Technische Universität München

TUM School of Computation, Information and Technology

Beyond Unique Decoding in the Sum-Rank Metric for Quantum-Resistant Cryptography

Thomas Jerkovits

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:	Prof. Dr.-Ing. Georg Sigl
Prüfende der Dissertation:	1. Prof. Dr.-Ing. Antonia Wachter-Zeh
	2. Assoc. Prof. Umberto Martínez-Peñas, Ph.D.

Die Dissertation wurde am 7.11.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 25.03.2025 angenommen.

To 姝姝

Acknowledgements

Completing this doctoral thesis has been a remarkable journey, filled with challenges, learning experiences, and moments of growth, both academically and personally. As I look back at the work that led to this point, I am immensely grateful to those who have helped me navigate through it all.

This journey took place at the German Aerospace Center (DLR), Institute of Communication and Navigation, in collaboration with the Technical University of Munich (TUM). I would like to take this opportunity to express my deepest gratitude to all the people from these institutions and beyond who played a crucial role in this experience.

First and foremost, I would like to sincerely thank my supervisor, Antonia Wachter-Zeh, and my mentor, Hannes Bartz, for offering me this opportunity and for their invaluable guidance and fruitful discussions. I am especially grateful to Hannes for always being available to answer my questions and for allowing me to openly discuss anything on my mind, whether it was related to academic topics or beyond.

A special thanks goes to Gianluigi Liva and Balázs Matuz for welcoming me to DLR and offering immense help during my early years. It was through them that I first came to DLR for my Master's thesis, and they were the ones who inspired me to embark on this doctoral path. Their expertise and vast knowledge in coding theory left a lasting impression on me, motivating me to continually strive for deeper understanding and to remain curious about new ideas.

I had the great fortune to collaborate with many talented researchers whose insights and expertise were invaluable to me and my work. I am especially grateful to my co-authors and collaborators, including Hannes Bartz, Antonia Wachter-Zeh, Felicitas Hörmann, Hugo Sauerbier Couvée, Jessica Bariffi, Julian Renner, Sven Puchinger, Vladimir Sidorenko, Gianluigi Liva, Balázs Matuz, Gerhard Kramer, Johan Rosenkilde, Pierre Loidreau, Mustafa Cemil Coşkun, Onur Günlü, Alexandre Graell i Amat, Giacomo Ricciutelli, Tudor Ninacs, Lorenzo Gaudio, and Hedongliang Liu.

My sincere appreciation goes to Umberto Martínez-Peñas for his interest in my work and for agreeing to be a reviewer of this dissertation.

To my colleagues in the Quantum-Resistant Cryptography (QRC) group at DLR—Felicitas, Svenja, Conny, Jessica, and Anna. Thank you for the fun team events, from escape rooms to dinners and bowling nights. Your team spirit and shared enthusiasm made this experience much more enjoyable and memorable. Special thanks to Svenja and Conny for proofreading parts of this thesis.

I would also like to acknowledge the seasoned veterans, Francisco Lázaro and Federico Clazzer from KN-SAN for the insightful discussions along the way, both on and off topic. Special thanks to Sandro Scalise for making it possible for me to work on this thesis.

I am also grateful to the fellow researchers and doctoral candidates at TUM for the many engaging workshops and events that greatly enriched my academic experience.

Lastly, I would like to thank everyone at DLR who brightened my days during the final months of this work, providing both support and a positive environment. Whether it was through shared lunches, extended coffee breaks, or simply their regular presence. They contributed more than they might realize. Special thanks to Davide, Riccardo, Manuel, Marcel, Alexander S., Alexander F., Benni, Stefan, Roshith, Umut, Pedro, Purva, my office mates Estefania and Stefano, and everyone else who made the everyday moments so enjoyable.

Finally, I owe my deepest gratitude to my family: my mother Gudrun, my father Willi, my sister Susi, my brother-in-law Roger, and especially my wife Shu. Shu's constant encouragement, endless patience, and unwavering support throughout my doctoral studies were pivotal in helping me persevere and reach the finish line. Her belief in me has been a source of strength during the most challenging moments of this journey.

Zusammenfassung

Die Fortschritte im Bereich der Quantencomputer bedrohen klassische kryptografische Systeme und unterstreichen die Notwendigkeit quantenresistenter Alternativen. Die codierungsbasierte Kryptografie mit ihrer robusten Sicherheitsgrundlage gilt hierbei als vielversprechender Ansatz. Insbesondere die Summenrangmetrik, welche sowohl die Hamming- als auch die Rangmetrik verallgemeinert, eröffnet ein interessantes Forschungsfeld.

In dieser Arbeit wird das Potenzial von Codes in der Summenrangmetrik für kryptografische Anwendungen untersucht. Dabei liegt ein besonderer Fokus auf der Weiterentwicklung von Dekodieralgorithmen und dem Dekodieren jenseits des eindeutigen Dekodierbereichs.

Zunächst wird ein Augenmerk auf die linearisierten Reed–Solomon (LRS)-Codes in der Summenrangmetrik gelegt, welche sowohl die Reed–Solomon-Codes bezüglich der Hamming-Metrik als auch die Gabidulin-Codes bezüglich der Rangmetrik verallgemeinern. Ein schneller Kötter–Nielsen–Høholdt-Interpolationsalgorithmus über Schiefpolynomringe für interleaved LRS-Codes wird vorgestellt. Dieser Algorithmus erreicht die beste bekannte asymptotische Komplexität und kommt dabei ohne Vorverarbeitung und spezielle Anforderungen an die Interpolationspunkte aus.

Des Weiteren werden Gabidulin-Codes untersucht, die durch schwach selbstorthogonale Basen definiert sind. Dabei werden raumsymmetrische Fehler analysiert, bei denen die Zeilen- und Spaltenräume der Fehlermatrix übereinstimmen. Es zeigt sich, dass das Dekodieren solcher Fehler mit hoher Wahrscheinlichkeit auch über den eindeutigen Dekodierbereich hinaus möglich ist.

Zusätzlich wird ein generischer Dekodieralgorithmus erforscht, der für die Kryptanalyse von Verfahren in der Summenrangmetrik nützlich ist. Durch eine Verallgemeinerung des Metzner–Kapturowski-Algorithmus von der Rang- und Hamming-Metrik auf die Summenrangmetrik wird ein Dekodierer mit polynomieller Laufzeit für hochgradig interleaved Summenrangmetrikcodes vorgestellt. Der vorgeschlagene Dekodierer ist auf beliebige lineare Komponentencodes anwendbar, einschließlich solcher ohne bekannte Struktur. Es zeigt sich, dass der Dekodierer bei ausreichend großer Interleaving-Ordnung stets bis knapp unterhalb der Minimaldistanz des Codes und mit hoher Wahrscheinlichkeit bis zur Singleton-artigen Schranke dekodieren kann. Durch diese Dekodiermethode wird jedoch eine Begrenzung der Interleaving-Ordnung erforderlich, um Sicherheitslücken in codierungsbasierten Kryptosystemen mit hoher Interleaving-Ordnung zu vermeiden.

Zum Schluss wird das Verständnis eines generischen “Support-Guessing”-Dekodierverfahrens für nicht-interleaved Summenrangmetrikcodes durch eine Analyse der durchschnittlichen Komplexität vertieft. Mithilfe von “Random Coding”-Argumenten werden genauere Schranken für das Dekodieren über den eindeutigen Dekodierbereich hinaus abgeleitet. Der “Support-Guessing”-Dekodieralgorithmus für Gabidulin-Codes in der Rangmetrik wird zudem auf LRS-Codes in der Summenrangmetrik verallgemeinert. Der Algorithmus nutzt dabei einen zugrunde liegenden Dekodierer, der sowohl Fehler als auch Auslöschungen dekodieren kann. Diese Anpassung reduziert die Dekodierkomplexität erheblich im Vergleich zu generischen Dekodierern, welche die Codestruktur nicht ausnutzen.

Mit diesen Beiträgen wird die Weiterentwicklung von Codes in der Summenrangmetrik gefördert, um die Praxistauglichkeit und Sicherheit zukünftiger quantenresistenter, codierungsbasierter Kryptosysteme zu erhöhen.

Abstract

The advent of quantum computing threatens classical cryptographic systems, highlighting the need for quantum-resistant alternatives. Code-based cryptography, with its strong security foundations, is a promising candidate. In particular, the sum-rank metric, generalizing both Hamming and rank metric, offers a potential avenue for exploration.

This work explores the potential of sum-rank metric codes in cryptography by advancing decoding algorithms with a focus on decoding beyond the unique radius.

We first focus on Linearized Reed–Solomon (LRS) codes in the sum-rank metric, which generalize Reed–Solomon (Hamming metric) and Gabidulin codes (rank metric). We present a fast skew Kötter–Nielsen–Høholdt interpolation algorithm for interleaved LRS codes. This algorithm matches the best-known asymptotic complexity while eliminating the need for pre-processing and specific interpolation point requirements.

Further, focusing on Gabidulin codes defined by weak self-orthogonal bases, we investigate space-symmetric errors, where the row and column spaces of the error matrix coincide. We show that decoding beyond the unique decoding radius for such errors is possible with high probability.

We also explore generic decoding algorithms useful for cryptanalysis of sum-rank metric schemes. By generalizing the Metzner–Kapturowski algorithm from rank and Hamming metric to the sum-rank metric, we introduce a polynomial-time decoder for high-order interleaved sum-rank metric codes. This decoder applies to any linear constituent code, including those without a known structure. We show that when the interleaving order is sufficiently large, our decoder can always decode up to just below the minimum distance of the code and up to the Singleton-like bound with high probability. Due to this decoder, it is necessary to limit the interleaving order to prevent vulnerabilities in code-based cryptosystems with high interleaving orders.

Finally, we improve the understanding of generic support-guessing decoding for non-interleaved sum-rank metric codes by considering an average-case complexity analysis. Using random coding arguments, we derive tighter bounds for beyond unique decoding. Furthermore, we adapt the support-guessing decoding algorithm for Gabidulin codes, utilizing an underlying error-and-erasure decoder, to LRS codes. This significantly reduces the decoding complexity compared to generic decoders that do not utilize the code structure.

These contributions advance sum-rank metric codes, aiming to enhance the practicality and security of future quantum-resistant code-based cryptosystems.

Contents

1 Introduction **1**
- 1.1 The Need for Post-Quantum Cryptography . . . 1
- 1.2 Code-Based Cryptosystems . . . 5
- 1.3 Motivation . . . 7
- 1.4 Contributions and Outline . . . 10

2 Preliminaries **13**
- 2.1 Notation . . . 13
  - 2.1.1 Sets, Vectors and Matrices . . . 13
  - 2.1.2 Finite Fields and Bases . . . 15
- 2.2 Row and Column Spaces . . . 16
- 2.3 Probabilities of Subspace Relationships . . . 17
- 2.4 Linear Block Codes over Finite Fields . . . 18
  - 2.4.1 Generator Matrix and Parity Check Matrix . . . 19
  - 2.4.2 Distance Properties of Linear Block Codes . . . 19
  - 2.4.3 Interleaved Codes . . . 23
- 2.5 Polynomials over Finite Fields . . . 25
  - 2.5.1 Conjugacy Class . . . 26
  - 2.5.2 Skew Polynomials . . . 26
  - 2.5.3 Generalized Operator Evaluation . . . 28
  - 2.5.4 Generalized Moore Matrix . . . 29
- 2.6 Codes in the Sum-Rank Metric . . . 31
  - 2.6.1 The Sum-Rank Metric and its Properties . . . 32
  - 2.6.2 Interleaved Sum-Rank-Metric Codes . . . 35
  - 2.6.3 Channel Models . . . 36
  - 2.6.4 Row and Column Support in the Sum-Rank Metric . . . 37
  - 2.6.5 Linearized Reed–Solomon Codes . . . 38
  - 2.6.6 Interleaved Linearized–Reed Solomon Codes . . . 39
- 2.7 Remark on the Notation of Complexity . . . 40

3 Efficient Decoding of Interleaved Linearized Reed–Solomon Codes **43**
- 3.1 Known Decoding Approaches . . . 44
  - 3.1.1 Syndrome-Based Decoding . . . 44

3.1.2 The Loidreau–Overbeck Decoder . . . 45
3.1.3 Interpolation-Based Decoding of Interleaved LRS Codes . . . 47
3.2 Weak Popov and Gröbner Bases . . . 49
3.3 Skew Kötter–Nielsen–Høhold Interpolation over Skew Polynomial Rings 52
3.4 Fast Skew Kötter–Nielsen–Høhold Interpolation . . . 56
3.4.1 Divide-and-Conquer Skew Kötter Interpolation . . . 56
3.4.2 Precomputing Minimal-Polynomial Vectors . . . 60
3.4.3 Application to Interleaved Linearized Reed–Solomon Codes . . . 61
3.5 Summary and Discussion . . . 62

4 Decoding of Space-Symmetric Rank Errors 65
4.1 Gabidulin Codes Generated by Weak Self-Orthogonal Bases . . . 66
4.2 Space-Symmetric Channel Model . . . 68
4.3 Syndrome-Based Decoding Approach . . . 68
4.4 Probability of Decoding Failure . . . 71
4.5 Numerical Results . . . 73
4.6 Number of Space-Symmetric Matrices . . . 74
4.7 Application to Code-Based Cryptography . . . 75
4.8 Summary and Discussion . . . 77

5 Decoding of High-Order Interleaved Sum-Rank-Metric Codes 79
5.1 Problem Description . . . 81
5.2 Recovering the Error Support . . . 83
5.3 A Metzner–Kapturowski-like Decoding Algorithm for Sum-Rank-Metric Codes . . . 87
5.4 Probabilistic Decoding for Uniform Random Errors . . . 89
5.4.1 Main Theorem . . . 95
5.4.2 Numerical Results . . . 96
5.5 Decoding Radius . . . 98
5.5.1 Numerical Results . . . 101
5.6 Examples . . . 105
5.7 Special Cases of the Algorithm for Hamming and Rank Metric . . . 109
5.8 Connection to the Loidreau–Overbeck Decoder . . . 114
5.9 Summary and Discussion . . . 114

6 Support-Guessing Decoding Algorithms in the Sum-Rank Metric 117
6.1 Overview of Decoding Problems . . . 120
6.1.1 Sum-Rank Syndrome Decoding Problem . . . 121
6.1.2 Decoding Beyond the Unique Radius . . . 121
6.1.3 Unique Decoding Problem . . . 123
6.1.4 Channel Model . . . 123
6.2 Ordered Rank Profiles . . . 124

6.3 Generic Decoding in the Sum-Rank Metric . . . 125
6.3.1 Improved Simple Bound on the Worst-Case Success Probability 127
6.3.2 Success Probability Analysis for the Average Case . . . 131
6.3.3 Optimizing the Support-Drawing Distribution via Linear Programming . . . 135
6.3.4 Efficient Optimization of the Support-Drawing Distribution . . . 136
6.3.5 Numerical Results . . . 141
6.4 Generic Decoding for Large Error Weights . . . 145
6.5 Randomized Decoding of Linearized Reed–Solomon Codes . . . 149
6.5.1 Erasures in the Sum-Rank Metric . . . 150
6.5.2 Randomized Decoding Algorithm . . . 151
6.5.3 Worst-Case Complexity . . . 154
6.5.4 Average Complexity . . . 159
6.5.5 Optimizing the Support-Drawing Distribution . . . 160
6.5.6 Numerical Results . . . 161
6.5.7 Weak Keys in the Faure–Loidreau Cryptosystem . . . 163
6.6 Summary and Discussion . . . 164

7 Concluding Remarks **165**

A Proofs **167**

B Appendix of Chapter 6 **175**

C Notations, Variables, and Abbreviations **181**

Related Publications by the Author **189**

Bibliography **190**

1 Introduction

1.1 The Need for Post-Quantum Cryptography

Classical cryptography is broadly divided into two main types: symmetric-key cryptography and public-key cryptography, as illustrated in Figure 1.1.

Symmetric-key cryptography, also known as *secret-key cryptography*, relies on a single key for both encryption and decryption processes. This method is favored for tasks such as database encryption, file encryption, and securing communications within closed networks due to its efficiency and speed in handling large volumes of data. Common examples of symmetric-key algorithms are the Advanced Encryption Standard (AES) and the Data Encryption Standard (DES). However, one of its key challenges is the need for secure key exchange between the communicating parties, which must occur before any encrypted communication can take place.

To address this key exchange challenge, *public-key cryptography* was developed. It uses a pair of keys: a public key for encryption and a private key for decryption. This approach is crucial for securely exchanging keys over public channels where a private or secure key exchange is not feasible. It plays a vital role in internet applications, enabling secure connections through protocols such as TLS used in HTTPS, and in ensuring the authenticity of messages or documents through digital signatures. Public-key cryptography is also employed for email encryption, such as in PGP. Notable practical examples include Rivest–Shamir–Adleman (RSA), Diffie–Hellman Key Exchange, and Elliptic Curve Cryptography (ECC).

In many real-world applications, public-key and symmetric-key cryptography are combined in what is known as a hybrid cryptosystem. Public-key cryptography is crucial for securely exchanging symmetric keys through processes like key-encapsulation mechanism (KEM), after which the symmetric key is used for efficient encryption and decryption of bulk data. This approach leverages the strengths of both cryptographic methods: the security of public-key cryptography for key exchange and the speed of symmetric-key cryptography for data encryption. Additionally, public-key cryptogra-

phy is crucial for authentication, where mechanisms like digital signatures ensure the integrity and authenticity of communications. Such systems often rely on a public-key infrastructure (PKI), with a certificate authority (CA) issuing digital certificates to bind public keys to verified identities, preventing so-called man-in-the-middle attacks.

The effectiveness of any cryptographic system hinges on its ability to resist attacks, which is inherently tied to the computational difficulty of breaking it. This leads to the concept of the security level (SL) of a cryptographic system. The SL of a cryptographic system is defined as the base-2 logarithm of the computational effort required to break it using the most efficient known attack [MW18; Fed24]. This computational effort is also known as the work factor (WF). For example, if the best-known attack requires a WF of 2^{128}, the system is said to offer 128-bit security. Cryptographic systems are designed to ensure that breaking them is computationally infeasible, relying on the difficulty of solving certain mathematical problems and on the assumption that more efficient attacks do not exist.

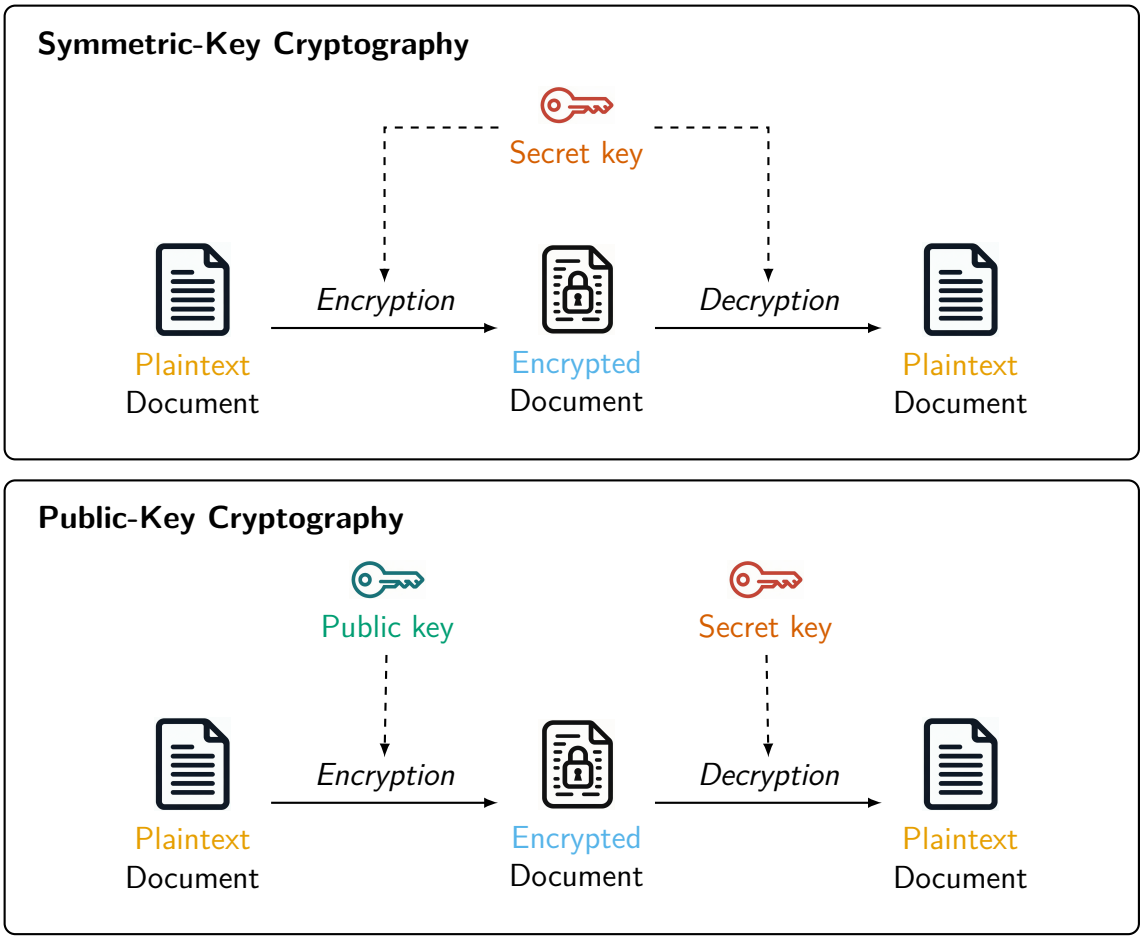


Figure 1.1: Illustration and comparison of symmetric-key cryptography and public-key cryptography.

However, advancements in computing technology, particularly the development of quantum computers, pose significant threats to the assumed SLs of current cryptographic systems. Quantum algorithms have the potential to drastically reduce the computational effort required to break certain cryptographic schemes, especially those used in public-key cryptography.

For example, Shor's algorithm can factorize large integers and compute discrete logarithms in polynomial time, effectively breaking the security of RSA, ECC, and other public-key schemes [Sho97; PZ03]. The existence of such a polynomial-time algorithm means that these cryptographic schemes are no longer considered secure in the presence of quantum computers. In cryptography, a problem that can be solved in polynomial time is considered tractable, making it unsuitable as a foundation for secure cryptographic schemes. Since the effort required to break these systems grows only polynomially with the key size, increasing the key size does not provide adequate security against quantum attacks.

Although current quantum computers are not yet powerful enough to factorize the large integers used in modern cryptographic applications, the potential for future advancements demonstrates the severity of this threat. For instance, IBM has successfully factorized the number 21 using a so-called "compiled" version of Shor's algorithm [ST21]. This approach leverages precomputed classical information and algorithmic simplifications to reduce the complexity of the quantum circuit, tailoring it specifically for factoring 21 and circumventing the full generality of Shor's algorithm. In 2012, a room-temperature adiabatic quantum computer was used to factor 143 [XZL+12], demonstrating that quantum annealing techniques like those implemented by D-Wave systems can also contribute to this progress. Additionally, Google's quantum supremacy experiment [AAB+19] demonstrated the ability of quantum computers to solve specific problems significantly faster than classical computers. Moreover, hybrid quantum-classical algorithms have already managed to factorize 48-bit numbers, indicating considerable progress toward practical quantum factorization [YTW+22]. While these results may not match Shor's algorithm in terms of scalability, they highlight the variety of quantum techniques being actively explored.

The security of symmetric-key cryptography is also at risk from quantum algorithms. Grover's algorithm [Gro96] offers a quadratic speedup for unstructured search problems, reducing the effective SL of symmetric-key systems such as AES and ChaCha20. Consequently, the key sizes of symmetric algorithms must be doubled to maintain the same level of security against quantum attacks. However, while the quadratic speedup of Grover's algorithm is theoretically significant, its practical impact is more limited. The overhead required for quantum error correction often outweighs the advantage offered by the speedup, making Grover's algorithm less practical on current or near-future quantum hardware [BMN+21; HHT23]. Additionally, classical systems can be highly parallelized, diminishing the relative benefit of Grover's quadratic speedup for realistic problem sizes [HHT23].

Nevertheless, the quadratic speedup provided by Grover's algorithm still poses a significant theoretical threat and must be accounted for when designing quantum-resistant cryptographic schemes. As a consequence, Grover's algorithm also impacts public-key cryptosystems and must be considered to derive quantum-secure SLs.

While it may take considerable time to develop quantum computers powerful enough to break modern cryptographic schemes, it is crucial to seek alternative solutions well in advance. This proactive approach is necessary due to the "store-now, decrypt-later" paradigm, where encrypted data can be stored today and decrypted once quantum computers become available. Moreover, the transition from research laboratories to widespread use of quantum computers in the near future could pose a significant threat to existing cryptographic systems, necessitating a paradigm shift in cryptographic design.

Recognizing the urgency of emerging quantum threats, the NIST initiated the post-quantum cryptography standardization process in 2016 [Moo16]. This effort aims to develop a diverse set of quantum-secure cryptographic schemes, including lattice-based, isogeny-based, multivariate-based, hash-based, and code-based cryptography. After a rigorous six-year evaluation, NIST [Nat22; Nat24a] selected four candidates for standardization, one KEM and three Digital Signature (DS) schemes, as shown in bold within Table 1.1.

Table 1.1: Quantum-Resistant Cryptographic Schemes Selected by NIST.

Algorithm	**Type**	**Based on**	**Status / Standardized as**
CRYSTALS-Kyber	**KEM**	**Lattices**	**ML-KEM [Nat24c]**
CRYSTALS-Dilithium	**DS**	**Lattices**	**ML-DSA [Nat24b]**
SPHINCS+	**DS**	**Hashes**	**SLH-DSA [Nat24d]**
FALCON	**DS**	**Lattices**	**Pending Standardization**
BIKE	KEM	Codes	4th Round
Classic McEliece	KEM	Codes	4th Round
HQC	KEM	Codes	4th Round
~~*SIKE*~~	~~KEM~~	~~Isogeny~~	Withdrawn

While four algorithms have been selected for standardization, NIST continues its evaluation into a fourth round. As indicated in Table 1.1, *BIKE*, *Classic McEliece*, and *HQC* are code-based cryptosystems under consideration in this round. The isogeny-based candidate *SIKE* was withdrawn from the process after vulnerabilities were discovered [Nat22].

The extensive evaluation period reflects the importance of ensuring that these new

cryptographic algorithms are secure against both classical and quantum attacks. Additionally, these algorithms require time to gain the "trust" of the research community, emphasizing the need to proactively develop and adopt quantum-secure solutions before quantum computers become a viable threat.

1.2 Code-Based Cryptosystems

The core concept of code-based cryptography is to leverage error-correcting codes to obscure a message's contents during "transmission". Traditionally, error-correcting codes serve to detect and correct bit errors that occur when messages are sent over unreliable channels. By tailoring the code to meet the channel's specific requirements, it's possible to set the number of bit errors the code can reliably correct. This process is illustrated in Figure 1.2.

In this example, the original message $\mathbf{m}$ is first transformed (encoded) into a codeword $\mathbf{c}$ corresponding to a particular code $\mathcal{C}$. Encoding adds redundancy, making the codeword longer than the message itself. The codeword $\mathbf{c}$ is then transmitted over the channel, where certain bits may be altered or "flipped" due to channel noise, resulting in a received, noisy version $\mathbf{y}$ of $\mathbf{c}$.

The decoder's job is to interpret this noisy version $\mathbf{y}$, mapping it back to the original message $\mathbf{m}$. Generally, the number of errors introduced by the channel is within the range the code can correct; otherwise, the decoder may fail. This capability to correct errors is a crucial feature that code-based cryptographic schemes exploit.

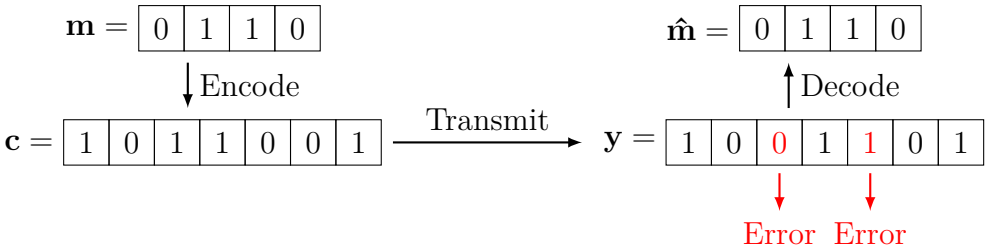


Figure 1.2: Illustration of error correction over an unreliable communication channel. The error-correcting code allows the receiver to identify and correct a specific number of bit errors introduced during transmission.

The McEliece cryptosystem, introduced by McEliece in 1978 [McE78], is one of the earliest public-key cryptographic schemes and remains unbroken to this day. It has gained substantial trust in its resilience within the cryptographic community.

In the original McEliece cryptosystem, binary Goppa codes are utilized for their efficient decoding properties, capable of correcting errors up to a certain weight (Hamming weight). The encryption process works as follows. The plaintext message $\boldsymbol{m}$ is encoded using a public generator matrix $\boldsymbol{G}'$, which is a disguised version of the original

generator matrix $\boldsymbol{G}$ of the code $\mathcal{C}$. An error vector $\boldsymbol{e}$, whose Hamming weight is within the code's error-correcting capability, is added to the encoded message to produce the ciphertext $\boldsymbol{y}$

$$\boldsymbol{y} = \boldsymbol{m} \cdot \boldsymbol{G}' + \boldsymbol{e}.$$

This ensures that the legitimate receiver can recover the original message uniquely.

Key Generation: The key generation process involves creating the public and secret keys as follows. The legitimate user selects a generator matrix $\boldsymbol{G}$ of the code $\mathcal{C}$, which allows efficient decoding. They also choose a random non-singular scrambling matrix $\boldsymbol{S}$ and a random permutation matrix $\boldsymbol{P}$. The public key is then computed as

$$\boldsymbol{G}' = \boldsymbol{S} \cdot \boldsymbol{G} \cdot \boldsymbol{P}.$$

The secret key consists of the matrices $\boldsymbol{S}$ and $\boldsymbol{P}$, along with the knowledge of the code $\mathcal{C}$ and its efficient decoding algorithm corresponding to $\boldsymbol{G}$.

Decryption: The decryption process relies on the secret key to reverse the transformations applied during encryption:

1. **Invert the permutation**: Multiply the received ciphertext $\boldsymbol{y}$ by $\boldsymbol{P}^{-1}$

   $$\boldsymbol{y}' = \boldsymbol{y} \cdot \boldsymbol{P}^{-1} = \boldsymbol{m} \cdot \boldsymbol{S} \cdot \boldsymbol{G} + \boldsymbol{e}',$$

   where $\boldsymbol{e}' = \boldsymbol{e} \cdot \boldsymbol{P}^{-1}$. Since $\boldsymbol{P}^{-1}$ is a permutation matrix, the Hamming weight of $\boldsymbol{e}'$ remains unchanged, ensuring it is still within the code's error-correcting capability.

2. **Decode the codeword**: Apply the efficient decoding algorithm of $\mathcal{C}$ to $\boldsymbol{y}'$ to correct $\boldsymbol{e}'$ and recover $\boldsymbol{m} \cdot \boldsymbol{S}$.

3. **Invert the scrambling**: Multiply by $\boldsymbol{S}^{-1}$ to obtain the original message

   $$\hat{\boldsymbol{m}} = \boldsymbol{m} = (\boldsymbol{m} \cdot \boldsymbol{S}) \cdot \boldsymbol{S}^{-1}.$$

An adversary, lacking knowledge of $\boldsymbol{S}$ and $\boldsymbol{P}$, cannot perform these steps efficiently. They are confronted with the problem of decoding a random linear code within its designed error-correcting capability, which is believed to be computationally hard. While Berlekamp et al. [BMV78] showed that the general decoding problem is NP-complete, it remains an open question whether bounded-distance decoding up to the unique decoding radius is NP-complete for random linear codes. However, no efficient algorithms are known for this problem, and it is widely assumed to be intractable for sufficiently large code parameters.

The public key $\boldsymbol{G}' = \boldsymbol{S} \cdot \boldsymbol{G} \cdot \boldsymbol{P}$ conceals the structure of the original code, making it appear as a random code to an adversary. This obfuscation prevents efficient decoding

without the secret key, as the adversary cannot exploit any specific code structure to facilitate decoding.

Variations of the McEliece cryptosystem exist, such as the Niederreiter variant, which uses a parity-check matrix for encryption. In this discussion, we focus on the original representation using the generator matrix.

Figure 1.3 illustrates the encryption and decryption process, including key generation, where the public key is derived by scrambling and permuting the original generator matrix.

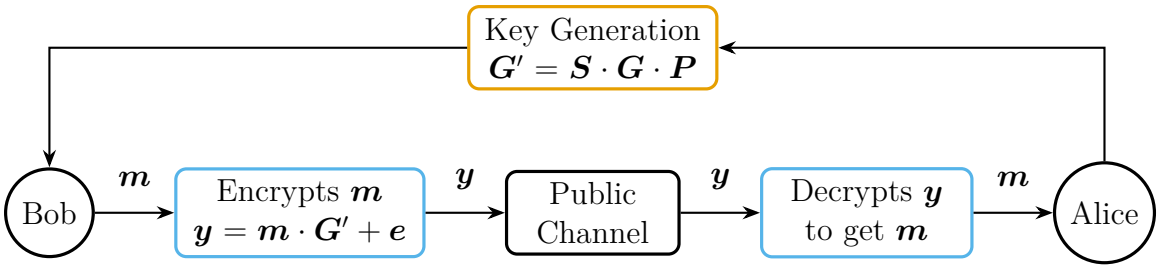


Figure 1.3: Illustration of the McEliece cryptosystem with key generation.

1.3 Motivation

Although the McEliece cryptosystem has many merits, it also suffers from certain drawbacks. One of the most significant challenges is the large key size, particularly the public key. The size of the public key is determined by the generator matrix, which, after the scrambling process, loses its structured form. As a result, the public key must be stored as a large unstructured matrix, making it difficult to represent efficiently.

Numerous attempts have been made to decrease the key size of the McEliece cryptosystem by modifying specific aspects. The original proposal by McEliece used Goppa codes [McE78]. Using error-correcting codes with better error-correcting capabilities than Goppa codes allows for smaller code parameters, resulting in smaller generator matrices and, consequently, reduced public-key sizes. However, such codes often introduce more structure, making them vulnerable to algebraic attacks. For instance, in 1986, Niederreiter suggested the use of generalized Reed–Solomon (GRS) codes [Nie86], but this scheme was later broken by Sidelnikov and Shestakov [SS92].

Further modifications to McEliece schemes based on algebraic codes, along with effective attack strategies, are detailed in [BL05; Wie10; Sid94; MS07; BCGO09; FOP+16; JM96; CMP15; Wan16; CLT19].

Sum-Rank Metric

An alternative approach to reduce the key size is to use codes with distance properties defined over metrics other than the Hamming metric. One of the first such schemes using alternative metrics, in particular the rank metric, was proposed by Gabidulin, Paramonov, and Trejakov [GPT91b], employing Gabidulin codes.

The rank metric is particularly attractive for cryptographic applications because, for the same SL, codes in the rank metric can utilize smaller code parameters than those in the Hamming metric, potentially resulting in smaller key sizes [Loi16]. This advantage arises from the increased complexity of generic decoding in the rank metric, where the best-known attacks, such as rank syndrome decoding, generally require significantly more computational effort than their Hamming-metric counterparts [GRS16]. As highlighted in [Loi16], generic decoding in the rank metric is exponentially more difficult than in the Hamming metric for the same set of parameters. This is due to the fact that errors are measured by the rank of a matrix, introducing dependencies across matrix entries and resulting in a more complex algebraic structure. This structural complexity, combined with the fact that attacks on rank-metric codes often involve operations over matrices rather than vectors, leads to higher decoding complexity. In particular, algorithms that rely on techniques such as information-set decoding (ISD) for Hamming-metric codes [Pra62; Ste89] do not directly apply to rank-metric codes without substantial increases in computational cost, further contributing to the security benefits of rank-metric cryptosystems.

However, systems using the rank metric, such as the Gabidulin-based cryptosystems, have been subject to several structural attacks [Gib95; Gib96; Ove05; Ove06; Ove08; HMR16; OKN18; HMR18], leading to multiple rounds of repairs and improvements [GO01; GOHA03; Loi10; RGH11; Gab08; GRH09; RGH10].

To address the trade-off between key size and security, the sum-rank metric presents a promising alternative. In the sum-rank metric, vectors are divided into several blocks. The sum-rank weight of a vector is computed by calculating the rank weight of each block separately and then summing these ranks. By adjusting the number of blocks, the sum-rank weight allows a smooth transition between the rank weight (if the vector is treated as one single block) and the Hamming weight (if each block has only a single column). This relationship between the sum-rank, rank, and Hamming metric is depicted in Figure 1.4, where ℓ is the number of blocks and n the code length.

By utilizing the sum-rank metric, it is possible to construct codes that retain strong error-correcting properties while offering the potential for smaller key sizes. Moreover, this metric may help avoid some of the structural vulnerabilities that have affected previous rank-metric-based systems, providing a promising avenue for further exploration in code-based cryptography [HBH23].

The sum-rank metric offers a potential solution to the trade-off between key size and security. Linearized Reed–Solomon (LRS) codes [Mar18], which are the sum-rank-metric analogue of Gabidulin codes in the rank metric and Reed–Solomon (RS) codes in

the Hamming metric, provide a promising class of codes within this framework. While cryptosystems based on the rank metric, such as Gabidulin codes, have been vulnerable to structural attacks like those proposed by Overbeck [Ove05; Ove06; Ove08], adapting these attacks to sum-rank-metric codes has proven more difficult. In [HBH23], it was demonstrated that generalizing Overbeck's attacks to the sum-rank metric for LRS codes requires prior knowledge of specific code parameters, suggesting that sum-rank-metric codes may provide more resilience against such vulnerabilities. Additionally, the sum-rank metric retains a higher generic decoding complexity compared to the Hamming metric, particularly for $\ell < n$, as analyzed in Chapter 6.

It is hoped that cryptosystems based on sum-rank-metric codes, such as LRS codes, will strike a favorable balance between security and efficiency. By positioning themselves between the Hamming and rank metrics, these codes may leverage the higher generic decoding complexity associated with the rank metric, potentially allowing for smaller key sizes than Hamming-metric systems, while remaining less susceptible to structural vulnerabilities. Though this balance appears promising, further analysis is necessary to fully assess the security and practical benefits of sum-rank-metric codes.

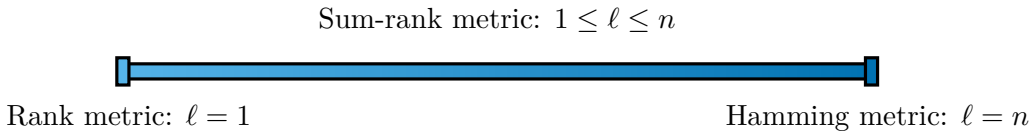


Figure 1.4: Illustration of the relation between the rank metric, Hamming metric, and sum-rank metric.

Interleaved Codes

The process of interleaving combines multiple codewords from the same underlying code, the so-called constituent code, to form a larger structure, allowing for improved error correction. Technical details on how interleaved codes are defined are provided in Section 2.4.3.

This technique has been proposed as an effective approach to mitigate the key size issue in McEliece cryptosystem variants based on both Hamming and rank metrics [EWZ18; HLPW19; RPW19]. Interleaving allows for the reuse of the constituent code's structure, which determines the public key size, while enabling the decoding of higher-weight errors compared to a single instance of the code. This increases the complexity of attacks, resulting in smaller public keys for the same SL.

The interleaving order is a critical parameter in this context, as it influences both the decoding process and the overall performance of the cryptosystem. While higher interleaving orders generally increase the complexity of attacks and thus enhance security, they can also introduce additional structure that attackers might exploit (see, e.g., Chapter 5).

Various decoding strategies have been developed for interleaved codes across different metrics. For example, list and probabilistic unique decoders exist for interleaved Reed–Solomon (IRS) codes in the Hamming metric [KL97], interleaved Gabidulin codes in the rank metric [Loi06], and interleaved linearized Reed–Solomon (ILRS) codes in the sum-rank metric [BP22]. These decoders are specifically tailored to each code family, utilizing the structure inherent to the interleaved codes.

Interleaved codes and their associated decoders thus present a promising direction for reducing key sizes in McEliece-type cryptosystems while maintaining the SL. By carefully selecting the interleaving order and the corresponding decoding strategy, it might be possible to optimize the balance between performance and resilience against attacks.

1.4 Contributions and Outline

This thesis makes several contributions to the field of decoding algorithms for codes in the sum-rank metric, with a particular focus on advancing the efficiency of decoding methods and analyzing their complexities. The key contributions are outlined below, with corresponding chapters that elaborate on each topic.

To establish the necessary mathematical background, Chapter 2 introduces key concepts such as linear block codes, polynomials, and the sum-rank metric. It also covers LRS codes, interleaved codes, and includes a remark on the complexity notation used in the thesis. A summary of notation and abbreviations is provided in Appendix C.

Chapter 3 revisits established decoding concepts for ILRS codes. We present a fast variant of the skew Kötter–Nielsen–Høholdt interpolation algorithm, which matches the best-known asymptotic complexity for interpolation-based decoding. Notably, our algorithm eliminates the need for pre-processing of the interpolation points as well as any specific requirements on them. The contribution of this chapter builds on work from [BJR24; BJPR19; BJPR21].

Chapter 4 investigates the decoding of *space-symmetric* rank-metric errors using Gabidulin codes. By restricting errors to those whose row and column spaces coincide, we demonstrate that Gabidulin codes can successfully decode such errors with rank up to $\frac{2(n-k)}{3}$ with high probability, where n is the code length and k is the code dimension. This restriction to space-symmetric errors allows for the decoding of errors with larger weights, which in turn enables the use of smaller code parameters and, consequently, smaller public-key sizes. While we do not propose a specific cryptosystem, this approach has the potential to enhance the practicality of rank-metric cryptosystems by reducing key sizes. This chapter is based on our prior work in [JSW21].

In Chapter 5, we introduce a novel approach for decoding high-order interleaved sum-rank metric codes by extending the Metzner–Kapturowski algorithm to the sum-rank metric. The proposed decoder is able to correct errors with a sum-rank weight up to $d_{\min} - 2$, where $d_{\min}$ is the minimum distance of the code. Moreover, with high

probability, it can decode errors with weight up to $n - k$. This decoder operates in polynomial time for any linear constituent code, including unstructured ones. An important takeaway from this chapter is the need for careful selection of the interleaving order when designing cryptosystems based on interleaved codes. If the interleaving order is too large, an attacker can exploit the decoding approach described here to decode without needing knowledge of the code structure, posing a security risk. This chapter builds on our previous work in [JHB23] and [JHB24].

In Chapter 6, we advance the analysis of decoding problems in the sum-rank metric by transitioning from worst-case to average-case complexity. We provide a detailed analysis of support-guessing algorithms, including the adaptation of a randomized decoding algorithm originally developed for Gabidulin codes. This chapter also includes a new heuristic approach for optimizing the support-drawing distribution to minimize decoding complexity. These generic decoding algorithms are essential for analyzing the security of future cryptosystems based on sum-rank metric codes. By better understanding the decoding complexity, one can derive cryptographic parameters that ensure both security and efficiency. The chapter draws from works including [JB19; RJB+20; JBW23; JBW24; CJB24].

Complementary Research Contributions

In addition to the contributions directly related to the thesis, further research was conducted in other areas during this period. For instance, in [JLG18], we introduced energy shaping techniques to enhance the iterative decoding threshold of tailbiting spatially coupled low-density parity-check (LDPC) code ensembles over the additive white Gaussian noise (AWGN) channel. This approach optimizes the transmission energy to improve decoding performance without sacrificing the code rate.

Moreover, in [JGSK20], we developed a nested convolutional code construction for key agreement using biometric or physical identifiers. This construction achieves points on the key-leakage-storage region for long block lengths and offers a flexible alternative to nested polar codes, balancing performance and complexity.

In [RJB19], which the author of this dissertation co-authored, we proposed and analyzed an efficient decoding algorithm for horizontally interleaved low-rank parity-check (LRPC) codes. We derived upper bounds on both the decoding failure rate and the computational complexity of the algorithm. The results demonstrate that interleaving reduces the decoding failure rate exponentially with respect to the interleaving order, while the computational complexity increases only linearly.

Although these publications are not directly included in the thesis, they reflect complementary areas of study carried out during the course of this work.

2
Preliminaries

In this chapter, we establish the mathematical background and introduce the notation and concepts used throughout this thesis. The chapter provides essential foundations in several key areas necessary for understanding the decoding algorithms presented later. A summary of all notations and abbreviations is provided in Appendix C.

We begin with the notation and mathematical conventions in Section 2.1, covering sets, vectors, matrices, finite fields, and bases. These concepts lay a foundation for various linear algebraic operations and representations. Section 2.4 introduces linear block codes over finite fields, discussing their properties and the associated encoding and decoding processes. In Section 2.5, we review the properties of polynomials over finite fields, which are important for constructing and analyzing different types of codes and their decoding methods. Section 2.6 examines codes in the sum-rank metric, which offer flexibility and potential for efficient decoding. This section includes discussions on the sum-rank metric, LRS codes, interleaved codes, and their relationships to other metrics. Finally, Section 2.7 provides a remark on the notation of complexity, explaining the conventions used to describe the computational complexity of various decoding algorithms.

2.1 Notation

We begin with the fundamental concepts and notation related to sets, vectors, and matrices in Section 2.1.1, which are key for many operations throughout this thesis. Next, we discuss finite fields and bases in Section 2.1.2, which are crucial for understanding the algebraic structures involved in the decoding strategies analyzed.

2.1.1 Sets, Vectors and Matrices

A *set* is a collection of unique elements, represented as $\mathcal{S} = \{s_1, s_2, \ldots, s_r\}$. The cardinality of a set $\mathcal{S}$, denoted as $|\mathcal{S}|$, is the number of elements in $\mathcal{S}$. We use similar

calligraphic notation for *tuples*, which are ordered sequences of (possibly non-unique) items, written as $\mathcal{T} = (t_1, t_2, \ldots, t_r)$.

We define the set of nonnegative integers as $\mathbb{Z}_{\geq 0} \stackrel{\text{def}}{=} \{0, 1, 2, \ldots\}$. *Vectors* are represented by bold lowercase letters, and their elements are indexed starting from 1.

For example, a vector of length v is written as $\boldsymbol{a} = [a_1, a_2, \ldots, a_v]$.

Similarly, for a *matrix* $\boldsymbol{A}$ of size $v \times w$, the entry in the i-th row and j-th column, where $i \in \{1, \ldots, v\}$ and $j \in \{1, \ldots, w\}$, is denoted as $A_{i,j}$.

For given integers c, d, e, and f such that $1 \leq c \leq d \leq v$ and $1 \leq e \leq f \leq w$, we use the following notation to denote a *submatrix* of $\boldsymbol{A}$ by

$$\boldsymbol{A}_{[c:d],[e:f]} \coloneqq \begin{bmatrix} A_{c,e} & \cdots & A_{c,f} \\ \vdots & \ddots & \vdots \\ A_{d,e} & \cdots & A_{d,f} \end{bmatrix}.$$

This submatrix consists of the elements from rows c to d and columns e to f of the original matrix $\boldsymbol{A}$. For a given integer e and f such that $1 \leq e \leq f \leq w$, we use the following notation to denote a *submatrix* of $\boldsymbol{A} \in \mathbb{F}_q^{v \times w}$ consisting of the selected columns

$$\boldsymbol{A}_{[e:f]} \coloneqq \begin{bmatrix} A_{1,e} & \cdots & A_{1,f} \\ \vdots & \ddots & \vdots \\ A_{v,e} & \cdots & A_{v,f} \end{bmatrix},$$

where $\boldsymbol{A}_{[e:f]}$ is a submatrix of $\boldsymbol{A}$ formed by all rows but only columns e to f. The size of $\boldsymbol{A}_{[e:f]}$ is $v \times (f - e + 1)$.

The transpose of a matrix $\boldsymbol{A}$ of size $v \times w$ is denoted as $\boldsymbol{A}^\top$, resulting in a matrix of size $w \times v$. For a square matrix $\boldsymbol{A}$ of size $v \times v$, we denote as $\boldsymbol{A}^{-\top}$ the inverse of the transpose of $\boldsymbol{A}$, if it exists. This matrix, also known as the inverse transpose or transposed inverse, satisfies the property

$$\boldsymbol{A}^{-\top} = \left(\boldsymbol{A}^\top\right)^{-1} = \left(\boldsymbol{A}^{-1}\right)^\top.$$

Let $\boldsymbol{a}^{(1)} = [a_1^{(1)}, a_2^{(1)}, \ldots, a_{\ell_1}^{(1)}]$ be a vector of length ℓ_1 and $\boldsymbol{a}^{(2)} = [a_1^{(2)}, a_2^{(2)}, \ldots, a_{\ell_2}^{(2)}]$ be a vector of length ℓ_2. We define and denote the concatenation of $\boldsymbol{a}^{(1)}$ with $\boldsymbol{a}^{(2)}$ as

$$\left[\boldsymbol{a}^{(1)} \mid \boldsymbol{a}^{(2)}\right] \stackrel{\text{def}}{=} \Big[\underbrace{a_1^{(1)}, a_2^{(1)}, \ldots, a_{\ell_1}^{(1)}}_{\boldsymbol{a}^{(1)}}, \underbrace{a_1^{(2)}, a_2^{(2)}, \ldots, a_{\ell_2}^{(2)}}_{\boldsymbol{a}^{(2)}} \Big],$$

which is a vector of length $\ell_1 + \ell_2$.

Let $\mathcal{S}$ be a finite set. We use the notation $a \stackrel{\$}{\leftarrow} \mathcal{S}$ to denote an element $a \in \mathcal{S}$ drawn uniformly at random.

Consider $\mathcal{S}$ as a finite index set, and let $f : \mathcal{S} \to \mathbb{R}$ be a real-valued function defined

on $\mathcal{S}$. The *argmin* and *argmax* operators are defined as follows

$$\arg\min_{i\in\mathcal{S}} f(i) \stackrel{\text{def}}{=} \left\{ i \in \mathcal{S} : f(i) = \min_{j\in\mathcal{S}} f(j) \right\},$$

$$\arg\max_{i\in\mathcal{S}} f(i) \stackrel{\text{def}}{=} \left\{ i \in \mathcal{S} : f(i) = \max_{j\in\mathcal{S}} f(j) \right\}.$$

2.1.2 Finite Fields and Bases

This subsection provides the necessary notations and concepts related to finite fields. For a detailed study of finite fields, their properties, and applications, see e.g., [LN96].

Let p be a prime, then by $\mathbb{F}_p$ we denote a finite field with p elements. Let q be a power of a prime p, then $\mathbb{F}_q$ denotes a finite field of order q, and p is called the characteristic of $\mathbb{F}_q$. By $\mathbb{F}_{q^m}$ we denote an extension field of degree m, meaning that $\mathbb{F}_q$ is a subfield of $\mathbb{F}_{q^m}$, written as $\mathbb{F}_q \subseteq \mathbb{F}_{q^m}$. The multiplicative group of a finite field $\mathbb{F}$ excluding the zero element is denoted by $\mathbb{F}^*$ and defined as $\mathbb{F}^* \stackrel{\text{def}}{=} \mathbb{F} \setminus \{0\}$.

We denote the set of all $v \times w$ matrices over any finite field $\mathbb{F}$ as $\mathbb{F}^{v\times w}$. The general linear group $\mathrm{GL}_v(\mathbb{F})$ consists of all $v \times v$ invertible matrices over the finite field $\mathbb{F}$. Similarly, the set of all row vectors of length v over $\mathbb{F}$ is denoted by $\mathbb{F}^v$, and the set of all column vectors of length v over $\mathbb{F}$ is denoted by $\mathbb{F}^{v\times 1}$.

Let $\mathcal{B} = \{b_1, \ldots, b_m\} \subset \mathbb{F}_{q^m}$ be a fixed basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ and define the vector $\boldsymbol{b} = [b_1, \ldots, b_m] \in \mathbb{F}_{q^m}^m$ as the corresponding *ordered basis* of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. We denote by $\mathrm{ext}(a) \in \mathbb{F}_q^{m\times 1}$ the column-wise expansion of an element $a \in \mathbb{F}_{q^m}$ over $\mathbb{F}_q$ with respect to the basis $\boldsymbol{b}$, i.e.

$$\mathrm{ext} : \mathbb{F}_{q^m} \to \mathbb{F}_q^{m\times 1}, \tag{2.1}$$

such that $a = \boldsymbol{b} \cdot \mathrm{ext}(a)$. For a vector $\boldsymbol{a} = [a_1, \ldots, a_v] \in \mathbb{F}_{q^m}^v$, this notation is extended element-wise, resulting in

$$\mathrm{ext}(\boldsymbol{a}) = [\mathrm{ext}(a_1), \ldots, \mathrm{ext}(a_v)] \in \mathbb{F}_q^{m\times v}, \tag{2.2}$$

where $\mathrm{ext}(\boldsymbol{a})$ is a matrix with each column being $\mathrm{ext}(a_i) \in \mathbb{F}_q^{m\times 1}$ for $i \in \{1, \ldots, v\}$. Similarly, for a matrix $\boldsymbol{M} \in \mathbb{F}_{q^m}^{v\times w}$, the notation is extended element-wise as

$$\mathrm{ext}(\boldsymbol{M}) = \begin{bmatrix} \mathrm{ext}(M_{1,1}) & \cdots & \mathrm{ext}(M_{1,w}) \\ \vdots & \ddots & \vdots \\ \mathrm{ext}(M_{v,1}) & \cdots & \mathrm{ext}(M_{v,w}) \end{bmatrix} \in \mathbb{F}_q^{mv\times w},$$

where $\mathrm{ext}(\boldsymbol{M})$ is a matrix formed by replacing each element $M_{i,j}$ of $\boldsymbol{M}$ with its corresponding column-wise expansion $\mathrm{ext}(M_{i,j}) \in \mathbb{F}_q^{m\times 1}$, for all $i \in \{1, \ldots, v\}$ and $j \in \{1, \ldots, w\}$.

The $\mathbb{F}_q$-rank of a matrix $\boldsymbol{A} \in \mathbb{F}_q^{v\times w}$ is denoted as $\mathrm{rk_q}(\boldsymbol{A})$, and similarly, the $\mathbb{F}_{q^m}$

rank of a matrix over $\mathbb{F}_{q^m}$ is denoted by $\mathrm{rk_q}^{\mathrm{m}}(\cdot)$. Thus, we can define the $\mathbb{F}_q$-rank of a matrix $\boldsymbol{M} \in \mathbb{F}_{q^m}^{v\times w}$ as

$$\mathrm{rk_q}(\boldsymbol{M}) \stackrel{\mathrm{def}}{=} \mathrm{rk_q}(\mathrm{ext}(\boldsymbol{M})).$$

From this definition, it directly follows that for a vector $\boldsymbol{a} \in \mathbb{F}_{q^m}^{v}$, we have

$$\mathrm{rk_q}(\boldsymbol{a}) = \mathrm{rk_q}(\mathrm{ext}(\boldsymbol{a})). \tag{2.3}$$

Note that the $\mathbb{F}_q$-rank is independent of the choice of the basis $\boldsymbol{b}$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Consequently, it corresponds to the dimension of the $\mathbb{F}_q$-span of the entries of $\boldsymbol{a}$.

2.2 Row and Column Spaces

The *row space* of a matrix $\boldsymbol{A} \in \mathbb{F}_{q^m}^{v\times w}$ over the field $\mathbb{F}_{q^m}$ is the $\mathbb{F}_{q^m}$-linear space formed by all possible linear combinations of its rows with coefficients from $\mathbb{F}_{q^m}$. We denote this $\mathbb{F}_{q^m}$-linear row space by $\mathcal{R}_{q^m}(\boldsymbol{A})$.

In some contexts, we are interested in the $\mathbb{F}_q$-linear row space of a matrix $\boldsymbol{A} \in \mathbb{F}_{q^m}^{v\times w}$, where $\mathbb{F}_{q^m}$ is viewed as a vector space over $\mathbb{F}_q$. In this case, we first expand each element of $\boldsymbol{A}$ over the field $\mathbb{F}_q$ and then consider the $\mathbb{F}_q$-linear row space of the resulting matrix. We denote this space by $\mathcal{R}_q(\boldsymbol{A})$, which is formally defined as

$$\mathcal{R}_q(\boldsymbol{A}) \stackrel{\mathrm{def}}{=} \mathcal{R}_q(\mathrm{ext}(\boldsymbol{A})),$$

where $\mathrm{ext}(\boldsymbol{A})$ denotes the matrix $\boldsymbol{A}$ after expansion of its entries over $\mathbb{F}_q$.

The same notation applies to the *column space* of a matrix $\boldsymbol{A}$, using $\mathcal{C}_q(\cdot)$ instead of $\mathcal{R}_q(\cdot)$ (or $\mathcal{C}_{q^m}(\cdot)$ instead of $\mathcal{R}_{q^m}(\cdot)$). The column space is formed by the columns of $\boldsymbol{A}$.

Vector spaces are denoted, similarly to sets, by calligraphic letters. For non-negative integers a and b, the number of b-dimensional subspaces of $\mathbb{F}_q^a$ is given by the Gaussian binomial coefficient $\begin{bmatrix} a \\ b \end{bmatrix}_q$ (see [Ber84]), defined as

$$\begin{bmatrix} a \\ b \end{bmatrix}_q \stackrel{\mathrm{def}}{=} \prod_{i=1}^{b} \frac{q^{a-b+i}-1}{q^i-1} = \prod_{i=0}^{b-1} \frac{q^{a-i}-1}{q^{b-i}-1}. \tag{2.4}$$

The Gaussian binomial coefficient satisfies [KK08]

$$q^{(a-b)b} \leq \begin{bmatrix} a \\ b \end{bmatrix}_q \leq \gamma_q q^{(a-b)b},$$

where γ_q is defined as

$$\gamma_q \stackrel{\mathrm{def}}{=} \prod_{i=1}^{\infty} \left(1-q^{-i}\right)^{-1}. \tag{2.5}$$

Note that γ_q is a monotonically decreasing function of q with a limit of 1 as q approaches infinity. We can observe this behavior through the following example values:

$$\begin{aligned}\gamma_2 &\approx 3.4627,\\ \gamma_4 &\approx 3.4524,\\ \gamma_8 &\approx 1.1636,\\ \gamma_{16} &\approx 1.0711.\end{aligned}$$

Given a matrix $\boldsymbol{A} \in \mathbb{F}_{q^m}^{v\times w}$, the right $\mathbb{F}_q$-kernel is denoted and defined as

$$\ker\left(\boldsymbol{A}\right)_{\mathbb{F}_q} \stackrel{\text{def}}{=} \{\boldsymbol{w} \in \mathbb{F}_q^w : \boldsymbol{A}\boldsymbol{w}^\top = \boldsymbol{0}\},$$

and similarly the right $\mathbb{F}_{q^m}$-kernel is

$$\ker\left(\boldsymbol{A}\right)_{\mathbb{F}_{q^m}} \stackrel{\text{def}}{=} \{\boldsymbol{w} \in \mathbb{F}_{q^m}^w : \boldsymbol{A}\boldsymbol{w}^\top = \boldsymbol{0}\}.$$

Finally, we denote the set of all k-dimensional subspaces of $\mathbb{F}_q^v$ by the Grassmannian $\mathcal{G}_k(\mathbb{F}_q^v)$. For a vector space $\mathcal{V}$, we denote its dual space by $\mathcal{V}^\perp$. We use dim to represent the general dimension of a vector space, $\dim_{\mathrm{q}}$ for the dimension of an $\mathbb{F}_q$-linear vector space obtained by expanding a vector space over $\mathbb{F}_{q^m}$ as an $\mathbb{F}_q$-linear space using $\mathrm{ext}(\cdot)$, and $\dim_{\mathrm{q^m}}$ to indicate the dimension of an $\mathbb{F}_{q^m}$-linear vector space.

2.3 Probabilities of Subspace Relationships

In this section, we present expressions for the probability that two subspaces intersect in a fixed-dimensional space and the probability that one subspace is contained within another.

We use $[0,1] \subset \mathbb{R}$ to denote the interval of real numbers between 0 and 1, inclusive. We define the set of all valid probability mass functions (PMFs) over a discrete set $\mathcal{A}$ as

$$\mathcal{D}(\mathcal{A}) = \left\{ \alpha_{\boldsymbol{s}} \in [0,1]^{|\mathcal{A}|} \subset \mathbb{R}^{|\mathcal{A}|} : \sum_{\boldsymbol{s}\in\mathcal{A}} \alpha_{\boldsymbol{s}} = 1 \right\}. \tag{2.6}$$

Here, $\alpha_{\boldsymbol{s}} \in [0,1]^{|\mathcal{A}|}$ denotes a vector of length $|\mathcal{A}|$ with real values in the interval $[0,1]$. The condition $\sum_{\boldsymbol{s}\in\mathcal{A}} \alpha_{\boldsymbol{s}} = 1$ ensures that the vector represents a valid PMF.

Since each component $\alpha_{\boldsymbol{s}}$ can take any real value in $[0,1]$ and the sum of these components must be 1, the set $\mathcal{D}(\mathcal{A})$ forms a probability simplex in $\mathbb{R}^{|\mathcal{A}|}$. Therefore, there are uncountably many PMFs over $\mathcal{A}$.

Let $\mathcal{A}$ and $\mathcal{B}$ be two subspaces of $\mathbb{F}_q^\mu$ with dimensions a and b, respectively.

We define the conditional probability $\mathsf{P}^{\cap}_{q,\mu,a,b}(j)$ as the probability that the intersection of $\mathcal{A}$ and $\mathcal{B}$ has dimension exactly j, given their dimensions a and b. This

probability is given by (see [RJB+20])

$$\mathsf{P}^{\cap}_{q,\mu,a,b}(j) \stackrel{\text{def}}{=} \Pr[\dim(\mathcal{A}\cap\mathcal{B}) = j \mid a,b] = \frac{\begin{bmatrix}\mu-a\\ b-j\end{bmatrix}_q \begin{bmatrix}a\\ j\end{bmatrix}_q q^{(a-j)(b-j)}}{\begin{bmatrix}\mu\\ b\end{bmatrix}_q}. \tag{2.7}$$

Next, we define the probability that $\mathcal{A}$ is a subspace of $\mathcal{B}$, denoted by $\mathsf{P}^{\subseteq}_{q,\mu}(a,b)$, where $a \leq b$. This probability is given by [KK08]

$$\mathsf{P}^{\subseteq}_{q,\mu}(a,b) \stackrel{\text{def}}{=} \Pr[\mathcal{A}\subseteq\mathcal{B} \mid a,b] = \frac{\begin{bmatrix}b\\ a\end{bmatrix}_q}{\begin{bmatrix}\mu\\ a\end{bmatrix}_q}. \tag{2.8}$$

Remark 2.1. *The probabilities $\mathsf{P}^{\cap}_{q,\mu,a,b}(j)$ and $\mathsf{P}^{\subseteq}_{q,\mu}(a,b)$ hold whether $\mathcal{A}$ and $\mathcal{B}$ are both drawn uniformly at random from $\mathcal{G}_q(\mathbb{F}_q^\mu)$, or one of them is fixed and the other is drawn uniformly at random from $\mathcal{G}_q(\mathbb{F}_q^\mu)$.*

Remark 2.2. *The probability $\mathsf{P}^{\subseteq}_{q,\mu}(a,b)$ is equal to the intersection probability $\mathsf{P}^{\cap}_{q,\mu,a,b}(b)$ (or $\mathsf{P}^{\cap}_{q,\mu,a,b}(a)$) when the dimension of the intersection is equal to the dimension of the smaller subspace $\mathcal{A}$. That is,*

$$\mathsf{P}^{\subseteq}_{q,\mu}(a,b) = \mathsf{P}^{\cap}_{q,\mu,a,b}(b) = \frac{\begin{bmatrix}\mu-a\\ b-a\end{bmatrix}_q}{\begin{bmatrix}\mu\\ b\end{bmatrix}_q} = \frac{\begin{bmatrix}b\\ a\end{bmatrix}_q}{\begin{bmatrix}\mu\\ a\end{bmatrix}_q} \qquad \text{if } a \leq b,$$

$$\mathsf{P}^{\subseteq}_{q,\mu}(b,a) = \mathsf{P}^{\cap}_{q,\mu,a,b}(a) = \frac{\begin{bmatrix}a\\ b\end{bmatrix}_q}{\begin{bmatrix}\mu\\ b\end{bmatrix}_q} \qquad \text{if } b \leq a.$$

This relationship holds because $\mathcal{A}$ is a subspace of $\mathcal{B}$ if and only if the dimension of their intersection is the same as that of $\mathcal{A}$.

2.4 Linear Block Codes over Finite Fields

Block codes are a fundamental concept in coding theory, essential for error correction in data transmission and storage. They are used to detect and correct errors, ensuring the accuracy and reliability of data. Among the most popular block codes in the Hamming metric are linear block codes such as RS codes, Bose–Chaudhuri–Hocquenghem (BCH) codes, LDPC codes, turbo codes, and polar codes, each with unique properties and use cases.

Beyond traditional data transmission and storage, block codes are also crucial in various electronic communication systems, including mobile communication, wireless networks, and fiber-optic communications, where they help maintain data integrity

despite noise and signal degradation. They are also fundamental in code-based cryptography. Other applications include distributed storage systems, satellite communications, and deep-space communications, where error correction is critical.

For more detailed information on block codes, including their construction and applications, the reader is referred to [MS77; Moo05; Rot06; PW08; HJ17].

Definition 2.1 (Block Codes [Moo05])**.** *A* ***block code*** $\mathcal{C}$ *of length* n *and cardinality* $|\mathcal{C}|$ *over a finite field* $\mathbb{F}$ *is a set* $\mathcal{C} \subseteq \mathbb{F}^n$ *consisting of* $|\mathcal{C}|$ *vectors, called* ***codewords*** *and the set* $\mathcal{C}$ *is called the* ***codebook****.*

The *encoder* maps a *message* vector $\boldsymbol{m}$ to its corresponding codeword $\boldsymbol{c} \in \mathcal{C}$. For effective error correction, there must be a one-to-one correspondence between each message $\boldsymbol{m}$ and its codeword $\boldsymbol{c} \in \mathcal{C}$. While a block code can be represented as an exhaustive list, this becomes impractical for large $|\mathcal{C}|$. To reduce complexity, mathematical structure, particularly *linearity*, is often imposed on the code.

Definition 2.2 (Linear Block Codes [Moo05])**.** *A block code* $\mathcal{C}$ *over a finite field* $\mathbb{F}_{q^m}$ *of length* n *is a* q^m*-ary* ***linear block code*** *if and only if its* q^{mk} *codewords form a* k*-dimensional vector subspace of* $\mathbb{F}_{q^m}^n$*. The rate of the code is given by* $R = k/n$ *with*

$$k \stackrel{\text{def}}{=} \log_{q^m} |\mathcal{C}|.$$

2.4.1 Generator Matrix and Parity Check Matrix

Following Definition 2.2, the codebook of an $\mathbb{F}_{q^m}$-linear code can be entirely characterized by a basis for its k-dimensional subspace, consisting of k basis vectors from $\mathbb{F}_{q^m}^n$. The code is then defined as the set of all $\mathbb{F}_{q^m}$-linear combinations of these basis vectors. A matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k \times n}$, whose rows form a basis for $\mathcal{C}$, is known as a *generator matrix* for $\mathcal{C}$.

Another way to describe a linear code is by its *parity-check matrix* $\boldsymbol{H}$. The parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k) \times n}$ defines the code $\mathcal{C}$ as the null space of $\boldsymbol{H}$. In other words, $\mathcal{C}$ consists of all vectors $\boldsymbol{c} \in \mathbb{F}_{q^m}^n$ that satisfy $\boldsymbol{H}\boldsymbol{c}^\top = \boldsymbol{0}$. The rows of $\boldsymbol{H}$ represent a set of linear constraints that every codeword must satisfy, providing a convenient way to check whether a given vector is a valid codeword.

The generator matrix and the parity-check matrix are dual representations of a linear block code, i.e. they satisfy the condition $\boldsymbol{G}\boldsymbol{H}^\top = \boldsymbol{0}$.

2.4.2 Distance Properties of Linear Block Codes

As discussed in Chapter 1, codes can be defined over various metric spaces. To formally analyze the distance properties of codes, we first define the concept of a *metric*, which is a specific type of distance measure satisfying certain properties.

Definition 2.3 (Metric)**.** *Given a set $\mathcal{A}$, a mapping $d(\cdot,\cdot) : \mathcal{A} \times \mathcal{A} \to \mathbb{Z}_{\geq 0}$ is called a metric on $\mathcal{A}$ if it satisfies the following axioms for all $a, b, c \in \mathcal{A}$:*

- ***Non-negativity:*** $d(a,b) \geq 0$,
- ***Identity of indiscernibles:*** $d(a,b) = 0 \iff a = b$,
- ***Symmetry:*** $d(a,b) = d(b,a)$,
- ***Triangle inequality:*** $d(a,c) \leq d(a,b) + d(b,c)$.

In coding theory, the most widely used metric is the *Hamming metric*, which is a specific example of the general metric concept.

The Hamming weight $\mathrm{wt}_H(\boldsymbol{a})$ of a vector $\boldsymbol{a} \in \mathbb{F}_{q^m}^n$ is the number of nonzero entries in the vector. Formally, it is given by

$$\mathrm{wt}_H(\boldsymbol{a}) \stackrel{\text{def}}{=} |\{i \in \{1,\ldots,n\} : a_i \neq 0\}|, \tag{2.9}$$

where $\boldsymbol{a} = [a_1, a_2, \ldots, a_n]$.

The *Hamming distance* between two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_{q^m}^n$ is induced by the Hamming weight and is defined as

$$d_H(\boldsymbol{a}, \boldsymbol{b}) \stackrel{\text{def}}{=} \mathrm{wt}_H(\boldsymbol{a} - \boldsymbol{b}), \tag{2.10}$$

which represents the number of positions in which $\boldsymbol{a}$ and $\boldsymbol{b}$ differ.

Given a metric $d(\cdot,\cdot)$ on $\mathbb{F}_{q^m}^n$, the *minimum distance* of a code $\mathcal{C}$ is defined as

$$d_{\min}(\mathcal{C}) \stackrel{\text{def}}{=} \min_{\substack{\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathcal{C} \\ \boldsymbol{c}_1 \neq \boldsymbol{c}_2}} d(\boldsymbol{c}_1, \boldsymbol{c}_2).$$

This definition of minimum distance applies to both linear and non-linear codes. However, for a linear code, the minimum distance is equal to the minimum weight of its nonzero codewords. The weight of a codeword $\boldsymbol{c} \in \mathbb{F}_{q^m}^n$ is defined as

$$\mathrm{wt}(\boldsymbol{c}) \stackrel{\text{def}}{=} d(\boldsymbol{c}, \boldsymbol{0}),$$

where $\boldsymbol{0}$ is the zero vector.

For *translation-invariant metrics* where the distance $d(\cdot,\cdot)$ also satisfies the following property

$$d(\boldsymbol{a} + \boldsymbol{c}, \boldsymbol{b} + \boldsymbol{c}) = d(\boldsymbol{a}, \boldsymbol{b}),$$

for all $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{F}_{q^m}^n$, the minimum distance of a linear code $\mathcal{C}$ simplifies to

$$d_{\min}(\mathcal{C}) = \min_{\substack{\boldsymbol{c} \in \mathcal{C} \\ \boldsymbol{c} \neq \boldsymbol{0}}} \mathrm{wt}(\boldsymbol{c}).$$

While this work primarily focuses on the sum-rank metric, which we formally introduce in Section 2.6, it is worth noting that the Hamming metric is a special case of the sum-rank metric. All metrics considered in this thesis, namely the sum-rank metric as well as its special cases of the Hamming and rank metrics, are translation-invariant.

The Singleton bound for the Hamming metric states that for a linear code $\mathcal{C}$ with length n and dimension k, the minimum distance $d_{\min}(\mathcal{C})$ satisfies $d_{\min}(\mathcal{C}) \leq n-k+1$. When it is clear from the context which code is being discussed, we may simply write $d_{\min}$.

In the following, we denote a linear block code $\mathcal{C}$ over $\mathbb{F}_{q^m}$ with length n, dimension k, and minimum distance $d_{\min}$ as $\mathcal{C}[n, k, d_{\min}]$. If the minimum distance is not known, we denote it as $\mathcal{C}[n, k]$.

With the notion of minimum distance, we can now explore basic decoding principles. Consider a linear block code $\mathcal{C}[n, k, d_{\min}]$, and suppose we receive a noisy codeword $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$, which has been corrupted during transmission over a channel or through other means, so generally $\boldsymbol{y} \notin \mathcal{C}$. Our goal is to map $\boldsymbol{y}$ back to a codeword $\hat{\boldsymbol{c}} \in \mathcal{C}$, estimating the correct codeword from the input vector $\boldsymbol{y}$. This process is called *decoding*, and in the following, we discuss the most important decoding principles considered in this thesis.

Maximum Likelihood Decoding

In practice, especially in channel transmission, the objective is often to maximize the likelihood of the received data given a particular transmitted codeword, taking the channel distribution into account. This likelihood is derived from the channel's statistical model and is often referred to as *soft information*. The decoding principle based on maximizing this likelihood is called *maximum-likelihood (ML)* decoding. For certain channels, such as the binary symmetric channel (BSC), maximizing the likelihood is equivalent to minimizing the Hamming distance between $\boldsymbol{y}$ and any codeword $\boldsymbol{c} \in \mathcal{C}$. This distance-based approach is known as *nearest neighbor decoding*, and it applies to any metric, not just the Hamming metric.

For a given metric $d(\cdot, \cdot)$, the estimated codeword is given by

$$\hat{\boldsymbol{c}} \xleftarrow{\$} \arg\min_{\boldsymbol{c} \in \mathcal{C}} \{d(\boldsymbol{y}, \boldsymbol{c})\} .$$

In cases where there is a tie, and several valid codewords exist, the arg min function returns a set of solutions $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_z\}$ such that $d(\boldsymbol{y}, \boldsymbol{c}_1) = d(\boldsymbol{y}, \boldsymbol{c}_2) = \ldots = d(\boldsymbol{y}, \boldsymbol{c}_z)$ for some $z \in \mathbb{Z}_{\geq 0}$ with $z \geq 1$. The decoder then outputs one of these z solutions uniformly at random.

This approach ensures the decoder consistently returns a valid solution, thus functioning as a *complete decoder*. Unlike other decoders that may signal a decoding failure when the error weight exceeds a certain constraint, a complete decoder always returns a valid codeword as a solution.

The decoder implicitly divides the space $\mathbb{F}_{q^m}^n$ into distinct decoding regions, commonly referred to as *Voronoi* regions. An example illustrating this concept is provided in Figure 2.1a.

While theoretically optimal, this decoding principle often presents significant practical challenges. For instance, implementing it efficiently is proven to be NP-complete for a generic code in the Hamming metric [BMV78]. Due to this computational complexity, practical systems often employ suboptimal decoders that utilize channel likelihood information, such as LDPC codes, turbo codes, or polar codes. According to Shannon's theorem, random codes can achieve channel capacity as their block length approaches infinity [Sha48]. While these practical codes are not random, they are designed to approximate this behavior and can perform close to the theoretical limit, particularly for large code lengths and with limited decoding complexity. Although they do not achieve optimal performance in all cases, they often operate near ML performance, providing an effective trade-off between decoding complexity and error correction capability.

Bounded Minimum Distance Decoding

For coding settings where soft information is unavailable, another common decoding principle is *bounded minimum distance (BMD)* decoding. Let $w = d(\boldsymbol{c}, \boldsymbol{y})$ be the distance between the originally transmitted codeword $\boldsymbol{c}$ and the received noisy codeword $\boldsymbol{y}$. We define the *unique decoding radius* as

$$\tau \stackrel{\text{def}}{=} \frac{d_{\min} - 1}{2}.$$

If $w \leq \tau$, there exists always and at most one solution, which is the unique codeword $\boldsymbol{c}$. In this case *BMD* decoding concides with *optimal decoding*. However, if $w > \tau$, there might be no solution at all, or there could be a solution $\hat{\boldsymbol{c}} \neq \boldsymbol{c}$. This decoding principle is depicted in Figure 2.1b.

List Decoding and Probabilistic Unique Decoding

List decoding was first proposed by Elias [Eli57] and Wozencraft [Woz58]. For a given received vector $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$, a linear block code $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$, and a specified decoding radius $\tau_{\mathcal{L}} \in \mathbb{Z}_{\geq 0}$, a list decoder outputs a set $\mathcal{L} \subseteq \mathcal{C}$ of codewords defined as

$$\mathcal{L} \stackrel{\text{def}}{=} \{\boldsymbol{c} \in \mathcal{C} : d(\boldsymbol{y}, \boldsymbol{c}) \leq \tau_{\mathcal{L}}\}.$$

This set $\mathcal{L}$ contains all codewords within a distance of $\tau_{\mathcal{L}}$ from the received vector $\boldsymbol{y}$. The complexity of list decoding typically depends on the size of the list $|\mathcal{L}|$. Ideally, the list size is bounded by a function polynomial in the code length n to ensure feasible decoding. Larger lists can lead to exponential computational costs, so effective list

decoders strive to keep $|\mathcal{L}|$ manageable. In the extreme case where $|\mathcal{L}| = |\mathcal{C}|$, the list decoder returns the entire code. However, by selecting the most likely codeword from the list, the process is equivalent to ML decoding. In practical settings, the goal is to minimize the list size and choose the most likely solution, as large list sizes increase computational complexity without necessarily improving decoding performance.

List decoders can operate for errors within the unique decoding radius τ as well as beyond. For error weights w such that $w \leq \tau$ and a list decoding radius of $\tau_{\mathcal{L}} = \tau$, the list decoder coincides with the BMD decoder, and we always have $|\mathcal{L}| = 1$.

For $\tau_{\mathcal{L}} > \tau$ and any error weight w, we might still have $|\mathcal{L}| = 1$ on average over all possible received noisy codewords. If the probability of the event $|\mathcal{L}| = 1$ is very high, the list decoder effectively behaves like a unique decoder, even beyond the unique decoding radius. In such cases, we refer to such a decoder as a *probabilistic unique decoder*. That is a list decoder that outputs a valid codeword if and only if $|\mathcal{L}| = 1$, and declares a decoding failure otherwise.

The concept of list decoding is illustrated in Figure 2.1c. In Chapter 6, specifically Section 6.1, we revisit different decoding problem statements for randomized and support-guessing decoders, as applied to decoding problems relevant to some code-based cryptosystems.

