Local correction of codes of high rate

Wu, Liyasi

2017

Wu, L. (2017). Local correction of codes of high rate. Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/69693>

<https://doi.org/10.32657/10356/69693>

LOCAL CORRECTION OF CODES OF HIGH RATE

WU LIYASI

SCHOOL OF PHYSICAL & MATHEMATICAL SCIENCES

2016

LOCAL CORRECTION OF CODES OF HIGH RATE

WU LIYASI

School of Physical & Mathematical Sciences

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2016

DECLARATION

This dissertation is a presentation of my original research work, carried out by myself, in collaboration with my supervisors Chee Yeow Meng and Xing Chaoping. Most of the material in this dissertation has been presented in various conferences and published in the conference proceedings. The material in Chapter 3 has been presented in IEEE International Symposium on Information Theory in 2014 [1]. The construction in Chapter 4 has been published in the 53rd Annual Allerton Conference on Communication, Control, and Computing in 2015 [2]. The work in Chapter 5 has been presented in the IEEE Information Theory Workshop - Fall in 2015 [3].

ACKNOWLEDGEMENTS

This dissertation is the fruit of discussions, collaborations and interactions with many people that I come in contact with during my graduate studies in NTU. I feel very fortunate to have many wonderful teachers and mentors.

Firstly, I would like to thank my supervisors, Professor Chee Yeow Meng and Professor Xing Chaoping, for their patience and guidance. They have voluminous knowledge on coding theory, and many results of this dissertation could not be realized without their valuable suggestions. They also give a lot of help in my academic writing, presentation skills and critical thinking.

Furthermore, I would like to thank my thesis committee for their time and support. I am also very grateful to the anonymous reviewers for their helpful comments.

Finally, I am blessed to have the love, support and encouragement from my parents during this period.

ABSTRACT

This thesis is devoted to the study of locally correctable codes of high rate. We are mainly interested in new constructions of high-rate locally correctable codes, local correction of multiple bits, and the inner connection between the lifted Reed-Solomon codes and the multiplicity codes.

Locally correctable codes (LCC) are error-correcting codes with efficient decoding schemes, which can recover any bit of a codeword by visiting a small number of locations of the codeword. LCCs have found numerous applications in complexity theory, cryptography and the theory of fault tolerant computation. Kopparty et. al. [4] presented the first family of high-rate locally correctable codes, which is known as the multiplicity codes, by considering the evaluations of multivariate polynomials together with their Hasse derivatives. After that work, Guo et al. [5], discovered another family of high-rate locally correctable codes, the lifted Reed-Solomon codes, by considering lifting of multivariate polynomials.

In this thesis, firstly, we extend the techniques of lifted Reed-Solomon codes by lifting multivariate polynomials on curves, and generalize the decoding on curve algorithm from Reed-Muller codes to these lifted codes to provide a correcting algorithm with success probability arbitrarily approaching 1. This gives a family of high-rate locally correctable codes which can be recovered with probability close to 1.

Furthermore, we take a deeper look into the method of multiplicities and provide a new construction of locally correctable codes of rate approaching 1, which is also a theoretical connection between the codes in [4] and [5].

At last, we generalize the concept of traditional local correction to locally correcting multiple bits, and present a family of codes which allow multiple bits to be recovered with probability approaching 1 by visiting a small number of locations of the received word, even when a constant fraction of errors exist. Moreover, our codes are of high rate.

CONTENTS

1. Introduction	1
1.1 Locally Correctable Codes	1
1.2 Previous Constructions of High-Rate Locally Correctable Codes	6
1.2.1 Multiplicity Codes	6
1.2.2 Lifted Reed-Solomon Codes	7
1.2.3 Some Other Constructions	8
1.3 Main Contribution	9
1.3.1 Local Correction of High-Rate Codes on Curves	10
1.3.2 Revisiting the Multiplicity Codes	12
1.3.3 Local Correction of Multiple Positions	13
1.4 Organization of this Thesis	16
2. Preliminaries	17
2.1 Notations	17
2.1.1 Locally Correctable Codes	19
2.2 Properties of Multiplicities and Multiplicity Codes	20
2.2.1 Hasse Derivatives and Multiplicities	20
2.2.2 Multiplicity Codes	22

2.2.3	Local Correction of Multiplicity Codes	24
2.3	The Method of Lifting and Lifted Reed-Solomon Codes	27
3.	Correcting on Curves and Codes of High Rate	29
3.1	Lifting of Multivariate Polynomial Codes on Lines	29
3.2	Lifting of Multivariate Polynomial Codes on Curves	30
3.3	Preliminaries	31
3.4	The Code Construction and Decoding on curves	35
3.4.1	Proof of Theorem 3.2.1: Part I	36
3.4.2	Proof of Theorem 3.2.1: Part II	37
4.	Revisiting the Multiplicity Codes: A New Class of High-Rate Locally Cor-	
	rectable Codes	43
4.1	A Generalization of the Multiplicity Codes	43
4.2	Preliminaries	44
4.3	The Code Construction	46
4.3.1	Notations and Some Properties of Multiplicities	46
4.3.2	The Construction	49
4.4	Proof of Theorem 4.1.1	50
4.5	Remark: The Connection Between Multiplicity Codes and Our Codes . .	57
5.	Local Correction of Multiple Bits	61
5.1	Background and Motivations	61
5.2	A Generalized Definition of Locally Correctable Codes	62
5.3	A Theorem on Local Correction on Multiple Bits	63
5.4	The Construction	64

Contents	ix
5.4.1 Proof of Theorem 5.3.1: Part 1	67
5.4.2 The Decoding Algorithm	71
5.4.3 Proof of Theorem 5.3.1: Part 2	73
6. Conclusions	79
6.1 Concluding Remarks and Future Directions	79
Bibliography	81
Index	86

1. INTRODUCTION

The classical error-correcting codes encode messages of length k into codewords of length n by introducing redundancy, such that any codeword can be recovered even if it is corrupted in a number of positions. In order to recover the information, the decoder needs to read the entire corrupted codeword. This traditional method, however, is not efficient enough when people are only interested in a few positions of the message or codeword.

Locally correctable codes are error-correcting codes with efficient decoding schemes, which can recover any bit of a codeword by visiting a small number of locations of the codeword. They have found numerous applications in complexity theory, cryptography and the theory of fault tolerant computation. This dissertation contributes to the study of codes which are locally correctable and have rate close to 1, and we mainly investigate the constructions of such codes. In this chapter, we introduce the background on high-rate locally correctable codes and take an overview of our results.

1.1 Locally Correctable Codes

A q -ary code C on the coordinate set D is a set of functions from D to the finite field $\mathbb{F}_q$. Each function in C is called a codeword. The size of D is called the length of the code. We define the rate to be the ratio $\frac{\log_q |C|}{|D|}$. For functions f and g from D to $\mathbb{F}_q$, the distance

between them is the number of coordinates in D where f and g differ. The minimum distance of a code C is the minimum value of the distances between two distinct functions in C .

The motivation behind error-correcting codes is that there is noise in the communication channel and people want to add some redundancy to make information robust. To this end, we encode an original message m into a codeword c . Noise may cause some errors such that a new word y is received. If the number of errors is within the error tolerance of the code, i.e., $1/2$ of the minimum distance, then the decoder can give a unique codeword which is within $1/2$ of the minimum distance of y . In this a way, message m can still be recovered even if codeword c gets corrupted in some positions.

A drawback of the traditional error-correcting code is that, in some situations, the length of the received word is very large, however we may only want to recover one symbol of the codeword. It is inefficient if we read and process the entire received word in order to recover that symbol in this situation. Thus, a natural question is whether we can recover one symbol by accessing a subset of the coordinates in stead of the entire received word.

Actually, codes with efficient decoding schemes are possible, which allow one to recover any position in a codeword by visiting only a small number of code positions. These codes are called locally correctable codes. Locally correctable codes are a special kind of error-correcting codes, which allows any coordinate of a codeword to be recovered with high probability by looking up a small subset of coordinates of the corrupted codeword. A closely related kind of codes, which is more popular and very well studied, is the locally decodable codes. In some situations, we want a coordinate of a message (instead of a codeword) to be recovered. A code which allows any coordinate of a message to

be recovered with high probability by looking up a small number of coordinates of the codeword is called a locally decodable code.

The local correctability is actually a property stronger than the local decodability. It can be proved that from an (r, δ, ϵ) -locally correctable code of length N and dimension k , one can obtain a locally decodable code with the same parameters. For the proof of this result, we refer the reader to lemma 2.3 in [27].

This concept originates in the works on program checking, (See [6] and [7],) and was first formally defined in [8] and [9]. In the research of theoretical computer science, locally decodable codes and locally correctable codes played important roles in the proof-checking revolution [7, 8, 10–13] and complexity theory [14–17]. In a probabilistic proof system, for a decision problem L and an input x , one need to check whether x belongs to L or not, with random access to the ‘proof’. Similar to locally correctable codes, we want the query complexity to be small. They also have applications in cryptography such as Private Information Retrieval schemes [18–21].

Here we should mention that, although locally decodable codes and locally correctable codes have the property of efficiency and locality, for some important stochastic channels, codes with iterative decoding algorithms, such as LDPC codes, can be very close to Shannon capacity, thus are also very efficient both in theory and in practice. For more details, we refer the readers to the survey paper [22] and references therein.

When we study locally correctable codes, the main parameters which measure its utility are the rate, error tolerance, query complexity and probability of success of the decoding algorithm. The rate measures the amount of redundancy which is introduced into the message by the encoder. The error tolerance measures the maximum amount of

errors that the decoder can tolerate in order to give a unique answer of the symbol we want to correct. The query complexity measures the number of coordinates in a codeword which the decoder needs to access when we want to recover one single coordinate. We employ probabilistic algorithms when doing local correction, thus the probability of success of the decoder is also concerned.

Suppose we have an error-correcting code of length N and minimum distance d . This code can correct $\lfloor (d-1)/2 \rfloor$ errors. In other words, if we want to recover an arbitrary coordinate in a codeword, we need to read $N - \lfloor (d-1)/2 \rfloor + 1$ other coordinates. Thus, we can say that the traditional error-correcting codes are also locally correctable codes, whose query complexity is $O(N)$. Despite this trivial case, some traditional codes such as Reed-Muller codes also have local correction algorithms which perform better.

Ideally, we would like to have the rate and the error tolerance as large as possible, while the query complexity is as small as possible. However, these can not be achieved simultaneously. There is a trade off. Codes with different parameters have found applications in different areas.

In this thesis, we mainly focus on those locally correctable codes with the codeword length close to the message length, in other words, codes of rate close to 1. They find applications in data storage and transmission. In these applications, the rate of codes has always been paramount. It would increase the storage required or transmission time if codes of very small rate are used, which means an increase of cost and is unsatisfactory for most practical applications. Nevertheless, we should mention that locally correctable codes of high rate require a relatively large query complexity. The codes that have a small query complexity find applications in other areas, such as cryptography.

Recently Kopparty et. al. [4] and Guo et. al. [5] presented multiplicity codes and lifted Reed-Solomon codes, respectively, which are two families of high-rate locally correctable codes. Based on their works, we are mainly interested in new constructions of high-rate locally correctable codes, the connection between the lifted Reed-Solomon code and multiplicity code, and local correction of multiple bits.

We at first will present two new families of locally correctable codes of high rate. More precisely, we will introduce a family of high-rate locally correctable codes which is inspired by the lifted Reed-Solomon codes and employs a decoder with a higher probability of success. We extend the ‘lifting’ method in lifted Reed Solomon codes by ‘lifting’ multivariate polynomials on curves, and generalize the decoding on curves algorithm from Reed-Muller codes to these lifted codes to provide correcting algorithms with success probability arbitrarily approaching 1. This gives a family of high rate locally correctable codes that is highly sound.

Then, we will revisit the famous multiplicity codes in order to present another new construction of codes based on it. We generalize the method of multiplicities and provide a new construction of locally correctable codes of rate approaching 1, which is also a theoretical connection between the codes in [4] and [5]. We will prove that the traditional multiplicity code can be seen as a subcode of our construction.

Furthermore, we will investigate a more general question on local correction, i.e., constructions of codes which allow multiple coordinates to be corrected simultaneously. We give a family of codes which can correct multiple bits successfully with probability approaching 1 by visiting a small number of locations of the received word, even when a constant fraction of errors exist. Moreover, our codes are of high rate.

1.2 Previous Constructions of High-Rate Locally Correctable Codes

1.2.1 Multiplicity Codes

Locally correctable codes origin in research on program checking and have been implicitly and explicitly studied in coding theory for a long time. Since then, the quest for understanding locally correctable codes has generated many developments. For a q -ary code C on coordinate set $\mathbb{F}_q^n$, we say that C is a (r, δ, ϵ) -locally correctable code, if we can correct any codeword with probability at least $1 - \epsilon$ by visiting at most r positions, even if there are δ fraction of errors. A formal definition of locally correctable codes will be given in chapter 2.

The classical Reed-Muller codes based on evaluations of multivariate polynomials can be seen as the first construction of locally correctable codes. More precisely, a Reed-Muller code over finite field $\mathbb{F}_q$ of length N admits local correction with $O(N^\epsilon)$ queries and time, for some small constant ϵ , at the cost of reducing its rate to $\Omega(1/q)$.

However, for codes of rate greater than $1/2$, for a long time it was unknown how to get any nontrivial construction which admits local correction algorithms. The first family of high-rate locally correctable codes is introduced by Kopparty, Saraf and Yekhanin [4], which is known as the multiplicity code. A multiplicity code over $\mathbb{F}_q$ is a linear code of length N with rate close to 1, which is locally correctable from a constant fraction of errors with sublinear-time decoding algorithm and sublinear query complexity. More precisely, they showed that for any $\epsilon, \alpha > 0$, for infinitely many N , there is a code of length N and rate $1 - \alpha$, which is $(O(N^\epsilon), \Omega(\epsilon\alpha), 0.2)$ -locally correctable.

Each codeword of the multiplicity code is a list of evaluations of a multivariate polynomial together with its Hasse derivatives. Multiplicity codes generalize traditional mul-

tivariate polynomial codes. They inherit the local-correctability of these codes, while achieving better trade-offs in the rate and minimum distance.

We will take a closer look at this family of codes in Chapter 2 and Chapter 4.

1.2.2 Lifted Reed-Solomon Codes

Soon after the construction of multiplicity codes, Guo, Kopparty, and Sudan [5] introduced an alternative construction of high-rate locally correctable codes, which is known as lifted Reed-Solomon codes. They showed that for every $\epsilon, \alpha > 0$, and infinitely many N , there is a code of length N and rate $1 - \alpha$, which is $(O(N^\epsilon), \tau, 1/3)$ -locally correctable for some constant τ . Here the error tolerance τ is a constant smaller than the error tolerance of multiplicity codes.

