UNIVERSITY OF CALIFORNIA,
IRVINE

Reed-Solomon Codes and the Deep Hole Problem

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mathematics

by

Matt Keti

Dissertation Committee:
Professor Daqing Wan, Chair
Professor Karl Rubin
Professor Song-Ying Li

2015

Table of Contents

List of Figures

List of Tables

Acknowledgements

My deepest gratitude to my advisor Professor Daqing Wan for his continued support and mathematical guidance over the years, and for helping me make the final push toward the completion of this dissertation. It has been a tremendous pleasure to have been his student.

I'd also like to thank Donna McConnell for her help navigating the bureaucracy of the program and advocacy benefitting the graduate students here.

Finally, much appreciation to my office neighbours: Cynthia Northrup for our enlightening discussions on education; Luke Smith for our collaboration on number theory research and logistical issues; Wei-Kuo Chen, Mustafa Said, and Ru-Fei Ren for sharing their personal research ideas and giving me general assistance. And to all of my colleagues, thank you for your friendship throughout the program. It's been a lot of fun, and I wish you the very best.

Curriculum Vitae

Matt Keti

Education

Ph.D. in Mathematics, University of California, Irvine, 2015
M.S. in Mathematics, University of California, Irvine, 2011
B.S. in Mathematics, University of California, Irvine, 2009

Publications

Computing Error Distance of Reed-Solomon Codes (2012)
with Daqing Wan and Guizhen Zhu. Submitted for publication.

Deep Holes in Reed-Solomon Codes Based on Dickson Polynomials (2015)
with Daqing Wan. Preprint.

Teaching

As the instructor

Pre-Calculus
Single-variable Calculus
Multi-variable Calculus

As the teaching assistant

Single-variable Calculus
Multi-variable Calculus
Linear Algebra
Introduction to Abstract Mathematics
Number Theory

Abstract of the Dissertation

Reed-Solomon Codes and the Deep Hole Problem

By

Matt Keti

Doctor of Philosophy in Mathematics
University of California, Irvine, 2015
Professor Daqing Wan, Chair

In many types of modern communication, a message is transmitted over a noisy medium. When this is done, there is a chance that the message will be corrupted. An error-correcting code adds redundant information to the message which allows the receiver to detect and correct errors accrued during the transmission. We will study the famous Reed-Solomon code (found in QR codes, compact discs, deep space probes,...) and investigate the limits of its error-correcting capacity. It can be shown that understanding this is related to understanding the "deep hole" problem, which is a question of determining when a received message has, in a sense, incurred the worst possible corruption. We partially resolve this in its traditional context, when the code is based on the finite field $\mathbb{F}_q$ or $\mathbb{F}_q^*$, as well as new contexts, when it is based on a subgroup of $\mathbb{F}_q^*$ or the image of a Dickson polynomial. This is a new and important problem that could give insight on the true error-correcting potential of the Reed-Solomon code.

Chapter 1

Introduction

1.1 History and the Basics

In many areas of modern communications, some data must be sent over a noisy medium. Action must be taken in order for the receiver to correctly interpret the data. One way to do this is to use "forward error correction" (FEC), that is, to add additional redundant information to the data so that the receiver can recover corrupted parts of a message. Some examples of where this may be necessary are:

1. QR codes: to protect damaged or obstructed codes
2. Compact discs: to recover from scratches or dust on the surface
3. Deep space probes: to transmit data in the presence of cosmic or planetary noise

1.1.1 A Simple Code

Before the theory of error-correcting codes was established, people employed very simple procedures, one of which is now called a repetition code. To set up a repetition code, one declares a number N to be the number of times a message is to be repeated. This number depends on the amount of noise present in the medium; a higher number is required for higher noise rates. Then, the sender simply sends a given message symbol N times. More formally,

Procedure (Repetition Code)**.** Fix N to be the repetition value.

Input: A (binary) message symbol B.

Output: The string $\underbrace{BBB\ldots B}_{N \text{ times}}$, to be transmitted.

The receiver can recover a corrupted string by taking the symbol that occurs most frequently.

Repetition codes can be implemented easily due to their simplicity, but they are very inefficient, due to the fact that their data output is very low. This would later motivate the search for better ways to protect data.

1.1.2 The First Nontrivial Code: Hamming $(7, 4)$

In 1950, Richard Hamming of Bell Labs published the first nontrivial error-correcting code, which was a result of his attempts to mitigate read errors in binary punchcard readers. His idea was add redundancy by associating 4 data bits with a system of 3 equations, creating a code of length 7. This is known today as Hamming $(7, 4)$.

Procedure (Hamming $(7, 4)$ Encoding)**.**

Input: The message vector $m = (m_1, m_2, m_3, m_4)^T$, where the m_i are elements of $\mathbb{F}_2$.

Output: The codeword vector $c = Gm$, to be transmitted, where

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Procedure (Hamming $(7, 4)$ Decoding)**.**

Input: A received message vector $r = (r_1, r_2, r_3, r_4, r_5, r_6, r_7)^T$.

Computation:

1. Calculate the syndrome vector Hr, where

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

2. If the syndrome vector is not zero, treat it as a binary number.
3. In the received message, flip the bit in the position corresponding to that number.

Output: The message vector $(r_3, r_5, r_6, r_7)^T$.

In addition to developing this code, Hamming also introduced new concepts in the area of error-correction.

Definition.

- Message block size: the number of data symbols to be sent, denoted by k
- Code block size: the number of data symbols plus redundant symbols, denoted by n
- Hamming distance: the number of coordinates in which two words differ, denoted by $d(\cdot, \cdot)$
- Minimum distance: the shortest distance between any two codewords
- Information rate: the measure of the amount of data sent versus total code block size, given by k/n

The minimum distance gives information on the error-correcting capacity of the code. The consequences can be seen from the following figure, where d is the minimum distance, C_i are codewords, and r is a received word:

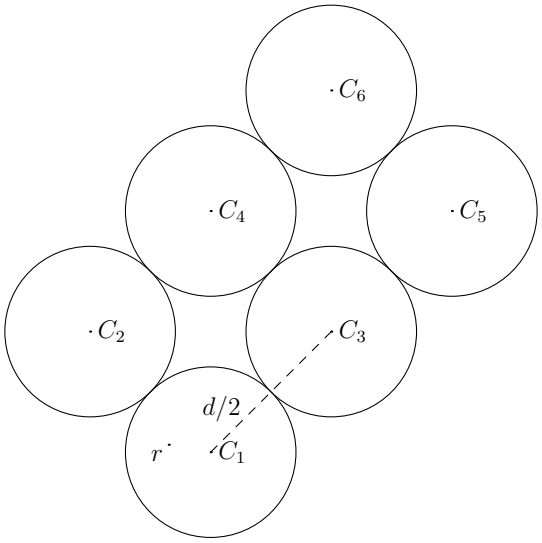

Figure 1.1: Geometric interpretation of minimum distance

The minimum distance tells us that any two codewords are separated at least by distance d. If a received message r falls within Hamming distance $d/2$ of a codeword, then it can be

unambiguously decoded to that codeword. In other words, a code can always correct less than $d/2$ errors in a received word.

Example.

Code	(odd) N-fold repetition	Hamming $(7, 4)$
Message block size	1	4
Code block size	N	7
Minimum distance	N	3
Correctable errors	$\lfloor N/2 \rfloor$	1
Information rate	$1/N$	$4/7$

Table 1.1: Properties of repetition and Hamming codes

Now with these concepts defined, the goal is to find better codes in terms of error-correcting capacity (i.e. large minimum distance) and high information rates.

1.2 Reed-Solomon Codes

1.2.1 Encoding and Decoding

In 1960, Irving S. Reed and Gustave Solomon published a paper titled 'Polynomial Codes Over Certain Finite Fields' [23]. Here they outlined a new error-correcting code that was based on sampling points on a polynomial. They used the idea that a polynomial of degree $k-1$ is determined by k of its points, and that if we know $n > k$ points, we can recover the original polynomial even if some of the points go missing or are corrupted. This was described in the following way:

Procedure (Reed-Solomon Encoding, Original Formulation)**.** Fix a finite field $\mathbb{F}_q$, a message block size k, and a subset $D = \{x_1, x_2, \ldots, x_n\} \subseteq \mathbb{F}_q$ so that $n > k$.

Input: The message $m = (m_0, m_1, m_2, \ldots, m_{k-1})$, represented by the polynomial

$$m(x) = m_0 + m_1 x + m_2 x^2 + \ldots + m_{k-1} x^{k-1}$$

where the m_i are elements of $\mathbb{F}_q$.

Output: The codeword $(m(x_1), m(x_2), \ldots, m(x_n))$, to be transmitted.

Example. Take the finite field $\mathbb{F}_{2^2}$, α a root of $x^2 + x + 1 \in \mathbb{F}_2[x]$, the message block length 2, and $D = \mathbb{F}_{2^2}$. Suppose $m = (10, 11)$. Then,

$$m(x) = 10 + 11x = \alpha + (\alpha + 1)x$$

and the transmitted codeword will be

$$(m(00), m(01), m(10), m(11)) = (10, 01, 11, 00)$$

In summary, our original message was 1011, and we will transmit the codeword 10011100.

Procedure (Reed-Solomon Decoding, Original Formulation)**.**

Input: A received message $r = (r_1, r_2, \ldots, r_n)$.

Computation: Find the interpolated polynomials of all combinations of k points from

$$(x_1, r_1), (x_2, r_2), \ldots, (x_n, r_n)$$

Output: The most popular polynomial.

Reed and Solomon gave an analysis of this algorithm and used a combinatorial argument to find the maximum number of correctable errors. To calculate this, suppose that there are t coordinate errors. The correct polynomial will therefore appear $\binom{n-t}{k}$ times, and any one incorrect polynomial will appear at most $\binom{t+(k-1)}{k}$ times. To recover the correct polynomial, we must have

$$\binom{n-t}{k} > \binom{t+(k-1)}{k}$$

which is true if and only if $n - t > t + (k - 1)$. Rearranging this inequality shows that we can correct up to $(n - k + 1)/2$ errors.

Example. Suppose the message $r = 10101100$ is received. Up to $\lfloor (4-2+1)/2 \rfloor = 1$ error can be corrected. $\binom{4}{2} = 6$ pairs of points must be examined. The frequencies are

# of occurrences	Polynomial
2	$10 + 11x$
1	$11 + 11x$
1	$11 + 01x$
1	$00 + 10x$
1	$10 + 00x$

Table 1.2: Frequencies of interpolated words

The most popular polynomial is $10 + 11x$, so the decoded message is $m = 1011$.

Unfortunately, Reed and Solomon's original decoding algorithm is infeasible except for very small codes. In many practical applications, we need to use fields such as $\mathbb{F}_{2^8}$. One popular setup is to use $D = \mathbb{F}_{2^8}^*$ and message block size $k = 223$ (i.e. a $(255, 223)$ Reed-Solomon code). Using this decoding procedure requires that we examine $\binom{255}{223} \approx 5.1 \times 10^{40}$ subsets. Even if we could examine one million subsets per second, it would take about 1.6×10^{27} years to complete. With this in mind, Reed-Solomon codes have been formulated in another way to allow for efficient decoding.

The most popular method to phrase Reed-Solomon codes is in terms of BCH (Bose-Chadhuri-Hocquenghem) codes. Here is the idea: suppose $g(x)$ is a polynomial with roots $\{x_1, x_2, \ldots, x_n\}$. Let $m(x)$ be some other polynomial. Then take $c(x) = m(x)g(x)$. If we make no mistake about $c(x)$, then

$$c(x_1) = c(x_2) = \ldots = c(x_n) = 0$$

If any of the terms is not zero, then some of the coefficients of $c(x)$ are in error, and we can use those values to try to recover $c(x)$.

Procedure (Reed-Solomon Encoding, BCH Formulation)**.** Fix a finite field $\mathbb{F}_q$, a generator α of $\mathbb{F}_q^*$, and an error tolerance t. Set

$$g(x) = (x - \alpha)(x - \alpha^2) \cdots (x - \alpha^{2t})$$

Input: The message $m = (m_0, m_1, m_2, \ldots, m_{k-1})$, represented by the polynomial

$$m(x) = m_0 + m_1 x + m_2 x^2 + \ldots + m_{k-1} x^{k-1}$$

where the m_i are elements of $\mathbb{F}_q$ and $k = q - 2t - 1$.

Output: The codeword formed by the coefficients of $c(x) = m(x)g(x)$, to be transmitted.

The polynomial $g(x)$ is referred to as the generator polynomial.