2.4.3 Interleaved Codes

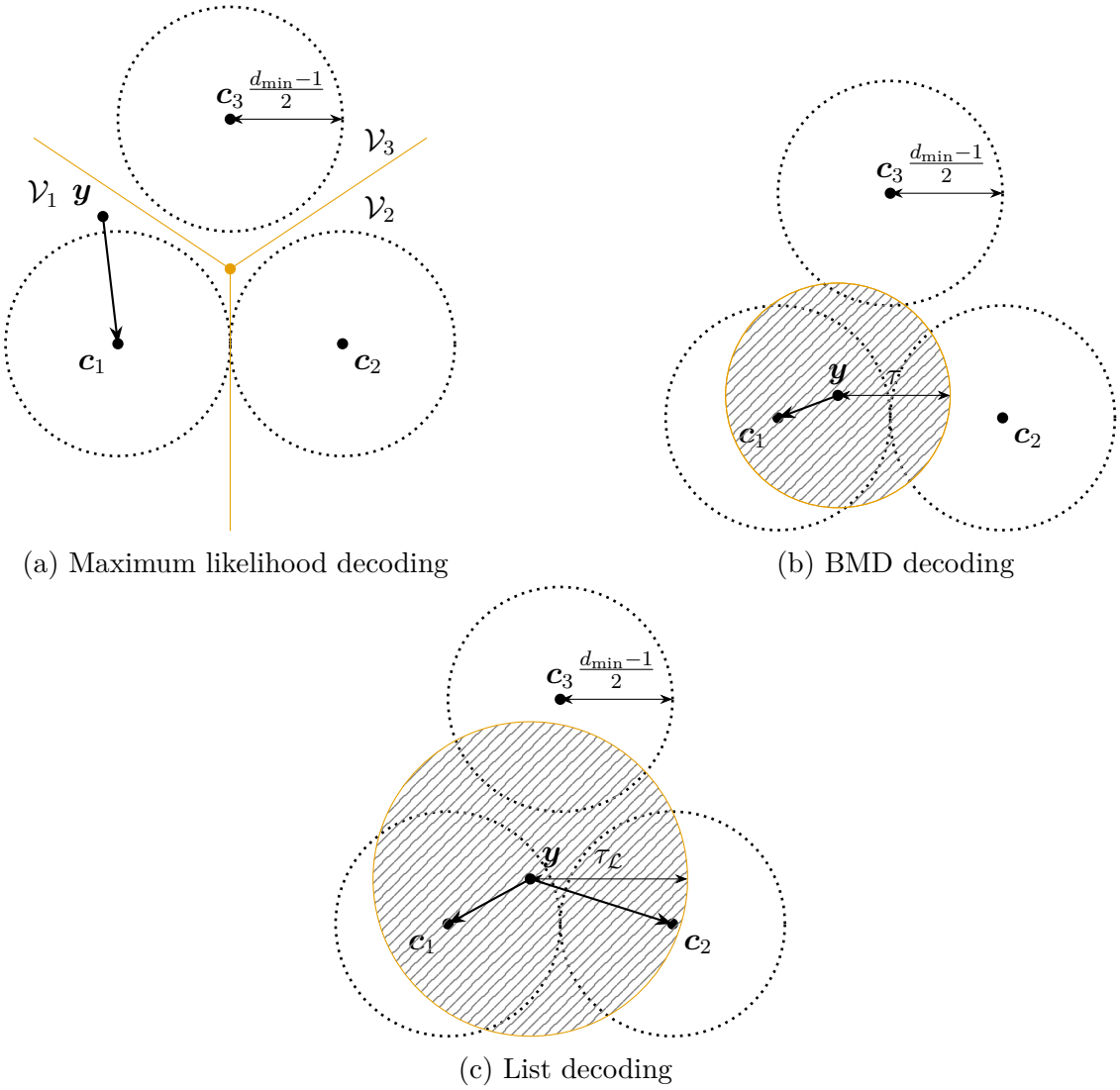
A vertically s-interleaved code is a direct sum of s codes, all having the same length n. The parameter s is referred to as the interleaving order. In general, interleaved codes can be formed by combining codewords from different codes that are subcodes of a common *supercode* (inhomogeneous interleaving). However, in this thesis, we focus on *homogeneous* interleaved codes, where codewords from the same constituent code are interleaved.

Definition 2.4 (Vertically Homogeneous Interleaved Code)**.** *Let $\mathcal{C}[n, k, d_{\min}] \subseteq \mathbb{F}_{q^m}^n$ be an $\mathbb{F}_{q^m}$-linear code of length n, dimension k, and minimum distance $d_{\min}$. The corresponding* ***homogeneous*** *s-interleaved code is defined as*

$$\mathcal{IC}[s; n, k, d_{\min}] \stackrel{\text{def}}{=} \left\{ \begin{bmatrix} \boldsymbol{c}_1 \\ \vdots \\ \boldsymbol{c}_s \end{bmatrix} : \boldsymbol{c}_j \in \mathcal{C}[n, k, d_{\min}],\ \forall j \in \{1, \ldots, s\} \right\} \subseteq \mathbb{F}_{q^m}^{s \times n},$$

where $\mathcal{C}[n, k, d_{\min}]$ is called the ***constituent code*** *of $\mathcal{IC}[s; n, k, d_{\min}]$.*

Interleaving does not increase the minimum distance of the overall interleaved code, as it is determined by the minimum distance of the constituent code. However, interleaving can significantly improve the decoding radius, allowing for the correction of errors beyond the unique decoding radius with efficient decoding algorithms, often



(a) Maximum likelihood decoding

(b) BMD decoding

(c) List decoding

Figure 2.1: Illustration of different decoding principles in the 2D plane. The dotted circles represent the unique decoding radius $\frac{d_{\min}-1}{2}$. τ is the decoding radius for BMD, while $\tau_{\mathcal{L}}$ is for list decoding. $\mathcal{V}_1$, $\mathcal{V}_2$, and $\mathcal{V}_3$ represent the Voronoi regions for codewords c_1, c_2, and c_3 respectively for ML decoding. In all cases, y represents the received word.

with a high probability of returning a unique solution. This improvement is due to the structure of the error patterns.

In particular for the vertically interleaved case, we consider channels of the form

$$\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E} \in \mathbb{F}_{q^m}^{s \times n},$$

where $\boldsymbol{Y}$ is the received matrix, $\boldsymbol{C}$ is the transmitted codeword matrix, and $\boldsymbol{E}$ is the error matrix. For example, in the Hamming metric, errors typically appear as "bursts", meaning that errors are distributed column-wise, affecting multiple codewords of the constituent code within the code matrix $\boldsymbol{C}$ simultaneously. This correlation between errors across codewords enables interleaving to correct more errors beyond the unique decoding radius of the constituent code. In the rank metric, improved performance is achieved for errors that lie within the same $\mathbb{F}_q$-row space.

Decoders for interleaved codes are available for RS codes in the Hamming metric [KL97; BKY03; CS03; BMS04; SSB07; WZB14; PR17; YL18] and for Gabidulin codes in the rank metric [Loi06; SB10; SJB11; WZ14; PRLS17; PMM$^+$17; BJPR21]. These decoders generally fall into two types: list decoders, which have an exponential list size in the worst case but a small list size on average, or probabilistic unique decoders that fail with very small probability.

Later in this thesis, we will extend this concept of interleaving and "burst errors" to the sum-rank metric, demonstrating how interleaving can further improve error correction capability in that context.

2.5 Polynomials over Finite Fields

Polynomials over finite fields are fundamental in the design and decoding of algebraic error-correcting block codes. A nonzero polynomial $f(x)$ over a field $\mathbb{F}_{q^m}$ is of the form

$$f(x) = \sum_{i \in \mathbb{Z}_{\geq 0}} f_i x^i, \quad \text{where} \quad f_i \in \mathbb{F}_{q^m}. \tag{2.11}$$

The *degree* of a polynomial $f(x)$ is given by

$$\deg(f) \stackrel{\text{def}}{=} \begin{cases} \max\{i \in \mathbb{Z}_{\geq 0} : f_i \neq 0\} & \text{if } f \neq 0, \\ -\infty & \text{otherwise} \end{cases}. \tag{2.12}$$

We now introduce a special class of polynomials called *skew polynomials*, first described by Ore in [Ore33b]. Skew polynomials facilitate the definition of LRS codes, which we discuss in Section 2.6.5.

2.5.1 Conjugacy Class

Let $\sigma : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ be a field automorphism of $\mathbb{F}_{q^m}$, and let $\delta : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ be a σ-derivation satisfying the following properties

$$\delta(a+b) = \delta(a) + \delta(b) \quad \text{and} \quad \delta(a \cdot b) = \delta(a) \cdot b + \sigma(a) \cdot \delta(b).$$

In the context of a finite field, all σ-derivations take the form described in [LMK14, Proposition 1] as

$$\delta(a) = b \cdot (\sigma(a) - a), \tag{2.13}$$

for some $b \in \mathbb{F}_{q^m}$.

From (2.13), it follows that the derivation δ becomes zero ($\delta = 0$) if the automorphism σ is the identity map ($\sigma = \mathsf{Id}$).

Let $a \in \mathbb{F}_{q^m}$ and $c \in \mathbb{F}_{q^m}^*$. We define the operation

$$a^c \stackrel{\text{def}}{=} \sigma(c) \cdot a \cdot c^{-1} + \delta(c) \cdot c^{-1},$$

where $\delta(c) \cdot c^{-1}$ is referred to as the *logarithmic derivative* of c.

Two elements $a, b \in \mathbb{F}_{q^m}$ are called (σ, δ)-conjugates if there exists an element $c \in \mathbb{F}_{q^m}^*$ such that $b = a^c$. If no such c exists, a and b are considered (σ, δ)-distinct.

The concept of (σ, δ)-conjugacy establishes an equivalence relation on $\mathbb{F}_{q^m}$, thereby partitioning $\mathbb{F}_{q^m}$ into conjugacy classes, as discussed in [LL88b]. To formalize this, we define the conjugacy class of an element $a \in \mathbb{F}_{q^m}$.

Definition 2.5 (Conjugacy Class [LL88b])**.** *Let $a \in \mathbb{F}_{q^m}$ and consider the automorphism σ and derivation δ on $\mathbb{F}_{q^m}$. The set*

$$\mathfrak{C}(a) \stackrel{\text{def}}{=} \left\{ a^c : c \in \mathbb{F}_{q^m}^* \right\},$$

is called the ***conjugacy class*** *of a.*

2.5.2 Skew Polynomials

Skew polynomials are *non-commutative* polynomials introduced by Ore [Ore33b].

Definition 2.6 (Skew Polynomial Ring)**.** *For a given automorphism $\sigma(\cdot)$ and a given derivation $\delta(\cdot)$, the set of all polynomials of the form specified in* (2.11)*, combined with ordinary polynomial addition and the multiplication rule*

$$x \cdot a = \sigma(a) \cdot x + \delta(a) \quad \textit{with} \quad a \in \mathbb{F}_{q^m},$$

forms the ***non-commutative*** *ring of* ***skew polynomials****, denoted by $\mathbb{F}_{q^m}[x; \sigma, \delta]$.*

The *degree* of a skew polynomial $f \in \mathbb{F}_{q^m}[x;\sigma,\delta]$ is defined in the same way as for ordinary polynomials (see (2.12)). Further, we denote by $\mathbb{F}_{q^m}[x;\sigma,\delta]_{<k}$ the set of skew polynomials from $\mathbb{F}_{q^m}[x;\sigma,\delta]$ with degree less than k, i.e.,

$$\mathbb{F}_{q^m}[x;\sigma,\delta]_{<k} \stackrel{\text{def}}{=} \{f \in \mathbb{F}_{q^m}[x;\sigma,\delta] : \deg(f) < k\}\,. \tag{2.14}$$

For skew polynomial rings with zero derivation, denoted as $\mathbb{F}_{q^m}[x;\sigma]$, there exists a ring isomorphism to $\mathbb{F}_{q^m}[x;\sigma,\delta]$ when δ is an inner derivation (as defined in (2.13)). This is always the case for finite fields. The isomorphism is described by the mapping (see [Mar18, Proposition 40] and [Liu16, Proposition 2.1.8]); for a $b \in \mathbb{F}_{q^m}$ we have

$$\begin{aligned}\mathbb{F}_{q^m}[x;\sigma,\delta] &\to \mathbb{F}_{q^m}[x;\sigma]\\ \sum_i f_i x^i &\mapsto \sum_i f_i (x-b)^i.\end{aligned}$$

The *monic* least-common left multiple (lclm) of a set $\{p_1, p_2, \ldots, p_n\} \subset \mathbb{F}_{q^m}[x;\sigma,\delta]$ of polynomials is denoted as

$$\operatorname{lclm}(p_i)_{1\le i\le n} \stackrel{\text{def}}{=} \operatorname{lclm}(p_1, p_2, \ldots, p_n)\,.$$

$\mathbb{F}_{q^m}[x;\sigma,\delta]$ is both a left and right Euclidean domain. For any $f \in \mathbb{F}_{q^m}[x;\sigma,\delta]$ and any nonzero $g \in \mathbb{F}_{q^m}[x;\sigma,\delta]$, there exist unique polynomials $q_L, r_L, q_R, r_R \in \mathbb{F}_{q^m}[x;\sigma,\delta]$ satisfying

$$f(x) = q_R(x)g(x) + r_R(x) = g(x)q_L(x) + r_L(x),$$

where $\deg(r_R), \deg(r_L) < \deg(g)$ (see [Ore33b]). This property enables both left and right division with remainder in $\mathbb{F}_{q^m}[x;\sigma,\delta]$.

Efficient algorithms for performing left and right skew polynomial division have been developed (e.g. [CL17b; CL17a; PW18]). We denote the remainder of the right division of f by g as $f \bmod_{\mathrm{r}} g$, where $f, g \in \mathbb{F}_{q^m}[x;\sigma,\delta]$.

Special Cases

There are several notable instances where skew polynomial rings align with other well-known polynomial rings:

- When the automorphism σ is the identity, the derivation δ becomes the zero derivation (see (2.13)). This means that the skew polynomial ring $\mathbb{F}_{q^m}[x;\sigma,\delta]$ is equivalent to the ordinary polynomial ring $\mathbb{F}_{q^m}[x]$.
- If the derivation δ is zero, the resulting ring $\mathbb{F}_{q^m}[x;\sigma]$ is known as the *twisted polynomial ring* (see [Gos96; Ros02]).
- For the Frobenius automorphism $\sigma_{\mathrm{Frob}}(x) = x^q$ of $\mathbb{F}_{q^m}$ and a zero derivation δ, the ring $\mathbb{F}_{q^m}[x;\sigma_{\mathrm{Frob}}]$ is isomorphic to the linearized polynomial ring [Ore33a;

Ore33b] denoted as $\mathbb{L}_{q^m}[x]$. Note that in Chapter 4 we are going to revisit the definition of linearized polynomials.

2.5.3 Generalized Operator Evaluation

The concept of generalized operator evaluation, as introduced in [Ler95], facilitates the $\mathbb{F}_q$-linearization of skew polynomial evaluation. This, in turn, establishes a connection between the skew polynomial ring and the linearized polynomial ring [Ore33b; Ore33a].

Consider an $\mathbb{F}_{q^m}$-automorphism σ, a derivation δ, and an element $a \in \mathbb{F}_{q^m}$. The (σ, δ) operator $\mathcal{D}_a^{\sigma,\delta}(b)\colon \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ is defined as

$$\mathcal{D}_a^{\sigma,\delta}(b) \stackrel{\text{def}}{=} \sigma(b)a + \delta(b) \qquad \forall b \in \mathbb{F}_{q^m}.$$

We denote this operator as $\mathcal{D}_a(b)$ when σ and δ are understood from the context. For an integer $i \geq 0$, we define $\mathcal{D}_a^{i+1}(b) = \mathcal{D}_a(\mathcal{D}_a^i(b))$ and $\mathcal{D}_a^0(b) = b$.

Definition 2.7 (Generalized Operator Evaluation [Mar18])**.** *For a skew polynomial $f \in \mathbb{F}_{q^m}[x;\sigma,\delta]$, the generalized operator evaluation $f(b)_a$ of f at an element $b \in \mathbb{F}_{q^m}$ with respect to the evaluation parameter $a \in \mathbb{F}_{q^m}$ is defined as*

$$f(b)_a \stackrel{\text{def}}{=} \sum_i f_i \mathcal{D}_a^i(b).$$

The generalized operator evaluation is an $\mathbb{F}_q$-linear map. For any $f \in \mathbb{F}_{q^m}[x;\sigma,\delta]$, $\lambda_1, \lambda_2 \in \mathbb{F}_q$, and $a, b_1, b_2 \in \mathbb{F}_{q^m}$, it holds that

$$f(\lambda_1 b_1 + \lambda_2 b_2)_a = \lambda_1 f(b_1)_a + \lambda_2 f(b_2)_a.$$

This result is shown in [Mar18, Lemma 23] and also discussed by [LL94].

For a vector $\boldsymbol{b} = [b_1, b_2, \ldots, b_n] \in \mathbb{F}_{q^m}^n$, the *generalized multipoint operator evaluation* of a skew polynomial $f \in \mathbb{F}_{q^m}[x;\sigma,\delta]$ with respect to an $a \in \mathbb{F}_{q^m}$ is defined as

$$f(\boldsymbol{b})_a \stackrel{\text{def}}{=} \left[f(b_1)_a, f(b_2)_a, \ldots, f(b_n)_a\right].$$

As demonstrated by [Car19], the *minimal skew polynomial* that evaluates to zero for all elements in $\boldsymbol{b}$ with respect to the evaluation parameters in $\boldsymbol{a}$ is defined as

$$M_{\boldsymbol{b}}^{\text{op}}(x)_{\boldsymbol{a}} = \operatorname{lclm}\left(x - \frac{\sigma(b_i)a_i + \delta(b_i)}{b_i}\right)_{\substack{1 \leq i \leq n \\ b_i \neq 0}}. \tag{2.15}$$

The degree of $M_{\boldsymbol{b}}^{\text{op}}(x)_{\boldsymbol{a}}$ satisfies

$$\deg(M_{\boldsymbol{b}}^{\text{op}}(x)_{\boldsymbol{a}}) \leq n.$$

Equality holds if and only if the elements b_i sharing the same evaluation parameter a_i are $\mathbb{F}_q$-linearly independent, and the a_i belong to distinct, non-trivial conjugacy classes. Additionally, all distinct evaluation parameters a_i must each lie in a separate conjugacy class. This is discussed in detail by [Car19].

Example 2.1. *Take the vectors $\boldsymbol{b} = [b_1, b_2, b_3, b_4] \in \mathbb{F}_{q^m}^4$ and $\boldsymbol{a} = [a_1, a_2, a_3, a_4] \in \mathbb{F}_{q^m}^4$, where $a_1 = a_2$ and $a_3 = a_4$ are representatives from different conjugacy classes. Then, $\deg(M_{\boldsymbol{b}}^{op}(x)_{\boldsymbol{a}}) = 4$ if and only if b_1 and b_2 are $\mathbb{F}_q$-linearly independent, and b_3 and b_4 are $\mathbb{F}_q$-linearly independent.*

Example 2.2. *Consider the vectors $\boldsymbol{b} = [b_1, b_2, b_3] \in \mathbb{F}_{q^m}^3$ and $\boldsymbol{a} = [a_1, a_2, a_3] \in \mathbb{F}_{q^m}^3$, where a_1, a_2, and a_3 are representatives from three distinct conjugacy classes. Then, $\deg(M_{\boldsymbol{b}}^{op}(x)_{\boldsymbol{a}}) = 3$ if and only if b_1, b_2, and b_3 are $\mathbb{F}_q$-linearly independent.*

The generalized operator evaluation of a polynomial modulo a particular minimal polynomial exhibits properties analogous to those of ordinary polynomials. We formalize this in the following lemma.

Lemma 2.1 ([BJR24])**.** *For any $f \in \mathbb{F}_{q^m}[x; \sigma, \delta]$, and vectors $\boldsymbol{b} = [b_1, \ldots, b_n]$ and $\boldsymbol{a} = [a_1, \ldots, a_n] \in \mathbb{F}_{q^m}^n$, the following equality holds*

$$f(b_i)_{a_i} = f \bmod_{\mathrm{r}} M_{\boldsymbol{b}}^{op}(x)_{\boldsymbol{a}}(b_i)_{a_i}, \quad \forall i \in \{1, \ldots, n\}.$$

Proof. The proof leverages the Euclidean domain property of $\mathbb{F}_{q^m}[x; \sigma, \delta]$. We can express $f(x)$ uniquely as

$$f(x) = q(x) M_{\boldsymbol{b}}^{\mathrm{op}}(x)_{\boldsymbol{a}} + r(x),$$

where $q, r \in \mathbb{F}_{q^m}[x; \sigma, \delta]$ and $\deg(r) < \deg(M_{\boldsymbol{b}}^{\mathrm{op}}(x)_{\boldsymbol{a}})$. By definition

$$r(x) = f(x) \bmod_{\mathrm{r}} M_{\boldsymbol{b}}^{\mathrm{op}}(x)_{\boldsymbol{a}}.$$

Given that $M_{\boldsymbol{b}}^{\mathrm{op}}(x)_{\boldsymbol{a}}$ vanishes for all (b_i, a_i) pairs, we have

$$f(b_i)_{a_i} = r(b_i)_{a_i} = f \bmod_{\mathrm{r}} M_{\boldsymbol{b}}^{\mathrm{op}}(x)_{\boldsymbol{a}}(b_i)_{a_i}, \quad \forall i \in \{1, \ldots, n\},$$

which concludes the proof. □

2.5.4 Generalized Moore Matrix

With the notion of the generalized operator evaluation we can define the *generalized Moore matrix*.

Definition 2.8 (Generalized Moore Matrix)**.** *Let $\boldsymbol{x} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(\ell)}] \in \mathbb{F}_{q^m}^n$ with $\boldsymbol{x}^{(i)} \in \mathbb{F}_{q^m}^{n_i}$ according to some length profile $\boldsymbol{n} = [n_1, n_2, \ldots, n_\ell] \in \mathbb{Z}_{\geq 0}^{\ell}$. Further let*

$\boldsymbol{a} = [a_1, a_2, \ldots, a_\ell] \in \mathbb{F}_{q^m}^{\ell}$ *and an integer* $z \in \mathbb{Z}_{\geq 0}$ *be given. The* ***generalized Moore matrix*** $\mathfrak{M}_z(\boldsymbol{x})_{\boldsymbol{a}}$ *is defined as*

$$\mathfrak{M}_z(\boldsymbol{x})_{\boldsymbol{a}} \stackrel{\text{def}}{=} \left[\begin{array}{c|c|c|c} \boldsymbol{M}_z\left(\boldsymbol{x}^{(1)}\right)_{a_1} & \boldsymbol{M}_z\left(\boldsymbol{x}^{(2)}\right)_{a_2} & \ldots & \boldsymbol{M}_z\left(\boldsymbol{x}^{(\ell)}\right)_{a_\ell} \end{array} \right] \in \mathbb{F}_{q^m}^{z \times n}, \tag{2.16}$$

where

$$\boldsymbol{M}_z\left(\boldsymbol{x}^{(i)}\right)_{a_i} \stackrel{\text{def}}{=} \begin{bmatrix} x_1^{(i)} & \ldots & x_{n_i}^{(i)} \\ \mathcal{D}_{a_i}(x_1^{(i)}) & \ldots & \mathcal{D}_{a_i}(x_{n_i}^{(i)}) \\ \vdots & \ddots & \vdots \\ \mathcal{D}_{a_i}^{z-1}(x_1^{(i)}) & \ldots & \mathcal{D}_{a_i}^{z-1}(x_{n_i}^{(i)}) \end{bmatrix} \quad \forall i \in \{1, \ldots, \ell\}.$$

Theorem 2.1 (Full Rank of Generalized Moore Matrix [Mar18, Theorem 14], [LL88a, Theorem 4.5])**.** *Let* $\boldsymbol{a}$ *include representatives from pairwise distinct nontrivial conjugacy classes of* $\mathbb{F}_{q^m}$*, and let* $\mathrm{rk_q}\left(\boldsymbol{x}^{(i)}\right) = n_i$ *for all* $i \in \{1, \ldots, \ell\}$*. Then, the generalized Moore matrix is of full rank, i.e.,*

$$\mathrm{rk_{q^m}}\left(\mathfrak{M}_z(\boldsymbol{x})_{\boldsymbol{a}}\right) = \min\{z, n\}.$$

Moore Matrix

The Moore matrix [Moo96] is a special case of the generalized Moore matrix with the following parameters:

- Automorphism: $\sigma(x) = x^q$ (Frobenius automorphism).
- Derivation: $\delta = 0$,
- Number of blocks: $\ell = 1$,
- $\boldsymbol{a} = [1,] \in \mathbb{F}_{q^m}^{1}$,
- Length profile: $\boldsymbol{n} = [n,] \in \mathbb{Z}_{\geq 0}^{1}$.

Given $\boldsymbol{x} \in \mathbb{F}_{q^m}^{n}$ and an integer $z \in \mathbb{Z}_{\geq 0}$, denote the i-th q-power by $x^{[i]}$ where $[i] \stackrel{\text{def}}{=} q^i$. Then the Moore matrix $\boldsymbol{M}_z(\boldsymbol{x}) \in \mathbb{F}_{q^m}^{z \times n}$ is

$$\boldsymbol{M}_z(\boldsymbol{x}) = \mathfrak{M}_z(\boldsymbol{x})_{\boldsymbol{a}} = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \\ x_1^{[1]} & x_2^{[1]} & \ldots & x_n^{[1]} \\ \vdots & \ddots & \vdots & \vdots \\ x_1^{[z-1]} & x_2^{[z-1]} & \ldots & x_n^{[z-1]} \end{bmatrix}. \tag{2.17}$$

According to Theorem 2.1, the Moore matrix is of full rank if and only if $\mathrm{rk_q}(\boldsymbol{x}) = n$.

Vandermonde Matrix

The Vandermonde matrix is another special case of the generalized Moore matrix with the following parameters:

- Automorphism: $\sigma(x) = x$ (identity automorphism),
- Derivation: $\delta = 0$ (implied by $\sigma = \mathsf{Id}$),
- Number of blocks: $\ell = n$,
- $\boldsymbol{v} = [1, 1, \ldots, 1] \in \mathbb{F}_{q^m}^n$,
- Length profile: $\boldsymbol{n} = [1, 1, \ldots, 1] \in \mathbb{Z}_{\geq 0}^n$.

Then the Vandermonde matrix $\boldsymbol{V}_z(\boldsymbol{x}) \in \mathbb{F}_{q^m}^{z \times n}$ is

$$\boldsymbol{V}_z(\boldsymbol{x}) \stackrel{\text{def}}{=} \mathfrak{M}_z(\boldsymbol{v})_{\boldsymbol{x}} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x^{(1)} & x^{(2)} & \ldots & x^{(n)} \\ (x^{(1)})^2 & (x^{(2)})^2 & \ldots & (x^{(n)})^2 \\ \vdots & \vdots & \ddots & \vdots \\ (x^{(1)})^{z-1} & (x^{(2)})^{z-1} & \ldots & (x^{(n)})^{z-1} \end{bmatrix}, \tag{2.18}$$

where $\boldsymbol{x} = [x^{(1)}, x^{(2)}, \ldots, x^{(n)}] \in \mathbb{F}_{q^m}^n$.

According to Theorem 2.1, the Vandermonde matrix is of full rank if and only if the elements of $\boldsymbol{x}$ are pairwise distinct and nonzero.

2.6 Codes in the Sum-Rank Metric

The sum-rank metric, introduced in [Mar18], is a metric that includes both the Hamming and rank metrics as special cases, effectively blending the properties of these two metrics. Initially referred to as the *extended rank metric* in 2010 for multi-shot network coding [NU10], the sum-rank metric has since gained significant attention for its applications in distributed storage [MK19b], network coding [MK19a], space-time coding [SK20], and more recently, in code-based cryptography [PRR22; HBH23].

The sum-rank metric combines aspects of both the Hamming and rank metrics by dividing a vector into several blocks and computing the rank of each block. The sum of these ranks gives the sum-rank metric. If each block is a single column, the sum-rank metric reduces to the Hamming metric, which counts the number of nonzero elements in the vector. Conversely, if the entire vector is treated as a single block, the sum-rank metric coincides with the rank metric, which measures the rank of the vector as a matrix over its base field.

Numerous code constructions and efficient decoding algorithms have been developed for the sum-rank metric [WSBZ11; WS12; WSS15; NPRV17; Mar18; MK19a; Bou20;

Car19; BJPR21]. Furthermore, several key results on the fundamental properties of sum-rank-metric codes have been established, including bounds on code parameters, MacWilliams identities, and the Gilbert–Varshamov bound [BGR21]. Further properties and extended research on sum-rank-metric codes have been explored in other studies, such as [OLW22; OPB21; CJB24].

In this section, we review fundamental definitions related to the sum-rank metric and the associated codes. We will also introduce the most prominent class of linear codes within the sum-rank metric, known as LRS codes.

2.6.1 The Sum-Rank Metric and its Properties

We consider vectors that are divided into blocks, such that

$$\boldsymbol{x} = \left[\boldsymbol{x}^{(1)} \mid \boldsymbol{x}^{(2)} \mid \ldots \mid \boldsymbol{x}^{(\ell)}\right] \in \mathbb{F}_{q^m}^n,$$

where each block $\boldsymbol{x}^{(i)} \in \mathbb{F}_{q^m}^{n_i}$ has length $n_i \in \mathbb{Z}_{\geq 0}$ for $i \in \{1, \ldots, \ell\}$. We use the term *block length* to denote the length of an individual block with respect to the sum-rank metric. Note that in some literature, the term "block length" may refer to the length of a block code. In this thesis, however, we use the term *code length* to refer to the length of a block code and *block length* to refer to the length of an individual block with respect to the sum-rank metric. The vector

$$\boldsymbol{n} \stackrel{\text{def}}{=} [n_1, n_2, \ldots, n_\ell] \in \mathbb{Z}_{\geq 0}^{\ell},$$

is referred to as the *length profile* and satisfies

$$n = \sum_{i=1}^{\ell} n_i.$$

The sum-rank weight of a vector $\boldsymbol{x} = [\boldsymbol{x}^{(1)} \mid \boldsymbol{x}^{(2)} \mid \ldots \mid \boldsymbol{x}^{(\ell)}] \in \mathbb{F}_{q^m}^n$ with respect to the length profile $\boldsymbol{n}$ is defined as

$$\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{x}) \stackrel{\text{def}}{=} \sum_{i=1}^{\ell} \mathrm{rk_q}\left(\boldsymbol{x}^{(i)}\right), \tag{2.19}$$

where $\mathrm{rk_q}(\boldsymbol{x}^{(i)}) = \dim_{\mathrm{q}}\left(\langle x_1^{(i)}, \ldots, x_{n_i}^{(i)}\rangle_q\right)$ denotes the dimension of the $\mathbb{F}_q$-span of the entries of $\boldsymbol{x}^{(i)}$ which is equal to the $\mathbb{F}_q$-rank of $\mathrm{ext}\left(\boldsymbol{x}^{(i)}\right)$ (see (2.3)).

The sum-rank distance between two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_{q^m}^n$ is then induced by the sum-rank weight and defined as

$$d_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{x} - \boldsymbol{y}). \tag{2.20}$$

We denote $\mathbb{F}_{q^m}$-linear codes considered with respect to the sum-rank metric and a given length profile $\boldsymbol{n}$ as

$$\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k, d_{\min}].$$

As mentioned before, the sum-rank metric includes well-known metrics as special cases:

- **Hamming metric:** When $\ell = n$, this implies $n_i = 1$ for all $i \in \{1, \ldots, \ell\}$, with $\boldsymbol{n} = [1, 1, \ldots, 1] \in \mathbb{Z}_{\geq 0}^{n}$. In this case, each $\boldsymbol{x}^{(i)} \in \mathbb{F}_{q^m}^{1}$, and the sum-rank metric reduces to the Hamming metric (see (2.9) and (2.10)). Therefore,

$$\mathrm{wt}_H(\boldsymbol{e}) = \mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e}) = \sum_{i=1}^{n} \mathrm{rk_q}\left(\boldsymbol{x}^{(i)}\right).$$

  Here, $\mathrm{rk_q}(\boldsymbol{x}^{(i)})$ is 1 if and only if $\boldsymbol{x}^{(i)}$ is a nonzero element, and 0 if $\boldsymbol{x}^{(i)}$ is zero, since $\mathrm{ext}\left(\boldsymbol{x}^{(i)}\right)$ is a column vector over $\mathbb{F}_q$.

- **Rank metric:** When $\ell = 1$, the sum-rank metric corresponds to the rank metric, with $\boldsymbol{x} = [\boldsymbol{x}^{(1)}] \in \mathbb{F}_{q^m}^{n}$ and $\boldsymbol{n} = [n] \in \mathbb{Z}_{\geq 0}^{1}$. Thus,

$$\mathrm{wt}_R(\boldsymbol{e}) = \mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e}) = \mathrm{rk_q}\left(\boldsymbol{x}^{(1)}\right).$$

In some parts of this thesis, we restrict ourselves to the case where each block has a constant length η. This means that the length profile is given by

$$\boldsymbol{n} = [\eta, \eta, \ldots, \eta] \in \mathbb{Z}_{\geq 0}^{\ell},$$

and thus $n = \ell\eta$. In this context, we omit $\boldsymbol{n}$ in the notation and simply write $\mathrm{wt}_{\Sigma R}(\cdot)$ and $d_{\Sigma R}(\cdot, \cdot)$, respectively. For notational convenience, we may switch between these notations and assume a constant length for each block. Whenever we state an algorithm using constant block length notation, we will provide a remark or note indicating if the algorithm can be easily adapted for variable block lengths. It should be clear from the context when we assume constant block lengths and when we consider variable lengths.

The maximum rank of each block is defined as

$$\mu_i \stackrel{\text{def}}{=} \max\{n_i, m\} \quad \forall i \in \{1, \ldots, \ell\}, \tag{2.21}$$

and for the case of constant block length as

$$\mu \stackrel{\text{def}}{=} \min\{\eta, m\}. \tag{2.22}$$

For a vector $\boldsymbol{x} = [\boldsymbol{x}^{(1)} \mid \ldots \mid \boldsymbol{x}^{(\ell)}] \in \mathbb{F}_{q^m}^{n}$, we may be interested in the sequence of rank weights of the individual blocks of $\boldsymbol{x}$.

Definition 2.9 (Rank Profile)**.** *Let $\ell \in \mathbb{Z}$ be the number of blocks, and let the vector $\boldsymbol{x} = [\boldsymbol{x}^{(1)} \mid \ldots \mid \boldsymbol{x}^{(\ell)}] \in \mathbb{F}_{q^m}^n$ be partitioned into blocks according to a length profile $\boldsymbol{n} \in \mathbb{Z}^\ell$. For each block $i \in \{1, \ldots, \ell\}$, let μ_i denote its maximum rank weight, as defined in* (2.21)*. The map*

$$\psi : \mathbb{F}_{q^m}^n \to \{0, \ldots, \mu_1\} \times \{0, \ldots, \mu_2\} \times \cdots \times \{0, \ldots, \mu_\ell\},$$

is defined by

$$\psi(\boldsymbol{x}) \mapsto \left[\mathrm{rk_q}(\boldsymbol{x}^{(1)}), \ldots, \mathrm{rk_q}(\boldsymbol{x}^{(\ell)})\right]. \tag{2.23}$$

*The image of $\psi(\boldsymbol{x})$ is called the **rank profile** of $\boldsymbol{x}$.*

Additionally, we define the set of all possible rank profiles. For convenience, we assume constant block length for some of the upcoming notations. However, extending the results to variable block lengths is straightforward.

Definition 2.10 (Set of Rank Profiles)**.** *Let $w \in \mathbb{Z}$ be an integer and $\ell \in \mathbb{Z}$ the number of blocks, and $\mu \in \mathbb{Z}$ the maximum rank weight of each block. Assume $w \leq \ell\mu$. We define the set*

$$\mathcal{T}_{w,\ell,\mu} \stackrel{\text{def}}{=} \left\{ [w_1, w_2, \ldots, w_\ell] \in \{0, \ldots, \mu\}^\ell : \sum_{i=1}^{\ell} w_i = w \right\},$$

which contains all possible sequences of rank weights (rank profiles) of a vector consisting of ℓ blocks and having a sum-rank weight of w [1].

In [PRR22], an upper bound on the cardinality of the set $\mathcal{T}_{w,\ell,\mu}$ has been derived as

$$|\mathcal{T}_{w,\ell,\mu}| \leq \binom{\ell + w - 1}{\ell - 1}.$$

The set of all vectors in $\mathbb{F}_{q^m}^n$ of sum-rank weight $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}) = w$ is denoted by

$$\mathcal{E}_{q,\eta,m,\ell}(w) \stackrel{\text{def}}{=} \{\boldsymbol{e} \in \mathbb{F}_{q^m}^n : \mathrm{wt}_{\Sigma R}(\boldsymbol{e}) = w\}. \tag{2.24}$$

The cardinality of this set is given by (see [PRR22])

$$|\mathcal{E}_{q,\eta,m,\ell}(w)| = \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m, \eta, w_i\right), \tag{2.25}$$

[1] Rank profiles are closely related to the concept known in the literature as *weak integer compositions*, where an integer is expressed as the sum of non-negative integers. In contrast to weak integer compositions, rank profiles have the additional constraint that each part (rank weight) does not exceed a fixed upper bound, which in our case is μ.

with a given rank profile $\boldsymbol{w} = [w_1, \ldots, w_\ell]$. The cardinality expression in (2.25) can be efficiently computed using a dynamic programming approach, as described in [PRR22].

The term $\mathrm{NM}_q\left(m, \eta, w_i\right)$ represents the number of matrices of size $m \times \eta$ of rank w_i over the finite field $\mathbb{F}_q$. It can be calculated using the following formula [MS77, Chapter 13][MMO04]

$$\begin{aligned} \mathrm{NM}_q\left(m, \eta, w_i\right) &= \prod_{j=0}^{w_i-1} \frac{(q^m - q^j)(q^\eta - q^j)}{q^{w_i} - q^j} \\ &= \begin{bmatrix} \eta \\ w_i \end{bmatrix}_q \prod_{j=0}^{w_i-1} (q^m - q^j) \\ &= \begin{bmatrix} m \\ w_i \end{bmatrix}_q \prod_{j=0}^{w_i-1} (q^\eta - q^j), \end{aligned} \tag{2.26}$$

where $\begin{bmatrix} a \\ b \end{bmatrix}_q$, with $a \geq b \geq 0$, is defined in (2.4).

2.6.2 Interleaved Sum-Rank-Metric Codes

The definition of vertically interleaved codes in the sum-rank metric is straightforward and follows the same principle as in Definition 2.4. We denote such a code as $\mathcal{IC}_{\Sigma R}[s; \boldsymbol{n}, k, d_{\min}]$.

Each codeword $\boldsymbol{C} \in \mathcal{IC}_{\Sigma R}[s; \boldsymbol{n}, k, d_{\min}]$ can be written as

$$\boldsymbol{C} = \left[\begin{array}{c|c|c|c} \boldsymbol{c}_1^{(1)} & \boldsymbol{c}_1^{(2)} & \ldots & \boldsymbol{c}_1^{(\ell)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{c}_s^{(1)} & \boldsymbol{c}_s^{(2)} & \ldots & \boldsymbol{c}_s^{(\ell)} \end{array}\right] \in \mathbb{F}_{q^m}^{s \times n},$$

or equivalently as

$$\boldsymbol{C} = \left[\boldsymbol{C}^{(1)} \mid \boldsymbol{C}^{(2)} \mid \ldots \mid \boldsymbol{C}^{(\ell)}\right],$$

where

$$\boldsymbol{C}^{(i)} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{c}_1^{(i)} \\ \boldsymbol{c}_2^{(i)} \\ \vdots \\ \boldsymbol{c}_s^{(i)} \end{bmatrix} \in \mathbb{F}_{q^m}^{s \times n_i},$$

for all $i \in \{1, \ldots, \ell\}$.

2.6.3 Channel Models

Consider a $\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k]$ linear sum-rank-metric code. Throughout this thesis, we mostly (if not specified otherwise) consider additive error channels of the form

$$\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e} \ \in \mathbb{F}_{q^m}^n,$$

where $\boldsymbol{c} \in \mathcal{C}_{\Sigma R}$ and $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ is an error vector with sum-rank weight $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e}) = w$.

The following theorem establishes the connection between the error vector $\boldsymbol{e}$ and the row and column spaces of its blocks.

Theorem 2.2 (Error Decomposition in the Sum-Rank Metric [MP74, Theorem 1], [PRR22, Lemma 10])**.** *Let the error vector $\boldsymbol{e} = [\boldsymbol{e}^{(1)} \mid \boldsymbol{e}^{(2)} \mid \ldots \mid \boldsymbol{e}^{(\ell)}] \in \mathbb{F}_{q^m}^n$ be partitioned into blocks with respect to the length profile $\boldsymbol{n} = [n_1, \ldots, n_\ell] \in \mathbb{Z}_{\geq 0}^\ell$. The error vector $\boldsymbol{e}$ can be decomposed, though not necessarily uniquely, as*

$$\boldsymbol{e} = \boldsymbol{a} \cdot \boldsymbol{B}, \tag{2.27}$$

where $\boldsymbol{a} = [\boldsymbol{a}^{(1)} \mid \ldots \mid \boldsymbol{a}^{(\ell)}]$ with $\boldsymbol{a}^{(i)} \in \mathbb{F}_{q^m}^{w_i}$, and

$$\boldsymbol{B} = \mathrm{diag}(\boldsymbol{B}^{(1)}, \ldots, \boldsymbol{B}^{(\ell)}) \in \mathbb{F}_q^{w \times n}, \tag{2.28}$$

with $\boldsymbol{B}^{(i)} \in \mathbb{F}_q^{w_i \times n_i}$ satisfying $\mathrm{rk_q}(\boldsymbol{a}^{(i)}) = \mathrm{rk_q}(\boldsymbol{B}^{(i)}) = w_i$ for all $i \in \{1, \ldots, \ell\}$, and $w = \sum_{i=1}^{\ell} w_i$. The rank profile is given by

$$\boldsymbol{w} = \psi(\boldsymbol{e}) = [w_1, w_2, \ldots, w_\ell]\,.$$

For each $i \in \{1, \ldots, \ell\}$, the entries of $\boldsymbol{a}^{(i)}$ form a basis over $\mathbb{F}_q$ of the $\mathbb{F}_q$-column space of $\boldsymbol{e}^{(i)}$, and the rows of $\boldsymbol{B}^{(i)}$ form a basis over $\mathbb{F}_q$ of its $\mathbb{F}_q$-row space.

In the interleaved case with interleaving order s, we extend from vectors to matrices by considering the additive sum-rank channel

$$\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E} \quad \in \mathbb{F}_{q^m}^{s \times n}, \tag{2.29}$$

where the error matrix $\boldsymbol{E}$ is structured as

$$\boldsymbol{E} = \left[\boldsymbol{E}^{(1)} \mid \boldsymbol{E}^{(2)} \mid \ldots \mid \boldsymbol{E}^{(\ell)}\right] \in \mathbb{F}_{q^m}^{s \times n}.$$

Here, each block $\boldsymbol{E}^{(i)} \in \mathbb{F}_{q^m}^{s \times n_i}$ has rank $\mathrm{rk_q}(\boldsymbol{E}^{(i)}) = w_i$ for all $i \in \{1, \ldots, \ell\}$. The overall sum-rank weight of the matrix $\boldsymbol{E}$ is $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{E}) = w = \sum_{i=1}^{\ell} w_i$.

This matrix $\boldsymbol{E}$ can be decomposed similarly to the vector case, as follows

$$\boldsymbol{E} = \boldsymbol{A} \cdot \boldsymbol{B}, \tag{2.30}$$

where $\boldsymbol{A} = \left[\boldsymbol{A}^{(1)} \mid \boldsymbol{A}^{(2)} \mid \ldots \mid \boldsymbol{A}^{(\ell)}\right] \in \mathbb{F}_{q^m}^{s\times w}$ is a matrix with $\boldsymbol{A}^{(i)} \in \mathbb{F}_{q^m}^{s\times w_i}$ satisfying $\mathrm{rk_q}(\boldsymbol{A}^{(i)}) = w_i$. The matrix $\boldsymbol{B}$ has the same block-diagonal matrix structure as $\boldsymbol{B}$ defined in (2.28).

Similarly to (2.21) and (2.22) for the non-interleaved case, we define $\mu^{(s)}$ as the maximum possible $\mathbb{F}_q$-rank of each block of the error matrix. It is given by

$$\mu_i^{(s)} \stackrel{\text{def}}{=} \min\{n_i, sm\} \quad \forall i \in \{1, \ldots, \ell\}, \tag{2.31}$$

and for constant block lengths, this becomes

$$\mu^{(s)} \stackrel{\text{def}}{=} \min\{\eta, sm\}. \tag{2.32}$$

2.6.4 Row and Column Support in the Sum-Rank Metric

In this subsection, we define the row and column support in the sum-rank metric. We provide the definition for the interleaved case, where the error is represented by a matrix $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s\times n}$. The non-interleaved case follows naturally when $s = 1$.

Before introducing the sum-rank support, we first define the notion of support with respect to the rank metric for interleaved codes.

Definition 2.11 (Rank Support in the Interleaved Setting)**.** *Let $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s\times n}$ be a matrix, the row and column rank supports of $\boldsymbol{E}$ are defined as follows:*

- ***Row Rank Support:*** *The* ***row rank support*** *of $\boldsymbol{E}$ is defined as*

$$\mathrm{supp}_R^{(R)}(\boldsymbol{E}) \stackrel{\text{def}}{=} \mathcal{R}_q(\boldsymbol{E})\,.$$

- ***Column Rank Support:*** *The* ***column rank support*** *of $\boldsymbol{E}$ is defined as*

$$\mathrm{supp}_R^{(C)}(\boldsymbol{E}) \stackrel{\text{def}}{=} \mathcal{C}_q(\boldsymbol{E})\,.$$

Using the rank support defined above, we now define the row and column support in the sum-rank metric for interleaved codes.

Definition 2.12 (Sum-Rank Support in the Interleaved Setting)**.** *Let $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s\times n}$ be a matrix and s be the interleaving order. The sum-rank row and column supports are defined as the Cartesian product of the row and column rank supports of each block:*

- ***Sum-Rank Row Support:*** *The sum-rank row support is given by*

$$\begin{aligned}\mathrm{supp}_{\Sigma R}^{(R)}(\boldsymbol{E}) &\stackrel{\text{def}}{=} \mathrm{supp}_R^{(R)}(\boldsymbol{E}_1) \times \mathrm{supp}_R^{(R)}(\boldsymbol{E}_2) \times \cdots \times \mathrm{supp}_R^{(R)}(\boldsymbol{E}_\ell) \\ &= \mathcal{R}_q(\boldsymbol{E}_1) \times \mathcal{R}_q(\boldsymbol{E}_2) \times \cdots \times \mathcal{R}_q(\boldsymbol{E}_\ell)\,.\end{aligned} \tag{2.33}$$

- ***Sum-Rank Column Support:*** *The sum-rank column support is given by*

$$\begin{aligned}\operatorname{supp}_{\Sigma R}^{(C)}(\boldsymbol{E}) &\stackrel{\text{def}}{=} \operatorname{supp}_{R}^{(C)}(\boldsymbol{E}_1) \times \operatorname{supp}_{R}^{(C)}(\boldsymbol{E}_2) \times \cdots \times \operatorname{supp}_{R}^{(C)}(\boldsymbol{E}_\ell) \\ &= \mathcal{C}_q(\boldsymbol{E}_1) \times \mathcal{C}_q(\boldsymbol{E}_2) \times \cdots \times \mathcal{C}_q(\boldsymbol{E}_\ell)\,.\end{aligned} \tag{2.34}$$

The second equality for both the sum-rank row and column supports follows directly from Theorem 2.2. For $s = 1$, these definitions reduce to the non-interleaved case for a vector $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$.

Assume $\boldsymbol{\mathcal{E}}$ to be either a row or column support. We denote by $\dim_\Sigma(\boldsymbol{\mathcal{E}})$ the *sum dimension*

$$\dim_\Sigma(\boldsymbol{\mathcal{E}}) \stackrel{\text{def}}{=} \sum_{i=1}^{\ell} \dim(\mathcal{E}^{(i)}).$$

The intersection of two supports $\boldsymbol{\mathcal{E}}_1$ and $\boldsymbol{\mathcal{E}}_2$ is defined as

$$\boldsymbol{\mathcal{E}}_1 \cap \boldsymbol{\mathcal{E}}_2 \stackrel{\text{def}}{=} \mathcal{E}_1^{(1)} \cap \mathcal{E}_2^{(1)} \times \cdots \times \mathcal{E}_1^{(\ell)} \cap \mathcal{E}_2^{(\ell)}.$$

Given two supports $\boldsymbol{\mathcal{E}}_1$ and $\boldsymbol{\mathcal{E}}_2$, we say that $\boldsymbol{\mathcal{E}}_2$ is a *super-support* of $\boldsymbol{\mathcal{E}}_1$, denoted by $\boldsymbol{\mathcal{E}}_1 \subseteq \boldsymbol{\mathcal{E}}_2$, if $\mathcal{E}_1^{(i)} \subseteq \mathcal{E}_2^{(i)}$ for all $i \in \{1, \ldots, \ell\}$. Conversely, $\boldsymbol{\mathcal{E}}_1$ is a *sub-support* of $\boldsymbol{\mathcal{E}}_2$.

For a given length profile $\boldsymbol{n} = [n_1, n_2, \ldots, n_\ell]$, we define the notation

$$\mathbb{F}_q^{\boldsymbol{n}} \stackrel{\text{def}}{=} \mathbb{F}_q^{n_1} \times \mathbb{F}_q^{n_2} \times \cdots \times \mathbb{F}_q^{n_\ell}. \tag{2.35}$$

2.6.5 Linearized Reed–Solomon Codes

LRS codes [LK05; Mar18; Car19] represent a special class of linear sum-rank-metric codes that satisfy the maximum sum-rank distance (MSRD) property. In other words, they attain the maximum possible minimum distance according to the Singleton-like bound in the sum-rank metric.

The fundamental properties and concepts of LRS codes are extensively explored in the literature. For a comprehensive overview, refer to [MSK22].

In the following, we define LRS codes as evaluation codes of degree-restricted skew polynomials and present their generator matrices.

Definition 2.13 (Linearized Reed–Solomon Code)**.** *Let $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_\ell] \in \mathbb{F}_{q^m}^\ell$ be a vector containing representatives from different conjugacy classes of $\mathbb{F}_{q^m}$, and let a length profile $\boldsymbol{n} = [n_1, n_2, \ldots, n_\ell] \in \mathbb{Z}_{\geq 0}^\ell$ be given. For each $i \in \{1, \ldots, \ell\}$, let the vector $\boldsymbol{\beta}^{(i)} = [\beta_1^{(i)}, \ldots, \beta_{n_i}^{(i)}] \in \mathbb{F}_{q^m}^{n_i}$ consist of $\mathbb{F}_q$-linearly independent elements from $\mathbb{F}_{q^m}$, and consider the vector $\boldsymbol{\beta} = [\boldsymbol{\beta}^{(1)} \mid \boldsymbol{\beta}^{(2)} \mid \ldots \mid \boldsymbol{\beta}^{(\ell)}] \in \mathbb{F}_{q^m}^n$.*

Let the following conditions hold

$$\ell \leq q - 1 \quad \textit{and} \quad m \geq n_i \quad \forall i \in \{1, \ldots, \ell\}.$$

A LRS code of length $n = \sum_{i=1}^{\ell} n_i$ *and dimension* k *is defined as*

$$\mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; \boldsymbol{n}, k] \stackrel{\text{def}}{=} \left\{ \left[f(\boldsymbol{\beta}^{(1)})_{\xi_1} \mid \cdots \mid f(\boldsymbol{\beta}^{(\ell)})_{\xi_\ell} \right] : f \in \mathbb{F}_{q^m}[x; \sigma, \delta]_{<k} \right\} \subseteq \mathbb{F}_{q^m}^n.$$

With respect to the length profile $\boldsymbol{n}$, each codeword $\boldsymbol{c} \in \mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; \boldsymbol{n}, k]$ is structured as

$$\boldsymbol{c} = \left[\boldsymbol{c}^{(1)} \mid \boldsymbol{c}^{(2)} \mid \ldots \mid \boldsymbol{c}^{(\ell)} \right],$$

where $\boldsymbol{c}^{(i)} = f(\boldsymbol{\beta}^{(i)})_{\xi_i} \in \mathbb{F}_{q^m}^{n_i}$ for all $i \in \{1, \ldots, \ell\}$.

Another way to describe an $\mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; \boldsymbol{n}, k]$ code is through its generator matrix, given by the generalized Moore matrix (see Definition 2.8)

$$\boldsymbol{G}_{\mathrm{LRS}} = \mathfrak{M}_k(\boldsymbol{\beta})_{\boldsymbol{\xi}}.$$

LRS codes achieve the Singleton-like bound in the sum-rank metric (see [Mar18, Proposition 34]) with equality, meaning the minimum sum-rank distance equals $n - k + 1$.

Efficient (polynomial-time) algorithms exist [Mar18; Car19; Bou20] that enable BMD decoding of errors with sum-rank weight w up to

$$w \leq \frac{n-k}{2}.$$

Note that for constant block lengths, we write $\mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; n, k]$ instead, where

$$n = \eta\ell,$$

and η is implicitly defined by $\boldsymbol{\beta}^{(i)} \in \mathbb{F}_{q^m}^{\eta}$ for $i \in \{1, \ldots, \ell\}$.

2.6.6 Interleaved Linearized–Reed Solomon Codes

Motivated by the results on IRS codes [KL97; KY98] and Gabidulin codes [Loi06] we define ILRS [HB23] as follows.

Definition 2.14 (ILRS Code)**.** *Let* $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_\ell] \in \mathbb{F}_{q^m}^{\ell}$ *be a vector containing representatives from different conjugacy classes of* $\mathbb{F}_{q^m}$*. Further let a length profile* $\boldsymbol{n} = [n_1, n_2, \ldots, n_\ell] \in \mathbb{Z}_{\geq 0}^{\ell}$ *be given and let the vectors* $\boldsymbol{\beta}^{(i)} = [\beta_1^{(i)}, \ldots, \beta_{n_i}^{(i)}] \in \mathbb{F}_{q^m}^{n_i}$ *contain* $\mathbb{F}_q$*-linearly independent elements from* $\mathbb{F}_{q^m}$ *for all* $i \in \{1, \ldots, \ell\}$*. Consider the vector* $\boldsymbol{\beta} = [\boldsymbol{\beta}^{(1)} \mid \boldsymbol{\beta}^{(2)} \mid \ldots \mid \boldsymbol{\beta}^{(\ell)}] \in \mathbb{F}_{q^m}^{n}$*. Assume* $\ell \leq q - 1$ *and* $m \geq n_i$ *for all* $i \in \{1, \ldots, \ell\}$*. Let* $\mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; \boldsymbol{n}, k]$ *be the constituent LRS code. The corresponding*

s-interleaved LRS code is defined as

$$\mathrm{ILRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell, s; \boldsymbol{n}, k] \stackrel{\text{def}}{=} \left\{ \begin{bmatrix} \boldsymbol{c}_1 \\ \vdots \\ \boldsymbol{c}_s \end{bmatrix} : \boldsymbol{c}_j \in \mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; \boldsymbol{n}, k] \right\} \subseteq \mathbb{F}_{q^m}^{s \times n}.$$

The minimum distance of $\mathrm{ILRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell, s; \boldsymbol{n}, k]$ *is* $d_{\min} = n - k + 1$.

Based on Definition 2.14 and Definition 2.13, we can characterize the structure of codewords in $\mathrm{ILRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell, s; \boldsymbol{n}, k]$. Each codeword $\boldsymbol{C}$ in this code can be expressed as

$$\boldsymbol{C} = \left[\boldsymbol{C}^{(1)} \mid \boldsymbol{C}^{(2)} \mid \ldots \mid \boldsymbol{C}^{(\ell)} \right], \tag{2.36}$$

where

$$\boldsymbol{C}^{(i)} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{c}_1^{(i)} \\ \boldsymbol{c}_2^{(i)} \\ \vdots \\ \boldsymbol{c}_s^{(i)} \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{\beta}^{(i)})_{\xi_i} \\ f_2(\boldsymbol{\beta}^{(i)})_{\xi_i} \\ \vdots \\ f_s(\boldsymbol{\beta}^{(i)})_{\xi_i} \end{bmatrix} \in \mathbb{F}_{q^m}^{s \times n_i},$$

for all $i \in \{1, \ldots, \ell\}$. Define the vector $\boldsymbol{f}$ that contains all message polynomials as

$$\boldsymbol{f} \stackrel{\text{def}}{=} [f_1, f_2, \ldots, f_s] \in \mathbb{F}_{q^m}[x; \sigma, \delta]_{<k}^s.$$

We use the shorthand notation for the codeword matrix evaluated at the corresponding message polynomials, as in (2.36) as

$$\boldsymbol{C}(\boldsymbol{f}) \stackrel{\text{def}}{=} \left[\boldsymbol{C}^{(1)}(\boldsymbol{f}) \mid \boldsymbol{C}^{(2)}(\boldsymbol{f}) \mid \ldots \mid \boldsymbol{C}^{(\ell)}(\boldsymbol{f}) \right] \in \mathrm{ILRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell, s; \boldsymbol{n}, k].$$

2.7 Remark on the Notation of Complexity

In this thesis, we use the big-O notation, also known as Bachmann–Landau notation [Knu82] and denoted as $O(\cdot)$, to describe the asymptotic runtime of algorithms. Additionally, we use the soft-O notation $\widetilde{O}(\cdot)$, where $f(n) \in \widetilde{O}(g(n))$ if there exists a k such that $f(n) \in O\big(g(n) \log^k g(n)\big)$. This notation is coincides with the big-O notation but ignores logarithmic factors.

For computational complexity regarding skew polynomials, we focus on those with zero derivations, as explicit results are available in the literature. The isomorphism between $\mathbb{F}_{q^m}[x; \sigma, \delta]$ and $\mathbb{F}_{q^m}[x; \sigma]$ (see Section 2.5.2 and [Liu16; Mar18]) allows us to derive complexity bounds even for nonzero derivations.

We denote by $\mathfrak{p}(n)$ the cost of multiplying two skew polynomials from $\mathbb{F}_{q^m}[x; \sigma]$ of

degree n. The currently best known cost bounds for $\mathfrak{p}(n)$ are

$$\mathfrak{p}(n) \in O\Big(n^{\min\left\{\frac{\zeta+1}{2}, 1.635\right\}}\Big),$$

operations in $\mathbb{F}_{q^m}$ (see [PW18]), and

$$\mathfrak{p}(n) \in \widetilde{O}\Big(\min\Big\{n^{\zeta-2}m^2,\, nm^{\zeta-1}\Big\}\Big),$$

operations in $\mathbb{F}_q$ (see [CL17b; CL17a]). Here, ζ denotes the matrix multiplication exponent, defined as the infimum of values $\zeta_0 \in [2, 3]$ for which an algorithm exists to multiply $n \times n$ matrices over $\mathbb{F}_{q^m}$ in $O(n^{\zeta_0})$ operations. The best-known bound is currently $\zeta < 2.37286$ (see [AW21]).

Notably, the following skew polynomial operations in $\mathbb{F}_{q^m}[x;\sigma]$ can be performed in $\widetilde{O}(\mathfrak{p}(n))$:

- Left/right division of two skew polynomials of degree at most n,
- Generalized operator / remainder evaluation of a skew polynomial of degree at most n at n elements from $\mathbb{F}_{q^m}$,
- Computation of the minimal polynomials $M^{\mathrm{op}}_{\mathcal{B}}(x)_{\boldsymbol{a}}$ for $|\mathcal{B}| \leq n$ w.r.t. the remainder and generalized operator evaluation, respectively,
- Computation of the lclm (see [CL17a, Theorem 3.2.7]).

While these notations provide asymptotic approximations and omit constant factors, they are useful for estimating performance with large input sizes. In some parts of the thesis, we evaluate or plot complexities using these notations. For finite input sizes, we emphasize that these values are approximations, as the exact complexities depend on the specific implementation details of the algorithms and the underlying hardware.

3

Efficient Decoding of Interleaved Linearized Reed–Solomon Codes

In code-based cryptography, decryption typically involves a decoding step where the error introduced in encryption is corrected to retrieve the original message. To enable efficient error-correction algorithms that support the decryption processes in potential code-based cryptosystems built on sum-rank-metric codes, we focus on the decoding of ILRS codes. As discussed in Chapter 1 interleaved structures, such as ILRS codes, offer a promising approach to improve key sizes by extending the decoding radius.

In this chapter, we revisit key decoding concepts for ILRS codes in Section 3.1. We then provide an overview of weak Popov forms and Gröbner bases in Section 3.2. Next, in Section 3.3, we explore the skew Kötter–Nielsen–Høholdt interpolation over skew polynomial rings and the corresponding algorithm for solving the interpolation step in interpolation-based decoding. In Section 3.4, we introduce an optimized version of this algorithm that matches the best-known asymptotic complexity while eliminating the need for pre-processing and specific requirements on the interpolation points. Finally, in Section 3.5, we summarize, provide further discussion, and outline future research directions.

The content presented in the first parts of the chapter serves primarily as a review of well-established decoding concepts for ILRS codes. Section 3.4 is based on prior works, including [BJPR21] and [BJR24], both of which the author of this dissertation co-authored. In these works, the author contributed to discussions and the overall development of these papers. Furthermore, Section 3.5 is largely based on [BJR24], published in *Designs, Codes and Cryptography* in 2023 and the conference version [BJR22]. Specifically, the author made significant contributions to the results shown in Section 3.4.2, which is about precomputing methods for the Skew Kötter–Nielsen–Høholdt interpolation algorithm.

3.1 Known Decoding Approaches

In this section, we provide a brief and comprehensive overview of three prominent decoding approaches for (interleaved) LRS codes: syndrome-based decoding (non-interleaved) LRS codes [HBP22], the Loidreau–Overbeck decoding method for ILRS codes [Loi06] and interpolation-based decoding [BP22] for ILRS codes.

3.1.1 Syndrome-Based Decoding

Syndrome-based decoding is a common method for error correction using linear codes, including RS codes [Mas69; Gao03], Gabidulin codes [Gab85; Rot91; PT91; Gab92; RP04b] and also LRS codes [HBP22].

Syndrome decoding can also be extended to interleaved versions of these code families. For example, interleaved Gabidulin codes are addressed in [SB10], and for ILRS codes, syndrome decoding has been considered in the non-interleaved case [HBP22], with potential extensions to interleaved codes[1].

The general process involves three main steps:

- **Syndrome calculation:** The syndrome is computed by multiplying the received word $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$ with the transpose of the parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$

$$\boldsymbol{s} = \boldsymbol{y} \cdot \boldsymbol{H}^\top = \boldsymbol{e} \cdot \boldsymbol{H}^\top \in \mathbb{F}_{q^m}^{(n-k)}.$$

  A zero syndrome indicates no errors, while a non-zero syndrome provides information about the error pattern.

- **Solving the key equation:** The next step is to solve a key equation that relates the syndrome to the error. For both RS and Gabidulin codes, this step involves solving a system of linear equations. Specialized algorithms, like the Berlekamp–Massey algorithm for RS codes and similar techniques for Gabidulin codes, can solve this system more efficiently [MS77; Gab85].

- **Finding the error locations and values:** Once the key equation is solved, the error locations and values are determined. In Gabidulin codes, this step can be more challenging than in RS codes [Wac16]. The error locations correspond to the roots of the error locator polynomial, and the error values are derived from the syndrome and the error positions.

[1] In [HBP22], syndrome decoding of LRS codes is considered for the non-interleaved case. The authors hint at the possibility of extending error-erasure decoding to interleaved LRS codes and discuss future work on the implications of errors and erasures in the skew metric, which is isomorphic to the sum-rank metric.

In Chapter 4, we revisit *syndrome decoding* in the context of the *rank metric*, particularly for space-symmetric errors. Additionally, for the *sum-rank metric*, syndrome decoding has been applied for efficient error-and-erasure decoding, as outlined in [HBP22]. For further details on syndrome decoding in the sum-rank metric, we refer the reader to this work.

3.1.2 The Loidreau–Overbeck Decoder

In [Loi06], Loidreau and Overbeck introduced a unique decoding method for interleaved Gabidulin codes. This subsection examines an adaptation of their approach for ILRS codes in the sum-rank metric, as presented in [BP22]. This algorithm enables decoding of interleaved codes beyond the unique decoding radius of the underlying constituent code, at the cost of a nonzero probability of decoding failure. However, the probability of successful decoding in general remains high. Specifically, the decoder can correct errors with sum-rank weight w up to

$$w \leq \tau_{\mathcal{L}},$$

where

$$\tau_{\mathcal{L}} \stackrel{\text{def}}{=} \frac{s}{s+1}(n-k). \tag{3.1}$$

A connection between this decoder and the Metzner–Kapturowski-like decoder will be established in Chapter 5.

Assume a codeword $\boldsymbol{C} \in \mathrm{ILRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell, s; \boldsymbol{n}, k]$ of an s-interleaved ILRS code is transmitted over a sum-rank error channel, resulting in the corrupted matrix

$$\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E},$$

where $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{E}) = w$.

The received matrix $\boldsymbol{Y}$ and the error matrix $\boldsymbol{E}$ are defined as follows

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_s \end{bmatrix} \in \mathbb{F}_{q^m}^{s \times n} \text{ and } \boldsymbol{E} = \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_s \end{bmatrix} \in \mathbb{F}_{q^m}^{s \times n},$$

with each row $\boldsymbol{e}_i \in \mathbb{F}_{q^m}^n$ for $i \in \{1, \ldots, s\}$.

To proceed with the decoding algorithm, we construct a matrix $\boldsymbol{L}$ as

$$\boldsymbol{L} \stackrel{\text{def}}{=} \begin{bmatrix} \mathfrak{M}_{n-w-1}(\boldsymbol{\beta})_{\boldsymbol{\xi}} \\ \mathfrak{M}_{n-w-k}(\boldsymbol{y}_1)_{\boldsymbol{\xi}} \\ \mathfrak{M}_{n-w-k}(\boldsymbol{y}_2)_{\boldsymbol{\xi}} \\ \vdots \\ \mathfrak{M}_{n-w-k}(\boldsymbol{y}_s)_{\boldsymbol{\xi}} \end{bmatrix} \in \mathbb{F}_{q^m}^{(n-w-1+s(n-k-w))\times n} \tag{3.2}$$

from the received matrix $\boldsymbol{Y}$ and the codeword parameters $\boldsymbol{\beta}$ and $\boldsymbol{\xi}$. Observe that

$$\mathcal{R}_{q^m}(\boldsymbol{L}) = \mathcal{R}_{q^m}\left(\begin{bmatrix} \mathfrak{M}_{n-w-1}(\boldsymbol{\beta})_{\boldsymbol{\xi}} \\ \mathfrak{M}_{n-w-k}(\boldsymbol{e}_1)_{\boldsymbol{\xi}} \\ \mathfrak{M}_{n-w-k}(\boldsymbol{e}_2)_{\boldsymbol{\xi}} \\ \vdots \\ \mathfrak{M}_{n-w-k}(\boldsymbol{e}_s)_{\boldsymbol{\xi}} \end{bmatrix} \right).$$

If $\mathrm{rk_{q^m}}(\boldsymbol{L}) = n-1$, then the $\mathbb{F}_{q^m}$-dimension of the kernel of $\boldsymbol{L}$ is 1, and there exists a nonzero vector $\boldsymbol{v} \in \ker(\boldsymbol{L})_{\mathbb{F}_{q^m}} \setminus \{0\} \subseteq \mathbb{F}_{q^m}^n$.

Partition this vector $\boldsymbol{v}$ into blocks

$$\boldsymbol{v} = \left[\boldsymbol{v}^{(1)} \mid \boldsymbol{v}^{(2)} \mid \ldots \mid \boldsymbol{v}^{(\ell)}\right],$$

with $\boldsymbol{v}^{(i)} \in \mathbb{F}_{q^m}^{n_i}$ for each $i \in \{1, \ldots, \ell\}$.

It can be shown that $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{v}) = w$ (see [BP22]), and thus $\mathrm{rk_q}\left(\boldsymbol{v}^{(i)}\right) = w_i$ for all $i \in \{1, \ldots, \ell\}$.

For each $\boldsymbol{v}^{(i)}$, find a transformation matrix $\boldsymbol{T}^{(i)} \in \mathbb{F}_q^{n_i \times n_i}$ such that the $n_i - w_i$ leftmost entries of $\boldsymbol{v}^{(i)}\boldsymbol{T}^{(i)}$ are zero

$$\tilde{\boldsymbol{v}}^{(i)} = \boldsymbol{v}^{(i)}\boldsymbol{T}^{(i)} = [0, \ldots, 0, \tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_{w_i}] \in \mathbb{F}_{q^m}^{n_i},$$

for all $i \in \{1, \ldots, \ell\}$.

This implies that the rightmost columns of the matrix $\boldsymbol{E}^{(i)}\left(\boldsymbol{T}^{(i)}\right)^{-\top}$ are zero. As a consequence, the rightmost columns of $\boldsymbol{Y}^{(i)}\left(\boldsymbol{T}^{(i)}\right)^{-\top}$ correspond to non-corrupted positions. Therefore, the transmitted codeword can be reconstructed by (column) erasure decoding on the $n-w$ positions of

$$\boldsymbol{Y} \cdot \mathrm{diag}\left(\boldsymbol{T}^{(1)}, \boldsymbol{T}^{(2)}, \ldots, \boldsymbol{T}^{(\ell)}\right).$$

These positions are formed by the $n_i - w_i$ rightmost positions of each block with $i \in \{1, \ldots, \ell\}$. The reconstruction can be achieved, for example, using Lagrange

interpolation [BP22].

The success of this decoder depends on the probability that the matrix $\boldsymbol{L}$, as defined in (3.2), has $\mathbb{F}_{q^m}$-rank equal to $n-1$. By Theorem 2.1, we know that

$$\mathrm{rk_{q^m}}\left(\mathfrak{M}_{n-w-1}(\boldsymbol{\beta})_{\boldsymbol{\xi}}\right) = n-w-1,$$

so the probability of decoding success reduces to the probability that the matrix

$$\tilde{\boldsymbol{L}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathfrak{M}_{n-w-k}(\boldsymbol{e}_1)_{\boldsymbol{\xi}} \\ \mathfrak{M}_{n-w-k}(\boldsymbol{e}_2)_{\boldsymbol{\xi}} \\ \vdots \\ \mathfrak{M}_{n-w-k}(\boldsymbol{e}_s)_{\boldsymbol{\xi}} \end{bmatrix} \in \mathbb{F}_{q^m}^{s(n-w-k)\times n}, \tag{3.3}$$

has $\mathbb{F}_{q^m}$-rank equal to w. This probability is bounded as follows [BP22]

$$\Pr\left[\text{success}\right] = \Pr\left[\mathrm{rk_{q^m}}\left(\tilde{\boldsymbol{L}}\right) = w\right] \geq 1 - \gamma_q^{\ell+1} q^{-m((s+1)(\tau_{\mathcal{L}}-w)+1)},$$

with $\tau_{\mathcal{L}}$ as in (3.1) and γ_q as in (2.5).

3.1.3 Interpolation-Based Decoding of Interleaved LRS Codes

Interpolation-based decoding is a powerful decoding approach for interleaved algebraic codes such as IRS codes in the Hamming metric, interleaved Gabidulin codes in the rank metric and ILRS in the sum-rank metric. This method relies on constructing a multivariate polynomial that interpolates through points derived from the received codeword, followed by finding suitable roots of the polynomial that give rise to the transmitted message. The interpolation-based decoding procedure for ILRS codes can be summarized in two key steps:

1. **Interpolation:** Construct a multivariate skew polynomial $Q(x, y_1, \ldots, y_s)$ with a specific degree constraint, of the form

   $$Q(x, y_1, \ldots, y_s) = Q_0(x) + Q_1(x)y_1 + \cdots + Q_s(x)y_s,$$

   where $Q_i(x) \in \mathbb{F}_{q^m}[x;\sigma,\delta]$ for all $i \in \{0,\ldots,s\}$. This polynomial must vanish at a set of interpolation points derived from the code locators and the received matrix $\boldsymbol{Y}$.

2. **Root-Finding:** Determine all degree restricted skew polynomials $f_1, \ldots, f_s$ with $f_i \in \mathbb{F}_{q^m}[x;\sigma,\delta]_{<k}$ for all $i \in \{1,\ldots,s\}$, that satisfy the equation

   $$Q_0(x) + Q_1(x)f_1(x) + \cdots + Q_s(x)f_s(x) = 0. \tag{3.4}$$

Definition 3.1 (Generalized Operator Vector Evaluation Map)**.** *Given an interpolation point set*

$$\mathcal{P} = \{[p_{i,0}, p_{i,1}, \ldots, p_{i,s}] : i \in \{1, \ldots, n\}\} \subseteq \mathbb{F}_{q^m}^{s+1},$$

a vector $\boldsymbol{Q} = [Q_0, Q_1, \ldots, Q_s] \in \mathbb{F}_{q^m}[x; \sigma, \delta]^{s+1}$*, and a vector* $\boldsymbol{a} = [a_1, a_2, \ldots, a_n] \in \mathbb{F}_{q^m}^n$ *containing the generalized operator evaluation parameters, we define the generalized vector evaluation map* $\mathscr{E}_i^{op} \colon \mathbb{F}_{q^m}[x; \sigma, \delta]^{s+1} \to \mathbb{F}_{q^m}$ *as*

$$\mathscr{E}_i^{op}(\boldsymbol{Q})_{a_i} \stackrel{\text{def}}{=} \sum_{j=0}^{s} Q_j(p_{i,j})_{a_i} \quad \forall i \in \{1, \ldots, n\}. \tag{3.5}$$

Note that for $i \in \{1, \ldots, n\}$ *the evaluation map* $\mathscr{E}_i^{op}(\boldsymbol{Q})_{a_i}$ *depends on* $[p_{i,0}, p_{i,1}, \ldots, p_{i,s}]$ *but for simplicity of notation we omit this dependency and always assume that an evaluation map is defined with respect to some interpolation point set* $\mathcal{P}$*.*

Consider now an ILRS$[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell, s; \boldsymbol{n}, k]$ code with vectors $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_\ell] \in \mathbb{F}_{q^m}^\ell$ and $\boldsymbol{\beta} = [\boldsymbol{\beta}^{(1)} \mid \boldsymbol{\beta}^{(2)} \mid \ldots \mid \boldsymbol{\beta}^{(\ell)}] \in \mathbb{F}_{q^m}^n$ transmitted over an additive channel as in (2.29). The interpolation point set is given as

$$\mathcal{P} = \{[\beta_i, y_{1,i}, \ldots, y_{s,i}] : i \in \{1, \ldots, n\}\} \subset \mathbb{F}_{q^m}^{s+1},$$

where $y_{j,i}$ for $j \in \{1, \ldots, s\}$ and $i \in \{1, \ldots, n\}$ are the entries of the received matrix $\boldsymbol{Y}$. We define the vector $\boldsymbol{a}$ as follows

$$\boldsymbol{a} = \left[\boldsymbol{a}^{(1)} \mid \boldsymbol{a}^{(2)} \mid \ldots \mid \boldsymbol{a}^{(\ell)}\right] \in \mathbb{F}_{q^m}^n, \tag{3.6}$$

where for each $i \in \{1, \ldots, \ell\}$, we have

$$\boldsymbol{a}^{(i)} = \underbrace{[\xi_i, \xi_i, \ldots, \xi_i]}_{n_i \text{ times}} \in \mathbb{F}_{q^m}^{n_i}.$$

Given a vector $\boldsymbol{Q} = [Q_0, Q_1, \ldots, Q_s] \in \mathbb{F}_{q^m}[x; \sigma, \delta]^{s+1}$ of skew polynomials and a *weighting vector* $\boldsymbol{\omega} = [\omega_0, \omega_2, \ldots, \omega_s] \in \mathbb{Z}_{\geq 0}^{s+1}$, we define the $\boldsymbol{\omega}$-weighted degree of $\boldsymbol{Q}$ as

$$\deg_{\boldsymbol{\omega}}(\boldsymbol{Q}) \stackrel{\text{def}}{=} \max_{0 \leq j \leq n} \{\deg(Q_j) + \omega_j\}.$$

With the notion of the generalized operator vector evaluation map established, we are now equipped to define the interpolation problem for ILRS codes.

Problem 3.1 (Vector Interpolation Problem)**.** *Given the interleaving order* $s \in \mathbb{Z}_{\geq 0}$*, a* ***degree constraint*** $D \in \mathbb{Z}_{\geq 0}$*, a set of* $\mathbb{F}_{q^m}$*-linear vector evaluation maps, i.e.* $\mathcal{E}^{op} = \{\mathscr{E}_i^{op} : i \in \{1, \ldots, n\}\}$ *as defined in* (3.5)*, a vector* $\boldsymbol{a} = [a_1, a_2, \ldots, a_n] \in \mathbb{F}_{q^m}^n$ *as defined in* (3.6) *and a vector* $\boldsymbol{\omega} = [0, k-1, \ldots, k-1] \in \mathbb{Z}_{\geq 0}^{s+1}$*, compute a vector* $\boldsymbol{Q} \in \mathbb{F}_{q^m}[x; \sigma, \delta]^{s+1}$ *that satisfies:*

- $\mathscr{E}_i^{\mathrm{op}}(\boldsymbol{Q})_{a_i} = 0,\ \forall i \in \{1, \ldots, n\}$,
- $\deg_{\boldsymbol{w}}(\boldsymbol{Q}) < D$.