These codes can be seen as another variant of Reed Muller codes. Unlike the q -ary Reed-Muller codes, which consist of all multivariate polynomials of degree at most d , the lifted Reed-Solomon codes employ all multivariate polynomials whose restriction on any line is of degree at most d , thus achieves a much higher rate than the Reed-Muller codes. Actually, they are supercodes of Reed-Muller codes with greater dimension but the same minimum distance.

The main idea behind this family of codes is the operation lifting. This operation takes a short-base code $C \subset \{\mathbb{F}_q^t \rightarrow \mathbb{F}_q\}$ and lifts it to a longer code $C \subset \{\mathbb{F}_q^n \rightarrow \mathbb{F}_q\}$, for integers $n > t$, such that codewords of C are those functions $f \in \{\mathbb{F}_q^n \rightarrow \mathbb{F}_q\}$ whose restriction to every t -dimension affine subspace lies in C .

Our Constructions in Chapter 3 and Chapter 5 extend the method of lifted Reed-Solomon codes, thus we will take a closer look at this method in Chapter 2 as a prelimi-

nary.

1.2.3 Some Other Constructions

Besides the families of multiplicity codes and lifted Reed-Solomon codes which inspire the construction of our codes in this thesis, there are some other high-rate locally correctable codes with very good performance. This section is an incomplete list of such codes.

Locally Correctable Codes Based on Distance-Amplification Method

In a surprising piece of work, Kopparty et. al. [23] in 2016 presented the first locally correctable codes with constant rate, constant relative distance, and sub-polynomial query complexity, which additionally have the property of approaching the Singleton bound. This is essentially as good as possible for general error-correcting codes. This shows that local correctability with $N^{o(1)}$ queries is not only possible with constant rate and constant relative distance, but it also does not require paying anything in terms of rate and relative distance.

This has the surprising consequence that asking for a locally correctable code of length N over a large alphabet with $\exp(O(\sqrt{\log N}))$ query complexity does not require any reducing in rate and distance. The rate of these codes can be arbitrarily close to 1. They have showed that, for every $\alpha > 0$, for infinitely many N , there is a code of length N , rate at least $1 - \alpha$, and relative distance at least $\alpha - o(1)$, which is $(\exp(\sqrt{\log N \cdot \log \log N}), \frac{\alpha - o(1)}{2}, 1/3)$ locally correctable.

These codes are based on the distance-amplification method presented by Alon and Luby in [24]. It turns out that this method increases the minimum distance significantly

when applied to suitably constructed locally correctable codes of sub-constant relative distance.

Lifted Algebraic Geometry Codes

Another family of high-rate locally correctable codes is introduced by Guo [25] by lifting algebraic geometry codes. They generalize the lifting of affine-invariant codes [5] and automorphic lifting which lifts algebraic geometry codes with respect to a group of automorphisms of the code. Like the multiplicity codes and the lifted Reed Solomon codes, these codes are locally correctable with sublinear query complexity while correcting a constant fraction of errors. (Note that lifting algebraic geometry codes has also been considered in [26] in a different way to construct locally correctable codes, which is known as degree lifting.)

1.3 Main Contribution

In this thesis, firstly, we extend the techniques of lifted Reed-Solomon codes by lifting multivariate polynomials on curves, and generalize the decoding on curves algorithm from Reed-Muller codes to these lifted codes to provide correcting algorithms with success probability arbitrarily approaching 1. This gives a family of high-rate locally correctable codes that is highly sound. This is the main contribution of Chapter 3.

Furthermore, we take a deeper look into the method of multiplicities and provide a new construction of locally correctable codes of rate approaching 1, which is also a theoretical connection between the codes in [4] and [5]. We will discuss this in Chapter 4.

Parameters

codes	query complexity	error tolerance	probability
Multiplicity Codes	$O(N^\epsilon)$	$\frac{\epsilon\alpha}{20}$	0.8
Lifted RS Codes	$O(N^\epsilon)$	$\Omega\left(\frac{1}{\exp(\exp(\log(1/\epsilon)/\epsilon)\log(1/\alpha))}\right)$	$\frac{2}{3}$
Lifted AG Codes	$O(N^\epsilon)$	$\Omega\left(\frac{1}{\exp(\exp(\log(1/\epsilon)/\epsilon)\log(1/\alpha))}\right)$	$\frac{2}{3}$
Codes in Chap. 3	$O(N^\epsilon)$	$\Omega\left(\frac{1}{\exp(\exp(\log(1/\epsilon)/\epsilon)\log(1/\epsilon)\log(1/\alpha))}\right)$	$1 - O_{\epsilon,\alpha}\left(\frac{1}{N^\epsilon}\right)$
LCCs based on Distance-Amplification Method	$\exp(O(\sqrt{\log N}))$	$\frac{\alpha - o(1)}{2}$	$\frac{2}{3}$

At last, we generalize the concept of traditional local correction to locally correcting multiple bits, and present a family of codes which allow multiple bits to be recovered with probability approaching 1 by visiting a small number of locations of the received word. Moreover, our codes are of high rate. This is the main contribution of Chapter 5.

Here we use a table to summarize the parameters for the known high-rate codes and our code in Chapter 3. Let $\alpha, \epsilon < 1$ be arbitrary positive real numbers. In this table, all the codes have length N and rate at least $1 - \alpha$.

1.3.1 Local Correction of High-Rate Codes on Curves

Recall that a q -ary Reed-Muller code consists of the evaluations on $\mathbb{F}_q^m$ of all m -variable polynomials of total degree at most d . Given a point $\mathbf{w} \in \mathbb{F}_q^m$, we want to recover the evaluation of a polynomial f at the point $\mathbf{w}$. To this end, we can shoot a random line ℓ through $\mathbf{w}$ and consider the restriction of the polynomial f to ℓ . By querying values of f at $d + 1$ points on the line, we can recover the evaluation of f at $\mathbf{w}$. This is the idea of decoding on lines for Reed-Muller codes. One drawback for this decoding on lines method is that the success probability $1 - \epsilon$ is small, namely $\epsilon = (d + 1)\delta$, where δ is the error tolerance (see [27]).

In order to improve this probability, a so-called decoding on curves strategy can be

used. In the decoding on curves algorithm, the decoder uses a random curve instead of a random line. As a result, it can decode with success probability arbitrarily approaching 1 (see [27, 28]), even when the fraction of errors is nearly $1/2$.

Unlike the q -ary Reed-Muller code, which consists of all multivariate polynomials of degree at most d , the lifted Reed-Solomon code employs all multivariate polynomials whose restriction on any line is of degree at most d , thus achieves a much higher rate than the traditional Reed-Muller code.

In Chapter 3, we explore the lifting of multivariate polynomial codes on curves and design a correcting algorithm based on decoding on curves. By choosing a random parametric curve of degree two over $\mathbb{F}_q$ and considering the restriction of functions to the curve, the corrector can recover any bit of the codeword with probability approaching 1 arbitrarily close. This gives a family of highly sound locally correctable codes that have high rate (rate tending to 1). More precisely, we will show that for arbitrarily small $\delta, \epsilon > 0$ and for infinitely many N , there is a family of $(N^\delta, \tau, O_{\epsilon, \delta}(1/N^\delta))$ -locally correctable codes of length N and rate $1 - \epsilon$, where $\tau = \Omega(1/\exp(\exp(\log(1/\delta)/\delta) \log(1/\delta) \log(1/\epsilon)))$.

All other parameters of our codes except for the rate are the same as those given in [5]. The code rate of our construction still approaches 1, but the convergence speed is slightly slower. On the other hand, if we want to keep the same convergence speed of the code rate, we have to tolerate a smaller number of errors. (This can be seen if we reformulate Theorem 3.2.1 in Chapter 3.)

Note that all previous locally decodable codes have decoding success probability bounded by a constant strictly less than 1. Comparing with the main result in [5], we obtain such a high success probability while keeping the codes rate and tolerance rate.

However, we should mention that the high probability of successful decoding is not necessarily an advantage, since one can simply use a majority vote to improve the success probability for any bit in a traditional LCC. As a cost, the algorithm will be polynomially slower than ours.

1.3.2 Revisiting the Multiplicity Codes

The first family of high-rate locally correctable codes is introduced by Kopparty, Saraf and Yekhanin [4], which is known as the multiplicity code. They showed that for any real numbers $\epsilon, \alpha > 0$, for infinitely many N , there is a code of length N and rate $1 - \alpha$, which is $(O(N^\epsilon), \Omega(\epsilon\alpha), 0.2)$ -locally correctable. Each codeword of the multiplicity code is a list of evaluations of a multivariate polynomial together with its Hasse derivatives.

We should notice that according to the construction in [4], the multiplicity code in n -variable is of high rate only if m , the order of Hasse derivatives, is significantly greater than n^2 . Thus a natural question is whether we can construct high-rate locally correctable codes by using the order m evaluations of n -variable polynomials when m is small. The lifted Reed-Solomon code can be seen as a construction of high-rate locally correctable codes by using the order 1 evaluations of multivariate polynomials.

In Chapter 4, we construct a large class of high-rate locally correctable codes by using the order m evaluations of multivariate polynomials for more general integer m . Our code consists of the order m evaluations of all n -variable polynomials over $\mathbb{F}_q$ whose restriction on any line is of degree at most d . We showed that when m is a prime power, this gives a family of high-rate locally correctable codes and we can apply the decoding algorithm of the multiplicity code to correct the desired coordinate of any codeword.

More precisely, we will show that for every $\epsilon, \alpha > 0$ and for infinitely many N , there exists a family of codes of length N and rate $1 - \alpha$, which is $(O(N^\epsilon), \tau, 0.2)$ -locally correctable, where $\tau = \Omega(1/\exp(\log(1/\alpha) \log(1/\epsilon) \exp(\log(1/\epsilon)/\epsilon)))$. Moreover, both multiplicity codes and lifted Reed-Solomon codes can be seen as special cases of our construction.

The contribution of this construction is not on parameters of the codes, but building a theoretical connection between two well known high-rate locally correctable codes, i.e., the multiplicity code and the lifted Reed-Solomon code, and answering the question whether we can construct high-rate locally correctable codes with fewer order of derivatives. Note that when $m = 1$, our construction becomes the lifted Reed-Solomon code. Moreover, we show that the traditional multiplicity code is actually a subcode of our code.

1.3.3 Local Correction of Multiple Positions

The event of multiple errors are not uncommon in reality, and we want to recover them efficiently. This motivates the concept of local correction for multiple bits. As a generalization of locally correctable codes, one may want to recover arbitrary t bits of a corrupted codeword, for some integer t . Ideally, one would like t to be large and the query complexity to be small. They however can not be achieved simultaneously. There is a trade-off. On one hand, we have the classical error-correcting code which allows the number of errors within one half of the minimum distance to be recovered with N queries, where N denotes the length of the code. On the other hand, we have the locally correctable code which allows 1 bit to be recovered with a small number of queries.

For a (r, δ, ϵ) -locally correctable code, if we correct t bits one by one, i.e., run the decoding algorithm t times, the number of queries will be rt , and the running time will

be increased as well. Thus a natural question is whether there exists a code which allows t bits to be recovered with less query complexity. Moreover, as with the classical error-correcting code, the rate is also a parameter of interest.

Let C be a q -ary code on coordinate set $\mathbb{F}_q^n$, for some integer n . We say the code C is $((r, \delta, \epsilon), t)$ -locally correctable if there exists a randomized decoding algorithm such that for all codewords in C , arbitrary t positions can be recovered with probability at least $1 - \epsilon$, by looking at other r positions in the codeword, even if there are δ fraction of errors. The formal definition of $((r, \delta, \epsilon), t)$ -locally correctable codes will be given in Chapter 5. Note that the traditional (r, δ, ϵ) -locally correctable code is actually $((r, \delta, \epsilon), 1)$ -locally correctable in the present notation.

In Chapter 5 we present a family of codes in which multiple bits of a codeword can be recovered by a small number of queries with probability approaching 1, even if there is a constant fraction of errors. At the same time, the rate of the code can be arbitrarily close to 1. Our code generalizes the construction of the lifted Reed-Solomon code, and the decoding on curves method in Gemmel and Sudan's work [28]. More precisely, we have the following result: For any $\epsilon, \alpha > 0$, there exists a constant τ , such that for infinitely many N , there is a code C of length N and rate $1 - \alpha$, which is $((O(N^\epsilon), \tau, O(\frac{1}{N^\epsilon})), t)$ -locally correctable, where t can be up to $O(\log N^\epsilon)^\epsilon$.

As a comparison, if we want to recover $O(\log N^\epsilon)^\epsilon$ positions in a codeword in the multiplicity code or lifted Reed-Solomon code by running their decoding algorithm repeatedly, where N denotes the length of the code, then the query complexity is $O(N^\epsilon \cdot (\log N^\epsilon)^\epsilon)$.

Note that if one want to correct t bits in a corrupted codeword, the query complexity will increase rapidly with t . Thus, the number of positions one want to recover can not

be too large. One obvious upper bound on t is $1/2$ of the minimum distance, since the decoder need to read the whole codeword in order to correct this number of errors. (For example, this bound is $\epsilon\alpha N/4$ for the multiplicity code.)

At the same time, the probability that our code can be recovered successfully is approaching 1. However, this is not necessarily an advantage, since one can simply use a majority vote to improve the success probability for any bit in a traditional LCC. As a cost, the algorithm will be polynomially slower than ours.

Remark

We should mention that recovering multiple erasures by using a small number of queries have been studied in the locally repairable codes (LRCs) world [29], which is known as the cooperative local repair and motivated by the fact that distributed storage systems can experience multiple simultaneous failures. In the literature on locally repairable codes, a code is said to have (r, l) -locality if it allows l failed code symbols to be recovered by contacting at most r other intact code symbols. For any code of length n and dimension k with (r, l) -locality, the minimum distance d is bounded by the following inequality given in [29]:

$$d \leq n - k + 1 - l \left(\left\lceil \frac{k}{r} \right\rceil - 1 \right).$$

While both locally correctable codes and locally repairable codes have the property of locality, they are actually designed for different purposes and under different assumptions. Locally correctable codes can help ensure reliable transmission of data over noisy channels, while locally repairable codes can repair node failures (erasures) in distributed storage systems. Locally repairable codes are a variant of the traditional erasure codes.

For more related works on locally repairable codes, we refer the readers to [30–32], and references therein.

1.4 Organization of this Thesis

The structure of this thesis is as follows. In Chapter 2, we introduce some basic terminologies and describe two previous constructions of locally correctable codes, which our codes are based on. We also review some important properties of Hasse derivatives and multiplicities as a preparation for our construction in Chapter 4.