Decoding requires some sophistication, following [3]. Suppose we receive a message $r(x) = c(x) + e(x)$, where $e(x)$ is the error polynomial accumulated during transmission. We can evaluate $r(x)$ at the α^j. If we do this, we have

$$r(\alpha^j) = c(\alpha^j) + e(\alpha^j) = g(\alpha^j)m(\alpha^j) + e(\alpha^j) = e(\alpha^j)$$

where the last equality holds because the roots of $g(x)$ were α^j by design. We call these values the message syndromes and denote them by $S_j = r(\alpha^j) = e(\alpha^j)$. If all of the $S_j = 0$, then there was no transmission error. Otherwise, we have to use the values of S_j to determine error locations and their values. If there are $0 \leq \nu \leq t$ errors occuring in unknown locations $i_1, i_2, \ldots, i_\nu$, we can write the error polynomial as

$$e(x) = e_{i_1} x^{i_1} + e_{i_2} x^{i_2} + \ldots + e_{i_\nu} x^{i_\nu}$$

noting that i_ℓ, e_{i_ℓ}, and ν are all unknown. In a change of notation, let $Y_\ell = e_{i_\ell}$ and $X_\ell = \alpha^{i_\ell}$. The syndromes are then written

$$S_j = Y_1 X_1^j + Y_2 X_2^j + \ldots + Y_\nu X_\nu^j = \sum_{\ell=1}^{\nu} Y_\ell X_\ell^j$$

for $1 \leq j \leq 2t$. This gives a system of $2t$ equations in ν unknowns X_ℓ and ν unknowns Y_ℓ, where ν itself is unknown. It can be shown that a solution always exists and is unique. Solving this system is aided by defining the so-called error locator polynomial

$$\Lambda(x) = \prod_{\ell=1}^{\nu} (1 - xX_\ell) = 1 + \Lambda_1 x + \Lambda_2 x^2 + \ldots + \Lambda_t x^\nu$$

whose inverted roots give us information about the error positions. The error locator polynomial can be related to the syndromes by first setting $x = X_\ell^{-1}$ and multiplying both sides by $Y_\ell X_\ell^{j+\nu}$, for each ℓ and j. This gives

$$0 = Y_\ell X_\ell^{j+\nu} \Lambda(X_\ell^{-1}) = Y_\ell (X_\ell^{j+\nu} + \Lambda_1 X_\ell^{j+\nu-1} + \ldots + \Lambda_\nu X_\ell^j)$$

If we sum all of these equations over $1 \leq \ell \leq \nu$,

$$\sum_{\ell=1}^{\nu} Y_\ell X_\ell^{j+\nu} + \Lambda_1 \sum_{\ell=1}^{\nu} Y_\ell X_\ell^{j+\nu-1} + \ldots + \Lambda_\nu \sum_{\ell=1}^{\nu} Y_\ell X_\ell^j = 0$$

Observe that each one of these sums corresponds to a message syndrome. Making the substitution produces the system of equations

$$\Lambda_1 S_{j+\nu-1} + \Lambda_2 S_{j+\nu-2} + \ldots + \Lambda_\nu S_j = -S_{j+\nu}$$

over $1 \leq j \leq \nu$, or equivalently,

$$\begin{pmatrix} S_1 & S_2 & \cdots & S_\nu \\ S_2 & S_3 & \cdots & S_{\nu+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_\nu & S_{\nu+1} & \cdots & S_{2\nu-1} \end{pmatrix} \begin{pmatrix} \Lambda_\nu \\ \Lambda_{\nu-1} \\ \vdots \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} -S_{\nu+1} \\ -S_{\nu+2} \\ \vdots \\ -S_{2\nu} \end{pmatrix}$$

This system can be solved efficiently using many famous algorithms, such as the one by Berelekamp-Massey, described in [2, 3, 21].

Procedure (Reed-Solomon Decoding, BCH Formulation)**.**

Input: A received message $r = (r_0, r_1, r_2, \ldots, r_{n-1})$, represented by

$$r(x) = r_0 + r_1 x + r_2 x^2 + \ldots + r_{n-1} x^{n-1}$$

Computation:

1. Calculate the syndromes

$$S_j = r(\alpha^j) = \sum_{\ell=1}^{\nu} e_{i_\ell} (\alpha^j)^{i_\ell}$$

for $j = 1, 2 \ldots, n-k$, where e_{i_ℓ} is the error value in the i_ℓ-th component of r, and ν is the number of errors.

2. Find the error positions α^{i_ℓ} and error values e_{i_ℓ} by finding the error locator polynomial and its roots, and make the appropriate corrections in $r(x)$.

Output: The polynomial $r(x)/g(x)$.

1.2.2 Code Properties

Reed-Solomon codes have many excellent properties.

Code	$(255, 223)$ Reed-Solomon	(n, k) Reed-Solomon
Message block size	223	k
Code block size	255	n
Minimum distance	33	$n - k + 1$
Correctable errors	16	$\lfloor (n - k + 1)/2 \rfloor$
Information rate	223/255	n/k

Table 1.3: Properties of Reed-Solomon codes

In 1964, R.C. Singleton published a very basic bound relating to the minimum distance of block codes.

Theorem 1.1 (Singleton)**.** Given a q-symbol (n, k) error-correcting code with minimum distance d, we must have

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

Proof. Consider only the first $k-1$ coordinates of our codewords, noting that there are only q^{k-1} possibilities. Since we have q^k codewords, there must be a collision. Therefore, any two codewords can differ in at most $n - (k - 1) = n - k + 1$ of the remaining coordinates. This is exactly $d \leq n - k + 1$. □

From the table above, we see that Reed-Solomon codes match the Singleton bound. We call this a maximum-distance separable (MDS) code.

Reed-Solomon codes are highly resistant against burst errors. This can be seen in our small example, where two bits were flipped in the transmission.

Transmitted	10011100
Received	10101100

Table 1.4: Comparison between transmitted and received words

Since the two flipped bits were part of the same symbol, this only counts as a single error. This effect is more dramatic when there are more bits per symbol (say, in a code over $\mathbb{F}_{2^8}$).

1.2.3 Applications in Technology

Compact Discs

- CDs make use of two concatenated cross-interleaved Reed-Solomon codes (CIRC), the first is a $(32, 28)$ code C_1 and the second is a $(28, 24)$ code C_2, both over $\mathbb{F}_{2^8}$.
- Interleaving of data symbols prevents burst errors from overwhelming any one block.
- If the C_1 or C_2 decoders fail to correct the errors in a block, the symbols in the block are flagged as erasures so that the hardware can attempt to conceal the corruption.
- The CIRC setup can correct a burst error lasting about 4000 bits, or 2.5mm in track length.
- If the errors are too overwhelming, data blocks can be interpolated or concealed for errors spanning around 12300 bits, or 7.7mm in track length.

QR Codes

- QR codes were invented by the Toyota subsidiary Denso Wave in 1994 and are freely available for use.
- A 21×21 QR code ('Version 1') contains 26 bytes of information.
- There are four levels of error-correction:

Level	Check Symbols	Data Bytes
L	7	19
M	10	16
Q	13	13
H	17	9

Table 1.5: Levels of error-correction in a 21×21 QR code

where the coding is done over $\mathbb{F}_{2^8}$.

- A 21×21 QR code with level M encoding uses the generator polynomial

$$g(x) = (x - \alpha)(x - \alpha^2) \cdots (x - \alpha^{10})$$

where α is a multiplicative generator of $\mathbb{F}_{2^8}^*$. This allows for the correction of up to five errors.

- Error correction in QR codes allows them to be read even if they are damaged or obstructed. Some use this property for artistic purposes by embedding images within a standard code.

Voyager Probes

- Reed-Solomon codes were first used for space exploration in the Voyager missions (1977). They were to be used in the transmission of full-colour 800×800 images at 8 bits per pixel.

- Compressing the colour images made them vulnerable to bit errors, so engineers employed a Reed-Solomon code composed with a convolutional code to compensate. The system, however, was considered too new and experimental, so it was used as a backup system for a more traditional setup. It was finally put into action after the basic Jupiter and Saturn mission.

- Voyager uses a $(255, 223)$ code over $\mathbb{F}_{2^8}$, which is represented by the primitive polynomial $f(x) = x^8 + x^4 + x^3 + x^2 + 1$. The generator polynomial is

$$g(x) = (x - \alpha)(x - \alpha^2) \cdots (x - \alpha^{32})$$

where α is a root of $f(x)$. This allows for the correction of up to 16 symbol errors.

1.3 Further Improving Error-Correction Limits

1.3.1 List Decoding

Can the error-correcting limit in Reed-Solomon codes be improved? We know that a received word can be decoded to a unique codeword as long as the number of errors is less than $d/2$, where d is the minimum distance. If we want to handle more errors, we have to drop unique decoding. This leads to the following concept:

Problem (List Decoding)**.** Given a received word r and an error tolerance t, to find all codewords c such that $d(c, r) \leq t$.

For Reed-Solomon codes, it was unknown whether or not this could be done efficiently, even for t a little larger than $d/2$. In 2001, Guruswami and Sudan published a random polynomial time algorithm that allowed decoding in the presence of up to $n - \sqrt{nk}$ errors [12, 13].

Procedure (Guruswami-Sudan List Decoding)**.**

Input: A received message $r = (r_1, r_2, \ldots, r_n)$ and an error tolerance t.

Computation:

1. Find a two-variable polynomial $Q(x, y)$ with suitably high degree so that $Q(x_i, r_i) = 0$ for $1 \leq i \leq n$.
2. Find all factors of $Q(x, y)$ of the form $y - p(x)$, where $\deg p(x) < k$.

Output: Those polynomials $y = p(x)$ such that $r_i = p(x_i)$ for at least $n - t$ of the (x_i, r_i).

A more detailed description and implementation of the algorithm can be found in [12].

1.3.2 New Problems

Given the breakthroughs given in the Guruswami-Sudan algorithm, we can formulate two new questions to further investigate decoding limitations.

Definition (Distance to a Code)**.** For an error-correcting code $\mathcal{C}$ and a word r, denote $d(r, \mathcal{C})$ to be the shortest distance between r and each codeword from $\mathcal{C}$.

Problem (Maximum Likelihood Decoding)**.** Given a received word r, to find an explict codeword c such that $d(r, c) = d(r, \mathcal{C})$.

Problem (Bounded Distance Decoding)**.** Given a received word r and a bound B, to find just one codeword c such that $d(r, c) \leq B$.

1.3.3 Hardness Results

Studying these problems has led to several hardness results. One major result comes from Guruswami and Vardy [14].

Theorem 1.2 (Guruswami-Vardy, 2005)**.** For an integer $t \geq 1$, let $m = 3t$, $k = t^3 - t - 1$, and $n = t^3$. There is a class of (n, k) Reed-Solomon codes over $\mathbb{F}_{2^m}$ with evaluation set of size $|D| = n$ such that maximum-likelihood decoding is NP-complete.

The proof gives a polynomial-time reduction of the maximum-likelihood decoding problem to the three-dimenisonal matching problem, which has been shown to be NP-complete. The main drawback is that this code uses a tremendously small evaluation set D compared to $\mathbb{F}_q$ (i.e. t^3 versus 2^{3t}), so in some sense, it isn't realistic for codes actually used in practise. Guruswami and Vardy noted this and suggested in passing that the maximum-likelihood decoding might be easier if D is much larger or has some algebraic structure.

Cheng and Wan also published hardness results on two occassions for codes with large evaluation sets, both relying on the assumed hardness of the discrete logarithm problem over finite fields [7, 8]. Their arguments hinged on giving a procedure to interpret a decoding problem in terms of a discrete logarithm problem (similar to using an index calculus algorithm).

Problem (Discrete Logarithm)**.** Given a finite field $\mathbb{F}_q$, a generator g of $\mathbb{F}_q^*$, and a nonzero element a, to find an integer i such that

$$g^i = a$$

The value of i is denoted $\log_g a$.

Theorem 1.3 (Cheng-Wan, 2004)**.** In an (n, k) Reed-Solomon code over $\mathbb{F}_q$, let $\hat{g}(n, k, q)$ be the smallest positive integer g such that $\binom{n}{g}/q^{g-k}$ is less than 1. If there exists an algorithm solving the list decoding problem of radius $n - \hat{g}$ in time $q^{O(1)}$, then discrete logarithm over the finite field $\mathbb{F}_{q^{\hat{g}-k}}$ can be computed in random time $q^{O(1)}$.

Theorem 1.4 (Cheng-Wan, 2004)**.** Let h be a positive integer satisfying

$$q \geq \max(g^2, (h-1)^{2+\varepsilon}) \quad \text{and} \quad g \geq (4/\varepsilon + 2)(h+1)$$

for a constant $\varepsilon > 0$. If the bounded distance decoding problem of radius $B = q - g$ for the $(q, g - h)$ Reed-Solomon code can be solved in time $q^{O(1)}$, the discrete logarithm problem over $\mathbb{F}_{q^h}$ can be solved in random time $q^{O(1)}$.

Theorem 1.5 (Cheng-Wan, 2010)**.** Let $\delta > 0$ be a constant and $m > 1$ be an integer. Suppose h and k are integers satisfying

$$h \leq \frac{q^{\frac{1}{2+\delta}}}{m} + \frac{1}{m}\ ,\ h \leq \frac{\sqrt{q}}{m(4/\delta + 2)} - \frac{1}{m}\ ,\ q \leq k \leq q^m - q$$

The discrete logarithm in $\mathbb{F}_{q^{mh}}^*$ can be solved in randomized time $(q^m)^{O(1)}$ with oracle access to a maximum-likelihood decoder for a (q^m, k) Reed-Solomon code over $\mathbb{F}_{q^m}$.

Theorem 1.6 (Cheng-Wan, 2010)**.** Let ε be a positive constant less than $1/3$ and $g = \frac{2+3\varepsilon}{1-3\varepsilon}(h+1)$. In an $(q, q-g-h)$ Reed-Solomon code over $\mathbb{F}_q$ for sufficiently large q, there does not exist a randomized polynomial time bounded distance decoder at distance $(2/3 + \varepsilon)d$, where d is the minimum distance, unless the discrete logarithm problem over $\mathbb{F}_{q^h}$ can be solved in randomized time $q^{O(1)}$ for any $h \leq q^{0.8\varepsilon}$.

Later, in 2012, Augot and Morain took Cheng and Wan's conversion idea and made it effective [1]. This allowed them to produce a new algorithm for computing discrete logarithms.