A nonzero solution of Problem 3.1 exists if the degree constraint satisfies [BP22]

$$D = \left\lceil \frac{n + s(k-1) + 1}{s+1} \right\rceil.$$

If the sum-rank weight of the error, $w = \mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{E})$, satisfies

$$w < \frac{s}{s+1}(n-k+1),$$

then, according to [BJR24, Theorem 2], a basis for a list $\mathcal{L}$ of candidate solutions of size

$$|\mathcal{L}| \leq q^{mk(s-1)},$$

can be found in polynomial time. This list $\mathcal{L}$ consists of all solutions to the root-finding problem, that means all polynomials

$$f_1, \ldots, f_s \in \mathbb{F}_{q^m}[x;\sigma]_{<k},$$

that satisfy (3.4). The list $\mathcal{L}$ also contains all possible message polynomial vectors $\boldsymbol{f} \in \mathbb{F}_{q^m}[x;\sigma,\delta]_{<k}^{s}$ corresponding to codewords $\boldsymbol{C} \in \mathrm{ILRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell, s; \boldsymbol{n}, k]$ that satisfy

$$d_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{C}(\boldsymbol{f}), \boldsymbol{Y}) < \frac{s}{s+1}(n-k+1).$$

The root-finding problem can be solved efficiently using the minimal approximant basis methods described in [BJPR19; BJPR21], requiring at most

$$\widetilde{O}\big(s^{\zeta}\mathfrak{p}(n)\big),$$

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

3.2 Weak Popov and Gröbner Bases

For vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^n$, we define the element-wise right modulo operation as

$$\boldsymbol{a} \bmod_{\mathrm{r}} \boldsymbol{b} \stackrel{\mathrm{def}}{=} \left[a_1 \bmod_{\mathrm{r}} b_1, a_2 \bmod_{\mathrm{r}} b_2, \ldots, a_n \bmod_{\mathrm{r}} b_n\right].$$

Similarly, we define the element-wise lclm for $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^n$ as

$$\mathrm{lclm}\left(\boldsymbol{a}, \boldsymbol{b}\right) \stackrel{\mathrm{def}}{=} \left[\mathrm{lclm}\left(a_1, b_1\right), \mathrm{lclm}\left(a_2, b_2\right), \ldots, \mathrm{lclm}\left(a_n, b_n\right)\right].$$

Furthermore, we introduce a $\boldsymbol{\omega}$-weighted monomial ordering $\prec_{\boldsymbol{\omega}}$ on $\mathbb{F}_{q^m}[x;\sigma,\delta]^n$. For $b_i, b_{i'} \in \mathbb{F}_{q^m} \setminus \{0\}$, we have

$$b_i x^i \boldsymbol{e}_j \prec_{\boldsymbol{\omega}} b_{i'} x^{i'} \boldsymbol{e}_{j'}, \tag{3.7}$$

if either $i + \omega_j < i' + \omega_{j'}$, or $i + \omega_j = i' + \omega_{j'}$ and $j < j'$, where $\boldsymbol{e}_j$ denotes the j-th unit vector over $\mathbb{F}_{q^m}[x;\sigma,\delta]$. This definition of $\prec_{\boldsymbol{\omega}}$ aligns with the $\boldsymbol{\omega}$-weighted term-over-position (TOP) ordering as described in [AL94].

Consider a nonzero vector $\boldsymbol{a} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^n$ of skew polynomials and a weighting vector $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_n] \in \mathbb{Z}_{\geq 0}^n$. We define the $\boldsymbol{\omega}$-pivot index $\mathrm{Ind}_{\boldsymbol{w}}(\boldsymbol{a})$ of $\boldsymbol{a}$ as the largest index j, where $1 \leq j \leq n$, satisfying

$$\deg(a_j) + \omega_j = \deg_{\boldsymbol{\omega}}(\boldsymbol{a}).$$

For any nonzero vector $\boldsymbol{a} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^n$, we define its *leading term* $\mathrm{LT}(\boldsymbol{a})$ as the maximal term $a_{i,j}x^j$ under the $\prec_{\boldsymbol{\omega}}$ ordering. In this case, j corresponds to the $\boldsymbol{\omega}$-pivot index of $\boldsymbol{a}$.

A matrix $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{a\times b}$ with $a \leq b$ is said to be in (row) $\boldsymbol{\omega}$-ordered weak Popov form if the $\boldsymbol{\omega}$-pivot indices of its rows form a strictly increasing sequence with respect to the row index [MS03].

A module over a ring, such as a $\mathbb{F}_{q^m}[x;\sigma,\delta]$-module, is a generalization of the concept of a vector space, where scalars are elements of a ring instead of a field.

Since $\mathbb{F}_{q^m}[x;\sigma,\delta]$ is not commutative, we distinguish between left and right modules.

Definition 3.2 (Left Module)**.** *A left $\mathbb{F}_{q^m}[x;\sigma,\delta]$-module $\mathcal{M}$ is an abelian group under addition with a scalar multiplication $\mathbb{F}_{q^m}[x;\sigma,\delta] \times \mathcal{M} \to \mathcal{M}$ satisfying:*

1. *$u(\boldsymbol{a}+\boldsymbol{b}) = u\boldsymbol{a} + u\boldsymbol{b}$ for all $u \in \mathbb{F}_{q^m}[x;\sigma,\delta]$ and $\boldsymbol{a},\boldsymbol{b} \in \mathcal{M}$,*
2. *$(u+v)\boldsymbol{a} = u\boldsymbol{a} + v\boldsymbol{a}$ for all $u, v \in \mathbb{F}_{q^m}[x;\sigma,\delta]$ and $\boldsymbol{a} \in \mathcal{M}$,*
3. *$(uv)\boldsymbol{a} = u(v\boldsymbol{a})$ for all $u, v \in \mathbb{F}_{q^m}[x;\sigma,\delta]$ and $\boldsymbol{a} \in \mathcal{M}$.*

A right $\mathbb{F}_{q^m}[x;\sigma,\delta]$-module is similarly defined, but the scalar multiplication is from the right, i.e., elements of the module are multiplied by elements of $\mathbb{F}_{q^m}[x;\sigma,\delta]$ from the right.

A free $\mathbb{F}_{q^m}[x;\sigma,\delta]$-module is a module that has a basis, a set of $\mathbb{F}_{q^m}[x;\sigma,\delta]$-linearly independent elements that span the module. The rank of this module is the number of elements in its basis.

We now consider specific bases for left $\mathbb{F}_{q^m}[x;\sigma,\delta]$-modules.

Definition 3.3 ($\boldsymbol{\omega}$-Ordered Weak Popov Basis [BJPR21])**.** *Let $\mathcal{M}$ be a left $\mathbb{F}_{q^m}[x;\sigma,\delta]$-submodule of $\mathbb{F}_{q^m}[x;\sigma,\delta]^b$. For $\boldsymbol{\omega} \in \mathbb{Z}_{\geq 0}^a$, a left $\boldsymbol{\omega}$-ordered weak Popov basis for $\mathcal{M}$ is a full-rank matrix $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{a\times b}$ satisfying:*

1. $\boldsymbol{A}$ *is in* $\boldsymbol{\omega}$*-ordered weak Popov form,*

2. *The rows of* $\boldsymbol{A}$ *form a basis of* $\mathcal{M}$*.*

We now explore the relationship between $\boldsymbol{\omega}$-ordered weak Popov bases and Gröbner bases with respect to $\prec_{\boldsymbol{\omega}}$ for left $\mathbb{F}_{q^m}[x;\sigma,\delta]$-submodules. This connection is well-established for ordinary commutative polynomial rings (see, e.g., [Fit95; Ale02; KRT07; Nie13; Nei16]). For skew polynomial rings, this relationship was derived in [Mid12, Chapter 6] and applied in [BJPR21].

For a concise introduction to Gröbner bases, readers are referred to [Stu05], while a comprehensive study can be found in [CLO92].

Let $h_1, h_2, \ldots, h_k$ be elements of a left module. The span of these elements, denoted by $\langle h_1, h_2, \ldots, h_k\rangle$, is defined as

$$\langle h_1, h_2, \ldots, h_k\rangle = \left\{ \sum_{i=1}^{k} r_i h_i : r_i \in \mathbb{F}_{q^m}[x;\sigma,\delta] \right\}.$$

Definition 3.4 (Gröbner Basis [CLO92])**.** *Let* $\mathcal{M}$ *be a left* $\mathbb{F}_{q^m}[x;\sigma,\delta]$*-submodule. A subset* $\mathcal{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_\nu\} \subset \mathcal{M}$ *is a Gröbner basis for* $\mathcal{M}$ *under* $\prec_{\boldsymbol{\omega}}$ *if the leading terms of* $\mathcal{B}$ *generate a left module containing all leading terms in* $\mathcal{M}$*, i.e., if*

$$\langle LT(\boldsymbol{b}_1), LT(\boldsymbol{b}_2), \ldots, LT(\boldsymbol{b}_\nu)\rangle = \langle LT(\mathcal{M})\rangle,$$

where $LT(\mathcal{M})$ *is the set of all leading terms of elements in* $\mathcal{M}$*, determined by the order* $\prec_{\boldsymbol{\omega}}$ *applied to the polynomial entries of the module vectors.*

A Gröbner basis for an $\mathbb{F}_{q^m}[x;\sigma,\delta]$-submodule $\mathcal{M}$ does not necessarily form a *minimal* generating set for $\mathcal{M}$, since any larger set within $\mathcal{M}$ that includes a Gröbner basis will also qualify as a Gröbner basis (see [CLO92; Stu05]). The following definition introduces a minimality condition on the size of Gröbner bases for an $\mathbb{F}_{q^m}[x;\sigma,\delta]$-submodule with respect to the order $\prec_{\boldsymbol{\omega}}$.

Definition 3.5 (Minimal Gröbner Basis [CLO92])**.** *For a given monomial ordering* $\prec_{\boldsymbol{\omega}}$*, a Gröbner basis* $\mathcal{B}$ *for a left* $\mathbb{F}_{q^m}[x;\sigma,\delta]$*-submodule* $\mathcal{M}$ *is minimal if for all* $\boldsymbol{p} \in \mathcal{B}$*, the leading term* $LT(\boldsymbol{p})$ *is not in the module* $\langle LT(\mathcal{B} \setminus \{\boldsymbol{p}\})\rangle$*.*

A minimal Gröbner basis $\mathcal{B}$ with respect to $\prec_{\boldsymbol{\omega}}$ is referred to as a *reduced* Gröbner basis if its leading terms are normalized, and no monomial of any element $\boldsymbol{p} \in \mathcal{B}$ is contained within $\langle \mathrm{LT}(\mathcal{B} \setminus \{\boldsymbol{p}\})\rangle$.

Theorem 6.29 in [Mid12] illustrates the relationship between the stronger $\boldsymbol{w}$-ordered Popov form and the corresponding *reduced* Gröbner basis with respect to $\prec_{\boldsymbol{\omega}}$. This reasoning can also be applied to establish a connection between the $\boldsymbol{w}$-ordered weak Popov form and the minimal Gröbner basis for $\prec_{\boldsymbol{\omega}}$.

For any given module monomial order $\prec$ and a basis $\mathcal{B} \subset \mathbb{F}_{q^m}[x;\sigma,\delta]^n$ of a submodule $\mathcal{M}$, there is an efficient approach to find a weight vector $\boldsymbol{\omega}$ and a column permutation

P such that the weak Popov form under $\prec_{\boldsymbol{\omega}}$ of $P(\mathcal{M})$ corresponds to the P-permuted *minimal* Gröbner basis of $\mathcal{M}$ with respect to $\prec$ (see [Nei16, Chapter 1.3.4]).

3.3 Skew Kötter–Nielsen–Høhold Interpolation over Skew Polynomial Rings

We now consider the skew Kötter–Nielsen–Høholdt (KNH) interpolation introduced by Liu et al. [LMK14], which extends the KNH interpolation concept from ordinary polynomial rings [WMW05] to the skew polynomial domain. Notably, when the automorphism σ is the Frobenius automorphism and the derivation δ is zero, $\mathbb{F}_{q^m}[x;\sigma,\delta]$ becomes isomorphic to the ring of linearized polynomials. Consequently, the KNH variant for linearized polynomial rings proposed by Xie et al. [XYS11] can be considered a specific instance of the more general approach presented in [LMK14].

In this section, we consider the general case of the interpolation algorithm, which works for any evaluation map. Later, we will apply this to ILRS codes using the evaluation maps defined in Definition 3.1.

As input to our problem, we consider n many $\mathbb{F}_{q^m}$-linear skew vector evaluation maps[2] $\mathscr{E}_i$, i.e.

$$\mathscr{E}_i : \mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1} \to \mathbb{F}_{q^m}, \tag{3.8}$$

where $i \in \{1,\dots,n\}$. Here, n represents the number of interpolation constraints, which will later coincide with the codeword length in the context of ILRS codes. The parameter s is an interpolation parameter and later coincides with the interleaving order of $ILRS$ codes.

For each skew vector evaluation map $\mathscr{E}_i$ we define the kernels

$$\mathcal{K}_i \stackrel{\text{def}}{=} \{\boldsymbol{Q} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1} : \mathscr{E}_i(\boldsymbol{Q}) = 0\}, \quad \forall i \in \{1,\dots,n\}.$$

For $i \in \{1,\dots,n\}$ the intersection $\overline{\mathcal{K}}_i \stackrel{\text{def}}{=} \mathcal{K}_1 \cap \mathcal{K}_2 \cap \ldots \cap \mathcal{K}_i$ contains all vectors from $\mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1}$ that are mapped to zero under $\mathscr{E}_1, \mathscr{E}_2, \dots, \mathscr{E}_i$, i.e.

$$\overline{\mathcal{K}}_i = \{\boldsymbol{Q} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1} : \mathscr{E}_j(\boldsymbol{Q}) = 0, \forall j \in \{1,\dots,i\}\}.$$

Assuming that each $\overline{\mathcal{K}}_i$ (for all $i \in \{1,\dots,n\}$) is a left $\mathbb{F}_{q^m}[x;\sigma,\delta]$-submodule (see [LMK14]), we can state the general skew polynomial vector interpolation problem.

[2] Previous works like [WMW05; LMK14; XYS11] use linear functionals for each interpolation point. Our approach defines evaluation maps on skew polynomial vectors, which is equivalent when $\mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1}$ is viewed as an $\mathbb{F}_{q^m}$-vector space.

Problem 3.2 (General Vector Interpolation Problem)**.** *Given the integer $s \in \mathbb{Z}_{\geq 0}$, a tuple of $\mathbb{F}_{q^m}$-linear vector evaluation maps $\mathcal{E} = (\mathscr{E}_1, \ldots, \mathscr{E}_n)$ and a vector $\boldsymbol{\omega} \in \mathbb{Z}_{\geq 0}^{s+1}$, compute a $\boldsymbol{\omega}$-ordered weak-Popov Basis for the left $\mathbb{F}_{q^m}[x;\sigma,\delta]$-module*

$$\overline{\mathcal{K}}_n = \{\boldsymbol{b} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1} : \mathscr{E}_i(\boldsymbol{b}) = 0, \forall i \in \{1,\ldots,n\}\}.$$

To solve Problem 3.2, we can use a modified version of the multivariate skew KNH interpolation algorithm from [LMK14]. The main adjustment lies in the output: rather than returning a single minimal polynomial vector, we adapt [LMK14, Algorithm 1] to return a $\boldsymbol{w}$-ordered weak Popov basis for the entire interpolation module $\overline{\mathcal{K}}_n$. This method resembles the approach employed in [Bar17] for linearized polynomial rings.

The modified multivariate skew KNH interpolation is summarized in Algorithm 1.

Algorithm 1: Modified Skew KNH Interpolation

Input : A tuple $(\mathscr{E}_1, \ldots, \mathscr{E}_n)$ of vector evaluation maps
A weighting vector $\boldsymbol{\omega} = [w_1, w_2, \ldots, w_{s+1}] \in \mathbb{Z}_{\geq 0}^{s+1}$

Output : A $\boldsymbol{\omega}$-ordered weak-Popov Basis $\boldsymbol{B} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{(s+1)\times(s+1)}$ for $\overline{\mathcal{K}}_n$

1 **Initialize:** $\boldsymbol{B} = \boldsymbol{I}_{s+1} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{(s+1)\times(s+1)}$

2 **for** $i \leftarrow 1$ **to** n **do**
3 **for** $j \leftarrow 1$ **to** $s+1$ **do**
4 $\Delta_j \leftarrow \mathscr{E}_i(\boldsymbol{b}_j)$
5 $\mathcal{J} \leftarrow \{j : \Delta_j \neq 0\}$
6 **if** $\mathcal{J} \neq \emptyset$ **then**
7 $j^* \leftarrow \min_{j\in\mathcal{J}}\{\arg\min_{j\in\mathcal{J}}\{\deg_{\boldsymbol{\omega}}(\boldsymbol{b}_j)\}\}$
8 $\boldsymbol{b}^* \leftarrow \boldsymbol{b}_{j^*}$
9 **for** $j \in \mathcal{J}$ **do**
10 **if** $j = j^*$ **then**
11 $\boldsymbol{b}_j \leftarrow \left(x - \frac{\mathscr{E}_i(x\boldsymbol{b}^*)}{\Delta_{j^*}}\right)\boldsymbol{b}^*$ `/* degree-increasing step */`
12 **else**
13 $\boldsymbol{b}_j \leftarrow \boldsymbol{b}_j - \frac{\Delta_j}{\Delta_{j^*}}\boldsymbol{b}^*$ `/* cross-evaluation step */`

14 **return** $\boldsymbol{B}$

Note that

$$\min_{j\in\mathcal{J}}\{\arg\min_{j\in\mathcal{J}}\{\deg_{\boldsymbol{\omega}}(\boldsymbol{b}_j)\}\},$$

in Line 7 returns the smallest index $j \in \mathcal{J}$ to break ties, i.e. the index j of the minimal vector $\boldsymbol{b}_j$ w.r.t. $\prec_{\boldsymbol{\omega}}$ for which $\Delta_j \neq 0$ (see (3.7)).

In each iteration of Algorithm 1 (and so [LMK14, Algorithm 1]) there are three possible update steps:

1. **No update**: The vector $\boldsymbol{b}_j$ is not updated if $\boldsymbol{b}_j$ is in the kernel $\overline{\mathcal{K}}_i$ already, i.e. if $\Delta_j = \mathscr{E}_i(\boldsymbol{b}_j) = 0$.

2. **Cross-evaluation** (or *order-preserving* [LMK14]) update: For any $\boldsymbol{b}_j$ that is not minimal w.r.t. $\prec_{\boldsymbol{\omega}}$ (i.e. $j \neq j^*$) the cross-evaluation update (Line 13) is performed such that

$$\mathscr{E}_i\left(\boldsymbol{b}_j - \frac{\Delta_j}{\Delta_{j^*}}\boldsymbol{b}^*\right) = \mathscr{E}_i(\boldsymbol{b}_j) - \frac{\mathscr{E}_i(\boldsymbol{b}_j)}{\mathscr{E}_i(\boldsymbol{b}_{j^*})}\mathscr{E}_i(\boldsymbol{b}_{j^*}) = 0.$$

   Note that the ($\boldsymbol{w}$-weighted) degree of $\boldsymbol{b}_j$ is not increased by this update.

3. **Degree-increasing** (or **order-increasing** [LMK14]) update: For the minimal vector $\boldsymbol{b}_{j^*} \stackrel{\text{def}}{=} \boldsymbol{b}^*$ w.r.t. $\prec_{\boldsymbol{\omega}}$ the degree-increasing update (Line 11) is performed such that

$$\mathscr{E}_i\left(\left(x - \frac{\mathscr{E}_i(x\boldsymbol{b}^*)}{\Delta_{j^*}}\right)\boldsymbol{b}^*\right) = \mathscr{E}_i(x\boldsymbol{b}^*) - \frac{\mathscr{E}_i(x\boldsymbol{b}^*)}{\mathscr{E}_i(\boldsymbol{b}^*)}\mathscr{E}_i(\boldsymbol{b}^*) = 0.$$

   The ($\boldsymbol{w}$-weighted) degree of $\boldsymbol{b}^*$ is increased by one in this case.

Define the sets

$$\mathcal{S}_j \stackrel{\text{def}}{=} \{\boldsymbol{Q} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1} : \operatorname{Ind}_{\boldsymbol{w}}(\boldsymbol{Q}) = j\} \cup \{\boldsymbol{0}\},$$

and

$$\mathcal{X}_{i,j} \stackrel{\text{def}}{=} \overline{\mathcal{K}}_i \cap \mathcal{S}_j,$$

for all $i \in \{1,\ldots,n\}$ and $j \in \{1,\ldots,s+1\}$. Note that $\mathcal{S}_j \cap \mathcal{S}_{j'} = \{\boldsymbol{0}\}$ for all $1 \leq j, j' \leq s+1$.

The following result from [LMK14] is fundamental for proving the correctness of Algorithm 1.

Theorem 3.1 ([LMK14, Theorem 5])**.** *Assume that $\overline{\mathcal{K}}_i$ are left $\mathbb{F}_{q^m}[x;\sigma,\delta]$-submodules for all $i \in \{1,\ldots,n\}$. Then after each iteration i of Algorithm 1, the updated $\boldsymbol{b}_j$ is a minimum w.r.t. $\prec_{\boldsymbol{\omega}}$ in $\mathcal{X}_{i,j} = \overline{\mathcal{K}}_i \cap \mathcal{S}_j$ for all $j \in \{1,\ldots,s+1\}$.*

In other words, after the i-th iteration each $\boldsymbol{b}_j$ has $\operatorname{Ind}_{\boldsymbol{w}}(\boldsymbol{b}) = j$ and the minimal $\boldsymbol{w}$-weighted degree among all vectors in $\mathcal{X}_{i,j}$. Therefore, after the i-th iteration, the matrix $\boldsymbol{B}$ is a $\boldsymbol{w}$-ordered weak Popov basis for $\overline{\mathcal{K}}_i$.

Lemma 3.1 (Correctness of Algorithm 1)**.** *Algorithm 1 is correct and provides a solution to the general vector interpolation problem in Problem 3.2.*

Proof. The update steps of Algorithm 1 and [LMK14, Algorithm 1] are equivalent and therefore we have by [LMK14, Theorem 5] that after the i-th iteration each $\boldsymbol{b}_j \in \overline{\mathcal{K}}_i$. We now show that after the i-th iteration of Algorithm 1 the matrix $\boldsymbol{B}$ is a $\boldsymbol{w}$-ordered weak Popov basis for $\overline{\mathcal{K}}_i$. By Theorem 3.1 ([LMK14, Theorem 5]) each $\boldsymbol{b}_j$ has the minimal $\boldsymbol{w}$-weighted degree among all polynomials in $\mathcal{X}_{i,j}$, which implies that the $\boldsymbol{w}$-pivot indices of $\boldsymbol{b}_0, \ldots, \boldsymbol{b}_s$ are increasing and distinct. Now assume that there exists a vector $\boldsymbol{p} \in \overline{\mathcal{K}}_i$ that can not be represented by a $\mathbb{F}_{q^m}[x;\sigma,\delta]$-linear combination of the form

$$\boldsymbol{p} = \sum_{j=0}^{s} a_j \boldsymbol{b}_j,$$

for some $a_j \in \mathbb{F}_{q^m}[x;\sigma,\delta]$. Then we must have that $\boldsymbol{p}$ can be written as

$$\boldsymbol{p} = \boldsymbol{r} + \sum_{j=0}^{s} a_j \boldsymbol{b}_j,$$

where $\deg_{\boldsymbol{w}}(\boldsymbol{r}) < \min_j\{\deg_{\boldsymbol{w}}(\boldsymbol{b}_j)\}$. This contradicts that $\boldsymbol{b}_j$ is a minimum w.r.t. $\prec_{\boldsymbol{\omega}}$ in $\mathcal{X}_{i,j}$ since $\mathrm{Ind}_{\boldsymbol{w}}(\boldsymbol{r}) \in \{0, \ldots, s\}$. Therefore we conclude that after the i-th iteration $\boldsymbol{B}$ is a $\boldsymbol{w}$-ordered weak Popov basis for $\overline{\mathcal{K}}_i$. □

Proposition 3.1 (Computational Complexity of Algorithm 1)**.** *The complexity of Algorithm 1 is dominated by the complexity of:*

- $O(sn)$ *evaluation maps* $\mathscr{E}_i$ *applied to a vector from* $\mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1}_{\leq n}$,
- n *multiplications involving a monic degree-1 skew polynomial and a vector from* $\mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1}_{\leq n}$ *(degree-increasing step),*
- $O(sn)$ *multiplications of an element from* $\mathbb{F}_{q^m}$ *with a vector from* $\mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1}_{\leq n}$ *(cross-evaluation step).*

Proof. In each of the n iterations we have:

- $s+2$ evaluation maps $\mathscr{E}_i$ applied to a vector from $\mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1}_{\leq n}$ (Line 4),
- one product of a skew polynomial of degree 1 with a vector from $\mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1}_{\leq n}$ (degree-increasing step in Line 11),
- s multiplications of an element from $\mathbb{F}_{q^m}$ with a vector from $\mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1}_{\leq n}$ (cross-evaluation step in Line 13),
- $s+1$ inversions/divisions in $\mathbb{F}_{q^m}$.

□

3.4 Fast Skew Kötter–Nielsen–Høhold Interpolation

In [Nie14], a fast divide-and-conquer (D&C) variant of the Kötter interpolation was introduced for the Guruswami–Sudan decoder of Reed–Solomon codes. We extend this approach to the skew KNH interpolation proposed in [LMK14].

The core strategy involves decomposing Problem 3.2 into a tree-like structure, typical of D&C approaches. This tree represents a hierarchy of increasingly smaller subproblems. At the lowest level, leaf nodes correspond to the smallest subproblems, each associated with a linear functional. Updates at this level are represented as skew polynomial matrices. Moving up the tree, inner nodes combine these updates through matrix multiplication. The algorithm's efficiency stems from a key insight: at any given node, only the intermediate basis's image on the linear functionals within that node's subtree is necessary, significantly reducing computational complexity.

We now present the general framework for the fast skew KNH interpolation algorithm. Its application to ILRS codes will be discussed in Section 3.4.3.

The transformations applied to the basis $\boldsymbol{B}$ during the inner loop of the i-th iteration in Algorithm 1 can be encapsulated by the matrix $\boldsymbol{U}$, defined as

$$\boldsymbol{U} \stackrel{\text{def}}{=} \left[\begin{array}{ccc|c|ccc} 1 & & & -\frac{\Delta_0}{\Delta_{j^*}} & & & \\ & \ddots & & \vdots & & & \\ & & 1 & -\frac{\Delta_{j^*-1}}{\Delta_{j^*}} & & & \\ & & & \left(x - \frac{\mathscr{E}_i(x\boldsymbol{b}_{j^*})}{\Delta_{j^*}}\right) & & & \\ & & & -\frac{\Delta_{j^*+1}}{\Delta_{j^*}} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -\frac{\Delta_s}{\Delta_{j^*}} & & & 1 \end{array}\right]. \tag{3.9}$$

After the i-th iteration, the resulting basis for $\overline{\mathcal{K}}_i$ is given by $\boldsymbol{UB}$. Note that when $\Delta_j = 0$, the corresponding row $\boldsymbol{b}_j$ remains unchanged, as the ratio Δ_j/Δ_{j^*} evaluates to zero.

3.4.1 Divide-and-Conquer Skew Kötter Interpolation

First, let us introduce the notation necessary for describing the subsequent algorithms. For $j \geq i$, let $\boldsymbol{m}_{[i,j]} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1}$ denote a skew polynomial vector associated with the index set $\{i, i+1, \ldots, j-1, j\}$, where $\boldsymbol{m}_{[i,i]} = \boldsymbol{m}_i$.

We define $\mathcal{P}$ as a globally accessible ordered set encompassing all possible skew polynomial vector elements $\boldsymbol{m}_{[i,j]}$ as

$$\mathcal{P} \stackrel{\text{def}}{=} \left\{ \boldsymbol{m}_{[0,n-1]}, \boldsymbol{m}_{[0,\lfloor n/2 \rfloor -1]}, \boldsymbol{m}_{[\lfloor n/2 \rfloor, n-1]}, \ldots, \boldsymbol{m}_0, \boldsymbol{m}_1, \ldots, \boldsymbol{m}_{n-1} \right\} \subseteq \mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1},$$

where $n \in \mathbb{N}$. This vector is assumed to be precomputed for the subsequent algorithms. The process of efficiently precomputing $\mathcal{P}$ is described in detail in Section 3.4.2.

For a tuple of evaluation maps $\mathcal{E} = (\mathscr{E}_1, \ldots, \mathscr{E}_n)$, we introduce a similar indexing convention to represent an ordered segment of $\mathcal{E}$ as

$$\mathcal{E}_{[i,j]} = (\mathscr{E}_i, \ldots, \mathscr{E}_j).$$

Depending on the considered interpolation problem, we will later on define the polynomial vectors $\boldsymbol{m}_{[i,j]}$ to contain minimal polynomials that depend on the interpolation points corresponding to the vector evaluation maps in $\mathcal{E}_{[i,j]}$.

In contrast to the general interpolation problem presented in Problem 3.2, which considers sets of evaluation maps, our approach employs the global set $\mathcal{P}$ and the segmentation notation. This notation facilitates the construction of the problem-solving tree within the D&C algorithm, providing a structured framework for the divide-and-conquer process.

To establish a general framework for the fast skew KNH interpolation, we introduce a key assumption. We demonstrate in Section 3.4.3 that this assumption holds for decoding ILRS codes.

Assumption 1. *Consider a tuple of linear functionals $\mathcal{E} = (\mathscr{E}_1, \ldots, \mathscr{E}_n)$ as defined in (3.8), and let $\mathcal{E}_{[i,j]} = (\mathscr{E}_i, \ldots, \mathscr{E}_j)$ be an ordered segment of $\mathcal{E}$ for $1 \leq i \leq j \leq n$.*

For any $\boldsymbol{Q} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1}$, we assume that the skew polynomial vector $\boldsymbol{m}_{[i,j]} \in \mathcal{P}$, containing minimal skew polynomials dependent on $\mathcal{E}_{[i,j]}$, satisfies

$$\mathscr{E}_l(\boldsymbol{Q}) = \mathscr{E}_l(\boldsymbol{Q} \bmod_{\mathrm{r}} \boldsymbol{m}_{[i,j]}), \quad \forall l \in \{i, \ldots, j\}.$$

Having established the necessary framework and assumptions, we now present an algorithm that forms a crucial component of our fast skew KNH interpolation method. Algorithm 2, named SkewInterpolatePoint, performs a key step in the interpolation process by updating a basis matrix to satisfy a single interpolation condition.

Algorithm 2: SkewInterpolatePoint

Input : A skew vector evaluation map $\mathscr{E}_i$
$\boldsymbol{B} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{(s+1)\times(s+1)}$
$\boldsymbol{d} = [d_1, d_2, \ldots, d_{s+1}] \in \mathbb{Z}_{\geq 0}^{s+1}$ s.t. $d_j = \deg_{\boldsymbol{\omega}}(\boldsymbol{b}_j) \quad \forall j \in \{1, \ldots, s+1\}$

Output : $\boldsymbol{T} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{(s+1)\times(s+1)}$ s.t. the rows of $\hat{\boldsymbol{B}} \stackrel{\text{def}}{=} \boldsymbol{T}\boldsymbol{B} \quad \forall \boldsymbol{\omega}$-ordered weak-Popov Basis for $\langle \boldsymbol{B} \rangle \cap \mathcal{K}_i$
$\hat{\boldsymbol{d}} = [\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_{s+1}] \in \mathbb{Z}_{\geq 0}^{s+1}$ s.t. $\hat{d}_j = \deg_{\boldsymbol{\omega}}(\hat{\boldsymbol{b}}_j) \quad \forall j \in \{1, \ldots, s+1\}$

1 $\hat{\boldsymbol{d}} \leftarrow \boldsymbol{d}$
2 **for** $j \leftarrow 0$ **to** s **do**
3 $\quad \Delta_j \leftarrow \mathscr{E}_i(\boldsymbol{b}_j)$
4 $\mathcal{J} \leftarrow \{j : d_j \neq 0\}$
5 $\boldsymbol{T} \leftarrow \boldsymbol{I}_{s+1} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{(s+1)\times(s+1)}$
6 **if** $\mathcal{J} \neq \emptyset$ **then**
7 $\quad j^* \leftarrow \min_{l \in \mathcal{J}}\{\arg\min_{l \in \mathcal{J}}\{d_l\}\}$
8 $\quad \boldsymbol{T} \leftarrow \boldsymbol{U}$ where $\boldsymbol{U}$ is as in (3.9)
9 $\quad \hat{d}_{j^*} \leftarrow \hat{d}_{j^*} + 1$
10 **return** $(\boldsymbol{T}, \hat{\boldsymbol{d}})$

The correctness of Algorithm 2 is established by the following lemma.

Lemma 3.2 (Correctness of Algorithm 2)**.** *The SkewInterpolatePoint procedure described in Algorithm 2 correctly transforms the input basis to satisfy the given interpolation condition.*

Proof. The structure of the matrix $\boldsymbol{U}$ is designed to perform two key operations:

1. For all columns except the j^*-th, $\boldsymbol{U}$ implements the cross-evaluation step for the non-minimal rows of $\boldsymbol{B}$. This operation, corresponding to Line 13 in Algorithm 1, is achieved through the entries in the j-th row and j^*-th column of $\boldsymbol{U}$.

2. The entry at position (j^*, j^*) in $\boldsymbol{U}$ executes the degree-increasing step, mirroring Line 11 in Algorithm 1.

As a result of these operations, the algorithm produces a transformation matrix $\boldsymbol{T}$ with the property that $\mathscr{E}_i$ maps all rows of $\boldsymbol{T}\boldsymbol{B}$ to zero. This demonstrates that the algorithm successfully adjusts the basis to meet the required interpolation condition. □

Having established the basic step with the SkewInterpolatePoint routine, we can now develop a D&C variant of the skew KNH interpolation. This approach, which we call SkewInterpolateTree, is presented in Algorithm 3.

Algorithm 3: SkewInterpolateTree

Input : Skew vector evaluation maps $\mathcal{E}_{[i_1,i_2]} = (\mathscr{E}_{i_1}, \ldots, \mathscr{E}_{i_2})$
$\boldsymbol{B} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{(s+1)\times(s+1)}$
$\boldsymbol{d} = [d_1, d_2, \ldots, d_{s+1}] \in \mathbb{Z}_{\geq 0}^{s+1}$ s.t. $d_j = \deg_{\boldsymbol{\omega}}(\boldsymbol{b}_j) \quad \forall j \in \{1, \ldots, s+1\}$

Output : A matrix $\boldsymbol{T} \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{(s+1)\times(s+1)}$ s.t. $\hat{\boldsymbol{B}} \stackrel{\text{def}}{=} \boldsymbol{T}\boldsymbol{B}$ is a $\boldsymbol{\omega}$-ordered weak-Popov Basis for $\langle \boldsymbol{B} \rangle \cap \mathcal{K}_{i_1} \cap \cdots \cap \mathcal{K}_{i_2}$
$\hat{\boldsymbol{d}} = [\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_{s+1}] \in \mathbb{Z}_{\geq 0}^{s+1}$ s.t. $\hat{d}_j = \deg_{\boldsymbol{\omega}}(\hat{\boldsymbol{b}}_j) \quad \forall j \in \{1, \ldots, s+1\}$

1 **if** $i_1 = i_2$ **then**
2 **return** *SkewInterpolatePoint*$(\mathscr{E}_{i_1}, \boldsymbol{B}, \boldsymbol{d})$
3 **else**
4 $z \leftarrow \left\lfloor \frac{i_1+i_2}{2} \right\rfloor$
5 $\boldsymbol{B}_1 \leftarrow \boldsymbol{B} \bmod_{\mathrm{r}} \boldsymbol{m}_{[i_1,z]}$
6 $(\boldsymbol{T}_1, \boldsymbol{d}_1) \leftarrow$ SkewInterpolateTree$(\mathcal{E}_{[i_1,z]}, \boldsymbol{B}_1, \boldsymbol{d})$
7 $\boldsymbol{B}_2 \leftarrow \boldsymbol{T}_1\boldsymbol{B} \bmod_{\mathrm{r}} \boldsymbol{m}_{[z+1,i_2]}$
8 $(\boldsymbol{T}_2, \boldsymbol{d}_2) \leftarrow$ SkewInterpolateTree$(\mathcal{E}_{[z+1,i_2]}, \boldsymbol{B}_2, \boldsymbol{d}_1)$
9 **return** $(\boldsymbol{T} = \boldsymbol{T}_2\boldsymbol{T}_1, \hat{\boldsymbol{d}} = \boldsymbol{d}_2)$

To establish the validity of our D&C approach, we now prove the correctness of the SkewInterpolateTree algorithm.

Lemma 3.3 (Correctness of Algorithm 3)**.** *Given Assumption 1, the SkewInterpolateTree procedure described in Algorithm 3 correctly computes a basis for the intersection of the input module with the kernels of the given evaluation maps.*

Proof. The correctness of Algorithm 3 can be established through the following key points:

1. The base case (when $i_1 = i_2$) is handled correctly by the SkewInterpolatePoint routine, as shown in Lemma 3.2.

2. For the recursive case, the algorithm correctly divides the problem into two subproblems, thanks to Assumption 1. This assumption ensures that the modulo operations in Lines 5 and 7 preserve the necessary information.

3. The combination of the subproblem solutions in Line 9 is valid due to the properties of the transformation matrix $\boldsymbol{U}$ defined in (3.9).

These elements together ensure that the algorithm produces a correct basis for the intersection of the input module with all the kernel modules defined by the given evaluation maps. □

3.4.2 Precomputing Minimal-Polynomial Vectors

We now introduce an efficient method to pre-compute the set $\mathcal{P}$ of minimal-polynomial vectors $\boldsymbol{m}_{[i,j]}$ required in Algorithm 3. Our approach builds upon the work presented in [CL17b, Theorem 3.2.7] and utilizes the concept of generalized operator evaluation, which can be constructed using the lclm of polynomial sequences (see (2.15)).

Algorithm 4 outlines this efficient procedure, which employs a D&C structure. To illustrate this structure, we provide a visual representation in Figure 3.1 for the case where $n = 4$.

The algorithm begins with the initial minimal-polynomial vectors $\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n$, from which all other minimal polynomials are computed via the lclm. The computation of these initial vectors is based on the general operator evaluation, as defined in Section 2.5.3. Later in (3.10) we show how this can be defined for decoding ILRS codes.

Algorithm 4: PreComputeMinVectorsTree

Input : Upper and lower index bound $a \in \mathbb{Z}_{\geq 0}$ and $b \in \mathbb{Z}_{\geq 0}$ with $b \geq a$
Minimal-polynomial vectors $\boldsymbol{m}_a, \boldsymbol{m}_{a+1}, \ldots, \boldsymbol{m}_b \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1}$

Output : A set

$$\{\boldsymbol{m}_{[a,b]}, \boldsymbol{m}_{[a,\lfloor (b-1)/2 \rfloor -1]}, \boldsymbol{m}_{[\lfloor (b-1)/2 \rfloor, b]}, \ldots \quad \ldots, \boldsymbol{m}_{[a,a]}, \boldsymbol{m}_{[a+1,a+1]}, \ldots \quad \ldots, \boldsymbol{m}_{[b,b]}\} \subseteq \mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1}$$

```
if a = b then
    return {m_a}
else
    δ ← ⌊(b−a+1)/2⌋
    P_1 ← PreComputeMinVectorsTree(a, a + δ − 1)
    P_2 ← PreComputeMinVectorsTree(a + δ, b)
    m_[a,b] ← lclm(m_[a,a+δ−1], m_[a+δ,b])      /* m_[a,a+δ−1] ∈ P_1 and m_[a+δ,b] ∈ P_2 */
    return P_1 ∪ P_2 ∪ {m_[a,b]}
```

Lemma 3.4 (Correctness of Algorithm 4)**.** *Algorithm 4 is correct.*

Proof. The correctness of Algorithm 4 follows directly from [CL17a, Theorem 3.2.7]. The algorithm proceeds in a recursive manner and splits the size of the set of considered minimal polynomials in half. When sets consist only of one element, $\boldsymbol{m}_{[a,a]}$ are computed, using the generalized operator (see (2.15)). The sets of minimal polynomials of larger size are then obtained by merging the smaller sets of minimal polynomials

using the relation $\boldsymbol{m}_{[a,b]} = \operatorname{lclm}\left(\boldsymbol{m}_{[a,a+\delta-1]}, \boldsymbol{m}_{[a+\delta,b]}\right)$ with $\delta = \lfloor \frac{b-a+1}{2} \rfloor$, also illustrated in Figure 3.1.

□

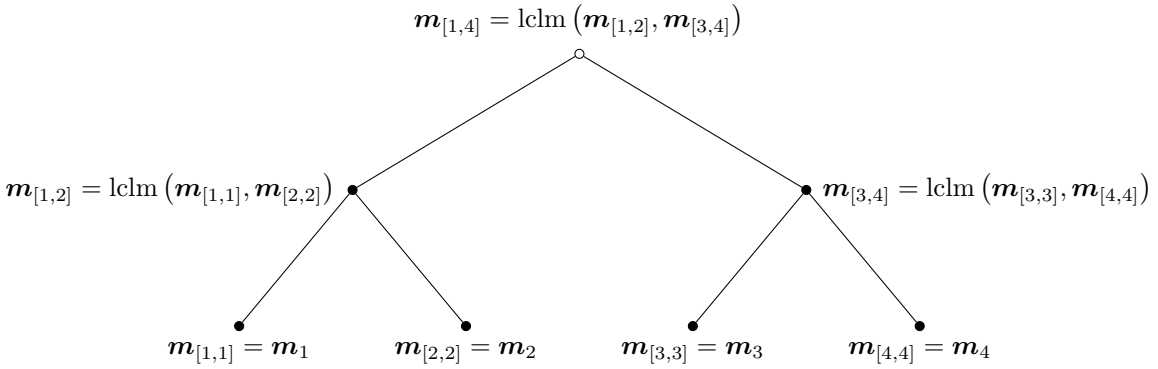


Figure 3.1: Illustration of the computation tree of Algorithm 4 to precompute all minimal-polynomial vectors in the set $\mathcal{P}$ for $n = 4$.

3.4.3 Application to Interleaved Linearized Reed–Solomon Codes

For an interpolation point set $\mathcal{P} = \{[p_{i,0}, p_{i,1}, \ldots, p_{i,s}] \mid i \in \{1, \ldots, n\}\} \subseteq \mathbb{F}_{q^m}^{s+1}$ define the vectors of minimal polynomials with respect to the *generalized operator evaluation* for $1 \leq i \leq j \leq n$ as

$$\boldsymbol{m}^{\mathrm{op}}_{[i,j]}(x)_{a_i} \stackrel{\text{def}}{=} \Big[M^{\mathrm{op}}_{\{p_{i,0},\ldots,p_{j,0}\}}(x)_{a_i}, M^{\mathrm{op}}_{\{p_{i,1},\ldots,p_{j,1}\}}(x)_{a_i}, \ldots \\ \ldots, M^{\mathrm{op}}_{\{p_{i,s},\ldots,p_{j,s}\}}(x)_{a_i} \Big] \in \mathbb{F}_{q^m}[x;\sigma,\delta]^{s+1}, \tag{3.10}$$

with the elements defined as in (2.15).

Lemma 3.5. *Let $\mathcal{E}^{op} = (\mathscr{E}_1^{op}, \ldots, \mathscr{E}_n^{op})$ be a tuple of generalized operator vector evaluation maps as defined in (3.5) and let $\mathcal{E}^{op}_{[i,j]} = (\mathscr{E}_i^{op}, \ldots, \mathscr{E}_j^{op})$. Then for any $\boldsymbol{Q} \in \mathbb{F}_{q^m}[x;\sigma]^{s+1}$ we have that*

$$\mathscr{E}_l^{op}(\boldsymbol{Q})_{a_l} = \mathscr{E}_l^{op}(\boldsymbol{Q} \bmod_{\mathrm{r}} \boldsymbol{m}^{op}_{[i,j]}(x)_{\boldsymbol{a}})_{a_l}, \quad \forall l \in \{i, \ldots, j\},$$

where $\boldsymbol{a} = [a_i, \ldots, a_j]$ contains the corresponding general operator evaluation parameters.

Proof. The lemma follows directly by applying the result from Lemma 2.1 to the elementary evaluations in the generalized operator vector operator evaluation maps defined in (3.5). □

Lemma 3.5 has a significant implication: the generalized operator vector evaluation maps from Definition 3.1 and the minimal-polynomial vectors defined in (3.10) satisfy Assumption 1. This compliance allows us to address Problem 3.1 using Algorithm 3.

To solve the problem, we invoke Algorithm 3 with the following parameters:

- **Evaluation set**: $\mathcal{E}^{op}$,
- **Basis**: $\boldsymbol{I}_{s+1}$ (the $(s+1) \times (s+1)$ identity matrix),
- **Initial degrees**: $\boldsymbol{\omega} = [0, k-1, \ldots, k-1]$.

This approach leverages the properties established in Lemma 3.5 to efficiently solve the interpolation problem for ILRS codes.

3.5 Summary and Discussion

In this chapter, we focused on efficient decoding algorithms for ILRS codes, motivated by the need for efficient decryption algorithms for legitimate users of potential code-based cryptosystems using sum-rank-metric codes. We revisited key decoding concepts for ILRS codes, setting the stage for our main contribution.

The primary result is a fast D&C variant of the KNH interpolation algorithm over free modules over skew polynomial rings. This variant solves the interpolation step of interpolation-based decoding of ILRS codes in $\widetilde{O}\big(s^{\zeta}\mathfrak{p}(n)\big)$ operations in $\mathbb{F}_{q^m}$. Our approach achieves the same asymptotic complexity as the fastest known methods for skew polynomial rings but uses the well-established bottom-up KNH algorithm, avoiding the more complex top-down minimal approximant bases techniques. Importantly, it also eliminates the need for pre-processing of interpolation points, making it more straightforward to implement.

Despite this improvement, Table 3.1 highlights that the optimal complexity of $\widetilde{O}\big(s^{\zeta-1}\mathfrak{p}(n)\big)$ achievable for ordinary polynomial rings is still out of reach for skew polynomials. Closing this gap remains an open area for future research.

This work advances the decoding efficiency for ILRS codes and applies to RS codes and Gabidulin codes as special cases. Our algorithmic improvements are crucial for enabling efficient decryption for legitimate users of potential code-based cryptosystems, making practical decoding more feasible.

Future research could focus on closing the complexity gap between skew and ordinary polynomial rings, as well as exploring applications of this algorithm to alternative metrics or algebraic structures. These developments could enhance the efficiency of decoding algorithms and the practical use of ILRS codes in cryptographic systems.

Table 3.1: Overview of the computational complexity of the proposed fast KNH interpolation approach compared to existing methods for the case of zero derivations ($\delta = 0$).

	Interpolation Method	Type	Polynomial Ring	Complexity ($\delta = 0$)
ordinary	KNH [WMW05]	bottom-up	$\mathbb{F}_{q^m}[x; \mathsf{Id}, 0]$	$O(s^2n^2)$
	DaC KNH [Nie14]	bottom-up	$\mathbb{F}_{q^m}[x; \mathsf{Id}, 0]$	$\widetilde{O}\big(s^\zeta \mathfrak{p}(n)\big)$
	Min. approximant bases [GJV03]	top-down	$\mathbb{F}_{q^m}[x; \mathsf{Id}, 0]$	$\widetilde{O}\big(s^\zeta \mathfrak{p}(n)\big)$
	Min. interpolation bases [JNSV17]	top-down	$\mathbb{F}_{q^m}[x; \mathsf{Id}, 0]$	$\widetilde{O}\big(s^{\zeta-1} \mathfrak{p}(n)\big)$
skew	Linearized KNH [XYS11]	bottom-up	$\mathbb{F}_{q^m}[x; \sigma_{\text{Frob}}, 0]$	$O(s^2n^2)$
	Skew KNH [LMK14]	bottom-up	$\mathbb{F}_{q^m}[x; \sigma, \delta]$	$O(s^2n^2)$
	DaC skew KNH [BJR24]	bottom-up	$\mathbb{F}_{q^m}[x; \sigma, \delta]$	$\widetilde{O}\big(s^\zeta \mathfrak{p}(n)\big)$
	Skew min. approximant bases [BJPR21]	top-down	$\mathbb{F}_{q^m}[x; \sigma, \delta]$	$\widetilde{O}\big(s^\zeta \mathfrak{p}(n)\big)$

4

Decoding of Space-Symmetric Rank Errors

In this chapter, we focus on Gabidulin codes, a specific type of rank-metric codes [Gab85; Rot91; Del78]. Note that the rank metric is a special case of the sum-rank metric, where the number of blocks is one. Similarly, Gabidulin codes are a specific instance of LRS codes. These codes, analogous to RS codes in the Hamming metric, play a crucial role in fields such as communication, cryptography, and network coding [Loi16; Loi17; LGB03; SKK08; SRV12; LCG19].

Previous work [GP04; PG06; GP06] has shown that Gabidulin codes containing a linear subcode of symmetric matrices can correct symmetric error matrices of rank up to $(n-1)/2$.

In this work, we relax the symmetry condition to focus on *space-symmetric errors*, where only the column and row spaces of error matrices are required to match. We demonstrate that for space-symmetric errors, decoding errors of rank up to $2(n-k)/3$ is possible with high probability.

This chapter extends the understanding of symmetric errors to space-symmetric ones, achieving enhanced decoding by using Gabidulin codes with a linear subcode of symmetric matrices. We cover the theoretical basis of this approach, provide proofs, and include simulation results. Additionally, we touch on potential applications in code-based cryptography.

The content of this chapter is based on the work presented at the IEEE International Symposium on Information Theory (ISIT 2021) and published in its proceedings [JSW21]. The author of this dissertation contributed significantly to all aspects of the paper, including the theoretical framework, proofs, and simulation results.

4.1 Gabidulin Codes Generated by Weak Self-Orthogonal Bases

Throughout this chapter, we focus on codewords that expand into square matrices over $\mathbb{F}_q$ for symmetry definitions, so we let $n = m$. In this case and analogous to the general definitions in (2.1) and (2.2), let the elements of the vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n] \in \mathbb{F}_{q^n}^n$ correspond to a fixed basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$. We define the map[1]

$$\begin{aligned} \phi : \mathbb{F}_{q^n}^n &\to \mathbb{F}_q^{n\times n} \\ \boldsymbol{a} &\mapsto \boldsymbol{A}, \end{aligned}$$

where $\boldsymbol{a} \in \mathbb{F}_{q^n}^n$ and $\boldsymbol{A} \in \mathbb{F}_q^{n\times n}$ is the unique matrix such that $\boldsymbol{a} = \boldsymbol{\alpha}\boldsymbol{A}$. The map ϕ is a bijection that preserves rank, so we have

$$\mathrm{rk}_q(\boldsymbol{a}) = \mathrm{rk}_q(\boldsymbol{A}).$$

For $\phi(\boldsymbol{a}) = \boldsymbol{A}$, let $\hat{\boldsymbol{a}}$ be the vector such that $\phi(\hat{\boldsymbol{a}}) = \boldsymbol{A}^\top$. We call $\hat{\boldsymbol{a}}$ the *ϕ-transposed vector* of $\boldsymbol{a}$. If $\boldsymbol{A}$ is a symmetric matrix, meaning $\boldsymbol{A} = \boldsymbol{A}^\top$, then $\boldsymbol{a} = \hat{\boldsymbol{a}}$. Furthermore let $\mathrm{GL}_n(\mathbb{F}_q)$ denote the set of all matrices in $\mathbb{F}_q^{n\times n}$ of full rank.

Gabidulin codes are constructed using *linearized polynomials*, which were introduced by Ore [Ore33a]. The ring of linearized polynomials, denoted by $\mathbb{L}_{q^n}[x]$, is isomorphic to the ring of skew polynomials (see Section 2.5.2 for details).

Recall that $x^{[i]} = x^{q^i}$ denotes the i-th power of the Frobenius automorphism. For a matrix $\boldsymbol{M} \in \mathbb{F}_{q^m}^{a\times b}$, we use $\boldsymbol{M}^{[i]}$ to indicate the elementwise application of the i-th power of the Frobenius automorphism to each entry of $\boldsymbol{M}$.

A linearized polynomial $f(x) \in \mathbb{L}_{q^n}[x]$ over $\mathbb{F}_{q^n}$ takes the form

$$f(x) = \sum_{i=0}^{d_f} f_{i+1} x^{[i]},$$

where $f_i \in \mathbb{F}_{q^n}$. We define the *q-degree* of $f(x)$, denoted $\deg_q f(x)$, as d_f when we have that $f_{d_f+1} \neq 0$. With $\boldsymbol{f} = [f_1, f_2, \ldots, f_{d_f+1}] \in \mathbb{F}_{q^n}^{d_f+1}$ we denote the vector containing the coefficients of $f(x)$. The multiplication of two linearized polynomials $g(x), f(x) \in \mathbb{L}_{q^n}[x]$ is defined as the (noncommutative) composition, i.e.

$$g(x) \cdot f(x) \stackrel{\text{def}}{=} g(f(x)).$$

Linearized polynomials possess the property of q-linearity. Specifically, for all

[1] Here, we use $\phi(\cdot)$ as a notation variant of $\mathrm{ext}(\cdot)$ (cf. (2.1)) from Chapter 2, specifically to distinguish this case where $m = n$.

$\alpha_1, \alpha_2 \in \mathbb{F}_q$ and $a, b \in \mathbb{F}_{q^n}$, the following holds

$$f(\alpha_1 a + \alpha_2 b) = \alpha_1 f(a) + \alpha_2 f(b).$$

A linearized polynomial of q-degree d is called the *minimal subspace polynomial* of a d-dimensional subspace if it has all elements of that subspace as roots.

One possible definition of Gabidulin codes is through a generator matrix, which coincides with the special case of the general Moore matrix described in (2.17).

Definition 4.1 (Gabidulin Code)**.** *Denote by $Gab_{\boldsymbol{\alpha}}[n,k]$ a Gabidulin code of dimension k and length n over $\mathbb{F}_{q^n}$ which is defined by its $k \times n$ generator matrix*

$$\boldsymbol{G}_k \stackrel{\text{def}}{=} \boldsymbol{M}_k(\boldsymbol{\alpha}),$$

where $\boldsymbol{\alpha} \in \mathbb{F}_{q^n}^n$ and $\alpha_1, \alpha_2, \ldots, \alpha_n$ are linearly independent over $\mathbb{F}_q$. The Moore matrix $\boldsymbol{M}_k(\boldsymbol{\alpha})$ is as defined in (2.17)*. The set of all Gabidulin codewords is then given by*

$$Gab_{\boldsymbol{\alpha}}[n,k] \stackrel{\text{def}}{=} \{\boldsymbol{u}\boldsymbol{G}_k : \forall \boldsymbol{u} \in \mathbb{F}_{q^n}^k\}.$$

Weak self-orthogonal bases play a significant role in coding theory and finite field applications [MB93; MS77; PG06]. We use such a basis for $\boldsymbol{\alpha}$, defined in the following.

Definition 4.2 (Weak Self-Orthogonal Basis)**.** *A basis $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n] \in \mathbb{F}_{q^n}^n$ of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ is called a* ***weak self-orthogonal basis*** *if*

$$\boldsymbol{M}_n(\boldsymbol{\alpha}) \cdot \boldsymbol{M}_n(\boldsymbol{\alpha})^\top = \boldsymbol{D},$$

where $\boldsymbol{D} \in \mathbb{F}_{q^n}^{n \times n}$ is a diagonal matrix.

In the following, we introduce the concept of transposed Gabidulin codes.

Definition 4.3 (Transposed Gabidulin Code)**.** *We define the transposed Gabidulin code as*

$$Gab_{\boldsymbol{\alpha}}^\top[n,k] \stackrel{\text{def}}{=} \{\phi^{-1}(\phi(\boldsymbol{c})^\top) : \forall \boldsymbol{c} \in Gab_{\boldsymbol{\alpha}}[n,k]\}.$$

The relationship between Gabidulin codes and their transposed counterparts becomes particularly interesting when considering their generator and parity-check matrices.

If the first row $\boldsymbol{\alpha}$ of a generator matrix of a Gabidulin code $\mathrm{Gab}_{\boldsymbol{\alpha}}[n,k]$ forms a weak self-orthogonal basis, then the parity-check matrix of the code is [PG06]

$$\boldsymbol{H}_{n-k} = \boldsymbol{M}_{n-k}(\boldsymbol{\alpha})^{[k]}.$$

Furthermore, under the same conditions, the parity-check matrix of the transposed code $\mathrm{Gab}_{\boldsymbol{\alpha}}^\top[n,k]$ takes a similar but distinct form [PG06]

$$\hat{\mathbf{H}}_{n-k} = \boldsymbol{M}_{n-k}(\boldsymbol{\alpha})^{[1]}.$$

4.2 Space-Symmetric Channel Model

We consider an additive channel model where a Gabidulin codeword $\boldsymbol{c} \in \mathrm{Gab}_{\boldsymbol{\alpha}}[n,k]$ is corrupted by an error $\boldsymbol{e}$ of rank $\mathrm{rk_q}(\boldsymbol{e}) = w$

$$\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e} \ \in \mathbb{F}_{q^m}^n. \tag{4.1}$$

In [GP04; GP06; PG06], the error matrix $\boldsymbol{E} = \phi(\boldsymbol{e}) \in \mathbb{F}_q^{m\times n}$ was considered to be *symmetric*, i.e.

$$\boldsymbol{E} = \boldsymbol{E}^\top.$$

Under this assumption, errors of rank up to $w \leq (n-1)/2$ can be corrected for certain rates.

In this paper, we relax the condition of $\boldsymbol{E}$ being symmetric to the condition that the row space of $\boldsymbol{E}$, denoted by $\mathcal{R}_q(\boldsymbol{E})$, equals its column space, denoted by $\mathcal{C}_q(\boldsymbol{E})$. That means

$$\mathcal{R}_q(\boldsymbol{E}) = \mathcal{C}_q(\boldsymbol{E})\,.$$

We call a matrix of rank w, whose row space equals its column space *space-symmetric*. Such a matrix can be decomposed as

$$\boldsymbol{E} = \boldsymbol{A}\boldsymbol{P}\boldsymbol{A}^\top, \tag{4.2}$$

where $\boldsymbol{A} \in \mathbb{F}_q^{n\times w}$ and $\boldsymbol{P} \in \mathrm{GL}_w(\mathbb{F}_q)$ are full-rank matrices. Note that the vector

$$\boldsymbol{a} = [a_1, a_2, \ldots, a_w] = \phi^{-1}(\boldsymbol{A}) \in \mathbb{F}_{q^n}^n, \tag{4.3}$$

forms a basis over $\mathbb{F}_q$ for both the column space and the row space of $\boldsymbol{E}$, since we have that $\mathcal{R}_q(\boldsymbol{E}) = \mathcal{C}_q(\boldsymbol{E})$.

4.3 Syndrome-Based Decoding Approach

This section introduces a syndrome-based decoding approach for Gabidulin codes. We then demonstrate how to transform the problem of decoding space-symmetric errors into decoding a specific interleaved Gabidulin code of interleaving order two.

Our method hinges on computing two syndromes:

1. One from the original code,
2. Another by transposing the received noisy codeword matrix and obtaining the syndrome from the transposed Gabidulin code.

These two syndromes are then used to jointly solve a linear system of equations, potentially increasing the decoding radius beyond $(n-k)/2$. The feasibility of finding a

solution depends on the matrix $\boldsymbol{P}$, as defined in (4.2). The resulting decoding process closely resembles that of a 2-interleaved Gabidulin code.

For more details on syndrome-based correction of up to $(n-k)/2$ errors in Gabidulin codes, we refer to [Gab85; Rot91; RP04a; SRB11]. For information on decoding interleaved Gabidulin codes, see [Loi06; SWC12; SB10; WZ14].

From the channel model in (4.1), we can derive two key syndromes

$$\boldsymbol{s}^{(1)} = \hat{\boldsymbol{y}}\hat{\boldsymbol{H}}_{n-k}^{\top} = \hat{\boldsymbol{e}}\hat{\boldsymbol{H}}_{n-k}^{\top}, \tag{4.4}$$

for the transposed code $\mathrm{Gab}_k^{\top}[\boldsymbol{\alpha}]$, and

$$\boldsymbol{s}^{(2)} = \boldsymbol{y}\boldsymbol{H}_{n-k}^{\top} = \boldsymbol{e}\boldsymbol{H}_{n-k}^{\top}, \tag{4.5}$$

for the non-transposed $\mathrm{Gab}_{\boldsymbol{\alpha}}[n,k]$ code. Each syndrome corresponds to a polynomial in $\mathbb{L}_{q^n}[x]$

$$s^{(i)}(x) = \sum_{j=1}^{n-k} s_j x^{[j-1]}, \quad \forall i \in \{1,2\}.$$

Given an error decomposed as in (4.2), we introduce the *row error span polynomial*. This polynomial is defined as the minimal subspace polynomial of the vector $\boldsymbol{a}$ (see [LN96]) as

$$\Gamma(x) \stackrel{\text{def}}{=} \prod_{\boldsymbol{u} \in \mathcal{R}_q(\boldsymbol{E})} (x - \phi^{-1}(\boldsymbol{u})) \in \mathbb{L}_{q^n}[x],$$

with $\deg_q(\Gamma(x)) = w$.

The space-symmetric nature of our error implies that $\mathcal{R}_q(\boldsymbol{E}) = \mathcal{C}_q(\boldsymbol{E})$. Consequently, the row error span polynomial is identical to the column error span polynomial, and satisfies

$$\Gamma(a_i) = 0,$$

for all $i \in \{1, \ldots, w\}$, where a_i are the entries of the vector $\boldsymbol{a}$ as defined in (4.3).

In the following, we give the key equation of the original code and the transposed code.

Theorem 4.1 (Key Equations)**.** *Let* $\Gamma(x) = \sum_{i=0}^{w} \Gamma_{i+1} x^{[i]} \in \mathbb{L}_{q^n}[x]$ *be the error span polynomial with* $w = \deg_q(\Gamma(x)) = \mathrm{rk_q}(\boldsymbol{e})$*. Then for each syndrome we obtain a key equation as follows*

$$\Gamma(s^{(i)}(x)) \equiv \Omega^{(i)}(x) \mod x^{[n-k]}, \forall i \in \{1,2\},$$

for some $\Omega^{(i)}(x)$ *with* $\deg_q(\Omega)^{(i)}(x)) < w$*.*

Proof. See Appendix A.1.1. □

Solving the key equation can be done by solving the linear system of equations

$$\boldsymbol{S}^{(i)} \cdot \boldsymbol{\Gamma}^{\top} = \boldsymbol{0},$$

where $\boldsymbol{\Gamma} = [\Gamma_1, \Gamma_2, \ldots, \Gamma_{w+1}]$ and $\boldsymbol{S}^{(i)}$

$$\boldsymbol{S}^{(i)} \stackrel{\text{def}}{=} \begin{bmatrix} s_{w+1}^{(i)\,[0]} & s_{w}^{(i)[1]} & \cdots & s_{1}^{(i)[w]} \\ s_{w+2}^{(i)\,[0]} & s_{w+1}^{(i)\,[1]} & \cdots & s_{2}^{(i)[w]} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n-k}^{(i)\,[0]} & s_{n-k-1}^{(i)\,[1]} & \cdots & s_{n-k-w+1}^{(i)\,[w]} \end{bmatrix}. \tag{4.6}$$

Since for each syndrome the error span polynomial in the key equation is the same, we can solve the two key equations jointly. This approach is similar to decoding a 2-interleaved Gabidulin code [Loi06; SWC12; SB10; WZ14] which yields the following linear system of equations

$$\boldsymbol{S} \cdot \boldsymbol{\Gamma}^{\top} = \begin{bmatrix} \boldsymbol{S}^{(1)} \\ \boldsymbol{S}^{(2)} \end{bmatrix} \cdot \boldsymbol{\Gamma}^{\top} = \boldsymbol{0}, \tag{4.7}$$

where (see Appendix A.1.2)

$$\boldsymbol{S}^{(1)} = \boldsymbol{M}_{n-k-w}(\boldsymbol{a})^{[w+1]} \cdot \boldsymbol{P} \cdot \boldsymbol{M}_{w+1}(\boldsymbol{a})^{\top}, \tag{4.8}$$

and

$$\boldsymbol{S}^{(2)} = \boldsymbol{M}_{n-k-w}(\boldsymbol{a})^{[w+k]} \cdot \boldsymbol{P}^{\top} \cdot \boldsymbol{M}_{w+1}(\boldsymbol{a})^{\top}. \tag{4.9}$$

Thus, $\boldsymbol{S}$ is as follows

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{M}_{n-k-w}(\boldsymbol{a})^{[w+1]} \cdot \boldsymbol{P} \\ \boldsymbol{M}_{n-k-w}(\boldsymbol{a})^{[w+k]} \cdot \boldsymbol{P}^{\top} \end{bmatrix} \cdot \boldsymbol{M}_{w+1}(\boldsymbol{a})^{\top}. \tag{4.10}$$

When $\mathrm{rk_q}(\boldsymbol{S}) = w$, we obtain a unique solution for $\Gamma(x)$, up to a scalar factor. Solving the key equation (4.7) yields the coefficients of $\Gamma(x)$, allowing us to determine a basis for its root space. This basis corresponds to a possible $\boldsymbol{a}$ in the decomposition given in (4.2). With a potential $\boldsymbol{a}$ identified, we can then determine the error.

Algorithm 5 outlines the complete decoding process, which has a complexity of at most $O(n^3)$ operations over $\mathbb{F}_{q^n}$. For a detailed method of obtaining the error matrix $\boldsymbol{E}$ from a possible vector $\boldsymbol{a}$, refer to Appendix A.1.3.

While Algorithm 5 suffices for our analysis, it's worth noting that more efficient methods exist for solving the joint syndrome key equation (4.7) and determining the matrix $\boldsymbol{B}$. The decoding techniques outlined in Chapter 3 are also applicable in this context. These sophisticated algorithms operate with quadratic or even sub-quadratic

complexity in terms of n. For further details on alternative decoding methods and algorithms, we refer the reader to [Loi06; SB10; SJB11; WZ14; PRLS17; PMM+17; SWC12; PW18; PW16; BJPR21].

Algorithm 5: DecodeSpaceSymmetric

Input : $\boldsymbol{y} = [y_1, y_2, \ldots, y_n] \in \mathbb{F}_{q^n}^n$
Parity-check matrix $\boldsymbol{H}_{n-k}$ of $\mathrm{Gab}_{\boldsymbol{\alpha}}[n,k]$

Output : $\boldsymbol{c} \in \mathrm{Gab}_{\boldsymbol{\alpha}}[n,k]$ or "decoding failure"

1 $\boldsymbol{s}^{(1)} \leftarrow \hat{\boldsymbol{y}}\hat{\boldsymbol{H}}_{n-k}^{\top}$
2 $\boldsymbol{s}^{(2)} \leftarrow \boldsymbol{y}\boldsymbol{H}_{n-k}^{\top}$
3 **if** $\boldsymbol{s}^{(2)} = \boldsymbol{0}$ **then**
4 **return** $\boldsymbol{y}$
5 **else**
6 $w \leftarrow \left\lfloor \frac{2}{3}(n-k) \right\rfloor$
7 Set up $\boldsymbol{S}^{(1)}$ and $\boldsymbol{S}^{(2)}$ as in (4.6)
8 $\boldsymbol{S} \leftarrow [(\boldsymbol{S}^{(1)})^{\top}, (\boldsymbol{S}^{(2)})^{\top}]^{\top}$
9 **while** $\mathrm{rk}(\boldsymbol{S}) < w$ **do**
10 $w \leftarrow w - 1$
11 Repeat Line 7 and 8
12 Solve: $\boldsymbol{S} \cdot \boldsymbol{\Gamma}^{\top} = \boldsymbol{0}$ for $\boldsymbol{\Gamma} = [\Gamma_1, \ldots, \Gamma_{w+1}] \in \mathbb{F}_{q^n}^{w+1}$
13 Find a basis $\boldsymbol{a} \in \mathbb{F}_{q^n}^{z}$ of the root space of $\Gamma(x)$
14 **if** $z = w$ **then**
15 Find $\boldsymbol{B}$ s.t. $\boldsymbol{e} = \boldsymbol{a}\boldsymbol{B}$ `/* cf. Appendix A.1.3 */`
16 $\boldsymbol{c} \leftarrow \boldsymbol{y} - \boldsymbol{a}\boldsymbol{B}$
17 **return** $\boldsymbol{c}$
18 **else**
19 **return** "decoding failure"

4.4 Probability of Decoding Failure

In this section, we demonstrate that decoding space-symmetric errors is likely to succeed with high probability. We focus on space-symmetric error matrices as represented in (4.2), and for simplicity in the analysis, we assume that the matrix

$$\boldsymbol{Q} \stackrel{\text{def}}{=} \boldsymbol{P}^{-1} \cdot \boldsymbol{P}^{\top},$$

is uniformly distributed over $\mathrm{GL}_w(\mathbb{F}_q)$. Section 4.5 provides simulation results to support this assumption.

Theorem 4.2 (Decoding of Space-Symmetric Errors)**.** *Let $Gab_{\boldsymbol{\alpha}}[n,k]$ be a Gabidulin code, where $\boldsymbol{\alpha}$ is a weak self-orthogonal basis. Let $\boldsymbol{r}$ be a noisy Gabidulin codeword as in* (4.1) *where $\boldsymbol{E}$ is a* ***space-symmetric*** *matrix of rank $w \leq 2(n-k)/3$. Then decoding is guaranteed with probability of at least $1 - P_f$, where P_f is the decoding failure probability.*

Assume that the matrix

$$\boldsymbol{Q} = \boldsymbol{P}^{-1} \cdot \boldsymbol{P}^{\top}, \tag{4.11}$$

where $\boldsymbol{P}$ is defined in (4.2)*, is uniformly drawn at random from $GL_w(\mathbb{F}_q)$. That means*

$$\boldsymbol{Q} \xleftarrow{\$} GL_w(\mathbb{F}_q).$$

Then P_f is bounded from above by

$$P_f \leq 4/q^n.$$

Proof. As discussed above, we obtain a unique solution for $\mathrm{rk_q}(\boldsymbol{S}) = w$ to succeed with decoding. To analyze the probability of failure, we restrict to the case for which the matrices $\boldsymbol{M}_{n-k-w}(\boldsymbol{a})^{[w+k]}$ and $\boldsymbol{M}_{n-k-w}(\boldsymbol{a})^{[w+1]}$ have no common rows, which means that $w > n-2k$. Consider the case of symmetric error matrices $\boldsymbol{E}$ for which $\boldsymbol{P} = \boldsymbol{P}^{\top}$, we have that

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{M}_{n-k-w}(\boldsymbol{a})^{[w+1]} \\ \boldsymbol{M}_{n-k-w}(\boldsymbol{a})^{[w+k]} \end{bmatrix} \cdot \boldsymbol{P} \cdot \boldsymbol{M}_{w+1}(\boldsymbol{a})^{\top},$$

for which we know that $\mathrm{rk_q}(\boldsymbol{P}) = w$ by definition, $\mathrm{rk_q}(\boldsymbol{M}_{w+1}(\boldsymbol{a})^{\top}) = w$ and since $n-k < w+k$ also the left part of the decomposition of $\boldsymbol{S}$ has always rank w for $w \leq 2(n-k)/3$.

For the case where $\boldsymbol{P}$ is not symmetric, we can rewrite (4.10) in a more compact form. Let us define

$$\tilde{\boldsymbol{M}}_{n-k-w} \stackrel{\mathrm{def}}{=} \boldsymbol{M}_{n-k-w}(\boldsymbol{a}) \cdot \boldsymbol{P}.$$

Using this definition, we can express the syndrome matrix $\boldsymbol{S}$ as

$$\boldsymbol{S} = \begin{bmatrix} \tilde{\boldsymbol{M}}_{n-k-w}^{[w+1]} \\ \tilde{\boldsymbol{M}}_{n-k-w}^{[w+k]} \cdot \boldsymbol{Q} \end{bmatrix} \cdot \boldsymbol{M}_{w+1}(\boldsymbol{a})^{\top}. \tag{4.12}$$

Assuming that $\boldsymbol{Q}$ is uniformly drawn at random from the set of all matrices in $\mathrm{GL}_w(\mathbb{F}_q)$ the matrix $\boldsymbol{S}$ is similar to the syndrome matrix of decoding a 2-interleaved Gabidulin code and we can bound the probability of decoding error P_f according to [SB10] and Theorem 4.2 follows. □

4.5 Numerical Results

We simulated a Gabidulin code using a weak self-orthogonal basis as locators, with parameters $n = 8$ and $k = 2$ over $\mathbb{F}_{2^8}$, in a space-symmetric error channel with a fixed error weight. Specifically, we set

$$w = \mathrm{rk}(\boldsymbol{E}) = \frac{2(n-k)}{3} = 4.$$

The maximum error weight possible for unique decoding of any rank error is

$$(n-k)/2 = 3.$$

We generated 10^6 noisy Gabidulin codeword samples and compared the outcomes across various scenarios:

1. **Space-symmetric errors:** Draw the matrix $\boldsymbol{A}$ and $\boldsymbol{P}$, both of rank w uniformly at random and decode using Algorithm 5.

2. **Uniform assumption:** The matrix $\boldsymbol{Q} \in \mathrm{GL}_w(\mathbb{F}_q)$ as in (4.11) is drawn uniformly at random instead of $\boldsymbol{P}$. We compute the matrix $\boldsymbol{S}$ as in (4.12) and check its rank. If $\mathrm{rk}(\boldsymbol{S}) \neq w$ we declare a decoding failure.

3. **2-interleaved Gabidulin code:** Simulation of a 2-interleaved Gabidulin code where the two error matrices are drawn uniformly at random such that the dimension of its column space is at most $2(n-k)/3 = 4$.

4. **Intersection probability:** Consider the probability that the intersection of two subspaces $\mathcal{U}$ and $\mathcal{V}$ of $\mathbb{F}_{q^n}^w$ with dimension ℓ drawn uniformly at random has dimension larger than or equal to z, This probability is [EV11]

$$\Pr[\dim(\mathcal{U} \cap \mathcal{V}) \geq z] = \frac{\sum_{i=z}^{\ell} \begin{bmatrix} w-\ell \\ \ell-i \end{bmatrix}_{q^n} \begin{bmatrix} \ell \\ i \end{bmatrix}_{q^n} \cdot q^{(\ell-i)^2}}{\begin{bmatrix} w \\ \ell \end{bmatrix}_{q^n}}. \tag{4.13}$$

   Consider the rows of $\boldsymbol{M}_{n-k-w}(\boldsymbol{a})^{[w+1]} \cdot \boldsymbol{P}$ being a basis of a subspace $\tilde{\mathcal{U}}$ of $\mathbb{F}_{q^n}^w$ of dimension $\ell = n-k-w$ and consider the rows of $\boldsymbol{M}_{n-k-w}(\boldsymbol{a})^{[w+k]} \cdot \boldsymbol{P}^\top$ being a basis of another subspace $\tilde{\mathcal{V}}$ also of dimension $\ell = n-k-w$. Use (4.13) as an estimation of the probability $\Pr[\dim(\tilde{\mathcal{U}} \cap \tilde{\mathcal{V}}) \geq z]$ for $z = 2(n-k) - 3w + 1$ which is equal to the probability of

$$\begin{bmatrix} \boldsymbol{M}_{n-k-w}(\boldsymbol{a})^{[w+1]} \\ \boldsymbol{M}_{n-k-w}(\boldsymbol{a})^{[w+k]} \end{bmatrix},$$

   having rank w and therefore $\mathrm{rk}(\boldsymbol{S}) = w$ according to (4.10).

Table 4.1 shows the simulation results, including the different scenarios for comparison. The decoding failure rate for space-symmetric errors using a Gabidulin code with a weak self-orthogonal basis is nearly identical to that observed under the uniform assumption. Furthermore, it aligns closely with the failure rate of decoding a 2-interleaved Gabidulin code over a standard rank-metric channel with fixed-rank errors. The upper bound on P_f is also provided, and we observe that the intersection probability offers a reliable estimate of the decoding failure rate.

Table 4.1: Simulation results for $n = 8$, $k = 2$ over $\mathbb{F}_{2^8}$ and $w = 4$.

Scenario	**Decoding failure rate**
1) Space-symmetric errors	0.004124
2) Uniform assumption	0.004229
3) 2-interleaved Gabidulin code	0.003965
4) Intersection probability	0.003921
Upper bound: $4/q^n$	0.015625

4.6 Number of Space-Symmetric Matrices

We now determine the exact number of space-symmetric matrices of a given rank over a finite field.

Theorem 4.3 (Number of Space-Symmetric Matrices)**.** *The number $\mathcal{N}_{sp\text{-}sym}(n, w, q)$ of $n \times n$ matrices over $\mathbb{F}_q$ of rank w that are space-symmetric is*

$$\mathcal{N}_{sp\text{-}sym}(n, w, q) = \prod_{i=0}^{w-1} (q^n - q^i). \tag{4.14}$$

Proof. Recall that the Gaussian binomial coefficient $\begin{bmatrix} n \\ w \end{bmatrix}_q$, defined in (2.4), gives the number of w-dimensional subspaces of $\mathbb{F}_q^n$ over $\mathbb{F}_q$. For square matrices, we can identify the column space with the image of the associated linear map from $\mathbb{F}_q^n$ to $\mathbb{F}_q^n$. Since column space and row space are equal for space-symmetric matrices, there are

$$\prod_{i=0}^{w-1} (q^w - q^i),$$

surjective linear maps from $\mathbb{F}_q^w$ to that w-dimensional image.

It follows that

$$\mathcal{N}_{\text{sp-sym}}(n, w, q) = \begin{bmatrix} n \\ w \end{bmatrix}_q \cdot \prod_{i=0}^{w-1} (q^w - q^i).$$

Substituting the definition of $\begin{bmatrix} n \\ w \end{bmatrix}_q$ from (2.4), we obtain

$$\begin{aligned}\mathcal{N}_{\text{sp-sym}}(n,w,q) &= \left(\prod_{i=0}^{w-1} \frac{q^n - q^i}{q^w - q^i}\right) \cdot \prod_{i=0}^{w-1} (q^w - q^i) \\ &= \prod_{i=0}^{w-1} (q^n - q^i),\end{aligned}$$

which proves (4.14). □

4.7 Application to Code-Based Cryptography

The Gabidulin–Paramonov–Tretjakov (GPT) cryptosystem, a McEliece-like scheme based on Gabidulin codes, was initially introduced in [GPT91a]. However, both the original version and many of its subsequent variants were later compromised by attacks from Gibson [Gib95; Gib96] and Overbeck [Ove06; Ove05; Ove08].

In this section, the potential application of space-symmetric rank errors in code-based cryptography is explored by comparing the key sizes of Loidreau's GPT variant [Loi16; Loi17] for arbitrary rank errors, symmetric errors, and space-symmetric errors. Although we do not offer security proofs, it is noted that both symmetric and space-symmetric errors introduce structure that could lead to new structural attacks, with space-symmetric errors offering less structure than symmetric ones.

The best known algorithm for decoding generic rank-metric codes [BBC+20] may be adaptable to these structured errors, potentially yielding improved complexity. Further analysis is necessary to rule out structural attacks for practical cryptosystems.

It's worth noting that restricting to symmetric or space-symmetric errors might enable reduced key sizes in other modern rank-metric code-based cryptographic schemes, such as the RQC scheme [MAB+20], a second-round submission to the NIST post-quantum cryptography standardization process.