In Chapter 3, we generalize the technic of ‘lifting’ and present our first construction of locally correctable codes, which is of high rate and highly sound.

In Chapter 4, we revisit the multiplicity codes and consider more general values for the order of the multiplicities. By doing this, we present our second construction of locally correctable codes, and we show that the traditional multiplicity codes can be seen as a special case of our construction. Moreover, we explore the theoretical connections between the multiplicity codes and lifted Reed-Solomon codes.

Chapter 5 is devoted to the study on local correction of multiple positions. We generalize the concept of locally correctable codes to multiple-error locally correctable codes, and provide a construction of codes which can correct multiple errors simultaneously.

2. PRELIMINARIES

2.1 Notations

We set up some notations which will be used in the remainder of the thesis.

- We use $|\cdot|$ to denote the cardinality of a set.
- The notation $[n]$ represents the set of integers $\{1, \dots, n\}$.
- $\mathbb{F}_q$ is a finite field of q elements, where q is a prime power.
- $\mathbb{F}_q^*$ is the multiplicative group of $\mathbb{F}_q$.
- We denote the set of non-negative integers and the set of positive integers by $\mathbb{Z}_{\geq 0}$ and $\mathbb{Z}_{>0}$, respectively.

Definition 2.1.1. A q -ary code C on coordinate set D is a set of functions from D to $\mathbb{F}_q$.

Each function in C is called a *codeword*. The size of D is called the *length* of the code and the number $\log_q |C|$ is called the *dimension*. We define the *rate* to be the ratio

$$\frac{\log_q |C|}{|D|}.$$

We say that C is a *linear code* if it is a linear space over $\mathbb{F}_q$. For a linear code, the dimension of the code can be seen as the dimension of the linear space over $\mathbb{F}_q$. In this work, most of the codes we discuss are linear codes, as they have good algebraic structures. Moreover, we often take the coordinate set to be $\mathbb{F}_q^n$ for some integer n .

For functions f and g from D to $\mathbb{F}_q$, the *distance* between them, which is denoted by $\text{dist}(f, g)$, is the number of coordinates in D where f and g differ. In other words,

$$\text{dist}(f, g) = |\{x \in D : f(x) \neq g(x)\}|.$$

The normalized quantity

$$\frac{|\{x \in D : f(x) \neq g(x)\}|}{|D|}$$

is called the *relative distance* between f and g , which is denoted by $\Delta(f, g)$. For a codeword f , we define its *Hamming weight* to be the number of coordinates in D where function f has nonzero evaluations.

Definition 2.1.2. The *minimum distance* of a code C , which is denoted by $\text{dist}(C)$, is the minimum value of the distance between two distinct functions in C . In other word,

$$\text{dist}(C) = \min_{f, g \in C, f \neq g} \text{dist}(f, g).$$

Naturally, the ratio

$$\frac{\min_{f, g \in C, f \neq g} \text{dist}(f, g)}{|D|}$$

is called the *relative distance* of C .

We denote a code C as an $(N, k, d)_q$ -code, if it is a q -ary code which has length

N , dimension k and minimum distance d . We should mention that codes of the same parameters are not unique.

2.1.1 Locally Correctable Codes

We say that C is a (r, δ, ϵ) -*locally correctable code*, if we can correct any codeword with probability at least $1 - \epsilon$ by visiting at most r positions, even if there are δ fraction of errors. A formal definition of the locally correctable code is as follows.

Definition 2.1.3. Let C be a q -ary code on coordinate set $\mathbb{F}_q^n$, for some integer n . We call the code C a (r, δ, ϵ) -locally correctable code if there exists a randomized decoding algorithm $\mathcal{A}$ such that

1. For all $\mathbf{w} \in \mathbb{F}_q^n$, all $f \in C$, and all functions $g : \mathbb{F}_q^n \rightarrow \mathbb{F}_q$ satisfying

$$\Delta(f, g) \leq \delta,$$

we have

$$\Pr[\mathcal{A}^g(\mathbf{w}) = f(\mathbf{w})] \geq 1 - \epsilon.$$

2. $\mathcal{A}$ makes at most r queries to g .

In this thesis, we mainly investigate the locally correctable codes of *high rate*.

Definition 2.1.4. We say a family of codes is of high rate if the rate tends to 1 as the length tends to infinity.

Definition 2.1.5. We call the parameter ϵ the *soundness error*. If the soundness error tends to 0 as the length tends to infinity, the code is said to be *highly sound*.

By convention, we use O , o , Ω and ω to describe the performance, such as the query complexity and error tolerance, of a family of codes with respect to the code length. For two functions $f(x)$ and $g(x)$, we say that $f(x) = O(g(x))$ if and only if $\lim_{x \rightarrow \infty} \sup \left| \frac{f(x)}{g(x)} \right|$ is finite. Similarly, $f(x) = o(g(x))$ if and only if $\lim_{x \rightarrow \infty} \left| \frac{f(x)}{g(x)} \right| = 0$. We say that $f(x) = \Omega(g(x))$ if and only if $g(x) = O(f(x))$. Moreover, $f(x) = \omega(g(x))$ if and only if $g(x) = o(f(x))$.

2.2 Properties of Multiplicities and Multiplicity Codes

In this section, we introduce the basic definitions, properties, and the decoding algorithm of multiplicity codes, which will be used in Chapter 4.

2.2.1 Hasse Derivatives and Multiplicities

At first, we need some preliminaries on multivariate polynomials, Hasse derivatives, and multiplicities.

Definition 2.2.1. Suppose $\mathbf{e} = (e_1, \dots, e_n)$ is an n -tuple, where $e_1, \dots, e_n$ are non-negative integers. We define its *weight*, which is denoted by $wt(\mathbf{e})$, to be the sum of its components, i.e.,

$$wt(\mathbf{e}) = \sum_{i=1}^n e_i.$$

We consider the polynomial ring

$$\mathbb{F}_q[\mathbf{x}] = \mathbb{F}_q[x_1, \dots, x_n],$$

where $\mathbb{F}_q$ is a finite field with q elements. Suppose

$$x_1^{e_1} \dots x_n^{e_n}$$

is a monomial in the ring, where $e_1, \dots, e_n$ are non-negative integers. We denote this monomial by $\mathbf{x}^{\mathbf{e}}$, where $\mathbf{e} = (e_1, \dots, e_n)$. The *degree* of a monomial, denoted Deg , is defined to be the sum of its exponents. Thus, we have

$$\text{Deg}(\mathbf{x}^{\mathbf{e}}) = \text{wt}(\mathbf{e}).$$

The degree of a polynomial is defined to be the highest degree of its monomials.

Now we introduce the *Hasse derivative* and *multiplicity* of polynomials, which play important roles in the construction of multiplicity codes. Most of the notations in this part follow [33] and [4].

Definition 2.2.2. Suppose

$$f(\mathbf{x}) \in \mathbb{F}_q[\mathbf{x}]$$

is a polynomial with n variables and $\mathbf{e} = (e_1, \dots, e_n)$ is a non-negative n -tuple. We define the $\mathbf{e}$ -th Hasse derivative of $f(\mathbf{x})$ to be the coefficient of $\mathbf{z}^{\mathbf{e}}$ in the polynomial

$$f(\mathbf{x} + \mathbf{z}),$$

where $\mathbf{z}$ is an indeterminate. We denote it by $f^{(\mathbf{e})}(\mathbf{x})$.

In other words, we have

$$f(\mathbf{x} + \mathbf{z}) = \sum_{\mathbf{e}} f^{(\mathbf{e})}(\mathbf{x}) \mathbf{z}^{\mathbf{e}}.$$

Definition 2.2.3. The multiplicity of a polynomial $f(\mathbf{x}) \in \mathbb{F}_q[\mathbf{x}]$ at some point $\mathbf{w} \in \mathbb{F}_q^n$ is the largest integer m such that the Hasse derivative $f^{(\mathbf{e})}(\mathbf{w})$ is always zero, for all non-negative n -tuples $\mathbf{e}$ with weight less than m . We denote it by

$$\text{mult}(f, \mathbf{w}).$$

In other words,

$$\text{mult}(f, \mathbf{w}) = \max \{ m \in \mathbb{Z} \mid f^{(\mathbf{e})}(\mathbf{w}) = 0 \text{ for all } \mathbf{e} \in \mathbb{Z}_{\geq 0}^n \text{ with } \text{wt}(\mathbf{e}) < m \}.$$

Note that if for a polynomial f and a point $\mathbf{w}$ the Hasse derivative $f^{(\mathbf{e})}(\mathbf{w})$ is zero for any $\mathbf{e} \in \mathbb{Z}_{\geq 0}^n$, then we say that the multiplicity of f at this point is infinity.

Similar to the normal derivative, it is easy to verify the linearity of the Hasse derivative.

Lemma 2.2.4. For any $f, g \in \mathbb{F}_q[\mathbf{x}]$, $a, b \in \mathbb{F}_q$, and non-negative n -tuple $\mathbf{e}$, we have that

$$(a \cdot f + b \cdot g)^{(\mathbf{e})}(\mathbf{x}) = a \cdot f^{(\mathbf{e})}(\mathbf{x}) + b \cdot g^{(\mathbf{e})}(\mathbf{x})$$

2.2.2 Multiplicity Codes

Now we introduce the concept of multiplicity codes formally.

Definition 2.2.5. Let Σ denote the set $\mathbb{F}_q^{\binom{n+m-1}{n}}$. The order m evaluation of a polynomial F at point $\mathbf{w}$ is defined as

$$F^{(<m)}(\mathbf{w}) = \left(F^{(\mathbf{i})}(\mathbf{w}) \right)_{wt(\mathbf{i}) < m}.$$

In other words, the order m evaluation of a polynomial F at some point $\mathbf{w}$ is a list of the Hasse derivatives of F at this point.

We recall the construction and properties of multiplicity codes, which will be generalized in Chapter 4.

Definition 2.2.6. Let $\mathbb{F}_q$ denote the finite field with q elements, and Σ be the set $\mathbb{F}_q^{\binom{n+m-1}{n}}$. The *multiplicity code* C of order m evaluations of degree d polynomials in n variable over $\mathbb{F}_q$ is a code of length q^n over alphabet Σ . For every polynomial F in $\mathbb{F}_q[x_1, \dots, x_n]$ with degree at most d , there is a codeword $Enc(F)$ in C , which is defined as

$$\left(F^{(<m)}(\mathbf{w}) \right)_{\mathbf{w} \in \mathbb{F}_q^n}.$$

In other words, the codeword correspond to a polynomial F is a vector coordinated by the points in $\mathbb{F}_q^n$, and each position is the order m evaluation of F at that point.

The following lemma gives the rate and relative distance of multiplicity codes, and shows that multiplicity codes are locally correctable.

Lemma 2.2.7. *Let C be the multiplicity code of order m evaluations of degree d polynomials in n variables over $\mathbb{F}_q$. Then the minimum distance of C is at least*

$$\delta = 1 - \frac{d}{mq},$$

and the rate is at least

$$(1 - \frac{n^2}{m})(1 - \delta)^n.$$

Moreover, if we assume that

$$q > \max \left\{ 10n, 12(m+1), \frac{d+15m}{m} \right\},$$

then the code C is $(O(m)^n q, \delta/10, 0.2)$ -locally correctable.

2.2.3 Local Correction of Multiplicity Codes

In this section we introduce the local correction algorithm of multiplicity codes, which will play an important role in the decoding of codes in Chapter 4. To this end, we start with some important properties of the restrictions of multivariate polynomial on lines, which imply the main idea of the decoding procedure.

We define the restriction of an n -variable polynomial F on a line to be the following univariate polynomial.

$$f(\lambda) = F(\mathbf{w} + \lambda \mathbf{v}) \in \mathbb{F}_{\mathbf{q}}[\lambda].$$

Thus, we have

$$\begin{aligned} f(\lambda) &= F^{(\mathbf{i})}(\mathbf{w})(\lambda \mathbf{v})^{\mathbf{i}} \\ &= \sum_{\mathbf{i}} F^{(\mathbf{i})}(\mathbf{w}) \mathbf{v}^{\mathbf{i}} \lambda^{\mathbf{wt}(\mathbf{i})}. \end{aligned}$$

This implies that

$$\sum_{\mathbf{i}, \mathbf{wt}(\mathbf{i})=e} F^{(\mathbf{i})}(\mathbf{w}) \mathbf{v}^{\mathbf{i}} = \text{coefficient of } \lambda^e \text{ in } f(\lambda).$$

Moreover, if we introduce another variable $\lambda_0 \in \mathbb{F}_q$, then we have that

$$\begin{aligned} F(\mathbf{w} + (\lambda_0 + \lambda)\mathbf{v}) &= f(\lambda_0 + \lambda) \\ &= \sum_j f^{(j)}(\lambda_0)\lambda^j. \end{aligned}$$

On the other hand, according to the definition of Hasse derivatives, we have

$$F(\mathbf{w} + (\lambda_0 + \lambda)\mathbf{v}) = \sum_{\mathbf{i}} F^{(\mathbf{i})}(\mathbf{w} + \lambda_0\mathbf{v})(\lambda\mathbf{v})^{\mathbf{i}},$$

which implies that

$$\sum_j f^{(j)}(\lambda_0)\lambda^j = \sum_{\mathbf{i}} F^{(\mathbf{i})}(\mathbf{w} + \lambda_0\mathbf{v})(\lambda\mathbf{v})^{\mathbf{i}}.$$

By comparing the coefficients of λ^j in the above equation, we can conclude that

$$f^{(j)}(\lambda_0) = \sum_{\mathbf{i}, wt(\mathbf{i})=j} F^{(\mathbf{i})}(\mathbf{w} + \lambda_0\mathbf{v})\mathbf{v}^{\mathbf{i}}. \quad (2.1)$$

The last equation shows that Hasse derivatives of the restriction of F on a line are linear combinations of Hasse derivatives of F .

Now we describe the decoding algorithm of multiplicity codes, which will be used in Chapter 4. Let C be the multiplicity code of order m evaluations of degree d polynomials in n variables over $\mathbb{F}_q$. Suppose

$$q > \max \left\{ 10n, 12(m+1), \frac{d+15m}{m} \right\}.$$

Let

$$G : \mathbb{F}_q^n \rightarrow \Sigma$$

be the received word, and $F \in \mathbb{F}_q[x_1, \dots, x_n]$ be a polynomial of degree at most d such that the distance between $Enc(F)$ and G are at most $\delta/10$. Suppose $\mathbf{w}$ is a point in $\mathbb{F}_q^n$ and we want to recover the order m evaluation $F^{(<m)}(\mathbf{w})$. We denote the $\mathbf{i}$ -th position of $G(\mathbf{w})$ by $G^{(\mathbf{i})}(\mathbf{w})$.