Theorem 1.7 (Augot-Morain, 2012)**.** Let $F = \mathbb{F}_{q^h}$ and $K = \mathbb{F}_q$. Take a fixed monic $Q(X)$ from $K[X]$, with $\deg Q(X) = h$, and a set $S \subset F$ with size n so that $Q(a) \neq 0$ for all $a \in S$. Let $1 \leq \mu \leq n$. For any $f(X)$ in $K[X]$ with $\deg f(X) < \mu$, there exists $A \subset S$ where $|A| = \mu$ such that

$$\prod_{a \in A}(X - a) \equiv f(X) \pmod{Q(x)}$$

if and only if the word represented by the polynomial

$$y(X) = -f(X)/Q(X) - X^k$$

is exactly distance $n - \mu$ from the Reed-Solomon code with $k = \mu - h$ and evaluation set $D = S$. All such sets A can be found by decoding the word $y(X)$ up to radius $n - \mu$.

Using this conversion, discrete logarithms can be computed using a procedure similar to index calculus. If $Q(X)$ is a primitive polynomial, take $f(X) = X^u$ for random u. After finding all relations of the form

$$\prod_{a \in A}(X - a) \equiv f(X) \pmod{Q(x)}$$

using Reed-Solomon decoding, we can try to set up a linear system to solve for the values of $\log(X - a)$ for all $a \in S$. The authors showed that their implementation of this procedure requires $\tilde{O}(h!\, q^2)$ operations over $\mathbb{F}_q$.

1.4 The Deep Hole Problem

In an (n, k) Reed-Solomon code, it can be shown that for any received word r, we always have $d(r, \mathcal{C}) \leq n - k$. This bound is the so-called covering radius, the maximum value of $d(r, \mathcal{C})$ over all possible words r. For r satisfying $d(r, \mathcal{C}) = n - k$, Guruswami and Vardy called this a deep hole. They observed that as a consequence of their results in their family of codes, determining whether or not r is a deep hole is actually NP-hard. As before, the situation might be different for codes with evaluation sets of larger size or some algebraic structure. This leads us to the following problem:

Problem (Determining Deep Holes)**.** Take an (n, k) Reed-Solomon code over $\mathbb{F}_q$ with some evaluation set D, preferably $D = \mathbb{F}_q$, $D = \mathbb{F}_q^*$, or some large or structured set. Given a received message r, to determine whether or not r is a deep hole.

One way to measure $d(u, \mathcal{C})$ is to run Lagrange Interpolation on the word $u = (u_1, \cdots, u_n)$ to get a fitted polynomial $u(x)$ satisfying $u(x_i) = u_i$ for all $1 \leq i \leq n$. Then, if $\deg u(x) \leq k-1$, then u is a codeword and $d(u, \mathcal{C}) = 0$. Otherwise, $k \leq \deg u(x) \leq n-1$, and Li and Wan in [16] gave the bound

Theorem 1.8. If $\deg u(x) \geq k$, then

$$n - \deg u(x) \leq d(u, \mathcal{C}) \leq n - k$$

Proof. To prove the right-hand inequality, let $\{x_1, x_2, \ldots, x_k\}$ a set of any k points from D. Let $g(x) = \prod_{i=1}^{k}(x - x_i)$. By the division algorithm, we can write

$$u(x) = g(x)q(x) + v(x)$$

where $v(x)$ is a codeword, since $\deg v(x) \leq k-1$. Then, $u(x) - v(x) = g(x)q(x)$ has at least k roots by the design of $g(x)$, meaning that u and v have at least k coordinates in common. Therefore, u differs from a codeword in no more than $n-k$ coordinates, or $d(u, \mathcal{C}) \leq n-k$.

Let $N(\cdot)$ denote the number of zeros of a polynomial. To prove the left-hand inequality, we measure the error distance:

$$\begin{aligned} d(u, \mathcal{C}) &= \min_{v \in \mathcal{C}} d(u, v) \\ &= n - \max_{v(x)} N(u(x) - v(x)) \\ &\geq n - \deg u(x) \end{aligned}$$

The last inequality holds because $u(x) - v(x)$ is a polynomial of degree $\deg u(x)$, so it has at most $\deg u(x)$ roots. This completes the proof. □

This simple bound shows that if $\deg u(x) = k$, then u is automatically a deep hole. Many of the results toward the deep hole problem are geared toward examining families of words by degree.

1.4.1 Results when $D = \mathbb{F}_q$

Cheng and Murray in [6] searched for deep holes when $D = \mathbb{F}_q$ (referred to as a standard Reed-Solomon code), and conjectured that the only deep holes were those satisfying $\deg u(x) = k$. More precisely,

Conjecture (Cheng-Murray)**.** All deep holes for standard Reed-Solomon codes are those words u satisfying $\deg u(x) = k$.

They weren't able to prove this, but they were able to reduce the problem to finding a rational point on an algebraic hypersurface to derive the first result on deep holes for Reed-Solomon codes over the prime field $\mathbb{F}_p$:

Theorem 1.9 (Cheng-Murray, 2007)**.** Let p be a prime and $1 < k < p^{1/4-\varepsilon}$ be a positive integer. Let u be a received word and $u(x)$ be its interpolated polynomial. If the degree of $u(x)$ satisfies

$$k < \deg u(x) < k + p^{3/13-\varepsilon}$$

then u is not a deep hole.

Example. Choose $p = 929$. Then in the $(929, k)$ Reed-Solomon code (for $1 < k < 5.25$), if the degree of $u(x)$ satisfies

$$k < \deg u(x) < k + 4.84$$

then u is not a deep hole.

In one of the newest papers, Cheng, Li, and Zhuang [4] were able to resolve in some cases the conjecture over $\mathbb{F}_p$ using the concept of deep hole trees.

Definition (Equivalent Functions)**.** Let $f(x)$ and $g(x)$ be functions over a finite field $\mathbb{F}_q$, and take an (n, k) Reed-Solomon code. f and g are said to be equivalent if there exists $a \in \mathbb{F}_q^*$ and a polynomial $h(x)$ with degree less than k such that

$$f(x) = ag(x) + h(x)$$

We refer to the set of functions equivalent to f as a class.

Theorem 1.10 (Cheng-Li-Zhuang, 2013)**.** Let $p > 2$ be a prime number, $k \geq \frac{p-1}{2}$, $D = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ with $k < n \leq p$. The only deep holes of $\mathcal{C}_p$ are generated by functions which are equivalent to the following:

$$f(x) = x^k, \qquad f_\delta(x) = \frac{1}{x - \delta}$$

where $\delta \in \mathbb{F}_p \backslash D$.

Over $\mathbb{F}_q$, they also showed

Theorem 1.11. Given a finite field $\mathbb{F}_q$ with characteristic $p > 2$, if $k + 1 \leq p$ or $3 \leq q - p + 1 \leq k + 1 \leq q - 2$, then the Cheng-Murray conjecture is true.

There is further evidence in favour of the conjecture. Li and Wan [16] studied the problem in terms of solving polynomial congruences, using character sums combined with Weil's character sum bounds to count solutions. They were able to give some exact distance measurements under the right conditions.

Theorem 1.12 (Li-Wan, 2010)**.** Let u be a received word and $u(x)$ be its interpolated polynomial. Suppose $1 \leq d := \deg u(x) - k \leq q - 1 - k$. If

$$q > \max((k+1)^2, d^{2+\varepsilon}) \text{ and } k > \left(\frac{2}{\varepsilon} + 1\right) d + \frac{8}{\varepsilon} + 2$$

for some constant $\varepsilon > 0$, then $d(u, \mathcal{C}) < q - k$. In other words, u is not a deep hole. Furthermore, if

$$q > \max((k+1)^2, (d-1)^{2+\varepsilon}) \text{ and } k > \left(\frac{4}{\varepsilon} + 1\right) d + \frac{4}{\varepsilon} + 2$$

for some constant $\varepsilon > 0$, then $d(u, \mathcal{C}) = q - (k + d)$.

Several other authors followed these techniques to get new bounds.

Theorem 1.13 (Liao, 2011 [20])**.** Let $r \geq 1$ be an integer. Let u be a received word and $u(x)$ be the interpolated polynomial of degree m. If $m \geq k + r$,

$$q > \max\left\{2\binom{k+r}{2} + (m-k), (m-k)^{2+\varepsilon}\right\}$$

and

$$k > \frac{1}{1+\varepsilon}\left(r + (2+\varepsilon)\left(\frac{m}{2}+1\right)\right)$$

for some constant $\varepsilon > 0$, then $d(u, \mathcal{C}) \leq q - k - r$.

Using some techniques from algebraic geometry, Cafure, Matera, and Privitelli in [5] slightly improved on one of Li-Wan's previous results with

Theorem 1.14 (Cafure-Matera-Privitelli, 2012)**.** Let u be a received word and $u(x)$ be is interpolated polynomial with $1 \leq d := \deg(u(x)) - k \leq q - 1 - k$. Assume that

$$q > \max((k+1)^2, 14d^{2+\varepsilon}) \text{ and } k > d\left(\frac{2}{\varepsilon}+1\right)$$

for some constant $\varepsilon > 0$. Then u is not a deep hole.

In the previous results, many of the conditions required $u(x)$ to be a polynomial with degree only slightly larger than k. Zhu and Wan improved upon this by observing that some high degree polynomials can also be represented by low-degree rational functions [29]. They came up with

Theorem 1.15 (Zhu-Wan, 2012)**.** Let $r \geq 1$ be an integer. Suppose we can write

$$\left(\frac{w(x_1)}{h(x_1)}, \frac{w(x_2)}{h(x_2)}, \ldots, \frac{w(x_q)}{h(x_q)}\right) = u$$

for some $h(x) \in \mathbb{F}_q[x]$, with $\gcd(h(x), x^q - x) = 1$, and $\deg h(x) + k \leq \deg w(x) \leq q - 1$. Let m be the smallest such degree of $w(x)$, and set $r \leq d := m - k \leq q - 1 - k$. There are positive constants c_1 and c_2 such that if

$$d < c_1 q^{1/2}\ ,\ \left(\frac{d+r}{2}+1\right)\log_2 q < k < c_2 q$$

then $d(u, \mathcal{C}) \leq q - k - r$.

Li and Zhu in [19] recently related the deep hole problem to explicitly counting solutions to certain polynomial equations to exactly calculate error distances.

Theorem 1.16 (Li-Zhu, 2015)**.** Let u be a received word represented by the class $u(x) = x^{k+1} - bx^k$. Then

$$d(u, \mathcal{C}) = \begin{cases} q-k & b = 0, p = 2, k = 1 \text{ or } q-3 \\ q-k-1 & \text{otherwise} \end{cases}$$

They also proved a similar distance formula for words in the class $u(x) = x^{k+2} - bx^{k+1} + cx^k$, showing that such words are not deep holes either.

1.4.2 Results when $D = \mathbb{F}_q^*$

Wu and Hong studied the deep hole problem when $D = \mathbb{F}_q^*$ (referred to as a primitive Reed-Solomon code) [27]. Using this evaluation set allowed them to use the BCH formulation of Reed-Solomon codes.

Theorem 1.17 (Wu-Hong, 2011)**.** Take a Reed-Solomon code over $\mathbb{F}_q$ with $q \geq 4$ and $2 \leq k \leq q-2$. Then polynomials in the class $u(x) = x^{q-2}$, represent deep holes.

This new family of deep holes shows that the Cheng-Murray conjecture is false when $D = \mathbb{F}_q^*$. They gave a revision to the conjecture:

Conjecture (Wu-Hong)**.** All deep holes for primitive Reed-Solomon codes are those words u represented by the class $u(x) = x^k$ or x^{q-2}.

Cheng, Li, and Zhuang in the previously referenced [4] were able to resolve this under some conditions.

Theorem 1.18 (Cheng-Li-Zhuang, 2013)**.** Given a finite field $\mathbb{F}_q$ with characteristic $p > 2$, if $3 \leq k < \frac{\sqrt{q}+1}{4}$ or $3 \leq k < \frac{p}{45}$ when $q = p$ is prime, then the Wu-Hong conjecture is true.

Zhang, Fu, and Liao proved an extension of Wu-Hong that allows D to be any evaluation set except $\mathbb{F}_q$ [28]. They also adapted work from Li-Wan to find more deep holes for a specific message length k. Finally, they found a class of received words that are not deep holes.

Theorem 1.19 (Zhang-Fu-Liao, 2012)**.** Take a Reed-Solomon code over $\mathbb{F}_q$ with evaluation set $D \neq \mathbb{F}_q$. Then for any $a \neq 0$, $b \notin D$, polynomials in the class

$$u(x) = (x - b)^{q-2}$$

represent deep holes.

Theorem 1.20 (Zhang-Fu-Liao, 2012)**.** Take a Reed-Solomon code over $\mathbb{F}_q$ for $q > 5$, $2 \leq k \leq q-3$, and $D = \mathbb{F}_q^*$. Polynomials in the class

$$u(x) = ax^{k+2} + bx^{k+1} + cx^k$$

where $a \in \mathbb{F}_q^*$, $b, c \in \mathbb{F}_q$, do not represent deep holes.

Theorem 1.21 (Zhang-Fu-Liao, 2012)**.** Let $q > 4$ be a power of 2 and take a Reed-Solomon code over $\mathbb{F}_q$ with evaluation set $D = \mathbb{F}_q^*$ or $D = \mathbb{F}_q^*/\{1\}$ and $k = q-4$. If $a \neq 0$, then polynomials in the class

$$u(x) = x^{q-3}$$

represent deep holes.

This last result shows that the deep hole problem in characteristic two, the most important setting for applications, may be very complicated.

