Higher Order MDS Codes for Combinatorial List Decoding and Distributed Multi-User Secret Sharing

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in Electronics and Communication Engineering by Research

by

Harshithanjani Athi
20171199
harshithanjani.athi@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
November 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Higher Order MDS Codes for Combinatorial List Decoding and Distributed Multi-User Secret Sharing" by Harshithanjani Athi, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Advisor: Dr. Lalitha Vadlamani

To

my parents,

my teachers and

my friends

Acknowledgments

I would like to thank my advisor Prof. Lalitha Vadlamani for her constant motivation and for pushing me to surpass my limits. Her guidance and expertise have been instrumental in shaping the direction of my research and overall development. I am grateful for her patience, encouragement and the trust she placed in me.

I am also grateful to my co-advisor Prof. Prasad Krishnan for his patience, constant insights and valuable guidance throughout my master's program. Working with him has been an enriching experience and I have learned a great deal from his expertise. His passion for the subject matter and his ability to deliver complex topics in a clear and engaging manner made the learning experience both enjoyable and rewarding. I would also like to extend my gratitude to Prof. Nikhil Karamchandani of IIT Bombay for his patience and valuable insights throughout my research journey. It has been a wonderful experience collaborating with him and I sincerely appreciate his guidance and support. I am also grateful to Dr. Myna Vajha for her insights while working on the Decoding of Reed-Muller codes.

I wish to extend my heartfelt gratitude to Qualcomm for their generous support of our research through the Qualcomm Innovation Fellowship 2022. I am thankful to my mentors Ashutosh Gore and Ahmed Zaki for their invaluable insights and guidance throughout the duration of the fellowship.

I am thankful to Rasagna for the collaborative efforts we have shared. Her insights have played a crucial role in propelling our work forward at every stage. I would also like to extend my gratitude to Jayadev and Praneeth for their valuable contributions in the form of helpful technical discussions. Whenever I found myself stuck, their insights and guidance were invaluable in overcoming challenges and progressing in my research.

I extend my heartfelt thanks to Varsha for her constant encouragement and inspiration. She has been a source of motivation, reminding me that great things can be achieved through dedication and perseverance. Our collaborative projects and the cherished memories of our long phone call gossip sessions during the challenging times of the COVID-19 pandemic will forever hold a special place in my heart. Furthermore, I would like to express my gratitude to Ruchitha for being a pillar of support and providing me with a sense of belonging away from home. Her caring nature and unwavering mental support have been crucial in keeping me motivated during the ups and downs of my academic journey. The late-night game sessions, gossip sessions and light-hearted moments we shared have not only brought joy and laughter but have also created enduring memories that I will treasure for a lifetime. A special thank you goes to my roommate Deepti for being a source of comfort and joy throughout my time in college.

Our late-night gossip sessions and the moments we shared over chocolates created a warm and friendly environment that I will cherish forever.

I would like to express my heartfelt gratitude to Niharika, Thejasvi and Sravani for enriching my college life with wonderful memories. Thanks to Sireesha and Meghana for making my final year in the college an absolute joy. The laughter, shared experiences, and bonds we formed will always hold a special place in my heart. Your presence has made my journey truly memorable and enjoyable.

I am grateful to Surendra and Mahee Surya for being constant pillars of support during my time away from home. Their encouragement and unwavering support during moments of extreme pressure and stress have been a source of strength for me. I will always remember their kindness and the impact it had on my journey.

Finally, I would like to express my sincere gratitude to my parents for their unwavering support and guidance throughout my academic journey. Their encouragement and belief in me have been invaluable and I am forever grateful for their love and affection.

To all those mentioned above and to anyone else who has played a part in shaping my journey, thank you from the bottom of my heart. Your support, encouragement and companionship have been truly invaluable and I am deeply grateful for your presence in my life.

Abstract

In this thesis two problems have been considered. The first one is related to the structure of higher order MDS codes which are a generalization of traditional MDS codes and the second one is related to distributed multi-user secret sharing.

When data is transmitted over noisy channels, errors can occur, corrupting the received information. Traditional error-correcting codes like Reed-Solomon or Hamming codes are designed to correct a specific number of errors. However, they have a limitation that they can only correct up to a certain number of errors beyond which they fail to recover the original message. In many scenarios, such as wireless communication or storage systems, the number of errors can be quite high, and standard error-correction techniques might not be sufficient. This is where list decoding, proposed independently by Elias and Wozencraft, comes into play. Instead of trying to pinpoint the exact original message, list decoding provides a list of possible messages that could have been transmitted. By doing so, it offers a more flexible and robust approach to error correction, allowing for a higher number of correctable errors. The decoding radius is the maximum number of errors that can be corrected by the code. By increasing the decoding radius and allowing for a list of possible codewords, list decodable codes can handle a higher number of errors and provide a more robust approach to error correction. In a recent work by Shangguan and Tamo, an upper bound on the size of a list deocodable codes, called the Generalized Singleton bound, is derived for a given list size and decoding radius. The codes which meet this bound with equality are optimally list decodable codes.

In our first problem, we study a recently introduced class of higher order MDS codes which are closely related to the optimally list decodable codes. For certain parameter regimes, we identify conditions under which performing expurgation and shortening (see Section 2.2) like operations on the higher order MDS codes based on Reed-Solomon codes obtained using Vandermonde matrix results in new higher order MDS codes with a related set of parameters. More specifically, we identify the conditions under which (n_1, k_1)-MDS(ℓ_1) codes can be obtained from (n_2, k_2)-MDS(ℓ_2) codes via various techniques where (n, k)-MDS(ℓ) codes denote higher order MDS codes of length n, dimension k and order $\ell \geq 1$. We also obtain a new field size upper bound for the existence of such codes which arguably improves over the best known existing bound in some parameter regimes. We believe that these results will aid in efficient constructions of higher order MDS codes.

In the second problem, we focus on a Distributed Multi-User Secret Sharing (DMUSS) problem. Distributed Multi-User Secret Sharing is motivated by the need for secure and efficient ways to share

sensitive information among multiple users in a decentralized manner. The traditional secret sharing scheme involves a single dealer dividing a secret into shares and distributing them among a group of users. However, this centralized approach has limitations, especially when dealing with large-scale systems or scenarios with multiple secrets and a large number of users. In DMUSS, each user can request access to a specific subset of secrets. This access control mechanism ensures that users only gain access to the secrets they are authorized to see. In this thesis we consider a DMUSS setting involving a dealer, n storage nodes and m secrets. Each user requests a subset of t out of the m secrets. Previous work in this setting addressed the case of $t = 1$ which we extend to handle general values of t. The users download shares from storage nodes based on the access structure and reconstruct their secrets. We establish a necessary condition on the access structure to ensure certain privacy conditions. Additionally, we establish a connection between access structures and *disjunct* matrices which are majorly used in the *Group testing*. In the Distributed Secret Sharing Protocol proposed for our setting, we utilize various disjunct matrix constructions and compare their performance in terms of the number of storage nodes and communication complexity. Moreover, we derive bounds on the optimal communication complexity of a distributed secret sharing protocol.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is based on two research problems. In the first part of this thesis, we study the structure of a newly introduced class of codes called higher order MDS codes which are a generalization of traditional MDS codes (Chapter 2) while in the second part we study the setting of a distributed multi-user secret sharing protocol (Chapter 3). In this chapter, we introduce some preliminaries as well as give an overview of our results.

The higher order MDS codes are closely connected to optimally list-decodable codes and maximally recoverable tensor codes (see Sub-section 1.2.2). List decoding and distributed multi-user secret sharing play vital roles in various cryptographic applications ranging from enhancing the security of cryptographic schemes to enabling secure multi-party computations and threshold cryptography. List decoding is utilized in post-quantum cryptography, enabling the construction of secure lattice-based encryption and digital signature schemes that withstand quantum attacks [1, 2]. Furthermore, list decoding can be employed in cryptanalysis to recover sensitive information when error rates are high enough. On the other hand, Distributed multi-user secret sharing is a crucial cryptographic technique that involves dividing a secret into multiple shares and distributing them among users. This method ensures data protection, privacy and fault tolerance by allowing the original secret to be reconstructed only when a sufficient number of shares collaborate. It finds applications in key management, secure communication, access control, digital signatures, secure multi-party computation and blockchain systems. By leveraging the collective effort of participants, distributed multi-user secret sharing enhances the security and resilience of cryptographic protocols and real-world systems, safeguarding sensitive information and cryptographic keys from unauthorized access and potential compromise.

Before delving into the prior work on both topics, let us establish some notation that will be consistently used throughout this thesis.

1.1 Notation

Let $\mathbb{F}$ be the finite field with q elements (we suppress the field size unless required explicitly). The notation $[n]$ denotes $\{1, \ldots, n\}$ and for $n_1, n_2 \in \mathbb{N} \cup \{0\}$, $n_1 \leq n_2$, define $[n_1 : n_2]$ as the set of

$\{n_1, n_1+1, \ldots, n_2\}$. The binomial coefficient with parameters m, r is denoted by $\binom{m}{r}$. A linear (n,k)-code $\mathcal{C}$ over $\mathbb{F}$ is a k-dimensional subspace of $\mathbb{F}^n$. We denote the dual code of $\mathcal{C}$ by $\mathcal{C}^\perp$, which is an $(n, n-k)$-code. The zero-vector is denoted by $\mathbf{0}$, which we also slightly abuse for denoting the zero-dimensional subspace of a vector space. For a matrix V with n columns over a field and subset $A \subseteq [n]$, we denote by V_A the span of columns of V indexed by A. For convenience, we also abuse this notation V_A slightly to occasionally denote the submatrix of V consisting of the columns in A (the exact meaning of V_A should be clear from the context). The phrase 'without loss of generality' is abbreviated as WLOG.

1.2 Higher Order MDS Codes

1.2.1 Background

The notion of list decoding was introduced in [3] as a generalization of unique decoding and helps in handling a greater number of errors than that allowed by unique decoding. A code of length n is said to be (combinatorially) (ρ, L)-list decodable if the Hamming ball of radius ρn around any vector in the ambient space does not contain more than L codewords. List decoding is an essential tool in capacity achieving codes in various channels, with applications in various fields like complexity theory and cryptography.

List decoding of Reed-Solomon (RS) codes is of specific interest, since RS codes are known to be Maximum Distance Separable (MDS) codes, i.e., they have the largest possible rate for a given minimum distance. A celebrated work of Guruswami and Sudan [4] showed that one could efficiently list decode RS codes (via a Berlekamp-Welch type polynomial interpolation algorithm) up to the so-called Johnson radius [5], a lower bound on the decoding radius for which polynomial list sizes are guaranteed. Formally, Johnson Bound is stated as follows:

Lemma 1.1. *Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a code with minimum distance d. If $\rho < J_q(\frac{d}{n})$, then $\mathcal{C}$ is a $(\rho, \rho dn)$-list decodable code, where $J_q(\delta)$ is defined as*

$$J_q(\delta) = \left(1 - \frac{1}{q}\right)\left(1 - \sqrt{1 - \frac{q\delta}{q-1}}\right)$$

Continuing this line of research, it was shown in [6] that there exists a large class of RS codes that are list decodable (that is, have small list sizes) beyond the Johnson radius as well.

Deviating from previous approaches of obtaining efficient algorithms for list decoding, recent work (for instance, [7, 8, 9]) has focused on the question of *combinatorial* list decodability of codes, especially MDS (and RS) codes.

Formally, a *(combinatorial) list decodable* code is defined as follows:

Definition 1.2. *A code $\mathcal{C}$ of length n over $\mathbb{F}$ is said to be (ρ, L)-list decodable if for every ball of radius ρn around any vector in $\mathbb{F}^n$, there are at most L codewords of $\mathcal{C}$.*

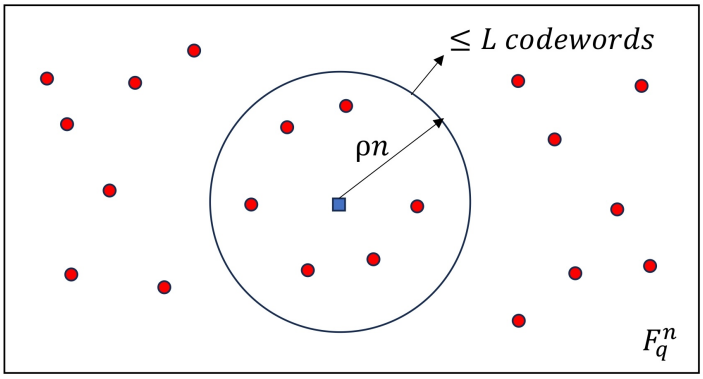

Figure 1.1: (ρ, L)-List decodable code

For $L = 1$, the maximum decoding radius is given by the Singleton bound:

$$\rho \leq \frac{1-R}{2}.$$

Codes for which the Singleton bound is met with equality are called Maximum Distance Separable (MDS) codes. In the paper by Shangguan and Tamo [7], the problem of list decoding RS codes beyond the Johnson radius is studied. They prove a Generalized Singleton bound which gives on upper bound on the size of a list decodable code for a given list size and decoding radius. It was conjectured that the bound is tight for most RS codes over large enough finite fields. They also show that the conjecture holds for list sizes 2 and 3, and as a byproduct, most RS codes with a rate of at least 1/9 are list-decodable beyond the Johnson radius. Lastly, they give the first explicit construction of such RS codes.