Step 1: Choose $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ from $\mathbb{F}_q^n$ independently and uniformly at random. Let M be a subset of $\mathbb{F}_q$ of size $12(m+1)$. We define a set of directions as follows:

$$V = \left\{ \mathbf{b}_0 + \sum_{i=1}^n a_i \mathbf{b}_i \mid a_i \in M \right\}.$$

Step 2: For each direction $\mathbf{v}$ in V and each element λ_0 in $\mathbb{F}_q$, we define $g_{\mathbf{v}}(\lambda_0)$ as follows:

$$g_{\mathbf{v}}(\lambda_0) = \left(\sum_{\mathbf{i} | wt(\mathbf{i})=j} G^{(\mathbf{i})}(\mathbf{w} + \lambda_0 \mathbf{v}) \mathbf{v}^{\mathbf{i}} \right).$$

The motivation behind this is the relationship represented in equation 2.1. Find a polynomial $h_{\mathbf{v}}(\lambda) \in \mathbb{F}_q[\lambda]$ such that the distance between $(g_{\mathbf{v}})_{\lambda \in \mathbb{F}_q}$ and the order m evaluation of h is less than $\delta/2$.

Step 3: For each integer e satisfying $0 \leq e < m$ and each direction $\mathbf{v}$, we define a linear equation with variables $u_{\mathbf{i}}$ as follows, where $\mathbf{i}$ are vectors with $wt(\mathbf{i}) = e$:

$$\sum_{\mathbf{i} | wt(\mathbf{i})=e} \mathbf{v}^{\mathbf{i}} u_{\mathbf{i}} = \text{coefficient of } \lambda^e \text{ in } h_{\mathbf{v}}.$$

For each e , we can construct a system of linear equations by using the equations for all $\mathbf{v} \in V$.

Note that the system of linear equations is noisy. Thus we want to find all $u_{\mathbf{i}}$ which satisfy at least $3/5$ of the equations.

Step 4: Output $\{u_{\mathbf{i}}\}_{wt(\mathbf{i}) < m}$.

2.3 The Method of Lifting and Lifted Reed-Solomon Codes

An alternative construction of high-rate LCCs is the lifted Reed-Solomon codes introduced by Guo, Kopparty, and Sudan [5]. Similar to the multiplicity codes, these codes can be seen as another variant of Reed-Muller codes. They have the same minimum distance as Reed-Muller codes, while the rate are significantly increased. More precisely, the rate of these codes can be arbitrarily close to 1.

The critical idea behind them is the technic of lifting, which is first introduced in [34]. The lifting operation transforms a short code

$$C \subset \{\mathbb{F}_q^n \rightarrow \mathbb{F}_q\}$$

to a longer code

$$C' \subset \{\mathbb{F}_q^N \rightarrow \mathbb{F}_q\},$$

where $N > n$, such that codewords of C' are those functions f from $\mathbb{F}_q^N$ to $\mathbb{F}_q$ whose restriction to every n -dimension affine subspace is a codeword of C .

Recall that in a Reed-Muller codes, all multivariate polynomials of total degree at

most d are collected to produce codewords. The idea employed in lifted Reed-Solomon codes is to collect a set of multivariate polynomials with total degree that may be greater than d , but that the restriction of every polynomial in this set to a line is still a univariate polynomial of degree at most d . As a result, Guo et al. are able to produce a new and natural family of high-rate locally correctable codes with a sublinear time correcting algorithm. The correcting strategy used is the method of decoding on lines, namely restricting polynomials to a line.

The main result of Guo et al. [5] may be summarized as follows.

Theorem 2.3.1. *For arbitrarily small $\delta, \alpha > 0$ and for infinitely many N , there is a family of $(N^\delta, \tau, 1/3)$ - locally correctable codes of length N with rate $1 - \alpha$, where $\tau = 2^{-c}/6$, $c = \lceil b2^{b^{1/\delta}} \log 1/\alpha \rceil$, and $b = 1 + \lceil \log[1/\delta] \rceil$, i.e.,*

$$\tau = \Omega(1/\exp(\exp(\log(1/\delta)/\delta) \log(1/\alpha))).$$

3. CORRECTING ON CURVES AND CODES OF HIGH RATE

3.1 Lifting of Multivariate Polynomial Codes on Lines

For a long time, people had not been able to construct locally correctable codes with rate greater than $1/2$. Kopparty, Saraf and Yekhanin [4] introduced the first family of high-rate locally correctable codes in 2011, which is known as multiplicity codes nowadays. Soon later in 2012, Guo, Kopparty and Sudan [5] presented a construction of high-rate locally correctable codes by applying the ‘lifting’ technic to multivariate polynomial codes.

Recall that in a Reed-Muller code, all multivariate polynomials of total degree at most d are collected to produce codewords and hence these polynomials become univariate polynomials of degree at most d when restricted to lines. The idea employed in [5] is to collect a set of multivariate polynomials with total degree that may be greater than d such that the restriction of every polynomial in this set to a line is still a univariate polynomial of degree at most d . As a result, the paper produces a new and natural family of high-rate locally correctable codes with sublinear time decoding. The decoding strategy used in [5] is still “decoding on lines”, namely restricting polynomials on a line.

To compare with our result in this chapter, we recall the main result of [5]. They

have proved that for arbitrarily small $\delta, \epsilon > 0$ and for infinitely many N , there is a family of $(N^\delta, \tau, 1/3)$ -locally correctable codes of length N with rate $1 - \epsilon$, where $\tau = \Omega(1/\exp(\exp(\log(1/\delta)/\delta) \log(1/\epsilon)))$.

3.2 Lifting of Multivariate Polynomial Codes on Curves

Let C be the q -ary Reed Muller code consisting of the evaluations on $\mathbb{F}_q^n$ of all n -variable polynomials of total degree at most d . Given a point $\mathbf{w}$, we want to recover the evaluation of a polynomial f at $\mathbf{w}$. To do so, the decoder shoots a random line ℓ through $\mathbf{w}$ and consider the restriction of the polynomial on ℓ . By querying values at $d+1$ points on the line, the decoder recovers the evaluation of f at $\mathbf{w}$.

One drawback for the “decoding on lines” algorithm is that the success probability $1 - \xi$ is too small, namely

$$\xi = (d+1)\delta$$

(see [27]). In order to improve this probability, the so-called “decoding on curves” algorithm was proposed [27]. In the “decoding on curves” algorithm, the decoder uses a random curve instead of the line ℓ . As a result, it can tolerate nearly $1/2$ fraction of errors with success probability arbitrarily approaching 1 (see [27] and [28]).

In the construction of our codes, we consider the polynomials whose degree is at most d when we restrict it on any curve of degree 2. We use the ‘decoding on curves’ algorithm to do the local correction. More precisely, we have the following result

Theorem 3.2.1. *For arbitrarily small $\delta, \epsilon > 0$ and for infinitely many N , there is a*

family of $(N^\delta, \tau, O_{\epsilon, \delta}(1/N^\delta))$ -locally correctable codes of length N and rate $1 - \epsilon$, where

$$\tau = 2^{-c}/6,$$

$$c = \lceil b2^{b\lceil 1/\delta \rceil} \log 1/\epsilon \rceil,$$

and

$$b = 2 + \lceil \log \lceil 1/\delta \rceil \rceil,$$

i.e.,

$$\tau = \Omega(1/\exp(\exp(\log(1/\delta)/\delta) \log(1/\delta) \log(1/\epsilon))).$$

3.3 Preliminaries

In this section, we introduce some basic terminologies required in this chapter. Most of the notations here follow [5] and [27]. Let us first recall the construction of the codes given in [5].

For a vector

$$\mathbf{d} = (d_1, \dots, d_m) \in \mathbb{Z}_{\geq 0}^m,$$

we use $\mathbf{x}^{\mathbf{d}}$ to denote the monomial

$$x_1^{d_1} \dots x_m^{d_m}.$$

The code C consists of all polynomials which have degree at most d when restricted to a

line. In other words, C is defined as the set

$$\{f = \sum \gamma x_1^{d_1} \cdots x_m^{d_m} \in \mathbb{F}_q[X_1, \dots, X_m] \mid d_i \leq q-1;$$

$$\text{for any } \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{F}_q^m, \deg(f|_{\boldsymbol{\alpha}+\lambda\boldsymbol{\beta}=\mathbf{0}}) \leq d\}.$$

Definition 3.3.1. Let

$$f = \sum_{\mathbf{d}} c_{\mathbf{d}} \mathbf{x}^{\mathbf{d}}$$

be a function from $\mathbb{F}_q^m$ to $\mathbb{F}_q$. The *support* of f is defined as

$$\text{supp}(f) = \{\mathbf{d} \mid c_{\mathbf{d}} \neq 0\}.$$

Given a vector

$$\mathbf{d} = (d_1, \dots, d_m) \in \mathbb{Z}_{\geq 0}^m,$$

we say it is in the *degree set* if there exists a function

$$f = \sum_{\mathbf{d}} c_{\mathbf{d}} \mathbf{x}^{\mathbf{d}}$$

in code C such that the monomial corresponding to this vector has nonzero coefficient.

More formally, we have the following definition.

Definition 3.3.2. Let C be a set of functions from $\mathbb{F}_q^m$ to $\mathbb{F}_q$. The degree set of C is defined as

$$\text{Deg}(C) = \bigcup_{f \in C} \text{supp}(f).$$

Let p be a prime number. For a non-negative integer a , we denote its base p expansion

by

$$a^{(0)}, a^{(1)}, a^{(2)}, \dots,$$

i.e.,

$$a = \sum_i a^{(i)} p^i$$

and

$$0 \leq a^{(i)} < p.$$

If a and b are non-negative integers and satisfy that

$$b^{(i)} \leq a^{(i)}$$

for every i , then we say that b is in the p -shadow of a , denoted

$$b \leq_p a.$$

For vectors

$$\mathbf{a} = (a_1, \dots, a_n)$$

and

$$\mathbf{b} = (b_1, \dots, b_n),$$

we say $\mathbf{b}$ is in the p -shadow of $\mathbf{a}$ if

$$b_i \leq_p a_i$$

for every $i \in \{1, \dots, n\}$. (see [5].)

Now, we recall the famous Lucas' Theorem, which expresses the congruence class of

binomial coefficient $\binom{a}{b}$ modulo p in terms of the base p expansions of integer a and b . We will use this theorem to count the number of codewords in the following sections.

Theorem 3.3.3. (Lucas' Theorem.) *For non-negative integers a and b , we have*

$$\binom{a}{b} \equiv \prod_i \binom{a^{(i)}}{b^{(i)}} \pmod{p}.$$

Note that

$$\binom{a}{b} \not\equiv 0 \pmod{p}$$

if and only if

$$b \leq_p a.$$

The following operation introduced in [5] is also needed when dealing with the degree set. Let q be a prime power. Let

$$(\bmod^* q)$$

denote an operation which maps non-negative integers to the set $\{0, \dots, q-1\}$. The operation $(\bmod^* q)$ is defined as follows: For an integer $a \geq 0$,

$$a \pmod^* q = \begin{cases} a \pmod{q-1} & \text{if } a \pmod{q-1} \neq 0; \\ q-1 & \text{if } a \neq 0 \text{ and } (q-1) \mid a; \\ 0 & \text{if } a = 0. \end{cases}$$

3.4 The Code Construction and Decoding on curves

In this section, we consider the code C consists of all polynomials which have degree at most d when restricted to a curve. In other words, C is defined as the set

$$\{f = \sum \gamma x_1^{e_1} \cdots x_m^{e_m} \in \mathbb{F}_q[X_1, \dots, X_m] \mid e_i \leq q-1;$$

$$\text{for any } \alpha, \beta, \gamma \in \mathbb{F}_q^m, \deg(f|_{\alpha + \lambda\beta + \lambda^2\gamma}) \leq d\}.$$

Given $\epsilon, \delta < 1$, we set

$$m = \lceil 1/\delta \rceil,$$

$$b = 2 + \lceil \log m \rceil$$

and

$$c = b2^{bm} \lceil \log 1/\epsilon \rceil.$$

Suppose $q = 2^s$ for some positive integer s . Let $N = q^m$. Let $\gamma = 2^{-c}$, $d = (1 - 2^{-c})q$ and $\tau = \gamma/6$. Suppose C is the code we defined above and g is a received function such that

$$\Delta(f, g) \leq \tau$$

for some f in C . Given a point $\alpha \in \mathbb{F}_q^m$, we want to recover the evaluation of f at this point by looking at the value of g at $O(N^\delta)$ positions. To this end, the corrector picks points $\beta, \gamma \in \mathbb{F}_q^m$ uniformly at random and consider the curve

$$\chi = \{\alpha + \lambda\beta + \lambda^2\gamma \mid \lambda \in \mathbb{F}_q\}.$$

By using evaluations of function g on this curve, the restriction of f can be recovered. Thus the value at the desired point can be computed easily.

Remark: From this part we can see that by using curves of degree two instead of lines in the lifted Reed-Solomon codes, the probability of success is improved. If we further consider the lifting on curves of degree 3 or higher degrees, the probability will again be improved but not significantly compared with curves of degree 2. Thus, we only discuss the results obtained from curves of degree 2.

3.4.1 Proof of Theorem 3.2.1: Part I

The proof of the main theorem in Chapter 3 is divided into the sections in the remainder of this chapter. In this part, we prove the statement on the decoding probability in Theorem 3.2.1. For the code C , we want to prove the following result: C is an $(N^\delta, \tau, O_{\epsilon, \delta}(1/q))$ -locally correctable code.

Recall that

$$d = (1 - 2^{-c})q.$$

If the fraction of errors on a curve is less than 6τ , we can recover the polynomial successfully.

Lemma 3.4.1 (Chebyshev's inequality). *Let X be a random variable. Its mean and variance are μ and η^2 respectively. Let k be a positive constant. Then the following inequality holds:*

$$\Pr[|X - \mu| \geq k] \leq \eta^2/k^2.$$

For more details about this inequality, see [35].

Now, we are ready to prove theorem 3.2.1.

proof of theorem 3.2.1. We consider a random variable X , which is the number of corruptions on the set

$$\chi^* = \{\mathbf{w} + \lambda \mathbf{v}_1 + \lambda^2 \mathbf{v}_2 \mid \lambda \in \mathbb{F}_q^*\}.$$

We can verify that

$$E[X] = (q-1)\tau,$$

and

$$D[X] \leq (q-1)(\tau - \tau^2).$$

An application of Chebyshev's inequality gives the desired result.

$$\begin{aligned} Pr[X \geq 6\tau(q-1)] &= Pr[X - E[X] \geq 6\tau(q-1) - E[X]] \\ &\leq D[X]/[5\tau(q-1)]^2 \\ &= O_\tau(1/q) \\ &= O_{\epsilon,\delta}(1/q). \end{aligned}$$

□

3.4.2 Proof of Theorem 3.2.1: Part II

In this part, we prove the statement on the code rate in Theorem 3.2.1.