Loidreau's GPT variant [Loi16; Loi17] includes a parameter λ that amplifies the error matrix's rank. Table 4.2 presents parameters assuming the possibility of embedding structured rank errors in this cryptosystem. We provide different hypothetical SLs, defined by the smallest WF of an attack in bits. We consider three WFs, the first two described in [Loi16]:

- **Decoding attack:** $\text{WF}_{\text{dec}} = n^3 q^{(w'-1)k}$,
- **Structural attack:** $\text{WF}_{\text{struc}} = n^3 q^{n(\lambda-1)-(\lambda-1)^2}$,
- **Brute-force attack on error patterns:** WF_{e}.

Here, $w' = w/\lambda$, where w represents the maximum number of correctable errors in each scenario:

1. **Conventional Gabidulin codes:** $w = \lfloor (n-k)/2 \rfloor$,
2. **Symmetric rank errors:** $w = \lfloor (n-1)/2 \rfloor$,
3. **Space-symmetric rank errors:** $w = \lfloor 2(n-k)/3 \rfloor$.

The work factor WF_{e} corresponds to the number of distinct error matrices, which varies for each case:

1. **Conventional rank errors:** The number of $n \times n$ matrices of rank w' over $\mathbb{F}_q$ is given by [LN96]:
$$\mathcal{N}_{\mathrm{rank}}(n, w', q) = \prod_{j=0}^{w'-1} \frac{(q^n - q^j)^2}{q^{w'} - q^j},$$
2. **Symmetric rank errors:** Let $\mathcal{N}_{\mathrm{symm}}(n, w', q)$ denote the number of symmetric matrices of size $n \times n$ with rank w' over $\mathbb{F}_q$. According to [MS77], we have
   - For even rank $w' = 2s$:
   $$\mathcal{N}_{\mathrm{symm}}(n, 2s, q) = \prod_{i=1}^{s} \frac{q^{2i}}{q^{2i} - 1} \cdot \prod_{i=0}^{2s-1} (q^{n-i} - 1),$$
   - For odd rank $w' = 2s + 1$:
   $$\mathcal{N}_{\mathrm{symm}}(n, 2s+1, q) = \prod_{i=1}^{s} \frac{q^{2i}}{q^{2i} - 1} \cdot \prod_{i=0}^{2s} (q^{n-i} - 1),$$
3. **Space-symmetric rank errors:** The number of space-symmetric matrices, denoted by $\mathcal{N}_{\text{sp-sym}}(n, w', q)$, is given by the expression in (4.14).

Table 4.2 presents the key sizes for a variant of the GPT cryptosystem [Loi16; Loi17] that utilizes different types of rank errors: conventional (Conv), symmetric (Sym), and space-symmetric (Sp-Sym), across various SLs. Each configuration employs codes with a code rate close to 1/2 to ensure comparability.

The columns show relevant parameters for each scheme, including the code length n, dimension k, the parameter λ and w'. While $\mathrm{WF}_{\mathrm{dec}}$, $\mathrm{WF}_{\mathrm{struc}}$, and WF_{e} denote the work factors associated with decoding attack, structural attacks, and brute-force, respectively.

The results indicate that both symmetric and space-symmetric rank errors can reduce the key size compared to conventional rank errors, while also maintaining or even improving the associated work factors. For instance, at the 256-bit security level, space-symmetric errors achieve a key size reduction compared to conventional errors, lowering the storage requirement from 27.65 KB to 17.87 KB. Similarly, symmetric

errors provide the most significant reduction at the 192-bit security level, achieving a key size of 7.45 KB versus 21.30 KB for conventional errors.

However, it is worth mentioning that symmetric errors, due to their inherent symmetry, may be more susceptible to structural attacks that could exploit this property. Space-symmetric errors, by contrast, could offer a balance by reducing key size while potentially being more resilient to such attacks.

Table 4.2: Key sizes of the GPT cryptosystem variant [Loi16; Loi17] using different types of errors: conventional rank errors (Conv), symmetric (Sym) and space-symmetric (Sp-Sym) rank errors for different SLs. The code rate of all codes is approximately 1/2.

SL	**Type**	n	k	λ	w'	$\mathbf{WF_{dec}}$	$\mathbf{WF_{struc}}$	$\mathbf{WF_e}$	**Keysize**
256	Conv	96	48	4	6	259.75	298.75	1117.77	27.65 KB
256	Sym	80	40	5	7	258.97	322.97	539.53	16.00 KB
256	**Sp-Sym**	**83**	**41**	**4**	**7**	**265.13**	**259.13**	**581.00**	**17.87 KB**
192	Conv	88	44	4	5	195.38	274.38	856.75	21.30 KB
192	Sym	62	31	4	7	203.86	194.86	413.53	7.45 KB
192	**Sp-Sym**	**71**	**35**	**4**	**6**	**193.45**	**222.45**	**426.00**	**11.18 KB**
128	Conv	59	29	3	5	133.65	131.65	566.75	6.41 KB
128	Sym	49	24	4	6	136.84	154.84	279.53	3.68 KB
128	**Sp-Sym**	**58**	**29**	**4**	**6**	**162.57**	**129.57**	**348.00**	**6.10 KB**

4.8 Summary and Discussion

In this chapter, we focused on decoding Gabidulin codes under the assumption of space-symmetric errors. Space-symmetric errors, where the row and column spaces of the error matrix coincide, allow for improved decoding performance compared to conventional rank errors. Specifically, we demonstrated that using Gabidulin codes, errors of rank up to $2(n-k)/3$ can be decoded with high probability, which extends the known decoding radius for Gabidulin codes.

We built on previous work that considered symmetric matrices, relaxing the symmetry condition to include space-symmetric errors. Our approach enables more flexible decoding while maintaining high decoding efficiency. The key insight is that by using Gabidulin codes with a linear subcode of symmetric matrices, we can exploit the error structure to correct higher-rank errors.

This chapter contributes to the broader objective of the thesis by exploring alternative error structures and decoding strategies to enhance the performance of code-based cryptosystems. While we do not propose a specific cryptosystem, we discussed potential applications of space-symmetric errors in cryptographic contexts, such as in the

GPT cryptosystem and other rank-metric schemes. As shown in Table 4.2, using structured errors like symmetric or space-symmetric errors could lead to reductions in key sizes, which is of particular importance for code-based cryptography. However, these structured errors also introduce new attack vectors, and further cryptanalysis is needed to evaluate the security implications.

The analysis presented here opens avenues for future research, particularly in refining error structures for cryptosystems and closing the gap between theoretical decoding improvements and practical security measures. Further exploration into the complexity of decoding algorithms for space-symmetric errors, along with rigorous cryptanalysis, could yield valuable insights for the design of more secure and efficient cryptosystems.

5

Decoding of High-Order Interleaved Sum-Rank-Metric Codes

In the previous chapters, we focused on decoding structured codes such as LRS codes and their interleaved versions (i.e. ILRS), as well as addressing space-symmetric errors for Gabidulin codes.

In this chapter, we extend our attention to a more general setting by presenting a Metzner–Kapturowski-like decoding algorithm that can be applied to *any* linear constituent code, including unstructured or random codes, as well as inherently structured codes whose structure is concealed (e.g., as in McEliece-like cryptosystems). This algorithm is designed for high-order (vertical) interleaved sum-rank-metric codes and generalizes the Metzner–Kapturowski decoding approach originally developed for the Hamming metric [MK90]. The ability to decode unstructured codes makes the proposed decoder highly versatile and broadens its applicability significantly.

Building upon the original Metzner–Kapturowski decoder for the Hamming metric, extensions to the rank metric were introduced in [PRW19; RPW21a], adapting the principles to accommodate the structure and properties of rank-metric codes. We further generalize this approach to the *sum-rank metric*.

A key contribution of our work is the introduction of the concept of an *error code*, which is spanned by the s rows of the error matrix. This new perspective provides a more intuitive understanding of the decoding process by relating it to properties of the error code. It also enables us to derive new interpretations, particularly for the special cases in the Hamming and rank metrics.

The computational complexity of the proposed algorithm is of the order

$$O\Big(\max\{n^3, n^2 s\}\Big),$$

operations over $\mathbb{F}_{q^m}$. Importantly, the decoding complexity is independent of the code structure of the constituent code, as the algorithm exploits properties of high-order interleaving only.

This generalization of the Metzner–Kapturowski decoder to the sum-rank metric not only recovers the original decoder for the Hamming metric [MK90] and its rank-metric analogs [PRW19; RPW21a], but also provides critical insights for the design of McEliece-like cryptosystems based on interleaved codes in the sum-rank metric. Specifically, our results indicate that the interleaving order cannot be chosen too large, as this would enable the proposed algorithm to efficiently recover the message or decrypt without the private key in polynomial time.

Generally, the decoder is capable of correcting errors with sum-rank weight w up to

$$w \leq d_{\min} - 2.$$

Moreover, under certain conditions, it can handle errors with sum-rank weight w up to

$$w \leq n - k - 1,$$

where n is the code length and k is the constituent code dimension. The decoder's success relies on the following assumptions:

- **High-order condition:** The interleaving order s must be at least the sum-rank weight of the error, i.e.,
$$s \geq w.$$
- **Full-rank condition:** The error matrix must have full $\mathbb{F}_{q^m}$-rank, meaning
$$\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = w.$$

It is important to note that the full-rank condition inherently implies the high-order condition, as explained in Section 5.5. This is because the $\mathbb{F}_{q^m}$-rank of a matrix $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s \times n}$ is bounded by the interleaving order s.

A key contribution of this work is the analysis of the decoder's success probability when dealing with errors chosen uniformly at random from the set of all possible errors with a fixed sum-rank weight w. In Section 5.4, we examine the likelihood that the full-rank condition is met under these circumstances.

In Section 5.5, we extend our analysis beyond the unique decoding radius, exploring the average success probability of the decoder when the full-rank assumption is satisfied. We utilize random coding techniques to assess the probability of successful decoding beyond $d_{\min} - 2$.

Figure 5.1 illustrates the decoding regions for the decoder in Algorithm 6 when the full-rank condition is satisfied.

The findings presented in this chapter have significant implications for the design and security analysis of code-based cryptosystems using interleaved sum-rank-metric codes. By offering new insights into the decoding process and demonstrating high success probabilities beyond the unique decoding radius, our work contributes to the

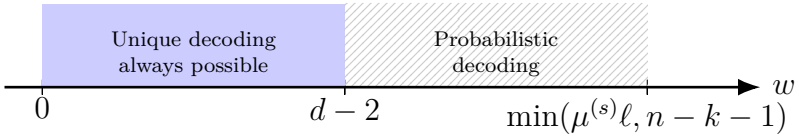


Figure 5.1: Illustration of the decoding regions for the proposed Metzner–Kapturowski-like decoder if the full-rank condition is satisfied.

ongoing development of robust post-quantum cryptographic solutions.

The content of this chapter is based on the work presented by the author at the Code-Based Cryptography Conference (CBCrypto 2022) [JHB23] and has been further extended in the journal version [JHB24], which is currently under review in IEEE Transactions on Information Theory. The author of this dissertation contributed significantly to all aspects of both papers. The probability analysis, as well as the simulations and numerical analysis detailed in Section 5.4 and Section 5.5, were primarily developed by the author, representing key contributions to the understanding of the decoding algorithm's performance.

5.1 Problem Description

This chapter focuses on homogeneous (vertically) s-interleaved sum-rank-metric codes over $\mathbb{F}_{q^m}$ as defined in Definition 2.4 and denoted as $\mathcal{IC}_{\Sigma R}[s; \boldsymbol{n}, k, d_{\min}]$. These codes have constituent codes $\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k, d_{\min}]$ defined by a parity-check matrix

$$\boldsymbol{H} = \left[\boldsymbol{H}^{(1)} \mid \boldsymbol{H}^{(2)} \mid \ldots \mid \boldsymbol{H}^{(\ell)}\right] \in \mathbb{F}_{q^m}^{(n-k)\times n},$$

where each $\boldsymbol{H}^{(i)} \in \mathbb{F}_{q^m}^{(n-k)\times n_i}$. For the sake of simplicity and convenience, we sometimes use the notation $\mathcal{C}$ to represent the constituent code $\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k, d_{\min}]$ throughout this chapter.

The objective is to recover codewords $\boldsymbol{C} \in \mathcal{IC}_{\Sigma R}[s; \boldsymbol{n}, k, d_{\min}]$ from corrupted matrices

$$\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E} \in \mathbb{F}_{q^m}^{s\times n},$$

where $\boldsymbol{E}$ is an error matrix with sum-rank weight $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{E}) = w$. We assume both the *high-order* and *full-rank* conditions are met.

Within this chapter we define the notion of error support as the *row support* of $\boldsymbol{E}$ as

$$\mathrm{supp}_{\Sigma R}(\boldsymbol{E}) \stackrel{\text{def}}{=} \mathrm{supp}_{\Sigma R}^{(R)}(\boldsymbol{E}), \tag{5.1}$$

with $\mathrm{supp}_{\Sigma R}^{(R)}(\boldsymbol{E})$ as in (2.33).

Additionally, we define the dual sum-rank support as

$$\begin{aligned}\operatorname{supp}_{\Sigma R}^{\perp}(\boldsymbol{E}) \stackrel{\text{def}}{=} &\operatorname{supp}_{R}^{\perp}\left(\boldsymbol{E}^{(1)}\right) \times \operatorname{supp}_{R}^{\perp}\left(\boldsymbol{E}^{(2)}\right) \times \cdots \times \operatorname{supp}_{R}^{\perp}\left(\boldsymbol{E}^{(\ell)}\right) \\ =&\mathcal{R}_q\left(\boldsymbol{B}^{(1)}\right)^{\perp} \times \mathcal{R}_q\left(\boldsymbol{B}^{(2)}\right)^{\perp} \times \cdots \times \mathcal{R}_q\left(\boldsymbol{B}^{(\ell)}\right)^{\perp}.\end{aligned}$$

Also recall that the error matrix can be decomposed as in (2.30) as $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B}$.

Inspired by the original Metzner–Kapturowski algorithm and its adaptation to the rank metric, the proposed decoding process involves two key steps:

1. **Recovering the error support:** Identify the error support $\operatorname{supp}_{\Sigma R}(\boldsymbol{E})$ to determine the locations of errors.
2. **Erasure decoding:** Use the syndrome matrix

   $$\boldsymbol{S} = \boldsymbol{H}\boldsymbol{Y}^{\top} = \boldsymbol{H}\boldsymbol{E}^{\top},$$

   to recover the error matrix $\boldsymbol{E}$. Consequently, retrieve the codeword $\boldsymbol{C}$ by computing $\boldsymbol{C} = \boldsymbol{Y} - \boldsymbol{E}$.

The following lemma, adapted from [PRR22], illustrates a method for reconstructing the error matrix $\boldsymbol{E}$ using its sum-rank support $\operatorname{supp}_{\Sigma R}(\boldsymbol{E})$ and the syndrome matrix $\boldsymbol{S}$. This lemma allows to solve the second step of the Metzner–Kapturowski-like decoder.

While the original theorem in [PRR22] assumes the condition $w < d_{\min}$, we present a more generalized version by relaxing this constraint. This modification enhances the lemma's applicability, particularly in scenarios where the error weight may exceed the minimum distance of the code.

Lemma 5.1 (Column-Erasure Decoder [PRR22, Theorem 13])**.** *Let the basis*

$$\boldsymbol{B} = \operatorname{diag}\left(\boldsymbol{B}^{(1)}, \ldots, \boldsymbol{B}^{(\ell)}\right) \in \mathbb{F}_q^{w \times n},$$

be a basis for the error support $\operatorname{supp}_{\Sigma R}(\boldsymbol{E})$ *of the error matrix* $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s \times n}$*, and let* $\boldsymbol{S} = \boldsymbol{H}\boldsymbol{E}^{\top} \in \mathbb{F}_{q^m}^{(n-k) \times s}$ *be the corresponding syndrome matrix.*

Assume that $\boldsymbol{H}\boldsymbol{B}^{\top}$ *is full-rank. Then,* $\boldsymbol{E}$ *can be uniquely recovered as* $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B}$*, where* $\boldsymbol{A} \in \mathbb{F}_{q^m}^{s \times w}$ *is the unique solution to the linear system*

$$\boldsymbol{S} = (\boldsymbol{H}\boldsymbol{B}^{\top})\boldsymbol{A}^{\top}.$$

Moreover, $\boldsymbol{E}$ *can be computed in* $O((n-k)^3 m^2)$ *operations over* $\mathbb{F}_q$*.*

Remark 5.1. *As noted in [PRR22, Lemma 12], when* $w < d_{\min}$*, the condition that* $\boldsymbol{H}\boldsymbol{B}^{\top}$ *is full-rank is inherently satisfied. We have relaxed this requirement to make the result applicable even when* $w \geq d_{\min}$*, since we consider this scenario in Section 5.5. The proof from [PRR22, Lemma 12] still holds.*

5.2 Recovering the Error Support

In this section, we focus on the first step of the Metzner–Kapturowski-like decoder as described in Section 5.1.

Let $\boldsymbol{w} = [w_1, \ldots, w_\ell] \in \mathbb{Z}_{\geq 0}^{\ell}$ denote the rank profile of the error matrix $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s \times n}$, with $w_i = \mathrm{rk_q}(\boldsymbol{E}^{(i)})$ for $i \in \{1, \ldots, \ell\}$. We assume that $\boldsymbol{E}$ fulfills the full-rank condition, meaning its $\mathbb{F}_{q^m}$-rank is equal to its sum-rank weight w. Note that the full-rank condition is satisfied if and only if $\mathrm{rk_{q^m}}(\boldsymbol{A}) = w$ for every $\boldsymbol{A} \in \mathbb{F}_{q^m}^{s \times w}$ as in (2.30).

With these assumptions, the rows of $\boldsymbol{E}$ span an $\mathbb{F}_{q^m}$-linear $\mathcal{E}_{\Sigma R}[\boldsymbol{n}, w]$ sum-rank metric code, denoted as

$$\mathcal{E} \stackrel{\text{def}}{=} \mathcal{R}_{q^m}(\boldsymbol{E}), \tag{5.2}$$

which we refer to as the *error code*.

Let $\boldsymbol{G}_{\mathcal{E}} \in \mathbb{F}_{q^m}^{w \times n}$ denote the generator matrix of $\mathcal{E}$. We can decompose $\boldsymbol{G}_{\mathcal{E}}$ as

$$\boldsymbol{G}_{\mathcal{E}} = \boldsymbol{A}_{\mathcal{E}} \boldsymbol{B}, \tag{5.3}$$

where $\boldsymbol{A}_{\mathcal{E}} = \left[\boldsymbol{A}_{\mathcal{E}}^{(1)} \mid \ldots \mid \boldsymbol{A}_{\mathcal{E}}^{(\ell)}\right] \in \mathbb{F}_{q^m}^{w \times w}$ with $\mathrm{rk_{q^m}}(\boldsymbol{A}_{\mathcal{E}}) = w$ and $\boldsymbol{B}$ is the same matrix as defined in the error decomposition (2.30) and (2.28). Each block $\boldsymbol{A}_{\mathcal{E}}^{(i)}$ is a matrix of size $w \times w_i$. The rank profile $\boldsymbol{w}$ specifies the rank of each block $\boldsymbol{A}_{\mathcal{E}}^{(i)}$, given by

$$\mathrm{rk_{q^m}}(\boldsymbol{A}_{\mathcal{E}}^{(i)}) = w_i.$$

It follows directly from (5.2), the definition of the error code that

$$\mathrm{supp}_{\Sigma R}(\mathcal{E}) = \mathrm{supp}_{\Sigma R}(\boldsymbol{E}).$$

Because of this property, we say that the error code $\mathcal{E}$ is *support-restricted by the row support of* $\boldsymbol{E}$ with $\mathcal{E} \subset \mathbb{F}_q^{\boldsymbol{n}}$.

Let us now consider the parity-check matrix $\boldsymbol{H}_{\mathcal{E}} \in \mathbb{F}_{q^m}^{(n-w) \times n}$ of the error code $\mathcal{E}$. By definition, the parity-check matrix satisfies

$$\boldsymbol{G}_{\mathcal{E}} \boldsymbol{H}_{\mathcal{E}}^{\top} = \boldsymbol{0}.$$

The following lemma establishes a relationship between the support of the parity-check matrix and the error matrix.

Lemma 5.2. *Let* $\boldsymbol{H}_{\mathcal{E}} = \left[\boldsymbol{H}_{\mathcal{E}}^{(1)} \mid \ldots \mid \boldsymbol{H}_{\mathcal{E}}^{(\ell)}\right] \in \mathbb{F}_{q^m}^{(n-w) \times n}$ *be the parity-check matrix of the* $[\boldsymbol{n}, w]$ *error code* $\mathcal{E}$ *with length partition* $\boldsymbol{n}$*. Then, we have*

$$\mathrm{supp}_{\Sigma R}(\boldsymbol{H}_{\mathcal{E}}) = \mathrm{supp}_{\Sigma R}^{\perp}(\boldsymbol{E}).$$

Proof. Since $\boldsymbol{H}_{\mathcal{E}}$ is a parity-check matrix of $\mathcal{E}$, we have $\mathrm{rk_{q^m}}(\boldsymbol{H}_{\mathcal{E}}) = n - w$. With

respect to the sum-rank metric, we can partition the parity-check matrix of the error code as

$$\boldsymbol{H}_{\mathcal{E}} = \left[\boldsymbol{H}_{\mathcal{E}}^{(1)} \mid \ldots \mid \boldsymbol{H}_{\mathcal{E}}^{(\ell)}\right], \tag{5.4}$$

such that $\boldsymbol{H}_{\mathcal{E}}^{(i)} \in \mathbb{F}_{q^m}^{(n-w)\times n_i}$ for all $i \in \{1, \ldots, \ell\}$.

To satisfy the check equations, we must have

$$\boldsymbol{G}_{\mathcal{E}}\boldsymbol{H}_{\mathcal{E}}^{\top} = \boldsymbol{0} \;\Leftrightarrow\; (\boldsymbol{A}_{\mathcal{E}}\boldsymbol{B})\boldsymbol{H}_{\mathcal{E}}^{\top} = \boldsymbol{0} \;\Leftrightarrow\; \boldsymbol{B}\boldsymbol{H}_{\mathcal{E}}^{\top} = \boldsymbol{0}.$$

From (5.4) and the block-diagonal structure of $\boldsymbol{B}$ (see (2.28)), it follows that

$$\boldsymbol{B}^{(i)}{\boldsymbol{H}_{\mathcal{E}}^{(i)}}^{\top} = \boldsymbol{0} \quad \forall i \in \{1, \ldots, \ell\}.$$

By the rank-nullity theorem and since $\boldsymbol{B}^{(i)}$ is over $\mathbb{F}_q$, we have

$$\dim\left(\mathcal{R}_q\left(\boldsymbol{H}_{\mathcal{E}}^{(i)}\right)\right) \leq n_i - w_i,$$

for all $i \in \{1, \ldots, \ell\}$. However, since $\boldsymbol{H}_{\mathcal{E}}$ must have $(n-w)$ many $\mathbb{F}_{q^m}$-linearly independent rows and $\sum_{i=1}^{\ell}(n_i - w_i) = n - w$, we conclude that

$$\dim\left(\mathcal{R}_q\left(\boldsymbol{H}_{\mathcal{E}}^{(i)}\right)\right) = n_i - w_i,$$

and hence

$$\mathcal{R}_q\left(\boldsymbol{H}_{\mathcal{E}}^{(i)}\right) = \mathcal{R}_q\left(\boldsymbol{B}^{(i)}\right)^{\top} \quad \forall i \in \{1, \ldots, \ell\}.$$

By the definition of the sum-rank support, this concludes the proof. □

The following theorem is crucial for decoding, as it reveals how the sum-rank support of an error matrix relates to the dual sum-rank support of the code resulting from combining the error code and the constituent code. This relationship helps us recover the error support.

Theorem 5.1. *Let $\mathcal{C}$ be an $\mathbb{F}_{q^m}$-linear $[\boldsymbol{n}, k]$ sum-rank-metric code with generator matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k\times n}$ and parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$. Let $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B} \in \mathbb{F}_{q^m}^{s\times n}$ be a matrix with $\boldsymbol{A} \in \mathbb{F}_{q^m}^{s\times w}$, $\boldsymbol{B} \in \mathbb{F}_q^{w\times n}$, $\mathrm{rk_{q^m}}(\boldsymbol{E}) = w$, and $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{E}) = w$. Let $w \leq n-k-1$ and suppose that*

$$\mathrm{rk_{q^m}}\left(\boldsymbol{H}\left[\frac{\boldsymbol{B}}{\boldsymbol{b}}\right]^{\top}\right) = w+1 \quad \forall\, \boldsymbol{b} \in \mathbb{F}_q^{\boldsymbol{n}} \setminus \mathrm{supp}_{\Sigma R}(\boldsymbol{E}) \;\; \textit{s.t.} \;\; \mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{b}) = 1. \tag{5.5}$$

Further, denote by $\boldsymbol{G}_{\mathcal{E}} \in \mathbb{F}_{q^m}^{w\times n}$ the generator matrix of the error code $\mathcal{E} = \mathcal{R}_{q^m}(\boldsymbol{E})$. Consider the $\mathbb{F}_{q^m}$-linear code $\mathcal{S} = \mathcal{E} + \mathcal{C}$ defined as

$$\mathcal{S} \stackrel{\text{def}}{=} \mathcal{R}_{q^m}(\boldsymbol{G}_{\mathcal{S}}), \tag{5.6}$$

with generator matrix

$$\boldsymbol{G}_{\mathcal{S}} \stackrel{\text{def}}{=} \left[\frac{\boldsymbol{G}}{\boldsymbol{G}_{\mathcal{E}}}\right].$$

Then, for any valid parity-check matrix $\boldsymbol{H}_{\mathcal{S}} \in \mathbb{F}_{q^m}^{(n-k-w)\times n}$ *of the* $\mathbb{F}_{q^m}$*-linear* $[\boldsymbol{n}, k+w]$ *sum-rank-metric code* $\mathcal{S}$*, we have*

$$\operatorname{supp}_{\Sigma R}^{\perp}(\boldsymbol{H}_{\mathcal{S}}) = \operatorname{supp}_{\Sigma R}(\boldsymbol{E}). \tag{5.7}$$

Proof. See Appendix A.2.1. □

The following remark highlights a key relationship between the row spaces of the code, the error, and the received matrix in interleaved decoding.

Remark 5.2 (Row Space Relationships)**.** *Due to the properties of the error code and the relationship* $\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E}$*, the following row spaces over* $\mathbb{F}_{q^m}$ *are the same*

$$\mathcal{R}_{q^m}\left(\left[\frac{\boldsymbol{G}}{\boldsymbol{G}_{\mathcal{E}}}\right]\right) = \mathcal{R}_{q^m}\left(\left[\frac{\boldsymbol{G}}{\boldsymbol{E}}\right]\right) = \mathcal{R}_{q^m}\left(\left[\frac{\boldsymbol{G}}{\boldsymbol{Y}}\right]\right).$$

Thus, the rows of all three matrices are generating sets for the code $\mathcal{S} = \mathcal{C} + \mathcal{E}$*.*

The next remark explores the implications of using very high-order interleaving, providing new insights for cases where $s \geq k + w$.

Remark 5.3 (Very High-Order Interleaving)**.** *Viewing the Metzner–Kapturowski-like algorithm from an error-code perspective provides valuable insights for cases with very high-order interleaving, specifically when* $s \geq k + w$*. Specifically, if the rows of the transmitted codeword* $\boldsymbol{C}$ *form a generating set for* $\mathcal{C}$ *(i.e.,* $\mathrm{rk_{q^m}}(\boldsymbol{C}) = k$*) and the error matrix* $\boldsymbol{E}$ *satisfies the full-rank condition, then* $\mathrm{rk_{q^m}}(\boldsymbol{Y}) = k + w$*, and the rows of* $\boldsymbol{Y}$ *form a generating set for* $\mathcal{S} = \mathcal{C} + \mathcal{E}$*.*

This enables the computation of a parity-check matrix $\boldsymbol{H}_{\mathcal{S}}$ *for* $\mathcal{S}$ *directly from the received matrix* $\boldsymbol{Y}$*, by finding a basis for the right* $\mathbb{F}_{q^m}$*-kernel of* $\boldsymbol{Y}$*. The support of the error can then be recovered as* $\operatorname{supp}_{\Sigma R}^{\perp}(\boldsymbol{H}_{\mathcal{S}}) = \operatorname{supp}_{\Sigma R}(\boldsymbol{E})$ *(see* (5.7) *in Theorem 5.1).*

Notably, this allows the error support $\operatorname{supp}_{\Sigma R}(\boldsymbol{E})$ *to be determined without knowledge of the codes* $\mathcal{C}$ *and* $\mathcal{S}$*.*

This observation could have significant implications for cryptosystems based on (secret) very high-order interleaved codes, as knowledge of the error support could substantially lower the security level as discussed in [HPR+22]. The security level of a cryptosystem refers to the computational difficulty of breaking the system.

After proving that $\operatorname{supp}_{\Sigma R}^{\perp}(\boldsymbol{H}_{\mathcal{S}}) = \operatorname{supp}_{\Sigma R}(\boldsymbol{E})$, we present a theorem that establishes a connection between the syndrome matrix $\boldsymbol{S}$ and the parity-check matrix $\boldsymbol{H}_{\mathcal{S}}$ of the sum code $\mathcal{S} = \mathcal{C} + \mathcal{E}$. This theorem provides a direct method to derive $\boldsymbol{H}_{\mathcal{S}}$ from $\boldsymbol{S}$.

Alternatively, a parity-check matrix can be obtained by stacking $\boldsymbol{G}_{\mathcal{S}}$ with $\boldsymbol{Y}$ and using Gaussian elimination.

Theorem 5.2. *Let $\mathcal{IC}_{\Sigma R}[s; \boldsymbol{n}, k, d_{\min}]$ be an $\mathbb{F}_{q^m}$-linear interleaved sum-rank-metric code with constituent code $\mathcal{C}$, which has parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$. Let $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s\times n}$ be an error matrix with $\mathrm{rk_{q^m}}(\boldsymbol{E}) = w$ and $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{E}) = w \leq n-k-1$ and let $\mathcal{E}$ be the error code spanned by the rows of $\boldsymbol{E}$. The received word is $\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E}$, where $\boldsymbol{C} \in \mathcal{IC}_{\Sigma R}[s; \boldsymbol{n}, k, d_{\min}]$. The syndrome matrix is $\boldsymbol{S} = \boldsymbol{H}\boldsymbol{Y}^\top = \boldsymbol{H}\boldsymbol{E}^\top$, where $\boldsymbol{S} \in \mathbb{F}_{q^m}^{(n-k)\times s}$.*

Let $\boldsymbol{P} \in \mathbb{F}_{q^m}^{(n-k)\times(n-k)}$ be a full-rank matrix such that $\boldsymbol{P}\boldsymbol{S}$ is in row-echelon form, i.e.,

$$\boldsymbol{P}\boldsymbol{S} = \begin{bmatrix} \boldsymbol{S}' \\ \boldsymbol{0} \end{bmatrix} \Longrightarrow \boldsymbol{P}\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}' \\ \boldsymbol{H}_{\mathcal{S}} \end{bmatrix},$$

where $\boldsymbol{S}' \in \mathbb{F}_{q^m}^{w\times s}$, $\boldsymbol{H}' \in \mathbb{F}_{q^m}^{w\times n}$. Then $\boldsymbol{H}_{\mathcal{S}} \in \mathbb{F}_{q^m}^{(n-k-w)\times n}$ is a parity-check matrix for the sum-rank-metric code $\mathcal{S} = \mathcal{E} + \mathcal{C}$ as defined in (5.6).

Proof. Since $\boldsymbol{P}$ is invertible, multiplying both sides of $\boldsymbol{S} = \boldsymbol{H}\boldsymbol{E}^\top$ by $\boldsymbol{P}$ yields

$$\boldsymbol{P}\boldsymbol{S} = \boldsymbol{P}\boldsymbol{H}\boldsymbol{E}^\top.$$

As $\boldsymbol{H}$ has full row rank $\mathrm{rk_{q^m}}(\boldsymbol{H}) = n-k$ and $\mathrm{rk_{q^m}}(\boldsymbol{E}) = w$, we have

$$\mathrm{rk_{q^m}}(\boldsymbol{S}) = \mathrm{rk_{q^m}}(\boldsymbol{H}\boldsymbol{E}^\top) = \min\{\mathrm{rk_{q^m}}(\boldsymbol{H}), \mathrm{rk_{q^m}}(\boldsymbol{E})\} = \min\{n-k, w\} = w.$$

By the rank-nullity theorem, $\mathrm{rk_{q^m}}(\boldsymbol{P}\boldsymbol{S}) = \mathrm{rk_{q^m}}(\boldsymbol{S}) = w$, so $\boldsymbol{P}\boldsymbol{S}$ has w nonzero rows. As $\boldsymbol{P}\boldsymbol{S}$ is in row-echelon form, we can write

$$\boldsymbol{P}\boldsymbol{S} = \begin{bmatrix} \boldsymbol{S}' \\ \boldsymbol{0} \end{bmatrix},$$

where $\boldsymbol{S}' \in \mathbb{F}_{q^m}^{w\times s}$ has full row rank.

Partitioning $\boldsymbol{P}\boldsymbol{H}$ conformally with $\boldsymbol{P}\boldsymbol{S}$, we have

$$\boldsymbol{P}\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}' \\ \boldsymbol{H}_{\mathcal{S}} \end{bmatrix},$$

where $\boldsymbol{H}' \in \mathbb{F}_{q^m}^{w\times n}$ and $\boldsymbol{H}_{\mathcal{S}} \in \mathbb{F}_{q^m}^{(n-k-w)\times n}$. Since $\boldsymbol{P}\boldsymbol{H}\boldsymbol{E}^\top = \boldsymbol{P}\boldsymbol{S}$, we have

$$\begin{bmatrix} \boldsymbol{H}' \\ \boldsymbol{H}_{\mathcal{S}} \end{bmatrix} \boldsymbol{E}^\top = \begin{bmatrix} \boldsymbol{S}' \\ \boldsymbol{0} \end{bmatrix},$$

which implies $\boldsymbol{H}_{\mathcal{S}}\boldsymbol{E}^\top = \boldsymbol{0}$. As the rows of $\boldsymbol{E}$ span $\mathcal{E}$, this means $\boldsymbol{H}_{\mathcal{S}}$ satisfies the parity-check equations for $\mathcal{E}$. By construction, $\boldsymbol{H}_{\mathcal{S}}$ also satisfies the parity-check equations

for $\mathcal{C}$, as it is a submatrix of $\boldsymbol{PH}$. And since $\boldsymbol{H}_{\mathcal{S}}$ has $n-k-w$ rows and is of full-rank, it is a parity-check matrix for the sum-rank-metric code $\mathcal{S}$ defined in (5.6), which contains both $\mathcal{C}$ and $\mathcal{E}$. □

5.3 A Metzner–Kapturowski-like Decoding Algorithm for Sum-Rank-Metric Codes

Using Theorem 5.1 and Theorem 5.2, we can formulate an efficient decoding algorithm for high-order interleaved sum-rank-metric codes. The algorithm is detailed in Algorithm 6 and follows a similar approach to the Metzner–Kapturowski-like decoding algorithm for Hamming and rank-metric codes.

Algorithm 6: Decoding High-Order Interleaved Sum-Rank-Metric Codes

Input : Parity-check matrix $\boldsymbol{H}$ of $\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k, d_{\min}]$
Received word $\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E}$
with $\boldsymbol{C} \in \mathcal{IC}_{\Sigma R}[s; \boldsymbol{n}, k, d_{\min}]$ and
$\mathrm{wt}^{(\boldsymbol{n})}_{\Sigma R}(\boldsymbol{E}) = \mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = w$ (**full-rank condition**)

Output : Transmitted codeword $\boldsymbol{C}$

$\boldsymbol{S} \leftarrow \boldsymbol{H}\boldsymbol{Y}^\top \in \mathbb{F}_{q^m}^{(n-k)\times s}$
Compute $\boldsymbol{P} \in \mathbb{F}_{q^m}^{(n-k)\times(n-k)}$ s.t. $\boldsymbol{PS} = \mathrm{REF}(\boldsymbol{S})$
$\boldsymbol{H}_{\mathcal{S}} = \left[\boldsymbol{H}_{\mathcal{S}}^{(1)} \mid \boldsymbol{H}_{\mathcal{S}}^{(2)} \mid \ldots \mid \boldsymbol{H}_{\mathcal{S}}^{(\ell)}\right] \leftarrow (\boldsymbol{PH})_{[w+1:n-k],[1:n]} \in \mathbb{F}_{q^m}^{(n-w-k)\times n}$
for $i \in \{1, \ldots, \ell\}$ **do**
  Compute $\boldsymbol{B}^{(i)} \in \mathbb{F}_q^{w_i \times n_i}$ s.t. $\mathrm{ext}(\boldsymbol{H}_{\mathcal{S}}^{(i)})(\boldsymbol{B}^{(i)})^\top = \boldsymbol{0}$ and $w_i = n_i - \mathrm{rk}_q(\boldsymbol{H}_{\mathcal{S}}^{(i)})$
$\boldsymbol{B} \leftarrow \mathrm{diag}(\boldsymbol{B}^{(1)}, \boldsymbol{B}^{(2)}, \ldots, \boldsymbol{B}^{(\ell)}) \in \mathbb{F}_q^{w\times n}$
Compute $\boldsymbol{A} \in \mathbb{F}_{q^m}^{s\times w}$ s.t. $(\boldsymbol{HB}^\top)\boldsymbol{A}^\top = \boldsymbol{S}$
$\boldsymbol{C} \leftarrow \boldsymbol{Y} - \boldsymbol{AB} \in \mathbb{F}_{q^m}^{s\times n}$
return $\boldsymbol{C}$

Once $\boldsymbol{H}_{\mathcal{S}}$ is derived from the syndrome matrix $\boldsymbol{S}$, the rank support of each block can be independently determined using Theorem 5.1. This involves finding a basis matrix $\boldsymbol{B}^{(i)} \in \mathbb{F}_q^{w_i\times n_i}$ such that

$$\mathrm{ext}(\boldsymbol{H}_{\mathcal{S}}^{(i)})(\boldsymbol{B}^{(i)})^\top = \boldsymbol{0},$$

for all $i \in \{1, \ldots, \ell\}$. Here, w_i is calculated using the rank-nullity theorem as

$$w_i = n_i - \mathrm{rk}_q(\boldsymbol{H}_{\mathcal{S}}^{(i)}),$$

according to (5.7).

In the following theorem, we address the computational complexity of Algorithm 6. This result demonstrates the efficiency of the decoding process under specific conditions.

Theorem 5.3. *Let $\boldsymbol{C}$ be a codeword of an s-interleaved sum-rank-metric code, denoted by $\mathcal{IC}_{\Sigma R}[s; \boldsymbol{n}, k, d_{\min}]$ and let $\boldsymbol{H}$ be the parity-check matrix of the corresponding constituent code $\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k, d_{\min}]$. Furthermore, let $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s \times n}$ be an error matrix of sum-rank weight*

$$\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{E}) = w,$$

that fulfills

$$w \leq s \quad (\textbf{\textit{high-order condition}}),$$

and

$$\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = w \quad (\textbf{\textit{full-rank condition}}).$$

Let $\boldsymbol{B}$ be a basis of the $\mathbb{F}_q$-row space of $\boldsymbol{E}$. If (5.5) holds, then $\boldsymbol{C}$ can be uniquely recovered from the received word $\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E}$ using Algorithm 6 in a time complexity equivalent to

$$O\Big(\max\{n^3, n^2 s\}\Big),$$

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

Proof. Lemma 5.1 states that the error matrix $\boldsymbol{E}$ can be factored as $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B}$. The decoding procedure in Algorithm 6 starts by finding a basis $\boldsymbol{B}$ of the error support $\mathrm{supp}_{\Sigma R}(\boldsymbol{E})$ and then uses erasure decoding with respect to Lemma 5.1 to recover $\boldsymbol{A}$. The matrix $\boldsymbol{B}$ is computed by transforming $\boldsymbol{S}$ into row-echelon form using a transformation matrix $\boldsymbol{P}$ (see Line 2).

In Line 3, $\boldsymbol{H}_{\mathcal{S}}$ is obtained by choosing the last $n - k - w$ rows of $\boldsymbol{P}\boldsymbol{H}$. According to Theorem 5.2, the matrix $\boldsymbol{H}_{\mathcal{S}}$ serves as a parity-check matrix for both the error code $\mathcal{E}$ associated with the error matrix $\boldsymbol{E}$ and the constituent code $\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k, d_{\min}]$. Then using Theorem 5.1 for each block (see Line 5) we find a matrix $\boldsymbol{B}^{(i)}$ whose rows form a basis for

$$\mathcal{R}_q\Big(\mathrm{ext}(\boldsymbol{H}_{\mathcal{S}}^{(i)})\Big)^{\top},$$

and therefore a basis for $\mathrm{supp}_R(\boldsymbol{E}^{(i)})$ for all $i \in \{1, \ldots, \ell\}$.

The matrix $\boldsymbol{B}$ is the block-diagonal matrix formed by $\boldsymbol{B}^{(i)}$ (cf. (2.28) and see Line 6) for $i \in \{1, \ldots, \ell\}$.

Finally, $\boldsymbol{A}$ can be computed from $\boldsymbol{B}$ and $\boldsymbol{H}$ using Lemma 5.1 in Line 7. Hence, Algorithm 6 returns the transmitted codeword in Line 9.

The complexities of the lines in the algorithm are as follows:

- **Line 1:** The syndrome matrix $\boldsymbol{S} = \boldsymbol{H}\boldsymbol{Y}^\top$ can be computed in at most $O(n^2 s)$ operations in $\mathbb{F}_{q^m}$.
- **Line 2:** The transformation of $[\boldsymbol{S} \mid \boldsymbol{I}]$ into row-echelon form requires
$$O\Big((n-k)^2(s+n-k)\Big) \subseteq O\Big(\max\{n^3, n^2 s\}\Big),$$
operations in $\mathbb{F}_{q^m}$.
- **Line 3:** The product $(\boldsymbol{P}\boldsymbol{H})_{[w+1:n-k],[1:n]}$ can be computed requiring at most
$$O(n(n-k-w)(n-k)) \subseteq O\Big(n^3\Big),$$
operations in $\mathbb{F}_{q^m}$.
- **Line 5:** The transformation of $[\mathrm{ext}(\boldsymbol{H}_{\mathcal{S}}^{(i)})^\top \mid \boldsymbol{I}^\top]^\top$ into column-echelon form requires $O(n_i^2((n-k-w)m + n_i))$ operations in $\mathbb{F}_q$ per block. Overall we get
$$O\left(\sum_{i=1}^{\ell} n_i^2((n-k-w)m + n_i)\right) \subseteq O\Big(n^3 m\Big)$$
operations in $\mathbb{F}_q$ since we have that $O\Big(\sum_{i=1}^{\ell} n_i^2\Big) \subseteq O(n^2)$.
- **Line 7:** According to Lemma 5.1, this step can be done in $O((n-k)^3 m^2)$ operations over $\mathbb{F}_q$.
- **Line 8:** The product $\boldsymbol{A}\boldsymbol{B} = \left[\boldsymbol{A}^{(1)}\boldsymbol{B}^{(1)} \mid \boldsymbol{A}^{(2)}\boldsymbol{B}^{(2)} \mid \ldots \mid \boldsymbol{A}^{(\ell)}\boldsymbol{B}^{(\ell)}\right]$ can be computed in $O\Big(\sum_{i=1}^{\ell} s w_i n_i\Big) \subseteq O(sn^2)$ and the difference of $\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{B}$ can be computed in $O(sn)$ operations in $\mathbb{F}_{q^m}$.

The complexities for Line 5 and Line 7 are given for operations in $\mathbb{F}_q$. The number of $\mathbb{F}_q$-operations of both steps together is in $O(n^3 m^2)$ and their execution complexity can be bounded by $O(n^3)$ operations in $\mathbb{F}_{q^m}$ (see [CL09]).

Thus, Algorithm 6 requires $O(\max\{n^3, n^2 s\})$ operations in $\mathbb{F}_{q^m}$. □

Note that the complexity of Algorithm 6 is independent of the structure of the underlying constituent code. This applies even when the code is random.

5.4 Probabilistic Decoding for Uniform Random Errors

In practical settings, the full-rank condition may not always hold. Therefore, we consider the performance of the decoder when the error is drawn uniformly at random from the set of all error matrices of a given sum-rank weight w. We then derive an

upper bound on the error probability, which, for fixed code parameters, decays exponentially with respect to the difference between the error weight w and the interleaving order s.

Note that we still require the high-order condition, i.e., $s \geq w$. Otherwise, no error can possibly satisfy the full-rank condition since

$$\mathrm{rk_{q^m}}(\boldsymbol{E}) \leq \sum_{i=1}^{\ell} \mathrm{rk_{q^m}}(\boldsymbol{E}^{(i)}) \leq \sum_{i=1}^{\ell} \mathrm{rk_q}(\boldsymbol{E}^{(i)}) = \sum_{i=1}^{\ell} w_i = w,$$

holds, and $\boldsymbol{E}$ has size $s \times n$ (with $s \leq n$).

For the sake of simplicity in the analysis, we focus on the case where the length partition $\boldsymbol{n} = [n_1, \ldots, n_\ell]$ has constant block lengths, i.e., there exists a positive integer η such that $n_i = \eta$ for all $i \in \{1, \ldots, \ell\}$.

We introduce the following sets, which are integral to the proofs of the forthcoming theorems in this section.

The set of all error matrices with a sum-rank weight of w in the interleaved case is defined as

$$\mathcal{E}_w^{(s)} \stackrel{\text{def}}{=} \left\{ \boldsymbol{E} = \left[\boldsymbol{E}^{(1)} \mid \cdots \mid \boldsymbol{E}^{(\ell)} \right] : \mathrm{wt}_{\Sigma R}(\boldsymbol{E}) = \sum_{i=1}^{\ell} \mathrm{rk_q}(\boldsymbol{E}^{(i)}) = w \right\} \subseteq \mathbb{F}_{q^m}^{s \times n},$$

which includes all error matrices with a total sum-rank weight of w and an interleaving order s. The corresponding set of all possible rank profiles is denoted by $\mathcal{T}_{w,\ell,\mu^{(s)}}$ (see Definition 2.9). For a fixed rank profile $\boldsymbol{w} = [w_1, w_2, \ldots, w_\ell] \in \mathcal{T}_{w,\ell,\mu^{(s)}}$ we define

$$\mathcal{E}_{\boldsymbol{w}}^{(s)} \stackrel{\text{def}}{=} \left\{ \boldsymbol{E} = \left[\boldsymbol{E}^{(1)} \mid \cdots \mid \boldsymbol{E}^{(\ell)} \right] : \mathrm{rk_q}(\boldsymbol{E}^{(i)}) = w_i \right\} \subseteq \mathbb{F}_{q^m}^{s \times n}.$$

From (2.30) we have that we can decompose the error into $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B}$ with $\boldsymbol{A} \in \mathbb{F}_{q^m}^{s \times w}$ and $\boldsymbol{B} \in \mathbb{F}_q^{w \times n}$ with $\boldsymbol{A}$ and $\boldsymbol{B}$ both of full-rank over $\mathbb{F}_q$. Let us define the set of all possible matrices $\boldsymbol{A}$

$$\mathcal{A}_{\boldsymbol{w}}^{(s)} \stackrel{\text{def}}{=} \left\{ \boldsymbol{A} \in \mathbb{F}_{q^m}^{s \times w} : \mathrm{wt}_{\Sigma R}^{(\boldsymbol{w})}(\boldsymbol{A}) = w \right\}, \tag{5.8}$$

and all possible matrices $\boldsymbol{B}$ as

$$\mathcal{B}_{\boldsymbol{w}} \stackrel{\text{def}}{=} \left\{ \mathrm{diag}\left(\boldsymbol{B}^{(1)}, \ldots, \boldsymbol{B}^{(\ell)} \right) : \right. \tag{5.9}$$
$$\left. \boldsymbol{B}^{(i)} \in \mathbb{F}_q^{w_i \times \eta} \text{ with } \mathrm{rk_q}(\boldsymbol{B}^{(i)}) = w_i \quad \forall i \in \{1, \ldots, \ell\} \right\} \subseteq \mathbb{F}_q^{w \times n}.$$

When drawing $\boldsymbol{E}$ uniformly at random from $\mathcal{E}_w^{(s)}$ the marginal distribution for the

corresponding rank profile $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu^{(s)}}$ is given by

$$\Pr[\boldsymbol{w}] = \frac{1}{|\mathcal{E}_{w}^{(s)}|} \prod_{i=1}^{\ell} \mathrm{NM}_q(sm, \eta, w_i),$$

where $\mathrm{NM}_q(sm, \eta, w_i)$ denotes the number of matrices over $\mathbb{F}_q$ of size $sm \times \eta$ of rank w_i as given in (2.26).

In the following lemma, we provide the conditional probability of an error matrix $\boldsymbol{E}$ drawn uniformly at random from $\mathcal{E}_{\boldsymbol{w}}^{(s)}$ having $\mathbb{F}_{q^m}$-rank w, given a rank profile $\boldsymbol{w}$.

Lemma 5.3. *For a given rank profile $\boldsymbol{w} = [w_1, w_2, \ldots, w_\ell]$, let $\boldsymbol{E}$ be an error matrix drawn uniformly at random from the set $\mathcal{E}_{\boldsymbol{w}}^{(s)}$. The probability that $\boldsymbol{E}$ has $\mathbb{F}_{q^m}$-rank equal to w, given $\boldsymbol{w}$, is*

$$\begin{aligned} \Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = w \mid \boldsymbol{w}] &= \Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{A}) = w \mid \boldsymbol{w}] \\ &= \frac{\prod_{j=0}^{w-1}(q^{sm} - q^{jm})}{\prod_{i=1}^{\ell}\prod_{j=0}^{w_i-1}(q^{sm} - q^{j})}, \end{aligned}$$

where $\boldsymbol{A}$ is a matrix drawn uniformly at random from the set defined in (5.8).

Proof. Every error matrix $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s\times n}$ can be decomposed as in (2.30), i.e., $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B}$. Since $\boldsymbol{A}$ is the only part influencing the $\mathbb{F}_{q^m}$-rank of $\boldsymbol{E}$ and is unique if an arbitrary block-diagonal matrix $\boldsymbol{B}$ with $\mathcal{R}_q(\boldsymbol{B}) = \mathcal{R}_q(\boldsymbol{E})$ is fixed (see, e.g., [MP74, Theorem 1]), we obtain

$$\Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = w \,|\, \boldsymbol{w}] = \Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{A}) = w \,|\, \boldsymbol{w}].$$

Recall that $\mathcal{B}_{\boldsymbol{w}}$ is defined in (5.9) as the set of all block-diagonal matrices $\boldsymbol{B}$ with $\mathcal{R}_q(\boldsymbol{B}) = \mathcal{R}_q(\boldsymbol{E})$ and rank profile $\boldsymbol{w}$. By the law of total probability, we then have

$$\begin{aligned} \Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = w \,|\, \boldsymbol{w}] &= \sum_{\boldsymbol{B}\in\mathcal{B}_{\boldsymbol{w}}} \Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{A}) = w \,|\, \boldsymbol{w}, \boldsymbol{B}] \cdot \Pr[\boldsymbol{B} \,|\, \boldsymbol{w}] \\ &= \sum_{\boldsymbol{B}\in\mathcal{B}_{\boldsymbol{w}}} \Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{A}) = w \,|\, \boldsymbol{w}] \cdot \Pr[\boldsymbol{B} \,|\, \boldsymbol{w}] \\ &= \Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{A}) = w \,|\, \boldsymbol{w}], \end{aligned}$$

where we used the fact that $\Pr[\boldsymbol{B} \,|\, \boldsymbol{w}] = \frac{1}{|\mathcal{B}_{\boldsymbol{w}}|}$ since $\boldsymbol{B}$ is uniformly distributed over $\mathcal{B}_{\boldsymbol{w}}$. The probability $\Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{A}) = w \,|\, \boldsymbol{w}]$ can be computed as

$$\Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{A}) = w \,|\, \boldsymbol{w}] = \frac{|\{\boldsymbol{A}' \in \mathbb{F}_{q^m}^{s\times w} : \mathrm{wt}_{\Sigma R}^{(\boldsymbol{w})}(\boldsymbol{A}') = \mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{A}') = w\}|}{|\{\boldsymbol{A}' \in \mathbb{F}_{q^m}^{s\times w} : \mathrm{wt}_{\Sigma R}^{(\boldsymbol{w})}(\boldsymbol{A}') = w\}|}.$$

Consider any matrix $\boldsymbol{A} \in \{\boldsymbol{A}' \in \mathbb{F}_{q^m}^{s\times w} : \mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{A}') = w\}$. Since $\boldsymbol{A}$ is full-rank over $\mathbb{F}_{q^m}$ and $s \geq w$, we can make several observations about the ranks of its blocks $\boldsymbol{A}^{(i)}$. First,

the $\mathbb{F}_{q^m}$-rank of each block $\boldsymbol{A}^{(i)}$ is equal to its corresponding rank profile component, i.e., $\mathrm{rk_{q^m}}(\boldsymbol{A}^{(i)}) = w_i$. Moreover, the $\mathbb{F}_q$-rank of each block $\boldsymbol{A}^{(i)}$ is lower bounded by its $\mathbb{F}_{q^m}$-rank, meaning that $w_i \leq \mathrm{rk_q}(\boldsymbol{A}^{(i)})$. At the same time, the $\mathbb{F}_q$-rank of each block $\boldsymbol{A}^{(i)}$ is upper bounded by $\min(w_i, s)$, because the rank of a matrix cannot exceed its number of rows or columns. In this case, each block $\boldsymbol{A}^{(i)}$ has dimensions $s \times w_i$, so its $\mathbb{F}_q$-rank is at most $\min\{w_i, s\}$. However, since $w = \sum_{i=1}^{\ell} w_i \leq s$, we have $w_i \leq s$ for all $i \in \{1, \ldots, \ell\}$, which implies that $\min\{w_i, s\} = w_i$. By combining the lower and upper bounds, we conclude that $\mathrm{rk_q}\left(\boldsymbol{A}^{(i)}\right) = w_i$ for all $i \in \{1, \ldots, \ell\}$. This implies that for any $\boldsymbol{A} \in \{\boldsymbol{A}' \in \mathbb{F}_{q^m}^{s \times w} : \mathrm{rk_{q^m}}(\boldsymbol{A}') = w\}$ we have that $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{w})}(\boldsymbol{A}) = w$ and therefore, we have the following equality

$$\{\boldsymbol{A}' \in \mathbb{F}_{q^m}^{s \times w} : \mathrm{wt}_{\Sigma R}^{(\boldsymbol{w})}\left(\boldsymbol{A}'\right) = \mathrm{rk_{q^m}}(\boldsymbol{A}') = w\} = \{\boldsymbol{A}' \in \mathbb{F}_{q^m}^{s \times w} : \mathrm{rk_{q^m}}(\boldsymbol{A}') = w\},$$

and hence

$$\Pr[\mathrm{rk_{q^m}}(\boldsymbol{A}) = w \,|\, \boldsymbol{w}] = \frac{|\{\boldsymbol{A}' \in \mathbb{F}_{q^m}^{s \times w} : \mathrm{rk_{q^m}}(\boldsymbol{A}') = w\}|}{|\{\boldsymbol{A}' \in \mathbb{F}_{q^m}^{s \times w} : \mathrm{wt}_{\Sigma R}^{(\boldsymbol{w})}(\boldsymbol{A}') = w\}|} = \frac{\prod_{j=0}^{w-1}(q^{sm} - q^{jm})}{\prod_{i=1}^{\ell}\prod_{j=0}^{w_i-1}(q^{sm} - q^j)},$$

where

$$\prod_{j=0}^{t-1}(q^{sm} - q^{jm}),$$

is the number of all full-rank matrices of size $s \times w$ over $\mathbb{F}_{q^m}$ and

$$\prod_{i=1}^{\ell}\prod_{j=0}^{w_i-1}(q^{sm} - q^j),$$

is the number of all matrices in $\mathbb{F}_{q^m}^{s \times w}$ with sum-rank weight w with corresponding length partition $\boldsymbol{w}$. Both relations can be derived from (2.26). □

In the next lemma, we provide the probability that an error matrix $\boldsymbol{E}$ drawn uniformly at random from the set $\mathcal{E}_{\boldsymbol{w}}^{(s)}$ has $\mathbb{F}_{q^m}$-rank w.

Lemma 5.4. *Let $\boldsymbol{E}$ be an error matrix drawn uniformly at random from the set $\mathcal{E}_{\boldsymbol{w}}^{(s)}$. Then, the probability that $\mathrm{rk_{q^m}}(\boldsymbol{E}) = w$ is given by*

$$\Pr[\mathrm{rk_{q^m}}(\boldsymbol{E}) = w] = \frac{\prod_{j=0}^{w-1}(q^{sm} - q^{jm})}{|\mathcal{E}_{\boldsymbol{w}}^{(s)}|} \cdot \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu^{(s)}}} \prod_{i=1}^{\ell} \begin{bmatrix} \eta \\ w_i \end{bmatrix}_q. \tag{5.12}$$

Proof. Recall the sets $\mathcal{A}_{\boldsymbol{w}}$ and $\mathcal{B}_{\boldsymbol{w}}$ defined in (5.8) and (5.9), respectively.

According to Lemma 5.3, for a fixed rank profile $\boldsymbol{w}$, we can draw $\boldsymbol{A} \in \mathcal{A}_{\boldsymbol{w}}$ and $\boldsymbol{B} \in \mathcal{B}_{\boldsymbol{w}}$ independently and uniformly from their corresponding domains and obtain $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B}$ with $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{E}) = w$ such that $\boldsymbol{E}$ is uniformly drawn at random from $\mathcal{E}_{\boldsymbol{w}}$.

This means the probability $\Pr[\mathrm{rk}_{\mathrm{q}^{\mathrm{m}}}(\boldsymbol{E}) = w]$ is

$$\begin{aligned}\Pr[\mathrm{rk}_{\mathrm{q}^{\mathrm{m}}}(\boldsymbol{E}) = w] &= \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu^{(s)}}} \Pr[\boldsymbol{w}] \cdot \Pr[\mathrm{rk}_{\mathrm{q}^{\mathrm{m}}}(\boldsymbol{A}) = w \mid \boldsymbol{w}] \\ &= \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu^{(s)}}} \frac{\prod_{i=1}^{\ell} \mathrm{NM}_q(sm,\eta,w_i)}{|\mathcal{E}_w^{(s)}|} \cdot \frac{\prod_{j=0}^{w-1}(q^{sm}-q^{jm})}{\prod_{i=1}^{\ell}\prod_{j=0}^{w_i-1}(q^{sm}-q^j)} \\ &= \frac{\prod_{j=0}^{w-1}(q^{sm}-q^{jm})}{|\mathcal{E}_w^{(s)}|} \cdot \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu^{(s)}}} \frac{\prod_{i=1}^{\ell} \mathrm{NM}_q(sm,\eta,w_i)}{\prod_{i=1}^{\ell}\prod_{j=0}^{w_i-1}(q^{sm}-q^j)}.\end{aligned}$$

Here, we first apply the law of total probability to express $\Pr[\mathrm{rk}_{\mathrm{q}^{\mathrm{m}}}(\boldsymbol{E}) = w]$ as a sum over all possible rank profiles $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu^{(s)}}$. Then, we use the fact that $\boldsymbol{A}$ and $\boldsymbol{B}$ are drawn independently and uniformly from their respective domains to compute the conditional probability $\Pr[\mathrm{rk}_{\mathrm{q}^{\mathrm{m}}}(\boldsymbol{A}) = w \,|\, \boldsymbol{w}]$.

Next, we simplify the expression using the definition of the Gaussian binomial coefficient

$$\begin{aligned}\Pr[\mathrm{rk}_{\mathrm{q}^{\mathrm{m}}}(\boldsymbol{E}) = w] &= \frac{\prod_{j=0}^{w-1}(q^{sm}-q^{jm})}{|\mathcal{E}_w^{(s)}|} \cdot \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu^{(s)}}} \frac{\prod_{i=1}^{\ell}\prod_{j=0}^{w_i-1} \frac{(q^{sm}-q^j)(q^{\eta}-q^j)}{(q^{w_i}-q^j)}}{\prod_{i=1}^{\ell}\prod_{j=0}^{w_i-1}(q^{sm}-q^j)} \\ &= \frac{\prod_{j=0}^{w-1}(q^{sm}-q^{jm})}{|\mathcal{E}_w^{(s)}|} \cdot \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu^{(s)}}} \prod_{i=1}^{\ell}\prod_{j=0}^{w_i-1} \frac{(q^{\eta}-q^j)}{(q^{w_i}-q^j)} \\ &= \frac{\prod_{j=0}^{w-1}(q^{sm}-q^{jm})}{|\mathcal{E}_w^{(s)}|} \cdot \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu^{(s)}}} \prod_{i=1}^{\ell} \begin{bmatrix}\eta \\ w_i\end{bmatrix}_q.\end{aligned}$$

In the first step, we rewrite the numerator using the definition of $\mathrm{NM}_q(sm,\eta,w_i)$. Then, we cancel out the common terms in the numerator and denominator, leaving only the Gaussian binomial coefficients in the final expression, which completes the proof. □

At first glance, the expression in (5.12) does not appear to be computationally efficient. However, in [PRR20], it was shown that the term $|\mathcal{E}_{\boldsymbol{w}}^{(s)}|$ can be efficiently computed using a dynamic programming approach. Inspired by this, we propose a similar procedure to compute the right-hand side of (5.12). To this end, let us define

$$\Phi_{q,\eta}(w,\ell) \stackrel{\text{def}}{=} \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu^{(s)}}} \prod_{i=1}^{\ell} \begin{bmatrix}\eta \\ w_i\end{bmatrix}_q,$$

where $\Phi_{q,\eta}(w,\ell)$ represents the sum over all possible rank profiles $\boldsymbol{w}$ for a given sum-rank weight w. For each rank profile, the q-binomial coefficient $\begin{bmatrix}\eta \\ w_i\end{bmatrix}_q$ counts the num-

ber of subspaces of dimension w_i in an η-dimensional space over $\mathbb{F}_q$. This expression can be computed recursively as

$$\Phi_{q,\eta}(w,\ell) = \begin{cases} \begin{bmatrix} \eta \\ w \end{bmatrix}_q & \text{if } \ell = 1 \\ \sum_{w'=0}^{\min\{\eta,w\}} \begin{bmatrix} \eta \\ w' \end{bmatrix}_q \cdot \Phi_{q,\eta}(w-w',\ell-1) & \text{else} \end{cases}. \tag{5.13}$$

The recursive relation can be understood as follows: For the base case, when the number of blocks $\ell = 1$, there is only one block, and the number of subspaces of dimension w in an η-dimensional space over $\mathbb{F}_q$ is given by the q-binomial coefficient $\begin{bmatrix} \eta \\ w \end{bmatrix}_q$. For $\ell > 1$, we consider all possible dimensions w' for the first block, ranging from 0 to $\min\{\eta, w\}$. For each choice of w', we multiply the number of subspaces of dimension w' in the first block, given by $\begin{bmatrix} \eta \\ w' \end{bmatrix}_q$, with the number of ways to distribute the remaining sum-rank weight $w - w'$ among the remaining $\ell - 1$ blocks, recursively computed by

$$\Phi_{q,\eta}(w-w',\ell-1).$$

Algorithm 7: Compute $\Phi_{q,\eta}(w,\ell)$

Input : Parameters: q, η, w and ℓ
Output : $\Phi_{q,\eta}(w,\ell)$
Initialize: $N(w',\ell') = 0 \quad \forall w' \in \{1,\dots,w\}$ and $\ell' \in \{1,\dots,\ell\}$

for $w' \in \{1,\dots,w\}$ **do**
  $N(w',1) \leftarrow \begin{bmatrix} \eta \\ w' \end{bmatrix}_q$
for $\ell' \in \{2,\dots,\ell\}$ **do**
  for $w' \in \{1,\dots,w\}$ **do**
    $N(w',\ell') \leftarrow \sum_{w''=0}^{\min\{\eta,w'\}} N(w'-w'',\ell'-1) \cdot \begin{bmatrix} \eta \\ w'' \end{bmatrix}_q$
return $N(w,\ell)$

The details of the algorithm are provided in Algorithm 7, and the following theorem establishes its correctness and complexity.

Theorem 5.4. *Algorithm 7 is correct and requires $\ell \cdot w^2$ integer multiplications.*

Proof. The correctness of Algorithm 7 follows from the recursive relationship established in (5.13) with the base cases $\Phi_{q,\eta}(w,1) = \begin{bmatrix} \eta \\ w \end{bmatrix}_q$.

Regarding the complexity, the algorithm performs $\ell \cdot w^2$ integer multiplications. This is because, for each $\ell' \in \{2,\dots,\ell\}$ and each $w' \in \{1,\dots,w\}$, the inner loop runs over

$\min\{\eta, w'\}$ values, leading to at most w iterations per combination of ℓ' and w'. Thus, the total number of iterations is $\ell \cdot w^2$. □

Building on the previous theorem, we derive the following corollary regarding computational complexity.

Corollary 5.1. *The success probability in* (5.12) *can be computed with polynomially-bounded complexity.*

Proof. The success probability in (5.12) is given by

$$\frac{\prod_{j=0}^{w-1}(q^{sm} - q^{jm})}{|\mathcal{E}_{\boldsymbol{w}}^{(s)}|} \cdot \Phi_{q,\eta}(w, \ell).$$

We analyze the complexity of computing each term in this expression:

- The cardinality $|\mathcal{E}_{\boldsymbol{w}}^{(s)}|$ can be computed with polynomially bounded complexity, as shown in [PRR22].
- The term $\Phi_{q,\eta}(w, \ell)$ can be computed with polynomially bounded complexity according to Theorem 5.4.
- The computation of $\prod_{j=0}^{w-1}(q^{sm} - q^{jm})$ is also polynomially bounded. The terms q^{sm} and q^{jm} can be computed using repeated squaring, and their differences and products involve polynomially-bounded integer operations.

The overall complexity is dominated by the complexity of computing $|\mathcal{E}_{\boldsymbol{w}}^{(s)}|$ and the term $\Phi_{q,\eta}(w, \ell)$ both of which are polynomially bounded. Thus, the success probability can be computed with polynomially bounded complexity. □

5.4.1 Main Theorem

The following theorem establishes an upper bound on the failure probability of the decoding algorithm.

Theorem 5.5. *Let $\mathcal{IC}_{\Sigma R}[s; \boldsymbol{n}, k, d_{\min}]$ be an $\mathbb{F}_{q^m}$-linear homogeneous s-interleaved sum-rank-metric code with component code $\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k, d_{\min}]$ of minimum sum-rank distance $d_{\min}$, and let $w \leq \min\{s, d_{\min} - 2\}$. Furthermore, let*

$$\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E},$$

where $\boldsymbol{C}$ is a codeword of the interleaved code $\mathcal{IC}_{\Sigma R}[s; \boldsymbol{n}, k, d_{\min}]$ and $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s \times n}$ is an error matrix uniformly drawn at random from $\mathcal{E}_w^{(s)}$. Then the probability that Algorithm 6 cannot decode, which is the probability that $\mathrm{rk_{q^m}}(\boldsymbol{E}) \neq w$, is bounded from above as

$$\Pr[\mathrm{rk_{q^m}}(\boldsymbol{E}) \neq w] \leq w q^{-m(s-w+1)}. \tag{5.14}$$

Proof. From Lemma 5.4, we have

$$\Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = w] = \frac{\prod_{j=0}^{w-1}(q^{sm} - q^{jm})}{|\mathcal{E}_w^{(s)}|} \cdot \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu^{(s)}}} \prod_{i=1}^{\ell} \begin{bmatrix} \eta \\ w_i \end{bmatrix}_q .$$

Next, we consider the following inequality to bound the denominator $|\mathcal{E}_w^{(s)}|$

$$\begin{aligned} |\mathcal{E}_w^{(s)}| &= \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu^{(s)}}} \prod_{i=1}^{\ell} \begin{bmatrix} \eta \\ w_i \end{bmatrix}_q \prod_{j=0}^{w_i-1} (q^{sm} - q^j) \\ &\leq \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu^{(s)}}} \prod_{i=1}^{\ell} \begin{bmatrix} \eta \\ w_i \end{bmatrix}_q \prod_{j=0}^{w_i-1} q^{sm} \\ &= \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu^{(s)}}} \left(\prod_{i=1}^{\ell} \begin{bmatrix} \eta \\ w_i \end{bmatrix}_q \right) q^{sm \sum_{i=1}^{\ell} w_i} \\ &= \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu^{(s)}}} \left(\prod_{i=1}^{\ell} \begin{bmatrix} \eta \\ w_i \end{bmatrix}_q \right) q^{smw}. \end{aligned}$$

Using this inequality, we can further bound $\Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = w]$ as follows

$$\begin{aligned} \Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = w] &\geq \frac{\prod_{j=0}^{w-1}(q^{sm} - q^{jm})}{q^{smw}} \\ &= \prod_{j=0}^{w-1}(1 - q^{m(j-s)}) \geq 1 - wq^{m(w-s-1)}. \end{aligned}$$

At this point, we have the same equation as in the rank-metric case. The last step follows from [RPW21a, Theorem 10].

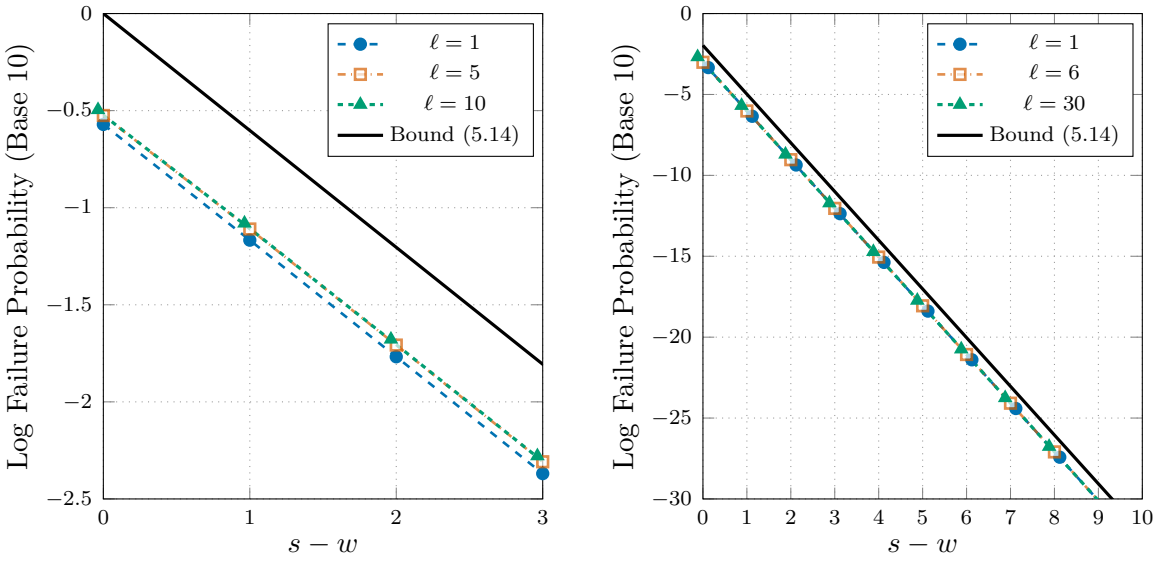
Finally, the claim of the theorem follows from the fact that

$$\Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) \neq w] = 1 - \Pr[\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = w].$$

□

5.4.2 Numerical Results

In Figure 5.2, we show the actual value of the failure probability, using Algorithm 7 to evaluate (5.12) and compare with the derived upper bound from (5.14). The failure probability is presented in logarithmic scale (base 10) versus the difference between the interleaving order s and the sum-rank error weight w for two different parameter sets:



(a) $q = 2$, $m = 2$, $n = 10$ and $w = 4$

(b) $q = 2$, $m = 10$, $n = 30$ and $w = 11$

Figure 5.2: Logarithmic failure probability vs. the difference $s - w$ for different values of ℓ with q, m, n and w.

- Figure 5.2a illustrates the failure probability for very small code parameters, with $q = 2$, $m = 2$, $n = 10$, and $w = 4$.

- Figure 5.2b shows the failure probability for larger, but still relatively small, code parameters, with $q = 2$, $m = 10$, $n = 30$, and $w = 11$.

From these plots, we can observe several key points:

1. As the code parameters increase, the difference in failure probability between the rank metric ($\ell = 1$), sum-rank metric ($1 \leq \ell \leq n$), and Hamming metric ($\ell = n$) becomes negligibly small. This suggests that for sufficiently large code parameters, the choice of metric has a diminishing impact on the failure probability.

2. The failure probability declines exponentially fast as $s - w$ increases, which is expected based on the expression of the upper bound in (5.14).

3. The gap between the upper bound and the actual failure probability narrows as the code parameters increase. In Figure 5.2b, with larger code parameters, the bound and the actual values are more closely aligned compared to Figure 5.2a. This suggests that the derived upper bound becomes tighter and more accurate for larger code parameters.

5.5 Decoding Radius

For the decoder presented in Algorithm 6 to succeed and uniquely recover the error, the following conditions must be satisfied:

1. The error matrix $\boldsymbol{E}$ must satisfy the *high-order* and *full-rank conditions*, i.e., $s \geq w$ and $\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = w$. Note that the full-rank condition already implies a high interleaving order, since for $\boldsymbol{E}$ to have rank w, the interleaving order s must be at least w.

2. The parity-check matrix $\boldsymbol{H}$ must satisfy the condition in (5.5), which can be expressed as

$$\mathrm{rk}_{\mathrm{q^m}}\left(\boldsymbol{H}\left[\frac{\boldsymbol{B}}{\boldsymbol{b}}\right]^{\top}\right) = w+1 \quad \forall\, \boldsymbol{b} \in \mathbb{F}_q^{\boldsymbol{n}} \setminus \mathrm{supp}_{\Sigma R}(\boldsymbol{E}) \text{ s.t. } \mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{b}) = 1$$

   where $\boldsymbol{B}$ is a basis of the row support of the error with respect to the sum-rank metric as in (2.28).

For $w \leq d_{\min} - 2$, the second condition is always satisfied, as shown in [PRR22, Lemma 8]. However, for $w \geq d_{\min} - 1$, the decoder becomes probabilistic and returns a unique solution to the decoding problem only if the second condition holds. When averaging over all error matrices $\boldsymbol{E}$, the probability of this condition being met becomes a property of the code itself, determined by the parity-check matrix $\boldsymbol{H}$ and thus the distance spectrum, which describes the distribution of distances between codewords. Note that the decoder in Algorithm 6 can correct errors with a maximum weight of

$$w \leq \min\{n-k-1, \mu^{(s)}\ell\}.$$

The term $n-k-1$ ensures that the common parity-check matrix of the error code and the component code has at least one nonzero row, which is necessary for successful decoding. This prevents reaching $n-k$, as it would result in no rows in the common parity-check matrix. The term $\mu^{(s)}\ell$ represents the maximum sum-rank weight for the given parameters, with $\mu^{(s)}$ defined in (2.32).

Figure 5.1 illustrates the decoding regions for Algorithm 6 when the error matrix $\boldsymbol{E}$ satisfies the full-rank condition, i.e., $\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = w$. This condition is crucial for the success of the decoding algorithm.

Consequently, this leads to two important results:

1. *Unique decoding* is always possible for

$$w \leq d_{\min} - 2,$$

   when the full-rank condition is satisfied.

2. Decoding is possible with *high probability* for

$$d_{\min} - 2 < w \leq n - k - 1,$$

when m is large.

In this section, we establish bounds on the probability for the second case. The following theorem provides a connection between the error matrix characteristics and the likelihood of satisfying the decoding condition.

Theorem 5.6. *Let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be a matrix chosen uniformly at random from $\mathbb{F}_{q^m}^{(n-k)\times n}$. We assume that q^m is large enough such that the probability of $\boldsymbol{H}$ having full $\mathbb{F}_{q^m}$-rank is close to 1. Consider an error matrix $\boldsymbol{E}$ picked uniformly at random from the set $\mathcal{E}_w^{(s)}$, where $\boldsymbol{E}$, $\boldsymbol{A}$, and $\boldsymbol{B}$ are as in* (2.30)*, and $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{E}) = w = \sum_{i=1}^{\ell} w_i$, satisfying the full-rank condition, i.e., $\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = w$. Then, on average, the probability that the condition* (5.5) *is satisfied is bounded from below and above as follows*

$$\mathrm{P_{LB}} \leq \Pr[(5.5) \textit{ is satisfied}] \leq \mathrm{P_{UB}},$$

where

$$\mathrm{P_{LB}} \stackrel{\text{def}}{=} \left(1 - \frac{1}{|\mathcal{E}_w^{(s)}|} \cdot \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}(s)} \prod_{i=1}^{\ell} \mathrm{NM}_q(sm, \eta, w_i) \cdot \frac{N_{\boldsymbol{w}}}{q^{m(n-k-w)}}\right) \cdot \cdot \prod_{j=0}^{w-1} \left(1 - \frac{1}{q^{m(n-k-j)}}\right), \tag{5.15}$$

with

$$N_{\boldsymbol{w}} \stackrel{\text{def}}{=} \min\{q^{m(n-k)}, \sum_{i=1}^{\ell}(q^{n_i} - q^{w_i})\}, \tag{5.16}$$

and

$$\mathrm{P_{UB}} \stackrel{\text{def}}{=} \prod_{j=0}^{w} \left(1 - \frac{1}{q^{m(n-k-j)}}\right).$$

Proof. The proof consists of two parts, one for the lower bound and one for the upper bound.