The Generalized Singleton bound derived in [7] is stated a follows and is proved here for the sake of completeness:

Proposition 1.3. *(Generalized Singleton bound) If an (n, k)-code $\mathcal{C}$ over an alphabet of size q is (ρ, L)-list-decodable, then*

$$|\mathcal{C}| \leq Lq^{n-\left\lfloor \frac{(L+1)\rho n}{L} \right\rfloor}.$$

Moreover, if $\mathcal{C}$ is a linear code over $\mathbb{F}_q^n$ with $q > L$, then

$$|\mathcal{C}| \leq q^{n-\left\lfloor \frac{(L+1)\rho n}{L} \right\rfloor}.$$

Proof. The proof of the first statement suffices, as the second statement readily derives from it. Consider $a := \left\lfloor \frac{(L+1)\rho n}{L} \right\rfloor = \rho n + \left\lfloor \frac{\rho n}{L} \right\rfloor$ (assuming ρn is an integer). Assume on the contrary that $|\mathcal{C}| >$

Lq^{n-a}. It can be shown that there exists a vector $y \in F_q^n$ satisfying the condition $|B_{\rho n}(y) \cap \mathcal{C}| > L+1$, where $B_{\rho n}(y)$ denotes the set of all vectors that are in a hamming ball of radius ρn around the vector y. This directly contradicts the (ρ, L) list-decodability requirement.

According to the pigeonhole principle, there must exist a minimum of $L+1$ distinct codewords, $c_1, c_2, ..., c_{L+1}$ of $\mathcal{C}$ that share the same values for their first $n-a$ coordinates, i.e., $(c_1)_{[n-a]} = (c_2)_{[n-a]} = \ldots = (c_{L+1})_{[n-a]}$. Partition arbitrarily the set of last a coordinates $\{n-a+1, \ldots, n\}$ as evenly as possible to $L+1$ subsets $I_1, \ldots, I_{L+1}$ each of size $\left\lfloor \frac{a}{L+1} \right\rfloor$ or $\left\lceil \frac{a}{L+1} \right\rceil$, and define the vector $y \in F_q^n$ to be

$$(y)_{[n-a]} = (c_1)_{[n-a]} \text{ and } (y)_{I_i} = (c_i)_{I_i} \text{ for } 1 \leq i \leq L+1.$$

Clearly, the vector y is well defined as $[n-a], I_1, \ldots, I_{L+1}$ form a partition of $[n]$. Moreover, for $1 \leq i \leq L+1$ the hamming distance between c_i and y is at most

$$\begin{aligned} a - |I_i| \leq a - \left\lfloor \frac{a}{L+1} \right\rfloor &= \rho n + \left\lfloor \frac{\rho n}{L} \right\rfloor - \left\lfloor \frac{\rho n}{L+1} + \left\lfloor \frac{\rho n}{L} \right\rfloor \frac{1}{L+1} \right\rfloor \\ &\leq \rho n + \left\lfloor \frac{\rho n}{L} \right\rfloor - \left\lfloor \left\lfloor \frac{\rho n}{L} \right\rfloor \frac{L}{L+1} + \left\lfloor \frac{\rho n}{L} \right\rfloor \frac{1}{L+1} \right\rfloor = \rho n. \end{aligned}$$

Hence, $\{c_1, c_2, \ldots, c_{L+1}\} \subseteq B_{\rho n}(y)$, which is a contradiction to our assumption that $\mathcal{C}$ is (ρ, L) list-decodable. □

Later in [10], the notion of average radius list decodability is introduced, and the codes that meet the Generalized Singleton bound, stated above, for average radius list decoding are identified as higher order generalizations of MDS codes [8]. The following definition captions the notion of average radius list decodability.

Definition 1.4. *[10] A code $\mathcal{C}$ of length n over $\mathbb{F}$ is said to be (ρ, L)-average-radius list decodable if for every $y \in \mathbb{F}^n$ there are no $L+1$ codewords $c_1, c_2, \ldots, c_{L+1} \in \mathcal{C}$ such that*

$$\sum_{m \in [1:L+1]} wt(y - c_m) \leq (L+1)(\rho n),$$

where $wt(.)$ denotes the Hamming weight.

1.2.2 Higher Order Generalizations of MDS codes

The codes that achieve the Generalized Singleton bound for average-radius list decoding are called List decodable MDS codes, defined as follows.

Definition 1.5. *(List decodable MDS codes [8]) Let $\mathcal{C}$ be an (n, k)-code over a field $\mathbb{F}_q$. For $L < q$, we say that $\mathcal{C}$ is list decodable-MDS(L) (in short LD-MDS(L)), if $\mathcal{C}$ is (ρ, L)-average-radius list decodable for*

$$\rho = \frac{L}{L+1}\left(1 - \frac{k}{n}\right).$$

In other words, for any $y \in \mathbb{F}_q^n$, if $\mathcal{C}$ is a LD-MDS(L) code, there do not exist $L+1$ distinct codewords $c_0, c_1, \ldots, c_L \in \mathcal{C}$ such that

$$\sum_{i=0}^{L} wt(c_i - y) \leq (L+1)\rho n = L(n-k),$$

where $wt(.)$ denotes the Hamming weight.

Note that the usual MDS codes are LD-MDS(1). In [11], a different notion of higher order MDS codes in relation to Maximally Recoverable tensor codes was introduced as a generalization of MDS codes, using the notion of generic matrices. Maximally Recoverable (MR) Tensor Codes, introduced by Gopalan et al. [12], are tensor codes that can correct every erasure pattern that is information theoretically possible to correct[1]. An (m, n, a, b)-tensor code consists of $m \times n$ matrices whose columns satisfy 'a' parity checks and rows satisfy 'b' parity checks. In other words, a tensor code is the tensor product of a column code and row code. Tensor codes are helpful in distributed storage because a single erasure can be corrected quickly either by reading its row or column. Brakenseik et al. in their work [11], mainly focused on MR tensor codes for the particular case where $a = 1$ and showed that constructing MR tensor codes even for this simple case, is quite challenging and leads to some very interesting generalization of MDS codes. Formally, these higher order generalizations of MDS codes are referred to as *higher order MDS* codes and are defined as follows.

Definition 1.6. *(Higher order MDS codes [13]) For a positive integer ℓ, we say that $\mathcal{C}$ is (n, k)-MDS(ℓ) if it has dimension k, length n, and a generator matrix V such that for any ℓ subsets $A_1, \ldots, A_\ell \subseteq [n]$, we have that*

$$dim(V_{A_1} \cap \cdots \cap V_{A_\ell}) = dim(W_{A_1} \cap \cdots \cap W_{A_\ell}),$$

where $W_{k\times n}$ is a generic matrix over the same field characteristic and V_{A_i}, W_{A_i} denote the span of columns of V and W indexed by A_i respectively.

The usual MDS codes are MDS(ℓ) for $\ell = 1, 2$ [11]. The relationship between Maximally Recoverable (MR) tensor codes and higher order MDS codes, as stated in [11], is captured by the following lemma:

[1]We have provided an informal definition of Maximally recoverable codes here, focusing on intuitive understanding rather than precise formalization.

Lemma 1.7. *Let $\mathcal{C} = \mathcal{C}_{col} \otimes \mathcal{C}_{row}$ be an $(m, n, a = 1, b)$-tensor code where $\mathcal{C}_{col}$ is a simple parity check code. Then $\mathcal{C}$ is MR if and only if $\mathcal{C}_{row}$ is an MDS(m) code.*

We say that a code $\mathcal{C}$ is LD-MDS$(\leq L)$ if it is LD-MDS(ℓ) for all $1 \leq \ell \leq L$. The List Decodable MDS codes introduced by Roth [8] and the Higher-order MDS codes introduced by Brakenseik et al. [11] are connected through a duality relation which is stated as follows:

Lemma 1.8. *[13] For all $\ell \leq 1$, a linear code $\mathcal{C}$ is LD-MDS$(\leq \ell)$ if and only if $\mathcal{C}^{\perp}$ is MDS$(\ell + 1)$.*

In a remarkable development, the work [13] showed that generic RS codes are optimally list decodable, using this dual relationship. This implies that there exists RS codes (with evaluation points in a large enough field) that achieve the list-decoding capacity, i.e., meet the Generalized Singleton bound for list-decoding with equality. Many questions remain open, however, regarding the minimum field size required for the existence of higher-order MDS codes. The best known lower bound on the field size of an (n, k)-MDS(ℓ) code is $\Omega_{\ell,k}(n^{min\{\ell,k,n-k\}-1})$ [11], whereas the best known (non-explicit) upper bound is $n^{O(min\{k,n-k\}(\ell-1))}$ [11, 13, 14] which is exponential in the dimension. In [7, 8], explicit constructions for MDS(3) codes are presented over fields of sizes 2^{k^n} and $n^{k^{O(k)}}$ respectively, which are double exponential in n for $k = \Theta(n)$. In a recent work[15], the GM-MDS theorem [2] [16] is used to give an explicit construction for (n, k)-MDS(ℓ) codes over fields of size $n^{(\ell k)^{O(\ell k)}}$. Further, explicit constructions for (n, k)-MDS(3) for $k = 3, 4, 5$ over field sizes $O(n^3)$, $O(n^7)$, $O(n^{50})$ respectively are also provided in [15].

1.2.3 Our Contributions

In Chapter 2, we present some new structural results relating to higher-order MDS codes.

- For certain parameter regimes, we show that (n, k)-MDS(k) codes which are also RS codes (generated by a Vandermonde matrix defined on a set of evaluation points) are closed under the expurgation operation. In other words, the removal of the last row of the generator matrix of such codes results in an $(n, k-1)$-MDS($k-1$) code, for certain parameter regimes. We extend this to MDS(ℓ) codes for general ℓ, under some special conditions.

- For (n, k)-MDS(k) RS codes, we show the preservation of the higher order MDS property on a *pseudo-shortening* operation (combining puncturing and expurgation). Specifically, given the Vandermonde generator matrix of an (n, k)-MDS(k) RS code, we show that removing the last row and any column results in $(n-1, k-1)$-MDS($k-1$) code.

- Using well-known probabilistic tools, we obtain a new field size upper bound for the existence of (n, k)-MDS(ℓ) codes which are arguably better than existing bounds for certain parameter regimes.

[2] Let M be an $m \times n$ binary matrix that satisfies the MDS condition. Then, there exists an $[n, m]_q$ MDS code whose generator matrix G, with entries in $\mathbb{F}_q$, fits M (i.e., M is the support matrix of G) for any field $\mathbb{F}_q$ of size $q \geq n + m - 1$.

1.3 Distributed Multi-User Secret Sharing

1.3.1 Background

A secret sharing scheme is a cryptographic technique that distributes a secret among multiple users while maintaining two key properties: secret recovery, which allows authorized subsets of parties to reconstruct the secret from their shares, and collusion resistance, which ensures that unauthorized subsets of parties cannot learn anything about the secret. These properties are crucial in maintaining the confidentiality and integrity of the secret.

The concept of secret sharing was first introduced by Shamir [17] and Blakley [18] in their independent works. In [17], Shamir proposes a secret sharing scheme based on polynomial interpolation, while in [18], Blakley introduces a secret sharing scheme based on the intersection of subspaces. Secret sharing has been extensively studied and applied in various areas of cryptography and distributed computing, such as secure multi-party computation [19], secure cloud computing [20], secure voting systems [21], to name a few. Moreover, recent advances in secret sharing have enabled its use in emerging technologies such as blockchain [22] and secure multi-party machine learning [23].

In a recent work by Soleymani et al. [24], a distributed multi-user secret sharing (DMUSS) system was considered, which consists of a dealer, n storage nodes, and m users. In this scenario, each user has a designated secret message and is given access to a specific subset of storage nodes, where the user can download the stored data. To ensure that certain privacy conditions are satisfied, the Sperner family [25] is used in obtaining these subsets of storage nodes. The dealer is treated as a master node controlling all the storage nodes. The aim of the dealer is to securely share a specific secret s_j with user j via the storage nodes. Under the multi-user context, two secrecy conditions are considered, and secret sharing schemes that achieve these secrecy conditions are constructed [24]. The *weak secrecy* condition requires that each user does not get any information about the individual secrets of other users, while the *perfect secrecy* condition requires that a user does not get any information about the collection of other users' secrets. Two major properties, namely, the storage overhead and the communication complexity, are defined for such distributed secret-sharing systems. Optimal values for storage overhead and communication complexity were derived for any given m and n, and protocols that achieve these optimal values simultaneously are constructed. The secret sharing protocols proposed in [24] are specific to the case where each user has a designated secret message. In [26], the capacity region of the distributed multi-user secret sharing system under weak secrecy is characterized, where they consider the set-up in which each user can have a secret message of a different size.

A distributed secret sharing protocol is defined as follows:

Definition 1.9. *A distributed secret sharing protocol (DSSP) is a bundle of* $(\mathcal{A}, \mathcal{E}, \boldsymbol{Z}_{n\times h}, \mathcal{D})$*, where*

- $\mathcal{A}$ is the access structure defined in equation (3.1).

- $\mathcal{E} : \mathbb{F}_q^m \to \mathbb{F}_q^h$ with $h \geq m$ is an encoding function which relates to storage overhead of the system. The input $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)^T$ is a vector of all secrets. The output $\mathbf{y} = \mathcal{E}(\boldsymbol{s}) = (y_1, \ldots, y_h)^T$ is a vector of all data (shares) to be distributed and stored in the storage nodes.

- $\boldsymbol{Z} = [z_{i,r}]_{n \times h}$, where $z_{i,r} = 1$ if y_r is stored in i^{th}-storage node, otherwise 0. We denote by $\mathbf{y}_j$ the vector of all shares stored in nodes indexed by the elements of storage set A_j. The matrix $\boldsymbol{Z}$ is referred to as *storing matrix*. The mapping of the output symbols to the storage nodes (specifying which output symbols are stored at each storage node) is referred to as the *storage profile*.

- $\mathcal{D}$ is a collection of m decoding functions $\mathcal{D}_j : \mathbb{F}_q^{|\mathbf{y}_j|} \to \mathbb{F}_q$, such that $\mathcal{D}_j(\mathbf{y}_j) = s_j$. In other words, each user can successfully reconstruct its secrets. This is referred to as *correctness* condition.

In order to assess the efficiency of the proposed DSSPs, we utilize the concepts of "storage overhead" and "communication complexity" as defined in [24]. These terms are consistently employed throughout the paper to evaluate the efficiency of the protocols are recalled below. Note that the total number of $\mathbb{F}_q$-symbols stored in storage nodes is $k' = \sum_{i=1}^{n} \sum_{r=1}^{h} z_{i,r}$, where $\boldsymbol{Z} = [z_{i,r}]_{n \times h}$ is specified in Definition 1.9.