We determine the degree set of C at first. Note that

$$\begin{aligned}
 & (\alpha_i + \beta_i \lambda + \gamma_i \lambda^2)^{e_i} \\
 = & \sum_{u_i \leq_2 e_i} \binom{e_i}{u_i} \alpha^{e_i - u_i} (\beta_i \lambda + \gamma_i \lambda^2)^{u_i} \\
 = & \sum_{u_i \leq_2 e_i} \binom{e_i}{u_i} \alpha^{e_i - u_i} \left[\sum_{w_i \leq_2 u_i} \binom{u_i}{w_i} \beta_i^{u_i - w_i} \lambda^{u_i - w_i} \gamma_i^{w_i} \lambda^{2w_i} \right].
 \end{aligned}$$

Thus, by Lucas' Theorem, we can verify that the degree set D is

$$\{\mathbf{e} \in \{0, \dots, q-1\}^m \mid \text{for any } \mathbf{u}, \mathbf{w} \in \{0, \dots, q-1\}^m \text{ satisfying } \mathbf{u} \leq_2 \mathbf{e} \text{ and } \mathbf{w} \leq_2 \mathbf{u},$$

$$\text{we have } \sum_{i=1}^m (u_i + w_i) \pmod{q} \leq d\}.$$

Lemma 3.4.2. *Let*

$$b = 2 + \lceil \log m \rceil.$$

Given $\mathbf{e} = (e_1, \dots, e_m) \in \{0, \dots, q-1\}^m$, if there exists an integer

$$j \in \{s - c, \dots, s - b\}$$

such that for every $i \in \{1, \dots, m\}$ we have

$$e_i^{(j)} = e_i^{(j+1)} = \dots = e_i^{(j+b-1)} = 0,$$

then $\mathbf{e}$ is in the degree set of the code.

Proof. Suppose

$$\mathbf{u}, \mathbf{w} \in \{0, \dots, q-1\}^m$$

satisfying

$$\mathbf{u} \leq_2 \mathbf{e}$$

and

$$\mathbf{w} \leq_2 \mathbf{u}.$$

Let $z_i = u_i$ for $i = 1, \dots, m$ and $z_i = w_{i-m}$ for $i = m+1, \dots, 2m$. Due to the property of $\mathbf{e}$, we have

$$z_i^{(j)} = \dots = z_i^{(j+b-1)} = 0$$

for every $i \in \{0, \dots, 2m\}$. Let

$$z = \sum_{i=1}^{2m} z_i \pmod{*q}.$$

We should show that $z \leq d$. Note that

$$d^{(0)} = \dots = d^{(s-c-1)} = 0$$

and

$$d^{(s-c)} = \dots = d^{(s-1)} = 1.$$

Thus, $z < d$ if and only if at least one of $z^{(s-c)}, \dots, z^{(s-1)}$ is 0.

Let

$$z'_i = 2^{s-j-b} z_i \pmod{*q}$$

for $i = 1, \dots, 2m$, and

$$z' = \sum_{i=1}^{2m} z'_i \pmod{*q}.$$

It is obvious that

$$z^{(k)} = z'^{(s-j-b+k \pmod{s})}.$$

Therefore, we have

$$z'_i{}^{(s-b)} = \dots = z'_i{}^{(s-1)} = 0,$$

which implies that

$$z'_i < 2^{s-b}.$$

Then

$$z' = \sum_{i=1}^{2m} z'_i < 2m \cdot 2^{s-b} \leq 2^{s-1}.$$

As a result, we have

$$z'^{(s-1)} = 0$$

thus

$$z^{j+b-1} = 0.$$

The fact that $j + b - 1$ lies in $\{s - c, \dots, s - 1\}$ completes the proof.

□

proof of theorem 3.2.1. We pick $\mathbf{u}$ randomly. For each

$$u_i = \sum_{j=0}^{s-1} u_i^{(j)} 2^j,$$

we consider the last c coordinates of the expansion,

$$u_i^{(s-c)}, \dots, u_i^{(s-1)},$$

as $\frac{c}{b}$ blocks of size b . The probability that there exists an all-zero block is

$$1 - (1 - 2^{-mb}),$$

which is at least

$$1 - e^{-\frac{c}{b2^{mb}}}.$$

Thus the dimension of the code is at least $(1 - \epsilon)N$.

□

4. REVISITING THE MULTIPLICITY CODES: A NEW CLASS OF HIGH-RATE LOCALLY CORRECTABLE CODES

4.1 A Generalization of the Multiplicity Codes

The multiplicity codes are an important family of high-rate algebraic error-correcting codes which admit local correction and are based on polynomial evaluations. A multiplicity code of length N is locally correctable from a constant fraction of errors with $O(N^\epsilon)$ query complexity, for some small constant ϵ . By constructing multiplicity codes, it can be proved that for any $\epsilon, \alpha > 0$, for infinitely many N , there is a code of length N and rate $1 - \alpha$, which is $(O(N^\epsilon), \Omega(\epsilon\alpha), 0.2)$ -locally correctable.

We should notice that according to the construction in [4], the multiplicity code in n -variable is of high rate only if m , the order of Hasse derivatives, is significantly greater than n^2 . Thus a natural question is whether we can construct high-rate LCCs by using the order m evaluations of n -variable polynomials when m is small. The lifted Reed-Solomon codes can be seen as a construction of high-rate LCCs by using the order 1 evaluations of multivariate polynomials.

In this chapter, we construct a large class of high-rate LCCs by using the order m evaluations of multivariate polynomials for more general m . Our code consists of the order m evaluations of all n -variable polynomials over $\mathbb{F}_q$ whose restriction on any line is of degree at most d . We show that when m is a prime power, this gives a family of high rate locally correctable codes and we can apply the decoding algorithm of the multiplicity code to correct the desired coordinate of a codeword. More precisely, we will establish the following theorem.

Theorem 4.1.1. *For every $\epsilon, \alpha > 0$ and for infinitely many N , there exists a family of codes of length N and rate $1 - \alpha$, which is $(O(N^\epsilon), \tau, 0.2)$ -locally correctable, where*

$$\tau = \Omega(1 / \exp(\log(1/\alpha) \log(1/\epsilon) \exp(\log(1/\epsilon)/\epsilon))).$$

The contribution of this part is not on parameters of the codes, but building a theoretical connection between two well known high-rate locally correctable codes, i.e., the multiplicity code and the lifted Reed-Solomon code, and answering the question whether we can construct high-rate LCCs with fewer order of derivatives. Note that when $m = 1$, our construction becomes the lifted Reed-Solomon code. Moreover, we show that the traditional multiplicity code is actually a subcode of our code.

4.2 Preliminaries

For a n -tuple

$$\mathbf{e} = (e_1, \dots, e_n) \in \mathbb{Z}_{\geq 0}^n,$$

we use $\mathbf{x}^{\mathbf{e}}$ to denote the monomial

$$\prod_{i=1}^n x_i^{e_i}.$$

Thus a polynomial f can be written as

$$f = \sum_{\mathbf{e}} c_{\mathbf{e}} \mathbf{x}^{\mathbf{e}}.$$

Here, we inherit the concept of support of polynomials and degree set from Chapter 3.

Let $\mathbf{w}$ be a point in $\mathbb{F}_q^n$. For polynomial $F \in \mathbb{F}_q[x_1, \dots, x_n]$, the order m evaluation of F at $\mathbf{w}$ is defined as

$$F^{(<m)}(\mathbf{w}) \triangleq (F^{(\mathbf{i})}(\mathbf{w}))_{wt(\mathbf{i}) < m}.$$

The multiplicity codes of order m evaluations of degree d polynomials in n variables, denoted by $Y_{n,m,d}$, consists of the order m evaluation of all polynomials of degree at most d , where d is smaller than mq , i.e.,

$$Y_{n,m,d} = \left\{ (F^{(<m)}(\mathbf{w}))_{\mathbf{w} \in \mathbb{F}_q^n} \mid \deg F \leq d \right\}.$$

The following result is proved in [4] and [27].

Theorem 4.2.1. *[Kopparty et. al. [4] and Yekhanin [27].] Let $Y_{n,m,d}$ be the multiplicity code of order m evaluations of degree d polynomials in n variables over $\mathbb{F}_q$. Suppose $q \geq \max\{10n, \frac{d+15m}{m}, 12(m+1)\}$. Let $\delta = 1 - \frac{d}{mq}$. Then the code is $(O(m)^n, \frac{\delta}{10}, 0.2)$ -locally correctable. The rate of the code is*

$$\frac{\binom{d+n}{n}}{\binom{m+n-1}{n} q^n},$$

which is at least $(1 - \frac{n^2}{m})(1 - \delta)$.

We now introduce an operation φ on the set of nonnegative integers, which will be used to describe the degree set of our code. For any nonnegative integer e ,

$$\varphi(e) = \begin{cases} e, & \text{if } e \leq mq - 1; \\ (e - m \bmod mq - m) + m, & \text{otherwise.} \end{cases}$$

The motivation behind the definition of φ is that if $\varphi(a) = b$, then we have

$$x^a \equiv x^b \pmod{(x^{mq} - x^m)}.$$

Let $\alpha, \beta \in \mathbb{F}_q^n$ and

$$l = \{\alpha + \lambda\beta \mid \lambda \in \mathbb{F}_q\}$$

be a line in $\mathbb{F}_q^n$. For a polynomial $F(\mathbf{x})$, we define the *restriction* of F on l as the univariate polynomial

$$F(\alpha + \lambda\beta) \pmod{\lambda^{mq} - \lambda^m}.$$

4.3 The Code Construction

4.3.1 Notations and Some Properties of Multiplicities

Let $q = p^s$ be a prime power and $\mathbb{F}_q$ be the finite field of q elements. Recall that the multiplicity code employs all the multivariate polynomials over $\mathbb{F}_q$ of degree at most d . In our construction, we employ multivariate polynomials with higher degree and require that the restriction on any line is of degree at most d . Our code consists of the order

m evaluations of such polynomials. To describe it more precisely, we start with some notations and properties of multiplicities which will be used in the construction of the code.

For an arbitrary positive integer m , we define

$$S_1(m) = \{e \in \mathbb{Z} \mid 0 \leq e < mq\}.$$

Let $S_n(m)$, for integer $n > 1$, be a set of n -tuples of integers defined inductively as follows.

$$\begin{aligned} S_n(m) = \bigcup_{k=1}^m \{ & (e_1, \dots, e_n) \in \mathbb{Z}^n \mid (k-1)q \leq e_n < kq \\ & \text{and } (e_1, \dots, e_{n-1}) \in S_{n-1}(m+1-k) \}. \end{aligned}$$

We denote $|S_n(m)|$ by $N_n(m)$.

Let $P_n(m)$ be the set of all polynomials whose support is contained in $S_n(m)$, i.e.,

$$P_n(m) = \{F(\mathbf{x}) \in \mathbb{F}_q[x_1, \dots, x_n] \mid F(\mathbf{x}) = \sum_{\mathbf{e} \in S_n(m)} c_{\mathbf{e}} \mathbf{x}^{\mathbf{e}}\}.$$

The following lemma shows that there is a one-to-one correspondence between the polynomials in $P_n(m)$ and their order m evaluations on $\mathbb{F}_q^n$.

Lemma 4.3.1. *Suppose*

$$F(\mathbf{x}) = \sum c_{\mathbf{e}} \mathbf{x}^{\mathbf{e}} \in P_n(m).$$

Then $F^{(<m)}(\mathbf{x})$ vanishes on all points in $\mathbb{F}_q^n$ if and only if F is the zero polynomial.

Proof. We prove it by induction on n . When $n = 1$, the lemma is trivial.

Suppose the lemma holds for $n - 1$ variables and any m . Now, write $F(\mathbf{x})$ as

$$\sum_{j=0}^{mq-1} f_j x_n^j,$$

where f_j is a polynomial in $x_1, \dots, x_{n-1}$.

Suppose

$$(k-1)q \leq e_n < kq$$

for some k . Fix a point

$$(a_1, \dots, a_{n-1}) \in \mathbb{F}_q^{n-1}.$$

For any $\mathbf{i} \in \mathbb{Z}_{\geq 0}^{n-1}$ with

$$wt(\mathbf{i}) \leq m - k,$$

we define

$$F_i(x_n) \triangleq F^{(\mathbf{i},0)}(x) = \sum f_j^{(\mathbf{i})}(a_1, \dots, a_{n-1}) x_n^j.$$

Note that

$$F_i^{(l)}(x_n) = F^{(\mathbf{i},l)}(a_1, \dots, a_{n-1}, x_n).$$

$F^{(<m)}$ vanishes on $\mathbb{F}_q^n$ implies $F_i^{(<k)}(x_n)$ vanishes on $\mathbb{F}_q$. Thus $F_i(x_n)$ is the zero polynomial. Therefore

$$f_j^{(\mathbf{i})}(a_1, \dots, a_{n-1}) = 0$$

for all $\mathbf{i}$ and all $(a_1, \dots, a_{n-1})$. In other words, $f_j^{(<m+1-k)}$ vanishes on $\mathbb{F}_q^{n-1}$. Then, f_j is the zero polynomial due to our assumption. $\square$

4.3.2 The Construction

Given $0 < \alpha, \epsilon < 1$, we set

$$n = \lceil \frac{1}{\epsilon} \rceil$$

and

$$b = \lceil \log_p n \rceil + 1.$$

Let t be a nonnegative integer. Let

$$m = p^t,$$

$$\alpha_t = \frac{\alpha}{1 - \frac{N_n(p^{t-b})}{N_n(p^t)}},$$

$$c = bp^{bn} \ln \frac{1}{\alpha_t} + t,$$

and

$$d = (1 - p^{-c})mq.$$

Let $D_{n,m,d}$ be a subset of $S_n(m)$ defined as follows.

$$\begin{aligned} D_{n,m,d} = \{ \mathbf{e} = (e_1, \dots, e_n) \in S_n(m) \mid & \text{for every } \mathbf{u} \in \mathbb{Z}_{\geq 0}^n \\ & \text{satisfying } \mathbf{u} \leq_p \mathbf{e}, \text{ we have } \varphi(\sum u_i) \leq d \}. \end{aligned}$$

Let $P_{n,m,d}$ be the set of polynomials whose support is contained in $D_{n,m,d}$, i.e.,

$$P_{n,m,d} = \{ F(\mathbf{x}) \in \mathbb{F}_q[x_1, \dots, x_n] \mid F(\mathbf{x}) = \sum_{\mathbf{e} \in D_{n,m,d}} c_{\mathbf{e}} \mathbf{x}^{\mathbf{e}} \}.$$

The code $C_{n,m,d}$ is defined as

$$C_{n,m,d} = \left\{ \left(F^{(<m)}(\mathbf{w}) \right)_{\mathbf{w} \in \mathbb{F}_q^n} \mid F \in P_{n,m,d} \right\}.$$

In other words, set $D_{n,m,d}$ is the degree set of the code $C_{n,m,d}$. The (relative) minimum distance of $C_{n,m,d}$ is p^{-c} . To recover one coordinate of the codeword, we can apply the local decoding algorithm of the multiplicity codes to $C_{n,m,d}$, thus if

$$q > \max\left\{10n, \frac{d+15m}{m}, 12(m+1)\right\},$$

the code $C_{n,m,d}$ is $(O(m)^n q, \frac{p^{-c}}{10}, 0.2)$ -locally correctable.