First, we show the lower bound. Condition (5.5) can only be satisfied if $\boldsymbol{H}\boldsymbol{B}^\top$ is of full $\mathbb{F}_{q^m}$-rank. Since $\boldsymbol{H}$ is chosen uniformly at random, $\boldsymbol{H}\boldsymbol{B}^\top$ is also a matrix uniformly distributed over $\mathbb{F}_{q^m}^{(n-k)\times w}$. The probability of $\boldsymbol{H}\boldsymbol{B}^\top$ having full $\mathbb{F}_{q^m}$-rank is given by

$$p_1 \stackrel{\text{def}}{=} \prod_{j=0}^{w-1} \left(1 - \frac{1}{q^{m(n-k-j)}}\right).$$

Now, consider a specific vector $\boldsymbol{b} \in \mathbb{F}_q^{\boldsymbol{n}} \setminus \operatorname{supp}_{\Sigma R}(\boldsymbol{E})$ and append it to $\boldsymbol{B}$. Note that for the bound we omit the restriction with $\operatorname{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{b}) = 1$. The probability that $\boldsymbol{H}[\boldsymbol{B}^\top \mid \boldsymbol{b}^\top]$ is of full $\mathbb{F}_{q^m}$-rank, given that $\boldsymbol{H}\boldsymbol{B}^\top$ is of full $\mathbb{F}_{q^m}$-rank, is equal to

$$p_2 \stackrel{\text{def}}{=} \left(1 - \frac{1}{q^{m(n-k-w)}}\right),$$

which is the probability that the $(w+1)$-th additional column in $\boldsymbol{H}[\boldsymbol{B}^\top \mid \boldsymbol{b}^\top]$ is linearly independent of the w remaining columns. This must hold true for any vector $\boldsymbol{b} \in \mathbb{F}_q^{\boldsymbol{n}} \setminus \operatorname{supp}_{\Sigma R}(\boldsymbol{E})$ simultaneously. Define the event $\mathcal{Z}_i$ as the $(w+1)$-th column in $\boldsymbol{H}[\boldsymbol{B}^\top \mid \boldsymbol{b}_i^\top]$ for a given $\boldsymbol{b}_i \in \mathbb{F}_q^{\boldsymbol{n}} \setminus \operatorname{supp}_{\Sigma R}(\boldsymbol{E})$ being linearly dependent on the remaining w columns, with $i \in \{1, \ldots, N_{\boldsymbol{w}}\}$ and

$$N_{\boldsymbol{w}} \geq \min\{q^{m(n-k)}, |\mathbb{F}_q^{\boldsymbol{n}} \setminus \operatorname{supp}_{\Sigma R}(\boldsymbol{E})|\}.$$

The cardinality $|\mathbb{F}_q^{\boldsymbol{n}} \setminus \operatorname{supp}_{\Sigma R}(\boldsymbol{E})|$ is given by

$$\sum_{i=1}^{\ell} (q^{n_i} - q^{w_i}),$$

which is the sum of the cardinalities of the blocks that correspond to

$$|\mathbb{F}_q^{n_i} \setminus \mathcal{R}_q\left(\boldsymbol{B}^{(i)}\right)| = q^{n_i} - q^{w_i}.$$

By applying the union bound on the events $\mathcal{Z}_i$, the probability that (5.5) is not satisfied is bounded from above as

$$\begin{aligned}
\Pr[\text{(5.5) is not satisfied}] &\leq (1 - p_1) + p_1 \cdot \sum_{\boldsymbol{w} \in \mathcal{T}_{t,\ell,\mu^{(s)}}} \Pr[\boldsymbol{w}] \cdot \Pr\left[\bigcup_{i=1}^{N_{\boldsymbol{w}}} \mathcal{Z}_i\right] \\
&\leq (1 - p_1) + p_1 \cdot \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu^{(s)}}} \Pr[\boldsymbol{w}] \cdot \sum_{i=1}^{N_{\boldsymbol{w}}} \Pr[\mathcal{Z}_i],
\end{aligned}$$

where $\Pr[\boldsymbol{w}]$ is the marginal distribution of the rank profiles, given by

$$\Pr[\boldsymbol{w}] = \frac{1}{|\mathcal{E}_w^{(s)}|} \prod_{i=1}^{\ell} \mathrm{NM}_q(sm, \eta, w_i).$$

Now, assuming that all $\mathcal{Z}_i$ are independent, we have that for a given $\boldsymbol{w}$,

$$\Pr[\mathcal{Z}_i] = 1 - p_2 = \frac{1}{q^{m(n-k-w)}}.$$

Therefore,

$$\begin{aligned}
&\Pr[(5.5) \text{ is not satisfied}] \leq (1-p_1) + p_1 \cdot \sum_{\boldsymbol{w} \in \mathcal{T}_{t,\ell,\mu^{(s)}}} \Pr[\boldsymbol{w}] \cdot \sum_{i=1}^{N_{\boldsymbol{w}}} \Pr[\mathcal{Z}_i] \\
&= (1-p_1) + p_1 \cdot \sum_{\boldsymbol{w} \in \mathcal{T}_{t,\ell,\mu^{(s)}}} \frac{1}{|\mathcal{E}_w^{(s)}|} \prod_{i=1}^{\ell} \mathrm{NM}_q(sm, \eta, w_i) \cdot \sum_{i=1}^{N_{\boldsymbol{w}}} \frac{1}{q^{m(n-k-w)}} \\
&= (1-p_1) + p_1 \cdot \sum_{\boldsymbol{w} \in \mathcal{T}_{t,\ell,\mu^{(s)}}} \frac{1}{|\mathcal{E}_w^{(s)}|} \prod_{i=1}^{\ell} \mathrm{NM}_q(sm, \eta, w_i) \cdot \frac{N_{\boldsymbol{w}}}{q^{m(n-k-w)}}.
\end{aligned}$$

Consequently,

$$\begin{aligned}
&\Pr[(5.5) \text{ is satisfied}] = 1 - \Pr[(5.5) \text{ is not satisfied}] \\
&\geq 1 - \left((1-p_1) + p_1 \cdot \sum_{\boldsymbol{w} \in \mathcal{T}_{t,\ell,\mu^{(s)}}} \frac{1}{|\mathcal{E}_w^{(s)}|} \prod_{i=1}^{\ell} \mathrm{NM}_q(sm, \eta, w_i) \cdot \frac{N_{\boldsymbol{w}}}{q^{m(n-k-w)}} \right) \\
&= p_1 - p_1 \cdot \sum_{\boldsymbol{w} \in \mathcal{T}_{t,\ell,\mu^{(s)}}} \frac{N_{\boldsymbol{w}} \cdot \prod_{i=1}^{\ell} \mathrm{NM}_q(sm, \eta, w_i)}{|\mathcal{E}_w^{(s)}| \cdot q^{m(n-k-w)}} \\
&= p_1 \cdot \left(1 - \sum_{\boldsymbol{w} \in \mathcal{T}_{t,\ell,\mu^{(s)}}} \frac{N_{\boldsymbol{w}} \cdot \prod_{i=1}^{\ell} \mathrm{NM}_q(sm, \eta, w_i)}{|\mathcal{E}_w^{(s)}| \cdot q^{m(n-k-w)}} \right) \\
&= \left(1 - \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu^{(s)}}} \frac{N_{\boldsymbol{w}} \cdot \prod_{i=1}^{\ell} \mathrm{NM}_q(sm, \eta, w_i)}{|\mathcal{E}_w^{(s)}| \cdot q^{m(n-k-w)}} \right) \prod_{j=0}^{w-1} \left(1 - \frac{1}{q^{m(n-k-j)}} \right),
\end{aligned}$$

which establishes the lower bound P_{LB}.

For the upper bound, we observe that the probability that condition (5.5) is satisfied is upper bounded by the event that at least one matrix $\boldsymbol{H}[\boldsymbol{B}^\top \mid \boldsymbol{b}^\top] \in \mathbb{F}_{q^m}^{(n-k)\times(w+1)}$ is of full $\mathbb{F}_{q^m}$-rank, where $\boldsymbol{b} \in \mathbb{F}_q^{\boldsymbol{n}} \setminus \mathrm{supp}_{\Sigma R}(\boldsymbol{E})$. This probability is equal to the probability that a random matrix in $\mathbb{F}_{q^m}^{(n-k)\times(w+1)}$ is of full $\mathbb{F}_{q^m}$-rank (see [LN96]), which is given by

$$\mathrm{P}_{\mathrm{UB}} = \prod_{j=0}^{w} \left(1 - \frac{1}{q^{m(n-k-j)}} \right).$$

This completes the proof. □

5.5.1 Numerical Results

We now investigate the tightness of the upper and lower bounds on the failure probability of condition (5.5) derived in Theorem 5.6. While these bounds provide theoretical

guarantees, they may not always give a precise estimate of the actual failure probability. To assess their accuracy and explore alternative approximations, we conduct simulations.

The following presents an approximation obtained by modifying the proof of Theorem 5.6. Although this approximation does not provide strict bounds, it may yield more realistic estimates of the failure probability and serves as a basis for comparison with the simulated results.

If, in the proof of Theorem 5.6, we ignore the dependence of the events $\mathcal{Z}_i$ for $i \in \{1, \ldots, N_{\boldsymbol{w}}\}$, we obtain neither a lower nor an upper bound on the failure/success probability of condition (5.5) for a random parity-check matrix $\boldsymbol{H}$. Nevertheless, we state the expression under that circumstance and use it as an approximation. We then show through simulation that this approximation provides a more realistic estimate of the success probability for relative small η. From the proof of Theorem 5.6, it is straightforward to show that, in this case

$$\begin{aligned}\Pr[\text{condition (5.5) is satisfied}] \approx \prod_{j=0}^{w-1} \left(1 - \frac{1}{q^{m(n-k-j)}}\right) \\ \cdot \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \Pr[\boldsymbol{w}] \left(1 - \frac{1}{q^{m(n-k-w)}}\right)^{N_{\boldsymbol{w}}}.\end{aligned} \tag{5.17}$$

In the case of the Hamming metric, the events $\mathcal{Z}_i$ for $i \in \{1, \ldots, N_{\boldsymbol{w}}\}$ are actually independent. This is because the rows of the matrix $\boldsymbol{B}$ consist solely of (scaled) unit vectors. For the Hamming metric, we have $N_{\boldsymbol{w}} = n - w$. When multiplying $\boldsymbol{B}^\top$ on the right side of $\boldsymbol{H}$, we effectively select specific columns of $\boldsymbol{H}$. Moreover, for any additional unit vector $\boldsymbol{b}$, we select another column from $\boldsymbol{H}$ to form $\boldsymbol{H} \cdot [\boldsymbol{B}^\top \mid \boldsymbol{b}^\top]$, choosing from the remaining $n - w$ columns. Given the assumption that the entries of $\boldsymbol{H}$ are independently and uniformly distributed, these selected columns are also independent.

In contrast, this independence does not hold for the rank metric. In the rank-metric case, the matrices $\boldsymbol{B}^\top$ and $[\boldsymbol{B}^\top \mid \boldsymbol{b}^\top]$ can be any full-rank matrices, rather than being limited to (scaled) unit vectors. Consequently, when multiplying these matrices on the right side of $\boldsymbol{H}$, we obtain linear combinations of the columns of $\boldsymbol{H}$ rather than simply selecting individual columns. These linear combinations introduce dependencies among the events $\mathcal{Z}_i$, violating the independence assumption.

We investigate the tightness of the upper and lower bounds on the failure probability of condition (5.5) derived in Theorem 5.6 by comparing them with simulated values and an approximation (given in (5.17)). The simulation was performed using a Monte Carlo approach with 10^5 samples for each point. Each sample involved picking a random parity-check matrix and evaluating the failure probability. The simulation was implemented using the computer-algebra system SageMath [The23].

Figure 5.3 shows parameters corresponding to the Hamming metric

$$\eta = 1,\ \ell = 24,\ n = 24,\ q = 2,\ m = 2,\ k = 8.$$

In this case, we observe that the approximation closely matches the simulated values, and both the upper and lower bounds hold. This reinforces our theory that the approximation is exact in the Hamming-metric case. In Figures 5.4 and 5.5, we increase η to $\eta = 2$ and $\eta = 3$, respectively, while keeping the code parameters constant (i.e., code length n and dimension k). As we move away from the Hamming metric by increasing η, we observe that the approximation becomes less accurate, and the lower bound provides a better estimate. In all plots, the upper bound is relatively loose compared to the lower bound.

Notably, across all scenarios, the success probability remains above 40% for error weights $w = 14$. This value represents the second-largest decodable error weight for the given code parameters and the Metzner–Kapturowski-like decoder described in Algorithm 6, where the theoretical maximum is $n - k - 1 = 15$. This level of success probability is relatively high, even near the upper limit of decodable errors.

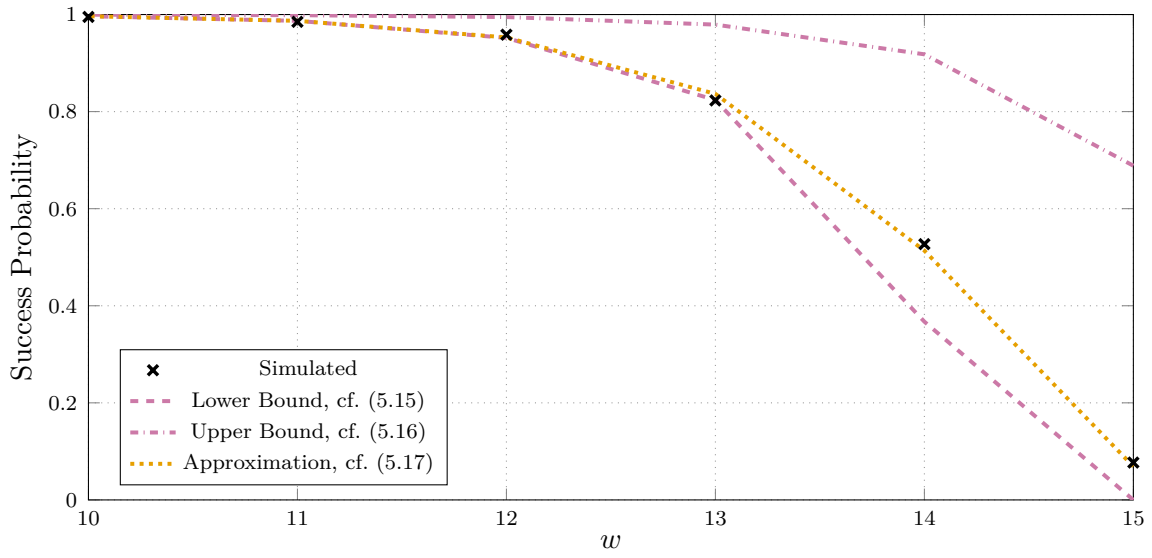


Figure 5.3: Success probability vs error weight w for $q = 2$, $m = 2$, $n = 24$, $k = 8$, $\eta = 1$, and $\ell = 24$ with interleaving order $s = w$.

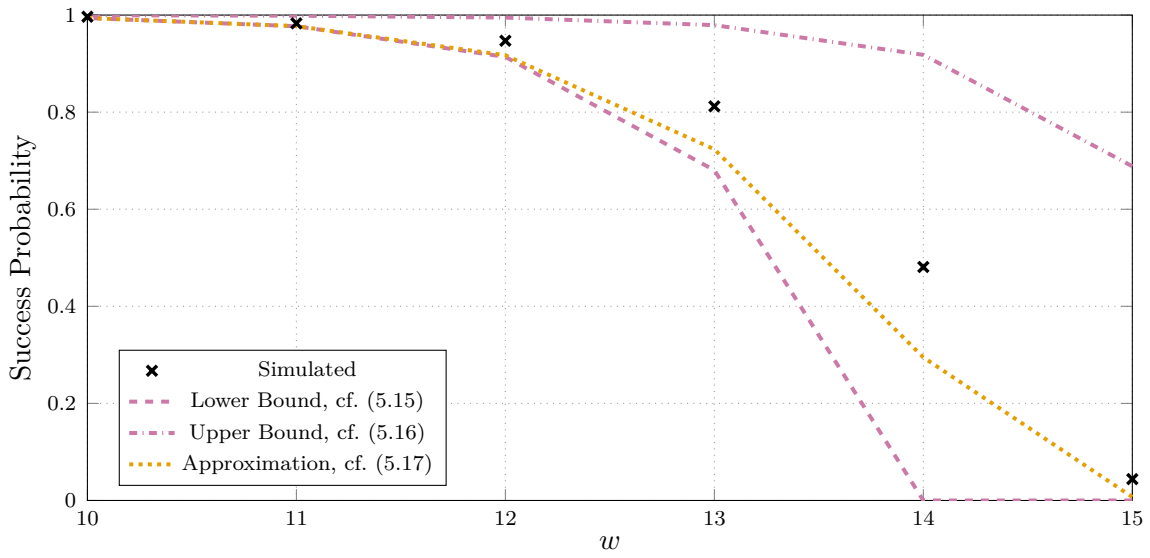


Figure 5.4: Success probability vs error weight w for $q = 2$, $m = 2$, $n = 24$, $k = 8$, $\eta = 2$, and $\ell = 12$ with interleaving order $s = w$.

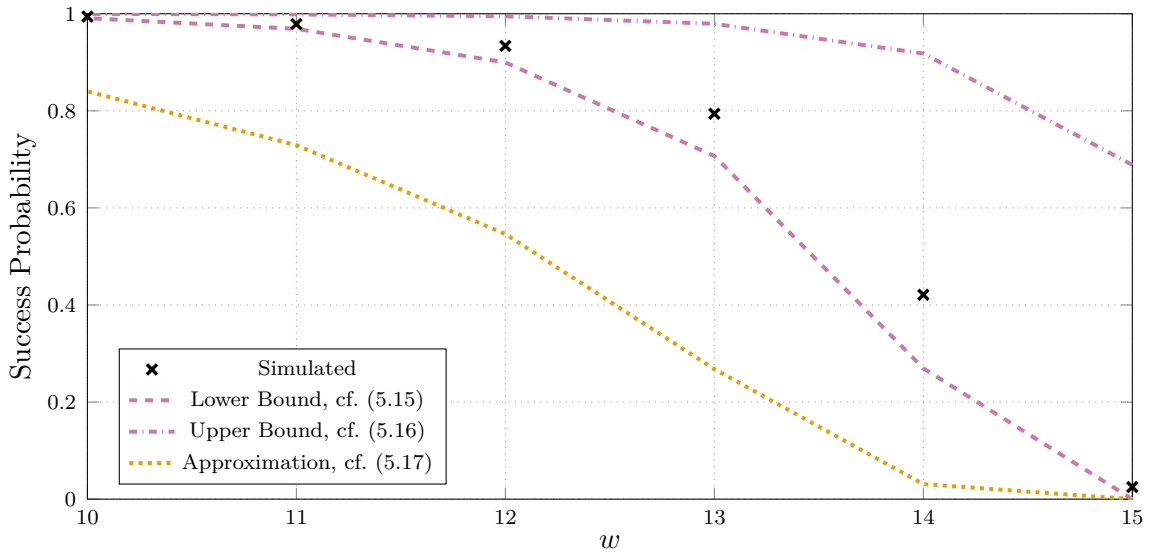


Figure 5.5: Success probability vs error weight w for $q = 2$, $m = 2$, $n = 24$, $k = 8$, $\eta = 3$, and $\ell = 8$ with interleaving order $s = w$.

5.6 Examples

In this section, we present two examples to illustrate the decoding process using Algorithm 6 with small code parameters and randomly chosen codes. The first example demonstrates a successful decoding, while the second example showcases a decoding failure where the condition in (5.5) is not satisfied.

Example 5.1 (Successful Decoding)**.** *Let $\mathbb{F}_{q^m} = \mathbb{F}_{2^3}$ with primitive element α and primitive polynomial $\alpha^3 + \alpha + 1$. Consider an interleaved sum-rank-metric code $\mathcal{IC}_{\Sigma R}[s; \boldsymbol{n}, k, d_{\min}]$ of length $n = 6$ with $\boldsymbol{n} = [2, 2, 2]$, $k = 2$, $\eta = 2$, $\ell = 3$, $d = 3$, and $s = 3$, defined by the parity-check matrix*

$$\boldsymbol{H} = \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & \alpha^2+1 & \alpha \\ 0 & 1 & 0 & 0 & 1 & \alpha^2 \\ 0 & 0 & 1 & 0 & \alpha & \alpha \\ 0 & 0 & 0 & 1 & \alpha^2+\alpha+1 & \alpha+1 \end{array}\right].$$

Suppose the codeword

$$\boldsymbol{C} = \left[\begin{array}{cc|cc|cc} \alpha^2+1 & 1 & 1 & 1 & \alpha^2+\alpha & \alpha+1 \\ \alpha+1 & \alpha^2+\alpha & \alpha^2 & \alpha & 1 & \alpha+1 \\ 0 & 0 & 1 & \alpha & \alpha^2 & 1 \end{array}\right],$$

is corrupted by an error

$$\boldsymbol{E} = \left[\begin{array}{cc|cc|cc} 0 & \alpha^2+1 & \alpha^2+1 & \alpha^2+1 & 0 & 0 \\ 1 & 0 & \alpha^2 & \alpha^2 & 0 & 0 \\ \alpha+1 & \alpha & \alpha+1 & \alpha+1 & 0 & 0 \end{array}\right],$$

with $\boldsymbol{C}\boldsymbol{H}^\top = \boldsymbol{0}$, $\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{E}) = \mathrm{wt}^{(\boldsymbol{n})}_{\Sigma R}(\boldsymbol{E}) = 3$ and $\boldsymbol{w} = [2, 1, 0]$. The received word is $\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E}$, given by

$$\boldsymbol{Y} = \left[\begin{array}{cc|cc|cc} \alpha^2+1 & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2+\alpha & \alpha+1 \\ \alpha & \alpha^2+\alpha & 0 & \alpha^2+\alpha & 1 & \alpha+1 \\ \alpha+1 & \alpha & \alpha & 1 & \alpha^2 & 1 \end{array}\right].$$

The syndrome $\boldsymbol{S} = \boldsymbol{H}\boldsymbol{Y}^\top$ is computed as

$$\boldsymbol{S} = \begin{bmatrix} 0 & 1 & \alpha+1 \\ \alpha^2+1 & 0 & \alpha \\ \alpha^2+1 & \alpha^2 & \alpha+1 \\ \alpha^2+1 & \alpha^2 & \alpha+1 \end{bmatrix}.$$

We find a matrix $\boldsymbol{P} \in \mathbb{F}_{q^m}^{(n-k)\times(n-k)}$ *with* $\mathrm{rk_q^m}(\boldsymbol{P}) = n-k = 4$*, such as*

$$\boldsymbol{P} = \begin{bmatrix} 1 & \alpha^2+1 & 0 & \alpha^2+\alpha+1 \\ \alpha+1 & \alpha^2+1 & 0 & \alpha^2+1 \\ \alpha^2+\alpha+1 & \alpha+1 & 0 & \alpha+1 \\ 0 & 0 & 1 & 1 \end{bmatrix},$$

which transforms $\boldsymbol{PS}$ *into row-echelon form.*

The last $n-k-w=1$ *rows of* $\boldsymbol{PH}$ *yield*

$$\boldsymbol{H}_{\mathcal{S}} = \left[\begin{array}{cc|cc|cc} 0 & 0 & 1 & 1 & \alpha^2+1 & 1 \end{array}\right].$$

Expanding each sub-block of $\boldsymbol{H}_{\mathcal{S}}$ *over* $\mathbb{F}_2$ *yields*

$$\mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(1)}\right) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(2)}\right) = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(3)}\right) = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Note that $\mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(1)}\right)$ *is an all-zero matrix, indicating that this block corresponds to a full-rank error.*

Next, we compute a basis for each of the right kernels of $\mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(1)}\right)$*,* $\mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(2)}\right)$*, and* $\mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(3)}\right)$ *such that*

$$\mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(1)}\right) {\boldsymbol{B}^{(1)}}^{\top} = \boldsymbol{0}, \quad \mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(2)}\right) {\boldsymbol{B}^{(2)}}^{\top} = \boldsymbol{0}, \quad \mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(3)}\right) {\boldsymbol{B}^{(3)}}^{\top} = \boldsymbol{0},$$

and

$$\begin{aligned} \mathrm{rk_q}(\boldsymbol{B}^{(1)}) &= n_1 - \mathrm{rk_q}(\boldsymbol{H}_{\mathcal{S}}^{(1)}) = 2, \\ \mathrm{rk_q}(\boldsymbol{B}^{(2)}) &= n_2 - \mathrm{rk_q}(\boldsymbol{H}_{\mathcal{S}}^{(2)}) = 1, \\ \mathrm{rk_q}(\boldsymbol{B}^{(3)}) &= n_3 - \mathrm{rk_q}(\boldsymbol{H}_{\mathcal{S}}^{(3)}) = 0. \end{aligned}$$

This gives us

$$\boldsymbol{B}^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{F}_q^{2\times 2}, \quad \boldsymbol{B}^{(2)} = \begin{bmatrix} 1 & 1 \end{bmatrix} \in \mathbb{F}_q^{1\times 2}, \quad \boldsymbol{B}^{(3)} = \begin{bmatrix} \end{bmatrix} \in \mathbb{F}_q^{0\times 2}.$$

The matrix $\boldsymbol{B}$ *is then given by*

$$\boldsymbol{B} = \mathrm{diag}\left(\boldsymbol{B}^{(1)}, \boldsymbol{B}^{(2)}, \boldsymbol{B}^{(3)}\right) = \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 \end{array}\right].$$

Finally, solving for $\boldsymbol{A}$*, i.e.,*

$$\begin{aligned}\boldsymbol{H}\boldsymbol{B}^\top\boldsymbol{A}^\top &= \boldsymbol{S}\\ \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\0&0&1\end{bmatrix}\boldsymbol{A}^\top &= \begin{bmatrix}0&1&\alpha+1\\ \alpha^2+1&0&\alpha\\ \alpha^2+1&\alpha^2&\alpha+1\\ \alpha^2+1&\alpha^2&\alpha+1\end{bmatrix},\end{aligned}$$

yields

$$\begin{aligned}\boldsymbol{A} &= \begin{bmatrix}0&\alpha^2+1&\alpha^2+1\\1&0&\alpha^2\\ \alpha+1&\alpha&\alpha+1\end{bmatrix}\\ \implies\quad \hat{\boldsymbol{E}} = \boldsymbol{A}\boldsymbol{B} &= \left[\begin{array}{cc|cc|cc}0&\alpha^2+1&\alpha^2+1&\alpha^2+1&0&0\\1&0&\alpha^2&\alpha^2&0&0\\ \alpha+1&\alpha&\alpha+1&\alpha+1&0&0\end{array}\right],\end{aligned}$$

and $\hat{\boldsymbol{E}} = \boldsymbol{E}$*. Note that decoding is possible since* $\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{H}[\boldsymbol{B}^\top \mid \boldsymbol{b}^\top]) = w+1 = 4$ *for all* $\boldsymbol{b} \in \mathbb{F}_q^{\boldsymbol{n}} \setminus \mathrm{supp}_{\Sigma R}(\boldsymbol{E})$ *such that* $\mathrm{wt}^{(\boldsymbol{n})}_{\Sigma R}(\boldsymbol{b}) = 1$*. The codeword* $\boldsymbol{C}$ *can be recovered as* $\boldsymbol{C} = \boldsymbol{Y} - \hat{\boldsymbol{E}}$*.*

Example 5.2 (Decoding Failure)**.** *Let* $\mathbb{F}_{q^m} = \mathbb{F}_{2^2}$ *with primitive element* α *and minimal polynomial* $\alpha^2+\alpha+1$*. Further let* $\mathcal{IC}_{\Sigma R}[s;\boldsymbol{n},k,d_{\min}]$ *be an interleaved sum-rank-metric code of length* $n=6$ *with* $\boldsymbol{n}=[2,2,2]$*,* $k=2$*,* $d_{\min}=4$*,* $\eta=2$*,* $\ell=3$ *and* $s=3$*, defined by the parity-check matrix*

$$\boldsymbol{H} = \left[\begin{array}{cc|cc|cc}1&0&0&\alpha+1&0&\alpha\\0&1&0&1&0&1\\0&0&1&\alpha&0&0\\0&0&0&0&1&\alpha+1\end{array}\right].$$

Suppose that the codeword

$$\boldsymbol{C} = \left[\begin{array}{cc|cc|cc}0&\alpha&1&\alpha+1&\alpha+1&1\\ \alpha&0&\alpha+1&\alpha&1&\alpha\\0&\alpha&1&\alpha+1&\alpha+1&1\end{array}\right],$$

is corrupted by an error

$$\boldsymbol{E} = \left[\begin{array}{cc|cc|cc}\alpha&0&\alpha&\alpha&0&0\\1&1&\alpha+1&\alpha+1&0&0\\ \alpha&1&\alpha+1&\alpha+1&0&0\end{array}\right],$$

with $\mathrm{rk_{q^m}}(\boldsymbol{E}) = \mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{E}) = 3$ *and* $\boldsymbol{w} = [2, 1, 0]$. *The resulting received word is then* $\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E}$ *and thus*

$$\boldsymbol{Y} = \left[\begin{array}{cc|cc|cc} \alpha & \alpha & \alpha+1 & 1 & \alpha+1 & 1 \\ \alpha+1 & 1 & 0 & 1 & 1 & \alpha \\ \alpha & \alpha+1 & \alpha & 0 & \alpha+1 & 1 \end{array}\right].$$

The syndrome is then

$$\boldsymbol{S} = \boldsymbol{H}\boldsymbol{Y}^\top = \begin{bmatrix} \alpha+1 & \alpha+1 & 0 \\ \alpha & \alpha & \alpha \\ 1 & \alpha & \alpha \\ 0 & 0 & 0 \end{bmatrix}.$$

We can find $\boldsymbol{P} \in \mathbb{F}_{q^m}^{(n-k)\times(n-k)}$ *with* $\mathrm{rk_{q^m}}(\boldsymbol{P}) = n - k = 4$, *hence*

$$\boldsymbol{P} = \begin{bmatrix} 0 & \alpha & \alpha & 0 \\ \alpha & \alpha & \alpha & 0 \\ \alpha & \alpha+1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

such that $\boldsymbol{PS}$ *is in row-echelon form. The last* $n - k - w = 1$ *rows of*

$$\boldsymbol{PH} = \left[\begin{array}{cc|cc|cc} 0 & \alpha & \alpha & 1 & 0 & \alpha \\ \alpha & \alpha & \alpha & 0 & 0 & 1 \\ \alpha & \alpha+1 & 0 & \alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \alpha+1 \end{array}\right],$$

yields

$$\boldsymbol{H}_{\mathcal{S}} = \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 1 & \alpha+1 \end{array}\right].$$

Next we expand every sub-block of $\boldsymbol{H}_{\mathcal{S}}$ *over* $\mathbb{F}_2$ *and obtain*

$$\mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(1)}\right) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(2)}\right) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(3)}\right) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

Next we compute a basis for each of the right kernels of $\mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(1)}\right)$, $\mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(2)}\right)$ *and* $\mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(3)}\right)$ *such that*

$$\mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(1)}\right) {\boldsymbol{B}^{(1)}}^\top = \boldsymbol{0}, \quad \mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(2)}\right) {\boldsymbol{B}^{(2)}}^\top = \boldsymbol{0}, \quad \mathrm{ext}\left(\boldsymbol{H}_{\mathcal{S}}^{(3)}\right) {\boldsymbol{B}^{(3)}}^\top = \boldsymbol{0},$$

and

$$\begin{aligned}\mathrm{rk_q}(\boldsymbol{B}^{(1)}) &= n_1 - \mathrm{rk_q}(\boldsymbol{H}_{\mathcal{S}}^{(1)}) = 2,\\ \mathrm{rk_q}(\boldsymbol{B}^{(2)}) &= n_2 - \mathrm{rk_q}(\boldsymbol{H}_{\mathcal{S}}^{(2)}) = 2,\\ \mathrm{rk_q}(\boldsymbol{B}^{(3)}) &= n_3 - \mathrm{rk_q}(\boldsymbol{H}_{\mathcal{S}}^{(3)}) = 0,\end{aligned}$$

which gives us

$$\boldsymbol{B}^{(1)} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \quad \boldsymbol{B}^{(2)} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \quad \boldsymbol{B}^{(3)} = \begin{bmatrix}\ & \end{bmatrix}.$$

The matrix $\boldsymbol{B}$ *is then given by*

$$\boldsymbol{B} = \mathrm{diag}\left(\boldsymbol{B}^{(1)}, \boldsymbol{B}^{(2)}, \boldsymbol{B}^{(3)}\right) = \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ \hline 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right].$$

In fact, we have that $\mathrm{rk_q}(\boldsymbol{B}^{(1)}) + \mathrm{rk_q}(\boldsymbol{B}^{(2)}) + \mathrm{rk_q}(\boldsymbol{B}^{(3)}) = 4 > w = 3$*, and therefore we cannot uniquely recover the error* $\boldsymbol{E}$ *anymore. This is because the decoding condition in* (5.5) *is not satisfied, since there exists* $\boldsymbol{b} \in \mathbb{F}_q^{\boldsymbol{n}} \setminus \mathrm{supp}_{\Sigma R}(\boldsymbol{E})$ *such that* $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{b}) = 1$ *and* $\mathrm{rk_{q^m}}(\boldsymbol{H}[\boldsymbol{B}^\top \mid \boldsymbol{b}^\top]) \neq w + 1 = 4$*. That is, for* $\boldsymbol{b} = \left[\begin{array}{cc|cc|cc} 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right]$*, we have*

$$\boldsymbol{H}\left[\boldsymbol{B}^\top \mid \boldsymbol{b}^\top\right] = \begin{bmatrix}1 & 0 & \alpha+1 & 0\\ 1 & 1 & 1 & 0\\ 0 & 0 & \alpha+1 & 1\\ 0 & 0 & 0 & 0\end{bmatrix} \quad \Longrightarrow \quad \mathrm{rk_{q^m}}\left(\boldsymbol{H}\left[\boldsymbol{B}^\top \mid \boldsymbol{b}^\top\right]\right) = 3 < 4.$$

5.7 Special Cases of the Algorithm for Hamming and Rank Metric

The decoder described in Algorithm 6 extends the Metzner–Kapturowski decoder originally developed for the Hamming metric [MK90] as well as the Metzner–Kapturowski-like decoder for the rank metric [PRW19; RPW21a] to the sum-rank metric. In this section, we outline the differences in the operation of the proposed decoder across three metrics: the Hamming metric, the rank metric, and the sum-rank metric. It is important to note that both the Hamming and rank metrics can be considered special cases of the sum-rank metric. We also highlight the similar definitions of error support applicable to all three metrics. To clarify the distinction between the error weights in each metric, we introduce the following notation: w_H for the Hamming metric, w_R for the rank metric, and $w_{\Sigma R}$ for the sum-rank metric.

In the Hamming metric (i.e. $\ell = n$ and $n_i = 1$), the support of an error matrix $\boldsymbol{E}$ is defined as the set of indices corresponding to the nonzero columns of $\boldsymbol{E}$, that is,

$$\operatorname{supp}_H(\boldsymbol{E}) \stackrel{\text{def}}{=} \{j \,:\, \text{the } j\text{-th column of } \boldsymbol{E} \text{ is nonzero}\}.$$

However, this classical notion of support does *not* directly align with the definition of sum-rank support given in Definition 2.12. Nonetheless, a one-to-one correspondence exists between the two concepts. We establish this by first describing $\operatorname{supp}_{\Sigma R}(\boldsymbol{E})$ and then linking it to $\operatorname{supp}_H(\boldsymbol{E})$. Since each of the blocks

$$\boldsymbol{E}^{(1)}, \ldots, \boldsymbol{E}^{(\ell)} \ \in \mathbb{F}_{q^m}^{s\times 1},$$

has length one (i.e. consist of one column) in the Hamming-metric setting, at most one rank error can occur per block. Thus, the i-th block $\boldsymbol{B}^{(i)}$ in the error decomposition

$$\boldsymbol{E} = \boldsymbol{A} \cdot \operatorname{diag}(\boldsymbol{B}^{(1)}, \ldots, \boldsymbol{B}^{(\ell)}),$$

from (2.30) has size $w_H^{(i)} \times 1$ with $w_H^{(i)} \in \{0, 1\}$. If the i-th block for an $i \in \{1, \ldots, \ell\}$ is erroneous, the matrix $\boldsymbol{B}^{(i)}$ contains one nonzero $\mathbb{F}_q$ element, which implies

$$\mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big) = \mathbb{F}_q.$$

If, on the other hand, the block $\boldsymbol{E}^{(i)}$ is error-free, the matrix $\boldsymbol{B}^{(i)}$ has size 0×1 and its row space $\mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big)$ is the trivial vector space $\{\boldsymbol{0}\} \subseteq \mathbb{F}_q$. Thus, the sum-rank support

$$\operatorname{supp}_{\Sigma R}(\boldsymbol{E}) = \mathcal{R}_q\Big(\boldsymbol{B}^{(1)}\Big) \times \cdots \times \mathcal{R}_q\Big(\boldsymbol{B}^{(\ell)}\Big),$$

of $\boldsymbol{E}$ is a Cartesian product containing copies of $\mathbb{F}_q \neq \{\boldsymbol{0}\}$ and $\{\boldsymbol{0}\}$ in the respective positions. This allows us to define a bijection between the sum-rank support and the classical definition of Hamming support given above. Namely,

$$\operatorname{supp}_H(\boldsymbol{E}) \mapsto \operatorname{supp}_{\Sigma R}(\boldsymbol{E}) = \boldsymbol{X}_1 \times \boldsymbol{X}_2 \times \cdots \times \boldsymbol{X}_n,$$

with

$$\boldsymbol{X}_i = \begin{cases} \mathbb{F}_q & \text{if } i \in \operatorname{supp}_H(\boldsymbol{E}) \\ \{\boldsymbol{0}\} & \text{if } i \notin \operatorname{supp}_H(\boldsymbol{E}) \end{cases},$$

maps a subset of the indices $\{1, \ldots, n\}$ to the corresponding sum-rank support contained in $\mathbb{F}_q^{\boldsymbol{n}}$ with $\boldsymbol{n} = [1, \ldots, 1] \in \mathbb{Z}_{\geq 0}^n$. We stick to $\operatorname{supp}_H(\boldsymbol{E})$ to explain the consequences for decoding in the Hamming metric in the following.

An error matrix $\boldsymbol{E}$ with w_H errors in the Hamming metric can be factored into $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B}$, where the rows of $\boldsymbol{B}$ are (scaled) unit vectors corresponding to the w_H

error positions. Consequently, the support of $\boldsymbol{E}$ is the union of the supports of the rows $\boldsymbol{B}_i$ ($\forall i \in \{1, \ldots, w_H\}$) of $\boldsymbol{B}$, i.e.,

$$\operatorname{supp}_H(\boldsymbol{E}) = \bigcup_{i=1}^{w_H} \operatorname{supp}_H(\boldsymbol{B}_i).$$

When the full-rank condition for the Metzner–Kapturowski decoder is satisfied, the zero columns in $\boldsymbol{H}_{\mathcal{S}}$ reveal the error positions and determine the error support. In this case, we have

$$\operatorname{supp}_H(\boldsymbol{E}) = \{1, \ldots, n\} \setminus \bigcup_{i=1}^{n-k-w_H} \operatorname{supp}_H(\boldsymbol{H}_{\mathcal{S},i}),$$

where $\boldsymbol{H}_{\mathcal{S},i}$ denotes the i-th row of $\boldsymbol{H}_{\mathcal{S}}$. Note that this equality corresponds to (5.7) in the general case. The process of recovering the error support $\operatorname{supp}_H(\boldsymbol{E})$ from $\boldsymbol{H}_{\mathcal{S}}$ is depicted in Figure 5.6.

The rank-metric case (i.e., $\ell = 1$ and $n_1 = n$) is analogous to the Hamming-metric case, with a different definition of error support. For an error matrix $\boldsymbol{E}$ with rank $\operatorname{rk_q}(\boldsymbol{E}) = w_R$, we can decompose it as $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B}$. The rank support $\operatorname{supp}_R(\boldsymbol{E})$ of $\boldsymbol{E}$ is defined as the row space of $\boldsymbol{B}$, spanned by the union of all rows $\boldsymbol{B}_i$ of $\boldsymbol{B}$, where $\boldsymbol{B}_i$ is the i-th row of $\boldsymbol{B}$. This definition matches exactly with the more general one in the sum-rank metric from Definition 2.12 when $\ell = 1$. Thus, the support of $\boldsymbol{E}$ is given by

$$\operatorname{supp}_R(\boldsymbol{E}) = \bigoplus_{i=1}^{w_R} \operatorname{supp}_R(\boldsymbol{B}_i),$$

where $\bigoplus$ denotes the addition of vector spaces, i.e., the span of the union of the considered spaces. If the full-rank condition on the error matrix is satisfied, the rank support of $\boldsymbol{E}$ can be determined by the $\mathbb{F}_q$-kernel of $\boldsymbol{H}_{\mathcal{S}}$ (see [RPW21a]).The $\mathbb{F}_q$-row space of $\boldsymbol{H}_{\mathcal{S}}$ can be computed by taking the span of the union of spaces $\operatorname{supp}_R(\boldsymbol{H}_{\mathcal{S},i})$, where $\boldsymbol{H}_{\mathcal{S},i}$ is the i-th row of $\boldsymbol{H}_{\mathcal{S}}$. Consequently, the support of $\boldsymbol{E}$ is given by

$$\operatorname{supp}_R(\boldsymbol{E}) = \left(\bigoplus_{j=1}^{n-k-w_R} \operatorname{supp}_R(\boldsymbol{H}_{\mathrm{sub},j}) \right)^{\perp}.$$

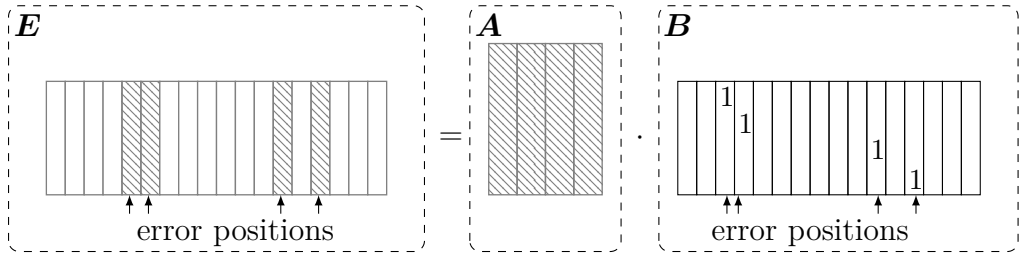
In the sum-rank metric, according to Definition 2.12, we have

$$\begin{aligned} \operatorname{supp}_{\Sigma R}(\boldsymbol{E}) &= \operatorname{supp}_R\left(\boldsymbol{B}^{(1)}\right) \times \operatorname{supp}_R\left(\boldsymbol{B}^{(2)}\right) \times \cdots \times \operatorname{supp}_R\left(\boldsymbol{B}^{(s)}\right) \\ &= \left(\bigoplus_{j=1}^{n-k-w_{\Sigma R}} \operatorname{supp}_R(\boldsymbol{B}_j^{(1)}) \right) \times \cdots \times \left(\bigoplus_{j=1}^{n-k-w_{\Sigma R}} \operatorname{supp}_R(\boldsymbol{B}_j^{(\ell)}) \right). \end{aligned}$$

Based on Theorem 5.1, we have

$$\operatorname{supp}_{\Sigma R}(\boldsymbol{E}) = \left(\bigoplus_{j=1}^{n-k-w_{\Sigma R}} \operatorname{supp}_R(\boldsymbol{H}_{\mathcal{S},j}^{(1)}) \right)^{\perp} \times \ldots \\ \cdots \times \left(\bigoplus_{j=1}^{n-k-w_{\Sigma R}} \operatorname{supp}_R(\boldsymbol{H}_{\mathcal{S},j}^{(s)}) \right)^{\perp}.$$

The relation between the error matrix $\boldsymbol{E}$, the matrix $\boldsymbol{H}_{\mathcal{S}}$, and the error supports for the Hamming metric, rank metric, and sum-rank metric are illustrated in Figure 5.6, Figure 5.7, and Figure 5.8, respectively. In particular, Figure 5.8 demonstrates the process of determining the sum-rank support $\operatorname{supp}_{\Sigma R}(\boldsymbol{E})$ from the row spaces of the blocks $\boldsymbol{H}_{\mathcal{S}}^{(i)}$ for $i \in \{1, \ldots, \ell\}$.



$$\boldsymbol{H}_{\mathcal{S}} = \quad \Rightarrow \operatorname{supp}_H(\boldsymbol{E}) = \bigcup_{i=1}^{w_H} \operatorname{supp}_H(\boldsymbol{B}_i) =$$

all-zero columns in error positions

$$= \{1, \ldots, n\} \setminus \bigcup_{i=1}^{n-k-w_H} \operatorname{supp}_H(\boldsymbol{H}_{\mathcal{S},i})$$

Figure 5.6: Illustration of the error support for the Hamming-metric case with an error matrix $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B} \in \mathbb{F}_{q^m}^{s \times n}$, $\boldsymbol{A} \in \mathbb{F}_{q^m}^{s \times w_H}$, $\boldsymbol{B} \in \mathbb{F}_q^{w_H \times n}$ and a parity-check matrix $\boldsymbol{H}_{\mathcal{S}} \in \mathbb{F}_{q^m}^{(n-k-w_H) \times n}$. $\boldsymbol{B}_i$ is the i-th row of $\boldsymbol{B}$ and $\boldsymbol{H}_{\mathcal{S},i}$ the i-th row of $\boldsymbol{H}_{\mathcal{S}}$.

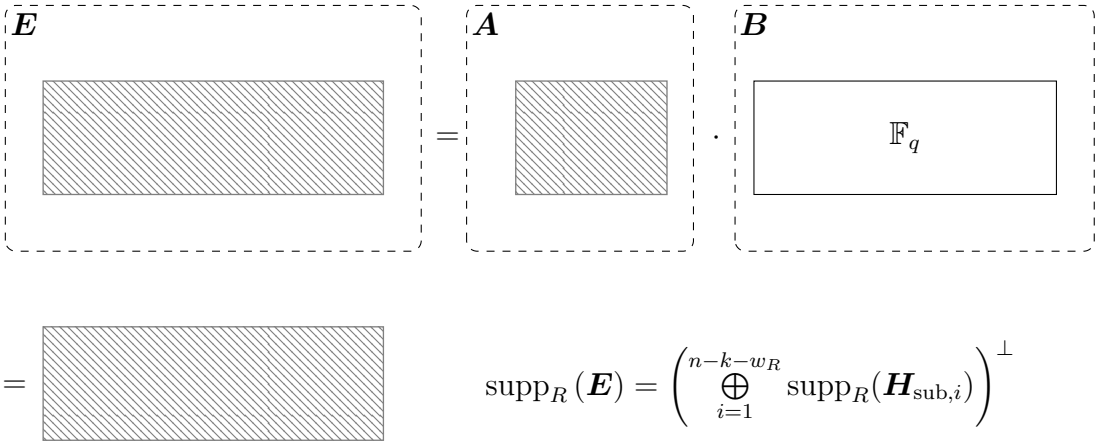


Figure 5.7: Illustration of the error support for the rank-metric case with an error matrix $\boldsymbol{E} = \boldsymbol{AB} \in \mathbb{F}_{q^m}^{s\times n}$, $\boldsymbol{A} \in \mathbb{F}_{q^m}^{s\times w_R}$, $\boldsymbol{B} \in \mathbb{F}_q^{w_R\times n}$ and a parity-check matrix $\boldsymbol{H}_{\mathcal{S}} \in \mathbb{F}_{q^m}^{(n-k-w_R)\times n}$. $\boldsymbol{H}_{\text{sub},i}$ the i-th row of $\boldsymbol{H}_{\mathcal{S}}$.

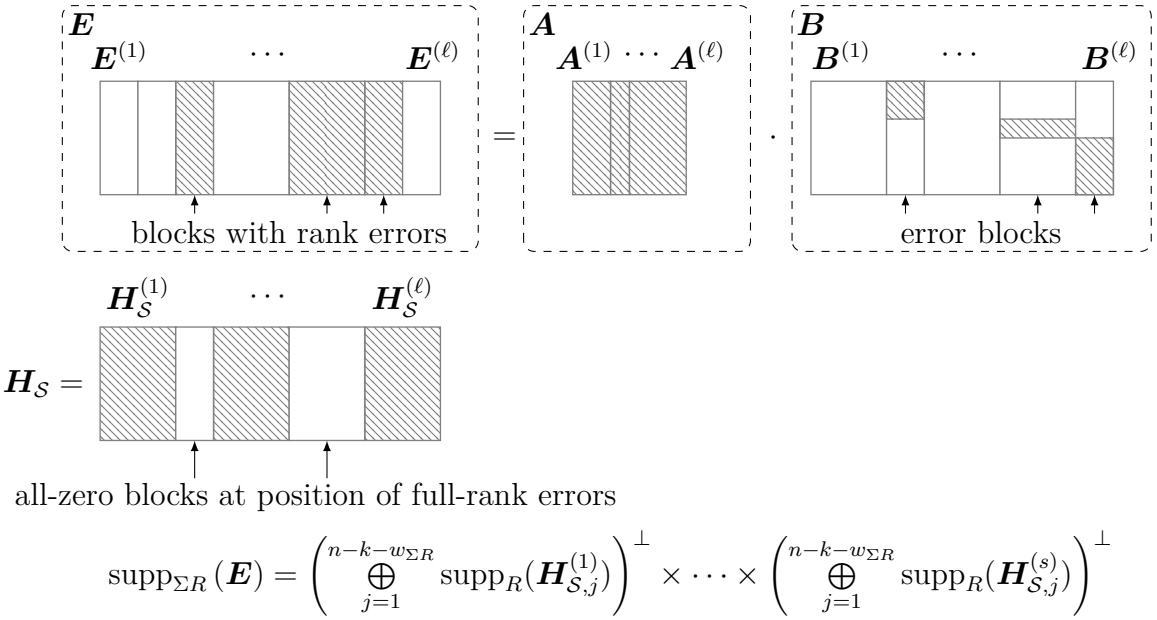


Figure 5.8: Illustration of the error support for the sum-rank-metric case with an error matrix $\boldsymbol{E} = \boldsymbol{AB} \in \mathbb{F}_{q^m}^{s\times n}$, $\boldsymbol{A} \in \mathbb{F}_{q^m}^{s\times w_{\Sigma R}}$, $\boldsymbol{B} \in \mathbb{F}_q^{w_{\Sigma R}\times n}$ and a parity-check matrix $\boldsymbol{H}_{\mathcal{S}} \in \mathbb{F}_{q^m}^{(n-k-w_{\Sigma R})\times n}$. $\boldsymbol{A}^{(i)}$ and $\boldsymbol{B}^{(i)}$ are the i-th block of $\boldsymbol{A}$ and $\boldsymbol{B}$ and $\boldsymbol{H}_{\mathcal{S},j}^{(i)}$ the j-th row of $\boldsymbol{H}_{\mathcal{S}}^{(i)}$.

5.8 Connection to the Loidreau–Overbeck Decoder

In this section, we explore the connection between the success conditions of the Loidreau–Overbeck decoder and the Metzner–Kapturowski-like decoder.

As introduced in (3.3), the matrix

$$\tilde{\boldsymbol{L}} = \begin{bmatrix} \mathfrak{M}_{n-w-k}(\boldsymbol{e}_1)_{\boldsymbol{\xi}} \\ \mathfrak{M}_{n-w-k}(\boldsymbol{e}_2)_{\boldsymbol{\xi}} \\ \vdots \\ \mathfrak{M}_{n-w-k}(\boldsymbol{e}_s)_{\boldsymbol{\xi}} \end{bmatrix} \in \mathbb{F}_{q^m}^{s(n-w-k)\times n},$$

determines the success of the Loidreau–Overbeck decoder, as described in Section 3.1.2. This decoder succeeds if the matrix $\tilde{\boldsymbol{L}}$ has rank equal to w.

Note that the first row of each submatrix $\mathfrak{M}_{n-w-k}(\boldsymbol{e}_i)_{\boldsymbol{\xi}}$ in $\tilde{\boldsymbol{L}}$ is equal to $\boldsymbol{e}_i$. Therefore, the matrix

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_s \end{bmatrix},$$

is a submatrix of $\tilde{\boldsymbol{L}}$. The Metzner–Kapturowski-like decoder succeeds if the matrix $\boldsymbol{E}$ has rank w.

Since $\boldsymbol{E}$ is a submatrix of $\tilde{\boldsymbol{L}}$, the success of the Metzner–Kapturowski-like decoder (i.e., $\boldsymbol{E}$ has rank w) implies that $\tilde{\boldsymbol{L}}$ also contains a submatrix of rank w. Thus, the success of this decoder is sufficient for the success of the Loidreau–Overbeck decoder. However, the condition for the Loidreau–Overbeck decoder (i.e., $\tilde{\boldsymbol{L}}$ having rank w) does not guarantee the success of the Metzner–Kapturowski-like decoder, as $\boldsymbol{E}$ could have a lower rank even if $\tilde{\boldsymbol{L}}$ meets the rank requirement.

This demonstrates a hierarchical relationship between the two decoders: while the Loidreau–Overbeck decoder may handle broader conditions, it is specifically designed for ILRS codes, whereas the Metzner–Kapturowski-like decoder is more general and can be applied to an interleaved sum-rank-metric code with any constituent code.

5.9 Summary and Discussion

In this chapter, we investigated a Metzner–Kapturowski-like decoding algorithm tailored for high-order interleaved sum-rank-metric codes. By introducing the novel concept of an error code, we provided a fresh perspective on the decoding process, enhancing our understanding of the decoder's functionality and offering new insights.

Our proposed algorithm demonstrates significant versatility, being applicable to any linear constituent code, including those that are unstructured or random. This gen-

eral applicability positions our decoder as a robust tool for cryptanalysis of code-based cryptosystems that utilize high order interleaving to reduce public key sizes. The ability to decode codes with arbitrary constituent codes suggests potential vulnerabilities in cryptosystems that rely on interleaving for security.

We analyzed the computational complexity of our algorithm, which is on the order of $O(\max\{n^3, n^2 s\})$ operations over $\mathbb{F}_{q^m}$, and found it to be independent of the structure of the constituent codes. This independence is particularly advantageous in cryptanalytic applications, where the codes may be designed to obscure their structure.

Furthermore, we explored the success probability of our decoder both within and beyond the error weight bound where unique decoding is guaranteed (up to $d_{\min} - 2$). Our analysis revealed that the decoder maintains a high success probability even for error weights exceeding this limit, as illustrated in Figure 5.9. This property is crucial for cryptanalysis, as it enables the decoding of ciphertexts encrypted with higher error weights, potentially compromising the security of the cryptosystem. Notably, since the sum-rank metric generalizes both the Hamming and rank metrics, our results also apply to these metrics, highlighting vulnerabilities in cryptosystems that utilize high interleaving in these contexts as well.

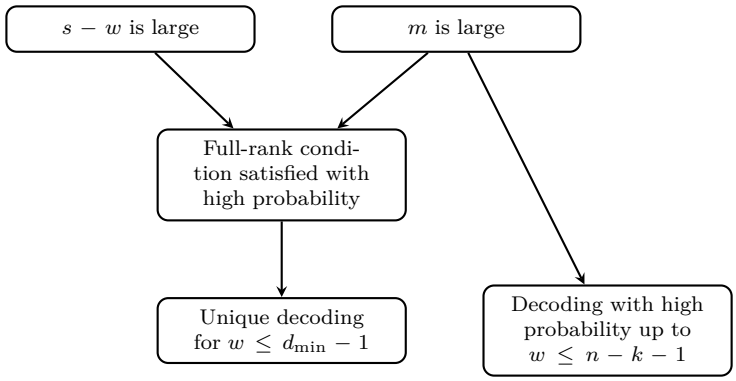


Figure 5.9: Relationships between parameters, conditions, and decoding success.

Our work extends previous studies and provides valuable insights for the design and security analysis of future code-based cryptosystems based on interleaving and the sum-rank metric. The decoder's guaranteed success for error weights up to $d_{\min} - 2$ under the full-rank condition, and its high success probability beyond this range, highlight the need for careful consideration when selecting code parameters for cryptographic purposes.

The algorithm considered in this work is designed for interleaved codes where the constituent codes are aligned vertically, also known as vertically interleaved codes. From the error code perspective, this means that the error code's row support is restricted. In an alternative model, codewords could be aligned horizontally (horizontal interleaving), resulting in the error code's column support being restricted. Adapting

the algorithm to this horizontal interleaving model is not straightforward and remains an open problem.

Future research could explore further optimizations of the decoding algorithm and its application to other metrics, such as the Lee metric [Lee58]. Additionally, investigating the adaptation of the algorithm to horizontal interleaving could reveal new avenues for cryptanalysis and contribute to the development of more secure code-based cryptosystems.

6

Support-Guessing Decoding Algorithms in the Sum-Rank Metric

Decoding problems in the sum-rank metric are crucial for the security analysis of code-based cryptosystems that utilize this metric. Continuing our focus on non-interleaved codes and generic decoding problems, in this chapter we further investigate decoding challenges within the sum-rank metric.

Efficient generic decoding algorithms have been extensively developed for the Hamming metric—such as Prange's algorithm [Pra62], Stern's algorithm [Ste89], the May–Meurer–Thomae algorithm (MMT) [MMT11], the Becker–Joux–May–Meurer algorithm (BJMM) [BJMM12], and many others. Significant progress has also been made in the rank metric with algorithms like Chabaud and Barbier's algorithm [CS96], combinatorial approaches by Gaborit et al. [GRS16], and improved syndrome decoding algorithms by Aragon et al. [AGHT17]. However, there has been relatively little advancement in developing such algorithms for the sum-rank metric, which bridges the Hamming and rank metrics. Notably, Puchinger et al. [PRR22] have initiated work in this area by providing upper and lower bounds on the decoding complexity based on worst-case rank profiles. Their analysis focuses on the worst-case scenarios over all possible rank profiles, offering valuable insights into the theoretical limits of decoding in the sum-rank metric. However, this worst-case perspective may not accurately reflect the average decoding complexity encountered in potential cryptographic applications. Therefore, further efforts are needed to understand and improve the complexity of decoding in the sum-rank metric from an average-case perspective.

In this chapter, we advance the complexity analysis of decoding problems in the sum-rank metric, addressing key limitations in existing work and providing insights essential for cryptographic applications. Specifically, we focus on improving the understanding of the expected decoding complexity for generic decoding algorithms by transitioning from worst-case to average-case analysis and introducing decoding algorithms for LRS codes beyond the unique decoding radius. The proposed algorithm for LRS codes

offers significant complexity reductions compared to fully generic decoding algorithms that do not exploit the underlying code structure.

While the lower bound presented in [PRR22] is tight within the unique decoding radius—where at most one decoding solution exists—it becomes less applicable beyond that radius. Beyond the unique decoding radius, multiple decoding solutions may exist, rendering the worst-case lower bound less representative of actual decoding complexity. This scenario is illustrated in Figure 6.1, where decoding beyond the unique radius leads to more than one possible solution. In such cases, the complexity of finding a solution can be significantly lower than the worst-case bound suggests. Relying solely on the previous lower bound may lead to overestimating the decoding complexity and underestimating an attacker's actual capabilities, potentially resulting in insecure parameter choices.

For cryptographic applications, especially in selecting key sizes and system parameters, it is crucial to establish lower bounds on the decoding complexity that reflect the minimum difficulty an attacker would face. Such bounds provide insights into the best-case scenario for the attacker, ensuring that security assessments are based on realistic estimates of the attack effort. By adopting an average-case analysis, we obtain more precise and realistic estimates of the decoding complexity over all possible error patterns, not just the worst-case scenarios. This ensures that the parameters chosen for a cryptosystem provide the desired level of security, safeguarding against potential attacks that exploit the existence of multiple decoding solutions.

In cryptography, it is essential that the complexity of the best-known attack does not lead to a lower security level than intended. This requires that the lower bound on the expected decoding complexity is sufficiently high to prevent efficient attacks. An accurate average-case analysis enables that cryptographic parameters can be chosen such that no known attack can break the system more efficiently than intended.

In this work, we extend the analysis to the average-case scenario by considering all possible rank profiles and deriving a tailored support-guessing distribution that optimizes the expected decoding complexity in the asymptotic setting. Specifically, we consider the case where the number of blocks tends to infinity, which allows us to assume independence between the probabilities of each block having a certain rank. With this independence assumption, we derive an optimal support-guessing distribution applicable to finite-length scenarios. Numerical evaluations demonstrate that this asymptotically derived distribution closely matches and often coincides with the distribution obtained by considering the dependencies between the blocks in the finite-length case, even though deriving the latter is more complex. Our analysis yields exact complexity values for unique decoding cases and provides tighter upper and lower bounds valid for decoding beyond the unique decoding radius, as illustrated in Figure 6.1. We introduce a new lower bound that accounts for alternative decoding solutions using random-coding union (RCU) bound arguments, offering a more precise estimation of decoding complexity in this regime.

In addition to the average-case analysis of the generic decoding algorithm, we investigate decoding LRS codes beyond the unique decoding radius. Previous work [RJB+20] introduced a randomized decoding algorithm for Gabidulin codes, which are the rank metric equivalent of LRS codes. We adapt and generalize this approach to the sum-rank metric for LRS codes. This work is based on our previous publications [JBW23; JBW24].

Both the generic decoder and the randomized decoder for LRS codes are support-guessing algorithms. They operate by randomly guessing the support of the error according to a predefined probability distribution, which needs to be optimized. In each iteration of the guessing loop, a decoding step is performed. For the generic decoder, this decoding step is an erasure decoder, requiring the guessed support to fully contain the actual error support to succeed. In contrast, the randomized decoder for LRS codes employs an error-and-erasure decoder, which only requires that the intersection between the guessed support and the actual error support is sufficiently large for successful decoding.

The error-and-erasure decoder leverages the underlying algebraic structure of LRS codes, allowing it to succeed in cases where the fully generic decoder would fail. By exploiting this structure, our randomized decoder achieves a higher success probability and lower expected computational complexity. Although the randomized decoding approach can conceptually be applied to any code equipped with an error-and-erasure decoder, we focus on LRS codes due to their significance in the sum-rank metric.

By combining the optimized average-case analysis of the generic decoder with the enhanced capabilities of the randomized decoder for LRS codes, we offer a comprehensive understanding of decoding complexities in the sum-rank metric. This work aligns with the overarching narrative of this thesis, which focuses on developing robust complexity analyses and efficient decoding algorithms for codes in the sum-rank metric.

Most parts of this chapter are based on [JBW24], submitted and currently under review at IEEE Transactions on Information Theory. The chapter presents tighter upper bounds on the worst-case setting, transitions from worst-case to average-case complexity analysis, and introduces a new support-drawing distribution for both fully generic and randomized decoding of LRS codes. These and all other new contributions of [JBW24] were the ideas of the author of this dissertation and were solely developed and worked out by the author.

Section 6.5 is primarily based on [JBW23] and [JBW24]. The conference version [JBW23] was first presented at the IEEE International Symposium on Information Theory (ISIT 2023) and was later extended in the journal version as discussed above [JBW24].

The concept of the randomized decoder, which combines an error-and-erasure decoder with a support-guessing approach, was first proposed in the work presented at the Code-Based Cryptography Conference (CBCrypto 2019) [JB19]. In that paper, we

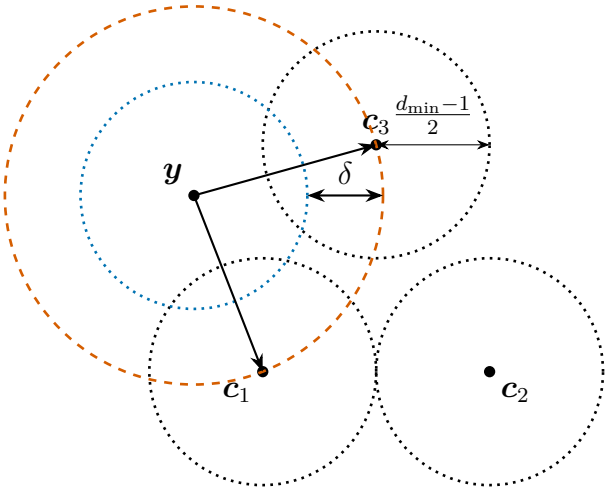


Figure 6.1: Illustration of decoding ambiguity in the 2D plane, where an error $\boldsymbol{e}$ exceeding the unique decoding radius causes the received point $\boldsymbol{y}$ to be equidistant from multiple codewords.

introduced the core idea of the decoder, focusing on a specific type of support related to criss-cross erasures. This initial concept was primarily developed by the author of this dissertation and is briefly discussed in Section 6.5.7. The analysis of the decoder was later generalized to handle a broader notion of support in the rank metric, as presented at the international conference on Post-Quantum Cryptography (PQCrypto 2020). The author contributed significantly to this generalization, including discussions, insights, algorithmic implementation, and simulations.

The work was further generalized to the sum-rank metric for LRS codes in [JBW23], and extended to average-case complexity analysis in the journal version [JBW24], as discussed earlier. These extensions enhance the understanding of decoding complexities and contribute valuable insights to the broader field of sum-rank-metric codes.

6.1 Overview of Decoding Problems

This section presents and categorizes decoding problems in the sum-rank metric, relevant to both coding theory and cryptographic applications. In Section 2.4.2, we introduced various decoding concepts aimed at finding an ”optimal” solution by minimizing either a distance metric or a likelihood.

In this chapter, we focus on problems where the error weight w is assumed to be known. This assumption is justified by the fact that one can iterate through a set of potential weights, i.e., $w \in \{0, \dots, n\}$, which has a cardinality polynomially bounded in n. The solution corresponding to the desired criterion, such as minimal weight, can then be selected. We begin with the most general decoding problem and then proceed

to more specific cases, formulating the corresponding problems along the way.

6.1.1 Sum-Rank Syndrome Decoding Problem

The *sum-rank syndrome decoding problem* is a generalization of both the syndrome decoding problem in the Hamming metric and the rank syndrome decoding problem, thereby covering a broad range of cryptosystems that rely on the hardness of this problem. Examples include WAVE [DST19], BIKE [ABB+20], and HQC [MAB+24] (based on the Hamming metric), as well as RQC [MAB+20] (based on the rank metric).

Problem 6.1 (Sum-Rank Syndrome Decoding Problem)**.**

- ***Instance:***
  - *A linear sum-rank-metric code* $\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k] \subseteq \mathbb{F}_{q^m}^{n}$ *with parity-check matrix* $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$*,*
  - *A syndrome* $\boldsymbol{s} \in \mathbb{F}_{q^m}^{n-k}$ *and an integer* $w > 0$*.*
- ***Objective:*** *Find an error vector* $\boldsymbol{e}$ *such that* $\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^\top$ *and* $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}) = w$*.*

The sum-rank syndrome decoding problem is particularly interesting due to its applicability across different metrics and cryptosystems. A solution to this problem is not guaranteed.

The algorithm introduced in [PRR22] provides a generic decoding approach that addresses the sum-rank syndrome decoding problem for error weights up to $n - k$.

6.1.2 Decoding Beyond the Unique Radius

The following problem can be seen as a special case of the sum-rank syndrome decoding problem (see Problem 6.1). In this case, we assume that a specific codeword, corrupted by an error of known weight, has been received. Under this assumption, the weight of the error, denoted by w, is known, and we set the decoding radius accordingly to w. This ensures that at least one solution exists, although it may not be unique, similar to the general syndrome decoding problem. We first consider the problem of decoding beyond the unique radius for arbitrary sum-rank-metric codes.

Problem 6.2 (Beyond Unique Decoding for Sum-Rank-Metric Codes)**.**

- ***Instance:***
  - *Linear sum-rank-metric code* $\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k] \subseteq \mathbb{F}_{q^m}^{n}$ *with unique decoding radius* τ*,*
  - *Error vector* $\boldsymbol{e} \xleftarrow{\$} \mathcal{E}_w$ *with* $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}) = w \geq \tau$*,*
  - *Received vector* $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ *with* $\boldsymbol{c} \in \mathcal{C}_{\Sigma R}[\boldsymbol{n}, k]$*.*

- ***Objective:*** *Find a codeword* $\boldsymbol{c} \in \mathcal{C}_{\Sigma R}[\boldsymbol{n}, k]$*, such that*

$$\mathrm{wt}_{\Sigma R}(\boldsymbol{y} - \boldsymbol{c}) = w.$$

Problem 6.2 extends the unique decoding problem by allowing error weights that exceed the unique decoding radius. Unlike unique decoding, multiple codewords may satisfy the decoding condition, and a solution is not guaranteed to be unique. The generic decoder introduced in [PRR22] can efficiently address this problem for any linear sum-rank-metric code without relying on its structure, handling error weights up to $n - k$.

Next, we specialize this problem for LRS codes.

Problem 6.3 (Beyond Unique Decoding for LRS codes)**.**

- ***Instance:***
  - *LRS code* $\mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; \boldsymbol{n}, k] \subseteq \mathbb{F}_{q^m}^n$*,* $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$*,*
  - *Error vector* $\boldsymbol{e} \xleftarrow{\$} \mathcal{E}_w$ *with* $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}) = w \geq \tau$*,*
  - *Received vector* $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e} \in \mathbb{F}_{q^m}^n$ *with* $\boldsymbol{c} \in \mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; \boldsymbol{n}, k]$*.*
- ***Objective:*** *Find a codeword* $\boldsymbol{c} \in \mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; \boldsymbol{n}, k]$*, such that*

$$\mathrm{wt}_{\Sigma R}(\boldsymbol{y} - \boldsymbol{c}) = w.$$

Problem 6.3 is a specialization of Problem 6.2 for LRS codes. This specialization allows us to exploit the inherent structure of LRS codes, potentially leading to more efficient decoding algorithms. We compare the complexity of the generic decoder for Problem 6.2, as introduced in [PRR22], with our proposed randomized decoder (see Section 6.5) tailored for LRS codes.

The Faure–Loidreau (FL) system is a rank-metric code-based cryptosystem that uses Gabidulin codes, which are a special case of LRS codes. Problem 6.3 is itself a generalization of the decoding problem considered in [RJB+20], which focused on Gabidulin codes. Our proposed randomized decoder generalizes the decoder introduced in [RJB+20] to the sum-rank metric. We first introduced this generalization in [JBW23].

Our proposed decoder and the complexity analysis for the sum-rank metric could be valuable for future cryptosystems that are similar to the FL system but operate in the sum-rank metric, or for different cryptosystems that rely on problems like Problem 6.3 in the sum-rank metric.

6.1.3 Unique Decoding Problem

The *unique decoding problem* (also known as *bounded minimum distance decoding problem*) is a specific case of Problem 6.2, where the error weight does not exceed the unique decoding radius, ensuring that there is exactly one solution within this radius.

Problem 6.4 (Unique Decoding Problem)**.**

- ***Instance:***
  - *A linear sum-rank-metric code* $\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k] \subseteq \mathbb{F}_{q^m}^n$ *with unique decoding radius* τ*.*
  - *An error vector* $\boldsymbol{e} \xleftarrow{\$} \mathcal{E}_w$ *with* $w = \mathrm{wt}_{\Sigma R}(\boldsymbol{e}) \leq \tau$*.*
  - *A received vector* $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ *where* $\boldsymbol{c} \in \mathcal{C}_{\Sigma R}[\boldsymbol{n}, k]$*.*
- ***Objective:*** *Determine the unique codeword* $\boldsymbol{c} \in \mathcal{C}_{\Sigma R}[\boldsymbol{n}, k]$ *such that*

$$\mathrm{wt}_{\Sigma R}(\boldsymbol{y} - \boldsymbol{c}) \leq \tau.$$

In this case, the decoder is guaranteed to find exactly one solution as long as the error weight is within the unique decoding radius. Efficient polynomial-time decoders are available for several well-known algebraic codes, such as Reed–Solomon codes, BCH codes, and Goppa codes in the Hamming metric [MS77], as well as Gabidulin codes in the rank metric [Gab85], and LRS codes in the sum-rank metric [Mar18], provided that the code structure is known.

However, in cryptosystems such as McEliece-like cryptosystems, where the code structure is intentionally obfuscated to increase security, decoding becomes much more challenging for unauthorized parties. This further highlights the importance of efficient decoders, particularly in cases where the error weight stays within the unique decoding radius.

6.1.4 Channel Model

In Problem 6.2, Problem 6.3 and Problem 6.4, the error vector $\boldsymbol{e}$ is drawn uniformly at random from $\mathcal{E}_w$, the set of all vectors in $\mathbb{F}_{q^m}^n$ with sum-rank weight w. This is done by first determining a rank profile $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}$ according to the marginal distribution of the rank profile of the error given by

$$\Pr[\boldsymbol{w}] = \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m, \eta, w_i\right). \tag{6.1}$$

Then, for each $i \in \{1, \ldots, \ell\}$, an element $\boldsymbol{e}^{(i)} \in \mathbb{F}_{q^m}^{\eta_i}$ of rank weight w_i is drawn independently and uniformly at random from the set of all elements in $\mathbb{F}_{q^m}^{\eta_i}$ of rank

weight w_i, as described in [PRR22]. The resulting error vector $\boldsymbol{e}$ satisfies $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}) = w$ and is uniformly distributed in $\mathcal{E}_w$. The received word $\boldsymbol{y}$ is then considered to be of the form

$$\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e},$$

with $\boldsymbol{c} \in \mathcal{C}_{\Sigma R}[\boldsymbol{n}, k] \subseteq \mathbb{F}_{q^m}^n$.

6.2 Ordered Rank Profiles

We now define the set that contains only the ordered rank profiles, as introduced in Definition 2.9. This concept helps to simplify the complexity of computing certain expressions regarding the sum-rank metric by exploiting symmetry properties that are the same for permuted versions of the rank profiles.

Definition 6.1 (Ordered Rank Profiles)**.** *Let w, ℓ, and μ be non-negative integers with $w \leq \ell\mu$. We define the set*

$$\mathcal{J}_{w,\ell,\mu} \stackrel{\text{def}}{=} \{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu} : w_1 \geq w_2 \geq \cdots \geq w_\ell\},$$

which contains the ordered rank profiles of a vector with ℓ blocks and a sum-rank weight of w.[1]

Note that every element $\boldsymbol{w} \in \mathcal{J}_{w,\ell,\mu}$ is also an element in $\mathcal{T}_{w,\ell,\mu}$, but not vice versa.

Next, for a given $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}$, we denote by $\mathfrak{S}_{\ell,\mu}(\boldsymbol{w})$ the set of all possible permutations of $\boldsymbol{w}$. Hence, each element σ in $\mathfrak{S}_{\ell,\mu}(\boldsymbol{w})$ is a permutation operator, i.e., $\sigma : \boldsymbol{w} \mapsto \sigma(\boldsymbol{w})$, where $\sigma(\boldsymbol{w})$ is a permutation of the entries of $\boldsymbol{w}$.

Additionally, let $\tilde{\boldsymbol{w}} = [\tilde{w}_1, \ldots, \tilde{w}_\ell]$ be the vector obtained by sorting the entries of $\boldsymbol{w}$ in decreasing order. We denote this operation by $\mathrm{sort}(\boldsymbol{w})$, so that $\tilde{\boldsymbol{w}} = \mathrm{sort}(\boldsymbol{w})$. In other words, $\mathrm{sort}(\boldsymbol{w})$ represents the permutation of the entries of $\boldsymbol{w}$ that yields the sorted vector $\tilde{\boldsymbol{w}}$, such that

$$\tilde{w}_1 \geq \tilde{w}_2 \geq \cdots \geq \tilde{w}_\ell.$$

This leads to the following two relations between the sets $\mathcal{J}_{w,\ell,\mu}$ and $\mathcal{T}_{w,\ell,\mu}$

$$\begin{aligned}
\mathcal{J}_{w,\ell,\mu} &= \{\mathrm{sort}(\boldsymbol{w}) : \forall \boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}\}, \\
\mathcal{T}_{w,\ell,\mu} &= \bigcup_{\boldsymbol{w} \in \mathcal{J}_{w,\ell,\mu}} \{\sigma(\boldsymbol{w}) : \forall \sigma \in \mathfrak{S}_{\ell,\mu}(\boldsymbol{w})\}.
\end{aligned}$$

[1] In literature, this set is closely related to the concept of *integer partitions* of w, where an integer is expressed as the sum of non-negative integers. Without the restrictions on the maximal number of blocks ℓ and the maximum rank weight μ, this set would coincide with the set of integer partitions. However, with these restrictions, it forms a constrained version, limited in both length by ℓ and value by μ.

The number of possible permutations of $\boldsymbol{w}$, denoted by $|\mathfrak{S}_{\ell,\mu}(\boldsymbol{w})|$, is given by the multinomial coefficient

$$|\mathfrak{S}_{\ell,\mu}(\boldsymbol{w})| = \binom{\ell}{\lambda_0, \lambda_1, \ldots, \lambda_\mu} = \frac{\ell!}{\lambda_0!\lambda_1! \cdots \lambda_\mu!},$$

where λ_i is the number of occurrences of the integer i in $\boldsymbol{w}$ for all $i \in \{0, \ldots, \mu\}$. Therefore, by considering all permutations of all elements in $\mathcal{J}_{w,\ell,\mu}$, we obtain the set $\mathcal{T}_{w,\ell,\mu}$.

For a given (non-ordered) rank profile $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}$ we also define the restricted set of rank profiles

$$\mathcal{T}_{w,\ell,\mu}^{\geq \boldsymbol{w}} \stackrel{\text{def}}{=} \{\boldsymbol{w}' \in \mathcal{T}_{w,\ell,\mu} : w'_1 \geq w_1, \ldots, w'_\ell \geq w_\ell\}.$$

6.3 Generic Decoding in the Sum-Rank Metric

In this section, we summarize the generic decoding algorithm for the sum-rank metric introduced by Puchinger et al. [PRR20; PRR22]. Their analysis focuses on solving a special instance of Problem 6.1, where the syndrome is chosen such that at least one solution exists. This scenario applies to both Problem 6.2 and Problem 6.4. In their derivation of the lower bound, they assume that at most one solution exists within the decoding radius, with no alternative solutions possible. As a result, the lower bound applies only to Problem 6.4.

The decoding algorithm proceeds by guessing possible error supports according to a probability distribution, which is a key design criterion that can be optimized to maximize the decoder's success probability. In each iteration of the decoding loop, a new support is guessed, and the decoder attempts to correct the error based on that support. If the guess is incorrect, the loop continues with another guess. The success probability for each iteration depends on the support-guessing distribution. The worst-case complexity is determined by evaluating all possible error rank profiles, with the worst-case profile, denoted by $\boldsymbol{w}_{\text{wc}}$, representing the maximum number of operations needed for the decoder to succeed.

Remark 6.1. *In [PRR22, Remark 17], it was shown that an error $\boldsymbol{e}$ with row support $\boldsymbol{\mathcal{E}}_R$ and column support $\boldsymbol{\mathcal{E}}_C$ can be uniquely recovered if either a row super-support $\boldsymbol{\mathcal{F}}_R \supseteq \boldsymbol{\mathcal{E}}_R$ or a column super-support $\boldsymbol{\mathcal{F}}_C \supseteq \boldsymbol{\mathcal{E}}_C$ is found, both with sum-rank weight v, such that $w \leq v < d_{\min}$. The sum-rank weight v cannot exceed*

$$v_{\max} \stackrel{\text{def}}{=} \min\left\{n-k, \left\lfloor \frac{m}{\eta}(n-k) \right\rfloor\right\}. \tag{6.2}$$

Decoding beyond the unique erasure decoding radius is possible up to $v_{\max}$. This situation is analogous to the classical ISD algorithm in the Hamming metric (i.e. $\eta = 1$

and $n = \ell$), where a support of size $v_{\max} = n - k$ is selected. Successful decoding occurs if the corresponding submatrix of the parity-check matrix formed by these columns is of full rank, ensuring a unique solution to the system of linear equations. In the Hamming metric, the probability that a submatrix is full rank is generally assumed to be close to 1, especially for large field sizes q^m. However, in Theorem 5.6, we derived bounds on this probability for randomly chosen parity-check matrices. As shown in Section 5.5.1, this probability can be significantly lower for the sum-rank metric, depending on the parameter choices. Consequently, the probability of finding a valid decoding support under these conditions can be much lower than expected. In this chapter, we similarly assume this probability to be close to 1 for simplicity in our analysis, but for certain parameter sets, it may be advisable to verify this assumption in practice.