Definition 1.10. *The storage overhead, SO, of the DSSP is defined as*

$$SO \triangleq \frac{k'}{m}. \tag{1.1}$$

Note that the correctness condition must be satisfied for m uniformly distributed and mutually independent secrets. Therefore, $k' \geq m$ and, consequently, $SO \geq 1$.

Definition 1.11. *Let c_u denote the number of symbols user u needs to download from the storage nodes to reconstruct the designated set of secrets S_u. Then the communication complexity C is defined as*

$$C \triangleq \sum_{u=1}^{\binom{m}{t}} c_u. \tag{1.2}$$

1.3.2 Our Contributions

In Chapter 3, we present our results related to distributed multi-user secret sharing where each user requests multiple secrets. We consider a setting where we have multiple users, and each user requests the same number of secrets. Our objective is to devise a secret sharing protocol that provides optimal storage efficiency while ensuring weak secrecy.

- We derive a necessary condition on the access structure of the distributed secret sharing protocol to ensure weak secrecy and establish a relation between the access structure and the t-disjunct matrices, which we later define in Chapter 3 (Definition 3.2).

- Using the access structure obtained from the t-disjunct matrix, we propose a secret sharing protocol that achieves optimal storage overhead.
- Using several constructions for t-disjunct matrices, we compare the system parameters and properties. We also show that t-disjunct matrices obtained using the Steiner system are better than those obtained from other known constructions in terms of accommodating more secrets.
- For the DMUSS system considered in our problem, we provide a range in which the communication complexity lies and derive bounds on the optimal communication complexity.

Chapter 2

On the Structure of Higher Order MDS Codes

In this chapter, we present some structural results for higher-order MDS codes. To set the stage for our work, we recall a brief collection of existing results for higher order MDS codes which will be further used in deriving our results. Later we present some results on the structure of Higher-order MDS codes based on Reed-Solomon codes obtained using the Vandermonde matrix. Finally, we give an upper bound on the field size for the existence of such codes, which arguably improves over the existing bounds in specific parameter regimes. The following definition of generic collections of sets is helpful in recalling some of the existing results on Higher-order MDS codes.

Definition 2.1. *Let ℓ be a positive integer and let n, k be integers such that $n \geq k \geq 0$. Consider $A_1, \ldots, A_\ell \subseteq [n]$ with $|A_i| \leq k$. These sets $(A_1, \ldots, A_\ell)$ are called an (n, k, ℓ)-generic collection if for all partitions $P_1 \cup \cdots \cup P_s = [\ell]$ we have*

$$\sum_{i=1}^{s} \left| \bigcap_{j \in P_i} A_j \right| \leq (s-1)k. \tag{2.1}$$

One of the basic properties of higher order MDS codes is stated in [11] as:

Lemma 2.2. *[11] Let $\mathcal{C}$ be an (n, k)-MDS(ℓ) code. If $\ell \geq 3$, then $\mathcal{C}$ is also an MDS$(\ell - 1)$ code.*

An equivalent definition of Higher-order MDS codes is given in terms of the intersection of sub-matrices is given in [11] as:

Lemma 2.3. *[11] Let V be a $k \times n$ matrix. Then V is (n, k)-MDS(ℓ) if and only if for all (n, k, ℓ)-generic collections $(A_1, \ldots, A_\ell)$ with $|A_1| + \cdots + |A_\ell| = (\ell - 1)k$, we have that $V_{A_1} \cap \cdots \cap V_{A_\ell} = \mathbf{0}$.*

An equivalent determinant-based criterion was also derived, which we recall.

Lemma 2.4. *[27, 11] Let V be a $k \times n$ matrix. Consider $A_1, A_2, \ldots, A_\ell \subseteq [n]$ with $|A_i| \leq k$ and $|A_1| + \cdots + |A_\ell| = (\ell - 1)k$, we have that $V_{A_1} \cap \cdots \cap V_{A_\ell} = \mathbf{0}$ if and only if*

$$\det \begin{pmatrix} I_k & V_{A_1} & & & \\ I_k & & V_{A_2} & & \\ \vdots & & & \ddots & \\ I_k & & & & V_{A_\ell} \end{pmatrix} \neq 0, \tag{2.2}$$

where V_{A_i} denotes the submatrix of V with columns indexed by A_i.

2.1 Properties of Higher Order MDS Codes Based on Vandermonde Matrix

For some $\beta_i \in \mathbb{F}, \forall i \in [n]$, denote the $k \times n$ Vandermonde matrix

$$\mathsf{Vand}_k(\{\beta_i : i \in [n]\}) \triangleq \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \beta_1 & \beta_2 & \cdots & \beta_n \\ \vdots & \vdots & \cdots & \vdots \\ \beta_1^{k-1} & \beta_2^{k-1} & \cdots & \beta_n^{k-1} \end{pmatrix}.$$

For the specific case of $(n, 3)$-RS codes, we have the following simplified determinant conditions.

Lemma 2.5. *[11, 8] Let V be $(n, 3)$-RS code generated using $G = \mathsf{Vand}_3(\{\beta_i : i \in [n]\})$. Then V is MDS(3) if and only if for all injective maps $\alpha : [6] \to [n]$ we have that*

$$\det \begin{pmatrix} 1 & \beta_{\alpha(1)} + \beta_{\alpha(2)} & \beta_{\alpha(1)}\beta_{\alpha(2)} \\ 1 & \beta_{\alpha(3)} + \beta_{\alpha(4)} & \beta_{\alpha(3)}\beta_{\alpha(4)} \\ 1 & \beta_{\alpha(5)} + \beta_{\alpha(6)} & \beta_{\alpha(5)}\beta_{\alpha(6)} \end{pmatrix} \neq 0.$$

In [15], an alternative characterization of the higher-order MDS conditions for Reed-Solomon codes is presented. For $A \subseteq [n]$, define the polynomial

$$\pi_A(x) \triangleq \prod_{i \in A} (x - \beta_i).$$

Define $\pi_A^d(x)$ to be the following (row) vector of polynomials:

$$\pi_A^d(x) \triangleq (\pi_A(x), x\pi_A(x), \ldots, x^{d-1}\pi_A(x)).$$

Then we have the following determinant condition from [15].

Lemma 2.6. *[15] Assume that $A_1, ..A_\ell \in [n]$ such that $|A_i| \leq k, \forall i \in [\ell]$. Let $|A_1| + \cdots + |A_\ell| = (\ell - 1)k$. Let $\delta_i = k - |A_i|$. Assume WLOG that $A_1 = \{1, 2, \ldots, k - \delta_1\}$. We have that $V_{A_1} \cap \cdots \cap V_{A_\ell} = \mathbf{0}$ iff*

$$\det \begin{pmatrix} \pi_{A_2}^{\delta_2}(\beta_1) & \pi_{A_3}^{\delta_3}(\beta_1) & \cdots & \pi_{A_\ell}^{\delta_\ell}(\beta_1) \\ \pi_{A_2}^{\delta_2}(\beta_2) & \pi_{A_3}^{\delta_3}(\beta_2) & \cdots & \pi_{A_\ell}^{\delta_\ell}(\beta_2) \\ \vdots & \vdots & \cdots & \vdots \\ \pi_{A_2}^{\delta_2}(\beta_{k-\delta_1}) & \pi_{A_2}^{\delta_3}(\beta_{k-\delta_1}) & \cdots & \pi_{A_\ell}^{\delta_\ell}(\beta_{k-\delta_1}) \end{pmatrix} \neq 0. \tag{2.3}$$

2.2 On Higher Order MDS Codes Based on RS Codes

In this section, we present the results that concern the properties of higher-order MDS codes based on Reed-Solomon codes generated from the Vandermonde matrix. We show that (n, k)-MDS(k) RS codes are closed under expurgation operation under certain parameter regimes and extend this to MDS(ℓ) codes for general ℓ. We also show the preservation of the higher order MDS property on a pseudo-shortening operation on the generator matrix of an RS (n, k)-MDS(k) code.

In the following Lemma, we show that removing the last row (*expurgation operation*) from the generator matrix of an MDS(k) Reed-Solomon code results in an MDS$(k-1)$ code.

Lemma 2.7. *Let V be an (n, k)-RS code generated using $G = \mathsf{Vand}_k(\{\beta_i : i \in [n]\})$ such that V is (n, k)-MDS(k) and $n \geq (k-2)(k-1)+1$. Then the code V' generated by $\mathsf{Vand}_{k-1}(\{\beta_i : i \in [n]\})$ is an $(n, k-1)$-MDS$(k-1)$ code.*

Proof. Suppose V is an (n, k)-MDS(k) code but V' is not $(n, k-1)$-MDS$(k-1)$ code. Then by Lemma 2.3, there exists some $(n, k-1, k-1)$-generic collection $A_1^{'}, .., A_{k-1}^{'} \subseteq [n]$, with

$$|A_1^{'}| + |A_2^{'}| + .. + |A_{k-1}^{'}| = (k-2)(k-1), \tag{2.4}$$

such that $V_{A_1^{'}}^{'} \cap ... \cap V_{A_{k-1}^{'}}^{'} \neq \mathbf{0}$. Now, we construct a (n, k, k)-generic collection $A_1, .., A_k \subseteq [n]$, using $A_1^{'}, .., A_{k-1}^{'}$ in the following way. Let $p \in [n]$ and $p \notin (A_1^{'} \cup .. \cup A_{k-1}^{'})$. Such a p exists as $n \geq (k-2)(k-1)+1$. Consider, $A_i = A_i^{'} \cup \{p\}, \forall i \in [k-1]$ and let $A_k = A_x^{'} \cup \{$one element from $(A_1^{'} \cup .. \cup A_{k-1}^{'}) \setminus A_x^{'}\}$, where $A_x^{'}$ is of size $(k-2)$ (such an $A_x^{'}$ of size $(k-2)$ exists, otherwise (2.4) will not be satisfied). Now, $|A_i| \leq k-1+1 = k$, $i \in [k-1]$, $|A_k| = k-1$. From definition (2.1), for all partitions $P_1 \cup ... \cup P_s = [k-1]$, we have,

$$\sum_{i=1}^{s} \left| \bigcap_{j \in P_i} A_j^{'} \right| \leq (s-1)(k-1).$$

As $A_i = A_i' \cup \{p\}, \forall i \in [k-1]$, we get

$$\sum_{i=1}^{s} \left| \bigcap_{j \in P_i} A_j \right| \leq (s-1)(k-1) + s = sk - k + 1. \tag{2.5}$$

Now consider a partition $Q_i : i \in [s']$ of $[k]$. Suppose that there exists a $Q_{i'}$ such that element $k \in Q_{i'}$ and $|Q_{i'}| \geq 2$. Then, the collection $\{Q_i : i \in [s'] \setminus \{i'\}\} \cup \{Q_{i'} \setminus \{k\}\}$ forms a partition of $[k-1]$. Since A_k does not contain the element p, by (2.5) we have that

$$\sum_{i=1}^{s'} \left| \bigcap_{j \in Q_i} A_j \right| \leq s'k - k = (s'-1)k. \tag{2.6}$$

If there however is no such $Q_{i'}$ such that $k \in Q_{i'}$ and $|Q_{i'}| \geq 2$, then there should be a $Q_{i''}$ such that $k \in Q_{i''}$ and $|Q_{i''}| = 1$. In that case, the collection $\{Q_i : i \in [s']\} \setminus \{Q_{i''}\}$ forms a partition of $[k-1]$. Hence, again using (2.5), we see that

$$\begin{aligned} \sum_{i=1}^{s'} \left| \bigcap_{j \in Q_i} A_j \right| &= \sum_{i=1:\ i \neq i''}^{s'} \left| \bigcap_{j \in Q_i} A_j \right| + |A_k| \\ &\leq (s'-1)k - k + 1 + (k-1) = (s'-1)k, \end{aligned} \tag{2.7}$$

as $|A_k| = k - 1$.

From equations (2.6) and (2.7) we can say that $A_1, A_2, .., A_k$ form a (n, k, k)-generic collection.

We now show that for the above obtained (n, k, k)-generic collection, $\cap_{i=1}^{k} V_{A_i} \neq \mathbf{0}$, thus showing that the code V is not MDS(k), leading to a contradiction with the given statement.