Li and Zhu from [19] also exactly calculated error distances for polynomials of degree $k+1$ and $k+2$ in this setting. One sample from their work:

Theorem 1.22 (Li-Zhu, 2015)**.** Take a Reed-Solomon code over $\mathbb{F}_q$ with evaluation set $D = \mathbb{F}_q^*$, and let u be a received word represented by the class $u(x) = x^{k+2} - bx^{k+1} + cx^k$. Then

$$d(u, \mathcal{C}) = \begin{cases} q-k-2 & b^2 = c \\ q-k-1 & b^2 \neq c \end{cases}$$

Chapter 2

A Schematic Approach to the Deep Hole Problem

Our first result is joint with Zhu and Wan, following [29]. We take a schematic approach to the deep hole in generalised Reed-Solomon codes. We can find a class of words that are not deep holes as long as a character sum over the evaluation set D can be estimated. An important special case will be derived by choosing D to be a subgroup of $\mathbb{F}_q^*$, which can be a substantially small portion of $\mathbb{F}_q$.

2.1 Main Theorem

Fix an enumeration $D = \{x_1, x_2, \ldots, x_{|D|}\}$. All of our specific results will hinge on

Theorem 2.1. Let $\mathcal{C}$ be the generalised Reed-Solomon code over $\mathbb{F}_q$ using the evaluation set D. Let u be a received word. Suppose we can write

$$\left(\frac{w(x_1)}{h(x_1)}, \frac{w(x_2)}{h(x_2)}, \ldots, \frac{w(x_{|D|})}{h(x_{|D|})}\right) = u$$

for some $h(x) \in \mathbb{F}_q[x]$, with no roots in $D \cup 0$, and $\deg h(x) + k \leq \deg w(x) \leq |D| - 1$. Let m be the smallest such degree of $w(x)$. Let $1 \leq r \leq d := m - k \leq |D| - k - 1$. If the bound

$$\left|\sum_{a \in D} \chi(1 - ax)\right| \leq K q^{1/2}$$

is true over all nontrivial characters $\chi : (\mathbb{F}[x]/(\bar{h}(x)))^* \to \mathbb{C}^*$ with $\chi(\mathbb{F}_q^*) = 1$ for some $K \geq d$ and $\bar{h}(x) = x^{m-k+1}h(1/x)$, there are positive constants c_1 and c_2 such that if

$$d \leq K < c_1 \frac{|D|}{q^{1/2}}\ ,\ \left(\frac{d+r}{2} + 1\right) \log_2 q < k < c_2 |D|$$

then $d(u, \mathcal{C}) \leq |D| - k - r$.

This statement is quite robust in that it can handle words that can either be represented by low-degree polynomials or low-degree rational functions. (The next section will show some examples.) In addition, it applies to code families with a positive information rate $k/|D|$. Setting $h(x) = 1$, the theorem reduces to the usual polynomial case, and we receive

Corollary 2.1. Let $\mathcal{C}$ be the generalised Reed-Solomon code over $\mathbb{F}_q$ using the evaluation set D. Let $r \geq 1$ be an integer and u a received word with interpolated polynomial $u(x)$

such that $r \leq d := \deg(u(x)) - k \leq |D| - k - 1$. If the bound

$$\left| \sum_{a \in D} \chi(1 - ax) \right| \leq K q^{1/2}$$

is true over all nontrivial characters $\chi : (\mathbb{F}[x]/(x^{d+1}))^* \to \mathbb{C}^*$ with $\chi(\mathbb{F}_q^*) = 1$ for some $K \geq d$, then there are positive constants c_1 and c_2 such that if

$$d \leq K < c_1 \frac{|D|}{q^{1/2}} \ , \ \left(\frac{d+r}{2} + 1 \right) \log_2 q < k < c_2 |D|$$

then $d(u, \mathcal{C}) \leq |D| - k - r$.

We can specialise this result to a code that uses the evaluation set $D = (\mathbb{F}_q^*)^{\frac{q-1}{\ell}}$ or $D = \mathbb{F}_q$. For these cases, we will be able to take $K = d$.

Corollary 2.2. Let $\mathcal{C}$ be the generalised Reed-Solomon code over $\mathbb{F}_q$ using the evaluation set $D = (\mathbb{F}_q^*)^{\frac{q-1}{\ell}}$ of size ℓ. Let $r \geq 1$ be an integer and u a received word with interpolated polynomial $u(x)$ such that $r \leq d := \deg(u(x)) - k \leq q - 2 - k$. There are positive constants c_1 and c_2 such that if

$$d < c_1 \frac{\ell}{q^{1/2}} \ , \ \left(\frac{d+r}{2} + 1 \right) \log_2 q < k < c_2 \ell$$

then $d(u, \mathcal{C}) \leq \ell - k - r$.

Note that the first condition $d < c_1 \frac{\ell}{q^{1/2}}$ will implicitly put a minimum on the size of ℓ due to the dependency between c_1 and c_2. This dependency can be seen in the upcoming example.

Corollary 2.3. Let $\mathcal{C}$ be the standard Reed-Solomon code over $\mathbb{F}_q$ using the evaluation set $D = \mathbb{F}_q$. Let $r \geq 1$ be an integer and u a received word with interpolated polynomial $u(x)$ such that $r \leq d := \deg(u(x)) - k \leq q - k - 1$. There are positive constants c_1 and c_2 such that if

$$d < c_1 q^{1/2} \ , \ \left(\frac{d+r}{2} + 1 \right) \log_2 q < k < c_2 q$$

then $d(u, \mathcal{C}) \leq q - k - r$.

This corollary recovers what Zhu and Wan proved in [29].

2.2 Examples

To get a better idea of what these theorems mean, take $\mathcal{C}$ to be the primitive Reed-Solomon code over $\mathbb{F}_{2^8}$ (i.e. $D = \mathbb{F}_{2^8}^*$). We will realise this field as $\mathbb{F}_{2^8} = \mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x^2 + 1)$, noting that this is formed using a primitive polynomial.

Now if $r \geq 1$, and u is a codeword with $r \leq d \leq 254 - k$, then we can find c_1 and c_2 such that if

$$d < \frac{255}{16} c_1 \quad \text{and} \quad 8\left(\frac{d+r}{2} + 1\right) < k < 255 c_2$$

then $d(u, \mathcal{C}) \leq 255 - k - r$. To be even more concrete, consider $d = r = 1$; in other words, we want to classify codewords whose polynomial (or rational) interpolations are degree $k+1$ (in the numerator). From the proof below, we can explicitly compute c_1 and c_2 using the formulas

$$1 < \frac{255}{16} c_1 \ , \ c_1 + c_2 = 256^{-\frac{1}{k+1}} - \frac{1}{2}$$

To obtain a wide range of k, fix $c_1 = .0628$. Then we have the condition

$$16 < k < 255\left(256^{-\frac{1}{k+1}} - \frac{1}{2} - .0628\right)$$

A computer algebra system shows that this is satisfied when $17 \leq k \leq 97$. Therefore, for codes using this range of message lengths, received words u represented by a polynomial (or rational function) of degree $k + 1$ (in the numerator) are not deep holes. More specifically, we can give the estimate $d(u, C) \leq 254 - k$.

Along the same lines, for $r = 1$ and $d = 2$, polynomials (or rational functions) of degree $k + 2$ (in the numerator) do not represent deep holes when the message length satisfies $21 \leq k \leq 86$. Again, such words u satisfy the estimate $d(u, C) \leq 254 - k$. Attempting to increase r or d any more does not yield additional information.

Here is a table with a few examples of words covered by our bounds. We will denote α to be a root of $x^8 + x^4 + x^3 + x^2 + 1$, so α will be a multiplicative generator for $\mathbb{F}_{2^8}^*$.

d	k	Polynomial Interpolation	Rational Interpolation
1	17	$x^{18} + 3x^2 + 1$	N/A
1	97	$x^{98} + x^{24} + x^{17} + 1$	N/A
2	30	$(\alpha^6 + \alpha^3 + 1)x^{254} + \ldots + (\alpha^6 + \alpha^5)$	$x^{32}/(x^2 + \alpha x + \alpha^7)$
2	86	$(\alpha^6 + \alpha^5)x^{254} + \ldots + (\alpha^7 + \alpha^6 + \alpha^2)$	$(x^{88} + 1)/(x^2 + x + \alpha^5)$

Table 2.1: Polynomial interpolation vs. rational interpolation

2.3 Preliminaries

There are a few theorems that we will need to establish our results.

2.3.1 Weil's Character Sum Bound

Definition (Multiplicative Character)**.** Let $h(x)$ be a polynomial from $\mathbb{F}_q[x]$. We say that a homomorphism $\chi : (\mathbb{F}_q[x]/(h(x)))^* \to \mathbb{C}^*$ is a multiplicative character of $(\mathbb{F}_q[x]/(h(x)))^*$. This can be extended to the entire group $\mathbb{F}_q[x]/(h(x))$ by setting $\chi(f(x)) = 0$ when $\gcd(f(x), h(x)) \neq 1$.

We will take advantage of the Weil bounds to estimate the number of solutions to certain polynomial equations. Let Λ denote the polynomial von Mangoldt function, defined as

$$\Lambda(f) = \begin{cases} \deg g & \text{if } f = g^k \text{ for some irreducible } g \\ 0 & \text{otherwise} \end{cases}$$

As stated in [24]:

Theorem 2.2 (Weil)**.** Let $h(x)$ be a polynomial of positive degree in the ring $\mathbb{F}_q[x]$, and let $\chi : (\mathbb{F}[x]/(h(x)))^* \to \mathbb{C}^*$ be a multiplicative character. If χ is not trivial, then

$$\left| \sum_{a \in \mathbb{F}_q} \chi(x - a) \right| \leq (\deg h(x) - 1) q^{1/2}$$

In general, for a sum over polynomials of degree k,

$$\left| \sum_{\deg g = k} \Lambda(g)\chi(g) \right| \leq (\deg h(x) - 1) q^{k/2}$$

Furthermore, if χ is not trivial but $\chi(\mathbb{F}_q^*) = 1$, then

$$\left| 1 + \sum_{a \in \mathbb{F}_q} \chi(x-a) \right| \leq (\deg h(x) - 2) q^{1/2}$$

Since we will be dealing with $(\mathbb{F}_q^*)^{(q-1)/\ell}$, these bounds need to be slightly modified:

Lemma 2.1. Let $h_1(x)$ be a polynomial from $\mathbb{F}_q[x]$ not divisible by x, $h(x) = x^k h_1(x)$ for $k \geq 1$, and $\chi : (\mathbb{F}_q[x]/(h_1(x)))^* \to \mathbb{C}^*$ with χ nontrivial and $\chi(\mathbb{F}_q^*) = 1$. For a subgroup $(\mathbb{F}_q^*)^{\frac{q-1}{\ell}}$ of $\mathbb{F}_q^*$, we have

$$\left| \sum_{a \in (\mathbb{F}_q^*)^{\frac{q-1}{\ell}}} \chi(x-a) \right| \leq (\deg h(x) - 1) q^{1/2}$$

Proof. Use the character sum

$$\sum_{a \in (\mathbb{F}_q^*)^{\frac{q-1}{\ell}}} \chi(x-a) = \sum_{a \in \mathbb{F}_q} \frac{\ell}{q-1} \sum_{(\chi')^{(q-1)/\ell} = 1} \chi'(a)\chi(x-a) = \frac{\ell}{q-1} \sum_{(\chi')^{(q-1)/\ell} = 1} \sum_{a \in \mathbb{F}_q} \chi'(a)\chi(x-a),$$

where $\chi' : \mathbb{F}_q^* \to \mathbb{C}^*$ denotes a multiplicative character. The two characters χ and χ' can be viewed as characters over the group

$$(\mathbb{F}_q[x]/(h(x)))^* \cong (\mathbb{F}_q[x]/(x^k))^* \times (\mathbb{F}_q[x]/(h_1(x)))^*$$

This is true because we can define the natural reduction map

$$(\mathbb{F}_q[x]/(x^k))^* \times (\mathbb{F}_q[x]/(h_1(x)))^* \to (\mathbb{F}_q[x]/(x))^* \times (\mathbb{F}_q[x]/(h_1(x)))^*$$

just by taking the terms in $(\mathbb{F}_q[x]/(x^k))^*$ modulo x. And since $\mathbb{F}_q^* \cong (\mathbb{F}_q[x]/(x))^*$, χ' lifts to a character over $(\mathbb{F}_q[x]/(x^k))^*$, which therefore extends to a character over $(\mathbb{F}_q[x]/(h(x)))^*$. Because of this, we also have $\chi'(a) = \chi'(-x+a) = \chi'(-1)\chi'(x-a)$. Then,

$$\begin{aligned}
\left| \sum_{a \in (\mathbb{F}_q^*)^{\frac{q-1}{\ell}}} \chi(x-a) \right| &\leq \frac{\ell}{q-1} \sum_{(\chi')^{(q-1)/\ell} = 1} \left| \sum_{a \in \mathbb{F}_q} \chi'(a)\chi(x-a) \right| \\
&= \frac{\ell}{q-1} \sum_{(\chi')^{(q-1)/\ell} = 1} \left| \sum_{a \in \mathbb{F}_q} (\chi'\chi)(x-a) \right| \\
&\leq \frac{\ell}{q-1} \sum_{(\chi)^{(q-1)/\ell} = 1} (\deg h(x) - 1) q^{1/2} \\
&= (\deg h(x) - 1) q^{1/2},
\end{aligned}$$

where we used the fact that the product $\chi'\chi$ is a nontrivial character of $(\mathbb{F}_q[x]/(h(x)))^*$. To see this, note that the restriction of $\chi'\chi$ to the second factor $(\mathbb{F}_q[x]/(h_1(x)))^*$ is precisely χ, which is already nontrivial. □