Algorithm 8 describes the decoding process, which selects the appropriate type of (row or column) support. When m is smaller than η, the algorithm selects the row support; otherwise, it chooses the column support. For the sake of simplicity, we will henceforth refer to the selected support type as simply "support" throughout the remainder of this chapter, with the understanding that the decoding algorithm makes this choice based on the given parameters.

Algorithm 8: Generic Sum-Rank Decoder [PRR22]

Input : Parameters: q, m, n, k, ℓ, w and v with $w \leq v \leq v_{\max}$
Received vector $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$
Parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ of an $\mathbb{F}_{q^m}$-linear sum-rank metric code $\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k]$

Output : Vector $\boldsymbol{c}' \in \mathcal{C}_{\Sigma R}[\boldsymbol{n}, k]$ such that $\mathrm{wt}_{\Sigma R}(\boldsymbol{y} - \boldsymbol{c}') = w$

$\boldsymbol{e}' \leftarrow \boldsymbol{0} \in \mathbb{F}_{q^m}^n$
$\eta \leftarrow n/\ell$
$\mu \leftarrow \min\{m, \eta\}$
while $\boldsymbol{H}(\boldsymbol{r} - \boldsymbol{e}')^\top \neq \boldsymbol{0}$ **or** $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}') \neq w$ **do**
  $\boldsymbol{\mathcal{F}} \leftarrow$ Draw random support $\boldsymbol{\mathcal{F}} \subseteq \mathbb{F}_q^\mu \times \cdots \times \mathbb{F}_q^\mu$ of sum dimension v
  if $\eta < m$ **then**
    $\boldsymbol{e}' \leftarrow$ Column erasure decoding: $\boldsymbol{\mathcal{F}}$, $\boldsymbol{H}$, $\boldsymbol{y}$ `/* cf. [PRR22, Theorem 13] */`
  else
    $\boldsymbol{e}' \leftarrow$ Row erasure decoding: $\boldsymbol{\mathcal{F}}$, $\boldsymbol{H}$, $\boldsymbol{y}$ `/* cf. [PRR22, Theorem 14] */`
return $\boldsymbol{y} - \boldsymbol{e}'$

6.3.1 Improved Simple Bound on the Worst-Case Success Probability

Recall that we denoted the worst-case rank profile as $\boldsymbol{w}_{\mathrm{wc}}$. In [PRR22, Theorem 16], the authors derive lower and upper bounds on the expected run time W_{gen} of Algorithm 8 to decode an error pattern with rank profile $\boldsymbol{w}_{\mathrm{wc}}$.

Given an error rank profile $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}$ and a super-support rank profile $\boldsymbol{v} \in \mathcal{T}_{v,\ell,\mu}$, we compute the probability $\varphi_{q,\mu}(\boldsymbol{v}, \boldsymbol{w})$ that the error support lies within the guessed super support as [PRR22]

$$\varphi_{q,\mu}(\boldsymbol{v}, \boldsymbol{w}) \stackrel{\text{def}}{=} \prod_{i=1}^{\ell} \mathsf{P}^{\subseteq}_{q,\mu}\left(\boldsymbol{w}_i, \boldsymbol{v}_i\right), \tag{6.3}$$

where $\mathsf{P}^{\subseteq}_{q,\mu}\left(\boldsymbol{w}_i, \boldsymbol{v}_i\right)$ is the probability as in (2.8).

Using this probability, we define $\varphi^{(\max)}_{q,\mu,v}(\boldsymbol{w})$ as the maximum probability over all super-support rank profiles $\boldsymbol{v} \in \mathcal{T}_{v,\ell,\mu}$

$$\varphi^{(\max)}_{q,\mu,v}(\boldsymbol{w}) \stackrel{\text{def}}{=} \max_{\boldsymbol{v} \in \mathcal{T}_{v,\ell,\mu}} \varphi_{q,\mu}(\boldsymbol{v}, \boldsymbol{w}).$$

Further let $\mathrm{vcomp}_{\mu}(\boldsymbol{w}, v)$ be a function $\mathrm{vcomp}_{\mu} : \mathcal{T}_{w,\ell,\mu} \times \mathbb{Z}_{\geq 0} \to \mathcal{T}_{v,\ell,\mu}$ that for a given integer $v \in \mathbb{Z}_{\geq 0}$ returns a rank profile such that

$$\varphi^{(\max)}_{q,\mu,v}(\boldsymbol{w}) = \varphi_{q,\mu}(\mathrm{vcomp}_{\mu}(\boldsymbol{w}, v), \boldsymbol{w}).$$

The function $\mathrm{vcomp}_{\mu}(\boldsymbol{w}, v)$ can be implemented efficiently, see [PRR22].

With these definitions in place, Puchinger et al. [PRR22, Theorem 16] provide the following bounds on the expected runtime W_{gen} of Algorithm 8 for Problem 6.4

$$W^{(\mathrm{LB})}_{\mathrm{gen,wc}} = W^{(\mathrm{iter})}_{\mathrm{gen}} |\mathcal{T}_{w,\ell,\mu}|^{-1} Q_{w,\ell,\mu}, \tag{6.4}$$

$$W^{(\mathrm{UB})}_{\mathrm{gen,wc}} = W^{(\mathrm{iter})}_{\mathrm{gen}} Q_{w,\ell,\mu}, \tag{6.5}$$

$$\widetilde{W}^{(\mathrm{UB})}_{\mathrm{gen,wc}} = W^{(\mathrm{iter})}_{\mathrm{gen}} \binom{\ell + w - 1}{\ell - 1} \gamma_q^{\ell} q^{w(\mu - \frac{v}{\ell})}, \tag{6.6}$$

with

$$Q_{w,\ell,\mu} \stackrel{\text{def}}{=} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi^{(\max)}_{q,\mu,v}(\boldsymbol{w})^{-1}, \tag{6.7}$$

where $W^{(\mathrm{iter})}_{\mathrm{gen}}$ represents the complexity of one iteration of the generic decoding algorithm. To be more precise, $W^{(\mathrm{iter})}_{\mathrm{gen}}$ is the sum of the complexities of two main components: drawing an error super support and performing the row/column erasure decoding, as outlined in Algorithm 8. The complexity of drawing an error super support is in the order of $\widetilde{O}(n^3 m^2 \log_2(q))$ bit operations. Meanwhile, the row/column-

erasure decoding involves $O((n-k)^3m^3)$ operations over $\mathbb{F}_q$. While these complexities are asymptotic approximations and neglect constant factors, they provide useful estimates for large input sizes. For finite lengths, the exact complexities depend on the specific implementation details of the underlying algorithms. Therefore, for the purpose of plotting and practical considerations, we approximate $W_{\text{gen}}^{(\text{iter})} \approx n^3m^3$, combining the dominant terms of both components.

The upper bound $W_{\text{gen,wc}}^{(\text{UB})}$ requires computing the term $Q_{w,\ell,\mu}$, a process that, while feasible in polynomial time, relies on algorithms that are challenging to implement. To simplify this, Puchinger et al. derived the closed-form bound $\widetilde{W}_{\text{gen,wc}}^{(\text{UB})}$ on the expected runtime of Algorithm 8. While convenient, this bound can be rather loose when the parameters approach the Hamming metric, i.e., as $\ell \to n$ for a fixed η and/or $\eta \to 1$ (see Figure 6.2). To address this, we introduce a new, tighter upper bound in Theorem 6.1.

Theorem 6.1. *Let $\boldsymbol{c}$ be a codeword of a sum-rank-metric code $\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k]$ with minimum sum-rank distance $d_{\min}$. Additionally, let $\boldsymbol{e}$ be an error of sum-rank weight $w < d_{\min}$ with a rank profile corresponding to the worst-case rank profile $\boldsymbol{w}_{\text{wc}}$. Then Algorithm 8 in the context of Problem 6.1 returns a solution w.r.t. an error $\boldsymbol{e}' \in \mathbb{F}_{q^m}^n$ with the weight w. Each iteration of Algorithm 8 has complexity $W_{\text{gen}}^{(\text{iter})}$. The overall expected worst-case runtime, also referred to as the complexity $W_{\text{gen,wc}}$ of Algorithm 8 is upper bounded by*

$$W_{\text{gen,wc}} \leq \widetilde{W}_{\text{gen,wc}}^{(\text{UB,improved})}, \tag{6.8}$$

with

$$\widetilde{W}_{\text{gen,wc}}^{(\text{UB,improved})} \stackrel{\text{def}}{=} W_{\text{gen}}^{(\text{iter})} \binom{\ell+w-1}{\ell-1} q^{w(\mu-\frac{v}{\ell})} \cdot \min\left(\gamma_q^{\ell}, \left(\frac{1-q^{-\mu}}{1-q^{-1}}\right)^w\right).$$

Proof. To prove the theorem, we will show that

$$W_{\text{gen,wc}} \leq W_{\text{gen}}^{(\text{iter})} \binom{\ell+w-1}{\ell-1} q^{w(\mu-\frac{v}{\ell})} \left(\frac{1-q^{-\mu}}{1-q^{-1}}\right)^w.$$

Starting from the bound in (6.5), it suffices to show that

$$Q_{w,\ell,\mu} \leq \binom{\ell+w-1}{\ell-1} q^{w(\mu-\frac{v}{\ell})} \left(\frac{1-q^{-\mu}}{1-q^{-1}}\right)^w.$$

By the definition of $Q_{w,\ell,\mu}$ in (6.7), we can bound $Q_{w,\ell,\mu}$ as (cf. [PRR22])

$$\begin{aligned} Q_{w,\ell,\mu} &\leq |\mathcal{T}_{w,\ell,\mu}| \cdot \max_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v}^{(\max)}(\boldsymbol{w})^{-1} \\ &\leq \binom{\ell+w-1}{\ell-1} \cdot \max_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v}^{(\max)}(\boldsymbol{w})^{-1}, \end{aligned} \tag{6.9}$$

where

$$\max_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v}^{(\max)}(\boldsymbol{w})^{-1} = \max_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \left\{ \varphi_{q,\mu}(\boldsymbol{v}',\boldsymbol{w})^{-1} : \boldsymbol{v}' = \mathrm{vcomp}_{\mu}(\boldsymbol{w},v) \right\}.$$

Next, we can bound $\varphi_{q,\mu}(\boldsymbol{v}',\boldsymbol{w})^{-1}$ as

$$\varphi_{q,\mu}(\boldsymbol{v}',\boldsymbol{w})^{-1} = \prod_{i=1}^{w} \frac{\begin{bmatrix}\mu\\ w_i\end{bmatrix}_q}{\begin{bmatrix}v_i'\\ w_i\end{bmatrix}_q} \leq \left(\frac{1-q^{-\mu}}{1-q^{-1}}\right)^{w} \cdot \prod_{i=1}^{w} q^{w_i(\mu-v_i')}, \tag{6.10}$$

where the inequality follows from

$$\frac{\begin{bmatrix}a\\ b\end{bmatrix}_q}{\begin{bmatrix}c\\ b\end{bmatrix}_q} = \frac{q^{b(a-b)}}{q^{b(c-b)}} \cdot \prod_{i=1}^{w} \frac{(1-q^{-a})(1-q^{-a+1})\cdots(1-q^{-a+b-1})}{(1-q^{-c})(1-q^{-c+1})\cdots(1-q^{-c+b-1})} \leq q^{b(a-c)} \cdot \frac{1-q^{-a}}{1-q^{-1}}.$$

Substituting (6.10) into (6.9) yields

$$Q_{w,\ell,\mu} \leq \binom{\ell+w-1}{\ell-1} \cdot \left(\frac{1-q^{-\mu}}{1-q^{-1}}\right)^{w} \cdot \max_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \left\{ q^{\sum_{i=1}^{\ell} w_i(\mu-v_i')} : \boldsymbol{v}' = \mathrm{vcomp}_{\mu}(\boldsymbol{w},v) \right\}.$$

The remainder of the proof follows from the proof in [PRR22, Proposition 21], which leads to the desired result

$$W_{\mathrm{gen,wc}} \leq \widetilde{W}_{\mathrm{gen,wc}}^{(\mathrm{UB,improved})}.$$

□

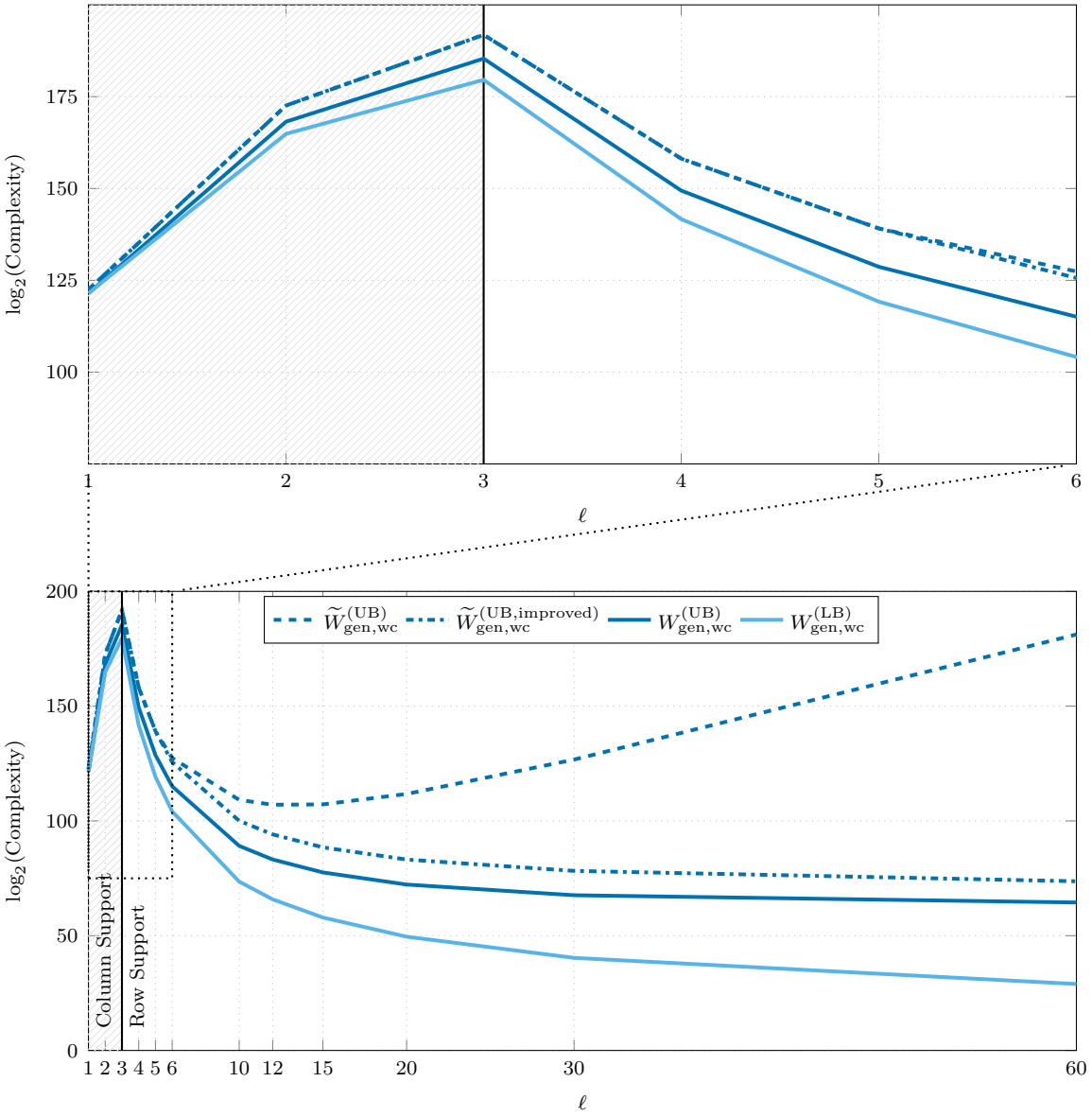


Figure 6.2: Illustration of the improved upper bound on the complexity of Algorithm 8 compared to the other bounds (6.4),(6.5) and (6.6) for parameters: $q = 2$, $m = 20$, $n = 60$, $k = 30$, $w = 9$, $s = 10$. At $\ell = 3$, the algorithm transitions from guessing the column support to guessing the row support.

Figure 6.2 illustrates the existing bounds from Puchinger et al. [PRR22] alongside the new improved bound $\widetilde{W}_{\text{gen,wc}}^{(\text{UB,improved})}$ given by (6.8). The figure shows the complexity as a function of the number of blocks ℓ for a fixed code length n, where the code length is defined as $n = \ell\eta$. This representation allows for the analysis of how the bounds and the improved bound behave as the number of blocks varies while keeping the code length and rate constant.

The illustrated bounds also include the lower bound $W_{\text{gen,wc}}^{(\text{LB})}$, the upper bound $W_{\text{gen,wc}}^{(\text{UB})}$, and the simplified upper bound $\widetilde{W}_{\text{gen,wc}}^{(\text{UB})}$, as defined in (6.4), (6.5), and (6.6), respectively. For a fair comparison, we use the same parameters as those presented in one of the figures from the original paper. The improved bound $\widetilde{W}_{\text{gen,wc}}^{(\text{UB,improved})}$ is significantly tighter and closer to the upper bound $W_{\text{gen,wc}}^{(\text{UB})}$, particularly in the region near the Hamming metric, which corresponds to cases where $\ell \to n$ and $\eta \to 1$. This suggests that the new simplified bound provides a better approximation of the algorithm's complexity compared to the previous simplified upper bound $\widetilde{W}_{\text{gen,wc}}^{(\text{UB})}$, especially when the sum-rank metric closely resembles the Hamming metric.

6.3.2 Success Probability Analysis for the Average Case

We now consider the channel model described in Section 6.1.4 and begin by deriving the success probability for the case of unique decoding, where exactly one solution exists. Additionally, we derive an upper bound on the success probability for decoding beyond the unique decoding radius, using RCU arguments. This upper bound accounts for alternative solutions that the decoder in Algorithm 8 may return in this scenario.

In this analysis, we assume that the support drawing distribution is known. In the subsequent section, we will use the probabilities derived here to formalize an optimization problem with respect to the support drawing distribution.

In Line 5 of Algorithm 8, we need to draw a suitable super support

$$\boldsymbol{\mathcal{F}} = \boldsymbol{\mathcal{F}}_1 \times \cdots \times \boldsymbol{\mathcal{F}}_\ell,$$

where $\boldsymbol{\mathcal{F}} \subseteq \mathbb{F}_q^\mu \times \cdots \times \mathbb{F}_q^\mu$, each $\boldsymbol{\mathcal{F}}_i$ has dimension v_i for $i \in \{1, \ldots, \ell\}$, and $\sum_{i=1}^{\ell} v_i = v$. The distribution from which these super supports are drawn is a critical design parameter of the algorithm and requires careful optimization to maximize the algorithm's performance.

To draw the super support $\boldsymbol{\mathcal{F}}$, we first draw a rank profile $\boldsymbol{v} = [v_1, v_2, \ldots, v_\ell] \in \mathcal{T}_{v,\ell,\mu}$. Let S be a discrete random variable over $\mathcal{T}_{v,\ell,\mu}$, and denote the probability distribution of S as $\alpha_{\boldsymbol{v}}$, i.e.,

$$\alpha_{\boldsymbol{v}} \stackrel{\text{def}}{=} \Pr[S = \boldsymbol{v}].$$

Moreover, let $\boldsymbol{\alpha}$ denote the probability vector for S, such that $\boldsymbol{\alpha} = [\alpha_{\boldsymbol{v}_1}, \ldots, \alpha_{\boldsymbol{v}_{|\mathcal{T}_{v,\ell,\mu}|}}]$, where $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{|\mathcal{T}_{v,\ell,\mu}|} \in \mathcal{T}_{v,\ell,\mu}$ and $\boldsymbol{\alpha} \in \mathcal{D}(\mathcal{T}_{v,\ell,\mu})$.

After drawing the rank profile $\boldsymbol{v}$ according to the distribution $\alpha_{\boldsymbol{v}}$, the next step

is to construct the super support $\boldsymbol{\mathcal{F}}$. We draw $\boldsymbol{\mathcal{F}}$ uniformly from the set $\Xi_{q,\mu}(\boldsymbol{v})$, which contains all valid super supports for the given rank profile $\boldsymbol{v}$. Each block $\boldsymbol{\mathcal{F}}_i$ is drawn independently from the set of subspaces of $\mathbb{F}_q^\mu$ with dimension v_i. By controlling the distribution $\alpha_{\boldsymbol{v}}$, we influence the distribution of the super support $\boldsymbol{\mathcal{F}}$ and aim to minimize the expected complexity of the decoding process.

The following theorem gives the success probability for uniquely decoding a solution using a single iteration of Algorithm 8 under the average-case setting.

Theorem 6.2. *Let $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$ be a linear sum-rank-metric code with length n and dimension k. Let $\boldsymbol{c} \in \mathcal{C}$ be a codeword and consider a channel model as described in Section 6.1.4, where the error $\boldsymbol{e}$ is drawn uniformly at random from $\mathcal{E}_{q,\eta,m,\ell}(w)$, as defined in* (2.24)*. Let $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}) = w$, and assume that $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$. Let $v_{\max}$ denote the maximum sum-rank weight of the guessed super support, as defined in* (6.2)*. Define the event $\mathcal{E}_{unique}$ as the event that Algorithm 8 outputs $\boldsymbol{c}$ in a single iteration for the scenario of Problem 6.2. The probability of $\mathcal{E}_{unique}$ is given by*

$$\Pr\left[\mathcal{E}_{unique}\right] = \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v'=w}^{v_{\max}} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\boldsymbol{w}) \cdot \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m, \eta, w_i\right), \tag{6.11}$$

where $\varphi_{q,\mu,v'}(\boldsymbol{w})$ is the average probability defined in (6.12)*.*

Proof. The average probability of decoding success can be expressed as a sum over all possible super space dimensions v' and error weight decompositions $\boldsymbol{w}$, weighted by the probability of each error weight decomposition

$$\Pr\left[\mathcal{E}_{\text{unique}}\right] = \sum_{v'=w}^{v_{\max}} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \Pr\left[\boldsymbol{w}\right] \cdot \varphi_{q,\mu,v'}(\boldsymbol{w}).$$

Substituting the expression for $\Pr\left[\boldsymbol{w}\right]$ from (6.1), we arrive at the expression given in (6.11)

$$\Pr\left[\mathcal{E}_{\text{unique}}\right] = \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v'=w}^{v_{\max}} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\boldsymbol{w}) \cdot \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m, \eta, w_i\right),$$

completing the proof. □

An upper bound on the probability of obtaining alternative solutions when using random linear sum-rank-metric codes is provided by the upcoming theorem. For the analysis, we define $\varphi_{q,\mu,v}(\boldsymbol{w})$ as the average probability over all super-support rank profiles

$$\varphi_{q,\mu,v}(\boldsymbol{w}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{v} \in \mathcal{T}_{v,\ell,\mu}} \alpha_{\boldsymbol{v}} \cdot \varphi_{q,\mu,v}(\boldsymbol{v}, \boldsymbol{w}), \tag{6.12}$$

where $\alpha_{\boldsymbol{v}}$ is the probability distribution over the super-support rank profiles.

Theorem 6.3 (Random Coding Union Bound)**.** *Let $\mathcal{C}$ be a random code of length n and cardinality $|\mathcal{C}| = q^{mk}$ over $\mathbb{F}_{q^m}$, where each codeword is drawn uniformly at random from the ambient space $\mathbb{F}_{q^m}^n$. Suppose that the received word $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$ is a noisy version of a codeword $\boldsymbol{c} \in \mathcal{C}$, corrupted by an error vector $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ of sum-rank weight w, i.e., $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$. Let v be an integer satisfying $w \leq v \leq v_{\max}$.*

The probability $\Pr[\mathcal{E}_{\mathrm{RCU}}]$ of one iteration of Algorithm 8 to output an alternative solution $\boldsymbol{c}'$ with $\boldsymbol{c}' \neq \boldsymbol{c}$ is upper bounded by

$$\Pr[\mathcal{E}_{\mathrm{RCU}}] \leq \mathrm{p}_{\mathrm{RCU}}^{(\mathrm{UB,gen})},$$

where

$$\mathrm{p}_{\mathrm{RCU}}^{(\mathrm{UB,gen})} \stackrel{\mathrm{def}}{=} q^{m(k-n)} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v}(\boldsymbol{w}) \prod_{i=1}^{\ell} \mathrm{NM}_q(m, \eta, w_i),$$

and $\varphi_{q,\mu,v}(\boldsymbol{w})$ is the average probability defined in (6.12)*.*

Proof. By assumption, each codeword in the codebook $\mathcal{C}$ is drawn uniformly at random over $\mathbb{F}_{q^m}^n$. Let $\boldsymbol{c}_j \in \mathcal{C}$ with $\boldsymbol{c}_j \neq \boldsymbol{c}$ be one such alternative codeword with $j \in \{1, \ldots, q^{mk} - 1\}$, and define $\mathcal{X}_j$ as the event that Algorithm 8 can decode this codeword. Then

$$\Pr[\mathcal{X}_j] = \sum_{\substack{\boldsymbol{e}' \in \mathbb{F}_{q^m}^n \\ \mathrm{wt}_{\Sigma R}(\boldsymbol{e}') = w}} \frac{1}{q^{mn}} \cdot \varphi_{q,\mu,v}(\psi(\boldsymbol{e}')).$$

Since $\varphi_{q,\mu,v}(\psi(\boldsymbol{e}'))$ only depends on the rank profile of $\boldsymbol{e}'$, we can change the sum to be over all rank profiles $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}$ and multiply by the number of error vectors that have the same rank profile

$$\Pr[\mathcal{X}_j] = \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \frac{1}{q^{mn}} \cdot \varphi_{q,\mu,v}(\boldsymbol{w}) \prod_{i=1}^{\ell} \mathrm{NM}_q(m, \eta, w_i).$$

The total probability of successful decoding is given by the union of the events

$$\mathcal{X}_1, \ldots, \mathcal{X}_{q^{mk}-1},$$

which can be upper bounded by

$$\Pr\left[\bigcup_{j=1}^{q^{mk}-1} \mathcal{X}_j\right] \leq \sum_{j=1}^{q^{mk}-1} \Pr[\mathcal{X}_j] \leq q^{mk} \Pr[\mathcal{X}_j] = \mathrm{p}_{\mathrm{RCU}}^{(\mathrm{UB,gen})}.$$

Substituting the expression for $\Pr[\mathcal{X}_j]$ yields the desired upper bound on the success probability. □

Combining Theorem 6.3 and Theorem 6.2, we can derive bounds on the success probability of Algorithm 8 for one iteration to return at least one solution. We state these bounds in the following lemma.

Theorem 6.4. *Let $\mathcal{C}$ be a random code of length n and size q^{mk} over $\mathbb{F}_{q^m}$, where each codeword is drawn uniformly at random from the ambient space $\mathbb{F}_{q^m}^n$. Suppose that the received word $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$ is a noisy version of a codeword $\boldsymbol{c} \in \mathcal{C}$, corrupted by an error vector $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ with $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}) = w$, i.e., $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$. The success probability of Algorithm 8 to output at least one solution satisfies*

$$\Pr[\textit{success}] \geq \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v'=w}^{v_{\max}} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\boldsymbol{w}) \cdot \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m, \eta, w_i\right),$$

and

$$\Pr[\textit{success}] \leq \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)} \right) \sum_{v'=w}^{v_{\max}} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\boldsymbol{w}) \cdot \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m, \eta, w_i\right).$$

Proof. First, we prove the lower bound on the success probability. Recall that we assume the received word $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$ is a noisy version of a codeword $\boldsymbol{c} \in \mathcal{C}$, corrupted by an error vector $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ with $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}) = w$, i.e., $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$. This implies that the codeword $\boldsymbol{c}$ is always within the decoding radius of the received word $\boldsymbol{y}$. Using the expression for $\Pr[\mathcal{E}_{\mathrm{unique}}]$ from Theorem 6.2, we have

$$\Pr[\text{success}] \geq \Pr[\mathcal{E}_{\mathrm{unique}}] = \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v'=w}^{v_{\max}} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\boldsymbol{w}) \cdot \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m, \eta, w_i\right).$$

Next, we prove the upper bound on the success probability. Using union bound arguments and the expressions for $\Pr[\mathcal{E}_{\mathrm{RCU}}]$ and $\Pr[\mathcal{E}_{\mathrm{unique}}]$ from Theorem 6.3 and Theorem 6.2, respectively, we obtain

$$\begin{aligned}
\Pr[\text{success}] &\leq \Pr[\mathcal{E}_{\mathrm{unique}}] + \Pr[\mathcal{E}_{\mathrm{RCU}}] \\
&= \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v'=w}^{v_{\max}} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\boldsymbol{w}) \cdot \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m, \eta, w_i\right) \\
&\quad + q^{m(k-n)} \sum_{v'=w}^{v_{\max}} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\boldsymbol{w}) \cdot \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m, \eta, w_i\right) \\
&= \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)} \right) \sum_{v'=w}^{v_{\max}} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\boldsymbol{w}) \cdot \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m, \eta, w_i\right),
\end{aligned}$$

which concludes the lemma. □

From Theorem 6.4 we get that to find an optimal distribution $\alpha_{\boldsymbol{v}}$ to draw $\boldsymbol{v}$ from $\mathcal{T}_{v,\ell,\mu}$ we need to maximize the term

$$\max_{\boldsymbol{\alpha}\in\mathcal{D}(\mathcal{T}_{v,\ell,\mu})} \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \sum_{\boldsymbol{v}\in\mathcal{T}_{v,\ell,\mu}} \alpha_{\boldsymbol{v}} \cdot \varphi_{q,\mu}(\boldsymbol{v},\boldsymbol{w}) \cdot \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m,\eta,w_i\right),$$

where $\mathcal{D}(\mathcal{T}_{v,\ell,\mu})$ is the set of all valid PMFs over $\mathcal{T}_{v,\ell,\mu}$ as defined in (2.6).

6.3.3 Optimizing the Support-Drawing Distribution via Linear Programming

The process of drawing a super support from a known distribution can be further broken down. Instead of drawing a rank profile $\boldsymbol{v}\in\mathcal{T}_{v,\ell,\mu}$ according to $\alpha_{\boldsymbol{v}}$, we can draw an ordered rank profile $\boldsymbol{v}'\in\mathcal{J}_{v,\ell,\mu}$ according to a distribution $\tilde{\alpha}_{\boldsymbol{v}'}$, where $\boldsymbol{v}'=\mathrm{sort}(\boldsymbol{v})$ is obtained by sorting the elements of $\boldsymbol{v}$ in non-increasing order. This simplification is possible due to symmetry, as the probability of drawing a particular rank profile remains the same for all permutations of that profile.

After drawing the ordered rank profile $\boldsymbol{v}'$, we perform a uniformly random permutation to obtain the final rank profile $\boldsymbol{v}$. The relation between the two probability distributions is given by

$$\alpha_{\boldsymbol{v}} = \frac{\tilde{\alpha}_{\mathrm{sort}(\boldsymbol{v})}}{|\mathfrak{S}_{\ell,\mu}(\mathrm{sort}(\boldsymbol{v}))|} = \frac{\tilde{\alpha}_{\mathrm{sort}(\boldsymbol{v})}}{|\mathfrak{S}_{\ell,\mu}(\boldsymbol{v})|}. \tag{6.13}$$

By reducing the problem to optimizing the distribution $\tilde{\alpha}_{\boldsymbol{v}'}$ of ordered rank profiles, we have reduced the number of unknowns since we have $|\mathcal{J}_{v,\ell,\mu}|\leq|\mathcal{T}_{v,\ell,\mu}|$.

In summary, the process of drawing a suitable super support $\boldsymbol{\mathcal{F}}$ can be broken down into three steps:

1) Draw an ordered rank profile $\boldsymbol{v}'\in\mathcal{J}_{v,\ell,\mu}$ according to a distribution $\tilde{\alpha}_{\boldsymbol{v}'}$, which is the criterion we need to optimize, and then apply a uniformly random permutation to obtain the rank profile $\boldsymbol{v}$,

2) For each $i\in\{1,\ldots,\ell\}$, draw $\boldsymbol{\mathcal{F}}_i$ from the set of all spaces of dimension v_i, independently for all blocks,

3) Combine the individual blocks $\boldsymbol{\mathcal{F}}_i$ to form the overall super support

$$\boldsymbol{\mathcal{F}} = \boldsymbol{\mathcal{F}}_1 \times \cdots \times \boldsymbol{\mathcal{F}}_\ell.$$

By making use of (6.13) we can reduce the number of unknowns and instead maxi-

mize

$$\begin{aligned}\max_{\tilde{\boldsymbol{\alpha}}\in\mathcal{D}(\mathcal{J}_{v,\ell,\mu})} &\sum_{\boldsymbol{w}\in\mathcal{J}_{w,\ell,\mu}} \sum_{\boldsymbol{v}\in\mathcal{T}^{\geq\boldsymbol{w}}_{v,\ell,\mu}} \frac{|\mathfrak{S}_{\ell,\mu}(\boldsymbol{w})|}{|\mathfrak{S}_{\ell,\mu}(\boldsymbol{v})|}\tilde{\alpha}_{\mathrm{sort}(\boldsymbol{v})}\varphi_{q,\mu}(\boldsymbol{v},\boldsymbol{w})\prod_{i=1}^{\ell}\mathrm{NM}_q\left(m,\eta,w_i\right)\\ &= \max_{\tilde{\alpha}_{\boldsymbol{v}'}\in\mathcal{D}(\mathcal{J}_{v,\ell,\mu})} \sum_{\boldsymbol{v}'\in\mathcal{J}_{v,\ell,\mu}} \tilde{\alpha}_{\boldsymbol{v}'}\cdot f(\boldsymbol{v}'),\end{aligned} \tag{6.14}$$

where

$$f(\boldsymbol{v}') \coloneqq \sum_{\boldsymbol{w}\in\mathcal{J}_{w,\ell,\mu}} \frac{|\mathfrak{S}_{\ell,\mu}(\boldsymbol{w})|}{|\mathfrak{S}_{\ell,\mu}(\boldsymbol{v}')|}\left(\sum_{\boldsymbol{v}''\in\mathfrak{S}_{\ell,\mu}(\boldsymbol{v}')}\varphi_{q,\mu,v}(\boldsymbol{v}'',\boldsymbol{w})\right)\prod_{i=1}^{\ell}\mathrm{NM}_q\left(m,\eta,w_i\right).$$

This optimization problem can be solved via linear program (LP) methods, where the objective function is (6.14) with $|\mathcal{J}_{v,\ell,\mu}|$ unknowns. Although we have reduced the number of unknowns by restricting the optimization to the ordered set $\mathcal{J}_{v,\ell,\mu}$, it is important to note that the cardinality of this set, and consequently the number of unknowns, can still grow super-polynomially with the parameters v, ℓ, and μ. Furthermore, computing the coefficients of the constraints requires summing over the set $\mathcal{J}_{w,\ell,\mu}$, which can be computationally demanding due to its potentially large cardinality. Even if we successfully derive the optimal support-drawing distribution through this process, implementing an efficient algorithm to sample from this distribution poses another significant challenge. This limitation motivates the need for alternative approaches to simplify the optimization problem and develop more practical sampling algorithms.

6.3.4 Efficient Optimization of the Support-Drawing Distribution

In this section, we propose an efficient method to optimize the support-drawing distribution, addressing the computational challenges discussed earlier. By assuming independence between the sum-rank metric blocks, we greatly simplify the problem. Instead of drawing a complete rank profile vector $\boldsymbol{v} \in \mathcal{T}_{w,\ell,\mu}$ with a fixed total rank v, we independently draw the rank v_i for each of the ℓ blocks. As a result, the sum rank $v = \sum_{i=1}^{\ell} v_i$ becomes a random variable. To prevent it from becoming unbounded, we constrain its expected value, $\mathbb{E}[v]$, to match a predetermined relative sum-rank weight v/ℓ. This assumption reduces the complexity of the optimization problem by focusing on the distributions for individual blocks, and it enables efficient sampling from the optimized distribution, overcoming the practical limitations of the previous approach.

Although this heuristic approach may not always yield the optimal solution that accounts for the dependencies between the ranks of the guessed supports across different blocks, it still provides a good approximation. We demonstrate this numerically in Section 6.3.5 by comparing the performance of the heuristic solution with solutions

that consider these dependencies, for parameters where the more complex optimization method is feasible.

Let $\alpha_i^{(\mathrm{m})}$ denote the marginal probability of drawing a super support $\mathcal{F}_i$ with dimension v_i, where $0 \leq v_i \leq \mu$ for $i \in \{1, \ldots, \ell\}$, and let

$$\boldsymbol{\alpha}^{(\mathrm{m})} = [\alpha_0^{(\mathrm{m})}, \ldots, \alpha_\mu^{(\mathrm{m})}],$$

represent the marginal probability vector. Assuming that the dimension of each subspace $\mathcal{F}_i$ is drawn independently according to $\boldsymbol{\alpha}^{(\mathrm{m})}$, the probability of a given rank profile $\boldsymbol{v} = [v_1, \ldots, v_\ell] \in \mathcal{T}_{v,\ell,\mu}$ is given by

$$\alpha_{\boldsymbol{v}} = \prod_{i=1}^{\ell} \alpha_{v_i}^{(\mathrm{m})}. \tag{6.15}$$

We define the following two quantities

$$\tilde{B}_{q,m,\eta}(w, v, \ell) \stackrel{\text{def}}{=} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \sum_{\boldsymbol{v} \in \mathcal{T}_{v,\ell,\mu}} \alpha_{\boldsymbol{v}} \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m, \eta, w_i\right) \mathsf{P}_{q,\mu}^{\subseteq}\left(w_i, v_i\right) \tag{6.16}$$

and

$$B_{q,m,\eta}(w, v, \ell) \stackrel{\text{def}}{=} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \sum_{\boldsymbol{v} \in \mathcal{T}_{v,\ell,\mu}} \prod_{i=1}^{\ell} \alpha_{v_i}^{(\mathrm{m})} \mathrm{NM}_q\left(m, \eta, w_i\right) \mathsf{P}_{q,\mu}^{\subseteq}\left(w_i, v_i\right), \tag{6.17}$$

where (6.17) is a special case of (6.16) using our independence assumption. In Appendix B.1, we show that (6.17) is efficiently computable in polynomial time. Using the definition of $B_{q,m,\eta}(w, v, \ell)$ from (6.17) and the relaxation in (6.15), we can restate Theorem 6.2, Theorem 6.3, and Theorem 6.4 in the following corollaries, respectively.

Corollary 6.1. *The probability* $\Pr[\mathcal{E}_{\mathrm{RCU}}]$ *of having an alternative solution in Algorithm 8 for a random linear code of length* n *and cardinality* $M = q^{mk}$ *over* $\mathbb{F}_{q^m}$ *can be upper bounded as*

$$\Pr[\mathcal{E}_{\mathrm{RCU}}] \leq q^{m(k-n)} \sum_{v'=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w, v', \ell).$$

Corollary 6.2. *The probability* $\Pr[\mathcal{E}_{unique}]$ *that Algorithm 8 outputs a unique solution* $\boldsymbol{c}$ *for a random linear code of length* n *and cardinality* $M = q^{mk}$ *over* $\mathbb{F}_{q^m}$ *is given by*

$$\Pr[\mathcal{E}_{unique}] = \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v'=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w, v', \ell).$$

Corollary 6.3. *The success probability of Algorithm 8 to output at least one solution*

satisfies

$$\Pr[\textit{success}] \geq \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v'=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w,v',\ell),$$

and

$$\Pr[\textit{success}] \leq \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)} \right) \sum_{v'=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w,v',\ell).$$

From Corollary 6.3, under our independence assumption, the success probability is proportional to the term

$$\sum_{v'=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w,v',\ell) = \sum_{v'=w}^{v_{\max}} B_{q,m,\eta}(w,v',\ell), \tag{6.18}$$

which we aim to maximize over all possible $\boldsymbol{\alpha}^{(\mathrm{m})} \in \mathcal{D}(\{0,\dots,\mu\})$.

In the following, we further upper bound the expression in (6.18) and propose a method to maximize this upper bound, to obtain a valid solution for $\boldsymbol{\alpha}^{(\mathrm{m})}$.

$$\begin{aligned}
\sum_{v'=w}^{v_{\max}} B_{q,m,\eta}(w,v',\ell) &= \sum_{v'=t}^{v_{\max}} \sum_{\boldsymbol{w}\in\mathcal{T}_{t,\ell,\mu}} \sum_{\boldsymbol{v}\in\mathcal{T}_{v',\ell,\mu}} \prod_{i=1}^{\ell} \alpha_{v_i}^{(\mathrm{m})} \,\mathrm{NM}_q\left(m,\eta,w_i\right) \mathsf{P}_{q,\mu}^{\subseteq}\left(w_i,v_i\right) \\
&\leq \sum_{\boldsymbol{w}\in\{0,\dots,\mu\}^{\ell}} \sum_{\boldsymbol{v}\in\{0,\dots,\mu\}^{\ell}} \prod_{i=1}^{\ell} \alpha_{v_i}^{(\mathrm{m})} \,\mathrm{NM}_q\left(m,\eta,w_i\right) \mathsf{P}_{q,\mu}^{\subseteq}\left(w_i,v_i\right) \\
&= \left(\sum_{w'=0}^{\mu} \sum_{v'=0}^{\mu} \alpha_{v'}^{(\mathrm{m})} \,\mathrm{NM}_q\left(m,\eta,w'\right) \mathsf{P}_{q,\mu}^{\subseteq}\left(w',v'\right) \right)^{\ell}.
\end{aligned} \tag{6.19}$$

To maximize the right-hand side of (6.19), it suffices to maximize the expression

$$\sum_{w'=0}^{\mu} \sum_{v'=0}^{\mu} \alpha_{v'}^{(\mathrm{m})} \,\mathrm{NM}_q\left(m,\eta,w'\right) \mathsf{P}_{q,\mu}^{\subseteq}\left(w',v'\right). \tag{6.20}$$

This expression is closely related to the average probability that a randomly drawn super space $\mathcal{F}_i$ contains the error space $\mathcal{E}_i$ in a single block, averaged over all possible rank weights w'. The average single-block success probability is given as

$$\sum_{w'=0}^{\mu} \Pr[w'] \sum_{v'=0}^{\mu} \alpha_{v'}^{(\mathrm{m})} \mathsf{P}_{q,\mu}^{\subseteq}\left(w',v'\right), \tag{6.21}$$

where $\Pr[w']$ is the marginal probability of an error of rank weight w' occurring in a single block.

In the asymptotic setting for $\ell \to \infty$, where η and m are fixed, the assumption of independence between blocks becomes valid due to the law of large numbers and the concept of typical sequences from statistical mechanics. In this regime, the empirical

distribution of error weights in the blocks converges to the marginal distribution $\Pr[w']$, which can be approximated by the Boltzmann distribution (see [CJB24]) as

$$\Pr[w'] = \frac{\mathrm{NM}_q\left(m, \eta, w'\right) e^{-\lambda w'}}{\sum_{w''=0}^{\mu} \mathrm{NM}_q\left(m, \eta, w''\right) e^{-\lambda w''}}, \tag{6.22}$$

where λ is the unique solution to the weight constraint

$$\mathbb{E}[w'] = \sum_{w''=0}^{\mu} w'' \cdot \Pr[w''] = \frac{w}{\ell}.$$

By substituting (6.22) into (6.21), we obtain the single-block success probability under the asymptotic error weight distribution.

Maximizing the single-block success probability in (6.21) effectively maximizes the overall success probability in the asymptotic regime. Although (6.20) represents an upper bound on the success probability, this upper bound becomes tight as $\ell \to \infty$ due to the convergence properties established by the law of large numbers. Therefore, optimizing this upper bound is justified because it aligns with maximizing the actual success probability in the asymptotic setting.

This connection reveals that optimizing (6.20) to obtain an optimal marginal distribution $\alpha_{v'}^{(\mathrm{m})}$ for the guessed super-support dimensions is beneficial for maximizing (6.18).

To optimize (6.20), our approach focuses on the marginal distribution $\boldsymbol{\alpha}^{(\mathrm{m})}$, rather than directly optimizing $\boldsymbol{\alpha}$, aiming to approximate the optimal average rank profile for the super support. Since directly optimizing $\boldsymbol{\alpha}^{(\mathrm{m})}$ results in a distribution independent of the number of blocks ℓ, we impose the constraint

$$\alpha_i^{(\mathrm{m})} = \frac{x_i}{\ell},$$

where $x_i \in \mathbb{Z}_{\geq 0}$ represents the number of occurrences of rank i across the ℓ blocks.

We then maximize the objective in (6.19) using linear integer programming with appropriate constraints and non-negativity conditions. This method assumes independence of rank weights across the ℓ blocks, which holds asymptotically as $\ell \to \infty$ for fixed η and m.

By applying this method, we obtain a solution $\boldsymbol{x} = [x_0, \ldots, x_\mu]$, from which we construct the ordered rank profile $\hat{\boldsymbol{v}} \in \mathcal{J}_{w,\ell,\mu}$ as

$$\hat{\boldsymbol{v}} = [\underbrace{\mu, \ldots, \mu}_{x_\mu \text{ times}}, \underbrace{\mu-1, \ldots, \mu-1}_{x_{\mu-1} \text{ times}}, \ldots, \underbrace{1, \ldots, 1}_{x_1 \text{ times}}, \underbrace{0, \ldots, 0}_{x_0 \text{ times}}], \tag{6.23}$$

where each element $i \in \{0, \ldots, \mu\}$ appears exactly x_i times in the vector $\hat{\boldsymbol{v}}$. We then

have that

$$\tilde{\alpha}_{\boldsymbol{v}'}^{(\mathrm{heu})} \stackrel{\mathrm{def}}{=} \begin{cases} 1 & \text{if } \boldsymbol{v}' = \hat{\boldsymbol{v}}, \\ 0 & \text{otherwise.} \end{cases} \tag{6.24}$$

Using the relation in (6.13), we obtain the overall probability for $\boldsymbol{v}$, i.e., $\alpha_{\boldsymbol{v}}^{(\mathrm{heu})}$. For this specific PMF we can write (6.16) as

$$\tilde{B}_{q,m,\eta}(w,v,\ell) = \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m,\eta,w_i\right) \mathsf{P}_{q,\mu}^{\subseteq}\left(w_i,\hat{v}_i\right).$$

As shown in Appendix B.1, this expression can be efficiently computed in polynomial time since it is a special case of (6.17) for a fixed vector $\boldsymbol{v} = \hat{\boldsymbol{v}}$.

Thus, the bounds on the overall expected runtime provided in the following theorem, which are general for any support-guessing distribution, can be efficiently computed for our specific support-guessing distribution given by (6.23) and (6.24).

Theorem 6.5. *Under the same assumptions as in Corollary 6.3, the overall expected runtime $W_{\mathrm{gen,RCU}}$ of Algorithm 8 to output at least one solution is bounded by*

$$W_{\mathrm{gen,RCU}}^{(\mathrm{LB})} \leq W_{\mathrm{gen,RCU}} \leq W_{\mathrm{gen,RCU}}^{(\mathrm{UB})},$$

with

$$W_{\mathrm{gen,RCU}}^{(\mathrm{LB})} \stackrel{\mathrm{def}}{=} W_{\mathrm{erasure-dec}} \left(\left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)} \right) \sum_{v=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w,v,\ell) \right)^{-1}, \tag{6.25}$$

and

$$W_{\mathrm{gen,RCU}}^{(\mathrm{UB})} \stackrel{\mathrm{def}}{=} W_{\mathrm{erasure-dec}} \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w,v,\ell) \right)^{-1}, \tag{6.26}$$

where $W_{\mathrm{erasure-dec}}$ denotes the cost of one iteration of Algorithm 8 and

$$W_{\mathrm{erasure-dec}} \in O\Big((n-k)^3 m^3\Big),$$

operations over $\mathbb{F}_q$ and we neglect the complexity of drawing from $\alpha_{\boldsymbol{v}}$.

Proof. For the lower bound on the complexity, we consider the worst-case scenario for the complexity of each iteration, denoted by $W_{\mathrm{erasure-dec}}$. The expected number of iterations until success is the reciprocal of the success probability. Using the upper bound on the success probability from Corollary 6.3, the lower bound on the overall expected runtime satisfies

$$W_{\mathrm{gen,RCU}}^{(\mathrm{LB})} = W_{\mathrm{erasure-dec}} \left(\left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)} \right) \sum_{v=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w,v,\ell) \right)^{-1}.$$

For the upper bound on the complexity, we use the cost of one iteration $W_{\text{erasure-dec}}$, which is $O((n-k)^3m^3)$ operations over $\mathbb{F}_q$ according to [PRR22, Theorem 13 and Theorem 14]. Using the heuristic probability distribution for the guessing super support as in (6.23), we can neglect the complexity of drawing the rank profile of the guessing support. The expected number of iterations until success is the reciprocal of the success probability. Using the lower bound on the success probability from Corollary 6.3, the upper bound on the overall expected runtime satisfies

$$W_{\text{gen,RCU}}^{(\text{UB})} = W_{\text{erasure-dec}} \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v=w}^{v_{\max}} B_{q,m,\eta}(w, \boldsymbol{v}, \ell) \right)^{-1},$$

which concludes the proof. □

Note that the assumption made in Theorem 6.5, which neglects the complexity of drawing from $\alpha_{\boldsymbol{v}}$, is valid since, in our solution, we only need to permute the support-guessing rank profile $\hat{\boldsymbol{v}}$ uniformly at random.

6.3.5 Numerical Results

In this section, we compare the complexity analysis of using a support-drawing distribution derived from the method described in Section 6.3.4 for the average case against the worst-case bounds from [PRR22]. We evaluate the average complexity over all error patterns for a specific sum-rank weight w and plot the logarithmic complexity (base 2) versus the number of blocks ℓ, while keeping the code parameters and field size q^m constant. The length of each individual block η is adjusted as ℓ varies.

Table 6.1 summarizes the bounds considered in this analysis and their applicability across different decoding scenarios, highlighting the conditions for each. This overview provides context for the subsequent figures and complexity comparisons.

Table 6.1: Overview of the bounds and their applicable scenarios.

	Bound	Applies to	Conditions
Worst case	$W_{\text{gen,wc}}^{(\text{LB})}$	Problem 6.4	Exactly one solution exists for the worst-case rank profile channel
	$W_{\text{gen,wc}}^{(\text{UB})}$	Problem 6.1	At least one solution exists for the worst-case rank profile channel
Average case	$W_{\text{gen,RCU}}^{(\text{UB})}$	Problem 6.1	At least one solution exists; exact when exactly one solution exists
	$W_{\text{gen,RCU}}^{(\text{LB})}$	Problem 6.1	At least one solution exists; accounts for alternative solutions

Figure 6.3 shows the complexity for generic decoding beyond the unique decoding radius with parameters $q = 2$, $m = 20$, $n = 60$, $k = 30$, $w = 9$, and $v = 10$ while in Figure 6.4, we increase v to $v_{\max}$. We include the upper bound $W_{\text{gen,RCU}}^{(\text{UB})}$ (6.26) and the lower bound $W_{\text{gen,RCU}}^{(\text{LB})}$ (6.25) for the expected complexity of Algorithm 8. The lower bound reflects the effect of decoding beyond the unique decoding radius and accounts for alternative solutions.

The figures show that the effect of alternative solutions, indicated by the divergence between the upper and lower bounds, becomes more prominent near the rank metric (i.e., when ℓ approaches 1) for the chosen parameters (cf. Figure 6.3 and Figure 6.4) and matches the upper bound for $\ell \to n$. However, the latter behavior can vary depending on the parameters, as seen in Figure 6.5, which uses the parameters $q = 2$, $m = 6$, $n = 36$, $k = 22$, $w = 10$, and $v = 10$. In this case, a significant difference between the lower and upper bounds persists even when the number of blocks is rather large, i.e. $\ell \to n$.

With the increase of v to $v_{\max}$ in Figure 6.4, the upper bound $W_{\text{gen,wc}}^{(\text{UB})}$ and lower bound $W_{\text{gen,wc}}^{(\text{LB})}$ for the worst-case scenario become even looser. With the increase of v to $v_{\max}$ in Figure 6.4, the upper bound $W_{\text{gen,wc}}^{(\text{UB})}$ and lower bound $W_{\text{gen,wc}}^{(\text{LB})}$ for the worst-case scenario become increasingly loose, making them less effective predictors of the actual complexity. In contrast, our average-case analysis provides a more accurate estimate of the decoding complexity.

The Prange algorithm [Pra62], a classic support-guessing decoding algorithm for the Hamming metric (i.e., $\ell = n = 60$), serves as a useful comparison for this case. The

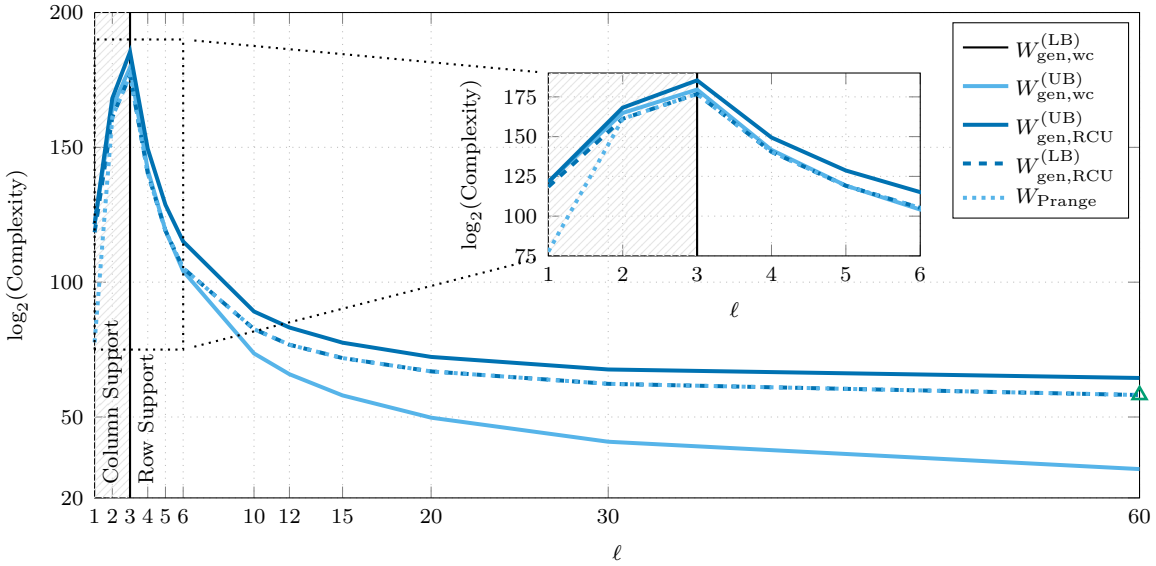


Figure 6.3: Complexity comparison for generic decoding using Algorithm 8 for codes with parameters: $q = 2$, $m = 20$, $n = 60$, $k = 30$, $w = 9$, and $v = 10$.

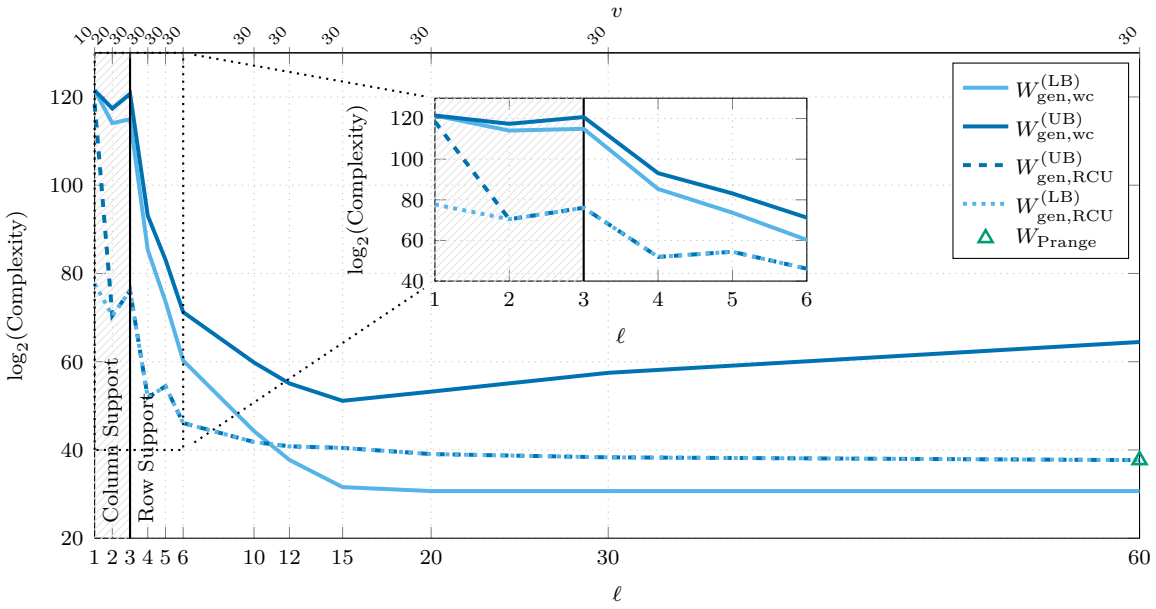


Figure 6.4: Complexity comparison for generic decoding using Algorithm 8 for codes with parameters: $q = 2$, $m = 20$, $n = 60$, $k = 30$, $w = 9$ and $v = v_{\max}$.

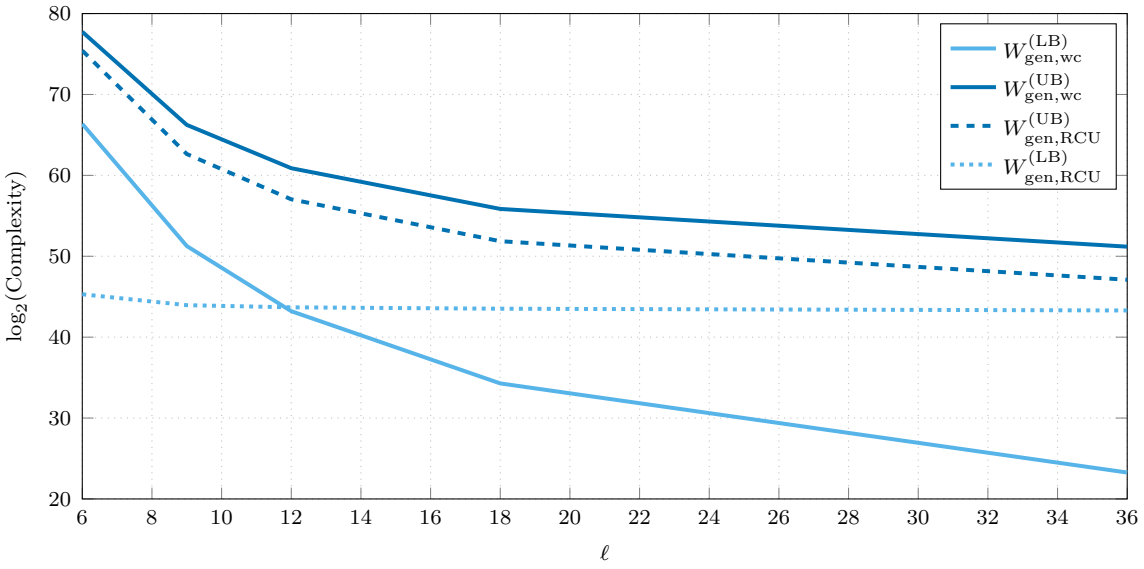


Figure 6.5: Complexity comparison for generic decoding using Algorithm 8 for codes with parameters: $q = 2$, $m = 6$, $n = 36$, $k = 22$, $w = 10$ and $v = 10$.

complexity of the Prange algorithm, denoted as W_{Prange} and given by [Pra62],

$$W_{\text{Prange}} = W^{(\text{iter})}_{\text{gen}} \frac{\binom{n}{w}}{\binom{v}{w}},$$

shows that our approach converges to Prange's performance for $\ell = n = 60$ (cf. Figure 6.3 and Figure 6.4). In this setting, our decoding algorithm not only matches the complexity of the Prange algorithm but is effectively the Prange algorithm itself.

Additionally, in Figure 6.4, for $\ell < 12$, the complexity of our solution falls below the lower bound of the worst-case scenario. This demonstrates that worst-case bounds may not always provide accurate estimates. For instance, when selecting parameters for cryptosystems based on sum-rank-metric codes, relying solely on worst-case bounds may lead to underestimating the actual complexity of practical attacks in certain regimes. The lower bound $W^{(\text{LB})}_{\text{gen,RCU}}$ provides a closer approximation to the actual complexity in practice than the worst-case bounds from [PRR22].

We performed extensive computations using LP to account for dependencies between the blocks, as described in Section 6.3.3, for the parameters used in Figure 6.3 and Figure 6.4. For these parameters this approach is feasible for ℓ up to 10 and we obtained the exact same support-guessing distribution corresponding to the rank profiles as our efficient solution. This is interesting, as our efficient method is expected to yield tighter results for larger block sizes, i.e., as $\ell \to n$. Thus, our findings indicate that for these parameters, the efficient approach remains effective even at lower ℓ.

6.4 Generic Decoding for Large Error Weights

In the previous section, we focused on decoding for low error weights, specifically when the error weight w satisfies $w \leq d_{\min}-1$, as erasure decoding is not possible beyond this threshold, according to [PRR22, Theorem 13 and Theorem 14]. We explored unique decoding for Problem 6.4 up to $w \leq d_{\min} - 1$ and going beyond unique decoding is possible for $w \leq \min\{n-k, \frac{m}{\eta}(n-k)\}$ (see [PRR22]).

In this section, we introduce a generic decoding algorithm (see Algorithm 9) that aims to solve Problem 6.1. The proposed algorithm generalizes the modified Prange algorithm for the Hamming metric, as detailed in [DST19]. Our analysis of the algorithm focuses on the asymptotic case $\ell \to \infty$ and its average performance. The following proposition establishes the range of relative weights for which solutions to Problem 6.1 can be found efficiently using Algorithm 9.

Algorithm 9: Prange-like Decoding Algorithm in the Sum-Rank Metric

Input : $\boldsymbol{H} \in \mathbb{F}_{q^m}^{\eta(\ell-\kappa)\times\eta\ell}$, $\boldsymbol{s} \in \mathbb{F}_{q^m}^{\eta(\ell-\kappa)}$ and $w \in \mathbb{Z}_{\geq 0}$
Output : $\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$ with $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}) = w$

$\mu \leftarrow \min\{m, \eta\}$
$\boldsymbol{e} \leftarrow \boldsymbol{0} \in \mathbb{F}_{q^m}^{\eta\ell}$
while $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e}) \neq w$ **do**
  $\boldsymbol{H}' \leftarrow \boldsymbol{0} \in \mathbb{F}_{q^m}^{\eta(\ell-\kappa)\times\eta\ell}$
  while $\mathrm{rk}_{\mathrm{q^m}}(\boldsymbol{H}'_{[1:\eta(\ell-\kappa)]}) \neq \eta(\ell-\kappa)$ **do**
    $\boldsymbol{P} \xleftarrow{\$}$ Set of $\ell \times \ell$ permutation matrices
    $\boldsymbol{P}' \leftarrow \boldsymbol{P} \otimes \boldsymbol{I}_\eta$
    $\boldsymbol{H}' \leftarrow \boldsymbol{H}\boldsymbol{P}'$
  $\boldsymbol{A} \leftarrow \boldsymbol{H}'_{[1:\eta(\ell-\kappa)]}$
  $\boldsymbol{B} \leftarrow \boldsymbol{H}'_{[\eta(\ell-\kappa)+1:\eta\ell]}$
  $w_1 \xleftarrow{\$} \{0, \ldots, \kappa\mu\}$
  $\boldsymbol{e}' \xleftarrow{\$} \{\boldsymbol{x} \in \mathbb{F}_{q^m}^{\kappa\ell} : \mathrm{wt}_{\Sigma R}^{[n_\kappa+1,\ldots,n_\ell]}(\boldsymbol{e}') = w_1\}$
  $\boldsymbol{e} \leftarrow ((\boldsymbol{s} - \boldsymbol{e}'\boldsymbol{B})\boldsymbol{A}^{-\top}, \boldsymbol{e}')(\boldsymbol{P}')^\top$
return $\boldsymbol{e}$

Proposition 6.1. *Consider a $\mathbb{F}_{q^m}$-linear sum-rank-metric code of length $n = \eta\ell$ and dimension $k = \eta\kappa$ with parity check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ and $R \overset{\text{def}}{=} \frac{k}{n} = \frac{\kappa}{\ell}$ where $\kappa \in \mathbb{Z}_{\geq 0}$ and $0 \leq \kappa \leq \ell$. Let the sum-rank weight be defined with respect to the length*

partition of constant block length, i.e., $\boldsymbol{n} = [n_1, \ldots, n_\ell] = [\eta, \ldots, \eta]$. *Define*

$$\bar{a} \stackrel{\text{def}}{=} \frac{\sum_{i=0}^{\mu} i \cdot \mathrm{NM}_q\left(m, \eta, i\right)}{q^{m\eta}},$$

as the average rank weight of a single block if drawn uniformly at random. Then, for the relative weight $w_{\text{rel}} \stackrel{\text{def}}{=} w/n$ *in the interval* $[w^-_{\text{easy}}, w^+_{\text{easy}}]$, *where*

$$w^-_{\text{easy}} \stackrel{\text{def}}{=} \frac{1-R}{\eta} \cdot \bar{a},$$
$$w^+_{\text{easy}} \stackrel{\text{def}}{=} \frac{1-R}{\eta} \cdot \bar{a} + \frac{R\mu}{\eta},$$

a solution to Problem 6.1 can be found in probabilistic polynomial time using the Prange-like Algorithm 9.

Proof. To address Problem 6.1, our goal is to find an error $\boldsymbol{e}$ with sum-rank weight w that satisfies the condition $\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$ for a given syndrome $\boldsymbol{s}$. The matrix $\boldsymbol{H}$ is a full-rank matrix and therefore contains an invertible submatrix $\boldsymbol{A} \in \mathbb{F}_{q^m}^{(n-k)\times(n-k)}$. Without loss of generality, assume that this matrix is formed by the first $n-k = \eta(\ell-\kappa)$ positions, i.e. we have

$$\boldsymbol{H} = [\boldsymbol{A} \mid \boldsymbol{B}]\,. \tag{6.27}$$

Assume $\boldsymbol{e} = [\boldsymbol{e}'', \boldsymbol{e}'] \in \mathbb{F}_{q^m}^{\eta\ell}$ with $\boldsymbol{e}'' \in \mathbb{F}_{q^m}^{\eta(\ell-\kappa)}$ and $\boldsymbol{e}' \in \mathbb{F}_{q^m}^{\eta\kappa}$. Then, by (6.27), we have

$$\boldsymbol{e}'' = (\boldsymbol{s} - \boldsymbol{e}'\boldsymbol{B}^\top)(\boldsymbol{A}^{-1})^\top.$$

The idea is to arbitrarily choose $\boldsymbol{e}'$ of length $k = \eta\kappa$. Then, on average, the expected (partial) sum-rank weight of the remaining $\ell - \kappa$ blocks is

$$\mathrm{E}\left[\mathrm{wt}_{\Sigma R}^{[n_{\kappa+1},\ldots,n_\ell]}(\boldsymbol{e}'')\right] = \bar{a} \cdot (\ell - \kappa).$$

The average probability of the sum-rank weight of $\boldsymbol{e}$ is then

$$\mathrm{E}\left[\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e})\right] = \underbrace{\mathrm{E}\left[\mathrm{wt}_{\Sigma R}^{([n_1,\ldots,n_\kappa])}(\boldsymbol{e}')\right]}_{\stackrel{\text{def}}{=}\bar{w}_1} + \mathrm{E}\left[\mathrm{wt}_{\Sigma R}^{([n_{\kappa+1},\ldots,n_\ell])}(\boldsymbol{e}'')\right] = \bar{w}_1 + \bar{a} \cdot (\ell - \kappa),$$

where $\bar{w}_1$ is determined by the distribution of $\boldsymbol{e}'$, which we can choose freely. Nonetheless, we have $0 \le \bar{w}_1 \le \mu\kappa$, and therefore

$$\underbrace{\bar{a} \cdot (\ell - \kappa)}_{=nw^-_{\text{easy}}} \le \mathrm{E}\left[\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e})\right] \le \underbrace{\mu\kappa + \bar{a} \cdot (\ell - \kappa)}_{=nw^+_{\text{easy}}}.$$

From this, we deduce that any weight in the interval $w \in [w^-_{\text{easy}}n, w^+_{\text{easy}}n]$ can be reached probabilistically in polynomial time using a distribution for $\boldsymbol{e}'$ with

$$\bar{w_1} = w - w^-_{\text{easy}}n \quad \text{s.t.} \quad \mathrm{E}\left[\mathrm{wt}^{(\boldsymbol{n})}_{\Sigma R}(\boldsymbol{e})\right] = w,$$

and which is sufficiently concentrated around its expectation. Algorithm 9 implements this approach, where in Line 6 to Line 8, the parity-check matrix $\boldsymbol{H}$ is permuted block-wise among the ℓ blocks, i.e., the permutation is applied to the block indices but not within the blocks. Here, $\otimes$ denotes the Kronecker product, which is used to construct the block-wise permutation matrix. This permutation is reversed in Line 13. □

The proposition above provides the interval $[w^-_{\text{easy}}, w^+_{\text{easy}}]$ for which a solution to Problem 6.1 can be found in probabilistic polynomial time using Algorithm 9. Combining this with the asymptotic Gilbert-Varshamov bound for the sum-rank metric [BGR21], we have the following summary of the relative weight intervals:

- $w_{\text{rel}} \in [w^-, w^+]$: A solution to Problem 6.1 is likely to exist (Gilbert-Varshamov bound for the sum-rank metric, see [BGR21])
- $w_{\text{rel}} \in [w^-_{\text{easy}}, w^+_{\text{easy}}]$: A solution to Problem 6.1 can be found in probabilistic polynomial time using a Prange-like algorithm, as stated in the proposition above.

Figure 6.6 and Figure 6.7 show the regions of hardness for finding a solution using Algorithm 9 and the bounds on the relative weight intervals for successful decoding plotted against the code rate R for different parameters. These results apply asymptotically ($\ell \to \infty$, for fixed m and μ) and on average. The "no solution", "hard", and "easy" regions indicate the difficulty of finding a solution for different code rates.

In Figure 6.6, the parameters are set to $m = \eta = 2$, $q = 2$, while in Figure 6.7, the parameters are $m = \eta = 6$, $q = 2$. Comparing the two figures, we observe that the "hard" region for large relative weights becomes smaller as the values of m and η increase. This indicates that it is easier for Algorithm 9 to decode errors of large relative weight when m and η are larger.

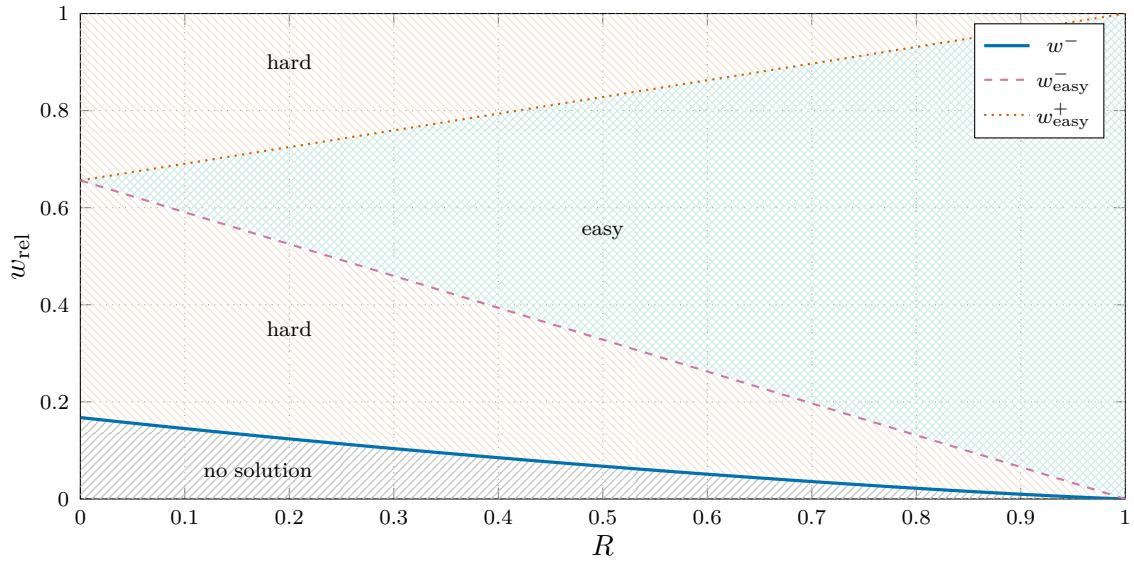


Figure 6.6: Regions of hardness for Algorithm 9 and bounds on the relative weight intervals for successful decoding vs code rate $R = k/n$ for parameters: $m = \eta = 2$, $q = 2$ ($\ell \to \infty$, average-case).

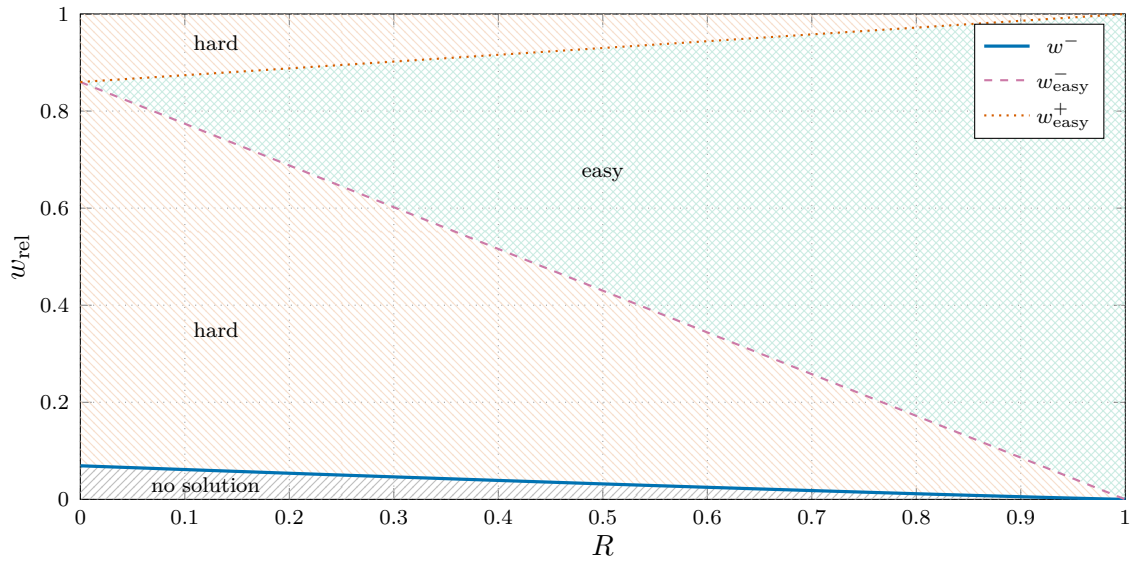


Figure 6.7: Regions of hardness for Algorithm 9 and bounds on the relative weight intervals for successful decoding vs code rate $R = k/n$ for parameters: $m = \eta = 6$, $q = 2$ ($\ell \to \infty$, average-case).

6.5 Randomized Decoding of Linearized Reed–Solomon Codes

In this section, we consider the probabilistic decoding algorithm introduced in [JBW23] to solve Problem 6.3. In [JBW23], the complexity of this decoder was analyzed for the worst-case rank profile. This algorithm generalizes the randomized decoder for Gabidulin codes from [RJB+20] to LRS codes.

We revisit the decoder from [JBW23] and present two new contributions:

- We adapt the methods from [PRR22] to efficiently compute worst-case bounds and sample the support rank profile for the randomized decoder for LRS codes.
- We extend the average-case analysis from Section 6.3 to the randomized decoder, deriving an objective function to optimize the support-drawing distribution, motivated by the asymptotic setting.

The proposed decoder relies on two key aspects:

First, we consider an underlying LRS code (see Section 2.6.5). Recall that LRS codes are MSRD codes with a minimum sum-rank distance as

$$d_{\min} = n - k + 1.$$

Efficient algorithms exist to decode LRS codes up to the unique decoding radius

$$\tau = \frac{n-k}{2}.$$

We consider LRS codes of length n partitioned into constant block lengths $\boldsymbol{n} = [n_1, \ldots, n_\ell] = [\eta, \ldots, \eta]$ and dimension k over $\mathbb{F}_{q^m}$, denoted by $\mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; \boldsymbol{n}, k]$. Additionally, LRS codes are restricted to

$$\ell \leq q-1 \quad \text{and} \quad n_i \leq m \quad \forall i \in \{1, \ldots, \ell\}.$$

Second, we focus on decoding beyond the unique decoding radius, where the sum-rank weight of the error $\boldsymbol{w}$ exceeds τ, i.e., $\mathrm{wt}_{\Sigma R}(\boldsymbol{w}) = w > \tau$. The error excess beyond the unique decoding radius is defined as

$$\xi \stackrel{\text{def}}{=} w - \tau.$$

Note that while 2ξ is always an integer, ξ itself does not necessarily need to be an integer.

As discussed in Section 6.1, for errors with weight $w \leq \tau$, Problem 6.3 has at most one solution. However, for $w > \tau$, multiple solutions may exist. The number of solutions can vary, being either polynomially or exponentially bounded in terms of the

code parameters and depending on the structure of the code. This behavior has been studied for LRS codes in [PR21]. Following the reasoning in [RJB+20], we analyze the complexity of finding at least one solution. If the code is list-decodable, this process can be repeated to obtain a list of solutions.