4.4 Proof of Theorem 4.1.1

We now estimate the size of the degree set of $C_{n,m,d}$ and show that the rate can be arbitrarily close to 1. To this end, we first present an equivalent definition of the set $S_n(m)$.

Lemma 4.4.1. *For positive integers n and m , we have*

$$S_n(m) = \left\{ \mathbf{e} = (e_1, \dots, e_n) \in \mathbb{Z}_{\geq 0}^n \mid \sum_{i=1}^n b_i \leq m-1 \text{ where } b_i = \left\lfloor \frac{e_i}{q} \right\rfloor \right\}.$$

Proof. We can write e_i as

$$a_i + qb_i$$

for every $i \in [n]$, where

$$0 \leq a_i < q.$$

Note that

$$\#\{\mathbf{e} \mid \sum_{i=1}^n b_i \leq m-1\} = \binom{n+m-1}{n} = \#S_n(m).$$

Thus, it is sufficient to prove that $S_n(m)$ contains the set

$$\{\mathbf{e} \mid \sum_{i=1}^n b_i \leq m-1\}.$$

We prove this by induction on n .

For $n = 1$, the lemma is trivial.

Suppose the lemma holds for $(n-1)$ -tuples. For n -tuples, suppose

$$\sum_{i=1}^n b_i \leq m-1.$$

If $b_n = k-1$, where $1 \leq k \leq m$, then

$$\sum_{i=1}^{n-1} b_i \leq m-k.$$

Thus we have

$$(e_1, \dots, e_{n-1}) \in S_n(m+1-k),$$

which implies $\mathbf{e} \in S_n(m)$.

□

The following lemma gives a characterization of the degree set.

Lemma 4.4.2. *Let b, c, d, t, n and m defined as before. If $\mathbf{e}$ lies in $S_n(m)$ and there exists*

$$j \in \{s + t - c, \dots, s + t - b\}$$

such that

$$e_i^{(j)} = \dots = e_i^{(j+b-1)} = 0$$

holds for every $i \in \{1, \dots, n\}$, then $\mathbf{e} \in D_{n,m,d}$.

Proof. Suppose $q = p^s$. Suppose $\mathbf{e}$ satisfies the required condition in the statement.

For $\mathbf{u} \leq_p \mathbf{e}$, for $1 \leq i \leq n$, we write u_i as

$$u_i = \alpha_i + p^{j+b}\beta_i + q\gamma_i,$$

where $\alpha, \beta, \gamma \in \mathbb{Z}_{\geq 0}$, $\alpha_i < p^j$ and $p^{j+b}\beta < q$. Let

$$A = \sum_{i=1}^n \alpha_i,$$

$$B = p^{j+b} \sum_{i=1}^n \beta_i$$

and

$$C = q \sum_{i=1}^n \gamma_i.$$

Suppose that

$$B + C = ymq + xp^{j+b}$$

for some

$$x < \frac{mq}{p^{j+b}}$$

and y . Due to lemma 4.4.1, we have

$$\sum_{i=1}^n \gamma_i \leq m - 1.$$

Moreover,

$$\begin{aligned} y &\leq \frac{B+C}{mq} \\ &= \frac{p^{j+b}n(p^{s-j-b}-1)+mq-q}{mq} \\ &= \frac{nq-p^{j+b}n+mq-q}{mq}. \end{aligned}$$

Thus we have

$$y \leq \left\lfloor \frac{n}{m} \right\rfloor + 1.$$

Therefore,

$$\begin{aligned} \varphi(B+C) &\leq \varphi(ymq) + \varphi(xp^{j+b}) \\ &\leq \left\lfloor \frac{n}{m} \right\rfloor + 1 + p^{j+b}(mp^{s-j-b} - 1) \\ &= \left\lfloor \frac{n}{m} \right\rfloor + 1 + mq - p^{j+b}. \end{aligned}$$

Then,

$$\begin{aligned} \varphi(A+B+C) &\leq \varphi(A) + \varphi(B+C) \\ &\leq n(p^j - 1) + \left\lfloor \frac{n}{m} \right\rfloor + 1 - mq - p^{j+b} \\ &\leq np^j + mq - p^{j+b} \\ &\leq p^{b-1}p^j + mq - p^{j+b} \\ &\leq mq - p^{j+b-1} \\ &< d. \end{aligned}$$

□

Now we are ready to estimate the rate.

Lemma 4.4.3. *Let b, c, d, t, m and n be defined as before. The rate of $C_{n,m,d}$ is at least*

$$1 - e^{-\frac{c-t}{bp^{nb}}} + e^{-\frac{c-t}{bp^{nb}}} \cdot \frac{N_n(p^{t-b})}{N_n(p^t)}.$$

Proof. Now we estimate size of the following set:

$$\{\mathbf{e} \mid \text{there exists } i \in [n] \text{ and } j \in \{s, \dots, s+b-1\} \text{ such that } e_i^{(j)} \neq 0\}.$$

For every i , we consider

$$e_i^{(s+t-c)}, \dots, e_i^{(s-1)}.$$

We divide these $c-t$ bits into $\frac{c-t}{b}$ blocks with size b . For any block, the probability that it is an all zero block is $(\frac{1}{p})^{bn}$. Thus, it is not an all zero block with probability

$$1 - \left(\frac{1}{p}\right)^{bn}.$$

The probability that there is at least one all zero block among these $\frac{c-t}{b}$ blocks is at least

$$1 - \left(1 - \left(\frac{1}{p}\right)^{bn}\right)^{\frac{c}{b}},$$

which is greater than

$$1 - e^{-\frac{c-t}{bp^{nb}}}.$$

Due to lemma 4.4.4, which we will prove later, there are

$$(1 - e^{-\frac{c-t}{bp^{bn}}})(N_n(m) - N_n(\frac{m}{p^b}))$$

vectors in the above set.

Now consider the set

$$\{\mathbf{e} | e_i^{(s)} = e_i^{(s+1)} \dots = e_i^{s+b-1} = 0\}.$$

This set is of size

$$N_n(\frac{m}{p^b})$$

due to lemma 4.4.4. The desired result then follows easily.

□

Lemma 4.4.4. *Suppose $m = p^t$ where p is a prime number. For $1 \leq l \leq t$, the set*

$$T_{n,m,l} = \{\mathbf{e} \in S_n(m) \mid \forall i, e_i^{(s)} = \dots = e_i^{(s+l-1)} = 0\}$$

is of size $N_n(\frac{m}{p^l})$.

Proof. Fix the integer l . We prove it by induction on n .

It is clear that the lemma holds when $n = 1$.

Suppose the lemma holds for $n < n'$.

$$\begin{aligned} T_{n',m,l} = \bigcup_{k \equiv 1 \pmod{p^l}} \{ & \mathbf{e} \in \mathbb{Z}^n \mid (k-1)q \leq e_n < kq \text{ and} \\ & (e_1, \dots, e_{n-1}) \in T_{n'-1, m+1-k, l} \}. \end{aligned}$$

Then we have

$$\begin{aligned}
|T_{n',m,l}| &= \sum_{k=0}^{\frac{m}{p^l}-1} q |T_{n'-1,m+1-(p^l k+1),l}| \\
&= \sum_{k=1}^{\frac{m}{p^l}} q |T_{n'-1,p^l k,l}| \\
&= \sum_{k=1}^{\frac{m}{p^l}} q^n \binom{n' + k - 2}{k - 1} \\
&= q^n \binom{n' - 1 + \frac{m}{p^l}}{\frac{m}{p^l} - 1} \\
&= N_{n'}\left(\frac{m}{p^l}\right).
\end{aligned}$$

□

Note that

$$\alpha = \left(1 - \frac{N_n(p^{t-b})}{N_n(p^t)}\right) e^{-\frac{c-t}{bp^{bn}}},$$

thus the rate of $C_{n,m,d}$ is at least $1 - \alpha$.

The following lemma estimates the size of $S_n(m)$.

Lemma 4.4.5. *For any positive integer n , we have*

$$N_n(m) = |S_n(m)| = \binom{n+m-1}{m-1} q^n.$$

Proof. We prove it by induction on n . When $n = 1$, the lemma is trivial.

Suppose the lemma holds for integers less than n . Then for $S_n(m)$ we have

$$\begin{aligned}
 |S_n(m)| &= \sum_{k=1}^m q |S_{n-1}(m+1-k)| \\
 &= \sum_{k=1}^m q |S_{n-1}(k)| \\
 &= \sum_{k=1}^m q^n \binom{n+k-2}{k-1} \\
 &= \binom{n+m-1}{m-1} q^n.
 \end{aligned}$$

□

4.5 Remark: The Connection Between Multiplicity Codes and Our Codes

Now we show that the traditional multiplicity code is a subcode of our code $C_{n,m,d}$.

Lemma 4.5.1. *The multiplicity code of order m evaluations of degree d polynomials in n variables over $\mathbb{F}_q$ is a subcode of $C_{n,m,d}$.*

Proof.

$$Y_n(m) = \{\mathbf{e} \mid \sum e_i < mq - m\}.$$

We want to show that

$$Y_n(m) \subseteq S_n(m).$$

We prove this by induction on n .

For one variable, it is clear that

$$Y_1(m) \subseteq S_1(m),$$

for any m .

Suppose that the lemma holds for less than n variables.

$$\begin{aligned} Y_n(m) &= \bigcup_{k=1}^m \{ \mathbf{e} \mid (k-1)q \leq e_n < kq \text{ and } \\ &\quad \sum_{i=1}^{n-1} e_i < mq - e_n \} \\ &\subseteq \bigcup_{k=1}^m \{ \mathbf{e} \mid (k-1)q \leq e_n < kq \text{ and } \\ &\quad \sum_{i=1}^{n-1} e_i < mq - (k-1)q \} \\ &= \bigcup_{k=1}^m \{ \mathbf{e} \mid (k-1)q \leq e_n < kq \text{ and } \\ &\quad (e_1, \dots, e_{n-1}) \in Y_n(m - k + 1) \} \\ &\subseteq \bigcup_{k=1}^m \{ \mathbf{e} \mid (k-1)q \leq e_n < kq \text{ and } \\ &\quad (e_1, \dots, e_{n-1}) \in S_n(m - k + 1) \} \\ &= S_n(m). \end{aligned}$$

If $\sum e_i$ is at most d , then for any $\mathbf{u}$ with $\mathbf{u} \leq_p \mathbf{e}$, we have

$$\sum u_i \leq \sum e_i \leq d < mq - m.$$

Thus,

$$\varphi(\sum u_i) = \sum u_i \leq d.$$

□

Here we should mention that, although we can obtain a larger family of high-rate

LCCs, to the best of our knowledge, the parameters of multiplicity codes are still the best in this family.

5. LOCAL CORRECTION OF MULTIPLE BITS

5.1 Background and Motivations

In reality, the event of multiple errors is not uncommon, and we want to recover them efficiently. In this chapter, we will explore the codes which admit multiple-error local correction algorithms.

Recall that a traditional locally decodable code is a code such that a single bit of a message can be recovered with high probability by visiting only a small number of bits of the corrupted codeword. If an arbitrary bit of a codeword (instead of a bit of a message) can be recovered, we say the code is locally correctable.

To compared with our generalized definition of locally correctable codes, we first state the original definition of locally correctable code formally. Let C be a q -ary code on coordinate set $\mathbb{F}_q^n$, for some integer n . We say the code C is (r, δ, ϵ) -locally correctable if there exists a randomized decoding algorithm $\mathcal{A}$ such that

1. For all $\mathbf{w} \in \mathbb{F}_q^n$, all $f \in C$, and all functions $g : \mathbb{F}_q^n \rightarrow \mathbb{F}_q$ satisfying

$$\Delta(f, g) \leq \delta,$$

we have

$$\Pr[\mathcal{A}^g(\mathbf{w}) = f(\mathbf{w})] \geq 1 - \epsilon.$$

2. $\mathcal{A}$ makes at most r queries to g .

However, in reality it is possible that multiple errors occur, and we want to recover them efficiently. This motivates the concept of local correction for multiple bits. As a generalization of locally correctable code, one may want to recover arbitrary t bits of the corrupted codeword, for some integer t .

Ideally, one would like t to be large and the query complexity to be small. They however can not be achieved simultaneously. There is a trade-off. On one hand, we have the classical error-correcting code which allows the number of errors within one half of the minimum distance to be recovered with N queries, where N denotes the length of the code. On the other hand, we have the locally correctable code which allows 1 bit to be recovered with a small number of queries.

For a (r, δ, ϵ) -locally correctable code, if we correct t bits one by one, i.e., run the decoding algorithm t times, the number of queries will be rt . Thus a natural question is whether there exists a code which allows t bits to be recovered with less query complexity.

5.2 A Generalized Definition of Locally Correctable Codes

We say a code C is $((r, \delta, \epsilon), t)$ -locally correctable if there exist a decoding algorithm such that any t positions in a codeword can be recovered with high probability by looking at only r other positions, even if there are δ fraction of errors. A formal definition is as follows.

Definition 5.2.1. Let C be a q -ary code on coordinate set $\mathbb{F}_q^n$, for some integer n . We call the code C a $((r, \delta, \epsilon), t)$ -locally correctable code if there exists a randomized decoding algorithm $\mathcal{A}$ such that

1. For all $\mathbf{w}_1, \dots, \mathbf{w}_t \in \mathbb{F}_q^n$, all $f \in C$, and all functions $g : \mathbb{F}_q^n \rightarrow \mathbb{F}_q$ satisfying

$$\Delta(f, g) \leq \delta,$$

we have

$$\Pr[\mathcal{A}^g(\mathbf{w}_i) = f(\mathbf{w}_i) \text{ for all } i \in [t]] \geq 1 - \epsilon.$$

2. $\mathcal{A}$ makes at most r queries to g .