2.3.2 Li-Wan's New Sieve

We also state Li-Wan's new sieve (as in [17, 29]): let D be a finite set and $D^k = D \times D \times \cdots \times D$ be the Cartesian product of k copies of D. Let X be a subset of D^k. Denote

$$\bar{X} = \{(x_1, x_2, \ldots, x_k) \in X \mid x_i \neq x_j, i \neq j\}$$

Let $f(x_1, x_2, \ldots, x_k)$ be a complex-valued function defined over X. Denote

$$F = \sum_{x \in \bar{X}} f(x_1, x_2, \ldots, x_k)$$

Let S_k be the symmetric group on $\{1, 2, \ldots, k\}$. Each permutation $\tau \in S_k$ can be uniquely factorised as a product of disjoint cycles and each fixed point is viewed as a trivial cycle of length 1. Namely,

$$\tau = (i_1 i_2 \ldots i_{a_1})(j_1 j_2 \ldots j_{a_2}) \cdots (l_1 l_2 \ldots l_{a_s})$$

with $a_i \geq 1$ and $1 \leq i \leq s$. Define

$$X_\tau = \{(x_1, x_2, \ldots, x_k) \mid x_{i_1} = \ldots = x_{i_{a_1}}, x_{j_1} = \ldots = x_{j_{a_2}}, \cdots, x_{l_1} = \ldots = x_{l_{a_s}}\}$$

Similarly define

$$F_\tau = \sum_{x \in X_\tau} f(x_1, x_2, \ldots, x_k)$$

We say that τ is of the type $(c_1, c_2, \ldots, c_k)$ if it has exactly c_i cycles of length i. Let $N(c_1, c_2, \ldots, c_k)$ be the number of permutations of type $(c_1, c_2, \ldots, c_k)$. Define

$$C_k(t_1, t_2, \ldots, t_k) = \sum_{\sum i c_i = k} N(c_1, c_2, \ldots, c_k) t_1^{c_1} t_2^{c_2} \cdots t_k^{c_k}$$

Now we have the following combinatorial result:

Lemma 2.2. Suppose $q \geq d$. If $t_i = q$ for $d|i$ and $t_i = s$ for $d \nmid i$, then we have

$$C_k(s, \ldots s, q, s, \ldots, s, q, \ldots) = k! \sum_{i=0}^{\lfloor k/d \rfloor} \binom{\frac{q-s}{d} + i - 1}{i} \binom{s + k - di - 1}{k - di}$$
$$\leq \left(s + k + \frac{q-s}{d} - 1 \right)_k$$

where $(x)_k = x(x-1)(x-2)\cdots(x-k+1)$.

Furthermore, we say that X is symmetric if for any $x \in X$ and any $g \in S_k$, we have $g \circ x \in X$. Also, if a complex-valued function f is defined on X, we say that it is normal on X if X is symmetric and for any two conjugate elements in S_k, τ and τ', we have

$$\sum_{x \in X_\tau} f(x_1, x_2, \ldots, x_k) = \sum_{x \in X_{\tau'}} f(x_1, x_2, \ldots, x_k)$$

Then, we have the result:

Lemma 2.3. If f is normal on X, then

$$F = \sum_{\sum i c_i = k} (-1)^{k - \sum c_i} N(c_1, c_2, \ldots, c_k) F_\tau$$

2.3.3 A Rephrasing of Error Distance

For our purposes, it is more convenient to state the error distance in this way:

Lemma 2.4. Let $\mathcal{C}$ be the generalised Reed-Solomon code over $\mathbb{F}_q$ using the evaluation set D. Let u be a received word and $u(x)$ its interpolated polynomial with $\deg u(x) = k + d$, where $k + 1 \leq k + d \leq q - 1$. The error distance satisfies $d(u, \mathcal{C}) \leq |D| - k - r$ for some $1 \leq r \leq d$ if and only if there exists a subset $\{x_{i_1}, x_{i_2}, \ldots, x_{i_{k+r}}\} \subset D$ and a polynomial $g(x) \in \mathbb{F}_q[x]$ of degree $d - r$ such that

$$u(x) - v(x) = (x - x_{i_1})(x - x_{i_2}) \cdots (x - x_{i_{k+r}}) g(x)$$

for some $v(x)$ with $\deg v(x) \leq k - 1$.

Proof. First suppose that $d(u, \mathcal{C}) \leq |D| - k - r$. Then, the coordinates of u differ from the coordinates of some codeword v in $|D| - k - r$ places (or less). Then, their interpolated polynomials $u(x)$ and $v(x)$ have (at least) $k + r$ roots in common in D.

The converse has a very similar structure, roughly following the above in reverse. Therefore, the result follows. □

2.4 The Proof

Suppose $w(x)$ is the polynomial with degree m, and $h(x)$ is the corresponding polynomial with no roots in D. Possibly by shifting u by a constant codeword, we can assume that $w(0) \neq 0$. Also let $\bar{h}(x) = x^{m-k+1}h(1/x)$. This is a polynomial of degree $m-k+1 = d+1$ and divisible by x since $h(0) \neq 0$ and $\deg(h(x)) \leq m-k$. Let $A = (\mathbb{F}_q[x]/(\bar{h}(x)))^*$ and $\hat{A}$ denote the group of all characters of A. Let $\hat{B}$ be the set of characters χ in $\hat{A}$ with $\chi(\mathbb{F}_q^*) = 1$. Note that $\hat{B}$ is an abelian subgroup of order $\leq q^d$.

Now, by Lemma 2.4, $d(u, \mathcal{C}) \leq |D| - k - r$ if and only if there is some polynomial $f(x) \in \mathbb{F}_q[x]$ with $\deg f(x) \leq k-1$ such that

$$\frac{w(x)}{h(x)} + f(x) = \frac{w(x) + f(x)h(x)}{h(x)}$$

has at least $k+r$ distinct roots in D. In other words, there are points $\{x_1, x_2, \ldots, x_{k+r}\} \subset D$ where

$$w(x) + f(x)h(x) = (x - x_1)(x - x_2)\cdots(x - x_{k+r})v(x)$$

for some polynomial $v(x)$ with $\deg v(x) = m - (k+r)$. Then replacing x with $1/x$ and multiplying through by x^m, it is enough to find such a subset for the equation

$$\tilde{w}(x) + \tilde{f}(x)\bar{h}(x) = (1 - x_1x)(1 - x_2x)\cdots(1 - x_{k+r}x)\tilde{v}(x)$$

where $\tilde{w}(x) = x^m w(1/x)$, $\tilde{f}(x) = x^{k-1}f(1/x)$, and $\tilde{v}(x) = x^{m-(k+r)}v(1/x)$. We can now further assume that $\tilde{w}(0) = 1$ and $\tilde{v}(0) = 1$. Then this equation is equivalent to

$$\frac{(1 - x_1x)(1 - x_2x)\cdots(1 - x_{k+r}x)\tilde{v}(x)}{\tilde{w}(x)} \equiv 1 \pmod{\bar{h}(x)}$$

Let the number of solutions to this equation be denoted by N_u, noting that the $x_i \in D$ are distinct, $\deg \tilde{v}(x) = m - (k+r) = d - r$, and $\tilde{v}(0) = 1$. In other words, N_u gives the number of codewords f in $\mathcal{C}$ where $d(u, f) \leq |D| - k - r$. If N_u is positive, then $d(u, \mathcal{C}) \leq |D| - k - r$. By character sums,

$$N_u = \frac{1}{|\hat{B}|} \sum_{\substack{x_i \in D \\ \text{distinct}}} \sum_{\substack{\tilde{v}(x), \tilde{v}(0)=1 \\ \deg \tilde{v}(x) = d-r}} \sum_{\chi \in \hat{B}} \chi\left(\frac{(1 - x_1x)(1 - x_2x)\cdots(1 - x_{k+r}x)\tilde{v}(x)}{\tilde{w}(x)}\right)$$

For ease of notation, define

$$S_\chi(x_1, x_2, \ldots, x_{k+r}, x) = \chi\left(\frac{(1-x_1x)(1-x_2x)\cdots(1-x_{k+r}x)\tilde{v}(x)}{\tilde{w}(x)}\right)$$

First we will handle the case where $r < d$. To do this, consider the weighted version

$$N = \frac{1}{|\hat{B}|}\sum_{\substack{x_i \in D \\ \text{distinct}}}\sum_{\substack{\tilde{v}(x),\tilde{v}(0)=1 \\ \deg\tilde{v}(x)=d-r}}\Lambda(\tilde{v})\sum_{\chi\in\hat{B}} S_\chi(x_1, x_2, \ldots, x_{k+r}, x)$$

Note that if $N > 0$, then $N_u > 0$. Separating the trivial character from the sum gives

$$\begin{aligned} N &= \frac{1}{|\hat{B}|}\sum_{\substack{x_i \in D \\ \text{distinct}}}\sum_{\substack{\tilde{v}(x),\tilde{v}(0)=1 \\ \deg\tilde{v}(x)=d-r}}\Lambda(\tilde{v}) + \frac{1}{|\hat{B}|}\sum_{\substack{x_i \in D \\ \text{distinct}}}\sum_{\substack{\tilde{v}(x),\tilde{v}(0)=1 \\ \deg\tilde{v}(x)=d-r}}\Lambda(\tilde{v})\sum_{\substack{\chi\in\hat{B} \\ \chi\neq 1}} S_\chi(x_1, x_2, \ldots, x_{k+r}, x) \\ &= \frac{1}{|\hat{B}|}(|D|)_{k+r}(q^{d-r}-1) + \frac{1}{|\hat{B}|}\sum_{\substack{x_i \in D \\ \text{distinct}}}\sum_{\substack{\tilde{v}(x),\tilde{v}(0)=1 \\ \deg\tilde{v}(x)=d-r}}\Lambda(\tilde{v})\sum_{\substack{\chi\in\hat{B} \\ \chi\neq 1}} S_\chi(x_1, x_2, \ldots, x_{k+r}, x) \end{aligned}$$

Now we have to estimate

$$\left|N - \frac{1}{|\hat{B}|}(|D|)_{k+r}(q^{d-r}-1)\right| = \left|\frac{1}{|\hat{B}|}\sum_{\substack{\tilde{v}(x),\tilde{v}(0)=1 \\ \deg\tilde{v}(x)=d-r}}\Lambda(\tilde{v})\sum_{\substack{\chi\in\hat{B} \\ \chi\neq 1}}\sum_{\substack{x_i \in D \\ \text{distinct}}} S_\chi(x_1, x_2, \ldots, x_{k+r}, x)\right|$$

To do this, apply Li-Wan's new sieve. Let $X = D^{k+r}$, $\bar{X} = \{(x_1, x_2, \ldots, x_{k+r}) \in D^{k+r} \mid x_i \neq x_j, i \neq j\}$, $f(x) = \chi((1-x_1x)(1-x_2x)\cdots(1-x_{k+r}x))$, and $F = \sum_{x\in\bar{X}} f(x)$. We have that X is symmetric and f is normal, so we can compute F. For our case, we take

$$F_\tau = \sum \chi(1-x_{11}x)\cdots\chi(1-x_{1c_1}x)\cdots\chi^{k+r}(1-x_{(k+r)1}x)\cdots\chi^{k+r}(1-x_{(k+r)c_{k+r}}x)$$

where the sum runs over $x_{st_s} \in D$, $1 \leq s \leq k+r$, and $1 \leq t_s \leq c_s$. We will use the Li-Wan sieve estimate and the bounds

$$\left|\sum_{a\in D}\chi(1-ax)\right| \leq Kq^{1/2} \quad \text{and} \quad K \geq d$$

over all nontrivial $\chi \in \hat{B}$ with $\chi(\mathbb{F}_q^*) = 1$. We have that

$$
\begin{aligned}
&\left| N - \frac{1}{|\hat{B}|}(|D|)_{k+r}(q^{d-r}-1) \right| \\
&\leq \frac{1}{|\hat{B}|} \left| \sum_{\substack{\tilde{v}(x),\tilde{v}(0)=1 \\ \deg \tilde{v}(x)=d-r}} \Lambda(\tilde{v})\chi(\tilde{v}) \right| \left| \sum_{\substack{\chi\in\hat{B} \\ \chi\neq 1}} \chi^{-1}(\tilde{w}) \sum_{\substack{x_i\in D \\ \text{distinct}}} \chi\left((1-x_1x)(1-x_2x)\cdots(1-x_{k+r}x)\right) \right| \\
&\leq \frac{1}{|\hat{B}|} dq^{\frac{d-r}{2}} \sum_{\substack{\chi\in\hat{B} \\ \chi\neq 1}} \sum_{\sum ic_i = k+r} N(c_1, c_2, \ldots, c_{k+r}) |F_\tau| \\
&\leq \frac{1}{|\hat{B}|} dq^{\frac{3d-r}{2}} C_{k+r}(Kq^{1/2}, |D|, Kq^{1/2}, |D|, \ldots) \\
&\leq \frac{1}{|\hat{B}|} dq^{\frac{3d-r}{2}} \left(Kq^{1/2} + k + r + \frac{|D| - Kq^{1/2}}{2} - 1 \right)_{k+r} \\
&\leq \frac{1}{|\hat{B}|} dq^{\frac{3d-r}{2}} \left(Kq^{1/2} + k + r - \frac{Kq^{1/2}}{2} + \frac{|D|}{2} \right)_{k+r} \\
&\leq \frac{1}{|\hat{B}|} dq^{\frac{3d-r}{2}} \left(Kq^{1/2} + k + \frac{|D|}{2} \right)_{k+r}
\end{aligned}
$$