6.5.1 Erasures in the Sum-Rank Metric

We consider an error of the form as described in Section 6.1.4. The error $\boldsymbol{e}$ can be further decomposed into a sum of three types of error vectors

$$\boldsymbol{e} = \boldsymbol{e}_{\mathrm{F}} + \boldsymbol{e}_{\mathrm{R}} + \boldsymbol{e}_{\mathrm{C}},$$

where $\boldsymbol{e}_{\mathrm{F}}$ represents *full errors*, $\boldsymbol{e}_{\mathrm{R}}$ represents *row erasures*, and $\boldsymbol{e}_{\mathrm{C}}$ represents *column erasures*. The sum-rank weights of these error vectors are denoted by w_F, w_R, and w_C, respectively, such that $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}_{\mathrm{F}}) = w_F$, $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}_{\mathrm{R}}) = w_R$, and $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}_{\mathrm{C}}) = w_C$ (see [HBP22]).

Each of the three error vectors can be decomposed as in (2.27)

$$\begin{aligned}
\boldsymbol{e}_{\mathrm{F}} &= \boldsymbol{a}_{\mathrm{F}}\boldsymbol{B}_{\mathrm{F}} \quad \text{with} \quad \boldsymbol{a}_{\mathrm{F}} \in \mathbb{F}_{q^m}^{w_F} \text{ and } \boldsymbol{B}_{\mathrm{F}} \in \mathbb{F}_q^{w_F \times n}, \\
\boldsymbol{e}_{\mathrm{R}} &= \boldsymbol{a}_{\mathrm{R}}\boldsymbol{B}_{\mathrm{R}} \quad \text{with} \quad \boldsymbol{a}_{\mathrm{R}} \in \mathbb{F}_{q^m}^{w_R} \text{ and } \boldsymbol{B}_{\mathrm{R}} \in \mathbb{F}_q^{w_R \times n}, \\
\boldsymbol{e}_{\mathrm{C}} &= \boldsymbol{a}_{\mathrm{C}}\boldsymbol{B}_{\mathrm{C}} \quad \text{with} \quad \boldsymbol{a}_{\mathrm{C}} \in \mathbb{F}_{q^m}^{w_C} \text{ and } \boldsymbol{B}_{\mathrm{C}} \in \mathbb{F}_q^{w_C \times n}.
\end{aligned}$$

For full errors, neither $\boldsymbol{a}_{\mathrm{F}}$ nor $\boldsymbol{B}_{\mathrm{F}}$ are known. For row erasures, $\boldsymbol{a}_{\mathrm{R}}$ is known but $\boldsymbol{B}_{\mathrm{R}}$ is unknown. For column erasures, $\boldsymbol{a}_{\mathrm{C}}$ is unknown but $\boldsymbol{B}_{\mathrm{C}}$ is known.

An efficient algorithm for LRS codes was proposed in [HBP22], capable of correcting combinations of *full errors*, *row erasures*, and *column erasures* up to

$$2w_F + w_C + w_R \leq n - k, \tag{6.28}$$

with a complexity of $O(n^2)$ operations over $\mathbb{F}_{q^m}$. We estimate this complexity, denoted as $W_{\mathrm{ee-dec}}$, over $\mathbb{F}_q$ as $O(m^2n^2)$ and approximate it, similar to the arguments in Section 6.3, as

$$W_{\mathrm{ee-dec}} \approx m^2n^2. \tag{6.29}$$

We denote this error-erasure decoder by $\mathrm{DEC}(\boldsymbol{y}, \boldsymbol{a}_{\mathrm{R}}, \boldsymbol{B}_{\mathrm{C}})$, which takes as input the received word $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$, along with a basis $\boldsymbol{a}_{\mathrm{R}}$ of the column support of $\boldsymbol{e}_{\mathrm{R}}$ (row erasures) and/or a basis $\boldsymbol{B}_{\mathrm{C}}$ of the row support of $\boldsymbol{e}_{\mathrm{C}}$ (column erasures). The decoder outputs a valid codeword $\hat{c}$ if the condition in (6.28) is satisfied; otherwise, it returns $\emptyset$.

6.5.2 Randomized Decoding Algorithm

We consider an error $\boldsymbol{e}$ with row support $\boldsymbol{\mathcal{E}}_R$ and column support $\boldsymbol{\mathcal{E}}_C$. Unlike the generic decoding algorithm, where we guess a super support that must completely contain the actual error support to succeed, the randomized approach aims to guess only parts of the error supports and utilizes an error-and-erasure decoder to succeed with a smaller number of guesses. The steps are as follows:

For each block $i \in \ell$, we have $\mathcal{E}_R^{(i)} \subseteq \mathbb{F}_q^{\eta}$ and $\mathcal{E}_C^{(i)} \subseteq \mathbb{F}_q^{m}$. As shown in [RJB+20], for Gabidulin codes, guessing a combination of row and column supports does not improve success, and it is more effective to guess from a smaller ambient space. Since this result applies block-wise in the sum-rank metric, we set $\mu = \min\{m, \eta\}$ and guess from $\boldsymbol{\mathcal{E}}_R$ if $\mu = \eta$, otherwise from $\boldsymbol{\mathcal{E}}_C$.

This ensures the guessed support is always a subspace of $\mathbb{F}_q^{\mu}$. For simplicity, we use $\boldsymbol{\mathcal{E}}$ to denote the error support, whether from $\boldsymbol{\mathcal{E}}_R$ or $\boldsymbol{\mathcal{E}}_C$.

To guess parts of $\boldsymbol{\mathcal{E}}$, we first draw a corresponding rank profile $\boldsymbol{u} = [u_1, \ldots, u_\ell] \in \mathbb{Z}_{\geq 0}^{\ell}$ according to some PMF denoted as $\beta_{\boldsymbol{u}} \stackrel{\text{def}}{=} \Pr[\boldsymbol{u}]$. Then, a support $\boldsymbol{\mathcal{U}}$ is drawn uniformly at random from $\Xi_{q,\mu}(\boldsymbol{u})$ with $u \stackrel{\text{def}}{=} \dim_{\Sigma}(\boldsymbol{\mathcal{U}})$. Define ϵ as the sum dimension of the intersection space between the guessed space $\boldsymbol{\mathcal{U}}$ and the actual error support $\boldsymbol{\mathcal{E}}$, i.e.,

$$\epsilon \stackrel{\text{def}}{=} \dim_{\Sigma}(\boldsymbol{\mathcal{U}} \cap \boldsymbol{\mathcal{E}}).$$

The number of full errors w_F is reduced by ϵ, so $w_F = w - \epsilon$, while the number of column or row erasures increases by u, corresponding to the guessed parts of $\boldsymbol{\mathcal{U}}$. For these guessed parts, we assume knowledge of the column support but not the row support (or vice versa), effectively trading errors for erasures [2].

The error-and-erasure decoder takes as input a vector containing $w - \epsilon$ full errors and u erasures. From the decoding condition in (6.28), we have

$$2(w - \epsilon) + u \leq n - k,$$

which implies that for successful decoding, we need to have

$$\epsilon \geq \epsilon_{\min} \stackrel{\text{def}}{=} w + \frac{u - (n-k)}{2} = \xi + \frac{u}{2}. \tag{6.30}$$

If the intersection between the guessed spaces and the actual error support is sufficiently large, an error-erasure decoder can successfully decode. From (6.30), the valid range for u is $u \in \{2\xi, \ldots, n-k\}$, with the lower bound ensuring $\epsilon \geq 0$ and the upper bound corresponding to the most favorable case, where $\epsilon = w$. In the latter case, $u \leq n - k$, which is the maximum erasure decoding capability for LRS codes. The

[2] This approach is reminiscent of the generalized minimum distance (GMD) decoding strategy introduced by Forney [For66] for Hamming metric codes, and later extended to the rank metric in [BCG+03] by Bossert *et al.*.

performance of the randomized decoder depends on the choice of the PMF $\beta_{\boldsymbol{u}}$ used to draw the rank profile $\boldsymbol{u}$. The optimal choice of $\beta_{\boldsymbol{u}}$ and the analysis of the algorithm's success probability will be discussed in the following sections.

Algorithm 10 outlines the proposed approach. It can be easily generalized to variable block lengths and other sum-rank-metric codes that support efficient error-and-erasure decoders.

Algorithm 10: Randomized Sum-Rank Metric Decoder for LRS codes

Input : Parameters: q, m, η, ℓ, w and u with $2\xi \leq u \leq n-k$ and $w \geq \tau$
Received vector $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$
LRS code $\mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; n = \eta\ell, k]$
Error-erasure decoder $\mathrm{DEC}(\cdot,\cdot,\cdot)$ for $\mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; n, k]$

Output : Vector $\boldsymbol{c}' \in \mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; n, k]$ such that $\mathrm{wt}_{\Sigma R}(\boldsymbol{y} - \boldsymbol{c}') = w$

$\boldsymbol{c}' \leftarrow \emptyset$
$\mu \leftarrow \min\{m, \eta\}$
while $\boldsymbol{c}' = \emptyset$ **or** $\mathrm{wt}_{\Sigma R}(\boldsymbol{y} - \boldsymbol{c}') \neq w$ **do**
  $\boldsymbol{u} \leftarrow$ Draw rank profile for the guess space according to $\beta_{\boldsymbol{u}}$
  $\boldsymbol{\mathcal{U}} \xleftarrow{\$} \Xi_{q,\mu}(\boldsymbol{u})$
  if $\eta < m$ **then**
    $\boldsymbol{B}_{\mathrm{C}} \leftarrow$ Basis of $\boldsymbol{\mathcal{U}}$
    $\boldsymbol{c}' \leftarrow \mathrm{DEC}(\boldsymbol{y}, [\,], \boldsymbol{B}_{\mathrm{C}})$ `/* Error-erasure decoding with row erasures */`
  else
    $\boldsymbol{a}_{\mathrm{R}} \leftarrow$ Basis of $\boldsymbol{\mathcal{U}}$
    $\boldsymbol{c}' \leftarrow \mathrm{DEC}(\boldsymbol{y}, \boldsymbol{a}_{\mathrm{R}}, [\,])$ `/* Error-erasure decoding with column erasures */`
return $\boldsymbol{c}'$

The following lemma provides a useful result that contributes to the derivation of the average complexity of the randomized approach in Algorithm 10.

Lemma 6.1. *For a fixed error vector $\boldsymbol{e} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_\ell] \in \mathbb{F}_{q^m}^n$ with a given rank profile $\boldsymbol{w} = [w_1, \ldots, w_\ell] \in \mathbb{Z}_{\geq 0}^\ell$ and another given rank profile $\boldsymbol{u} = [u_1, \ldots, u_\ell] \in \mathbb{Z}_{\geq 0}^\ell$, let $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\mathcal{U}}$ be the error space and guessed space, respectively, where $\boldsymbol{\mathcal{U}}$ is chosen uniformly from $\Xi_{q,\mu}(\boldsymbol{u})$. Further, let S_j denote the event that $\dim_\Sigma(\boldsymbol{\mathcal{E}} \cap \boldsymbol{\mathcal{U}}) = j$. The probability of S_j given $\boldsymbol{e}$ and $\boldsymbol{u}$ is then*

$$\Pr[S_j|\boldsymbol{e}, \boldsymbol{u}] = \Pr[S_j|\boldsymbol{w}, \boldsymbol{u}] = \left(\circledast_{i=1}^{\ell} \mathsf{P}^{\cap}_{q,\mu,w_i,u_i} \right)(j), \tag{6.31}$$

with

$$\left(\circledast_{i=1}^{\ell} \mathsf{P}^{\cap}_{q,\mu,w_i,u_i}\right)(j) \stackrel{\text{def}}{=} \left(\mathsf{P}^{\cap}_{q,\mu,w_1,u_1} \circledast \cdots \circledast \mathsf{P}^{\cap}_{q,\mu,w_\ell,u_\ell}\right)(j),$$

being the ℓ-fold discrete convolution (denoted by $\circledast$) of the probability distributions $\mathsf{P}^{\cap}_{q,\mu,w_i,u_i}$ evaluated at j for all $i \in \{1,\ldots,\ell\}$. Here, $\mathsf{P}^{\cap}_{q,\mu,w_i,u_i}$ represents the probability distribution of the intersection dimension between the error space and the guessed space in the i-th shot.

Proof. Given the error $\boldsymbol{e} = [\boldsymbol{e}_1,\ldots,\boldsymbol{e}_\ell] \in \mathbb{F}_{q^m}^n$ with rank profile $\boldsymbol{w} = [w_1,\ldots,w_\ell] \in \mathbb{Z}_{\geq 0}^\ell$ and the rank profile $\boldsymbol{u} = [u_1,\ldots,u_\ell] \in \mathbb{Z}_{\geq 0}^\ell$ of the guessed support, let V_i be a random variable that corresponds to the dimension of the intersection of the i-th guessed space $\mathcal{U}^{(i)}$ with the i-th actual error space $\mathcal{E}^{(i)}$ for $i \in \{1,\ldots,\ell\}$. By (2.7), we have that $\mathsf{P}^{\cap}_{q,\mu,w_i,u_i}(j)$ is the probability of that event, i.e.,

$$\Pr[V_i = j|\boldsymbol{e},\boldsymbol{u}] = \mathsf{P}^{\cap}_{q,\mu,w_i,u_i}(j).$$

Note that the probability $\Pr[V_i = j|\boldsymbol{e},\boldsymbol{u}]$ depends only on the rank weights w_i and u_i of the error and guessed support in the i-th shot, respectively, and not on the specific error vector $\boldsymbol{e}$. Thus, we can write

$$\Pr[V_i = j|\boldsymbol{e},\boldsymbol{u}] = \Pr[V_i = j|\boldsymbol{w},\boldsymbol{u}] = \mathsf{P}^{\cap}_{q,\mu,w_i,u_i}(j).$$

Since we are interested in the probability distribution of the sum of random variables, i.e., $V = \sum_{i=1}^{\ell} V_i$, the resulting probability distribution is given by the ℓ-fold discrete convolution of the probability distributions of the random variables V_i for $i \in \{1,\ldots,\ell\}$. Thus

$$\Pr[V = j|\boldsymbol{e},\boldsymbol{u}] = \Pr[V = j|\boldsymbol{w},\boldsymbol{u}] = \left(\circledast_{i=1}^{\ell} \mathsf{P}^{\cap}_{q,\mu,w_i,u_i}\right)(j),$$

with

$$\left(\mathsf{P}^{\cap}_{q,\mu,w_1,u_1} \circledast \mathsf{P}^{\cap}_{q,\mu,w_2,u_2}\right)(j) \stackrel{\text{def}}{=} \sum_{r=-\infty}^{\infty} \mathsf{P}^{\cap}_{q,\mu,w_1,u_1}(r)\, \mathsf{P}^{\cap}_{q,\mu,w_2,u_2}(j-r). \tag{6.32}$$

Finally, let S_j be the event that $V = j$, which proves the claim. □

The probability of the event S_j given the rank profiles of the error and the guessed support, as stated in Lemma 6.1, can be further expanded using the concept of rank profiles. The following proposition expresses this probability.

Proposition 6.2. *Let $\boldsymbol{e} = [\boldsymbol{e}_1,\ldots,\boldsymbol{e}_\ell] \in \mathbb{F}_{q^m}^n$ be a fixed error with given rank profile $\boldsymbol{w} = [w_1,\ldots,w_\ell] \in \mathbb{Z}_{\geq 0}^\ell$, and let $\boldsymbol{u} = [u_1,\ldots,u_\ell] \in \mathbb{Z}_{\geq 0}^\ell$ be another given rank profile. Further, let $\mu = \min\{\eta, m\}$ and $\epsilon \in \{0,\ldots,\ell\mu\}$. Then, we can write* (6.31) *from*

Lemma 6.1 as

$$\Pr[S_\epsilon|\boldsymbol{w},\boldsymbol{u}] = \sum_{\boldsymbol{\epsilon}\in\mathcal{T}_{\epsilon,\ell,\mu}} \prod_{i=1}^{\ell} \mathsf{P}^{\cap}_{q,\mu,w_i,u_i}(\epsilon_i). \tag{6.33}$$

Proof. Starting from the right-hand side of (6.33), we have

$$\begin{aligned}\sum_{\boldsymbol{\epsilon}\in\mathcal{T}_{\epsilon,\ell,\mu}} \prod_{i=1}^{\ell} \mathsf{P}^{\cap}_{q,\mu,w_i,u_i}(\epsilon_i) &= \sum_{\substack{\mu\geq\epsilon_1,\ldots,\epsilon_\ell\geq 0\\ \epsilon_1+\cdots+\epsilon_\ell=\epsilon}} \prod_{i=1}^{\ell} \mathsf{P}^{\cap}_{q,\mu,w_i,u_i}(\epsilon_i)\\ &= \left(\circledast_{i=1}^{\ell} \mathsf{P}^{\cap}_{q,\mu,w_i,u_i}\right)(\epsilon)\\ &= \Pr[S_\epsilon|\boldsymbol{w},\boldsymbol{u}],\end{aligned}$$

where the first equality follows from the definition of the set of rank profiles, the second equality follows from the definition of discrete convolution as defined in (6.32), and the last equality follows from the recursive application of the definition of discrete convolution of two PMFs. This completes the proof. □

The probability of successful decoding is given by the sum of the probabilities of the events S_ϵ over all feasible values of the total intersection dimension ϵ. Since these events are mutually exclusive, we define

$$\phi_\mu(\boldsymbol{u},\boldsymbol{w}) \stackrel{\text{def}}{=} \sum_{\epsilon=\epsilon_{\min}}^{\min\{u,w\}} \Pr[S_\epsilon \mid \boldsymbol{u},\boldsymbol{w}], \tag{6.34}$$

where $\phi_\mu(\boldsymbol{u},\boldsymbol{w})$ denotes the probability of successful decoding. The lower bound $\epsilon_{\min}$, defined in (6.30), ensures that the intersection support has enough dimensions for successful decoding. The upper bound $\min\{u,w\}$ reflects that the intersection cannot have a greater dimension than either of the supports.

Given a probability distribution $\beta_{\boldsymbol{u}}$ over the rank profiles $\boldsymbol{u}$, the overall probability of successful decoding for a fixed error rank profile $\boldsymbol{w}$ is

$$\phi_{\mu,u}(\boldsymbol{w}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{u}\in\mathcal{T}_{u,\ell,\mu}} \beta_{\boldsymbol{u}} \cdot \phi_\mu(\boldsymbol{u},\boldsymbol{w}). \tag{6.35}$$

6.5.3 Worst-Case Complexity

When decoding beyond the unique decoding radius for LRS codes (Problem 6.3), multiple solutions may exist, and the proposed decoder (Algorithm 10) lacks a mechanism to identify the original codeword among them. To make the complexity analysis exact, we assume a genie-aided version of the decoder that can identify the correct solution and allows us to stop the decoder at the iteration when the original codeword is found. Consequently, only the upper bound on the expected complexity applies to

the non-genie-aided decoder, similar to the bounds derived in [PRR22] for the generic decoder.

We analyze the worst-case expected complexity over all possible rank profiles of errors with $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}) = w$. We derive lower and upper bounds on this complexity and provide algorithms to efficiently compute these bounds. Additionally, we present methods to optimize the guessing-support distribution for this worst-case scenario.

Determining the optimal sum-rank weight $u \in \{2\xi, \dots, n-k\}$ for the guessing support to minimize the expected complexity of Algorithm 10 is not straightforward. The success probability bounds from [RJB+20] for the rank metric are convex in u, suggesting that the probability is maximized at $u = 2\xi$ or $u = n-k$. However, this does not guarantee that these values always yield the optimal expected complexity. Similarly, for the sum-rank metric, we cannot be certain, though these extreme values for u remain important points of interest. For these two specific values of u, we can express the success probability from (6.34) as follows. When $u = 2\xi \leq w$, we have

$$\phi_\mu(\boldsymbol{u}, \boldsymbol{w}) = \sum_{\epsilon=u}^{u} \Pr[S_\epsilon | \boldsymbol{u}, \boldsymbol{w}] = \prod_{i=1}^{\ell} \mathsf{P}_{q,\mu}^{\subseteq}(u_i, w_i)\,. \tag{6.36}$$

On the other hand, when $u = n-k \geq w$, we have

$$\phi_\mu(\boldsymbol{u}, \boldsymbol{w}) = \sum_{\epsilon=w}^{w} \Pr[S_\epsilon | \boldsymbol{u}, \boldsymbol{w}] = \prod_{i=1}^{\ell} \mathsf{P}_{q,\mu}^{\subseteq}(w_i, u_i)\,. \tag{6.37}$$

Notably, choosing either $u = 2\xi$ or $u = n-k$ simplifies the expression for the success probability, as the convolution in (6.34) reduces to a simple product over the blocks. This simplification is advantageous for our analysis of the decoder's performance.

When $u = n-k$, the success probability matches that of the fully generic decoding algorithm, as the expressions in (6.36) and (6.3) are identical. In this case, the bounds from [PRR22] apply directly, with the only difference being the per-iteration complexity, which is replaced by that of the error-and-erasure decoder for LRS codes. Therefore, setting $u = n-k$ offers no improvement in success probability over the fully generic approach.

When $u = 2\xi$ and $u \leq w$, the algorithms from [PRR22] require adjustments. We present these modifications below, using the notation

$$\phi'_\mu(\boldsymbol{u}, \boldsymbol{w}) \stackrel{\text{def}}{=} \prod_{i=1}^{\ell} \mathsf{P}_{q,\mu}^{\subseteq}(u_i, w_i)\,. \tag{6.38}$$

From (6.37) follows the worst-case expected number of iterations of Algorithm 10

for a given PMF of the guessing support, denoted as $\tilde{\beta}_{\boldsymbol{u}} \stackrel{\text{def}}{=} \Pr[\boldsymbol{u}]$ by

$$\max_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \mathbb{E}[\#\text{iterations}] = \max_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \left(\sum_{\boldsymbol{u}\in\mathcal{T}_{u,\ell,\mu}} \tilde{\beta}_{\boldsymbol{u}} \phi'_{\mu}(\boldsymbol{u},\boldsymbol{w}) \right)^{-1}. \tag{6.39}$$

The problem of minimizing the worst-case expected number of iterations over all valid distributions $\tilde{\beta}_{\boldsymbol{u}}$ on $\mathcal{T}_{u,\ell,\mu}$ can be formulated as a linear program. While this linear program can be solved numerically using standard methods for small values of μ, ℓ, and u, the number of unknowns, i.e., $\tilde{\beta}_{\boldsymbol{u}} \in [0,1]$, grows rapidly as these parameters increase. Consequently, solving the linear program directly becomes computationally prohibitive for larger problem instances.

To tackle this computational challenge, we use the approach described in [PRR22], which introduces a randomized mapping

$$\text{ucomp}_{\mu} \colon \mathcal{T}_{w,\ell,\mu} \times \mathbb{Z}_{\geq 0} \to \mathcal{T}_{u,\ell,\mu}.$$

This mapping aims to maximize the probability

$$\phi'_{\mu}(\text{ucomp}_{\mu}(\boldsymbol{w},u),\boldsymbol{w}),$$

for a given rank profile $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}$ by randomly selecting an output vector from multiple possible candidates for each input, providing a more computationally tractable approach to the problem.

Rather than directly choosing a rank profile $\boldsymbol{u} \in \mathcal{T}_{u,\ell,\mu}$, we first select a rank profile $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}$ at random according to a designed distribution $\gamma_{\boldsymbol{w}}$ on $\mathcal{T}_{w,\ell,\mu}$, and then set

$$\boldsymbol{u} \leftarrow \text{ucomp}_{\mu}(\boldsymbol{w},u).$$

For a fixed error $\boldsymbol{e}$, this allows us to bound the probability as follows

$$\Pr(\boldsymbol{\mathcal{U}} \subseteq \boldsymbol{\mathcal{E}}_{\boldsymbol{e}}) = \sum_{\boldsymbol{u}\in\mathcal{T}_{u,\ell,\mu}} \tilde{\beta}_{\boldsymbol{u}} \cdot \phi'_{\mu}(\boldsymbol{u},\boldsymbol{w}_{\boldsymbol{e}}) \geq \gamma_{\boldsymbol{w}_{\boldsymbol{e}}} \cdot \phi'_{\mu}(\text{ucomp}_{\mu}(\boldsymbol{w}_{\boldsymbol{e}},u),\boldsymbol{w}_{\boldsymbol{e}}).$$

Using this bound, we can minimize the following upper bound on the worst-case expected number of iterations, instead of directly minimizing (6.39)

$$\max_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \mathbb{E}[\#\text{iterations}] \leq \max_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \left(\gamma_{\boldsymbol{w}} \cdot \phi'_{\mu}(\text{ucomp}_{\mu}(\boldsymbol{w},u),\boldsymbol{w})\right)^{-1},$$

over all valid probability mass functions $\tilde{\beta}_{\boldsymbol{u}}$ on $\mathcal{T}_{u,\ell,\mu}$.

The randomized mapping ucomp_{μ} is formally defined in Appendix B.2.1 and its correctness is proofed in Lemma B.1.

We adapt the support-drawing algorithm from [PRR22] to handle cases where the

sum dimension of the guessed support is smaller than that of the error support. The modified version, shown in Algorithm 11, retains the structure of the original algorithm but is adjusted to account for this dimension difference.

Algorithm 11: DrawRandomSupport(u, w, μ)

Input : Integers $u, \mu, w \in \mathbb{Z}_{\geq 0}$ with $u \leq w$
Output : $\boldsymbol{\mathcal{U}}$ of sum dimension u

Draw $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}$ according to the distribution $\gamma_{\boldsymbol{w}}$ defined in (6.40).
$\boldsymbol{u} \leftarrow \mathrm{ucomp}_{\mu}(\boldsymbol{w}, u)$
$\boldsymbol{\mathcal{U}} \xleftarrow{\$} \Xi_{\mu,\zeta}(\boldsymbol{u})$
return $\boldsymbol{\mathcal{U}}$

We define the probability distribution $\gamma_{\boldsymbol{w}}$ as follows

$$\gamma_{\boldsymbol{w}} \stackrel{\text{def}}{=} \left(\phi'_{\mu}(\mathrm{ucomp}_{\mu}(\boldsymbol{w}, u), \boldsymbol{w}) \cdot \tilde{Q}_{\ell,w,\mu}\right)^{-1} \quad \forall \boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}, \tag{6.40}$$

where $\tilde{Q}_{\ell,w,\mu}$ is defined as

$$\tilde{Q}_{\ell,w,\mu} \stackrel{\text{def}}{=} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \phi'_{\mu}(\mathrm{ucomp}_{\mu}(\boldsymbol{w}, u), \boldsymbol{w})^{-1}. \tag{6.41}$$

The following proposition presents bounds on the expected number of iterations.

Proposition 6.3. *Let $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ be an error of sum-rank weight w and let u be an integer with $u \leq w$. If $\boldsymbol{\mathcal{U}}$ is a sub-support that is drawn by Algorithm 11 with input u and w, then we have*

$$|\mathcal{T}_{w,\ell,\mu}|^{-1} \tilde{Q}_{\ell,w,\mu} \leq \Pr(\boldsymbol{\mathcal{U}} \subseteq \boldsymbol{\mathcal{E}}_{\boldsymbol{e}})^{-1} \leq \tilde{Q}_{\ell,w,\mu},$$

where $\tilde{Q}_{\ell,w,\mu}$ is defined as in (6.41)*, and $\boldsymbol{\mathcal{E}}_{\boldsymbol{e}}$ denotes the error support corresponding to the error vector $\boldsymbol{e}$.*

Proof. Denote by $\gamma_{\boldsymbol{w}}$ the distribution of $\boldsymbol{w} = \mathrm{ucomp}_{\mu}(\boldsymbol{\mathcal{E}}_{\boldsymbol{e}})$, where $\boldsymbol{w}$ is a random variable with probability mass function $\gamma_{\boldsymbol{w}}$ as defined in (6.40). By the law of total probability, we have

$$\begin{aligned}
\Pr(\boldsymbol{\mathcal{U}} \subseteq \boldsymbol{\mathcal{E}}_{\boldsymbol{e}}) &= \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \gamma_{\boldsymbol{w}} \cdot \phi'_{\mu}(\mathrm{ucomp}_{\mu}(\boldsymbol{w}, u), \boldsymbol{w}_{\boldsymbol{e}}) \\
&\geq \gamma_{\boldsymbol{w}_{\boldsymbol{e}}} \cdot \phi'_{\mu}(\mathrm{ucomp}_{\mu}(\boldsymbol{w}_{\boldsymbol{e}}, u), \boldsymbol{w}_{\boldsymbol{e}}) \\
&= \tilde{Q}^{-1}_{\ell,w,\mu},
\end{aligned}$$

where the last equality follows from the definition of $\gamma_{\boldsymbol{w}}$ in (6.40). This proves the upper bound on $\Pr(\boldsymbol{\mathcal{U}} \subseteq \boldsymbol{\mathcal{E}}_{\boldsymbol{e}})^{-1}$. For the lower bound, we observe that for all $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}$

$$\phi'_\mu(\mathrm{ucomp}_\mu(\boldsymbol{w}, u), \boldsymbol{w}_{\boldsymbol{e}}) \leq \phi'_\mu(\mathrm{ucomp}_\mu(\boldsymbol{w}, u), \boldsymbol{w}),$$

which yields

$$\begin{aligned}\Pr(\boldsymbol{\mathcal{U}} \subseteq \boldsymbol{\mathcal{E}}_{\boldsymbol{e}}) &= \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \gamma_{\boldsymbol{w}} \cdot \phi'_\mu(\mathrm{ucomp}_\mu(\boldsymbol{w}, u), \boldsymbol{w}_{\boldsymbol{e}}) \\ &\leq \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \gamma_{\boldsymbol{w}} \cdot \phi'_\mu(\mathrm{ucomp}_\mu(\boldsymbol{w}, u), \boldsymbol{w}) \\ &= \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \tilde{Q}^{-1}_{\ell,w,\mu} = |\mathcal{T}_{w,\ell,\mu}| \tilde{Q}^{-1}_{\ell,w,\mu},\end{aligned}$$

where the last equality follows from the definitions of $\tilde{Q}_{\ell,w,\mu}$ in (6.41) and the design distribution $\gamma_{\boldsymbol{w}}$ in (6.40), which proves the claim. □

Using Proposition 6.3, we can formulate the following theorem about the expected runtime of the genie-aided version of Algorithm 10.

Theorem 6.6. *Consider a genie-aided version of Algorithm 10 for an LRS code* $\mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; \boldsymbol{n}, k]$ *of length* n*, dimension* k*, and length partition* $\boldsymbol{n}$ *with constant block length* $n_i = \eta$ *for all* $i \in \{1, \ldots, \ell\}$*. Let* $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ *be an error of sum-rank weight* $\tau < w \leq n - k$*, and let* $\boldsymbol{c} \in \mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; \boldsymbol{n}, k]$ *be a codeword. We consider the success event of the algorithm returning the originally trasnmitted codeword* $\boldsymbol{c}$ *when given input* $\boldsymbol{y} = \boldsymbol{e} + \boldsymbol{c}$ *and parameter* u *with* $u = 2\xi$*.*

Each iteration of Algorithm 10 costs $W_{\mathrm{rand}}^{(\mathrm{iter})}$*. By including also the expected number of iterations, we can bound the overall expected runtime* W_{rand} *of the genie-aided version of Algorithm 10 by*

$$W_{\mathrm{rand}}^{(\mathrm{LB})} \leq W_{\mathrm{rand}} \leq W_{\mathrm{rand}}^{(\mathrm{UB})},$$

where, for $\mu = \min\{\eta, m\}$ *and* $\tilde{Q}_{\ell,w,\mu}$ *as in* (6.41)*, we define*

$$\begin{aligned}W_{\mathrm{rand}}^{(\mathrm{LB})} &\stackrel{\mathrm{def}}{=} |\mathcal{T}_{w,\ell,\mu}|^{-1} \cdot \tilde{Q}_{\ell,w,\mu}, \\ W_{\mathrm{rand}}^{(\mathrm{UB})} &\stackrel{\mathrm{def}}{=} W_{\mathrm{rand}}^{(\mathrm{iter})} \cdot \tilde{Q}_{\ell,w,\mu}.\end{aligned}$$

Proof. The bounds follow directly from Proposition 6.3 by multiplying the cost of a single iteration W_{iter} by the expected number of iterations:

$$\begin{aligned}W_{\mathrm{rand}}^{(\mathrm{LB})} &\stackrel{\mathrm{def}}{=} |\mathcal{T}_{w,\ell,\mu}|^{-1} \cdot \tilde{Q}_{\ell,w,\mu} \leq W_{\mathrm{rand}} \\ &\leq W_{\mathrm{iter}} \cdot \tilde{Q}_{\ell,w,\mu} \stackrel{\mathrm{def}}{=} W_{\mathrm{rand}}^{(\mathrm{UB})}.\end{aligned}$$

□

Remark 6.2. *The complexity of one iteration $W_{\mathrm{rand}}^{(\mathrm{iter})}$ in Theorem 6.6 is determined by two main components. First, the support drawing algorithm from [PRR22] can be easily adapted to our case, which yields a complexity of $\widetilde{O}(n^3m^2\log_2(q))$ bit operations.*

Second, the overall complexity $W_{\mathrm{rand}}^{(\mathrm{iter})}$ is then the sum of the complexity of the support drawing algorithm and the complexity of the error and erasure decoder, which is of the order of $O(n^2m^2)$ operations over $\mathbb{F}_q$ (see (6.29)). Similar to the complexity analysis for the worst-case scenario in the generic decoding algorithm, we approximate $W_{\mathrm{rand}}^{(\mathrm{iter})}$ with the two dominating terms in each of the complexities as $W_{\mathrm{rand}}^{(\mathrm{iter})} \approx n^3m^2$ when we plot or evaluate the complexities.

To evaluate the bounds from Theorem 6.6, we must compute $\tilde{Q}_{\ell,w,\mu}$. Direct computation using (6.41) is infeasible, as the number of summands $|\mathcal{T}_{w,\ell,\mu}|$ can grow superpolynomially with w, depending on ℓ and μ. In Appendix B.2.2, we show, following [PRR22, Lemma 22], how to compute this efficiently in $\widetilde{O}(wun^3\mu^3\log_2(q))$ bit operations.

6.5.4 Average Complexity

We analyze the average complexity of the randomized decoding algorithm over all possible error vectors $\boldsymbol{e}$ with sum-rank weight w, similar to the analysis for the generic decoder in Section 6.3.2. Although Algorithm 10 operates on an LRS code with significant structure, we use random coding arguments akin to the generic decoding approach to estimate the average success probability when decoding beyond the unique decoding radius. This accounts for the additional codeword solutions that may appear in this regime. Note that since LRS codes are not random codes, applying random coding arguments only yields an approximation for the lower bound.

Adapting Theorem 6.4, we replace the expression for the success probability of one iteration of the decoding loop to obtain bounds for our randomized decoder.

Corollary 6.4. *Let $\mathcal{C}$ be a random $\mathbb{F}_{q^m}$-linear code of length n and dimension k over $\mathbb{F}_{q^m}$, where each codeword is drawn uniformly at random from $\mathbb{F}_{q^m}^n$. Suppose the received word $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$, where $\boldsymbol{c} \in \mathcal{C}$ and $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ with $\mathrm{wt}_{\Sigma R}(\boldsymbol{e}) = w$. Assume we have an error-and-erasure decoder that can correct combinations of errors and erasures up to the condition in (6.28). Then, the success probability of Algorithm 10 to output at least one solution satisfies*

$$\Pr[\mathit{success}] \geq \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{u'=w}^{v_{\max}} \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \phi_{q,\mu,u'}(\boldsymbol{w}) \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m,\eta,w_i\right),$$

and

$$\Pr[\mathit{success}] \leq \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)}\right) \sum_{u'=w}^{v_{\max}} \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \phi_{q,\mu,u'}(\boldsymbol{w}) \prod_{i=1}^{\ell} \mathrm{NM}_q\left(m,\eta,w_i\right).$$

Proof. The proof follows the same steps as in Theorem 6.4, replacing $\varphi_{q,\mu,v'}(\boldsymbol{w})$ with $\phi_{q,\mu,u'}(\boldsymbol{w})$. □

6.5.5 Optimizing the Support-Drawing Distribution

We propose a heuristic approach to optimize the support-drawing distribution used in Algorithm 10. Inspired by the asymptotic analysis in Section 6.3.4, we aim at maximizing the average intersection between the guessed support and the actual error support, thereby increasing the probability of successful decoding.

To simplify the optimization, we consider an asymptotic setting where the number of blocks ℓ tends to infinity. In this context, we approximate the rank weight distributions of the error and the guessed support by their marginal distributions for a single block, assuming they are independently and identically distributed across blocks.

Let $\beta_u^{(\mathrm{m})} \stackrel{\text{def}}{=} \Pr[\dim(\mathcal{U}^{(i)}) = u]$ denote the marginal distribution of the rank weight of the guessed support $\mathcal{U}^{(i)}$ for a single block i, where $u \in \{0, \ldots, \mu\}$. The joint distribution of the guessing rank profile $\boldsymbol{u}$ is then

$$\beta_{\boldsymbol{u}} = \prod_{i=1}^{\ell} \beta_{u_i}^{(\mathrm{m})}. \tag{6.42}$$

We introduce the random variable Z, representing the dimension of the intersection between the guessed support and the error support for a single block. Our objective is to maximize the expected value $\mathbb{E}[Z]$, as a larger average intersection increases the probability of successful decoding with an error-and-erasure decoder, which requires a sufficiently large intersection dimension (see (6.30)). The expectation $\mathbb{E}[Z]$ can be computed as

$$\mathbb{E}[Z] = \sum_{\epsilon=0}^{\mu} \sum_{w'=0}^{\mu} \sum_{u'=0}^{\mu} \beta_{u'}^{(\mathrm{m})} \cdot \Pr[w'] \cdot \mathsf{P}^{\cap}_{q,\mu,w',u'}(\epsilon),$$

where $\Pr[w']$ is the marginal distribution of the error rank weight for a single block.

The distribution $\boldsymbol{\beta}^{(\mathrm{m})}$ can be optimized using LP methods to maximize $\mathbb{E}[Z]$, similar to the approach in Section 6.3.4.

The following theorem provides bounds on the expected runtime of the randomized sum-rank decoder (Algorithm 10) based on the success probability bounds derived in Corollary 6.4.

Theorem 6.7. *Under the same assumptions as in Corollary 6.4, the overall expected runtime $W_{\mathrm{rand,RCU}}$ of Algorithm 10 to output at least one solution is bounded as*

$$W_{\mathrm{rand,RCU}}^{(\mathrm{LB})} \leq W_{\mathrm{rand,RCU}} \leq W_{\mathrm{rand,RCU}}^{(\mathrm{UB})},$$

where the lower and upper bounds are given by

$$W_{\text{rand,RCU}}^{(\text{LB})} \stackrel{\text{def}}{=} W_{\text{ee-dec}} \left(\left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)} \right) \sum_{u'=w}^{v_{\max}} C_{q,m,\eta}(\ell, w, u') \right)^{-1},$$

and

$$W_{\text{rand,RCU}}^{(\text{UB})} \stackrel{\text{def}}{=} W_{\text{ee-dec}} \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{u'=w}^{v_{\max}} C_{q,m,\eta}(\ell, w, u') \right)^{-1}.$$

Here, $W_{\text{ee-dec}} \in O(n^2m^2)$ over $\mathbb{F}_q$ represents the complexity of the error-and-erasure decoder from [HBP22], which is discussed in detail in Section 6.5.2. The term $C_{q,m,\eta}(\ell, w, u)$ is defined as

$$C_{q,m,\eta}(\ell, w, u) \stackrel{\text{def}}{=} \sum_{\varepsilon=\epsilon_{\min}}^{\min\{u,w\}} \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \sum_{\boldsymbol{u} \in \mathcal{T}_{u,\ell,\mu}} \sum_{\boldsymbol{\epsilon} \in \mathcal{T}_{\varepsilon,\ell,\mu}} \prod_{i=1}^{\ell} \beta_{u_i}^{(\text{m})} \cdot \mathsf{P}^{\cap}_{q,\mu,w_i,u_i}(\epsilon_i) \cdot \text{NM}_q\left(m, \eta, w_i\right).$$

Proof. The proof follows similar arguments as in Theorem 6.5. The main difference is that the bounds on the success probability are replaced by the expressions derived in Corollary 6.4, which involve the definitions from (6.33), (6.34), (6.35), and (6.42). The complexity of one iteration of Algorithm 10 is given by $W_{\text{ee-dec}}$, which is the complexity of the error and erasure decoder used in the randomized algorithm. □

Remark 6.3. *The function $C_{q,m,\eta}(\ell, w, u)$ plays a crucial role in determining the complexity bounds of the randomized sum-rank syndrome decoder. It can be computed efficiently using a dynamic programming routine similar to Algorithm 12 in polynomial time.*

6.5.6 Numerical Results

We compare the performance of the randomized decoding algorithm for LRS codes with the generic decoder . Figure 6.8 and Figure 6.9 illustrate the expected complexities for both algorithms under two different parameter sets, ensuring that the total number of bits, calculated as $m \log_2(q) = 144$, remains constant.

In both figures, we set $n = 48$ and $k = 24$, resulting in a minimum sum-rank distance $d_{\min} = 25$, and a unique decoding radius $\tau = \left\lfloor \frac{n-k}{2} \right\rfloor = 12$. We consider errors with sum-rank weight $w = \tau + \xi = 13$, where the error excess is $\xi = 1$.

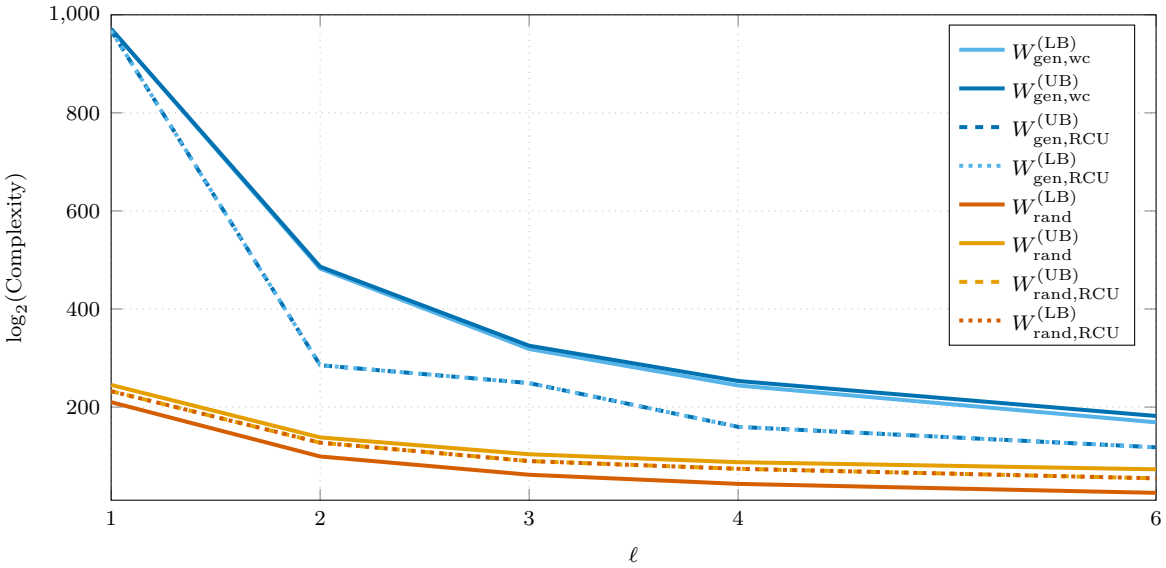


Figure 6.8: Complexity comparison of generic decoding vs. randomized decoding beyond the unique decoding radius for parameters: $q = 2^3$, $m = 48$, $n = 48$, $k = 24$, $w = 13$, $u = 2$, $v = 24$.

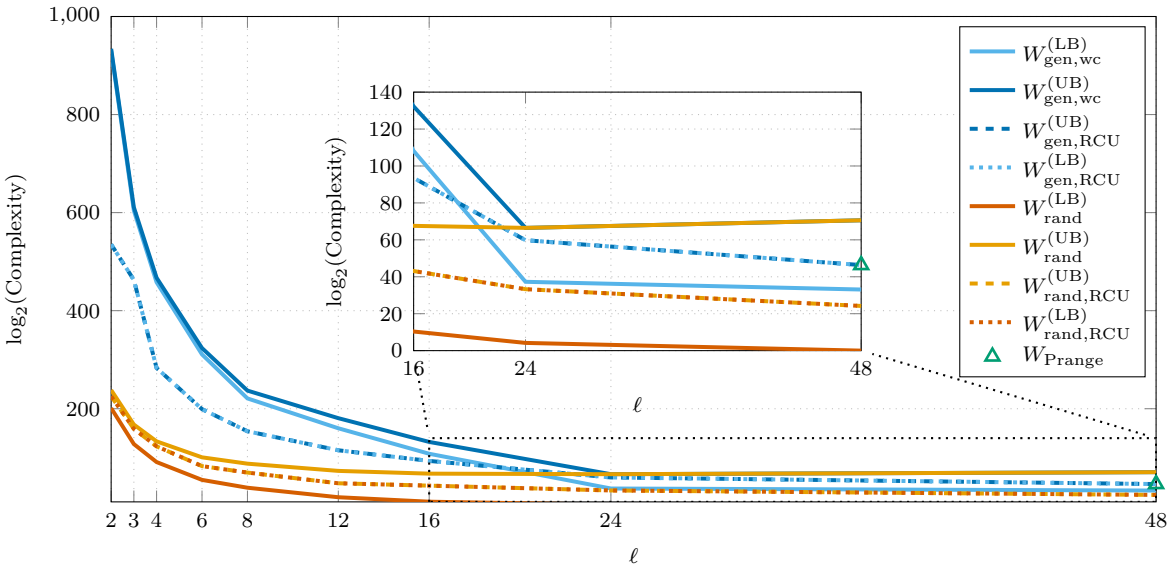


Figure 6.9: Complexity comparison of generic decoding vs. randomized decoding beyond the unique decoding radius for parameters: $q = 2^6$, $m = 24$, $n = 48$, $k = 24$, $w = 13$, $u = 2$, $v = 24$.

The results show a significant reduction in complexity for the randomized decoding algorithm compared to the generic decoder across both parameter sets. This improvement highlights the advantage of leveraging the structural properties of LRS codes. Although the relative gain decreases as the parameters approach those of the Hamming metric, the randomized decoder still maintains a lower complexity in both the worst-case and average-case settings.

6.5.7 Weak Keys in the Faure–Loidreau Cryptosystem

In 2006, Faure and Loidreau proposed a cryptosystem [FL06] based on the problem of reconstructing linearized polynomials. The FL cryptosystem is the rank-metric analogue of the Augot–Finiasz cryptosystem [AF03], offering very small public keys (around 2KB for 80-bit security).

In 2018, Gaborit et al. demonstrated that the private key of the FL cryptosystem could be recovered from the public key in polynomial time with high probability [GOK18]. Later, it was shown in [WPR18] that this attack is equivalent to list decoding interleaved Gabidulin codes [Loi06]. In this context, the private key corresponds to a noisy codeword, with an error weight just beyond the unique decoding radius. Such noisy codewords can often be recovered via probabilistic unique decoding, as the list size returned by the decoder is typically one (as discussed in Section 2.4.2).

To repair the FL cryptosystem, Wachter-Zeh et al. [WPR18] proposed restricting to error patterns that cause the probabilistic unique decoder for interleaved Gabidulin codes to fail, mitigating the vulnerability exploited by Gaborit's attack. However, the system remains vulnerable if the decoding problem for Gabidulin codes can be solved beyond the unique decoding radius as discussed in this section. For the original system parameters, this vulnerability exists only slightly beyond the unique decoding radius.

In [JB19], we introduced an initial version of the randomized decoder, which served as a precursor to the more generalized approach presented in this section. This earlier decoder operates on a narrower notion of support and focused on identifying weak keys, corresponding to specific subsets of error patterns in the FL cryptosystem. We demonstrated that this decoder could solve the decoding problem for weak keys with a significant reduction in complexity, lowering the security level to approximately 2^{25} operations for 80-bit security. We also characterized these weak keys and showed that a key-recovery attack is feasible for the parameters suggested in [FL06; WPR18].

It is worth noting that the modified FL system [WPR18] was adapted in [RPW18] and later introduced as LIGA in [RPW21b]. RAMESSES [LLP19] is another FL-based system, that also emerged as a variant. However, despite these developments, both systems were ultimately compromised two years later by a message recovery attack [BC21] that fully breaks the security of the systems. Our work [JB19] in 2019 predates this attack, and at the time, these systems were still viable cryptographic contributions.

6.6 Summary and Discussion

In this chapter, we developed and analyzed algorithms to address the general sum-rank metric decoding problem, with a focus on both worst-case and average-case complexities. We derived a tighter upper bound for the results in [PRR22] and extended their work to the average-case scenario, with a particular focus on decoding beyond the unique decoding radius. Additionally, we improved the randomized decoding algorithm for LRS codes, building on our previous work [JBW23]. Furthermore, we introduced a Prange-like algorithm for the sum-rank metric that effectively handles larger error weights in the asymptotic setting, where $\ell \to \infty$.

Future research could adapt techniques that improved Prange's original ISD algorithm in the Hamming metric [Ste89; BLP11; MMT11; BJMM12] to the generic and randomized decoding algorithms in the sum-rank metric. Given the hybrid nature of the sum-rank metric, these methods may particularly benefit the Hamming-like error structure and reduce complexity, especially when ℓ is large.

Another direction is to apply improvements from generic decoding algorithms in the rank metric, such as those in [AGHT17], or extend algebraic techniques like [BBC+20] to the sum-rank metric.

These approaches could yield substantial complexity reductions for specific parameter regimes in the sum-rank metric.

A list decoding algorithm for Gabidulin codes based on Gröbner bases was introduced in [HK17], enabling error correction beyond the unique decoding radius. This approach can be easily adapted for LRS codes. However, as no upper bound on the list size is known, it is difficult to assess the overall complexity of the algorithm, making it challenging to compare with our approach. Establishing tighter complexity bounds for this algorithm remains an open problem and could be a promising direction for future research.

7
Concluding Remarks

This thesis has contributed to the field of decoding algorithms for codes in the sum-rank metric, with an emphasis on improving decoding efficiency and analyzing complexities.

Chapter 3 revisited established decoding concepts for ILRS codes, presenting a fast Skew Kötter–Nielsen–Høholdt interpolation algorithm that addresses interpolation-based decoding of LRS codes. This is particularly relevant for the decryption process of potential code-based cryptosystems using ILRS codes.

In Chapter 4, we focused on decoding rank metric errors in Gabidulin codes that are space-symmetric. By relaxing the symmetry conditions, we demonstrated that errors with rank up to $\frac{2(n-k)}{3}$ can be decoded with high probability, thus extending the decoding capabilities of Gabidulin codes for these error types.

Chapter 5 introduced a novel approach for decoding high-order interleaved sum-rank-metric codes, generalizing the Metzner–Kapturowski algorithm to the sum-rank metric. The proposed decoder operates efficiently across a variety of linear codes, correcting errors of sum-rank weight up to $d_{\min} - 2$.

Chapter 6 advanced the complexity analysis of sum-rank metric decoding problems by transitioning from worst-case to average-case scenarios. We explored support-guessing algorithms and proposed a randomized decoder for LRS codes. Additionally, a new heuristic for optimizing the support-drawing distribution was introduced to minimize the average-case decoding complexity.

At the end of each chapter, we have provided future research directions and outlined open problems, offering valuable insights and potential paths for further exploration in decoding and cryptographic applications.

A
Proofs

A.1 Proofs of Chapter 4

Define $\boldsymbol{B} \stackrel{\text{def}}{=} \boldsymbol{P}\boldsymbol{A}^\top$ and $\boldsymbol{C} \stackrel{\text{def}}{=} \boldsymbol{P}^\top\boldsymbol{A}^\top$. Thus $\boldsymbol{E} = \boldsymbol{A}\boldsymbol{B}$ and $\boldsymbol{E}^\top = \boldsymbol{A}\boldsymbol{C}$. The vector representation $\boldsymbol{e}$ of $\boldsymbol{E}$ and its transposed $\hat{\boldsymbol{e}}$ of $\boldsymbol{E}^\top$ can therefore be written as

$$\begin{aligned}\boldsymbol{e} &= \boldsymbol{\alpha}\boldsymbol{E} = \boldsymbol{\alpha}\boldsymbol{A}\boldsymbol{B} = \boldsymbol{a}\boldsymbol{B},\\ \hat{\boldsymbol{e}} &= \boldsymbol{\alpha}\boldsymbol{E}^\top = \boldsymbol{\alpha}\boldsymbol{A}\boldsymbol{C} = \boldsymbol{a}\boldsymbol{C},\end{aligned}$$

with $\boldsymbol{a} = \boldsymbol{\alpha}\boldsymbol{A}$.

From (4.4) and (4.5) follows for all $j \in \{1, \ldots, n-k\}$ that

$$s_j^{(1)} = \sum_{i=1}^{n}\sum_{l=1}^{t} a_l C_{l,i}\alpha_i^{[j]} = \sum_{l=1}^{t} a_l \hat{c}_l^{[j]}, \tag{A.1}$$

$$s_j^{(2)} = \sum_{i=1}^{n}\sum_{l=1}^{t} a_l B_{l,i}\alpha_i^{[k+j-1]} = \sum_{l=1}^{t} a_l \hat{b}_l^{[k+j-1]}, \tag{A.2}$$

with $\hat{c}_l$ being the l-th entry of the vector $\hat{\boldsymbol{c}} = \boldsymbol{\alpha}\boldsymbol{C}^\top$ and $\hat{b}_l$ of $\hat{\boldsymbol{b}} = \boldsymbol{\alpha}\boldsymbol{B}^\top$, respectively.

A.1.1 Proof of the Key Equations

Proof. According to [Ore33a] the p-th coefficient of $\Omega^{(i)} = \Gamma(s^{(i)}(x))$ for $i \in \{1, 2\}$ can be computed as

$$\Omega_p^{(i)} = \sum_{j=1}^{p} \Gamma_j \left(s_{p-j+1}^{(i)}\right)^{[j-1]},$$

with $\Gamma_j = 0$ for $j > \deg_q(\Gamma(x))$ and $s_j^{(i)} = 0$ for $j > \deg_q(s^{(i)}(x))$.

From (A.1) and (A.2) it follows that

$$\Omega_p^{(1)} = \sum_{j=1}^{p} \Gamma_j \left(\sum_{l=1}^{t} a_l \hat{c}_l^{[p-j+1]} \right)^{[j-1]} = \sum_{l=1}^{t} \hat{c}_l^{[p]} \sum_{j=1}^{p} \Gamma_j a_l^{[j-1]},$$

and

$$\Omega_p^{(2)} = \sum_{j=1}^{p} \Gamma_j \left(\sum_{l=1}^{t} a_l \hat{b}_l^{[k+p-j]} \right)^{[j-1]} = \sum_{l=1}^{t} \hat{b}_l^{[p+k-1]} \sum_{j=1}^{p} \Gamma_j a_l^{[j-1]}.$$

For any $p \geq t$ this gives $\Omega_p^{(i)} = 0$, since $\Gamma(a_l) = \sum_{j=1}^{t} \Gamma_j a_l^{[j-1]} = 0$ by definition and therefore $\deg_q \left(\Omega^{(i)}(x) \right) < \deg_q \left(\Gamma(x) \right) = t$ for $i \in \{1, 2\}$. □

A.1.2 Derivation of Equation 4.8 and Equation 4.9

Using (A.1) and (A.2) we can decompose (4.6) as

$$\boldsymbol{S}^{(1)} = \begin{bmatrix} \hat{c}_1^{[t+1]} & \hat{c}_2^{[t+1]} & \dots & \hat{c}_t^{[t+1]} \\ \hat{c}_1^{[t+2]} & \hat{c}_2^{[t+2]} & \dots & \hat{c}_t^{[t+2]} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{c}_1^{[n-k]} & \hat{c}_2^{[n-k]} & \dots & \hat{c}_t^{[n-k]} \end{bmatrix} \cdot \boldsymbol{M}_{t+1}(\boldsymbol{a})^\top,$$

and

$$\mathbf{S}^{(2)} = \begin{bmatrix} \hat{b}_1^{[t+k]} & \hat{b}_2^{[t+k]} & \dots & \hat{b}_t^{[t+k]} \\ \hat{b}_1^{[t+k+1]} & \hat{b}_2^{[t+k+1]} & \dots & \hat{b}_t^{[t+k+1]} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{b}_1^{[n-1]} & \hat{b}_2^{[n-1]} & \dots & \hat{b}_t^{[n-1]} \end{bmatrix} \cdot \boldsymbol{M}_{t+1}(\boldsymbol{a})^\top.$$

The left-hand sides can be decomposed accordingly to the definition of $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{b}}$ and we have for $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$

$$\mathbf{S}^{(1)} = \begin{bmatrix} \alpha_1^{[t+1]} & \alpha_2^{[t+1]} & \dots & \alpha_n^{[t+1]} \\ \alpha_1^{[t+2]} & \alpha_2^{[t+2]} & \dots & \alpha_n^{[t+2]} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{[n-k]} & \alpha_2^{[n-k]} & \dots & \alpha_n^{[n-k]} \end{bmatrix} \cdot \mathbf{C}^\top \cdot \boldsymbol{M}_{t+1}(\boldsymbol{a})^\top,$$

$$\mathbf{S}^{(2)} = \begin{bmatrix} \alpha_1^{[t+k]} & \alpha_2^{[t+k]} & \dots & \alpha_n^{[t+k]} \\ \alpha_1^{[t+k+1]} & \alpha_2^{[t+k+1]} & \dots & \alpha_n^{[t+k+1]} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{[n-1]} & \alpha_2^{[n-1]} & \dots & \alpha_n^{[n-1]} \end{bmatrix} \cdot \mathbf{B}^\top \cdot \boldsymbol{M}_{t+1}(\boldsymbol{a})^\top.$$

Since $\boldsymbol{C}^\top = \boldsymbol{A}\boldsymbol{P}$, $\boldsymbol{B}^\top = \boldsymbol{A}\boldsymbol{P}^\top$ and $\boldsymbol{a} = \boldsymbol{\alpha}\boldsymbol{A}$ we obtain (4.8) and (4.9).

A.1.3 Recovering the Error Matrix

In the following, we describe the process of determining $\boldsymbol{B}$ such that $\boldsymbol{e} = \boldsymbol{a}\boldsymbol{B}$.

Let us define $d_l \stackrel{\text{def}}{=} \hat{b}_l^{[k]}$. From equation (A.2), we have

$$s_j^{(2)} = \sum_{l=1}^{t} a_l d_l^{[j-1]}.$$

Given the vector $\boldsymbol{a} = [a_1, a_2, \ldots, a_t]$, we can solve for $\boldsymbol{d} = [d_1, d_2, \ldots, d_t]$ using the following linear system of equations

$$\begin{bmatrix} a_1^{[0]} & a_2^{[0]} & \cdots & a_t^{[0]} \\ a_1^{[-1]} & a_2^{[-1]} & \cdots & a_t^{[-1]} \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{[-v]} & a_2^{[-v]} & \cdots & a_t^{[-v]} \end{bmatrix} \cdot \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_t \end{bmatrix} = \begin{bmatrix} (s_1^{(2)})^{[0]} \\ (s_2^{(2)})^{[-1]} \\ \vdots \\ (s_{v+1}^{(2)})^{[-v]} \end{bmatrix},$$

with $v = n - k - 1$. It remains to find $\boldsymbol{B}$ such that $d_l = \sum_{j=0}^{n-1} B_{l,j}\alpha_j^{[k]}$.

A.2 Proofs of Chapter 5

A.2.1 Proof of Theorem 5.1

Proof. First, partition $\boldsymbol{H}_{\mathcal{S}}$ into blocks according to the length partition $\boldsymbol{n}$, i.e.,

$$\boldsymbol{H}_{\mathcal{S}} = \left[\boldsymbol{H}_{\mathcal{S}}^{(1)} \mid \cdots \mid \boldsymbol{H}_{\mathcal{S}}^{(\ell)}\right],$$

with $\boldsymbol{H}_{\mathcal{S}}^{(i)} \in \mathbb{F}_{q^m}^{(n-k-w)\times n_i}$ for all $i \in \{1, \ldots, \ell\}$.

We want to show that $\operatorname{supp}_{\Sigma R}^{\perp}(\boldsymbol{H}_{\mathcal{S}}) = \operatorname{supp}_{\Sigma R}(\boldsymbol{E})$. By the definition of the support for the sum-rank metric, this means that we need to show that

$$\operatorname{supp}_R^{\perp}(\boldsymbol{H}_{\mathcal{S}}^{(i)}) = \operatorname{supp}_R(\boldsymbol{E}^{(i)}) = \mathcal{R}_q\left(\boldsymbol{B}^{(i)}\right) \quad \forall i \in \{1, \ldots, \ell\}.$$

Define $\mu_i \stackrel{\text{def}}{=} \operatorname{rk_q}(\boldsymbol{H}_{\mathcal{S}}^{(i)})$ for all $i \in \{1, \ldots, \ell\}$. Then, $\boldsymbol{H}_{\mathcal{S}}^{(i)}$ can be decomposed as

$$\boldsymbol{H}_{\mathcal{S}}^{(i)} = \boldsymbol{C}_{\mathcal{S}}^{(i)}\boldsymbol{D}_{\mathcal{S}}^{(i)},$$

with $\boldsymbol{C}_{\mathcal{S}}^{(i)} \in \mathbb{F}_{q^m}^{(n-k-w)\times\mu_i}$, $\boldsymbol{D}_{\mathcal{S}}^{(i)} \in \mathbb{F}_q^{\mu_i\times n_i}$, and $\operatorname{rk_q}(\boldsymbol{C}_{\mathcal{S}}^{(i)}) = \operatorname{rk_q}(\boldsymbol{D}_{\mathcal{S}}^{(i)}) = \mu_i$.

Recall from the definition of the sum-rank support (5.1) and its dual support (5.7) that we have

$$\mathrm{supp}_{\Sigma R}^{\perp}(\boldsymbol{H}_{\mathcal{S}}) = \mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(1)}\Big)^{\perp} \times \cdots \times \mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(\ell)}\Big)^{\perp},$$

and

$$\mathrm{supp}_{\Sigma R}(\boldsymbol{E}) = \mathcal{R}_q\Big(\boldsymbol{B}^{(1)}\Big) \times \cdots \times \mathcal{R}_q\Big(\boldsymbol{B}^{(\ell)}\Big),$$

respectively. The goal is to show that

$$\mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(i)}\Big)^{\perp} = \mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big),$$

for all $i \in \{1, \ldots, \ell\}$, which is equivalent to proving

$$\mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(i)}\Big) = \mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big)^{\perp}.$$

This will be achieved in two steps:

1. Show that $\mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(i)}\Big) \subseteq \mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big)^{\perp}$ for all $i \in \{1, \ldots, \ell\}$.
2. Show that $\mu_i < \dim\big(\mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big)^{\perp}\big) = n_i - w_i$ is not possible for any $i \in \{1, \ldots, \ell\}$, implying $\mu_i = n_i - w_i$ and hence $\mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(i)}\Big) = \mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big)^{\perp}$ for all $i \in \{1, \ldots, \ell\}$.

Step 1: Proving $\mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(i)}\Big) \subseteq \mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big)^{\perp}$ for all $i \in \{1, \ldots, \ell\}$.
To prove

$$\mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(i)}\Big) \subseteq \mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big)^{\perp},$$

we instead show that

$$\mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big) \subseteq \mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(i)}\Big)^{\perp}.$$

By definition, $\boldsymbol{H}_{\mathcal{S}}$ is a parity-check matrix for $\mathcal{S} = \mathcal{E} + \mathcal{C}$. Thus,

$$\boldsymbol{H}_{\mathcal{S}}\boldsymbol{G}_{\mathcal{E}}^{\top} = \boldsymbol{0} \quad \Leftrightarrow \quad \boldsymbol{H}_{\mathcal{S}}\boldsymbol{B}^{\top}\boldsymbol{A}_{\mathcal{E}}^{\top} = \boldsymbol{0},$$

where $\boldsymbol{G}_{\mathcal{E}}$ is the generator matrix of the error code as defined in (5.3). Since $\boldsymbol{A}_{\mathcal{E}} \in \mathbb{F}_{q^m}^{w \times w}$ is non-singular, we have that

$$\boldsymbol{H}_{\mathcal{S}}\boldsymbol{B}^{\top} = \boldsymbol{0} \Leftrightarrow \boldsymbol{H}_{\mathcal{S}}^{(i)}{\boldsymbol{B}^{(i)}}^{\top} = \boldsymbol{0} \quad \forall i \in \{1, \ldots, \ell\}. \tag{A.3}$$

This implies that all rows of $\boldsymbol{B}^{(i)}$ are in the $\mathbb{F}_{q^m}$-right kernel of $\boldsymbol{H}_{\mathcal{S}}^{(i)}$, and since $\boldsymbol{B}^{(i)}$ is over $\mathbb{F}_q$, we have that

$$\mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big) \subseteq \mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(i)}\Big)^{\perp}.$$

Consequently,

$$\mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(i)}\Big) \subseteq \mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big)^{\perp}.$$

Step 2: Showing that

$$\mu_i < \dim\left(\mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big)^{\perp}\right) = n_i - w_i,$$

is impossible for any $i \in \{1, \ldots, \ell\}$. Since

$$\mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(i)}\Big) \subseteq \mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big)^{\perp},$$

with $\mu_i > n_i - w_i$ is not possible for any $i \in \{1, \ldots, \ell\}$. Assume that $\mu_{i'} < n_{i'} - w_{i'}$ for at least one $i' \in \{1, \ldots, \ell\}$, i.e., let $\mu_{i'} = n_{i'} - w_{i'} - \delta \in \mathbb{Z}$ with $\delta > 0$. Without loss of generality, set $i' = \ell$.

Given that $\mathrm{rk_q}\left(\boldsymbol{D}_{\mathcal{S}}^{(\ell)}\right) = n_\ell - w_\ell - \delta$, there exists a full-rank matrix $\boldsymbol{Q}^{(\ell)} \in \mathbb{F}_q^{n_\ell \times n_\ell}$ that allows us to bring $\boldsymbol{D}_{\mathcal{S}}^{(\ell)}$ into column-echelon form. Hence,

$$\boldsymbol{D}_{\mathcal{S}}^{(\ell)}\boldsymbol{Q}^{(\ell)} = \Big[\ \underbrace{\boldsymbol{0}}_{\in \mathbb{F}_q^{(n_\ell - w_\ell - \delta)\times(w_\ell+\delta)}} \Big|\ \widetilde{\boldsymbol{D}}_{\mathcal{S}}^{(\ell)}\ \Big],$$

where $\widetilde{\boldsymbol{D}}_{\mathcal{S}}^{(\ell)} \in \mathbb{F}_q^{(n_\ell - w_\ell - \delta)\times(n_\ell - w_\ell - \delta)}$ with $\mathrm{rk_q}\left(\widetilde{\boldsymbol{D}}_{\mathcal{S}}^{(\ell)}\right) = n_\ell - w_\ell - \delta$.

Further, let

$$\boldsymbol{Q}^{(\ell)} = [\boldsymbol{Q}_1^{(\ell)} \mid \boldsymbol{Q}_2^{(\ell)}],$$

with $\boldsymbol{Q}_1^{(\ell)} \in \mathbb{F}_q^{n_\ell \times (w_\ell+\delta)}$ and $\boldsymbol{Q}_2^{(\ell)} \in \mathbb{F}_q^{n_\ell \times (n_\ell - w_\ell - \delta)}$. Since $\boldsymbol{Q}^{(\ell)}$ is full-rank, we have that $\boldsymbol{Q}_1^{(\ell)}$ is full-rank too, i.e., $\mathrm{rk_q}\left(\boldsymbol{Q}_1^{(\ell)}\right) = w_\ell + \delta$. Thus,

$$\boldsymbol{D}_{\mathcal{S}}^{(\ell)}\boldsymbol{Q}_1^{(\ell)} = \boldsymbol{0}. \tag{A.4}$$

That means we can multiply (A.4) from the right with some full-rank transformation matrix $\boldsymbol{T} \in \mathbb{F}_q^{(w_\ell+\delta)\times(w_\ell+\delta)}$ such that

$$\boldsymbol{D}_{\mathcal{S}}^{(\ell)} \underbrace{\left[\boldsymbol{B}^{(\ell)\top} \mid \widetilde{\boldsymbol{B}}^{(\ell)\top}\right]}_{=\boldsymbol{Q}_1^{(\ell)}\boldsymbol{T}} = \boldsymbol{0}. \tag{A.5}$$

Define the following block-diagonal matrix

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{B}^{(1)} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}^{(2)} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{B}^{(\ell)} \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \widetilde{\boldsymbol{B}}^{(\ell)} \end{bmatrix} \in \mathbb{F}_q^{(w+\delta)\times n}.$$

Then we have that

$$\boldsymbol{D}_{\mathcal{S}}\boldsymbol{Q}^{\top} = \boldsymbol{0}, \tag{A.6}$$

since

$$\boldsymbol{D}_{\mathcal{S}}^{(i)}\boldsymbol{B}^{(i)^{\top}} = \boldsymbol{0},$$

for $i \in \{1, \ldots, \ell - 1\}$ and by assumption (A.5),

$$\boldsymbol{D}_{\mathcal{S}}^{(\ell)}\left[\boldsymbol{B}^{(\ell)\top} \mid \widetilde{\boldsymbol{B}}^{(\ell)\top}\right] = \boldsymbol{0}.$$

Now, without loss of generality, let $\delta = 1$. By the decoding condition (5.5), we have that

$$\mathrm{rk}_{\mathrm{q^m}}\left(\boldsymbol{H}\boldsymbol{Q}^{\top}\right) = w + 1,$$

must hold. Thus, there exists a vector $\boldsymbol{g} \in \mathcal{R}_{q^m}(\boldsymbol{H})$ such that

$$\boldsymbol{g}\boldsymbol{Q}^{\top} = \begin{bmatrix} 0 & \ldots & 0 & g_{w+1} \end{bmatrix} \neq \begin{bmatrix} 0 & \ldots & 0 \end{bmatrix} \in \mathbb{F}_{q^m}^{w+1}.$$

Since the first w leftmost positions of $\boldsymbol{g}\boldsymbol{Q}^{\top}$ are zero, by (A.3) and the fact that the matrix formed by the w leftmost columns in $\boldsymbol{Q}^{\top}$ forms a basis of all $\mathcal{R}_q\left(\boldsymbol{B}^{(i)}\right)^{\perp}$ for all $i \in \{1, \ldots, \ell\}$, which are also bases for $\mathcal{R}_{q^m}\left(\boldsymbol{B}^{(i)}\right)^{\perp}$ for all $i \in \{1, \ldots, \ell\}$, this implies that $\boldsymbol{g} \in \mathcal{R}_{q^m}(\boldsymbol{B})^{\perp}$.

Also recall that $\boldsymbol{H}_{\mathcal{S}}$ fulfills the parity-check constraints for both codes simultaneously: the error code $\mathcal{E}$ and the component code $\mathcal{C}$. That means that

$$\begin{aligned} \mathcal{S} = \mathcal{C} + \mathcal{E} &\Leftrightarrow \mathcal{S}^{\perp} = \mathcal{C}^{\perp} \cap \mathcal{E}^{\perp} \\ &\Leftrightarrow \mathcal{R}_{q^m}(\boldsymbol{H}_{\mathcal{S}}) = \mathcal{R}_{q^m}(\boldsymbol{H}) \cap \mathcal{R}_{q^m}(\boldsymbol{B})^{\perp}. \end{aligned}$$

Since for this specific $\boldsymbol{g}$ we have that $\boldsymbol{g} \in \mathcal{R}_{q^m}(\boldsymbol{H})$ and also $\boldsymbol{g} \in \mathcal{R}_{q^m}(\boldsymbol{B})^{\perp}$, it follows that $\boldsymbol{g} \in \mathcal{R}_{q^m}(\boldsymbol{H}_{\mathcal{S}})$. Expanding $\boldsymbol{g}$ over $\mathbb{F}_q$ also implies that there exists a vector $\boldsymbol{g}' \in \mathcal{R}_q(\boldsymbol{H}_{\mathcal{S}}) = \mathcal{R}_q(\boldsymbol{D}_{\mathcal{S}})$ such that

$$\boldsymbol{g}'\boldsymbol{Q}^{\top} = \begin{bmatrix} 0 & \ldots & 0 & g'_{w+1} \end{bmatrix} \neq \begin{bmatrix} 0 & \ldots & 0 \end{bmatrix} \in \mathbb{F}_{q}^{w+1}.$$

But by (A.6), for all $\boldsymbol{g}' \in \mathcal{R}_q(\boldsymbol{D}_{\mathcal{S}})$ we need to have that

$$\boldsymbol{g}'\boldsymbol{Q}^{\top} = \begin{bmatrix} 0 & \ldots & 0 & 0 \end{bmatrix} \in \mathbb{F}_{q}^{w+1}.$$

This constitutes a contradiction, and thus $\mu_\ell < n_\ell - w_\ell$ is not possible. This also holds for any other $i' \neq \ell$, and therefore $\mu_i < n_i - w_i$ is not possible for any $i \in \{1, \ldots, \ell\}$.

When $\delta = 1$, we obtain one additional zero column in $\boldsymbol{g}'\boldsymbol{Q}^{\top}$. Similarly, when $\delta = 2$, we get two additional zero columns. Since a contradiction arises for $\delta = 1$, it follows

that the assumption cannot hold for any $\delta > 1$ as well. For $\delta = 0$, we do not get a contradiction, and thus $\mu_i = n_i - w_i$ for all $i \in \{1, \ldots, \ell\}$ is the only valid option.

This proves that

$$\mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(i)}\Big) = \mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big)^{\perp},$$

for all $i \in \{1, \ldots, \ell\}$, and therefore

$$\mathcal{R}_q\Big(\boldsymbol{D}_{\mathcal{S}}^{(i)}\Big)^{\perp} = \mathcal{R}_q\Big(\boldsymbol{B}^{(i)}\Big),$$

for all $i \in \{1, \ldots, \ell\}$, hence

$$\operatorname{supp}_{\Sigma R}^{\perp}(\boldsymbol{H}_{\mathcal{S}}) = \operatorname{supp}_{\Sigma R}(\boldsymbol{E}).$$

□

B
Appendix of Chapter 6

B.1 Efficient computation of $B_{q,m,\eta}(w, v, \ell)$

The quantity $B_{q,m,\eta}(w, v, \ell)$ defined in (6.17) can also be computed recursively as follows

$$B_{q,m,\eta}(w,v,\ell) = \begin{cases} \alpha_v^{(\mathrm{m})}\, \mathrm{NM}_q\left(m,\eta,w\right)\mathsf{P}_{q,\mu}^{\subseteq}\left(w,v\right) & \text{if } \ell = 1 \\ \displaystyle\sum_{w'=0}^{\min\{\mu,w\}} \sum_{v'=w'}^{\min\{\mu,v\}} \alpha_{v'}^{(\mathrm{m})}\, \mathrm{NM}_q\left(m,\eta,w'\right)\mathsf{P}_{q,\mu}^{\subseteq}\left(w',v'\right) & \\ \qquad \cdot B_{q,m,\eta}(w-w', v-v', \ell-1) & \text{else} \end{cases} .$$

This expression can be computed in polynomial time using dynamic programming; see Algorithm 12.

Algorithm 12: Compute $B_{q,m,\eta}(w,v,\ell)$ in the sum (6.18) for a given v.

Input : Parameters: q, m, n, k, ℓ, w and v with $v \geq 0$
Output : Value of $B_{q,m,\eta}(w,v,\ell)$
Initialize: $N(v',w',\ell') = 0 \quad \forall w' \in \{0,\dots,w\},\ v' \in \{0,\dots,v\},\ \ell' \in \{0,\dots,\ell\}$

if $v < w$ **then**
  return 0
for $w' \in \{0,\dots,w\}$ **do**
  for $v' \in \{w',\dots,v\}$ **do**
    if $v' \leq \mu$ **then**
      $N(v',w',1) \leftarrow \alpha_{v'}^{(\mathrm{m})}\,\mathrm{NM}_q\,(m,\eta,w')\mathsf{P}_{q,\mu}^{\subseteq}\,(v',w')$
for $\ell' \in \{2,\dots,\ell\}$ **do**
  for $w' \in \{0,\dots,w\}$ **do**
    for $v' \in \{w',\dots,v\}$ **do**

$$N(v',w',\ell') \leftarrow \sum_{w''=0}^{\min\{\mu,w'\}} \sum_{v''=w''}^{\min\{\mu,v'\}} N(v'-v'',w'-w'',\ell'-1) \cdot \alpha_{v''}^{(\mathrm{m})}\,\mathrm{NM}_q\,(m,\eta,w'')\mathsf{P}_{q,\mu}^{\subseteq}\,(v'',w'')$$

return $N(v,w,\ell)$

B.2 Appendix for Section 6.5

B.2.1 Definition of ucomp and Proof of Correctness

The computation of $\mathrm{ucomp}_\mu(\boldsymbol{w}, u)$ is described in Algorithm 13. and Lemma B.1 proofs its correctness. The randomization step in Line 6 of Algorithm 13 is crucial to avoid bias towards specific positions, particularly when ℓ is large. This contrasts with a deterministic choice, which may lead to suboptimal results. In the Hamming case, where $\eta = 1$ and $n = \ell$, such randomization is essential for the effectiveness of Prange's generic decoder. However, our analysis does not explicitly take this randomness property into account and instead relies on $\phi'_\mu(\mathrm{ucomp}_\mu(\boldsymbol{w}, u), \boldsymbol{w})$, which is not randomized, despite ucomp_μ being a randomized function.