From Lemma 2.6, assuming WLOG $A_k = \{1, \ldots, k-1\}$, we have $V_{A_1} \cap ... \cap V_{A_k} = \mathbf{0}$ if and only if

$$\det \begin{pmatrix} \pi_{A_1}^{\delta_1}(\beta_1) & \pi_{A_2}^{\delta_2}(\beta_1) & .. & \pi_{A_{k-1}}^{\delta_{k-1}}(\beta_1) \\ \pi_{A_1}^{\delta_1}(\beta_2) & \pi_{A_2}^{\delta_2}(\beta_2) & .. & \pi_{A_{k-1}}^{\delta_{k-1}}(\beta_2) \\ . & . & .. & . \\ . & . & .. & . \\ \pi_{A_1}^{\delta_1}(\beta_{k-1}) & \pi_{A_2}^{\delta_2}(\beta_{k-1}) & .. & \pi_{A_{k-1}}^{\delta_{k-1}}(\beta_{k-1}) \end{pmatrix} \neq 0 \tag{2.8}$$

We now show that the determinant in (2.8) is zero so that $V_{A_1} \cap ... \cap V_{A_k} \neq \mathbf{0}$, which implies that our assumption is incorrect. As $p \in (A_1 \cap A_2 \cap .. \cap A_{k-1})$, the equation (2.9) holds, where $\delta_i^{'} = k-1-|A_i'|$.

$$\det \begin{pmatrix} \pi_{A_1}^{\delta_1}(\beta_1) & \pi_{A_2}^{\delta_2}(\beta_1) & .. & \pi_{A_{k-1}}^{\delta_{k-1}}(\beta_1) \\ \pi_{A_1}^{\delta_1}(\beta_2) & \pi_{A_2}^{\delta_2}(\beta_2) & .. & \pi_{A_{k-1}}^{\delta_{k-1}}(\beta_2) \\ . & . & .. & . \\ . & . & .. & . \\ \pi_{A_1}^{\delta_1}(\beta_{k-1}) & \pi_{A_2}^{\delta_2}(\beta_{k-1}) & .. & \pi_{A_{k-1}}^{\delta_{k-1}}(\beta_{k-1}) \end{pmatrix} = \prod_{j=1}^{k-1}(\beta_j - \beta_p) \times \det \begin{pmatrix} \pi_{A_1'}^{\delta_1'}(\beta_1) & \pi_{A_2'}^{\delta_2'}(\beta_1) & .. & \pi_{A_{k-1}'}^{\delta_{k-1}'}(\beta_1) \\ \pi_{A_1'}^{\delta_1'}(\beta_2) & \pi_{A_2'}^{\delta_2'}(\beta_2) & .. & \pi_{A_{k-1}'}^{\delta_{k-1}'}(\beta_2) \\ . & . & .. & . \\ . & . & .. & . \\ \pi_{A_1'}^{\delta_1'}(\beta_{k-1}) & \pi_{A_2'}^{\delta_2'}(\beta_{k-1}) & .. & \pi_{A_{k-1}'}^{\delta_{k-1}'}(\beta_{k-1}) \end{pmatrix} \tag{2.9}$$

Since $\beta_j \neq \beta_p, \forall j \in A_k$, the term $\prod_{j=1}^{k-1}(\beta_j - \beta_p)$ on RHS of (2.9), is non-zero. From Lemma 2.6, we have that the determinant on RHS is 0 as $V_{A_1'}^{'} \cap ... \cap V_{A_{k-1}'}^{'} \neq \mathbf{0}$. So, the determinant on LHS is also 0, which implies $V_{A_1} \cap ... \cap V_{A_k} \neq \mathbf{0}$ (again by invoking Lemma 2.6), which is a contradiction to our assumption. This completes the proof. □

Lemma 2.8 extends the result of Lemma 2.7 to more general ℓ, under some conditions.

Lemma 2.8. *Let V be (n,k)-RS code generated using $G = \mathsf{Vand}_{k+1}(\{\beta_i : i \in [n]\})$ such that V is $(n,k+1)$-MDS(ℓ) and $n \geq (\ell-1)k+1$. Then the code V' generated by $\mathsf{Vand}_k(\{\beta_i : i \in [n]\})$ is an (n,k)-MDS(ℓ) code.*

Proof. Suppose V is $(n,k+1)$-MDS(ℓ) and V' is not (n,k)-MDS(ℓ). Then by Lemma 2.3, there exists an (n,k,ℓ)-generic collection $(A_1^{'},..,A_\ell^{'})$ with $|A_1^{'}|+|A_2^{'}|+...|A_\ell^{'}| = (\ell-1)k$ such that $V_{A_1'}^{'} \cap ... \cap V_{A_\ell'}^{'} \neq \mathbf{0}$. Observe that this means that $|A_i'| \geq 1, \forall i$. Now, we construct a $(n,k+1,\ell)$-generic collection $(A_1,...A_\ell)$ from $(A_1^{'},..,A_\ell^{'})$ such that $|A_1| + |A_2| + ...|A_\ell| = (\ell-1)(k+1)$. Let $p \in [n]$ and $p \notin (A_1^{'} \cup .. \cup A_\ell^{'})$. Such a p exists as $n \geq (\ell-1)(k)+1$. Consider $A_i = A_i^{'} \cup \{p\}, \forall i \in [\ell-1]$ and $A_\ell = A_\ell^{'}$. As $(A_1^{'},..,A_\ell^{'})$ is a (n,k,ℓ)-generic collection,from definition (2.1), for any partition $P_1 \cup ... \cup P_s = [\ell]$, we have,

$$\sum_{i=1}^{s} \left| \bigcap_{j \in P_i} A_j^{'} \right| \leq (s-1)k. \tag{2.10}$$

Observe that $|A_i| = |A_i'| + 1$, forall $i \in [\ell-1]$, while $|A_\ell| = |A_\ell'|$. Further $p \in \cap_{i=1}^{\ell-1} A_i \setminus A_\ell$. Thus, $\left|\bigcap_{j\in P_i} A_j\right| = \left|\bigcap_{j\in P_i} A_j'\right| + 1$ if $\ell \notin P_i$, while $\left|\bigcap_{j\in P_i} A_j\right| = \left|\bigcap_{j\in P_i} A_j'\right|$ for that P_i such that $\ell \in P_i$.

Thus we get

$$\sum_{i=1}^{s}\left|\bigcap_{j\in P_i} A_j\right| \leq (s-1)k+(s-1)$$
$$\leq (s-1)(k+1) \tag{2.11}$$

From (2.11) we can say that $A_1,..A_\ell$ form a $(n,k+1,\ell)$-generic collection. WLOG, let $A_\ell = \{1,\ldots,k+1-\delta_\ell\}$, for $\delta_\ell = (k+1-|A_\ell|) \in \{0,\ldots,k\}$ (as $|A_\ell| = |A'_\ell| \geq 1$) and $\delta_\ell = k+1-|A_\ell| = \delta'_\ell + 1$. From Lemma 2.6, we have $V_{A_1} \cap ... \cap V_{A_\ell} = \mathbf{0}$ if and only if

$$\det\begin{pmatrix} \pi^{\delta_1}_{A_1}(\beta_1) & \pi^{\delta_2}_{A_2}(\beta_1) & .. & \pi^{\delta_{\ell-1}}_{A_{\ell-1}}(\beta_1) \\ \pi^{\delta_1}_{A_1}(\beta_2) & \pi^{\delta_2}_{A_2}(\beta_2) & .. & \pi^{\delta_{\ell-1}}_{A_{\ell-1}}(\beta_2) \\ . & . & .. & . \\ . & . & .. & . \\ \pi^{\delta_1}_{A_1}(\beta_{k+1-\delta_\ell}) & \pi^{\delta_2}_{A_2}(\beta_{k+1-\delta_\ell}) & .. & \pi^{\delta_{\ell-1}}_{A_{\ell-1}}(\beta_{k+1-\delta_\ell}) \end{pmatrix} \neq 0, \tag{2.12}$$

We now show that the determinant in (2.12) is zero so that $V_{A_1} \cap ... \cap V_{A_\ell} \neq \mathbf{0}$, leading to a contradiction with the given fact that V is $(n,k+1)$-MDS(ℓ). As $p \in (A_1 \cap A_2 \cap .. \cap A_{\ell-1})$, the equation (2.13) holds.

$$\det\begin{pmatrix} \pi^{\delta_1}_{A_1}(\beta_1) & \pi^{\delta_2}_{A_2}(\beta_1) & .. & \pi^{\delta_{\ell-1}}_{A_{\ell-1}}(\beta_1) \\ \pi^{\delta_1}_{A_1}(\beta_2) & \pi^{\delta_2}_{A_2}(\beta_2) & .. & \pi^{\delta_{\ell-1}}_{A_{\ell-1}}(\beta_2) \\ . & . & .. & . \\ . & . & .. & . \\ \pi^{\delta_1}_{A_1}(\beta_{k+1-\delta_\ell}) & \pi^{\delta_2}_{A_2}(\beta_{k+1-\delta_\ell}) & .. & \pi^{\delta_{\ell-1}}_{A_{\ell-1}}(\beta_{k+1-\delta_\ell}) \end{pmatrix}$$
$$= \prod_{j=1}^{k+1-\delta_\ell}(\beta_j - \beta_p) \times \det\begin{pmatrix} \pi^{\delta'_1}_{A'_1}(\beta_1) & \pi^{\delta'_2}_{A'_2}(\beta_1) & .. & \pi^{\delta'_{\ell-1}}_{A'_{\ell-1}}(\beta_1) \\ \pi^{\delta'_1}_{A'_1}(\beta_2) & \pi^{\delta'_2}_{A'_2}(\beta_2) & .. & \pi^{\delta'_{\ell-1}}_{A'_{\ell-1}}(\beta_2) \\ . & . & .. & . \\ . & . & .. & . \\ \pi^{\delta'_1}_{A'_1}(\beta_{k-\delta'_\ell}) & \pi^{\delta'_2}_{A'_2}(\beta_{k-\delta'_\ell}) & .. & \pi^{\delta'_{\ell-1}}_{A'_{\ell-1}}(\beta_{k-\delta'_\ell}) \end{pmatrix}. \tag{2.13}$$

Since $\beta_j \neq \beta_p, \forall j \in A_k$, the first term of RHS of (2.13) is non-zero. From Lemma 2.6, We have that the determinant on RHS is 0 as $V'_{A'_1} \cap ... \cap V'_{A'_\ell} \neq \mathbf{0}$. So, the determinant on LHS is also 0, which implies $V_{A_1} \cap ... \cap V_{A_\ell} \neq \mathbf{0}$ (again using Lemma 2.6), which is a contradiction to our assumption. □

In the following lemma, we show that removing the last row and any column (*expurgation and puncturing together, an operation we have called pseudo-shortening*) from the generator matrix of an MDS(k) RS code also results in an MDS$(k-1)$ code.

Lemma 2.9. *Let V be (n,k)-RS code generated using $G = \mathsf{Vand}_k(\{\beta_i : i \in [n]\})$ such that V is (n,k)-MDS(k). Then, for any $j \in [n]$, the code V' generated by $\mathsf{Vand}_{k-1}(\{\beta_i : i \in [n]\} \setminus \beta_j)$ is an $(n-1,k-1)$-MDS($k-1$) code.*

Proof. WLOG, we assume that the column being left out is $j = n$. Suppose V is an (n,k)-MDS(k) code but V' is not $(n-1,k-1)$-MDS$(k-1)$. Then there exists some $(n-1,k-1,k-1)$-generic collection $A_1^{'},..,A_{k-1}^{'} \subseteq [n-1]$, with

$$|A_1^{'}| + |A_2^{'}| + .. + |A_{k-1}^{'}| = (k-2)(k-1), \tag{2.14}$$

such that $V'_{A_1^{'}} \cap \ldots \cap V'_{A_{k-1}^{'}} = \mathbf{0}$. Clearly, there exists an element $p \in [n] \setminus (A_1^{'} \cup .. \cup A_{k-1}^{'})$, as we have $A_i^{'} \subset [n-1], \forall i \in \{1,\ldots,k-1\}$. Now, using the arguments identical to the proof of Lemma 2.7 using this element p, we can construct a (n,k,k)-generic collection $A_1,..,A_k \subseteq [n]$ with

$$|A_1| + |A_2| + .. + |A_k| = (k-1)k, \tag{2.15}$$

such that such that $V_{A_1} \cap \ldots \cap V_{A_{k-1}} = \mathbf{0}$. By Lemma 2.6, this leads to a contradiction, as we are given that V is (n,k)-MDS(k). This completes the proof. □

2.3 A New Field Size Bound

In this section, we show a new upper bound for the field size of (n,k)-MDS(ℓ) codes. To do this, we essentially rely upon some probabilistic tools, specifically Lovasz's Local Lemma and Schwarz Zippel Lemma, which we now recall.

Lemma 2.10. ***(Schwarz Zippel Lemma)*** *Let $p(x_1,\ldots,x_n)$ be a non-zero polynomial of n variables with degree d with coefficients from a field $\mathbb{F}$. Let S be a finite subset of $\mathbb{F}$, with at least d elements in it. If we assign $x_1,\ldots,x_n$ values from S independently and uniformly at random, then*

$$Pr[p(x_1,\ldots,x_n) = 0] \leq \frac{d}{|S|}.$$

Lemma 2.11. ***(Lovasz Local Lemma** [28]) Let $B_1, B_2, \ldots, B_n$ be events in an arbitrary probability space. Suppose that each event B_i is mutually independent of a set of all other events B_j but at most d,*

and that $Pr(B_i) \leq p$ *for all* $1 \leq\ i \leq n$. *If*

$$e \cdot p \cdot (d+1) \leq 1$$

then $Pr(\bigcap_{i=1}^{n} \bar{B}_i) > 0$, *where* $\bar{B}_i$ *denotes the complement of event* B_i, *and* e *denotes the base of the natural logarithm.*

In the context of Lemma 2.11, we use the phrase 'bad' events to represent the events B_i. Thus, what Lemma 2.11 assures us is that if the probability of any bad event is small enough and any bad event is not dependent on too many others, then there is a positive probability that none of them occur.

We are now ready to present our results. Let $\mathfrak{A}$ denote the set of all collection of sets $\underline{A} = \{\{A_1, A_2, \ldots, A_\ell\} : A_j \subset [n], \forall j\}$ that satisfy the condition in Lemma 2.4. Throughout this section, we consider that the matrix $V = \mathsf{Vand}_k(\{\beta_i : i \in [n]\})$.

Claim 2.12. *The determinant in (2.2) in Lemma 2.4 corresponding to the matrix* V, *as a polynomial in* $\beta_i : i \in [n]$, *has total degree* $\leq \ell k^2$.