Therefore, it is sufficient to prove that

$$
(|D|)_{k+r}(q^{d-r}-1) > dq^{\frac{3d-r}{2}} \left(Kq^{1/2} + k + \frac{|D|}{2} \right)_{k+r}
$$

And since $d > r$, we have another sufficient condition:

$$
\frac{|D|}{Kq^{1/2} + k + \frac{|D|}{2}} > \left(\frac{dq^{\frac{3d-r}{2}}}{q^{d-r}-1} \right)^{\frac{1}{k+r}}
$$

If we take $K < c_1 \dfrac{|D|}{q^{1/2}}$ and $k < c_2|D|$, then it suffices to find c_1 and c_2 satisfying

$$
c_1 + c_2 < \left(\frac{dq^{\frac{3d-r}{2}}}{q^{d-r}-1} \right)^{-\frac{1}{k+r}} - \frac{1}{2}
$$

which means that it is enough to find

$$
\frac{1}{2} < \left(\frac{dq^{\frac{3d-r}{2}}}{q^{d-r}-1} \right)^{-\frac{1}{k+r}}
$$

By some rearrangement and simplification, we have

$$
k > \log_2 d + \left(\frac{d+r+1}{2} \right) \log_2 q - r
$$

A simpler condition can be prescribed. Since $r \leq d \leq K < q^{1/2}$, we can just replace $\log_2 d - r$ with $\frac{1}{2}\log_2 q$ to receive:

$$k > \left(\frac{d+r}{2} + 1\right)\log_2 q$$

For the case $r = d$, we use the unweighted counting function N_u, noting that $\tilde{v}(x) = 1$. Then we have

$$\begin{aligned} N_u &= \frac{1}{|\hat{B}|}\sum_{\substack{x_i \in D \\ \text{distinct}}} 1 + \frac{1}{|\hat{B}|}\sum_{\substack{x_i \in D \\ \text{distinct}}}\sum_{\substack{\chi \in \hat{B} \\ \chi \neq 1}} \chi\left(\frac{(1-x_1x)(1-x_2x)\cdots(1-x_{k+r}x)}{\tilde{w}(x)}\right) \\ &= \frac{1}{|\hat{B}|}(|D|)_{k+d} + \frac{1}{|\hat{B}|}\sum_{\substack{x_i \in D \\ \text{distinct}}}\sum_{\substack{\chi \in \hat{B} \\ \chi \neq 1}} \chi\left(\frac{(1-x_1x)(1-x_2x)\cdots(1-x_{k+d}x)}{\tilde{w}(x)}\right) \end{aligned}$$

Applying the same method as the previous case gives the estimate

$$\left| N_u - \frac{1}{|\hat{B}|}(|D|)_{k+d} \right| \leq \frac{q^d}{|\hat{B}|}\left(Kq^{1/2} + k + \frac{|D|}{2}\right)_{k+d}$$

and then it is enough to have

$$\frac{1}{|\hat{B}|}(|D|)_{k+d} > \frac{q^d}{|\hat{B}|}\left(Kq^{1/2} + k + \frac{|D|}{2}\right)_{k+d}$$

which actually gives a better bound than before. This concludes the proof. □

2.4.1 The Second Corollary

To prove the second corollary, we only need to find a number K satisfying the bounds

$$\left| \sum_{a \in (\mathbb{F}_q^*)^{\frac{q-1}{\ell}}} \chi(1-ax) \right| \leq Kq^{1/2} \quad \text{and} \quad K \geq d$$

over all characters χ from $\hat{B}$ in the main proof. This is where we can use Lemma 2.1, noting

that $\deg \bar{h}(x) = d+1$ and $\chi(\mathbb{F}_q^*) = 1$:

$$\begin{aligned}
\left| \sum_{a \in (\mathbb{F}_q^*)^{\frac{q-1}{\ell}}} \chi(1-ax) \right| &= \left| \sum_{a \in (\mathbb{F}_q^*)^{\frac{q-1}{\ell}}} \chi(-a)\chi(x-a^{-1}) \right| \\
&= \left| \sum_{a \in (\mathbb{F}_q^*)^{\frac{q-1}{\ell}}} \chi(x-a^{-1}) \right| \\
&= \left| \sum_{a \in (\mathbb{F}_q^*)^{\frac{q-1}{\ell}}} \chi(x-a) \right| \\
&\leq dq^{1/2}
\end{aligned}$$

So we have $K = d$, and the first bound is satisfied. The second bound $K \geq d$ is automatically satisfied. □

2.4.2 The Third Corollary

To prove the third corollary, we again need to find K satisfying

$$\left| \sum_{a \in \mathbb{F}_q} \chi(1-ax) \right| \leq Kq^{1/2} \quad \text{and} \quad K \geq d$$

As before, $\deg \bar{h}(x) = d+1$. Using Theorem 2.2 and the fact that $\chi(x) = 0$, since $\bar{h}(x)$ is divisible by x, we have

$$\left| \sum_{a \in \mathbb{F}_q} \chi(1-ax) \right| = \left| 1 + \sum_{a \in \mathbb{F}_q} \chi(x-a) \right| \leq (d-1)q^{1/2} \leq dq^{1/2}$$

Therefore $K = d$ is the suitable choice. □

Chapter 3

Deep Holes in Reed-Solomon Codes Based on Dickson Polynomials

Much of the previous work took Reed-Solomon codes with $D = \mathbb{F}_q$, $\mathbb{F}_q^*$, a subgroup of $\mathbb{F}_q^*$, or else a very large subset of $\mathbb{F}_q$. In joint work with Wan, we study the case where D is slightly more general - it will be the image of a Dickson polynomial over $\mathbb{F}_q$, which is defined as follows:

Definition (Dickson Polynomial)**.** Let n be a positive integer and $a \in \mathbb{F}_q$. The Dickson polynomial of degree n over $\mathbb{F}_q$ is defined as

$$D_n(x, a) = \sum_{i=0}^{\lfloor n/2 \rfloor} \frac{n}{n-i} \binom{n-i}{i} (-a)^i x^{n-2i}$$

A few examples of such polynomials:

n	$D_n(x, a)$
1	x
2	$x^2 - 2a$
3	$x^3 - 3ax$
4	$x^4 - 4ax^2 + 2a^2$

Table 3.1: Dickson polynomials for the parameter a

Note that when $a = 0$, $D_n(x, 0) = x^n$, so Dickson polynomials are a sort of generalisation of monomials. Of particular use to us is the size of the image of these polynomials, also known as the value set. A simple fact for the monomial $D_n(x, 0) = x^n$ is that the image of the map $D_n : \mathbb{F}_q^* \to \mathbb{F}_q^*$ has size $q - 1$ if $\gcd(n, q-1) = 1$ and size $(q-1)/\ell$ if $\gcd(n, q-1) = \ell$. In the first case, the map is 1-1; in the latter case, the map is ℓ-1. It turns out an analogous preimage-counting statement holds when $a \neq 0$. Chou, Mullen, and Wassermann in [9] studied this by using the functional form of $D_n(x, a)$. If $x = y + a/y$, then it can be shown that

$$D_n(y + a/y, a) = y^n + \frac{a}{y^n}$$

They used this to associate the size of a preimage of a point to the number of solutions to a certain equation. Using a character sum argument to count, they showed

Theorem 3.1. Let $n \geq 2$ and $a \in \mathbb{F}_q^*$. If q is even, then $|D_n^{-1}(D_n(x_0, a))| =$

$$\begin{cases} \gcd(n, q-1) & \text{if condition A holds} \\ \gcd(n, q+1) & \text{if condition B holds} \\ \dfrac{\gcd(n, q-1) + \gcd(n, q+1)}{2} & D_n(x_0, a) = 0 \end{cases}$$

where 'condition A' holds if $x^2 + x_0 x + a$ is reducible over $\mathbb{F}_q$ and $D_n(x_0, a) \neq \pm 0$; 'condition B' holds if $x^2 + x_0 x + a$ is irreducible over $\mathbb{F}_q$ and $D_n(x_0, a) \neq \pm 0$.

If q is odd, let η be the quadratic character of $\mathbb{F}_q$. If $2^r || (q^2 - 1)$ then $|D_n^{-1}(D_n(x_0, a))| =$

$$\begin{cases} \gcd(n, q-1) & \text{if } \eta(x_0^2 - 4a) = 1 \text{ and } D_n(x_0, a) \neq \pm 2a^{n/2} \\ \gcd(n, q+1) & \text{if } \eta(x_0^2 - 4a) = -1 \text{ and } D_n(x_0, a) \neq \pm 2a^{n/2} \\ \dfrac{\gcd(n, q-1)}{2} & \text{if } \eta(x_0^2 - 4a) = 1 \text{ and condition C holds} \\ \dfrac{\gcd(n, q+1)}{2} & \text{if } \eta(x_0^2 - 4a) = -1 \text{ and condition C holds} \\ \dfrac{\gcd(n, q-1) + \gcd(n, q+1)}{2} & \text{otherwise} \end{cases}$$

where 'condition C' holds if

$$2^t || n \text{ with } 1 \leq t \leq r-1, \eta(a) = -1, \text{ and } D_n(x_0, a) = \pm 2a^{n/2}$$

or

$$2^t || n \text{ with } 1 \leq t \leq r-2, \eta(a) = 1, \text{ and } D_n(x_0, a) = -2a^{n/2}$$

They also derived an explicit formula for the size of the value set of $D_n(x, a)$, denoted $|V_{D_n(x,a)}|$.

Theorem 3.2. Let $a \in \mathbb{F}_q^*$. If $2^r || (q^2 - 1)$ and η is the quadratic character on $\mathbb{F}_q$ when q is odd, then

$$|V_{D_n(x,a)}| = \frac{q-1}{2\gcd(n, q-1)} + \frac{q+1}{2\gcd(n, q+1)} + \delta$$

where

$$\delta = \begin{cases} 1 & \text{if } q \text{ is odd, } 2^{r-1} || n \text{ and } \eta(a) = -1 \\ \dfrac{1}{2} & \text{if } q \text{ is odd, } 2^t || n \text{ with } 1 \leq t \leq r-2 \\ 0 & \text{otherwise} \end{cases}$$

These results lead us to our main theorem.

3.1 Main Theorem

Theorem 3.3. Let $\mathcal{C}$ be the Reed-Solomon code over $\mathbb{F}_q$ with message length k, using the evaluation set $D = \{D_n(x, a) \mid x \in \mathbb{F}_q\}$, for $a \in \mathbb{F}_q^*$. Let u be a received word and $u(x)$ its interpolated polynomial with $\deg u(x) = k+1$. There exist positive constants c_1 and c_2 such that if the conditions

$$\frac{n+2}{2}\sqrt{q} < c_1|D| \text{ and } (\log_2 q) - 1 < k < c_2|D|$$

are satisfied, then u is not a deep hole.

Theorem 3.2 shows that D takes on a variety of sizes, large and small, depending on the parameters. This implies that progress is possible toward the deep hole problem for some small evaluation sets without any obvious algebraic structure to rely on.

3.1.1 Examples

Take $q = 2^8$ to consider a Reed-Solomon code over $\mathbb{F}_{2^8}$. When $n = 2$, the Dickson polynomial is $D_2(x, a) = x^2 - 2a = x^2$. By Theorem 3.2, $|V_{D_2(x,a)}| = |D| = 2^8$, which indicates that D_2 actually permutes the elements of $\mathbb{F}_{2^8}$, giving the evaluation set $D = \mathbb{F}_{2^8}$. Our theorem shows that if the message size satisfies

$$\frac{2+2}{2}\sqrt{2^8} < c_1 \cdot 2^8 \text{ and } \log_2 2^8 - 1 < k < c_2 \cdot 2^8$$

or simplified,

$$32 < 256c_1 \text{ and } 7 < k < 256c_2$$

for positive constants c_1 and c_2, then a received word of degree $k+1$ is not a deep hole. Our proof below will show that these constants satisfy the relation

$$256^{-\frac{1}{k+1}} - \frac{1}{2} > c_1 + c_2$$

For a wide range of k, fix $c_1 = .126$. The largest choice of c_2 is then

$$c_2 = 256^{-\frac{1}{k+1}} - \frac{1}{2} - .126$$

and we check the condition

$$7 < k < 256\left(256^{-\frac{1}{k+1}} - \frac{1}{2} - .126\right)$$

A computer algebra system shows that the supported message sizes are $14 \leq k \leq 78$.

Now take $q = 2^{16}$ to consider a Reed-Solomon code over $\mathbb{F}_{2^{16}}$. When $n = 3$, the Dickson polynomial family is $D_3(x, a) = x^3 - 3ax = x^3 + ax$. Theorem 3.2 gives $|D| = 43691$. The conditions to satisfy are

$$640 < 43691c_1 \text{ and } 15 < k < 43691c_2$$

Fix $c_1 = .015$. The largest choice of c_2 is

$$c_2 = 65536^{-\frac{1}{k+1}} - \frac{1}{2} - .015$$

to examine

$$15 < k < 43691\left(65536^{-\frac{1}{k+1}} - \frac{1}{2} - .015\right)$$

The supported message sizes are $16 \leq k \leq 21182$.

3.2 Preliminaries

We will prove a few prerequisite results need to prove this theorem. For convenience, a few theorems seen in the previous chapter will be stated.