Algorithm 13: $\mathrm{ucomp}_\mu(\boldsymbol{w}, u)$

Input : $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}$ and $u \in \mathbb{Z}_{\geq 0}$ with $u = 2\xi \leq w$
Output : $\boldsymbol{u} \in \mathcal{T}_{u,\ell,\mu}$ such that $\boldsymbol{u} = \arg\max_{\boldsymbol{u}' \in \mathcal{T}_{u,\ell,\mu}} \phi'_\mu(\boldsymbol{u}', \boldsymbol{w})$

1 $\boldsymbol{u} = [u_1, \ldots, u_\ell] \leftarrow \boldsymbol{w}$
2 $\delta \leftarrow w - u$
3 **while** $\delta > 0$ **do**
4 $\quad \mathcal{J}_1 \leftarrow \{i \in \{1, \ldots, n\} : u_i > 0\}$
5 $\quad \mathcal{J}_2 \leftarrow \{i \in \mathcal{J}_1 : w_i = \min_{j \in \mathcal{J}_1}\{w_j\}\}$
6 $\quad \mathcal{J}_3 \leftarrow \{i \in \mathcal{J}_2 : u_i = \max_{j \in \mathcal{J}_2}\{u_j\}\}$
7 $\quad h \xleftarrow{\$} \mathcal{J}_3$
8 $\quad u_h \leftarrow u_h - 1$
9 $\quad \delta \leftarrow \delta - 1$
10 **return** $\boldsymbol{u}$

Lemma B.1. *Let $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}$ and let $u \leq w$. Then, $\boldsymbol{u} = \mathrm{ucomp}_\mu(\boldsymbol{w}, u)$, with ucomp_μ as in Algorithm 13, maximizes $\phi'_\mu(\boldsymbol{u}, \boldsymbol{w})$, i.e.,*

$$\phi'_\mu(\mathrm{ucomp}_\mu(\boldsymbol{w}, u), \boldsymbol{w}) = \max_{\boldsymbol{u} \in \mathcal{T}_{u,\ell,\mu}} \phi'_\mu(\boldsymbol{u}, \boldsymbol{w}).$$

Proof. By (6.38) we have that

$$\phi'_\mu(\boldsymbol{u}, \boldsymbol{w}) \stackrel{\text{def}}{=} \prod_{i=1}^{\ell} \mathsf{P}^{\subseteq}_{q,\mu}(u_i, w_i) = \frac{\begin{bmatrix} w_i \\ u_i \end{bmatrix}_q}{\begin{bmatrix} \mu \\ u_i \end{bmatrix}_q}.$$

The factor by what this expression is increased if we decrease u_i by 1 is

$$\begin{aligned}\frac{\begin{bmatrix} w_i \\ u_i-1 \end{bmatrix}_q \Big/ \begin{bmatrix} \mu \\ u_i-1 \end{bmatrix}_q}{\begin{bmatrix} w_i \\ u_i \end{bmatrix}_q \Big/ \begin{bmatrix} \mu \\ u_i \end{bmatrix}_q} &= \prod_{j=1}^{u_i} \frac{q^{\mu-u_i+j}-1}{q^{w_i-u_i+j}-1} \cdot \prod_{j=1}^{u-1} \frac{q^{w_i-u_i+1+j}-1}{q^{\mu-u_i+1+j}-1} \\ &= \frac{q^\mu-1}{q^{w_i}-1} \cdot \underbrace{\prod_{i=j}^{u_i-1} \frac{(q^{w_i-u_i+1+j}-1)(q^{\mu-u_i+j}-1)}{(q^{\mu-u_i+1+j}-1)(q^{w_i-u_i+j}-1)}}_{=\frac{(q^{w_i}-1)(q^{\mu+1}-q^{u_i})}{(q^{\mu}-1)(q^{w_i+1}-q^{u_i})}} \\ &= \frac{q^{\mu+1}-q^{u_i}}{q^{w_i+1}-q^{u_i}}.\end{aligned}$$

This increase factor is monotonically increasing in u_i for a fixed w_i and μ, and decreasing in w_i for a fixed u_i. Consequently, the maximum increase of (6.38) is obtained by decreasing the largest u_i among the smallest w_i. By adopting a greedy approach and incrementally adjusting such positions, a global maximum can be reached as this strategy ensures optimal increase in subsequent steps. Thus, (6.38) is optimized by incrementally decreasing u_i by one while maintaining $u_i \geq 0$ and ensuring $\sum_{i=1}^{\ell} u_i \geq u$. This method aligns with the operations performed by $\mathrm{ucomp}_\mu(\boldsymbol{w}, u)$ as described in Algorithm 13. □

B.2.2 Efficient Computation of $\tilde{Q}_{\ell,w,\mu}$

Fortunately, we can employ a similar approach to [PRR22, Lemma 22] and compute $\tilde{Q}_{\ell,w,\mu}$ as

$$\tilde{Q}_{\ell,w,\mu} = \ell! \cdot M(w, \ell, \mu, u),$$

where $M(w, \ell, \mu, u)$ can be computed using Algorithm 14. To do so, we first initialize a global table $\{M(w', \ell', \mu', u')\}_{w' \leq w, \ell' \leq \ell}^{\mu' \leq \mu, u' \leq u}$ with $M(w', \ell', \mu', u') = -1$ for all entries. Then, we call Algorithm 14 with input parameters w, ℓ, μ, and u.

By applying arguments similar to those in [PRR22, Proposition 23], we can show that the complexity of computing $\tilde{Q}_{\ell,w,\mu}$ using this approach is $\widetilde{O}(wun^3\mu^3 \log_2(q))$ and thus polynomially bounded.

Algorithm 14: Fill Table $\{M(w', \ell', \mu', u')\}_{w' \le w, \ell' \le \ell}^{\mu' \le \mu, u' \le u}$

Input : Integers $w' \le w$, $\ell' \le \ell$, $\mu' \le \mu$, $u' \le u$
Global table $\{M(w', \ell', \mu', u')\}_{w' \le w, \ell' \le \ell}^{\mu' \le \mu, u' \le u}$
Global parameters q and μ

Output : $M(w', \ell', \mu', u')$

if $M(w', \ell', \mu', u') = -1$ **then**
  if $\ell' = w' = u' = 0$ **then**
    $x \leftarrow 1$
  else
    if $\ell' \ge 1$ **and** $0 \le w' \le \ell'\mu'$ **and** $0 \le u' \le \min\{\ell'\mu, w'\}$ **then**
      $x \leftarrow 0$
      for $w_1 \in \{\mu, \dots, \lfloor w/\ell \rfloor\}$ **do**
        $\delta_{\min} \leftarrow \max\{1, \ell'(w_1 + 1) - w)\}$
        $\delta_{\max} \leftarrow \max\{i \in \mathbb{Z}_{\ge 0} : 1 \le i \le \ell' + 1, w_1 i \le w\}$
        for $\delta \in \{\delta_{\min}, \dots, \delta_{\max}\}$ **do**
          $u_1 \leftarrow \max\{u' - (w' - \delta w_1), 0\}$
          $\boldsymbol{u}^{(1)} \leftarrow \mathrm{ucomp}_{\mu}([w_1, \dots, w_1], u_1)$
          $\rho \leftarrow \frac{1}{\delta!} \cdot \prod_{i=1}^{\delta} \left(\begin{bmatrix} \mu \\ u_i^{(1)} \end{bmatrix}_q \cdot \begin{bmatrix} w_1 \\ u_i^{(1)} \end{bmatrix}_q^{-1} \right)$
          $x \leftarrow x + \rho \cdot M(w' - \delta w_1, \ell' - \delta, w_1 + 1, u' - u_1)$
    else
      $x \leftarrow 0$
return $M(w', \ell', \mu', u')$

C

Notations, Variables, and Abbreviations

Below, we provide a comprehensive list of the notations, variables, and abbreviations used throughout this dissertation. Note that notation and variables defined only in specific contexts or chapters are not included here.

Abbreviations

AES	Advanced Encryption Standard
AWGN	additive white Gaussian noise
BCH	Bose–Chaudhuri–Hocquenghem
BMD	bounded minimum distance
BJMM	Becker–Joux–May–Meurer algorithm
BSC	binary symmetric channel
CA	certificate authority
D&C	divide-and-conquer
DES	Data Encryption Standard
DS	Digital Signature
ECC	Elliptic Curve Cryptography
FL	Faure–Loidreau
GPT	Gabidulin–Paramonov–Tretjakov
GRS	generalized Reed–Solomon
ILRS	interleaved linearized Reed–Solomon
IRS	interleaved Reed–Solomon
ISD	information-set decoding

KEM	key-encapsulation mechanism
KNH	Kötter–Nielsen–Høholdt
lclm	least-common left multiple
LDPC	low-density parity-check
LP	linear program
LRS	linearized Reed–Solomon
LRPC	low-rank parity-check
ML	maximum-likelihood
MMT	May–Meurer–Thomae algorithm
MSRD	maximum sum-rank distance
PKI	public-key infrastructure
PMF	probability mass function
REF	row echelon form
RCU	random-coding union
RS	Reed–Solomon
RSA	Rivest–Shamir–Adleman
SL	security level
TOP	term-over-position
WF	work factor

Acronyms

BIKE	Bit Flipping Key Encapsulation
HTTPS	HyperText Transfer Protocol Secure
HQC	Hamming Quasi-Cyclic
LIGA	List-decoding and Interleaved Gabidulin Approach
ML-DSA	Module-Lattice Digital Signature Algorithm
ML-KEM	Module-Lattice Key Encapsulation Mechanism
NIST	National Institute of Standards and Technology
PGP	Pretty Good Privacy
RAMESSES	Rank-Metric Encryption Scheme with Short Keys
RQC	Rank Quasi-Cyclic
SIKE	Supersingular Isogeny Key Encapsulation
SLH-DSA	Stateless Hash-based Digital Signature Algorithm
SSL	Secure Sockets Layer
TLS	Transport Layer Security

Basics

Id	Identity map
$\mathbb{Z}$	Set of all integers
$\mathbb{Z}_{\geq 0}$	Set of nonnegative integer $\mathbb{Z}_{\geq 0} = \{0, 1, 2, \ldots\}$
$\mathbb{R}$	Set of real numbers
$a \xleftarrow{\$} \mathcal{A}$	Indicates that a is sampled uniformly at random from the set $\mathcal{A}$
$\mathbb{E}[X]$	The expected value (or expectation) of the random variable X
ζ	Matrix multiplication exponent, the infimum of values $\zeta_0 \in [2, 3]$ for which an algorithm exists to multiply $n \times n$ matrices over $\mathbb{F}_{q^m}$ in $O(n^{\zeta_0})$ operations
$\mathfrak{p}(n)$	Cost of multiplying two skew polynomials from $\mathbb{F}_{q^m}[x; \sigma]$ of degree n

Finite Fields, Matrices, Sets and Vector Spaces

$\boldsymbol{A}_{[e:f]}$	Submatrix of $\boldsymbol{A}$ consisting of all rows and columns e to f
$\boldsymbol{A}^{-\top}$	Inverse of the transpose of a square matrix $\boldsymbol{A}$
q	Power of a prime
$\mathbb{F}_q$	Finite field of order q
$\mathbb{F}_{q^m}$	Finite extension field of $\mathbb{F}_q$ of degree m
m	Degree of a finite extension field $\mathbb{F}_{q^m}$
$\mathbb{F}^*$	Multiplicative group of a finite field $\mathbb{F}$
$\mathbb{F}^{m \times n}$	Set of all $m \times n$ matrices over a finite field $\mathbb{F}$
$\mathbb{F}^{1 \times n}$	Set of all row vectors of length n over a finite field $\mathbb{F}$
$\mathrm{ext}(\cdot)$	The expansion map, representing elements of $\mathbb{F}_{q^m}$ as column vectors over $\mathbb{F}_q$

$\mathrm{rk_q}\,(\boldsymbol{A})$	The $\mathbb{F}_q$-rank of a matrix $\boldsymbol{A}$ over the field $\mathbb{F}_{q^m}$ after expanding over $\mathbb{F}_q$ using ext$(\cdot)$
$\mathrm{rk_{q^m}}\,(\boldsymbol{A})$	The $\mathbb{F}_{q^m}$-rank of a matrix $\boldsymbol{A}$ over the field $\mathbb{F}_{q^m}$
$\mathrm{GL}_v(\mathbb{F})$	The general linear group consisting of all $v \times v$ invertible matrices over a finite field $\mathbb{F}$
$\mathcal{G}_k(\mathbb{F}_q^v)$	The Grassmannian, representing the set of all k-dimensional subspaces of the vector space $\mathbb{F}_q^v$ over the finite field $\mathbb{F}_q$
$\begin{bmatrix} a \\ b \end{bmatrix}_q$	Gaussian binomial coefficient
$\langle \boldsymbol{a}_1, \ldots, \boldsymbol{a}_v \rangle_q$	The $\mathbb{F}_q$-linear vector space spanned by the vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_v$
$\mathcal{V}^\perp$	The dual space of the vector space $\mathcal{V}$
dim	The dimension of a vector space
$\mathrm{dim_q}$	The dimension of an $\mathbb{F}_q$-linear vector space, specifically when expanding a vector space over $\mathbb{F}_{q^m}$ as an $\mathbb{F}_q$-linear space
$\mathrm{dim_{q^m}}$	The dimension of an $\mathbb{F}_{q^m}$-linear vector space
$\ker\,(\boldsymbol{A})_{\mathbb{F}_q}$	The right $\mathbb{F}_q$-kernel of the matrix $\boldsymbol{A}$
$\ker\,(\boldsymbol{A})_{\mathbb{F}_{q^m}}$	The right $\mathbb{F}_{q^m}$-kernel of the matrix $\boldsymbol{A}$
$\mathcal{R}_q(\boldsymbol{A})$	The $\mathbb{F}_q$-linear row space of the matrix $\boldsymbol{A} \in \mathbb{F}_{q^m}^{v \times w}$, obtained by expanding each element of $\boldsymbol{A}$ over $\mathbb{F}_q$ and considering the $\mathbb{F}_q$-linear span of the resulting rows
$\mathcal{C}_q(\boldsymbol{A})$	The $\mathbb{F}_q$-linear column space of the matrix $\boldsymbol{A} \in \mathbb{F}_{q^m}^{v \times w}$, obtained by expanding each element of $\boldsymbol{A}$ over $\mathbb{F}_q$ and considering the $\mathbb{F}_q$-linear span of the resulting columns
$\mathsf{P}^{\cap}_{q,\mu,a,b}(j)$	The conditional probability that the intersection of two subspaces of $\mathbb{F}_q^\mu$, with dimensions a and b, has dimension exactly j (see (2.7))
$\mathsf{P}^{\subseteq}_{q,\mu}\,(a,b)$	The probability that a subspace of $\mathbb{F}_q^\mu$ with dimension a is contained within another subspace of dimension b, where $a \leq b$ (see (2.8))
$\deg(f)$	The degree of a polynomial f (see (2.12))

Polynomials over Finite Fields

$\mathbb{F}_{q^m}[x;\sigma,\delta]$	The non-commutative ring of skew polynomials with automorphism $\sigma(\cdot)$ and derivation $\delta(\cdot)$, as defined in Section 2.5.2
$\mathbb{F}_{q^m}[x;\sigma,\delta]_{<k}$	The set of skew polynomials in $\mathbb{F}_{q^m}[x;\sigma,\delta]$ with degree less than k (see (2.14))
$\mathbb{L}_{q^m}[x]$	The linearized polynomial ring (see [Ore33a; Ore33b])
σ_{Frob}	The Frobenius automorphism, defined as $\sigma_{\text{Frob}}(x) = x^q$
$\mathfrak{M}_z(\boldsymbol{x})_{\boldsymbol{a}}$	The generalized Moore matrix, defined as in (2.16)
$\boldsymbol{M}_z(\boldsymbol{x})$	The Moore matrix, a special case of the generalized Moore matrix $\mathfrak{M}_z(\boldsymbol{x})_{\boldsymbol{a}}$ (see (2.17))
$\boldsymbol{V}_z(\boldsymbol{x})$	The Vandermonde matrix (see (2.18))

Linear Codes

n	Length of a code
k	Dimension of a code
R	Rate of a code $R = k/n$
$d_{\min}$	The minimum distance of a code, defined as the smallest distance between any two distinct codewords. The applicable metric (Hamming, Rank, or Sum-Rank) varies based on context
$\mathcal{C}[n,k,d_{\min}]$	Linear code of length n, dimension k and minimum distance $d_{\min}$ over $\mathbb{F}_{q^m}$
$\mathcal{IC}[s;n,k,d_{\min}]$	A vertically homogeneous s-interleaved code of length n, dimension k, and minimum distance $d_{\min}$ (see Definition 2.4)
s	Interleaving order
$\text{wt}_H(\boldsymbol{a})$	Hamming weight of vector $\boldsymbol{a}$, representing the number of non-zero elements in the vector
$d_H(\boldsymbol{a},\boldsymbol{b})$	Hamming distance between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, defined as the number of positions at which the corresponding elements differ

$\mathrm{wt}_R(\boldsymbol{a})$	Rank weight of a vector $\boldsymbol{a}$
w	Error weight
τ	Unique decoding radius
$\tau_{\mathcal{L}}$	List decoding radius

Sum-Rank Metric

ℓ	Number of blocks
η	Length of a single block. Used for constant block length. The overall length is then $n = \ell\eta$
μ_i	Maximum rank of block i (see (2.21))
μ	Maximum rank of a block in the case of constant block length (see (2.22))
$\mu_i^{(s)}$	Minimum of n_i and sm for block i (see (2.31))
$\mu^{(s)}$	Minimum of η and sm for the case of constant block lengths (see (2.32))
$\boldsymbol{n}$	Length profile $\boldsymbol{n} = [n_1, n_2, \ldots, n_\ell] \in \mathbb{Z}_{\geq 0}^{\ell}$
ψ	A map which assigns a vector $\boldsymbol{x}$ to its rank profile (see (2.23))
$\mathcal{T}_{w,\ell,\mu}$	Set of rank profiles with block weights in $\{0, \ldots, \mu\}$ summing to w (see Definition 2.10)
$\mathrm{NM}_q(m, \eta, w_i)$	Number of $m \times \eta$ matrices of rank w_i over $\mathbb{F}_q$ (see (2.26))
$\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{a})$	Sum-rank weight of a vector $\boldsymbol{a}$, defined as the sum of the rank weights of its blocks with respect to a length profile $\boldsymbol{n}$ (see (2.19))
$d_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{x}, \boldsymbol{y})$	Sum-rank distance between vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ (see (2.20))
$\mathcal{C}_{\Sigma R}[\boldsymbol{n}, k, d_{\min}]$	An $\mathbb{F}_{q^m}$-linear code in the sum-rank metric with length profile $\boldsymbol{n}$, dimension k, and minimum distance $d_{\min}$
$\mathcal{C}_{\Sigma R}[n, k, d_{\min}]$	An $\mathbb{F}_{q^m}$-linear code in the sum-rank metric with of length $n = \eta\ell$, dimension k, and minimum distance $d_{\min}$

$\mathcal{IC}_{\Sigma R}[s; \boldsymbol{n}, k, d_{\min}]$	Vertically interleaved code in the sum-rank metric with interleaving order s, length profile $\boldsymbol{n}$, dimension k, and minimum distance $d_{\min}$
$\mathrm{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell; \boldsymbol{n}, k]$	LRS code of length $n = \sum_{i=1}^{\ell} n_i$ and dimension k with conjugacy class representatives $\boldsymbol{\xi}$ and evaluation parameters $\boldsymbol{\beta}$ (see Definition 2.13)
$\mathrm{ILRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, \ell, s; \boldsymbol{n}, k]$	ILRS code of length $n = \sum_{i=1}^{\ell} n_i$ and dimension k, defined with conjugacy class representatives $\boldsymbol{\xi}$ and evaluation parameters $\boldsymbol{\beta}$ (see Definition 2.13)
$\mathrm{Gab}_{\boldsymbol{\alpha}}[n, k]$	Gabidulin code with evaluation parameters $\boldsymbol{\alpha}$, of length n and dimension k (see Definition 4.1)
$\mathrm{supp}_R^{(R)}(\boldsymbol{E})$	Row rank support of $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s \times n}$
$\mathrm{supp}_R^{(C)}(\boldsymbol{E})$	Column rank support of $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s \times n}$
$\mathrm{supp}_{\Sigma R}^{(R)}(\boldsymbol{E})$	Sum-rank row support of $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s \times n}$ (see (2.33))
$\mathrm{supp}_{\Sigma R}^{(C)}(\boldsymbol{E})$	Sum-rank column support of $\boldsymbol{E} \in \mathbb{F}_{q^m}^{s \times n}$ (see (2.34))
$\dim_{\Sigma}(\boldsymbol{\mathcal{E}})$	Sum dimension of the support $\boldsymbol{\mathcal{E}}$
$\boldsymbol{\mathcal{E}}_1 \cap \boldsymbol{\mathcal{E}}_2$	Intersection of two supports $\boldsymbol{\mathcal{E}}_1$ and $\boldsymbol{\mathcal{E}}_2$
$\mathbb{F}_q^{\boldsymbol{n}}$	Notation for $\mathbb{F}_q^{n_1} \times \mathbb{F}_q^{n_2} \times \cdots \times \mathbb{F}_q^{n_\ell}$ given a length profile $\boldsymbol{n} = [n_1, n_2, \ldots, n_\ell]$ (see (2.35))

Related Publications by the Author

[BJPR19] H. Bartz, T. Jerkovits, S. Puchinger, and J. Rosenkilde. "Fast Root Finding for Interpolation-Based Decoding of Interleaved Gabidulin Codes". In: *2019 IEEE Information Theory Workshop (ITW)*. IEEE Information Theory Workshop. Visby, Sweden: IEEE, Aug. 2019, pp. 1–5. ISBN: 978-1-5386-6900-6. DOI: 10.1109/ITW44776.2019.8989290.

[BJPR21] H. Bartz, T. Jerkovits, S. Puchinger, and J. Rosenkilde. "Fast Decoding of Codes in the Rank, Subspace, and Sum-Rank Metric". In: *IEEE Transactions on Information Theory* 67.8 (2021). ISSN: 15579654. DOI: 10.1109/TIT.2021.3067318.

[BJR22] H. Bartz, T. Jerkovits, and J. Rosenkilde. "Fast Kötter–Nielsen–Høholdt Interpolation Over Skew Polynomial Rings". In: *IFAC-PapersOnLine* 55.30 (2022), pp. 1–6. ISSN: 24058963. DOI: 10.1016/j.ifacol.2022.11.019.

[BJR24] H. Bartz, T. Jerkovits, and J. Rosenkilde. "Fast Kötter–Nielsen–Høholdt Interpolation over Skew Polynomial Rings and its Application in Coding Theory". In: *Designs, Codes and Cryptography* 92.2 (Feb. 2024), pp. 435–465. ISSN: 0925-1022, 1573-7586. DOI: 10.1007/s10623-023-01315-4.

[CJB24] H. S. Couvée, T. Jerkovits, and J. Bariffi. *Bounds on Sphere Sizes in the Sum-Rank Metric and Coordinate-Additive Metrics*. Version Number: 2 Submitted to Designs, Codes and Cryptography, Special Issue WCC2024. 2024. DOI: 10.48550/ARXIV.2404.10666.

[JB19] T. Jerkovits and H. Bartz. "Weak Keys in the Faure–Loidreau Cryptosystem". In: *Code-Based Cryptography*. Cham: Springer, 2019, pp. 102–114. ISBN: 978-3-030-25922-8.

[JBW23] T. Jerkovits, H. Bartz, and A. Wachter-Zeh. "Randomized Decoding of Linearized Reed–Solomon Codes Beyond the Unique Decoding Radius". In: *2023 IEEE International Symposium on Information Theory (ISIT)*. 2023, pp. 820–825. DOI: 10.1109/ISIT54713.2023.10206957.

[JBW24] T. Jerkovits, H. Bartz, and A. Wachter-Zeh. *Support-Guessing Decoding Algorithms in the Sum-Rank Metric*. Submitted to IEEE Transactions on Information Theory, Version Number: 1. 2024. DOI: 10.48550/ARXIV.2410.15806.

[JGSK20] T. Jerkovits, O. Günlü, V. Sidorenko, and G. Kramer. "Nested Tailbiting Convolutional Codes for Secrecy, Privacy, and Storage". In: *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*. IH&MMSEC '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 79–89. ISBN: 978-1-4503-7050-9. DOI: 10.1145/3369412.3395063.

[JHB23] T. Jerkovits, F. Hörmann, and H. Bartz. "On Decoding High-Order Interleaved Sum-Rank-Metric Codes". In: *Code-Based Cryptography*. Cham: Springer, 2023, pp. 90–109. ISBN: 978-3-031-29689-5.

[JHB24] T. Jerkovits, F. Hörmann, and H. Bartz. *An Error-Code Perspective on Metzner–Kapturowski-Like Decoders*. Submitted to IEEE Transactions on Information Theory, Version Number: 1. 2024. DOI: 10.48550/ARXIV.2409.18488.

[JLG18] T. Jerkovits, G. Liva, and A. Graell i Amat. "Improving the Decoding Threshold of Tailbiting Spatially Coupled LDPC Codes by Energy Shaping". In: *IEEE Communications Letters* 22.4 (2018). ISSN: 10897798. DOI: 10.1109/LCOMM.2018.2802488.

[JSW21] T. Jerkovits, V. Sidorenko, and A. Wachter-Zeh. "Decoding of Space-Symmetric Rank Errors". In: *IEEE International Symposium on Information Theory - Proceedings*. IEEE International Symposium on Information Theory (ISIT). Vol. 2021-July. ISSN: 21578095. 2021. ISBN: 978-1-5386-8209-8. DOI: 10.1109/ISIT45174.2021.9518115.

[RJB+20] 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: *Post-Quantum Cryptography*. Cham: Springer, 2020, pp. 3–19. ISBN: 978-3-030-44223-1.

[RJB19] J. Renner, T. Jerkovits, and H. Bartz. "Efficient Decoding of Interleaved Low-Rank Parity-Check Codes". In: *2019 16th International Symposium "Problems of Redundancy in Information and Control Systems", REDUNDANCY 2019*. 2019. ISBN: 978-1-72811-944-1. DOI: 10.1109/REDUNDANCY48165.2019.9003356.

Bibliography

[AAB+19] F. Arute et al. “Quantum Supremacy Using a Programmable Superconducting Processor”. In: *Nature* 574.7779 (Oct. 24, 2019), pp. 505–510. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-019-1666-5.

[ABB+20] N. Aragon et al. *BIKE: Bit Flipping Key Encapsulation*. Submission to NIST Post-Quantum Cryptography Standardization Process, Round 3. 2020.
URL: https://bikesuite.org/ (accessed on 06/20/2024).

[AF03] D. Augot and M. Finiasz. “A Public Key Encryption Scheme Based on the Polynomial Reconstruction Problem”. In: *LNCS: Revised Selected Papers of Eurocrypt 2003* 2656 (2003), pp. 229–249.

[AGHT17] N. Aragon, P. Gaborit, A. Hauteville, and J.-P. Tillich. “Improvement of Generic Attacks on the Rank Syndrome Decoding Problem”. Oct. 2017.
URL: https://hal.archives-ouvertes.fr/hal-01618464 (accessed on 02/29/2024).

[AL94] W. W. Adams and P. Loustaunau. *An Introduction to Gröbner Bases*. Graduate studies in mathematics 3. Providence (R.I.): American mathematical society, 1994. ISBN: 978-0-8218-3804-4.

[Ale02] M. Alekhnovich. “Linear Diophantine Equations Over Polynomials and Soft Decoding of Reed–Solomon Codes”. In: *The 43rd Annual IEEESymposium on Foundations of Computer Science, 2002. Proceedings.* IEEE, 2002, pp. 439–448.

[AW21] J. Alman and V. V. Williams. “A Refined Laser Method and Faster Matrix Multiplication”. In: *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*. Soda ’21. USA: Society for Industrial and Applied Mathematics, 2021, pp. 522–539. ISBN: 978-1-61197-646-5.

[Bar17] H. Bartz. “Algebraic Decoding of Subspace and Rank-Metric Codes”. ISBN: 978-3-8439-3174-8 Series: Informationstechnik Publisher: Dr. Hut Verlag. PhD thesis. Munich, Germany: Technical University of Munich, 2017.

[BBC+20] M. Bardet et al. “Improvements of Algebraic Attacks for Solving the Rank Decoding and MinRank Problems”. In: *Advances in cryptology – ASIACRYPT 2020*. Cham: Springer, 2020, pp. 507–536. ISBN: 978-3-030-64837-4.

[BC21] M. Bombar and A. Couvreur. “Decoding Supercodes of Gabidulin Codes and Applications to Cryptanalysis”. In: *Post-Quantum Cryptography*. Vol. 12841. Series Title: Lecture Notes in Computer Science. Cham: Springer, 2021, pp. 3–22. ISBN: 978-3-030-81292-8. DOI: 10.1007/978-3-030-81293-5_1.

[BCG+03] M. Bossert, E. Costa, E. M. Gabidulin, E. Schulz, and M. Weckerle. “Verfahren und Kommunikationsvorrichtung zum Dekodieren von mit einem Rang-Code codierten Daten”. European pat. 20040104458. 2003.

[BCGO09] T. P. Berger, P.-L. Cayrel, P. Gaborit, and A. Otmani. “Reducing Key Length of the McEliece Cryptosystem”. In: *Progress in Cryptology – AFRICACRYPT 2009*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2009, pp. 77–97. ISBN: 978-3-642-02384-2. DOI: 10.1007/978-3-642-02384-2_6.

[Ber84] E. R. Berlekamp. *Algebraic Coding Theory*. Revised. Paperback. Aegean Park Press, June 1, 1984. ISBN: 0-89412-063-8.

[BGR21] E. Byrne, H. Gluesing-Luerssen, and A. Ravagnani. “Fundamental Properties of Sum-Rank-Metric Codes”. In: *IEEE Transactions on Information Theory* 67.10 (2021). ISSN: 15579654. DOI: 10.1109/TIT.2021.3074190.

[BJMM12] A. Becker, A. Joux, A. May, and A. Meurer. “Decoding Random Binary Linear Codes in 2^(n/20): How 1 + 1 = 0 Improves Information Set Decoding”. In: *Advances in Cryptology – EUROCRYPT 2012*. Red. by D. Hutchison et al. Vol. 7237. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 520–536. ISBN: 978-3-642-29010-7. DOI: 10.1007/978-3-642-29011-4_31.

[BKY03] D. Bleichenbacher, A. Kiayias, and M. Yung. “Decoding of Interleaved Reed Solomon Codes over Noisy Data”. In: *Automata, Languages and Programming*. Red. by G. Goos, J. Hartmanis, and J. Van Leeuwen. Vol. 2719. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 97–108. ISBN: 978-3-540-40493-4. DOI: 10.1007/3-540-45061-0_9.

[BL05] T. P. Berger and P. Loidreau. “How to Mask the Structure of Codes for a Cryptographic Use”. In: *Designs, Codes and Cryptography* 35.1 (Apr. 1, 2005), pp. 63–79. ISSN: 1573-7586. DOI: 10.1007/s10623-003-6151-2.

[BLP11] D. J. Bernstein, T. Lange, and C. Peters. "Smaller Decoding Exponents: Ball-Collision Decoding". In: *Advances in Cryptology – CRYPTO 2011*. Red. by D. Hutchison et al. Vol. 6841. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2011, pp. 743–760. ISBN: 978-3-642-22791-2. DOI: 10.1007/978-3-642-22792-9_42.

[BMN+21] R. Babbush, J. R. McClean, M. Newman, C. Gidney, S. Boixo, and H. Neven. "Focus Beyond Quadratic Speedups for Error-Corrected Quantum Advantage". In: *PRX Quantum* 2.1 (Mar. 29, 2021). Publisher: American Physical Society, p. 010103. DOI: 10.1103/PRXQuantum.2.010103.

[BMS04] A. Brown, L. Minder, and A. Shokrollahi. "Probabilistic Decoding of Interleaved Rs-Codes on the Q-Ary Symmetric Channel". In: *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.* International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings. Chicago, Illinois, USA: IEEE, 2004, pp. 326–326. ISBN: 978-0-7803-8280-0. DOI: 10.1109/ISIT.2004.1365363.

[BMV78] E. Berlekamp, R. McEliece, and H. Van Tilborg. "On the Inherent Intractability of Certain Coding Problems (corresp.)" In: *IEEE Transactions on Information Theory* 24.3 (May 1978), pp. 384–386. ISSN: 0018-9448. DOI: 10.1109/TIT.1978.1055873.

[Bou20] D. Boucher. "An Algorithm for Decoding Skew Reed–Solomon Codes with Respect to the Skew Metric". In: *Designs, Codes and Cryptography* 88.9 (Sept. 2020), pp. 1991–2005. ISSN: 0925-1022, 1573-7586. DOI: 10.1007/s10623-020-00789-w.

[BP22] H. Bartz and S. Puchinger. *Fast Decoding of Interleaved Linearized Reed-Solomon Codes and Variants*. Version Number: 3, Submitted to Advances in Mathematics of Communications (AMC). 2022. DOI: 10.48550/ARXIV.2201.01339.

[Car19] X. Caruso. "Residues of Skew Rational Functions and Linearized Goppa Codes". tex.hal_id: hal-02268790 tex.hal_version: v1. Aug. 2019. URL: https://hal.science/hal-02268790 (accessed on 10/18/2024).

[CL09] J.-M. Couveignes and R. Lercier. "Elliptic Periods for Finite Fields". In: *Finite Fields and Their Applications* 15.1 (2009), pp. 1–22. DOI: 10.1016/j.ffa.2008.07.004.

[CL17a] X. Caruso and J. Le Borgne. "A New Faster Algorithm for Factoring Skew Polynomials Over Finite Fields". In: *Journal of Symbolic Computation* 79 (Mar. 2017), pp. 411–443. ISSN: 07477171. DOI: 10.1016/j.jsc.2016.02.016.

[CL17b] X. Caruso and J. Le Borgne. "Fast Multiplication for Skew Polynomials". In: *Proceedings of the 2017 ACM International Symposium on Symbolic and Algebraic Computation*. ISSAC '17: International Symposium on Symbolic and Algebraic Computation. Kaiserslautern Germany: ACM, July 23, 2017, pp. 77–84. ISBN: 978-1-4503-5064-8. DOI: 10.1145/3087604.3087617.

[CLO92] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Red. by J. H. Ewing, F. W. Gehring, and P. R. Halmos. Undergraduate Texts in Mathematics. New York, NY: Springer, 1992. ISBN: 978-1-4757-2181-2. DOI: 10.1007/978-1-4757-2181-2.

[CLT19] A. Couvreur, M. Lequesne, and J.-P. Tillich. "Recovering Short Secret Keys of RLCE in Polynomial Time". In: *Post-Quantum Cryptography*. Vol. 11505. Series Title: Lecture Notes in Computer Science. Cham: Springer, 2019, pp. 133–152. ISBN: 978-3-030-25510-7. DOI: 10.1007/978-3-030-25510-7_8.

[CMP15] A. Couvreur, I. Márquez-Corbella, and R. Pellikaan. "Cryptanalysis of Public-Key Cryptosystems That Use Subcodes of Algebraic Geometry Codes". In: *Coding Theory and Applications*. Vol. 3. Series Title: CIM Series in Mathematical Sciences. Cham: Springer, 2015, pp. 133–140. ISBN: 978-3-319-17296-5. DOI: 10.1007/978-3-319-17296-5_13.

[CS03] D. Coppersmith and M. Sudan. "Reconstructing Curves in Three (and Higher) Dimensional Space from Noisy Data". In: *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. STOC03: The 35th Annual ACM Symposium on Theory of Computing. San Diego CA USA: ACM, June 9, 2003, pp. 136–142. ISBN: 978-1-58113-674-6. DOI: 10.1145/780542.780563.

[CS96] F. Chabaud and J. Stern. "The Cryptographic Security of the Syndrome Decoding Problem for Rank Distance Codes". In: *Advances in Cryptology — ASIACRYPT '96*. Berlin, Heidelberg: Springer, 1996, pp. 368–381. ISBN: 978-3-540-70707-3.

[Del78] P. Delsarte. "Bilinear Forms Over a Finite Field with Applications to Coding Theory". In: *J. Comb. Theory Ser. A* 25.3 (Nov. 1978), pp. 226–241.

[DST19] T. Debris-Alazard, N. Sendrier, and J.-P. Tillich. "Wave: A New Family of Trapdoor One-Way Preimage Sampleable Functions Based on Codes". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 11921 LNCS. ISSN: 16113349. 2019. DOI: 10.1007/978-3-030-34578-5_2.

[Eli57] P. Elias. *List Decoding for Noisy Channels*. 335. Research Laboratory of Electronics, Massachusetts Institute of Technology, 1957.
URL: http://hdl.handle.net/1721.1/4484 (accessed on 10/28/2024).

[EV11] T. Etzion and A. Vardy. "Error-Correcting Codes in Projective Space". In: *IEEE Transactions on Information Theory* 57.2 (Feb. 2011), pp. 1165–1173. ISSN: 0018-9448, 1557-9654. DOI: 10.1109/TIT.2010.2095232.

[EWZ18] M. Elleuch, A. Wachter-Zeh, and A. Zeh. *A Public-Key Cryptosystem from Interleaved Goppa Codes*. Version Number: 1. 2018. DOI: 10.48550/ARXIV.1809.03024.

[Fed24] Federal Office for Information Security. *BSI TR-02102-1: Cryptographic Mechanisms: Recommendations and Key Lengths*. Version: 2024-1. Feb. 2, 2024.
URL: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf (accessed on 10/25/2024).

[Fit95] P. Fitzpatrick. "On the Key Equation". In: *IEEE Transactions on Information Theory* 41.5 (1995). Publisher: IEEE, pp. 1290–1302.

[FL06] C. Faure and P. Loidreau. "A New Public-Key Cryptosystem Based on the Problem of Reconstructing p-Polynomials". In: *Coding and Cryptography*. Springer, 2006, pp. 304–315.

[FOP+16] J.-C. Faugère, A. Otmani, L. Perret, F. De Portzamparc, and J.-P. Tillich. "Structural Cryptanalysis of Mceliece Schemes with Compact Keys". In: *Designs, Codes and Cryptography* 79.1 (Apr. 2016), pp. 87–112. ISSN: 0925-1022, 1573-7586. DOI: 10.1007/s10623-015-0036-z.

[For66] G. Forney. "Generalized Minimum Distance Decoding". In: *IEEE Transactions on Information Theory* 12.2 (Apr. 1966), pp. 125–131. ISSN: 0018-9448. DOI: 10.1109/TIT.1966.1053873.

[Gab08] E. M. Gabidulin. "Attacks and Counter-Attacks on the GPT Public Key Cryptosystem". In: *Designs, Codes and Cryptography* 48.2 (Aug. 2008), pp. 171–177. ISSN: 0925-1022, 1573-7586. DOI: 10.1007/s10623-007-9160-8.

[Gab85] E. M. Gabidulin. "Theory of Codes with Maximum Rank Distance". In: *Problems of Information Transmission* 21.1 (1985), pp. 3–16.

[Gab92] E. M. Gabidulin. "A Fast Matrix Decoding Algorithm for Rank-Error-Correcting Codes". In: *Algebraic Coding*. Vol. 573. Series Title: Lecture Notes in Computer Science. Berlin/Heidelberg: Springer, 1992, pp. 126–133. ISBN: 978-3-540-55130-0. DOI: 10.1007/BFb0034349.

[Gao03] S. Gao. "A New Algorithm for Decoding Reed–Solomon Codes". In: *Communications, Information and Network Security*. Springer, 2003, pp. 55–68.

[Gib95] J. K. Gibson. "Severely Denting the Gabidulin Version of the Mceliece Public Key Cryptosystem". In: *Designs, Codes and Cryptography* 6.1 (July 1995), pp. 37–45. ISSN: 0925-1022, 1573-7586. DOI: 10.1007/BF01390769.

[Gib96] K. Gibson. "The Security of the Gabidulin Public Key Cryptosystem". In: *Advances in Cryptology — EUROCRYPT '96*. Red. by G. Goos, J. Hartmanis, and J. Van Leeuwen. Vol. 1070. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1996, pp. 212–223. ISBN: 978-3-540-68339-1. DOI: 10.1007/3-540-68339-9_19.

[GJV03] P. Giorgi, C.-P. Jeannerod, and G. Villard. "On the Complexity of Polynomial Matrix Computations". In: *Proceedings of the 2003 international symposium on symbolic and algebraic computation*. 2003, pp. 135–142.

[GO01] E. M. Gabidulin and A. Ourivski. "Modified GPT PKC with Right Scrambler". In: *Electronic Notes in Discrete Mathematics* 6 (Apr. 2001), pp. 168–177. ISSN: 15710653. DOI: 10.1016/S1571-0653(04)00168-4.

[GOHA03] E. M. Gabidulin, A. Ourivski, B. Honary, and B. Ammar. "Reducible Rank Codes and Their Applications to Cryptography". In: *IEEE Transactions on Information Theory* 49.12 (Dec. 2003), pp. 3289–3293. ISSN: 0018-9448. DOI: 10.1109/TIT.2003.820038.

[GOK18] P. Gaborit, A. Otmani, and H. T. Kalachi. "Polynomial-Time Key Recovery Attack on the Faure–Loidreau Scheme Based on Gabidulin Codes". In: *Designs, Codes and Cryptography* 86.7 (July 2018), pp. 1391–1403. ISSN: 0925-1022, 1573-7586. DOI: 10.1007/s10623-017-0402-0.

[Gos96] D. Goss. *Basic Structures of Function Field Arithmetic*. Ergebnisse der Mathematik und ihrer Grenzgebiete Folge 3, Bd. 35. Berlin Heidelberg: Springer, 1996. 422 pp. ISBN: 978-3-540-61087-8.

[GP04] E. M. Gabidulin and N. Pilipchuk. "Symmetric Rank Codes". In: *Problems of Information Transmission* 40 (Apr. 2004), pp. 103–117. DOI: 10.1023/B:PRIT.0000043925.67309.c6.

[GP06] E. M. Gabidulin and N. I. Pilipchuk. "Symmetric Matrices and Codes Correcting Rank Errors Beyond the (d-1)/2 Bound". In: *Discrete Applied Mathematics* 154.2 (2006), pp. 305–312. ISSN: 0166-218X. DOI: 10.1016/j.dam.2005.03.012.

[GPT91a] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. "Ideals Over a Non-Commutative Ring and Their Application in Cryptology". In: *Advances in Cryptology — EUROCRYPT '91*. Vol. 547. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1991, pp. 482–489. ISBN: 978-3-540-54620-7. DOI: 10.1007/3-540-46416-6_41.

[GPT91b] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. "Ideals Over a Non-Commutative Ring and Their Application in Cryptology". In: *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 1991, pp. 482–489.

[GRH09] E. M. Gabidulin, H. Rashwan, and B. Honary. "On Improving Security of GPT Cryptosystems". In: *2009 IEEE International Symposium on Information Theory*. 2009 IEEE International Symposium on Information Theory - ISIT. Seoul, South Korea: IEEE, June 2009, pp. 1110–1114. ISBN: 978-1-4244-4312-3. DOI: 10.1109/ISIT.2009.5206029.

[Gro96] L. K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing - STOC '96*. the twenty-eighth annual ACM symposium. Philadelphia, Pennsylvania, United States: ACM Press, 1996, pp. 212–219. ISBN: 978-0-89791-785-8. DOI: 10.1145/237814.237866.

[GRS16] P. Gaborit, O. Ruatta, and J. Schrek. "On the Complexity of the Rank Syndrome Decoding Problem". In: *IEEE Trans. Inform. Theory* 62.2 (2016), pp. 1006–1019. ISSN: 0018-9448. DOI: 10.1109/TIT.2015.2511786.

[HB23] F. Hörmann and H. Bartz. "Fast Gao-Like Decoding of Horizontally Interleaved Linearized Reed–Solomon Codes". In: *Code-Based Cryptography*. Vol. 14311. Series Title: Lecture Notes in Computer Science. Cham: Springer, 2023, pp. 14–34. ISBN: 978-3-031-46494-2. DOI: 10.1007/978-3-031-46495-9_2.

[HBH23] F. Hörmann, H. Bartz, and A.-L. Horlemann. "Distinguishing and Recovering Generalized Linearized Reed–Solomon Codes". In: *Code-Based Cryptography*. Vol. 13839. Series Title: Lecture Notes in Computer Science. Cham: Springer, 2023, pp. 1–20. ISBN: 978-3-031-29689-5. DOI: 10.1007/978-3-031-29689-5_1.

[HBP22] F. Hörmann, H. Bartz, and S. Puchinger. "Error-Erasure Decoding of Linearized Reed–Solomon Codes in the Sum-Rank Metric". In: *2022 IEEE International Symposium on Information Theory (ISIT)*. 2022, pp. 7–12. DOI: 10.1109/ISIT50566.2022.9834742.

[HHT23] T. Hoefler, T. Häner, and M. Troyer. “Disentangling Hype from Practicality: On Realistically Achieving Quantum Advantage”. In: *Commun. ACM* 66.5 (Apr. 21, 2023), pp. 82–87. ISSN: 0001-0782. DOI: 10.1145/3571725.

[HJ17] T. Høholdt and J. Justesen. *A Course in Error-Correcting Codes: Second Edition*. 2nd ed. EMS Textbooks in Mathematics. EMS Press, July 11, 2017. ISBN: 978-3-03719-679-3. DOI: 10.4171/179.

[HK17] A.-L. Horlemann and M. Kuijper. “A Module Minimization Approach to Gabidulin Decoding Via Interpolation”. In: *Journal of Algebra Combinatorics Discrete Structures and Applications* 5 (Dec. 2017), pp. 29–43. DOI: 10.13069/jacodesmath.369863.

[HLPW19] L. Holzbaur, H. Liu, S. Puchinger, and A. Wachter-Zeh. “On Decoding and Applications of Interleaved Goppa Codes”. In: *2019 IEEE International Symposium on Information Theory (ISIT)*. 2019, pp. 1887–1891.

[HMR16] A.-L. Horlemann-Trautmann, K. Marshall, and J. Rosenthal. “Considerations for Rank-Based Cryptosystems”. In: IEEE International Symposium on Information Theory (ISIT). Barcelona, Spain: IEEE, July 2016, pp. 2544–2548. ISBN: 978-1-5090-1806-2. DOI: 10.1109/ISIT.2016.7541758.

[HMR18] A.-L. Horlemann-Trautmann, K. Marshall, and J. Rosenthal. “Extension of Overbeck’s Attack for Gabidulin-Based Cryptosystems”. In: *Designs, Codes and Cryptography* 86.2 (Feb. 2018), pp. 319–340. ISSN: 0925-1022, 1573-7586. DOI: 10.1007/s10623-017-0343-7.

[HPR+22] A.-L. Horlemann, S. Puchinger, J. Renner, T. Schamberger, and A. Wachter-Zeh. “Information-Set Decoding with Hints”. In: *Code-Based Cryptography*. Cham: Springer, 2022, pp. 60–83. ISBN: 978-3-030-98365-9.

[JM96] H. Janwa and O. Moreno. “McEliece Public Key Cryptosystems Using Algebraic-Geometric Codes”. In: *Designs, Codes and Cryptography* 8.3 (1996), pp. 293–307. ISSN: 1573-7586. DOI: 10.1023/A:1027351723034.

[JNSV17] C.-P. Jeannerod, V. Neiger, É. Schost, and G. Villard. “Computing Minimal Interpolation Bases”. In: *Journal of Symbolic Computation* 83 (2017). Publisher: Elsevier, pp. 272–314.

[KK08] R. Kötter and F. R. Kschischang. “Coding for Errors and Erasures in Random Network Coding”. In: *IEEE Transactions on Information Theory* 54.8 (2008). Publisher: IEEE, pp. 3579–3591.

[KL97] V. Y. Krachkovsky and Y. X. Lee. “Decoding for Iterative Reed–Solomon Coding Schemes”. In: *IEEE Transactions on Magnetics* 33.5 (1997), pp. 2740–2742.

[Knu82] D. E. Knuth. *The Art of Computer Programming. 1: Fundamental Algorithms*. 2. ed., 7. print. Reading, Mass: Addison-Wesley, 1982. 634 pp. ISBN: 978-0-201-03801-9.

[KRT07] C. Kojima, P. Rapisarda, and K. Takaba. "Canonical Forms for Polynomial and Quadratic Differential Operators". In: *Systems & Control Letters* 56.11 (Nov. 2007), pp. 678–684. ISSN: 01676911. DOI: 10.1016/j.sysconle.2007.06.004.

[KY98] V. Y. Krachkovsky and Yuan Xing Lee. "Decoding of Parallel Reed–Solomon Codes with Applications to Product and Concatenated Codes". In: *Proceedings. 1998 IEEE International Symposium on Information Theory (Cat. No.98CH36252)*. 1998 IEEE International Symposium on Information Theory. Cambridge, MA, USA: IEEE, 1998, p. 55. ISBN: 978-0-7803-5000-7. DOI: 10.1109/ISIT.1998.708636.

[LCG19] P. Lefèvre, P. Carré, and P. Gaborit. "Application of Rank Metric Codes in Digital Image Watermarking". In: *Signal Processing: Image Communication* 74 (2019). Publisher: Elsevier, pp. 119–128.

[Lee58] C. Lee. "Some Properties of Nonbinary Error-Correcting Codes". In: *IEEE Transactions on Information Theory* 4.2 (June 1958), pp. 77–82. ISSN: 0018-9448. DOI: 10.1109/TIT.1958.1057446.

[Ler95] A. Leroy. "Pseudolinear Transformations and Evaluation in Ore Extensions". In: *Bulletin of the Belgian Mathematical Society-Simon Stevin* 2.3 (1995). Publisher: The Belgian Mathematic Society, pp. 321–347.

[LGB03] P. Lusina, E. M. Gabidulin, and M. Bossert. "Maximum Rank Distance Codes as Space-Time Codes". In: *Institute of Electrical and Electronics Engineers* 49.10 (Oct. 2003). Publisher: Dept. for Telecommun. & Appl. Inf. Theor., Univ. of Ulm, Germany Publisher: IEEE tex.posted-at: 2013-03-01 16:03:44 tex.priority: 2, pp. 2757–2760. ISSN: 0018-9448.

[Liu16] S. Liu. "Generalized Skew Reed-Solomon Codes and Other Applications of Skew Polynomial Evaluation". PhD thesis. University of Toronto (Canada), 2016.

[LK05] H. F. Lu and P. V. Kumar. "A Unified Construction of Space-Time Codes with Optimal Rate-Diversity Tradeoff". In: *IEEE Transactions on Information Theory* 51.5 (2005). ISSN: 00189448. DOI: 10.1109/TIT.2005.846403.

[LL88a] T.-Y. Lam and A. Leroy. "Algebraic Conjugacy Classes and Skew Polynomial Rings". In: *Perspectives in Ring Theory*. Springer, 1988, pp. 153–203.

[LL88b] T.-Y. Lam and A. Leroy. "Vandermonde and Wronskian Matrices Over Division Rings". In: *Journal of Algebra* 119.2 (1988). Publisher: Academic Press, pp. 308–336.

[LL94] T.-Y. Lam and A. Leroy. "Hilbert 90 Theorems Over Divison Rings". In: *Transactions of the American Mathematical Society* 345.2 (Oct. 1994), p. 595. ISSN: 00029947. DOI: 10.2307/2154989.

[LLP19] J. Lavauzelle, P. Loidreau, and B.-D. Pham. *RAMESSES a Rank Metric Encryption Scheme with Short Keys*. Version Number: 1. 2019. DOI: 10.48550/ARXIV.1911.13119.

[LMK14] S. Liu, F. Manganiello, and F. R. Kschischang. "Kötter Interpolation in Skew Polynomial Rings". In: *Designs, Codes and Cryptography* 72.3 (2014). Publisher: Springer, pp. 593–608.

[LN96] R. Lidl and H. Niederreiter. *Finite Fields*. Encyclopedia of Mathematics and its Applications. Published: Hardcover. Cambridge University Press, Oct. 1996. ISBN: 0-521-39231-4.

[Loi06] P. Loidreau. "Decoding Rank Errors Beyond the Error Correcting Capability". In: *Tenth International Workshop on Algebraic and Combinatorial Coding Theory,(ACCT)*. Zvenigorod, Russia, Sept. 2006, pp. 186–190.

[Loi10] P. Loidreau. "Designing a Rank Metric Based McEliece Cryptosystem". In: *Post-Quantum Cryptography*. Red. by D. Hutchison et al. Vol. 6061. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 142–152. ISBN: 978-3-642-12929-2. DOI: 10.1007/978-3-642-12929-2_11.

[Loi16] P. Loidreau. "An Evolution of GPT Cryptosystem". In: International Workshop on Algebraic and Combinatorial Coding Theory (ACCT). 2016.

[Loi17] P. Loidreau. "A New Rank Metric Codes Based Encryption Scheme". In: *Post-Quantum Cryptography*. Vol. 10346. Series Title: Lecture Notes in Computer Science. Cham: Springer, 2017, pp. 3–17. ISBN: 978-3-319-59879-6. DOI: 10.1007/978-3-319-59879-6_1.

[MAB+20] C. A. Melchor et al. *Rank Quasi-Cyclic (RQC)*. Submission to NIST Post-Quantum Cryptography Standardization Process, Second Round version - update for April 21st, 2020. 2020.
URL: https://pqc-rqc.org (accessed on 10/09/2024).

[MAB+24] C. A. Melchor et al. *Hamming Quasi-Cyclic (HQC)*. Submission to NIST Post-Quantum Cryptography Standardization Process, Fourth round version. 2024.
URL: https://pqc-hqc.org (accessed on 10/09/2024).

[Mar18] U. Martínez-Peñas. "Skew and Linearized Reed–Solomon Codes and Maximum Sum Rank Distance Codes Over Any Division Ring". In: *Journal of Algebra* 504 (2018). Publisher: Elsevier, pp. 587–612.

[Mas69] J. Massey. "Shift-Register Synthesis and BCH Decoding". In: *IEEE Transactions on Information Theory* 15.1 (Jan. 1969), pp. 122–127. ISSN: 0018-9448. DOI: 10.1109/TIT.1969.1054260.

[MB93] A. J. Menezes and I. F. Blake, eds. *Applications of Finite Fields*. The Kluwer International Series in Engineering and Computer Science; Communications and Information Theory SECS199. Boston: Kluwer Academic Publishers, 1993. 218 pp. ISBN: 978-0-7923-9282-8.

[McE78] R. J. McEliece. "A Public-Key Cryptosystem Based On Algebraic Coding Theory". In: *The Deep Space Network Progress Report* 42-44 (1978). Publisher: Jet Propulsion Laboratory, pp. 114–116.

[Mid12] J. Middeke. "A Computational View on Normal Forms of Matrices of Ore Polynomials". In: *ACM Communications in Computer Algebra* 45.3 (Jan. 23, 2012), pp. 190–191. ISSN: 1932-2240. DOI: 10.1145/2110170.2110182.

[MK19a] U. Martínez-Peñas and F. R. Kschischang. "Reliable and Secure Multishot Network Coding Using Linearized Reed–Solomon Codes". In: *IEEE Transactions on Information Theory* 65.8 (2019). Publisher: IEEE, pp. 4785–4803.

[MK19b] U. Martínez-Peñas and F. R. Kschischang. "Universal and Dynamic Locally Repairable Codes With Maximal Recoverability via Sum-Rank Codes". In: *IEEE Transactions on Information Theory* 65.12 (2019), pp. 7790–7805. DOI: 10.1109/TIT.2019.2924888.

[MK90] J. J. Metzner and E. J. Kapturowski. "A General Decoding Technique Applicable to Replicated File Disagreement Location and Concatenated Code Decoding". In: *IEEE Transactions on Information Theory* 36.4 (1990). Publisher: IEEE, pp. 911–917.

[MMO04] T. Migler, K. E. Morrison, and M. Ogle. "Weight and Rank of Matrices Over Finite Fields". In: (2004). Publisher: arXiv. DOI: 10.48550/ARXIV.MATH/0403314.

[MMT11] A. May, A. Meurer, and E. Thomae. "Decoding Random Linear Codes in O~(2^0.054n)". In: *Advances in Cryptology – ASIACRYPT 2011*. Red. by D. Hutchison et al. Vol. 7073. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2011, pp. 107–124. ISBN: 978-3-642-25384-3. DOI: 10.1007/978-3-642-25385-0_6.

[Moo05] T. K. Moon. *Error Correction Coding: Mathematical Methods and Algorithms*. 1st ed. Wiley, May 13, 2005. ISBN: 978-0-471-73921-0. DOI: 10.1002/0471739219.

[Moo16] D. Moody. *Post Quantum Cryptography Team, National Institute of Standards and Technology (NIST)*. 2016.
URL: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/pqcrypto-2016-presentation.pdf (accessed on 10/09/2024).

[Moo96] E. H. Moore. "A Two-Fold Generalization of Fermat's Theorem". In: *Bulletin of the American Mathematical Society* 2 (1896), pp. 189–199.

[MP74] G. Matsaglia and G. P. H. Styan. "Equalities and Inequalities for Ranks of Matrices †". In: *Linear and Multilinear Algebra* 2.3 (Jan. 1974), pp. 269–292. ISSN: 0308-1087, 1563-5139. DOI: 10.1080/03081087408817070.

[MS03] T. Mulders and A. Storjohann. "On Lattice Reduction for Polynomial Matrices". In: *Journal of Symbolic Computation* 35.4 (Apr. 2003), pp. 377–401. ISSN: 07477171. DOI: 10.1016/S0747-7171(02)00139-6.

[MS07] L. Minder and A. Shokrollahi. "Cryptanalysis of the Sidelnikov Cryptosystem". In: *Advances in Cryptology - EUROCRYPT 2007*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2007, pp. 347–360. ISBN: 978-3-540-72540-4. DOI: 10.1007/978-3-540-72540-4_20.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. Published: Hardcover. North Holland Publishing Co., 1977. ISBN: 0-444-85193-3.

[MSK22] U. Martínez-Peñas, M. Shehadeh, and F. R. Kschischang. "Codes in the Sum-Rank Metric: Fundamentals and Applications". In: *Foundations and Trends in Communications and Information Theory* 19.5 (2022), pp. 814–1031. ISSN: 1567-2190, 1567-2328. DOI: 10.1561/0100000120.

[MW18] D. Micciancio and M. Walter. "On the Bit Security of Cryptographic Primitives". In: *Advances in Cryptology – EUROCRYPT 2018*. Vol. 10820. Series Title: Lecture Notes in Computer Science. Cham: Springer, 2018, pp. 3–28. ISBN: 978-3-319-78380-2. DOI: 10.1007/978-3-319-78381-9_1.

[Nat22] National Institute of Standards and Technology. *PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates*. CSRC | NIST. Mar. 24, 2022.
URL: https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4 (accessed on 10/16/2024).

[Nat24a] National Institute of Standards and Technology. *Announcing Approval of Three Federal Information Processing Standards (FIPS) for Post-Quantum Cryptography*. CSRC | NIST. Aug. 6, 2024.
URL: https://csrc.nist.gov/News/2024/postquantum-cryptography-fips-approved (accessed on 10/16/2024).

[Nat24b] National Institute of Standards and Technology. *Module-Lattice-Based Digital Signature Standard*. FIPS 204. Gaithersburg, MD: National Institute of Standards and Technology, Aug. 13, 2024, FIPS 204. DOI: 10.6028/NIST.FIPS.204.
URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf (accessed on 10/18/2024).

[Nat24c] National Institute of Standards and Technology. *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. FIPS 203. Gaithersburg, MD: National Institute of Standards and Technology, Aug. 13, 2024, FIPS 203. DOI: 10.6028/NIST.FIPS.203.
URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf (accessed on 10/18/2024).

[Nat24d] National Institute of Standards and Technology. *Stateless Hash-Based Digital Signature Standard*. FIPS 205. Gaithersburg, MD: National Institute of Standards and Technology, Aug. 13, 2024, FIPS 205. DOI: 10.6028/NIST.FIPS.205.
URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf (accessed on 10/18/2024).

[Nei16] Neiger, Vincent. "Bases of Relations in One or Several Variables: Fast Algorithms and Applications". PhD thesis. École Normale Supérieure de Lyon - University of Waterloo, 2016.

[Nie13] J. S. R. Nielsen. "List Decoding of Algebraic Codes". Number: 309 Series: DTU compute PHD-2013. PhD thesis. Technical University of Denmark, 2013.

[Nie14] J. S. Nielsen. "Fast Kötter-Nielsen-Høholdt Interpolation in the Guruswami-Sudan Algorithm". In: *14th International Workshop on Algebraic and Combinatorial Coding Theory (ACCT)*. Svetlogorsk , Russia, 2014.

[Nie86] H. Niederreiter. "Knapsack-Type Cryptosystems and Algebraic Coding Theory". In: *Problems of Control and Information Theory / Problemy Upravleniya i Teorii Informatsii* 15.2 (1986), pp. 159–166.

[NPRV17] D. Napp, R. Pinto, J. Rosenthal, and P. Vettori. "MRD Rank Metric Convolutional Codes". In: *2017 IEEE International Symposium on Information Theory (ISIT)*. 2017 IEEE International Symposium on Information Theory (ISIT). Aachen, Germany: IEEE, June 2017, pp. 2766–2770. ISBN: 978-1-5090-4096-4. DOI: 10.1109/ISIT.2017.8007033.

[NU10] R. W. Nóbrega and B. F. Uchôa-Filho. "Multishot Codes for Network Coding Using Rank-Metric Codes". In: *2010 Third IEEE International Workshop on Wireless Network Coding*. IEEE, 2010, pp. 1–6.

[OKN18] A. Otmani, H. T. Kalachi, and S. Ndjeya. "Improved Cryptanalysis of Rank Metric Schemes Based on Gabidulin Codes". In: *Designs, Codes and Cryptography* 86.9 (Sept. 2018), pp. 1983–1996. ISSN: 0925-1022, 1573-7586. DOI: 10.1007/s10623-017-0434-5.

[OLW22] C. Ott, H. Liu, and A. Wachter-Zeh. "Covering Properties of Sum-Rank Metric Codes". In: *2022 58th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2022*. 2022. DOI: 10.1109/Allerton49937.2022.9929421.

[OPB21] C. Ott, S. Puchinger, and M. Bossert. "Bounds and Genericity of Sum-Rank-Metric Codes". In: *2021 17th International Symposium Problems of Redundancy in Information and Control Systems, REDUNDANCY 2021*. 2021. DOI: 10.1109/REDUNDANCY52534.2021.9606442.

[Ore33a] Ø. Ore. "On a Special Class of Polynomials". In: *Transactions of the American Mathematical Society* 35 (1933). tex.citeulike-article-id: 8303181 tex.posted-at: 2010-11-24 11:13:56 tex.priority: 2, pp. 559–584.

[Ore33b] Ø. Ore. "Theory of Non-Commutative Polynomials". In: *Annals of Mathematics* (1933). Publisher: JSTOR, pp. 480–508.

[Ove05] R. Overbeck. "A New Structural Attack for GPT and Variants". In: *Progress in Cryptology – Mycrypt 2005*. Red. by D. Hutchison et al. Vol. 3715. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 50–63. ISBN: 978-3-540-32066-1. DOI: 10.1007/11554868_5.

[Ove06] R. Overbeck. "Extending Gibson's Attacks on the GPT Cryptosystem". In: *Coding and Cryptography*. Vol. 3969. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 178–188. ISBN: 978-3-540-35482-6. DOI: 10.1007/11779360_15.

[Ove08] R. Overbeck. "Structural Attacks for Public Key Cryptosystems Based on Gabidulin Codes". In: *Journal of Cryptology* 21.2 (Apr. 2008), pp. 280–301. ISSN: 0933-2790, 1432-1378. DOI: 10.1007/s00145-007-9003-9.

[PG06] N. I. Pilipchuk and E. M. Gabidulin. “On Codes Correcting Symmetric Rank Errors”. In: *Coding and cryptography*. Berlin, Heidelberg: Springer, 2006, pp. 14–21. ISBN: 978-3-540-35482-6.

[PMM+17] S. Puchinger, S. Müelich, D. Mödinger, J. Rosenkilde Né Nielsen, and M. Bossert. “Decoding Interleaved Gabidulin Codes Using Alekhnovich’s Algorithm”. In: *Electronic Notes in Discrete Mathematics* 57 (Mar. 2017), pp. 175–180. ISSN: 15710653. DOI: 10.1016/j.endm.2017.02.029.

[PR17] S. Puchinger and J. Rosenkilde Ne Nielsen. “Decoding of Interleaved Reed-Solomon Codes Using Improved Power Decoding”. In: *2017 IEEE International Symposium on Information Theory (ISIT)*. 2017 IEEE International Symposium on Information Theory (ISIT). Aachen, Germany: IEEE, June 2017, pp. 356–360. ISBN: 978-1-5090-4096-4. DOI: 10.1109/ISIT.2017.8006549.

[PR21] S. Puchinger and J. Rosenkilde. “Bounds on List Decoding of Linearized Reed–Solomon Codes”. In: *2021 IEEE International Symposium on Information Theory (ISIT)*. 2021, pp. 154–159. DOI: 10.1109/ISIT45174.2021.9517777.

[Pra62] E. Prange. “The Use of Information Sets in Decoding Cyclic Codes”. In: *IEEE Transactions on Information Theory* 8.5 (Sept. 1962), pp. 5–9. ISSN: 0018-9448. DOI: 10.1109/TIT.1962.1057777.

[PRLS17] S. Puchinger, J. Rosenkilde né Nielsen, W. Li, and V. Sidorenko. “Row Reduction Applied to Decoding of Rank-Metric and Subspace Codes”. In: *Designs, Codes and Cryptography* 82.1 (Jan. 2017), pp. 389–409. ISSN: 0925-1022, 1573-7586. DOI: 10.1007/s10623-016-0257-9.

[PRR20] S. Puchinger, J. Renner, and J. Rosenkilde. “Generic Decoding in the Sum-Rank Metric”. In: *2020 IEEE International Symposium on Information Theory (ISIT)*. Vol. 2020-June. ISSN: 21578095. 2020. DOI: 10.1109/ISIT44484.2020.9174497.

[PRR22] S. Puchinger, J. Renner, and J. Rosenkilde. “Generic Decoding in the Sum-Rank Metric”. In: *IEEE Transactions on Information Theory* 68.8 (2022). ISSN: 15579654. DOI: 10.1109/TIT.2022.3167629.

[PRW19] S. Puchinger, J. Renner, and A. Wachter-Zeh. *Decoding High-Order Interleaved Rank-Metric Codes*. Version Number: 1. 2019. DOI: 10.48550/ARXIV.1904.08774.

[PT91] A. V. Paramonov and O. V. Tretjakov. “An Analogue of Berlekamp-Massey Algorithm for Decoding Codes in Rank Metric”. In: *Moscow inst. Physics and technology (MIPT)*. Proceedings MIPT. tex.citeulike-article-id: 6188113 tex.posted-at: 2009-11-22 13:59:12 tex.priority: 2. 1991.

[PW08] W. W. Peterson and E. J. Weldon. *Error-Correcting Codes*. 2. ed., 16. print. Cambridge, Mass.: MIT Press, 2008. 560 pp. ISBN: 978-0-262-16039-1.

[PW16] S. Puchinger and A. Wachter-Zeh. "Sub-Quadratic Decoding of Gabidulin Codes". In: *IEEE Int. Symp. Inf. Theory (ISIT)*. Place: Barcelona, Spain. July 2016, pp. 2554–2558.

[PW18] S. Puchinger and A. Wachter-Zeh. "Fast Operations on Linearized Polynomials and Their Applications in Coding Theory". In: *Journal of Symbolic Computation* 89 (Nov. 2018), pp. 194–215. ISSN: 07477171. DOI: 10.1016/j.jsc.2017.11.012.

[PZ03] J. Proos and C. Zalka. "Shor's Discrete Logarithm Quantum Algorithm for Elliptic Curves". In: *Quantum Information & Computation* 3 (Feb. 2003). DOI: 10.26421/QIC3.4-3.

[RGH10] H. Rashwan, E. M. Gabidulin, and B. Honary. "A Smart Approach for GPT Cryptosystem Based on Rank Codes". In: *2010 IEEE International Symposium on Information Theory*. 2010 IEEE International Symposium on Information Theory - ISIT. Austin, TX, USA: IEEE, June 2010, pp. 2463–2467. ISBN: 978-1-4244-7891-0. DOI: 10.1109/ISIT.2010.5513549.

[RGH11] H. Rashwan, E. M. Gabidulin, and B. Honary. "Security of the GPT Cryptosystem and Its Applications to Cryptography". In: *Security and Communication Networks* 4.8 (Aug. 2011), pp. 937–946. ISSN: 1939-0114, 1939-0122. DOI: 10.1002/sec.228.

[Ros02] M. I. Rosen. *Number Theory in Function Fields*. Graduate texts in mathematics 210. New York: Springer, 2002. ISBN: 978-0-387-95335-9.

[Rot06] R. M. Roth. *Introduction to Coding Theory*. OCLC: ocm61757112. Cambridge, UK ; New York: Cambridge University Press, 2006. 566 pp. ISBN: 978-0-521-84504-5.

[Rot91] R. M. Roth. "Maximum-Rank Array Codes and Their Application to Crisscross Error Correction". In: *IEEE Trans. Inform. Theory* 37.2 (Mar. 1991), pp. 328–336.

[RP04a] G. Richter and S. Plass. "Fast Decoding of Rank-Codes with Rank Errors and Column Erasures". In: *IEEE int. Symp. Inf. Theory (ISIT)*. International Symposium on Information Theory 2004. Place: Chicago, IL, USA tex.citeulike-article-id: 3744725 tex.posted-at: 2009-11-04 20:28:11 tex.priority: 2. 2004, p. 398. DOI: 10.1109/ISIT.2004.1365435.

[RP04b] G. Richter and S. Plass. "Error and Erasure Decoding of Rank-Codes with a Modified Berlekamp-Massey Algorithm". In: ITG-Fachbericht. Jan. 2004.

[RPW18] J. Renner, S. Puchinger, and A. Wachter-Zeh. "On a Rank-Metric Code-Based Cryptosystem with Small Key Size". In: *arXiv preprint arXiv:1812.04892* (2018).

[RPW19] J. Renner, S. Puchinger, and A. Wachter-Zeh. "Interleaving Loidreau's Rank-Metric Cryptosystem". In: *2019 XVI International Symposium "Problems of Redundancy in Information and Control Systems" (REDUNDANCY)*. IEEE, 2019, pp. 127–132.

[RPW21a] J. Renner, S. Puchinger, and A. Wachter-Zeh. "Decoding High-Order Interleaved Rank-Metric Codes". In: *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 19–24.

[RPW21b] J. Renner, S. Puchinger, and A. Wachter-Zeh. "LIGA: A Cryptosystem Based on the Hardness of Rank-Metric List and Interleaved Decoding". In: *Designs, Codes and Cryptography* 89.6 (June 2021), pp. 1279–1319. ISSN: 0925-1022, 1573-7586. DOI: 10.1007/s10623-021-00861-z.

[SB10] V. Sidorenko and M. Bossert. "Decoding Interleaved Gabidulin Codes and Multisequence Linearized Shift-Register Synthesis". In: *2010 Ieee International Symposium on Information Theory*. 2010, pp. 1148–1152. DOI: 10.1109/ISIT.2010.5513676.

[Sha48] C. E. Shannon. "A Mathematical Theory of Communication". In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[Sho97] P. W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. ISSN: 0097-5397, 1095-7111. DOI: 10.1137/S0097539795293172.

[Sid94] V. M. Sidelnikov. "A Public-Key Cryptosystem Based on Binary Reed-Muller Codes". In: 4.3 (Jan. 1, 1994). Publisher: De Gruyter Section: Discrete Mathematics and Applications, pp. 191–208. ISSN: 1569-3929. DOI: 10.1515/dma.1994.4.3.191.

[SJB11] V. Sidorenko, L. Jiang, and M. Bossert. "Skew-Feedback Shift-Register Synthesis and Decoding Interleaved Gabidulin Codes". In: *IEEE Trans. Inform. Theory* 57.2 (Feb. 2011). Backup Publisher: Inst. of Telecommun. & Appl. Inf. Theor., Ulm Univ., Ulm, Germany Publisher: IEEE, pp. 621–632. ISSN: 0018-9448.

[SK20] M. Shehadeh and F. R. Kschischang. "Rate-Diversity Optimal Multiblock Space-Time Codes via Sum-Rank Codes". In: *2020 IEEE International Symposium on Information Theory (ISIT)*. 2020 IEEE International Symposium on Information Theory (ISIT). Los Angeles, CA,

USA: IEEE, June 2020, pp. 3055–3060. ISBN: 978-1-72816-432-8. DOI: 10.1109/ISIT44484.2020.9174329.

[SKK08] D. Silva, F. R. Kschischang, and R. Kötter. "A Rank-Metric Approach to Error Control in Random Network Coding". In: *IEEE Transactions on Information Theory* 54.9 (Sept. 2008), pp. 3951–3967.

[SRB11] V. Sidorenko, G. Richter, and M. Bossert. "Linearized Shift-Register Synthesis". In: *IEEE Transactions on Information Theory* 57.9 (2011), pp. 6025–6032. DOI: 10.1109/TIT.2011.2162173.

[SRV12] N. Silberstein, A. S. Rawat, and S. Vishwanath. "Error Resilience in Distributed Storage Via Rank-Metric Codes". In: *2012 50th annual allerton conference on communication, control, and computing (allerton)*. 2012, pp. 1150–1157. DOI: 10.1109/Allerton.2012.6483348.

[SS92] V. M. Sidelnikov and S. O. Shestakov. "On Insecurity of Cryptosystems based on Generalized Reed–Solomon Codes". In: *Discrete Mathematics and Applications* 2.4 (1992). Publisher: Walter de Gruyter GmbH.

[SSB07] G. Schmidt, V. Sidorenko, and M. Bossert. "Enhancing the Correcting Radius of Interleaved Reed-Solomon Decoding Using Syndrome Extension Techniques". In: *2007 IEEE International Symposium on Information Theory*. 2007 IEEE International Symposium on Information Theory. Nice: IEEE, June 2007, pp. 1341–1345. ISBN: 978-1-4244-1397-3. DOI: 10.1109/ISIT.2007.4557409.

[ST21] U. Skosana and M. Tame. "Demonstration of Shor's Factoring Algorithm for N = 21 on Ibm Quantum Processors". In: *Scientific Reports* 11.1 (Aug. 16, 2021), p. 16599. ISSN: 2045-2322. DOI: 10.1038/s41598-021-95973-w.

[Ste89] J. Stern. "A Method for Finding Codewords of Small Weight". In: *Coding Theory and Applications*. Vol. 388. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1989, pp. 106–113. ISBN: 978-3-540-51643-9. DOI: 10.1007/BFb0019850.

[Stu05] B. Sturmfels. "What is ... a Gröbner Basis". In: *Notices of the American Mathematical Society* 52.10 (2005), pp. 1199–1200.

[SWC12] V. Sidorenko, A. Wachter-Zeh, and D. Chen. "On Fast Decoding of Interleaved Gabidulin Codes". In: *Int. Symp. Probl. Redundancy Inf. Control Systems*. St. Petersburg, Russia, Sept. 2012, pp. 78–83.

[The23] The Sage Developers. *SageMath, the Sage Mathematics Software System (version 9.8)*. manual. tex.key: SageMath. 2023.

[Wac16] A. Wachter-Zeh. "Decoding of Block and Convolutional Codes in Rank Metric". PhD thesis. Universität Ulm, 2016. DOI: 10.18725/OPARU-2515. URL: https://oparu.uni-ulm.de/xmlui/handle/123456789/2542 (accessed on 10/18/2024).

[Wan16] Y. Wang. "Quantum Resistant Random Linear Code Based Public Key Encryption Scheme RLCE". In: 2016 IEEE International Symposium on Information Theory (ISIT). Barcelona, Spain: IEEE, July 2016, pp. 2519–2523. ISBN: 978-1-5090-1806-2. DOI: 10.1109/ISIT.2016.7541753.

[Wie10] C. Wieschebrink. "Cryptanalysis of the Niederreiter Public Key Scheme Based on GRS Subcodes". In: *Post-Quantum Cryptography*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 61–72. ISBN: 978-3-642-12929-2. DOI: 10.1007/978-3-642-12929-2_5.

[WMW05] B. Wang, R. J. McEliece, and K. Watanabe. "Kötter Interpolation Over Free Modules". In: *Proceedings of*. 2005, pp. 2197–2206.

[Woz58] J. M. Wozencraft. *List Decoding*. Quarterly Progress Report 48. Research Laboratory of Electronics, Massachusetts Institute of Technology, Jan. 1958, pp. 90–95.

[WPR18] A. Wachter-Zeh, S. Puchinger, and J. Renner. "Repairing the Faure-Loidreau Public-Key Cryptosystem". In: *IEEE International Symposium on Information Theory (ISIT)*. Place: Vail, Colorado, USA. June 2018, pp. 2426–2430.

[WS12] A. Wachter-Zeh and V. Sidorenko. "Rank Metric Convolutional Codes for Random Linear Network Coding". In: *2012 International Symposium on Network Coding (NetCod)*. 2012 International Symposium on Network Coding (NetCod). Cambridge, MA, USA: IEEE, June 2012, pp. 1–6. ISBN: 978-1-4673-1892-1. DOI: 10.1109/NETCOD.2012.6261875.

[WSBZ11] A. Wachter, V. Sidorenko, M. Bossert, and V. V. Zyablov. "On (Partial) Unit Memory Codes Based on Gabidulin Codes". In: *Problems of Information Transmission* 47.2 (June 1, 2011), pp. 117–129. ISSN: 1608-3253. DOI: 10.1134/S0032946011020049.

[WSS15] A. Wachter-Zeh, M. Stinner, and V. Sidorenko. "Convolutional Codes in Rank Metric With Application to Random Network Coding". In: *IEEE Transactions on Information Theory* 61.6 (June 2015), pp. 3199–3213. ISSN: 0018-9448, 1557-9654. DOI: 10.1109/TIT.2015.2424930.

[WZ14] A. Wachter-Zeh and A. Zeh. "List and Unique Error-Erasure Decoding of Interleaved Gabidulin Codes with Interpolation Techniques". In: *Designs, Codes and Cryptography* 73.2 (Nov. 1, 2014), pp. 547–570. ISSN: 1573-7586. DOI: 10.1007/s10623-014-9953-5.

[WZB14] A. Wachter-Zeh, A. Zeh, and M. Bossert. "Decoding Interleaved Reed–Solomon Codes Beyond Their Joint Error-Correcting Capability". In: *Designs, Codes and Cryptography* 71.2 (May 2014), pp. 261–281. ISSN: 0925-1022, 1573-7586. DOI: 10.1007/s10623-012-9728-9.

[XYS11] H. Xie, Z. Yan, and B. W. Suter. "General Linearized Polynomial Interpolation and Its Applications". In: *2011 International Symposium on Networking Coding*. 2011 International Symposium on Network Coding (NetCod). Beijing, China: IEEE, July 2011, pp. 1–4. ISBN: 978-1-61284-138-0. DOI: 10.1109/ISNETCOD.2011.5978942.

[XZL+12] N. Xu, J. Zhu, D. Lu, X. Zhou, X. Peng, and J. Du. "Quantum Factorization of 143 on a Dipolar-Coupling Nuclear Magnetic Resonance System". In: *Physical Review Letters* 108.13 (Mar. 30, 2012). Publisher: American Physical Society, p. 130501. DOI: 10.1103/PhysRevLett.108.130501.

[YL18] J.-H. Yu and H.-A. Loeliger. "Simultaneous Partial Inverses and Decoding Interleaved Reed–Solomon Codes". In: *IEEE Transactions on Information Theory* 64.12 (Dec. 2018), pp. 7511–7528. ISSN: 0018-9448, 1557-9654. DOI: 10.1109/TIT.2018.2868701.

[YTW+22] B. Yan et al. *Factoring Integers with Sublinear Resources on a Superconducting Quantum Processor*. Version Number: 1. 2022. DOI: 10.48550/ARXIV.2212.12372.