Proof. As V matrix is obtained using Vandermonde construction, each V_{A_i} ($|A_i| \leq k$) in (2.2) will be of the form

$$V_{A_i} = \begin{pmatrix} 1 & \ldots & \ldots & 1 \\ \beta_{i_1} & . & . & \beta_{i_{|A_i|}} \\ \beta_{i_1}^2 & \ldots & . & \beta_{i_{|A_i|}}^2 \\ . & . & . & . \\ \beta_{i_1}^{k-1} & \ldots & \ldots & \beta_{i_{|A_i|}}^{k-1} \end{pmatrix},$$

where $A_i = \{i_1, \ldots, i_{|A_i|}\}$. If the matrix V_{A_i} is square (i.e., $|A_i| = k$), then the degree of the determinant of the above matrix as a polynomial in $\beta_1, .., \beta_n$ is at most $k(k-1)/2$. In the determinant for the matrix in (2.2), ℓ such submatrices (V_{A_i}s) are involved. So, the total degree of the determinant as a polynomial in $\beta_1, .., \beta_n$ is at most $\ell k(k-1)/2 \leq \ell k^2$. □

For $\underline{A} = \{A_1, \ldots, A_\ell\}, \underline{A}' = \{A'_1, \ldots, A'_\ell\} \in \mathfrak{A}$, we say that $\underline{A} \not\perp\!\!\!\perp \underline{A}'$ if there is at least one $s \in \left(\cup_j^\ell A_j\right) \cap (\cup_j^\ell A'_j)$. We then say that $\underline{A}$ and $\underline{A}'$ are dependent (on each other). For $\underline{A} \in \mathfrak{A}$, let $E_{\underline{A}}$ denote the event that the determinant given by (2.2) in Lemma 2.4 is zero, for the collection $\underline{A}$. We then have the following claim.

Claim 2.13. *If* $q \geq e \cdot \ell k^2 \cdot \left(\max_{\underline{A} \in \mathfrak{A}} |\{\underline{A}' \in \mathfrak{A} \colon \underline{A} \not\perp\!\!\!\perp \underline{A}'\}|\right)$, *then there exist evaluation points* $\beta_i : i \in [n]$ *such that the code generated by* $\mathsf{Vand}_k(\{\beta_i : i \in [n]\})$ *is MDS(*ℓ*).*

Proof. We use the Lemma 2.11 with the bad events being the events $E_{\underline{A}} : \underline{A} \in \mathfrak{A}$. By Claim 2.12 and the Schwarz Zippel Lemma, the probability that the determinant (2.2) in Lemma 2.4 is 0, for any given $\underline{A} \in \mathfrak{A}$, when choosing the evaluation points independently and uniformly at random from field $\mathbb{F}$ (with q elements) is at the most $\frac{\ell k^2}{q}$. Thus, the probability of any bad event $E_{\underline{A}}$ is at the most $\frac{\ell k^2}{q}$. Invoking Lemma 2.11, the proof is complete. □

The following claim gives an upper bound to the quantity $\max_{\underline{A}\in\mathfrak{A}} |\{\underline{A}' \in \mathfrak{A} \colon \underline{A} \not\perp \underline{A}'\}|$ in Claim 2.13.

Claim 2.14. *For any $\underline{A} \in \mathfrak{A}$, we have*

$$|\{\underline{A}' \in \mathfrak{A} \colon \underline{A} \not\perp \underline{A}'\}| \leq \min\left(2^n, \Delta\left(\frac{en}{\Delta/\ell}\right)^{\Delta}\right) \cdot \min\left(2^{\ell \min(\Delta,n)}, k^{\ell} \min(\Delta, n)^{k\ell}\right).$$

Proof. Assume that there are j distinct entries in $\underline{A}$. Note that $j \leq \Delta \triangleq (\ell-1)k$. For some candidate $\underline{A}' \in \mathfrak{A}$, let j' denote the number of distinct entries in $\underline{A}'$. We will use i to denote the number of entries in $\underline{A}'$, which also appear in the set of distinct entries of $\underline{A}$. Note that since we want to count the number of tuples $\underline{A}'$ which are dependent on $\underline{A}$, we have that $1 \leq i \leq \min(j, j')$.

Using these observations, we see that the quantity $|\{\underline{A}' \in \mathfrak{A} \colon \underline{A} \not\perp \underline{A}'\}|$ will then be bounded as follows.

$$|\{\underline{A}' \in \mathfrak{A} \colon \underline{A} \not\perp \underline{A}'\}| \overset{(a)}{\leq} \sum_{j'=\lceil \Delta/\ell \rceil}^{\min(\Delta,n)} \sum_{i=1}^{\min(j,j')} \binom{j}{i} \binom{n-j}{j'-i} \times \binom{j'}{\leq k}^{\ell}$$

where the term $\binom{j'}{\leq k} \triangleq \sum_{k'=1}^{k} \binom{j'}{k'}$. Now, we explain the occurrence of the terms on the RHS as follows.

- The second summation accounts for the number of possible common entries between $\underline{A}$ and $\underline{A}'$, while the upper limit in the first summation is because we can have at the most $\min(\Delta, n)$ distinct entries in $\underline{A}'$. The lower limit in the first summation is because of the condition that $\sum_{j\in[\ell]} |A'_j| = \Delta$ which implies that there should be at least one A'_j with size $\lceil \Delta/\ell \rceil$.

- Within the summations, the term $\binom{j}{i}$ counts the number of possible subsets of i distinct entries of $\underline{A}$ that occur in $\underline{A}'$ also.

- The term $\binom{n-j}{j'-i}$ counts the number of possible ways to select the remaining $j' - i$ entries in $\underline{A}'$ from the $n - j$ entries in $[n]$ not appearing in $\underline{A}$.

- $\binom{j'}{\leq k}^{\ell}$ occurs to account for the number of ways to obtain the collection $\underline{A}'$ from the specific j' distinct entries that have been chosen from previous steps.

$$
\begin{aligned}
|\{\underline{A}' \in \mathfrak{A} \colon \underline{A} \not\perp \underline{A}'\}| &\overset{(a)}{\leq} \sum_{j'=\lceil \Delta/\ell \rceil}^{\min(\Delta,n)} \sum_{i=1}^{\min(j,j')} \binom{j}{i}\binom{n-j}{j'-i} \times \binom{j'}{\leq k}^{\ell} \\
&\leq \sum_{j'=\lceil \Delta/\ell \rceil}^{\min(\Delta,n)} \binom{n}{j'} \times \binom{j'}{\leq k}^{\ell} \\
&\leq \sum_{j'=\lceil \Delta/\ell \rceil}^{\min(\Delta,n)} \binom{n}{j'} \times \binom{\min(\Delta,n)}{\leq k}^{\ell} \\
&\overset{(a)}{\leq} \min\left(2^n, \Delta\left(\frac{en}{\Delta/\ell}\right)^{\Delta}\right) \times \min\left(2^{\ell \min(\Delta,n)}, k^{\ell}\min(\Delta,n)^{k\ell}\right), \qquad (2.16)
\end{aligned}
$$

where in (a) we have used the well known relationships $\sum_{r'=r_1}^{r_2}\binom{m}{r'} \leq 2^m$ and $\binom{m}{r_1} \leq \left(\frac{e\cdot m}{r_1}\right)^{r_1}$, for all non-negative integers r_1, r_2, m such that $r_1, r_2 \leq m$. Observing that $\lceil \Delta/\ell \rceil \leq j \leq \min(\Delta, n)$ and $i \geq 1$, we now refine the bound as shown in (2.16), thus completing the proof. □

Theorem 2.15. *There exists an (n,k)-MDS(ℓ) code over a field of size q, if*

$$
q \geq e\ell k^2 \cdot \min\left(2^n, \Delta\left(\frac{en}{\Delta/\ell}\right)^{\Delta}\right) \cdot \min\left(2^{\ell \min(\Delta,n)}, k^{\ell}\min(\Delta,n)^{k\ell}\right).
$$

Proof. From Claim 2.13 we have that, if

$$
q \geq e.\ell k^2.\left(\max_{\underline{A}\in\mathfrak{A}} |\{\underline{A}' \in \mathfrak{A} \colon \underline{A} \not\perp \underline{A}'\}|\right)
$$

then there exists a choice of evaluation points $\beta_i \in \mathbb{F} : i \in [n]$ ($\mathbb{F}$ being a field of size q) such that V is (n,k)-MDS(ℓ). Using Claim 2.13, Theorem 2.15 follows.

□

Remark 2.16. *A previously known upper bound from [8, 13, 11] was*

$$
q \geq \ell n^2 \binom{n}{\leq k}^{\ell} = n^{\mathcal{O}(\ell \min(k,n-k))}.
$$

However, note that $\binom{n}{\leq k} \leq 2^n$. Hence, we see that this existing bound is essentially

$$
\min(2^{\mathcal{O}(\ell n)}, n^{\mathcal{O}(\ell \min(k,n-k))}).
$$

Observe that our new bound in Theorem 2.15 arguably tightens this bound in some special regimes when $\Delta = (\ell-1)k < n$. *Indeed, when* $\Delta < n$, *our bound is at the most* $\mathcal{O}\left(\ell^2 k^3 \left(\frac{n}{k}\right)^{\ell k} 2^{\ell^2 k}\right) = \mathcal{O}\left(\ell^2 k^3 \left(\frac{2^\ell n}{k}\right)^{\ell k}\right)$, *which is tighter than* $\mathcal{O}(n^{\ell k})$ *for small values of* ℓ *(as compared to* k*).*

All the above results on Higher-order MDS codes also translate to results on LD-MDS codes due to the duality relationship stated in Lemma 1.8. We provide the proof of Lemma 1.8 here for the sake of completeness. The following lemma on the dimension of generic intersection is useful in proving the duality relationship between LD-MDS and higher order MDS codes.

Lemma 2.17. *[11] Given* $A_1, \ldots, A_\ell \subseteq [n]$ *of size at most* k, *for a generic matrix* $W_{k\times n}$, *we have that*

$$dim(W_{A_1} \cap W_{A_2} \cap \ldots \cap W_{A_\ell}) = \max_{P_1 \cup P_2 \cup \ldots P_s = [\ell]} \left(\sum_{i\in[s]} \left| \bigcap_{j\in P_i} A_j \right| - (s-1)k \right)$$

There is a duality relationship between LD-MDS codes and the Higher-order MDS codes, and is stated in Lemma 1.8. In order to prove the equivalence stated in the Lemma 1.8, we will break the proof into the following two propositions.

2.3.1 Proof of Lemma 1.8

Proposition 2.18. *If* $\mathcal{C}$ *is LD-MDS*$(\leq \ell)$ *then* $\mathcal{C}^\perp$ *is MDS*$(\ell+1)$.

Proof. We will prove the contrapositive: If $\mathcal{C}^\perp$ is not MDS$(\ell+1)$ then $\mathcal{C}$ is not LD-MDS$(\leq \ell)$. Let $\mathcal{C}$ be an (n,k) code whose parity check matrix is $H_{(n-k)\times n}$. As $\mathcal{C}^\perp$ is not MDS$(\ell+1)$ from Lemma 2.3, there exists an (n,k,ℓ)-generic collection $A_1, A_2, \ldots, A_\ell \subseteq [n]$, $|A_1| + |A_2| + \ldots + |A_\ell| = (\ell-1)k$ such that $H_{A_1} \cap H_{A_2} \cap \ldots \cap H_{A_\ell} \neq \mathbf{0}$. This implies there exists non-zero $u_1, \ldots, u_{\ell+1} \in \mathbb{F}_q^n$ such that

$$\text{supp}(u_i) \subset A_i \text{ and } Hu_1 = Hu_2 = \ldots = Hu_{\ell+1}.$$

Suppose there are s distinct vectors among $u_1, \ldots, u_{L+1}$. Let $P_1 \cup P_2 \ldots \cup P_s = \{1, 2, \ldots, \ell+1\}$ be the partition of $\ell+1$ into s parts such that all the $\{u_j : j \in P_i\}$ are equal for every $i \in [s]$. Let u_{P_i} be the common vector equal to $\{u_j : j \in P_i\}$. Note that $\text{supp}(u_{P_i}) \subset \cap_{j\in P_i} A_j = A_{P_i}$. Therefore, we have s distinct non-zero vectors $u_{P_1}, u_{P_2}, \ldots, u_{P_s}$ such that

$$\sum_{i=1}^{s} wt(u_{P_i}) \leq \sum_{i=1}^{s} |A_{P_i}| \leq (s-1)(n-k) \text{ and } Hu_{P_1} = Hu_{P_2} = \ldots = Hu_{P_s}.$$

If $s = 1$, we get $wt(u_{P_1}) \leq 0$ which is not possible since u_{P_1} is non-zero. Therefore $s \geq 2$ and this violates LD-MDS$(s-1)$ and therefore LD-MDS$(\leq \ell)$. □

Proposition 2.19. *If $\mathcal{C}^\perp$ is MDS$(\ell+1)$ then $\mathcal{C}$ is LD-MDS$(\leq \ell)$.*

Proof. Since MDS$(\ell+1)$ implies MDS$(\leq \ell+1)$, it is enough to show that if $\mathcal{C}^\perp$ is MDS$(\ell+1)$ then $\mathcal{C}$ is LD-MDS(ℓ) for $\ell \geq 2$. We will prove the contrapositive: If $\mathcal{C}$ is MDS and $\mathcal{C}$ is not LD-MDS(ℓ) then $\mathcal{C}^\perp$ is not MDS$(\ell+1)$.

Let $\mathcal{C}$ be an MDS (n,k)-code and $H_{(n-k)\times n}$ be its parity check matrix. Note that H is the generator matrix of $\mathcal{C}^\perp$. Since $\mathcal{C}$ is not LD-MDS(ℓ), there exists distinct $e_1, e_2, \ldots, e_{\ell+1} \in \mathbb{F}_q^n$ such that

$$\sum_{i=1}^{\ell+1} wt(e_i) \leq (\ell)(n-k) \text{ and } He_1 = He_2 = \ldots = He_{\ell+1}.$$

Let $J_i = \text{supp}(e_i)$. WLOG, we can assume that $|J_i| \leq n-k$. We can also infer that all the e_i are non-zero, because if say $e_1 = 0$, then $wt(e_i) \geq n-k+1$ for all $i \geq 2$, which violates the $\sum_{i=1}^{\ell+1} wt(e_i) \leq (\ell)(n-k)$ condition. For a generic $W_{(n-k)\times n}$ $dim(H_{J_1} \cap H_{J_2} \cap \ldots \cap H_{J_{\ell+1}}) > dim(W_{J_1} \cap W_{J_2} \cap \ldots \cap W_{J_{\ell+1}})$, hence $\mathcal{C}^\perp$ is not MDS$(\ell+1)$. □

Chapter 3

On Distributed Multi-User Secret Sharing with Multiple Secrets per User

In this chapter, we focus on designing a distributed secret sharing protocol (DSSP) that meets specific privacy requirements in a scenario where each user requests multiple secrets. We introduce a necessary condition on access structures to ensure weak secrecy. Additionally, we establish a relationship between access structures and t-disjunct matrices. We investigate different constructions of t-disjunct matrices in this context, comparing their performance based on the number of storage nodes required and the associated communication complexity. Furthermore, we derive bounds on the optimal communication complexity for a distributed secret-sharing protocol.