Note that the traditional (r, δ, ϵ) -locally correctable code is actually $((r, \delta, \epsilon), 1)$ -locally correctable in the present notation. We call the parameter ϵ the *soundness error*. If the soundness error ϵ tends to 0 as the length tends to infinity, the code is said to be *highly sound*. Moreover, as with the classical error-correcting code, rate is also a parameter of interest.

5.3 A Theorem on Local Correction on Multiple Bits

In the following sections, we present a family of codes in which multiple bits of a codeword can be recovered by a small number of queries with probability approaching 1, even if there is a constant fraction of errors. At the same time, the rate of the code can be arbitrarily close to 1.

Our code generalizes the construction of the lifted Reed-Solomon code in [5], the

“decoding on curves” method in Gemmel and Sudan’s work [28], and the construction in Chapter 3.

More precisely, we have the following result:

Theorem 5.3.1. *For any $\alpha, \epsilon > 0$, there exists a constant τ , such that for infinitely many N , there is a code C of length N and rate $1 - \alpha$, which is $((O(N^\epsilon), \tau, O(\frac{1}{N^\epsilon})), t)$ -locally correctable, where t can be up to $O(\log N^\epsilon)^\epsilon$.*

As a comparison, if we want to recover $O(\log N^\epsilon)^\epsilon$ bits of a codeword in multiplicity code or lifted Reed-Solomon code by running their decoding algorithm repeatedly, where N denotes the length of the code, then the query complexity is $O(N^\epsilon(\log N^\epsilon)^\epsilon)$.

Note that if one want to correct t bits in a corrupted codeword, the query complexity will increase rapidly with t . Thus the number of positions one want to recover can not be too large. One obvious upper bound is $1/2$ of the minimum distance, since the decoder need to read the whole codeword in order to correct this number of errors. (For example, this bound is $\frac{\epsilon\alpha N}{4}$ for the multiplicity code.)

At the same time, the probability that our code can be recovered successfully is approaching 1. However, this is not necessarily an advantage, since one can simply use a majority vote to improve the success probability for any bit in a traditional locally correctable codes.

5.4 The Construction

At first we fix some notations which will be used in the sequel. Most of our notations follow [5] and [27].

For vector

$$\mathbf{e} = (e_1, \dots, e_n) \in \mathbb{Z}_{\geq 0}^n,$$

we use $\mathbf{x}^{\mathbf{e}}$ to denote the monomial

$$\prod_{i=1}^n x_i^{e_i}.$$

Thus a polynomial f can be written as

$$f = \sum_{\mathbf{e}} c_{\mathbf{e}} \mathbf{x}^{\mathbf{e}}.$$

For a integer a and a prime p , we use

$$\sum_i p^i a^{(i)}$$

to denote its base p expansion, where

$$0 \leq a^{(i)} \leq p - 1.$$

For two integers a and b , if we have $a^{(i)} \leq b^{(i)}$ for every i , then we say a is in the p -shadow of b , denoted by

$$a \leq_p b.$$

For two vectors $\mathbf{a} = (a_1, \dots, a_n), \mathbf{b} = (b_1, \dots, b_n) \in \mathbb{Z}^n$, we say $\mathbf{a}$ is in the p -shadow of $\mathbf{b}$ if $a_i \leq_p b_i$ holds for every i .

For a prime power q , the operation $\text{mod } *q$ is the map from nonnegative integers to

the set $\{0, \dots, q-1\}$ defined in Chapter 3. For a nonnegative integer a ,

$$a \bmod^* q = \begin{cases} a \bmod (q-1) & \text{if } a \bmod (q-1) \neq 0 \\ q-1 & \text{if } a \neq 0 \text{ and } (q-1) \nmid a \\ 0 & \text{if } a = 0. \end{cases}$$

Note that if

$$a \bmod^* q = b,$$

then we have

$$x^a \equiv x^b \pmod{(x^q - x)}.$$

We use $\mathbb{F}_q$ to denote the finite field of q elements, where $q = p^s$ for some integer s and p is a prime number. Suppose α and ϵ are constants in $(0, 1)$. Let

$$n = \lceil \frac{1}{\epsilon} \rceil,$$

$$b = \lceil \log_p(nt + n) \rceil + 1$$

and

$$c = \lceil bp^{bn} \ln \frac{1}{\alpha} \rceil.$$

Let

$$d = (1 - p^{-c})q.$$

Suppose t is an integer less than

$$\frac{1}{np^2} \left(\frac{s}{\ln \frac{1}{\alpha}} \right)^{\frac{1}{n+1}}.$$

We use S_0 to denote the set of curves in $\mathbb{F}_q^n$ of degree at most $t+1$. In other words, we

define

$$S_0 = \{\chi(\lambda) \mid \chi(\lambda) = \sum_{i=0}^{t+1} \lambda^i \mathbf{v}_i, \mathbf{v}_i \in \mathbb{F}_q^n \text{ for } 0 \leq i \leq t+1, \lambda \in \mathbb{F}_q\}.$$

Let

$$C = \{F(\mathbf{x}) \in \mathbb{F}_q[x_1, \dots, x_n] \mid \forall \chi(\lambda) \in S_0, \deg F(\chi(\lambda)) \leq d\}.$$

(Note that $\lambda^q = \lambda$ when $\lambda \in \mathbb{F}_q$.)

In other words, C consists of all the polynomials whose restriction on every curve in S_0 is of degree at most d . Each codeword is the evaluation of a polynomial in C at all points in $\mathbb{F}_q^n$. Denote the length of the code by N . Thus we have $N = q^n$. Moreover, the minimum distance of the code is at least $p^{-c}q$.

5.4.1 Proof of Theorem 5.3.1: Part 1

The proof of the main theorem in Chapter 5 is divided into the sections in the remainder of this chapter. In this section, we present the decoding algorithm and show that the code is $((O(N^\epsilon), \tau, O(\frac{1}{N^\epsilon})), t)$ -locally correctable.

Suppose $\mathbf{w}_1, \dots, \mathbf{w}_t$ are t distinct points in $\mathbb{F}_q^n$. Suppose F is a function from $\mathbb{F}_q^n$ to $\mathbb{F}_q$ and there exist a polynomial $G \in C$ such that

$$\Delta(F, G) \leq \tau,$$

where

$$\tau = \frac{p^{-c}}{4}.$$

We want to recover $G(\mathbf{w}_1), \dots, G(\mathbf{w}_t)$.

Let $\alpha_1 = 0$. Suppose $\alpha_2, \dots, \alpha_t$ are distinct elements in $\mathbb{F}_q \setminus \{0\}$. Let

$$S = \{\chi(\lambda) = \mathbf{w}_1 + \sum_{i=1}^{t+1} \lambda^i \mathbf{v}_i \mid \mathbf{v}_i \in \mathbb{F}_q^n, \chi(\alpha_i) = \mathbf{w}_i \text{ for } 2 \leq i \leq t\}.$$

Note that $\chi(\lambda) \in S$ implies

$$\chi(\alpha_1) = \mathbf{w}_1.$$

Before we describe the decoding procedure formally, we need to establish some properties of the set of curves, which plays an important role in the decoding algorithm.

Lemma 5.4.1. *Given $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{F}_q^n$, there is exactly one curve χ in S such that*

$$\chi(\lambda) = \mathbf{w}_1 + \lambda \mathbf{v}_1 + \lambda^2 \mathbf{v}_2 + \sum_{i=3}^{t+1} \lambda^i \mathbf{v}_i.$$

Proof. Consider the following equation system.

$$\begin{aligned} \chi(\alpha_2) &= \mathbf{w}_2 \\ &\dots \\ \chi(\alpha_t) &= \mathbf{w}_t \end{aligned}$$

This is equivalent to

$$\begin{aligned} \sum_{i=3}^{t+1} \alpha_2^i \mathbf{v}_i &= \mathbf{w}_2 - \mathbf{w}_1 - \alpha_2 \mathbf{v}_1 - \alpha_2^2 \mathbf{v}_2 \\ &\dots \\ \sum_{i=3}^{t+1} \alpha_t^i \mathbf{v}_i &= \mathbf{w}_t - \mathbf{w}_1 - \alpha_t \mathbf{v}_1 - \alpha_t^2 \mathbf{v}_2 \end{aligned}$$

We can write it coordinate-wise as follows:

$$= \begin{bmatrix} \begin{bmatrix} \alpha_2^3 & \cdots & \alpha_2^{t+1} \\ \vdots & \ddots & \vdots \\ \alpha_t^3 & \cdots & \alpha_t^{t+1} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{3,k} \\ \vdots \\ \mathbf{v}_{t+1,k} \end{bmatrix} \\ \mathbf{w}_{2,k} - \mathbf{w}_{1,k} - \alpha_2 \mathbf{v}_{1,k} - \alpha_2^2 \mathbf{v}_{2,k} \\ \vdots \\ \mathbf{w}_{2,k} - \mathbf{w}_{1,k} - \alpha_t \mathbf{v}_{1,k} - \alpha_t^2 \mathbf{v}_{2,k} \end{bmatrix},$$

for $1 \leq k \leq n$. The system for $\mathbf{v}_{3,k}, \dots, \mathbf{v}_{t+1,k}$ has a unique solution.

□

As a corollary, we have

$$|S| = q^{2n}.$$

Now, we denote the curve $\chi(\lambda)$ in the lemma by

$$\chi_{\mathbf{v}_1, \mathbf{v}_2}(\lambda).$$

Lemma 5.4.2. *For any $\mathbf{a}$ and $\mathbf{b}$ in $\mathbb{F}_q^n$, and distinct elements λ_1 and λ_2 in $\mathbb{F}_q \setminus \{\alpha_1, \dots, \alpha_t\}$, we have*

$$\begin{aligned} & \Pr_{\mathbf{v}_1, \mathbf{v}_2} [\chi_{\mathbf{v}_1, \mathbf{v}_2}(\lambda_1) = \mathbf{a}, \chi_{\mathbf{v}_1, \mathbf{v}_2}(\lambda_2) = \mathbf{b}] \\ &= \Pr_{\mathbf{v}_1, \mathbf{v}_2} [\chi_{\mathbf{v}_1, \mathbf{v}_2}(\lambda_1) = \mathbf{a}] \Pr_{\mathbf{v}_1, \mathbf{v}_2} [\chi_{\mathbf{v}_1, \mathbf{v}_2}(\lambda_2) = \mathbf{b}] \\ &= \frac{1}{q^{2n}}. \end{aligned}$$

Proof. Fix λ_1 and λ_2 . We want to compute the size of the following set:

$$\{(\chi_{\mathbf{v}_1, \mathbf{v}_2}(\lambda_1), \chi_{\mathbf{v}_1, \mathbf{v}_2}(\lambda_2)) \mid \chi_{\mathbf{v}_1, \mathbf{v}_2} \in S, \mathbf{v}_1, \mathbf{v}_2 \in \mathbb{F}_q^n\}.$$

Denote the above set by A . We claim that for any $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^n$, there is $\mathbf{v}_1$ and $\mathbf{v}_2$ such that

$$\chi_{\mathbf{v}_1, \mathbf{v}_2}(\lambda_1) = \mathbf{a}$$

and

$$\chi_{\mathbf{v}_1, \mathbf{v}_2}(\lambda_2) = \mathbf{b}.$$

To prove this claim, we should consider the following system of equations:

$$\begin{aligned} \mathbf{w}_1 + \sum_{i=1}^{t+1} \alpha_2^i \mathbf{v}_i &= \mathbf{w}_2 \\ &\dots \\ \mathbf{w}_1 + \sum_{i=1}^{t+1} \alpha_t^i \mathbf{v}_i &= \mathbf{w}_t \cdot \\ \mathbf{w}_1 + \sum_{i=1}^{t+1} \lambda_1^i \mathbf{v}_i &= \mathbf{a} \\ \mathbf{w}_1 + \sum_{i=1}^{t+1} \lambda_2^i \mathbf{v}_i &= \mathbf{b} \end{aligned}$$

We rewrite it by coordinates:

$$\begin{bmatrix} \alpha_2 & \dots & \alpha_2^{t+1} \\ \vdots & \ddots & \vdots \\ \alpha_t & \dots & \alpha_t^{t+1} \\ \lambda_1 & \dots & \lambda_1^{t+1} \\ \lambda_2 & \dots & \lambda_2^{t+1} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{1,k} \\ \vdots \\ \mathbf{v}_{t+1,k} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_{2,k} - \mathbf{w}_{1,k} \\ \vdots \\ \mathbf{w}_{t,k} - \mathbf{w}_{1,k} \\ \mathbf{a}_k - \mathbf{w}_{1,k} \\ \mathbf{b}_k - \mathbf{w}_{1,k} \end{bmatrix},$$

for $1 \leq k \leq n$. There is a unique solution for each system. Thus set A is of size q^{2n} . $\square$

Thus, we have showed that the points on any curve in S is pairwise independent. Now

we are ready to present the decoding algorithm.

5.4.2 The Decoding Algorithm

The decoding procedure generalizes the “decoding on curves” method in [28] and can be described as follows:

Input: $\mathbf{w}_1, \dots, \mathbf{w}_t$ and a function F from $\mathbb{F}_q^n$ to $\mathbb{F}_q$ such that there exists $G \in C$ and

$$\Delta(F, G) \leq \tau.$$

Step 1: Choose distinct elements

$$\alpha_2, \dots, \alpha_t \in \mathbb{F}_q \setminus \{0\}.$$

Let S be the set of curves as we defined before.

Step 2: Choose $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{F}_q^n$ randomly.

Step 3: Let

$$f(\lambda) = F(\chi_{\mathbf{v}_1, \mathbf{v}_2}(\lambda)).$$

Decode on $\chi_{\mathbf{v}_1, \mathbf{v}_2}$ by using the Reed-Solomon decoding algorithm.

Output: $G(\mathbf{w}_1), \dots, G(\mathbf{w}_t)$.

Note that the restriction on a curve $\chi \in S$ of any $G \in C$ is a polynomial of degree at most

$$(1 - p^{-c})q.$$

In other words, if we denote $G(\chi(\lambda))$ by $g(\lambda)$, then

$$\deg g(\lambda) \leq (1 - p^{-c})q.$$

If the number of errors on χ is less than

$$\frac{p^{-c}}{2}q,$$

then we can recover $g(\lambda)$. The following lemma analyses the probability that $g(\lambda)$ can be recovered successfully.