3.2.1 Weil's Character Sum Bound

Our results rely on the following generalisation of Weil's classical character sum bound proved by Fu and Wan in [10]:

Theorem 3.4. Let $f_i(t)$ $(1 \leq i \leq n)$ be polynomials in $\mathbb{F}_q[t]$, let $f_{n+1}(t)$ be a rational function in $\mathbb{F}_q(t)$, let D_1 be the degree of the highest square free divisor of $\prod_{i=1}^{n} f_i(t)$, let $D_2 = 0$ if $\deg(f_{n+1}) \leq 0$ and $D_2 = \deg(f_{n+1})$ if $\deg(f_{n+1}) > 0$, let D_3 be the degree of the denominator of f_{n+1}, and let D_4 be the degree of the highest square free divisor of the denominator of $f_{n+1}(t)$ which is relatively prime to $\prod_{i=1}^{n} f_i(t)$. Let $\chi_i : \mathbb{F}_q^* \to \mathbb{C}^*$ $(1 \leq i \leq n)$

be multiplicative characters of $\mathbb{F}_q$, and let $\psi = \psi_p \circ \mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}$ for a non-trivial additive character $\psi_p : \mathbb{F}_p \to \mathbb{C}^*$ of $\mathbb{F}_p$. Extend χ_i to $\mathbb{F}_q$ by setting $\chi_i(0) = 0$. Suppose that $f_{n+1}(t)$ is not of the form $r(t)^p - r(t) + c$ in $\mathbb{F}_q(t)$. Then for any $m \geq 1$, we have

$$\left| \sum_{a \in \mathbb{F}_{q^m}, f_{n+1}(a) \neq \infty} \chi_1(\mathrm{N}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(f_1(a))) \cdots \chi_n(\mathrm{N}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(f_n(a))) \psi(\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(f_{n+1}(a))) \right| \leq (D_1 + D_2 + D_3 + D_4 - 1) q^{m/2}$$

where the sum is taken over those $a \in \mathbb{F}_{q^m}$ such that $f_{n+1}(a)$ is well-defined.

We specify the parameters to obtain various character sum bounds.

Corollary 3.1. Let $\psi_{\mathrm{Tr}} = \psi_p \circ \mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}$ be as above, $\psi : \mathbb{F}_q \to \mathbb{C}^*$ a non-trivial additive character, and $\eta : \mathbb{F}_q^* \to \mathbb{C}^*$ the quadratic character if q is odd. Set $m = 1$.

1. If $f_1(x) = D_n(x, a)$ with $a \neq 0$:
$$\left| \sum_{x \in \mathbb{F}_q} \psi(D_n(x, a)) \right| \leq (n-1)\sqrt{q}$$
2. If q is odd, $f_1(x) = x^2 - 4a$, and $f_2(x) = D_n(x, a)$ with $a \neq 0$:
$$\left| \sum_{x \in \mathbb{F}_q} \eta(x^2 - 4a) \psi(D_n(x, a)) \right| \leq (n+1)\sqrt{q}$$
3. If q is even and $f_1(x) = bD_n(x, a) + a/x^2$ with $a, b \neq 0$:
$$\left| \sum_{x \in \mathbb{F}_q^*} \psi_{\mathrm{Tr}} \left(bD_n(x, a) + a/x^2 \right) \right| \leq (n+2)\sqrt{q}$$

Note that none of the polynomials in place of $f_{n+1}(x)$ are of the form $r(t)^2 - r(t) + c$. For instance, in part 3, such an r would have to take the form $r(t) = d/x + f(x)$, where $f(x)$ is a polynomial. Expanding this shows that $d^2/x^2 + d/x = a/x^2$, or $d = 0$, which is a contradiction.

Lemma 3.1. Let $D = \{D_n(x, a) \mid x \in \mathbb{F}_q\}$ for $a \in \mathbb{F}_q^*$. If $\psi : \mathbb{F}_q \to \mathbb{C}^*$ is a non-trivial additive character, then the following estimates hold:

1. If q is even:
$$\left|\sum_{x\in D}\psi(x)\right| \leq (n+2)\sqrt{q}$$
2. If q is odd:
$$\left|\sum_{x\in D}\psi(x)\right| \leq (n+1)\sqrt{q}$$

Proof. The sum can be rewritten in the following way:
$$\sum_{y\in D}\psi(y) = \sum_{x\in\mathbb{F}_q}\psi(D_n(x,a))\frac{1}{N_x}$$
where $N_x = |D_n^{-1}(D_n(x,a))|$ is size of the preimage of the value $D_n(x,a)$.

When q is even:

By Theorem 3.1, N_x can be quantified. Let $\mathrm{Tr}:\mathbb{F}_q\to\mathbb{F}_2$ denote the absolute trace. Using the fact that z^2+xz+a is reducible over $\mathbb{F}_q$ if and only if $\mathrm{Tr}(a/x^2)=0$,
$$\begin{aligned}= &\sum_{\substack{x\in\mathbb{F}_q^*\\ \mathrm{Tr}(a/x^2)=0}}\frac{1}{\gcd(n,q-1)}\psi(D_n(x,a)) + \sum_{\substack{x\in\mathbb{F}_q^*\\ \mathrm{Tr}(a/x^2)=1}}\frac{1}{\gcd(n,q+1)}\psi(D_n(x,a))\\ &+\frac{1}{\gcd(n,q-1)}\psi(D_n(0,a)) + O(1)\end{aligned}$$
where $O(1)$ is a constant of size at most 1, which we accept by dropping the $D_n(x,0)=0$ case. Denote $\psi_1:\mathbb{F}_2\to\mathbb{C}^*$ as the order two additive character and $\psi_{\mathrm{Tr}}=\psi_1\circ\mathrm{Tr}$, which is an additive character from $\mathbb{F}_q\to\mathbb{C}^*$. Simplifying and rearranging gives
$$\begin{aligned}&= \frac{1}{2\gcd(n,q-1)}\sum_{x\in\mathbb{F}_q^*}\psi(D_n(x,a))(1+\psi_{\mathrm{Tr}}(a/x^2))\\ &\quad+\frac{1}{2\gcd(n,q+1)}\sum_{x\in\mathbb{F}_q^*}\psi(D_n(x,a))(1-\psi_{\mathrm{Tr}}(a/x^2)) + \frac{1}{\gcd(n,q-1)}\psi(D_n(0,a)) + O(1)\\ &= \left(\frac{1}{2\gcd(n,q-1)}+\frac{1}{2\gcd(n,q+1)}\right)\sum_{x\in\mathbb{F}_q^*}\psi(D_n(x,a))\\ &\quad+\left(\frac{1}{2\gcd(n,q-1)}-\frac{1}{2\gcd(n,q+1)}\right)\sum_{x\in\mathbb{F}_q^*}\psi(D_n(x,a))\psi_{\mathrm{Tr}}(a/x^2)\\ &\qquad+\frac{1}{\gcd(n,q-1)}\psi(D_n(0,a)) + O(1)\end{aligned}$$

We add and subtract $\left(\frac{1}{2\gcd(n,q-1)}+\frac{1}{2\gcd(n,q+1)}\right)\psi(D_n(0,a))$ to complete the first sum:

$$\begin{aligned}
&= \left(\frac{1}{2\gcd(n,q-1)}+\frac{1}{2\gcd(n,q+1)}\right)\sum_{x\in\mathbb{F}_q}\psi(D_n(x,a)) \\
&\quad + \left(\frac{1}{2\gcd(n,q-1)}-\frac{1}{2\gcd(n,q+1)}\right)\sum_{x\in\mathbb{F}_q^*}\psi(D_n(x,a))\psi_{\mathrm{Tr}}(a/x^2) \\
&\quad - \left(\frac{1}{2\gcd(n,q-1)}-\frac{1}{2\gcd(n,q+1)}\right)\psi(D_n(0,a)) + O(1)
\end{aligned}$$

In order to estimate the sum in second term, take $b\in\mathbb{F}_q$ so that $\psi(x)=\psi_{\mathrm{Tr}}(bx)$. Then,

$$\sum_{x\in\mathbb{F}_q^*}\psi(D_n(x,a))\psi_{\mathrm{Tr}}(a/x^2) = \sum_{x\in\mathbb{F}_q^*}\psi_{\mathrm{Tr}}(bD_n(x,a)+a/x^2)$$

Applying the bounds in Corollary 3.1,

$$\begin{aligned}
\left|\sum_{y\in D}\psi(y)\right| &\le \left(\frac{1}{2\gcd(n,q-1)}+\frac{1}{2\gcd(n,q+1)}\right)(n-1)\sqrt{q} \\
&\quad + \left|\frac{1}{2\gcd(n,q-1)}-\frac{1}{2\gcd(n,q+1)}\right|(n+2)\sqrt{q}+2 \\
&\le (n+2)\sqrt{q}
\end{aligned}$$

When q is odd:

We use Theorem 3.1 again to calculate N_x. Let η be the quadratic character of $\mathbb{F}_q$.

$$= \sum_{\substack{x\in\mathbb{F}_q \\ \eta(x^2-4a)=1}}\frac{1}{\gcd(n,q-1)}\psi(D_n(x,a)) + \sum_{\substack{x\in\mathbb{F}_q \\ \eta(x^2-4a)=-1}}\frac{1}{\gcd(n,q+1)}\psi(D_n(x,a)) + O(1)$$

The term $O(1)$ is a constant of size at most 2, which we accept by dropping the complicated 'condition C' and 'otherwise' cases. Simplifying and rearranging gives

$$\begin{aligned}
&= \frac{1}{2\gcd(n,q-1)}\sum_{x\in\mathbb{F}_q}\psi(D_n(x,a))(1+\eta(x^2-4a)) \\
&\quad + \frac{1}{2\gcd(n,q+1)}\sum_{x\in\mathbb{F}_q}\psi(D_n(x,a))(1-\eta(x^2-4a)) + O(1) \\
&= \left(\frac{1}{2\gcd(n,q-1)}+\frac{1}{2\gcd(n,q+1)}\right)\sum_{x\in\mathbb{F}_q}\psi(D_n(x,a)) \\
&\quad + \left(\frac{1}{2\gcd(n,q-1)}-\frac{1}{2\gcd(n,q+1)}\right)\sum_{x\in\mathbb{F}_q}\psi(D_n(x,a))\eta(x^2-4a) + O(1)
\end{aligned}$$

Again applying the bounds in Corollary 3.1,

$$\begin{aligned}\left|\sum_{x\in D}\psi(x)\right| &\le \left(\frac{1}{2\gcd(n,q-1)}+\frac{1}{2\gcd(n,q+1)}\right)(n-1)\sqrt{q}\\ &\quad+\left|\frac{1}{2\gcd(n,q-1)}-\frac{1}{2\gcd(n,q+1)}\right|(n+1)\sqrt{q}+2\\ &\le (n+1)\sqrt{q}\end{aligned}$$

which was to be shown. □

3.2.2 Li-Wan's New Sieve

Let D be a finite set and $D^k = D\times D\times\cdots\times D$ be the Cartesian product of k copies of D. Let X be a subset of D^k. Denote

$$\overline{X} = \{(x_1,x_2,\ldots,x_k)\in X \mid x_i\neq x_j, i\neq j\}$$

Let $f(x_1,x_2,\ldots,x_k)$ be a complex-valued function defined over X. Denote

$$F=\sum_{\mathbf{x}\in\overline{X}} f(x_1,x_2,\ldots,x_k)$$

Let S_k be the symmetric group on $\{1,2,\ldots,k\}$. Each permutation $\tau\in S_k$ can be uniquely factorised as a product of disjoint cycles and each fixed point is viewed as a trivial cycle of length 1. Namely,

$$\tau=(i_1i_2\ldots i_{a_1})(j_1j_2\ldots j_{a_2})\cdots(l_1l_2\ldots l_{a_s})$$

with $a_i\ge 1$ and $1\le i\le s$. Define

$$X_\tau=\{(x_1,x_2,\ldots,x_k)\mid x_{i_1}=\ldots=x_{i_{a_1}}, x_{j_1}=\ldots=x_{j_{a_2}},\cdots,x_{l_1}=\ldots=x_{l_{a_s}}\}$$

Similarly define

$$F_\tau=\sum_{\mathbf{x}\in X_\tau} f(x_1,x_2,\ldots,x_k)$$

We say that τ is of the type $(c_1,c_2,\ldots,c_k)$ if it has exactly c_i cycles of length i. Let $N(c_1,c_2,\ldots,c_k)$ be the number of permutations of type $(c_1,c_2,\ldots,c_k)$. Define

$$C_k(t_1,t_2,\ldots,t_k)=\sum_{\sum ic_i=k} N(c_1,c_2,\ldots,c_k)t_1^{c_1}t_2^{c_2}\cdots t_k^{c_k}$$

Now we have the following combinatorial result:

Lemma 3.2. Suppose $q \geq d$. If $t_i = q$ for $d|i$ and $t_i = s$ for $d \nmid i$, then we have

$$C_k(s, \ldots s, q, s, \ldots, s, q, \ldots) = k! \sum_{i=0}^{\lfloor k/d \rfloor} \binom{\frac{q-s}{d} + i - 1}{i} \binom{s + k - di - 1}{k - di}$$
$$\leq \left(s + k + \frac{q-s}{d} - 1 \right)_k$$

where $(x)_k = x(x-1)(x-2)\cdots(x-k+1)$.

Furthermore, we say that X is symmetric if for any $x \in X$ and any $g \in S_k$, we have $g \circ x \in X$. Also, if a complex-valued function f is defined on X, we say that it is normal on X if X is symmetric and for any two conjugate elements in S_k, τ and τ', we have

$$\sum_{x \in X_\tau} f(x_1, x_2, \ldots, x_k) = \sum_{x \in X_{\tau'}} f(x_1, x_2, \ldots, x_k)$$

Then, we have the result:

Lemma 3.3. If f is normal on X, then

$$F = \sum_{\sum i c_i = k} (-1)^{k - \sum c_i} N(c_1, c_2, \ldots, c_k) F_\tau$$

3.2.3 A Rephrasing of Error Distance

We will take a specialised version of the error distance characterisation seen in the previous chapter.