3.1 System Model

We consider a distributed secret sharing system that comprises n storage nodes, m ($m \geq n$) secrets, and $P = \binom{m}{t}$ users (Fig. 3.1), with the primary goal of enabling the dealer to convey a specific set of secrets to each user securely via storage nodes. In this system model:

- Each secret s_j has a *storage set* $A_j \subseteq [n]$, which represents the set of all storage nodes that store the shares corresponding to secret s_j. For each $i \in A_j$, a share corresponding to secret s_j is stored in i-th storage node. The set of all these storage sets is called the *access structure*, and it is denoted as

$$\mathcal{A} \triangleq \{A_j : j \in [m]\}. \tag{3.1}$$

- Storage nodes are passive, which means they do not communicate with each other. The users do not communicate with each other either.

- All the secrets s_j, $j \in [m]$, are uniformly distributed and mutually independent. Each user u requests a subset S_u of $[m]$ secrets, where $|S_u| = t$.

- The dealer has access to all storage nodes but has no access to the users.

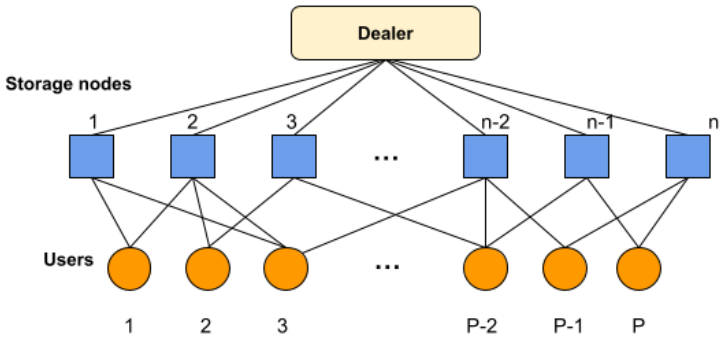

Figure 3.1: System Model

The objective is to design a distributed secret-sharing protocol that encodes secrets into shares and distributes them among storage nodes. The protocol should ensure that each user, denoted as u, can reconstruct their designated set of t secrets. Additionally, the protocol should satisfy the secrecy condition in a weak sense. In the protocol, the *weak secrecy* condition requires that a user does not get any information, in an information-theoretic sense, about the individual secrets of any other user. Let U_j denote the set of all the data the user j has access to, S_j is the set of secrets requested by user j. Then,

$$\forall j \in \left[\binom{m}{t}\right],\ \ell \in [m] \setminus S_j :\ H(s_\ell | U_j) = H(s_\ell). \tag{3.2}$$

A distributed secret sharing protocol (DSSP) that satisfies the weak secrecy condition is commonly referred to as a weakly secure DSSP.

3.1.1 Shamir's Secret Sharing Scheme

The (k, t) secret sharing scheme proposed by Shamir is described as follows and is used for encoding the secrets. Let's assume we have a secret value $s \in \mathbb{F}_q$, where $\mathbb{F}_q$ represents a finite field with q elements. The scheme generates k secret shares, denoted as $d_1, d_2, \ldots, d_k$, in such a way that the following conditions are satisfied:

(i) Reconstruction Property: With t or more secret shares, it is possible to reconstruct the original secret value s.

(ii) Secrecy Property: In an information-theoretic sense, possessing $t-1$ or fewer secret shares does not provide any information about the secret value s.

To construct the secret shares, a $(t-1)$-degree polynomial $P(x)$ is formed as follows:

$$P(x) = s + \sum_{i=1}^{t-1} p_i x^i$$

Here, the coefficients p_i are independent and identically distributed (i.i.d.) random variables, uniformly selected from $\mathbb{F}_q$. Additionally, k distinct non-zero elements $\gamma_1, \gamma_2, \ldots, \gamma_k$ are chosen from $\mathbb{F}_q$. The secret shares are then generated by evaluating $P(x)$ at these chosen points:

$$d_i = P(\gamma_i), \text{ for all } i \in [k].$$

By utilizing any t secret shares, the polynomial $P(x)$ can be reconstructed and is uniquely determined since its degree is at most $t-1$. This reconstruction process allows for the recovery of the original secret s. Now, we describe the (k, t) secret sharing scheme proposed by Shamir. Given a secret $s \in \mathbb{F}_q$, the output of the scheme consists of k secret shares $d_1, d_2, \ldots, d_k \in \mathbb{F}_q$ satisfying the following conditions:

(i) Given t or more secret shares, it is possible to reconstruct the secret s.

(ii) In the information-theoretic sense, the knowledge of $t-1$ or fewer shares does not disclose any information regarding s.

Consider a $(t-1)$-degree polynomial $P(x)$ given by $P(x) = s + \sum_{i=1}^{t-1} p_i x^i$, where p_is are i.i.d and are selected uniformly at random from $\mathbb{F}_q$. Let $\gamma_1, \gamma_2, \ldots, \gamma_k$ denote k distinct non-zero elements from F_q. The secret shares are then constructed by evaluating $P(x)$ at γ_is, i.e., $d_i = P(\gamma_i), \forall\ i \in [k]$. Given any t secret shares, $P(x)$ can be interpolated and is uniquely determined since the degree of $P(x)$ is at most $t-1$.

3.2 DSSP with Optimal Storage Overhead

In this section, we establish a necessary condition on storage sets in a distributed secret sharing protocol (DSSP) that ensures weak secrecy. This condition is closely related to the property of disjunct matrices, which are primarily used in the field of group testing (refer to [29, 30, 31] for more information). Furthermore, we present a scheme for constructing DSSPs with optimal storage overhead. This scheme leverages the access structure derived from disjunct matrices, allowing us to minimize the amount of storage required while maintaining weak secrecy.

3.2.1 Conditions on storage sets to ensure weak secrecy

Lemma 3.1. *For any weakly secure DSSP with an access structure $\mathcal{A}$ defined in* (3.1)*, we have*

$$A_{j_{t+1}} \nsubseteq (A_{j_1} \cup A_{j_2} \cup \cdots \cup A_{j_t}),$$

$\forall\ j_1, j_2, \ldots, j_{t+1} \in [m]$ *with* $j_1 \neq j_2 \neq \cdots \neq j_{t+1}$.

Proof. Let's assume the contrary, that is, $A_{j_{t+1}} \subseteq (A_{j_1} \cup A_{j_2} \cup \cdots \cup A_{j_t})$ for some distinct indices $j_1, j_2, \ldots, j_{t+1}$. This implies that the user who has access to the secrets $s_{j_1}, s_{j_2}, \ldots, s_{j_t}$ also has access to the secret $s_{j_{t+1}}$. However, this contradicts the weak secrecy condition stated in equation (3.2). The

weak secrecy condition ensures that knowledge of $t-1$ or fewer secrets does not disclose any information about the secret s. In our case, if $A_{j_{t+1}}$ is contained within the union of $A_{j_1}, A_{j_2}, \ldots, A_{j_t}$, it means that a user with access to $s_{j_1}, s_{j_2}, \ldots, s_{j_t}$ would also have access to $s_{j_{t+1}}$. This violates the weak secrecy condition. Hence, the assumption that $A_{j_{t+1}} \subseteq (A_{j_1} \cup A_{j_2} \cup \cdots \cup A_{j_t})$ leads to a contradiction. $\square$

The collection of subsets satisfying the Lemma 3.1 can be related to the columns of *t-disjunct* matrices and are defined as follows:

Definition 3.2. *A $n \times m$ binary matrix is t-disjunct if the union of supports of any t columns does not contain the support of any other column.*

Some well-known constructions for t-disjunct matrices are described in Section 3.3. We make the correspondence between the storage sets and t-disjunct matrices as follows: Consider a t-disjunct matrix where the columns correspond to the secrets, the rows correspond to the storage nodes, and the support of each column corresponds to the storage set of each secret.

3.2.2 DSSP with Optimal Storage Overhead

Consider a system with m secrets, and each user wants to access a subset of t secrets, $t < m$. There can be at most $\binom{m}{t}$ users in the system. Let $\mathcal{A} = \{A_i : i \in [m]\}$ be the access structure which consists of m subsets, each corresponding to the storage sets of m secrets. Suppose a user requests $\mathcal{P} \subset [m]$ secrets, then the user is given access to all the nodes in $\bigcup_{i \in \mathcal{P}} A_i$. To ensure weak secrecy, the access structure $\mathcal{A}$ must satisfy the condition specified in Lemma 3.1. So, we consider $\mathcal{A}$ to be a set of supports of each column of a t-disjunct matrix. We consider t-disjunct matrices to have the same column weight (i.e., the same number of non-zero positions in each column). Therefore each storage set is of the same size. Consider, $|A_i| = r$, $A_i = \{n_{i,1}, n_{i,2}, \ldots, n_{i,r}\}, \forall i \in [m]$.

To initialize the protocol, we pick n secrets such that the union of their access sets is $[n]$; WLOG let these be the first n secrets. To encode the secrets $s_j, \forall\, j \in [n]$ their $(r-1)$-degree polynomials $P_j(x)$s in (3.3) are constructed by the following system of linear equations:

$$P_j(x) = s_j + \sum_{l=1}^{r-1} p_{j,l} x^l \tag{3.3}$$

$$\begin{array}{c|c|c|c} P_1(\gamma_1) = y_{n_{1,1}} & P_2(\gamma_1) = y_{n_{2,1}} & \cdots & P_n(\gamma_1) = y_{n_{n,1}} \\ P_1(\gamma_2) = y_{n_{1,2}} & P_2(\gamma_2) = y_{n_{2,2}} & \cdots & P_n(\gamma_2) = y_{n_{n,2}} \\ \vdots & \vdots & \vdots & \vdots \\ P_1(\gamma_r) = y_{n_{1,r}} & P_2(\gamma_r) = y_{n_{2,r}} & \cdots & P_n(\gamma_r) = y_{n_{n,r}} \end{array}$$

Under certain conditions (Lemma 1 in [26]), the system has a unique solution for $p_{j,l}$s and $y_{n_{i,k}}$s, $i, j \in [n], l \in [r-1], k \in [r]$. Hence, there is a one-to-one mapping between $y_{[1:n]}$ and $s_{[1:n]}$.

The encoding of the remaining $m-n$ secrets is the same as the one proposed in [24] ($t=1$ case) to achieve optimal storage overhead. As $\bigcup_{i=1}^{n} A_i = [n]$, $\{y_{n_{1,1}}, \ldots, y_{n_{1,r}}, \ldots, y_{n_{n,1}} \ldots, y_{n_{n,r}}\} = y_{[1:n]}$. The data symbols $y_{[1:n]}$ correspond to the shares of the first n secrets, and $y_{[n+1:m]}$ correspond to the shares of remaining $m-n$ secrets.

Lemma 3.3. *The proposed protocol is a weakly secure DSSP satisfying all the conditions in Definition 1.9.*

Proof. In this protocol, each user j has access to all the $\sum_{i\in S_j} |A_i|$ evaluations of the polynomials associated with the secrets the user has requested. Hence, the correctness condition is satisfied by invoking Shamir's decoder. Now, we show that the proposed DSSP is indeed a weakly secure DSSP by showing that the condition specified in (3.2) holds.

First, we show that the data symbols $y_1, y_2, \ldots, y_m$ generated according to the proposed protocol are uniformly distributed and mutually independent. As the vector of all secrets is assumed to be full entropy, $(s_1, s_2, \ldots, s_n)$ is also full entropy. Also, there is a one-on-one relationship between $(s_1, s_2, \ldots, s_n)$ and $(y_1, y_2, \ldots, y_n)$ [24]. This implies $(y_1, y_2, \ldots, y_n)$ is also full entropy. Then,

$$H(y_{[n+1:m]}|y_{[1:n]}) = H(s_{[n+1:m]}|y_{[1:n]}) \tag{3.4}$$

$$\overset{(a)}{=} H(s_{[n+1:m]}) \overset{(b)}{=} (m-n)\log q, \tag{3.5}$$

where (3.4) holds since, given $y_{[1:n]}$ there is a one-on-one mapping between $y_{[n+1:m]}$ and $s_{[n+1:m]}$[24], (3.5(a)) holds since $y_{[1:n]}$ is independent of $s_{[n+1:m]}$ and (3.5(b)) holds since it is assumed that the vector of all secrets is full entropy. Using this together with the chain rule, we have

$$\begin{aligned} H(y_{[1:m]}) &= H(y_{[1:n]}) + H(y_{[n+1:m]}|y_{[1:n]}) \\ &= n\log q + (m-n)\log q = m\log q. \end{aligned} \tag{3.6}$$

Hence, from (3.6), we can say that the data symbols have full entropy and are mutually independent. As the storage sets are assumed to satisfy the t-disjunctness condition, there exists at least one γ_i, $i \in [r]$ such that $P_\ell(\gamma_i)$ is not accessed by user j. Let this data symbol $P_\ell(\gamma_i)$ be denoted by $y_\ell^{(-j)}$. Then $\forall\, j \in \left[\binom{m}{t}\right], \ell \in [m] \setminus S_j$

$$H(s_\ell|U_j) \geq H(s_\ell|\mathbf{y} \setminus y_\ell^{(-j)}) \tag{3.7}$$

$$\overset{(a)}{=} H(y_\ell^{(-j)}|\mathbf{y} \setminus y_\ell^{(-j)}) \overset{(b)}{=} H(y_\ell^{(-j)}) \overset{(c)}{=} \log q. \tag{3.8}$$

where (3.7) holds since conditioning does not increase the entropy, (3.8(a)) holds because given any $r-1$ evaluations of P_ℓ which is the evaluation polynomial corresponding to the ℓ-th user, out of r available ones, there is a one-to-one mapping between the remaining evaluation of P_ℓ and s_ℓ. (3.8(b)) because data symbols are independent and (3.8(c)) because data symbols are full entropy. Also, we have

$$H(s_\ell|U_j) \leq H(s_\ell) = \log q, \tag{3.9}$$

$\forall j \in \left[\binom{m}{t}\right], \ell \in [m] \setminus S_j$. From (3.8(c)) and (3.9), the weak secrecy condition (3.2) is satisfied. □

3.3 Comparison between Different Constructions of Disjunct Matrices

In this section, we will define several well-known constructions for t-disjunct matrices. We will then compare the number of storage nodes, denoted as n, and the communication complexity, denoted as C, when each of these constructions is employed as the access structure in the distributed secret sharing protocol (DSSP) discussed in Section 3.2.