Lemma 5.4.3. *Suppose χ is an arbitrary curve in S . Denote the number of errors on χ by Y . Then, we have*

$$\Pr[Y \geq \frac{p^{-c}}{2}q] \leq (\frac{1}{\tau} - 1)\frac{1}{q}.$$

Proof. We define a variable Y_λ as follows:

$$Y_\lambda = \begin{cases} 0 & \text{if there is no error at } \chi(\lambda) \\ 1 & \text{otherwise} \end{cases}.$$

Then we have the expectation

$$\mathbb{E}[Y_\lambda] = \tau$$

and the variance

$$\text{Var}[Y_\lambda] = \tau - \tau^2.$$

It is clear that

$$Y = \sum_{\lambda \in \mathbb{F}_q} Y_\lambda.$$

Since Y_λ s are pairwise independent, we have

$$\mathbb{E}[Y] = \tau q$$

and

$$\text{Var}[Y] = (\tau - \tau^2)q.$$

Now, an application of Chebyshev's inequality on variable Y completes the proof. $\square$

In other words, we can recover the desired bits successfully with probability at least $1 - O_{\epsilon, \alpha}(\frac{1}{q})$.

5.4.3 Proof of Theorem 5.3.1: Part 2

In this section, we show that the code C is of high rate. To this end, we need the following two lemmas to characterize the degree set.

Lemma 5.4.4. *For any n -tuple*

$$\mathbf{e} \in \{0, \dots, q-1\}^n,$$

we have $\mathbf{x}^e$ lies in C if and only if $\mathbf{e}$ lies in the following set:

$$\begin{aligned} D = & \{ \mathbf{e} \in \{0, \dots, q-1\}^n \mid \text{for all } n\text{-tuples} \\ & \mathbf{u}_1, \dots, \mathbf{u}_{t+1} \in \{0, \dots, q-1\}^n \text{ satisfying} \\ & \mathbf{u}_{t+1} \leq_p \dots \leq_p \mathbf{u}_1 \leq_p \mathbf{e}, \text{ we have} \\ & \sum_{i=1}^n \sum_{j=i}^{t+1} u_{j,i} \pmod{*q} \leq d \}. \end{aligned}$$

Proof. Let us consider the i -th coordinate where $1 \leq i \leq n$. If we expand

$$(v_{0,i} + \sum_{j=1}^{t+1} \lambda^j v_{j,i})^{e_i},$$

due to Lucas' Theorem, all the terms other than

$$\lambda^{u_{1,i} + \dots + u_{t+1,i}},$$

where

$$u_{t+1,i} \leq_p \dots \leq_p u_{1,i} \leq_p e_i,$$

will have coefficients zero. Note that

$$\lambda^k \pmod{(\lambda^q - \lambda)} = \lambda^{k \pmod{*q}}$$

for $k \in \mathbb{Z}_{\geq 0}$. The lemma then follows easily.

□

Thus, the set D is the degree set of C .

Lemma 5.4.5. *If there exists*

$$j \in \{s - c, \dots, s - b\}$$

such that for $1 \leq i \leq n$ we have

$$e_i^{(j)} = \dots = e_i^{(j+b-1)} = 0,$$

then $\mathbf{e} \in D$.

Proof. Let

$$A = \sum_{i=1}^n \sum_{k=1}^{t+1} \sum_{l=0}^{j-1} u_{k,i}^{(l)} p^l,$$

and

$$B = \sum_{i=1}^n \sum_{k=1}^{t+1} \sum_{l=j+b}^{s-1} u_{k,i}^{(l)} p^l.$$

Then we have

$$A \leq n(t+1)(p^j - 1)$$

and

$$B = p^{j+b} \sum_{i=1}^n \sum_{k=1}^{t+1} \sum_{l=0}^{s-1-j-b} u_{k,i}^{(l+j+b)} p^l.$$

Suppose

$$B = xq + yp^{j+b}$$

for some $x, y \in \mathbb{Z}$ and

$$yp^{j+b} \leq q - 1.$$

It is obvious that

$$x < n(t+1)$$

and

$$y \leq p^{s-j-b} - 1.$$

The following equation completes the proof.

$$\begin{aligned}
A + B \bmod^* q &\leq (A \bmod^* q) + (xq \bmod^* q) \\
&\quad + (yp^{j+b} \bmod^* q) \\
&< n(t+1)(p^j - 1) + n(t+1) + p^s - p^{j+b} \\
&= n(t+1)p^j + p^s - p^{j+b} \\
&\leq p^{b-1+j} + p^s - p^{j+b} \\
&\leq p^s - p^{j+b-1} \\
&\leq p^s - p^{s-c+b-1} \\
&< p^s - p^{s-c} \\
&< d.
\end{aligned}$$

□

Note that we should set t such that $c < s$. Under this requirement, t can be up to $O(\log_p q)^{\frac{1}{n+1}}$.

For a random vector $\mathbf{e} = (e_1, \dots, e_n)$, write e_i as

$$\sum_{j=0}^{s-1} e_i^{(j)} p^j.$$

We divide the last c bits of the expansion,

$$e_i^{(s-c)}, \dots, e_i^{(s-1)},$$

into $\frac{c}{b}$ blocks of size b . For any block, the probability that it is an all zero block is $\left(\frac{1}{p}\right)^{bn}$. Thus, it is not an all zero block with probability

$$1 - \left(\frac{1}{p}\right)^{bn}.$$

The probability that there is at least one all zero block among these $\frac{c}{b}$ blocks is at least

$$1 - \left(1 - \left(\frac{1}{p}\right)^{bn}\right)^{\frac{c}{b}},$$

which is greater than $1 - e^{-\frac{c}{bp^{bn}}}$. Thus, the dimension of the code is at least $(1 - \alpha)N$, which implies that the rate is at least $1 - \alpha$.

6. CONCLUSIONS

This chapter summarizes this dissertation and gives some concluding remarks on the problems studied and the results achieved. We also give suggestions for future work on this topic.

6.1 Concluding Remarks and Future Directions

In this thesis, we generalize the results of Guo et al. [5] by lifting multivariate polynomials on curves to construct high-rate codes. A corresponding correcting algorithm that uses the decoding on curves strategy is developed to correct errors with high probability, even when the number of errors is a constant fraction of the code length. This gives a family of high-rate locally correctable codes that are highly sound.

Then, we take a deeper look at the famous multiplicity codes. We construct a new class of locally correctable codes by combining the lifting method in [5] with multiplicity codes in [4], and show that this class of codes can achieve high rate. Moreover, we prove that the traditional multiplicity code can be seen as a subcode of our construction.

Furthermore, we generalize the concept of local correction to locally correcting multiple bits. We present a family of high-rate and highly sound code which allows multiple bits to be recovered with a small query complexity, even when a constant fraction of errors exist, by generalizing the construction of lifted Reed-Solomon codes and multiplicity

codes, and the decoding on curves method.

The distance-amplification method has been applied in [23] to construct locally correctable codes which have good relationship between the rate and minimum distance. This method works well with the suitably chosen base code in [23], and a locally correctable codes approaching the Singleton bound is produced. When we consider the local correction of multiple bits, a future direction is to investigate whether we can generalize the distance-amplification method and produce codes which admit local correction on multiple bits and have better parameters.

Moreover, the lifting method has been applied to algebraic geometry codes successfully in [25] and [26] to construct new locally correctable codes. A natural question is whether it is possible to construct high-rate lifted algebraic geometry codes which can correct multiple errors.

BIBLIOGRAPHY

- [1] Y. M. Chee, L. Wu, and C. Xing, “Correcting on curves and highly sound locally correctable codes of high rate,” in *ISIT 2014 - Proceedings of the 2014 IEEE International Symposium on Information Theory*, 2014, pp.2964-2966. [i]
- [2] L. Wu, “Revisiting the multiplicity codes: A new class of high-rate locally correctable codes,” in *53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015, pp. 509-513. [i]
- [3] L. Wu, “Locally correcting multiple bits and codes of high rate,” in *IEEE Information Theory Workshop - Fall (ITW)*, 2015, pp. 124-128. [i]
- [4] S. Kopparty, S. Saraf, and S. Yekhanin, “High-rate codes with sublinear-time decoding,” in *STOC’ 11 - Proceedings of the 43rd ACM Symposium on Theory of Computing*, 2011, pp. 167-176. [v, vi, 5, 6, 9, 12, 21, 29, 43, 45, 79]
- [5] A. Guo, S. Kopparty, and M. Sudan, “New affine-invariant codes from lifting”, in *ITCS’ 13 - Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*. New York: ACM, 2013, pp. 529-540. [v, vi, 5, 7, 9, 11, 27, 28, 29, 31, 33, 34, 63, 64, 79]

-
- [6] M. Blum and S. Kannan, "Designing programs that check their work," *J. Assoc. Comput. Mach.*, vol. 42, no. 1, 1995, pp. 269-291. [3]
- [7] R. J. Lipton, "Efficient checking of computations," in *Proceedings of the 7th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, Lecture Notes in Computer Science, vol. 415. Springer, 1990, pp. 207-215. [3]
- [8] L. Babai, L. Fortnow, and L. Levin, and M. Szegedy, "Checking computations in polylogarithmic time," in *23rd ACM Symposium on Theory of Computing*, 1991, pp. 21-31. [3]
- [9] J. Katz and L. Trevisan, "On the efficiency of local decoding procedures for error-correcting codes," in *32nd ACM Symposium on Theory of Computing*, 2000, pp. 80-86. [3]
- [10] D. Beaver and J. Feigenbaum, "Hiding instances in multioracle queries," In *Proceedings of 7th International Symposium on Theoretical Aspects of Computer Science (STACS)*, Lecture Notes in Computer Science, vol. 415, Springer, 1990, pp. 3748. [3]
- [11] C. Lund, L. Fortnow, H. J. Karloff, and N. Nisan, "Algebraic methods for interactive proof systems," *Journal of the ACM*, Vol. 39, No. 4, 1992, pp. 859-868. [3]
- [12] S. Arora and S. Safra, "Probabilistic checkable proofs: A new characterization of NP," *Journal of ACM*, vol. 45 No. 1, 1998, pp. 701-722. [3]
- [13] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, "Proof verification and the hardness of approximation problems," *Journal of the ACM*, vol. 45, No. 3, 1998, pp. 501-555. [3]

-
- [14] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson, "BPP has subexponential time simulations unless EXPTIME has publishable proofs," in *Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, 1991, pp. 213 - 219. [3]
- [15] M. Sudan, L. Trevisan, and S. Vadhan, "Pseudorandom generators without the XOR lemma," In *Proceedings of the 39th ACM Symposium on Theory of Computing (STOC)*, 1999, pp. 537-546. [3]
- [16] S. Arora and M. Sudan, "Improved low-degree testing and its applications," *Combinatorica*, vol. 23, no. 3, 2003, pp. 365-426. [3]
- [17] R. Shaltiel and C. Umans, "Simple extractors for all min-entropies and a new pseudorandom generator," *Journal of the ACM*, vol. 52, no. 2, 2005, pp. 172-216. [3]
- [18] A. Ambainis, "Upper bound on the communication complexity of private information retrieval," In *32nd International Colloquium on Automata, Languages and Programming (ICALP)*, 1997. [3]
- [19] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," *Journal of the ACM*, vol 45, 1998, pp. 965-981. [3]
- [20] O. Goldreich, H. Karloff, L. Schulman, and L. Trevisan, "Lower bounds for locally decodable codes and private information retrieval", in *17th IEEE Annual Conference on Computational Complexity*, 2002. [3]
- [21] O. Barkol, Y. Ishai, and E. Weinreb, "On locally decodable codes, self-correctable codes, and t-private PIR," In *International Workshop on Randomization and Computation (RANDOM)*, 2007, pp 311-325. [3]

-
- [22] A. Shokrollahi, “LDPC codes: An introduction,” In *Coding, Cryptography and Combinatorics*, 2004, pp. 85-110. [3]
- [23] S. Kopparty, O. Meir, N. Ron-Zewi, and S. Saraf, “High-rate locally-correctable and locally-testable codes with sub-polynomial query complexity,” *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, 2016, pp. 202-215. [8, 80]
- [24] N. Alon and M. Luby, “A linear time erasure-resilient code with nearly optimal recovery,” *IEEE Transactions on Information Theory*, vol. 42, no. 6, 1996, pp. 1732-1736. [8]
- [25] A. Guo, “High rate locally correctable codes via lifting,” *IEEE Transactions on Information Theory*, 2016, to appear. [9, 80]
- [26] E. Ben-Sasson, A. Gabizon, Y. Kaplan, S. Kopparty, and S. Saraf, *A new family of locally correctable codes based on degree-lifted algebraic geometry codes*. “Electronic Colloquium on Computational Complexity (ECCC)”, TR12-148, 2012. [9, 80]
- [27] S. Yekhanin, *Locally Decodable Codes*, ser. Foundations and Trends in Theoretical Computer Science. NOW, 2012, vol. 6, no. 3. [3, 10, 11, 30, 31, 45, 64]
- [28] P. Gemmell and M. Sudan, “Highly resilient correctors for polynomials,” *Inform. Process. Lett.*, vol.43, no.4, 1992, pp.169-174. [11, 14, 30, 64, 71]
- [29] A. S. Rawat, A. Mazumdar, and S. Vishwanath, “On cooperative local repair in distributed storage,” in *CISS 2014 - 48th Annual Conference on Information Sciences and Systems*, 2014, pp. 1-5. [15]

-
- [30] A. S. Rawat, D. S. Papailiopoulos, A. G. Dimakis, and S. Vishwanath, “Locality and availability in distributed storage,” *Proceedings of the 2014 IEEE International Symposium on Information Theory*, 2014, pp. 681-685. [16]
- [31] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, “On the locality of codeword symbols,” *IEEE Transactions on Information Theory*, vol. 58, no. 11, Nov. 2012, pp. 6925-6934. [16]
- [32] I. Tamo and A. Barg, “A family of optimal locally recoverable codes,” *IEEE Transactions on Information Theory*, vol. 60, no. 80, Aug. 2014, pp. 4661-4676. [16]
- [33] Z. Dvir, S. Kopparty, S. Saraf, and M. Sudan, “Extensions to the method of multiplicities, with applications to Kakeya sets and mergers,” *In 50th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2009, pp. 181-190. [21]
- [34] E. Ben-Sasson, G. Maatouk, A. Shpilka, and M. Sudan, “Symmetric LDPC codes are not necessarily locally testable,” *In IEEE 26th annual Conference on Computational Complexity (CCC)*, 2011. [27]
- [35] G. Grimmett and D. Stirzaker. *Probability and Random Processes*, 3rd edition. Oxford University Press, 2001. [36]

INDEX

code, 17

codeword, 17

degree of a polynomial, 21

degree set, 32

dimension, 17

distance, 18

Hamming weight, 18

Hasse derivative, 21

high-rate locally correctable code, 19

highly sound code, 19

length, 17

linear code, 18

locally correctable code, 19

minimum distance, 18

multiplicity, 21

multiplicity code, 23

rate, 17

relative distance, 18

soundness error, 19

support of a polynomial, 32

weight of a vector, 20