Lemma 3.4. Let $\mathcal{C}$ be a Reed-Solomon code over $\mathbb{F}_q$ using the evaluation set D. Let u be a received word and $u(x)$ be its interpolated polynomial with $\deg u(x) = k + 1$. The error distance satisfies $d(u, \mathcal{C}) \leq |D| - k - 1$ if and only if there exists a subset $\{x_{i_1}, x_{i_2}, \ldots, x_{i_{k+1}}\} \subset D$ such that

$$u(x) - v(x) = (x - x_{i_1})(x - x_{i_2}) \cdots (x - x_{i_{k+1}})$$

for some $v(x)$ with $\deg v(x) \leq k - 1$.

3.3 The Proof

Proof. For the received word u, write the interpolated polynomial as

$$u(x) = x^{k+1} - b_1 x^k + \ldots + (-1)^{k+1} b_{k+1}$$

By Lemma 3.4, if $u(x)$ is not a deep hole, then there exists a codeword $v(x)$ of degree $\leq k-1$ where

$$u(x) - v(x) = (x - x_1) \cdots (x - x_k)(x - x_{k+1})$$

for distinct values of $x_1, \ldots, x_k, x_{k+1}$ in D. By expanding this product, we see that u will not be a deep hole if and only if the equation

$$x_1 + \ldots + x_k + x_{k+1} = b_1$$

has a solution with distinct coordinates for every b_1 in $\mathbb{F}_q$. Let N_u be the number of solutions to this equation and G be the group of additive characters $\psi : \mathbb{F}_q \to \mathbb{C}^*$. By the orthogonality of characters,

$$N_u = \frac{1}{q} \sum_{\substack{x_i \in D \\ \text{distinct}}} \sum_{\psi \in G} \psi(x_1 + \ldots + x_k + x_{k+1} - b_1)$$

Removing the trivial character and exchanging the sums,

$$\begin{aligned}
\left| N_u - \frac{1}{q}(|D|)_{k+1} \right| &= \left| \frac{1}{q} \sum_{\substack{x_i \in D \\ \text{distinct}}} \sum_{\substack{\psi \in G \\ \psi \neq 1}} \psi(b_1)^{-1} \psi(x_1 + \ldots + x_k + x_{k+1}) \right| \\
&= \left| \frac{1}{q} \psi(b_1)^{-1} \sum_{\substack{\psi \in G \\ \psi \neq 1}} \sum_{\substack{x_i \in D \\ \text{distinct}}} \psi(x_1 + \ldots + x_k + x_{k+1}) \right| \\
&\leq \left| \sum_{\substack{x_i \in D \\ \text{distinct}}} \psi(x_1 + \ldots + x_k + x_{k+1}) \right|
\end{aligned}$$

Now we can estimate the right hand side using Li-Wan's new sieve. Let $X = D^{k+1}$, $f(x_1, \ldots, x_k, x_{k+1}) = \psi(x_1 + \ldots + x_k + x_{k+1}) = \psi(x_1) \cdots \psi(x_k)\psi(x_{k+1})$, so $F = \sum_{\mathbf{x} \in \overline{X}} f(\mathbf{x})$. F is symmetric and normal. Then

$$F_\tau = \sum \psi(x_{11}) \cdots \psi(x_{1c_1}) \cdots \psi^{k+1}(x_{(k+1)c_1}) \cdots \psi^{k+1}(x_{(k+1)c_{k+1}})$$

where the sum runs over $x_{st_s} \in D$, $1 \leq s \leq k+1$, and $1 \leq t_s \leq c_s$. Applying the larger of the estimates in Lemma 3.1 for a bound independent of of the evenness or oddness of q,

$$\begin{aligned}
\left| N_u - \frac{1}{q}(|D|)_{k+1} \right| &\leq \left| \sum_{\substack{x_i \in D \\ \text{distinct}}} \psi(x_1 + \ldots + x_k + x_{k+1}) \right| \\
&\leq \sum_{\sum ic_i = k+1} N(c_1, \ldots, c_{k+1}) |F_\tau| \\
&\leq C_{k+1}((n+2)\sqrt{q}, |D|, (n+2)\sqrt{q}, |D|, \ldots, (n+2)\sqrt{q}, |D|) \\
&\leq \left((n+2)\sqrt{q} + (k+1) + \frac{|D| - (n+2)\sqrt{q}}{2} - 1 \right)_{k+1} \\
&= \left(\frac{(n+2)\sqrt{q}}{2} + k + \frac{|D|}{2} \right)_{k+1}
\end{aligned}$$

To guarantee that $N_u > 0$, it suffices to have

$$\frac{1}{q}(|D|)_{k+1} > \left(\frac{(n+2)\sqrt{q}}{2} + k + \frac{|D|}{2} \right)_{k+1}$$

which is true if

$$\frac{|D|}{\frac{n+2}{2}\sqrt{q} + k + \frac{|D|}{2}} > q^{\frac{1}{k+1}}$$

If we take $\frac{n+2}{2}\sqrt{q} < c_1|D|$ and $k < c_2|D|$ for positive constants c_1 and c_2, we calculate the condition

$$q^{-\frac{1}{k+1}} - \frac{1}{2} > c_1 + c_2$$

Therefore, it is enough to find

$$q^{-\frac{1}{k+1}} > \frac{1}{2}$$

With some rearrangement, we have

$$k > (\log_2 q) - 1$$

□

Chapter 4

Conclusions

4.1 Summary and Future Work

In this dissertation, we searched for families of received words in Reed-Solomon codes that are not deep holes. For an (n, k) code, we managed to show that words represented by polynomials or rational functions of degree close to k (in the numerator) are not deep holes, given some restrictions on k and choosing $D = \mathbb{F}_q$ or a subgroup of $D = \mathbb{F}_q^*$. Our techniques in this case also allowed us to estimate the error distance. The approach is general in that for other choices of D, one only needs to fill in the character sum estimate

$$\left| \sum_{a \in D} \chi(1 - ax) \right| \leq Kq^{1/2}$$

with a suitable value of K to immediately produce similar results. In an initial attempt to study the problem of choosing D as the image of a Dickson polynomial, we tried to estimate this sum. For a Dickson polynomial $D_n(x, b)$, the sum can be written as

$$\left| \sum_{y \in \mathbb{F}_q} \chi(1 - D_n(y, b)x) \right|$$

Trying to estimate this combined with various Dickson polynomial forms and character sum techniques seen in the Chou, Mullen, and Wassermann paper [9] turned out to be difficult. Daqing Wan has suggested that it might work to factor $1 - D_n(y, b)x$ into linear terms over an appropriate extension field and deal with a product of characters. An appropriate theory still needs to be developed to continue this idea.

Instead of taking the character sum route, we looked into the weaker problem of showing that polynomials of degree $k+1$ do not represent deep holes when D is the image of $D_n(x, b)$. We reduced it to a type of restricted subset sum problem, trying to solve an equation of the form

$$x_1 + \ldots + x_r = c$$

where each of the $x_i \in D$ are distinct, and $c \in \mathbb{F}_q$ is arbitrary. We found conditions on q, n, and r which guarantee solutions. Another way to view this problem is to write $x_i = D_n(u_i, b)$ for suitable $u_i \in \mathbb{F}_q$. Then we see the equation as

$$D_n(u_1, b) + \ldots + D_n(u_r, b) = c$$

requiring distinct $D_n(u_i, b)$. This is actually a slightly more difficult version of Waring's problem for Dickson polynomials. The original, without the distinctness restriction, was studied by Gomez and Winterhof [11] and improved later by Ostafe and Shparlinski [22].

The deep hole problem on large or structured evaluation sets, as it stands currently, has progress from two sides: one classifying words whose degree is slightly larger than k, and the other classifying very specific words whose degree is close to n but with a small number of terms. If we stay with this strategy, the way to bridge this gap is to either guarantee solutions to complicated systems of equations or to improve sieving techniques and the Weil bound, all of which are difficult.

Another idea is to instead investigate the idea of deep hole trees by Cheng, Li, and Zhuang, which allowed them to partially resolve the problem. The limitation of their results to $q = p$ is caused by the use of a very specialised combinatorial result that applies only to prime fields. It doesn't seem to have an obvious generalisation.

One last direction is to follow the suggestion of Guruswami and Vardy that it may be easier to determine deep holes in codes using large or structured evaluation sets D. We can look for sets D where at least partial results are possible. Using techniques from Wan would call for producing estimates for incomplete character sums. This might be aided by understanding the behaviour of the value sets of certain polynomials, as seen with the Dickson polynomials in our results.

Hopefully, some combination of these concepts will bring this problem to a total resolution, giving us a better idea of the true capability of the Reed-Solomon code.

BIBLIOGRAPHY

[1] D. Augot and F. Morain. Discrete Logarithm Computations Over Finite Fields Using Reed-Solomon Codes. arXiv:1202.4361v1. 2012.

[2] E.R. Berlekamp. Algebraic Coding Theory. New York: McGraw-Hill, 1968.

[3] V.K. Bhargava, S.B. Wicker, et al. Reed-Solomon Codes and Their Applications. IEEE Press, Piscataway, NJ. 1994.

[4] Q. Cheng, J. Li, and J. Zhuang. On Determining Deep Holes of Generalized Reed-Solomon Codes. arXiv:1309.3546.

[5] A. Cafure, G. Matera, and M. Privitelli. Singularities of Symmetric Hypersurfaces and Reed-Solomon Codes. Advances in Mathematics of Communications, Vol. 6(1), 2012, pp.69-94.

[6] Q. Cheng and E. Murray. On deciding deep holes of Reed-Solomon codes. Proceedings of TAMC 2007, LNCS 4484, pp. 296-305

[7] Q. Cheng and D. Wan. On the List and Bounded Distance Decodibility of Reed-Solomon Codes. Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04).

[8] Q. Cheng and D. Wan. Complexity of Decoding Positive-Rate Primitive Reed-Solomon Codes. IEEE Transactions on Information Theory, Vol. 56, No. 10, October 2010: 5217-5222.

[9] W. Chou, G.L. Mullen, and B. Wassermann. On the number of solutions of equations of Dickson polynomials over finite fields. Taiwanese Journal of Mathematics Vol. 12, No. 4, pp. 917-931, July 2008.

[10] L. Fu and D. Wan. L-functions and Character Sums over Finite Fields. Preprint.

[11] D. Gomez and A. Winterhof. Warings Problem in Finite Fields with Dickson Polynomials. Finite fields: Theory and applications, Contemp. Math., vol. 477, Amer. Math. Soc., 2010, 185192.

[12] V. Guruswami. List Decoding of Error-Correcting Codes. Springer-Verlag Berlin Heidelberg, 2004.

[13] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. IEEE Transactions on Information Theory, Vol. 45, No. 6, September 1999: 1757-1767.

[14] V. Guruswami and A. Vardy. Maximum-Likelihood Decoding of Reed-Solomon Codes is NP-hard. SODA '05 Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms: 470-478. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2005.

[15] M. Keti, D. Wan, and G. Zhu. Computing Error Distance of Reed-Solomon Codes. Preprint.

[16] J. Li and D. Wan. On the subset sum problem over finite fields. Finite Fields and Their Applications, Volume 14, Issue 4, November 2008: 911-929

[17] J. Li and D. Wan. A new sieve for distinct coordinate counting. Science China Mathematics, Vol. 53 No.9: 2351-2362. Science China Press and Springer-Verlag Berlin Heidelberg, 2010.

[18] Y. Li and D. Wan. On error distance of Reed-Solomon codes, Science China Mathematics, Vol. 51, No. 11, 1982-1988. Science China Press and Springer-Verlag Berlin Heidelberg, 2008.

[19] Y. Li and G. Zhu. On error distance of received words with fixed degrees to Reed-Solomon code. Preprint.

[20] Q. Liao. On Reed-Solomon Codes. Chinese Annals of Mathematics, 32B(1), 89-98. Springer-Verlag Berlin Heidelberg, 2011.

[21] J.L. Massey. Shift-Register Syntehsis and BCH Decoding. IEEE Transactions on Information Theory, Vol. IT-15, No. 1, January 1969.

[22] A. Ostafe and I.E. Shparlinski. On the Waring Problem with Dickson Polyonmials in Finite Fields. Proceedings of the American Mathematical Society, Vol. 139, No. 11, November 2011, Pages 3815-3820.

[23] I.S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. Society for Industrial and Applied Mathematics, Vol. 8, No. 2, June 1960.

[24] D. Wan. Generators and irreducible polynomials over finite fields. Mathematics of Computation, 66, 119-1212 (1997).

[25] D. Wan and M. Keti. Deep Holes in Reed-Solomon Codes Based on Dickson Polynomials. Preprint.

[26] A. Weil. Basic Number Theory. Springer-Verlag, 1973.

[27] R. Wu and S. Hong. On deep holes of generalized Reed-Solomon codes. arXiv:1108.3524v2.

[28] J. Zhang, Fang-Wei Fu, and Qun-Ying Liao. New Deep Holes of Generalized Reed-Solomon Codes. arXiv:1205.6593v1.

[29] G. Zhu and D. Wan. Computing Error Distance of Reed-Solomon Codes. TAMC 2012, LNCS 7287, pp. 214-224, 2012. Springer-Verlag