3.3.1 Kautz-Singleton Construction [29]

A $[q, k, q-k+1]_q$ Reed-Solomon code is picked as the outer code $\mathcal{C}_{out}$ while the inner code $\mathcal{C}_{in} : \mathbb{F}_q \to \{0,1\}^q$ is defined as follows. For any $i \in \mathbb{F}_q$, $\mathcal{C}_{in}(i) = e_i$, where e_i is nothing but a one-hot vector. The concatenated code $\mathcal{C}^* = \mathcal{C}_{out} \circ \mathcal{C}_{in}$ is a $n \times m$ t-disjunct matrix, where $n = q^2, m = q^k$, and $t = \lfloor \frac{q-1}{k-1} \rfloor$. Here, each column in $\mathcal{C}^*$ has q ones. The set of storage sets obtained from the Kautz-Singleton construction is called the Kautz-Singleton access structure.

3.3.2 Sparse Disjunct Matrices [31]

In a $n \times m$ Sparse disjunct matrix, the number of ones in each column of a t-disjunct matrix is restricted to $\ell t + 1$, where $\ell \geq 1$. Such a matrix can be constructed by replacing the outer code in the Kautz-Singleton construction with $[\ell t + 1, k = \ell + 1]$-RS code over a field of size $q = \sqrt[\ell+1]{m}$. The concatenated code $\mathcal{C}^*$ is a $(\ell t + 1)q \times m$ t-disjunct matrix. The set of storage sets obtained from constructing the Sparse Disjunct Matrix is called Sparse Disjunct access structure.

3.3.3 Porat-Rothschild Construction [30]

A linear code that meets the Gilbert-Varshamov bound is picked as the outer code. Here, $\mathcal{C}_{out}$ is $[\mathscr{z}, k, \delta\mathscr{z}]_q$ linear code, where $\mathscr{z} \leq \frac{k}{1-H_q(\delta)}$ and $t+1 = \lceil \frac{1}{1-\delta} \rceil$. The inner code $\mathcal{C}_{in} : \mathbb{F}_q \to \{0,1\}^q$ is defined as follows. For any $i \in \mathbb{F}_q$, $\mathcal{C}_{in}(i) = e_i$, where e_i is nothing but a one-hot vector. The concatenated code $\mathcal{C}^* = \mathcal{C}_{out} \circ \mathcal{C}_{in}$ is a $n \times m$ t-disjunct matrix, where $n = \mathscr{z}q, m = q^k$ and $t =$

	Kautz-Singleton	Porat-Rothschild	Sparse disjunct ($\ell = 1$)
Code	$\mathcal{C}_{out} : [q_1, k, q_1 - k + 1]_{q_1}$- RS code $\mathcal{C}_{in} : I_{q_1}$	$\mathcal{C}_{out} : [\mathscr{z}, k, \delta\mathscr{z}]_{q_1}$-Linear code where $\mathscr{z} \leq \frac{k}{1-H_{q_1}(\delta)}$ $\mathcal{C}_{in} : I_{q_1}$	$\mathcal{C}_{out} : [t+1, k=2]_{q_2}$-RS code where $q_2 = \sqrt{m}$ $\mathcal{C}_{in} : I_{q_2}$
Disjunctness	$t = \lfloor \frac{q_1 - 1}{k-1} \rfloor$	$t + 1 = \lceil \frac{1}{1-\delta} \rceil$	$t + 1 \leq \sqrt{m}$
t-disjunct Matrix	$\mathcal{C}^* : q_1^2 \times m$ $\approx t^2 \log_t^2 m \times m$	$\mathcal{C}^* : \mathscr{z} q_1 \times m$ $\approx t^2 \log m \times m$	$\mathcal{C}^* : (t+1)q_2 \times m$ $= (t+1)\sqrt{m} \times m$
Column weight	$q_1 \approx t \log_t m$	$\mathscr{z}$	$t + 1$
Storage Overhead	1	1	1
Comm. Complexity	$\binom{m-1}{t-1} \times q_1 \times m$	$\binom{m-1}{t-1} \times \mathscr{z} \times m$	$\binom{m-1}{t-1} \times (t+1) \times m$

Table 3.1: Comparison of disjunct matrix constructions.

$\lceil \frac{1}{1-\delta} \rceil - 1$. Here, each column in $\mathcal{C}^*$ has $\mathscr{z}$ ones. The set of storage sets obtained from Porat-Rothschild Construction is called the Porat-Rothschild access structure.

For a given total number of secrets m and the number of secrets t each user requests ($t << m$), we compare the number of storage nodes n and the communication complexity C across the above-mentioned constructions for t-disjunct matrices (see also Table 3.1).

Kautz-Singleton (KS) Vs Porat-Rothschild (PR)

In the literature, two distinct regimes are commonly considered: (i) when the value of t is bounded by $O(poly(\log m))$, and (ii) when t is bounded by $O(m^\alpha)$, α in the range $(0, 1/2)$.

In regime (i), it has been established that the parameters n_{PR} and C_{PR} associated with the PR construction satisfy $n_{PR} < n_{KS}$ and $C_{PR} < C_{KS}$. This implies that PR outperforms KS in this regime, as it performs better in terms of the number of nodes (n) and the storage overhead (C). On the other hand, in regime (ii), the reverse inequalities hold: $n_{PR} > n_{KS}$ and $C_{PR} > C_{KS}$. Consequently, KS surpasses PR in this regime, offering superior performance in terms of the number of nodes and storage overhead. Thus, depending on the specific regime and the value of t, the comparison between PR and KS constructions reveals distinct outcomes, highlighting the relative advantages of each construction under different scenarios.

Kautz-Singleton (KS) Vs Sparse Disjunct ($\ell = 1$) (SD)

Since we have $t << m$, $n_{KS} < n_{SD}$ and $C_{KS} > C_{SD}$. There is a trade-off between these access structures, so if, for example, one seeks to minimize the number of storage nodes, KS is better than SD while paying for a higher communication complexity and vice versa.

Porat-Rothschild (PR) vs Sparse disjunct ($\ell = 1$) (SD)

Since we have $t << m$, $n_{PR} < n_{SD}$ and $C_{PR} < C_{SD}$. Thus, PR is better than SD.

Another interesting construction of the t-disjunct matrix uses the Steiner system, defined below.

Definition 3.4. *Let X be an n-element set. A Steiner system $\mathfrak{S}(n, b, p)$ is defined as $\mathfrak{S} \subset \binom{X}{b}$ such that for every $A \in \binom{X}{p}$ there is exactly one $B \in \mathfrak{S}$ with $A \subset B$, where $\binom{X}{i}$ here denotes the collection of all the i-sized subsets of X. The largest set $\mathfrak{S}$ which satisfies this property is called the maximum Steiner system.*

The conjecture relating Steiner systems and constant column weight disjunct matrices states that a Steiner system $\mathfrak{S}(n, b, p)$ can be used to construct a $\lfloor \frac{b-1}{p-1} \rfloor$-disjunct matrix with a constant column weight. This conjecture suggests a potential connection between the combinatorial structure of Steiner systems and the properties of disjunct matrices. If proven true, it would provide a valuable tool for constructing balanced collections and achieving balanced storage profiles in distributed storage systems, as discussed earlier. In our work, we proved this conjecture for the special cases where the matrix M is obtained using the Kautz-Singleton and Sparse Disjunct matrix constructions.

Conjecture 1. *[32] Let M be a $n \times m$ t-disjunct matrix with constant column weight b. Let $p = \frac{b+t-1}{t}$ (we assume that p is an integer). The maximum $\mathfrak{S}(n, b, p)$ Steiner system gives a matrix M' that is no worse than M. In other words, $|\mathfrak{S}(n, b, p)| \geq m$.*

Remark 3.5. *If Conjecture 1 is true, then for a given n, the number of storage nodes, and b, the size of storage sets, the access structure obtained from Steiner system $\mathfrak{S}(n, b, p)$ is the best in terms of accommodating more number of secrets.*

Consider a $q^2 \times q^k$ t-disjunct matrix obtained from Kautz-Singleton construction, $t = \lfloor \frac{q-1}{k-1} \rfloor$. We set $k = \frac{q+t-1}{t}$ to accommodate most secrets in this construction. Then the corresponding Steiner system with the same number of storage nodes is given by $\mathfrak{S}(q^2, q, k)$. The following lemma compares the total number of secrets accommodated by both constructions.

Lemma 3.6. *If the Steiner system $\mathfrak{S}(q^2, q, p = \frac{q+t-1}{t})$ exists, then it can accommodate more secrets compared to the access structure obtained from the Kautz-Singleton construction with the same number of nodes q^2 and constant column weight q. In other words,*

$$q^p \leq \left|\mathfrak{S}(q^2, q, p)\right| = \binom{q^2}{p} \Big/ \binom{q}{p}. \tag{3.10}$$

Proof. Expanding the binomial coefficients on RHS of (3.10) and observing that for all $\ell \in [1, q)$, $\frac{q^2-\ell}{q-\ell} > q$ gives (3.10). □

Similarly, a comparison between a sparse disjunct matrix and the Steiner system is given below.

Lemma 3.7. *If the Steiner system $\mathfrak{S}((t+1)q, t+1, 2)$ exists, then it can accommodate more secrets compared to the access structure obtained from the Sparse Disjunct matrix with the same number of nodes $(t+1)q$ and constant column weight $t+1$. In other words,*

$$q^2 \leq |\mathfrak{S}((t+1)q, t+1, 2)| = \binom{(t+1)q}{2} \Big/ \binom{t+1}{2}.$$

3.3.4 Balanced Storage Profile

In large-scale distributed storage systems, maintaining a balanced distribution of data across nodes is crucial for efficient performance and reliability. To achieve this, balanced collections are employed, where each element belongs to an equal number of subsets. The Kautz-Singleton construction and Sparse disjunct matrices are two techniques used to create balanced collections. These constructions provide t-disjunct matrices with constant row and column weights, ensuring an even distribution of data. By utilizing these matrices in a distributed storage system, a balanced storage profile can be achieved, mitigating issues such as slower access times and system failures and improving overall system performance and fault tolerance.

3.4 Bounds on Optimal Communication Complexity

In this section, we derive bounds on the minimum communication complexity of DSSPs where each user requests a subset of t secrets. Similar to [24], we will use the notion of *tight* DSSPs in deriving these bounds. The DSSP which attains the lower bound of the minimum communication complexity with equality is called *communication-optimal* DSSP.

Definition 3.8. *A DSSP is said to be tight DSSP (T-DSSP) if every user downloads exactly one $\mathbb{F}_q$-symbol from each node in the storage set corresponding to each secret in the user's designated set of secrets.*

Let b_k denote the number of t-subsets whose union is of size k in the access structure of a T-DSSP. Then its communication complexity lies between

$$\sum_{k=t}^{n} k b_k \leq C < t \sum_{k=t}^{n} k b_k, \tag{3.11}$$

where we obtain the lower and upper bounds when each user downloads exactly one share and t shares, respectively, from each node, the user has access to.

In [24], it is proved that for every DSSP with communication complexity C, there exists a T-DSSP with the same number of storage nodes and users with communication complexity $\tilde{C} \leq C$. Therefore,

we can minimize (3.11) to find communication-optimal DSSP, provided that access structure exists satisfying Lemma 3.1 with such b_k's.

We derive a necessary condition for the access structure satisfying Lemma 3.1 to exist, which is a generalization of the LYM inequality [33].

Lemma 3.9. *Consider an access structure $\mathcal{A}$ satisfying Lemma 3.1, then $\sum_{k=t}^{n} b_k / \binom{n}{k} \leq 1$.*

Proof. The permutations of $[n]$ can be counted in two different ways using the double-counting argument. One is by counting all permutations of $[n]$ identified with $\{1, \ldots, n\}$ directly, and there are $n!$ of them, and the other by generating a permutation of the $[n]$ by selecting sets $(S_{i_1}, \ldots, S_{i_t})$, each $S_{i_j} \in \mathcal{A}$ and choosing a map that sends $\{1, \ldots, |\cup_{j\in[t]} S_{i_j}|\}$ to $\cup_{j\in[t]} S_{i_j}$.

If $|\cup_{j\in[t]} S_{i_j}| = k$, the sets $(S_{i_1}, \ldots, S_{i_t})$ are associated in this way with $k!(n-k)!$ permutations, and in each of them the image of first k elements of $[n]$ is exactly $\cup_{j\in[t]} S_{i_j}$. Each permutation may only be associated with a single $\cup_{j\in[t]} S_{i_j}$. If a permutation is associated with $(S_{i_1}, \ldots, S_{i_t})$ and $(S_{i'_1}, \ldots, S_{i'_t})$, then one union would be a subset of the other. The number of permutations that this procedure can generate is less than or equal to $n!$, i.e.,

$$\sum_{\substack{S_{i_j}\in A \\ \forall j\in[t]}} |\cup_{j\in[t]} S_{i_j}|(1 - |\cup_{j\in[t]} S_{i_j}|) = \sum_{k=t}^{n} b_k k!(n-k)! \leq n!.$$

Dividing the above inequality by $n!$ gives the result. □

From (3.11) and Lemma 3.9, we consider the following discrete optimization problem to derive the bounds on optimal communication complexity.

$$min \quad \sum_{k=t}^{n} k b_k \tag{3.12}$$

$$\text{s.t.} \quad b_k \in \mathbb{N} \cup \{0\} \; \forall \; k \in [t:n] \tag{3.13}$$

$$\sum_{k=t}^{n} b_k = \binom{m}{t} \tag{3.14}$$

$$\sum_{k=t}^{n} \frac{b_k}{\binom{n}{k}} \leq 1. \tag{3.15}$$

The constraint (3.13) is set to ensure b_k's are non-negative, (3.14) is set because the sum of b_k's is equal to the number of users $\binom{m}{t}$ and (3.15) is a necessary condition for access structure satisfying Lemma 3.1 to exist.

Consider the continuous version of the previous optimization problem.

$$
\begin{aligned}
min \quad & \sum_{k=t}^{n} k\beta_k \\
\text{s.t.} \quad & \beta_k \geq 0 \; \forall \; k \in [t:n] \\
& \sum_{k=t}^{n} \beta_k = \binom{m}{t} \\
& \sum_{k=t}^{n} \frac{\beta_k}{\binom{n}{k}} \leq 1.
\end{aligned} \tag{3.16}
$$

The Lagrangian of the problem (3.16) is given by

$$
\mathcal{L} = \sum_{k=t}^{n} k\beta_k - \lambda_1 \left(\sum_{k=t}^{n} \beta_k - \binom{m}{t} \right) - \lambda_2 \left(1 - \sum_{k=t}^{n} \frac{\beta_k}{\binom{n}{k}} \right) - \sum_{k=t}^{n} \mu_k \beta_k,
$$

where $\lambda_1, \lambda_2, \mu_k$'s are Lagrange multipliers. The corresponding sufficient KKT conditions for the optimality of the variables are given below.

$$
\forall \; k: \quad k - \lambda_1 + \frac{\lambda_2}{\binom{n}{k}} - \mu_k = 0 \tag{3.17}
$$

$$
\forall \; k: \quad \mu_k \geq 0 \tag{3.18}
$$

$$
\forall \; k: \quad \beta_k^* \geq 0 \tag{3.19}
$$

$$
\forall \; k: \quad \mu_k \beta_k^* = 0 \tag{3.20}
$$

$$
\lambda_2 \left(1 - \sum_{k=t}^{n} \frac{\beta_k^*}{\binom{n}{k}} \right) = 0 \tag{3.21}
$$

$$
\lambda_2 \geq 0 \tag{3.22}
$$

$$
\sum_{k=t}^{n} \beta_k^* = \binom{m}{t} \tag{3.23}
$$

$$
\sum_{k=t}^{n} \frac{\beta_k^*}{\binom{n}{k}} \leq 1. \tag{3.24}
$$

Lemma 3.10. *[24] The solution of the problem* (3.16) *has at most two non-zero β_k's. Furthermore, if two of them are non-zero, then their indices are consecutive and $\lambda_2 > 0$.*

Using the above lemma, we derive the non-zero β_k^*'s. Since $\lambda_2 > 0$, (3.21) turns the inequality (3.24) into equality. Then

$$
\frac{\beta_i^*}{\binom{n}{i}} + \frac{\beta_{i+1}^*}{\binom{n}{i+1}} = 1.
$$

Using (3.23), we get

$$
\beta_i^* + \beta_{i+1}^* = \binom{m}{t}.
$$

By solving these two linear equations, we get

$$\beta_i^* = \frac{\binom{n}{i+1} - \binom{m}{t}}{\binom{n}{i+1} - \binom{n}{i}} \binom{n}{i}, \quad \beta_{i+1}^* = \frac{\binom{m}{t} - \binom{n}{i}}{\binom{n}{i+1} - \binom{n}{i}} \binom{n}{i+1}.$$

The inequality in (3.19) implies that the index i be such that $\binom{n}{i} \leq \binom{m}{t}$.

Thus, for a given number of secrets m and number of storage nodes n, any T-DSSP with an access structure $\mathcal{A}$ that has $\lfloor \beta_i^* \rfloor$ t-subsets whose union is of size i and $\lceil \beta_{i+1}^* \rceil$ t-subsets whose union is of size $i+1$ is a communication-optimal DSSP.

Chapter 4

Conclusion

In this chapter, we summarize the results presented in this thesis and briefly discuss potential directions for future research.

On the Structure of Higher Order MDS codes: In Chapter 2, we have derived some structural results for higher order MDS codes.

In specific parameter regimes, we establish that (n, k)-MDS(k) codes, which are also Reed-Solomon (RS) codes generated by a Vandermonde matrix over a set of evaluation points, remain closed under the expurgation operation. We extend this result to MDS(ℓ) codes for general ℓ, subject to certain special conditions. For (n, k)-MDS(k) RS codes, we demonstrate the preservation of the higher-order MDS property through a combination of puncturing and expurgation, referred to as the *pseudo-shortening* operation. Specifically, by removing the last row and any column from the Vandermonde generator matrix of an (n, k)-MDS(k) RS code, we establish the resulting code as an $(n-1, k-1)$-MDS($k-1$) code. Leveraging well-established probabilistic tools, we derive a new upper bound on the field size required for the existence of (n, k)-MDS(ℓ) codes, which is potentially superior to existing bounds in certain parameter regimes.

Future Work: An interesting open problem is to come up with low field size constructions for Higher-order MDS codes. In practical Distributed Storage Systems (DSS), the storage nodes are often spread out across different geographic locations and connected through a network, which may include intermediate devices like switches or routers. Traditionally, the main focus when designing codes for data storage in DSS was on ensuring reliability. The goal is to design codes that can effectively handle erasure patterns. While some erasure patterns are inherently uncorrectable, for those that can be potentially corrected, it is highly desirable to have a code that can correct all of them by appropriately assigning field elements while considering the code topology constraints. Such a code is known as a *maximally recoverable* code, capable of correcting every erasure pattern that is information-theoretically correctable. The concept of maximally recoverable codes was first introduced in [34] and has been further investigated in [35] and [12]. A recent work, [36], demonstrates that the instantiation of a maximally recoverable code can be seen as a special case of generic network coding utilizing a four-layer network topology. Similar to the approach described in [36] , we aim to establish a connection between the

MR instantiation of the parity check matrix of a tensor code and Generic network codes. By doing so, we can leverage the constructions available for Generic network codes to obtain constructions for MR tensor codes.

Distributed Multi-User Secret Sharing: We have considered a DMUSS, where the dealer conveys a specific subset of secrets to each user via the storage nodes. Our main result is the design of optimal weakly secure DSSP and compare its performance using well-known constructions for t-disjunct matrices. We also derived bounds on the optimal communication complexity of a DSSP and characterized the capacity region of the DMUSS considered.

Future Work: It remains open to solve for the exact value of optimal communication complexity. Another interesting direction for future work is to design a perfectly secure DSSP at the cost of increased storage overhead (due to the trade-off between SO and security level [24]).

Publications

1. Harshitanjani Athi, Rasagna Chigullapally, Prasad Krishnan, V Lalitha. "On the Structure of Higher Order MDS codes." in *Proc. IEEE International Symposium on Information Theory (ISIT), 2023.*

2. Rasagna Chigullapally, Harshithanjani Athi, Nikhil Karamchandani, V Lalitha. "On Distributed Multi-User Secret Sharing with Multiple Secrets per User." submitted to *IEEE GLOBECOM2023*.

Bibliography

[1] T. Kawachi and S.-i. Yamakami, "Quantum list decoding of classical block codes," *Information Theory, IEEE Transactions on*, vol. 56, no. 7, pp. 3222–3233, 2010.

[2] S. Tessaro and A. Vardy, "Quantum list decoding for codes over finite fields," *Information Theory, IEEE Transactions on*, vol. 59, no. 11, pp. 7328–7342, 2013.

[3] P. Elias, "List decoding for noisy channels," in *Technical Report 335*, (Research Laboratory of Electronics, MIT), 1957.

[4] V. Guruswami and M. Sudan, "Improved decoding of reed-solomon and algebraic-geometry codes," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1757–1767, 1999.

[5] S. Johnson, "A new upper bound for error-correcting codes," *IRE Transactions on Information Theory*, vol. 8, no. 3, pp. 203–207, 1962.

[6] A. Rudra and M. Wootters, "Every list-decodable code for high noise has abundant near-optimal rate puncturings," in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, (New York, NY, USA), p. 764–773, Association for Computing Machinery, 2014.

[7] C. Shangguan and I. Tamo, "Combinatorial list-decoding of reed-solomon codes beyond the johnson radius," in *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, (New York, NY, USA), p. 538–551, Association for Computing Machinery, 2020.

[8] R. M. Roth, "Higher-order MDS codes," *IEEE Transactions on Information Theory*, vol. 68, no. 12, pp. 7798–7816, 2022.

[9] E. Goldberg, C. Shangguan, and I. Tamo, "Singleton-type bounds for list-decoding and list-recovery, and related results," in *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 2565–2570, 2022.

[10] V. Guruswami and S. Narayanan, "Combinatorial limitations of average-radius list-decoding," *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 5827–5842, 2014.

[11] J. Brakensiek, S. Gopi, and V. Makam, "Lower bounds for maximally recoverable tensor codes and higher order MDS codes," *IEEE Transactions on Information Theory*, vol. 68, no. 11, pp. 7125–7140, 2022.

[12] P. Gopalan, G. Hu, S. Kopparty, S. Saraf, C. Wang, and S. Yekhanin, "Maximally recoverable codes for grid-like topologies," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2092–2108, SIAM, 2017.

[13] J. Brakensiek, S. Gopi, and V. Makam, "Generic reed-solomon codes achieve list-decoding capacity," 2022.

[14] X. Kong, J. Ma, and G. Ge, "New bounds on the field size for maximally recoverable codes instantiating grid-like topologies," *Journal of Algebraic Combinatorics*, vol. 54, pp. 529 – 557, 2019.

[15] J. Brakensiek, M. Dhar, and S. Gopi, "Improved field size bounds for higher order MDS codes," 2022.

[16] S. Lovett, "MDS matrices over small fields: A proof of the GM-MDS conjecture," in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 194–199, 2018.

[17] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, p. 612–613, 1979.

[18] G. R. Blakley, "Safeguarding cryptographic keys," *1979 International Workshop on Managing Requirements Knowledge (MARK)*, pp. 313–318, 1899.

[19] R. Cramer, I. Damgård, and U. Maurer, "General secure multi-party computation from any linear secret-sharing scheme," *Advances in Cryptology—EUROCRYPT 2000*, pp. 316–334, 2000.

[20] S. Takahashi and K. Iwamura, "Secret sharing scheme suitable for cloud computing," *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 530–537, 2013.

[21] Y. Liu and Q. Zhao, "E-voting scheme using secret sharing and k-anonymity," *World Wide Web*, vol. 22, pp. 1657–1667, 2019.

[22] R. K. Raman and L. R. Varshney, "Distributed storage meets secret sharing on the blockchain," *2018 information theory and applications workshop (ITA)*, pp. 1–6, 2018.

[23] A. Baccarini, M. Blanton, and C. Yuan, "Multi-party replicated secret sharing over a ring with applications to privacy-preserving machine learning," *Cryptology ePrint Archive*, 2020.

[24] M. Soleymani and H. Mahdavifar, "Distributed multi-user secret sharing," *IEEE Transactions on Information Theory*, vol. PP, 01 2018.

[25] E. Sperner, "Ein satz über untermengen einer endlichen menge," *Mathematische Zeitschrift*, vol. 27, no. 1, pp. 544–548, 1928.

[26] A. Khalesi, M. Mirmohseni, and M. A. Maddah-Ali, "The capacity region of distributed multi-user secret sharing," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 1057–1071, 2021.

[27] Y. Tian, "Formulas for calculating the dimensions of the sums and the intersections of a family of linear subspaces with applications," in *Beitr äge zur Algebra und Geometrie/Contributions to Algebra and Geometry*, vol. 60-3, pp. 471–485, 2019.

[28] P. Erdős and L. Lovász, "Problems and results on 3-chromatic hypergraphs and some related questions," *Infinite and finite sets*, vol. 10, pp. 609–627, 1975.

[29] W. Kautz and R. Singleton, "Nonrandom binary superimposed codes," *IEEE Trans. Inf. Theor.*, vol. 10, p. 363–377, sep 2006.

[30] E. Porat and A. Rothschild, "Explicit nonadaptive combinatorial group testing schemes," *IEEE Trans. Inf. Theor.*, vol. 57, p. 7982–7989, dec 2011.

[31] H. A. Inan, P. Kairouz, and A. Özgür, "Sparse combinatorial group testing," *IEEE Transactions on Information Theory*, vol. 66, no. 5, pp. 2729–2742, 2020.

[32] G. G. T. Balint, "Construction in non-adaptive group testing steiner systems and latin squares," *Ph.D. thesis, Illinois Institute of Technology*, 2014.

[33] D. Lubell, "A short proof of sperner's lemma," *Journal of Combinatorial Theory, Series A*, vol. 1, p. 299, 1966.

[34] M. Chen, C. Huang, and J. Li, "On the maximally recoverable property for multi-protection group codes," in *2007 IEEE International Symposium on Information Theory*, pp. 486–490, 2007.

[35] P. Gopalan, C. Huang, B. Jenkins, and S. Yekhanin, "Explicit maximally recoverable codes with locality," *IEEE Transactions on Information Theory*, vol. 60, no. 9, pp. 5245–5256, 2014.

[36] C. W. Sung, K. W. Shum, Q. Yu, and G. Xu, "Maximally recoverable codes: Connections to generic network coding and maximal matching," in *2017 IEEE Information Theory Workshop (ITW)*, pp. 36–40, 2017.