Reed-Solomon Error-correcting Codes
The Deep Hole Problem

Matt Keti
(Advisor: Professor Daqing Wan)

Department of Mathematics
University of California, Irvine

November 8, 2012

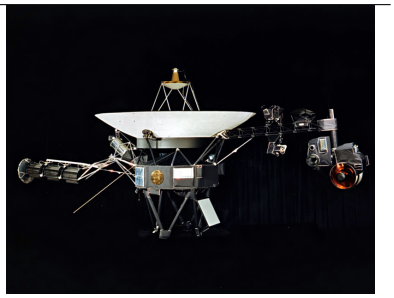
Abstract

In many types of modern communication, a message is transmitted over a noisy medium. This creates a chance that the message will become corrupted. The purpose of an error-correcting code is to add some redundant information to the message which allows the receiver to detect and correct those errors accrued during the transmission. We will study the famous Reed-Solomon code (found in QR codes, compact discs, deep space probes,...) and consider its error-correcting capacity. This will lead us to studying the "deep hole" problem, which is a question of determining when a received message has, in a sense, incurred the worst possible corruption. It is a new and important problem that could give insight on finding the upper bound for the error-correcting capacity of the Reed-Solomon code.

Contents

1 Introduction to Error-correcting Codes

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:

QR codes	Compact discs	Deep space probes
		

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.

1.2 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 relating 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:

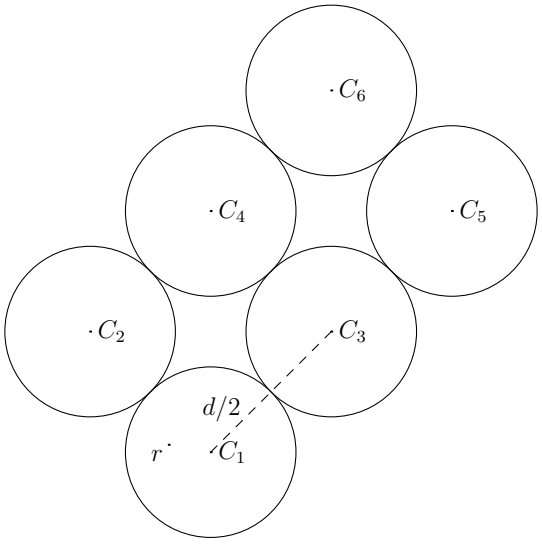


Figure 1: A geometric interpretation of minimum distance, where d is the minimum distance, C_i are codewords, and r is a received word

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	$1/N$	$4/7$

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.

2 Reed-Solomon Codes

2.1 Encoding and Decoding

In 1960, Irving S. Reed and Gustave Solomon published a paper titled 'Polynomial Codes Over Certain Finite Fields' [16]. 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 mixed-up. 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	$10 + 00x$

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 $c(x)$ were mixed-up, 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.

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

Decoding requires some sophistication. 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^i. If we do this, we have

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

where the last equality holds because the roots of $g(x)$ were α^i by design. We call these values the message syndromes and denote them by $S_i = r(\alpha^i) = e(\alpha^i)$. If all of the $S_i = 0$, then there was no transmission error. Otherwise, we have to use the values of S_i to determine error locations and their values. This idea can be thought of in terms of determining the error locator polynomial

$$\Lambda(x) = \prod_{i=1}^{t} (1 - x\alpha^i) = 1 + \Lambda_1 x + \Lambda_2 x^2 + \ldots + \Lambda_t x^t$$

noting that its roots give us information about the error positions. The error locator polynomial is related to the syndrome values by a set of linear equations (due to W. Peterson). This system can be solved using many famous algorithms, such as the one by Berelekamp-Massey, described in [2] [3] [15].

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
$$S_j = r(\alpha^j) = \sum_{k=1}^{t} e_{i_k} (\alpha^j)^{i_k}$$
for $j = 1, 2 \ldots, n-k$, where e_i is the error value in the i-th component of r, and t is the number of errors.
2. Find the error positions α^{i_k} and error values e_{i_k} by finding the error locator polynomial, and make the appropriate corrections in $r(x)$.

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

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

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

Theorem (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

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

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

  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:

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.

3 List Decoding

Given the highly applied nature of Reed-Solomon codes, one major problem has cropped up over the years: can the error-correcting limit 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 [8] [9].

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 [8].

4 Further Improving Error-Correction Limits

4.1 New Problems

For the same reasons as before, researchers are still interested in investigating the full error-correction capacity of Reed-Solomon codes. We can formulate two new decoding questions.

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 $d(r, c) \leq B$.

4.2 Hardness Results

Studying these problems has led to several hardness results. One major result comes from Guruswami and Vardy [10].

Theorem (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 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, both relying on the assumed hardness of the discrete logarithm problem over finite fields [6] [7]. 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 (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 (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 (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 (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 (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 $\mathbb{F}_q$.

4.3 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 set with algebraic structure. Given a received message r, to determine whether or not r is a deep hole.

5 Previous Results on Deep Holes

Let $u = (u_1, u_2, \ldots, u_n)$ be a received message. Use Lagrange Interpolation to find the associated polynomial $u(x)$. If $\deg u(x) \leq k - 1$, then u is a codeword and $d(u, \mathcal{C}) = 0$. Otherwise, we can put some simple bounds on the value of $d(u, \mathcal{C})$.

Theorem. 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 $N(\cdot)$ be 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. □

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

Cheng and Murray in [5] considered the deep hole problem for codes with evaluation set $D = \mathbb{F}_q$. They came up with the following conjecture:

Conjecture (Cheng-Murray)**.** The only deep holes are those words u such that $\deg u(x) = k$.

Though they could not prove this, they were able to reduce the problem to finding a rational point on an algebraic hypersurface and derived the following result, only for $q = p$:

Theorem (Cheng-Murray)**.** 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. As an 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.

Li and Wan [12] 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 (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 (Liao, 2011 [11])**.** 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 [4] slightly improved on one of Li-Wan's previous results with

Theorem (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 [21]. They came up with

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

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

Wu and Hong studied the deep hole problem when $D = \mathbb{F}_q^*$ [19]. Using this evaluation set allowed them to use the BCH formulation of Reed-Solomon codes.

Theorem (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 of the form $u(x) = ax^{q-2} + v(x)$ with $a \neq 0$, where $\deg v(x) \leq k - 1$, represent deep holes.

This work shows that the Cheng-Murray conjecture is false when $D = \mathbb{F}_q^*$. Wu and Hong attempted to modify the conjecture to include certain polynomials of degree $q - 2$. This would later be disproved by Zhang, Fu, and Liao.

Zhang, Fu, and Liao proved an extension of Wu-Hong that allows D to be any evaluation set except $\mathbb{F}_q$ [20]. 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 (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 of the form

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

where $\deg v(x) \leq k - 1$, represent deep holes.

Theorem (Zhang-Fu-Liao, 2012)**.** For $q > 4$, 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 of the form

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

where $\deg v(x) \leq k - 1$, represent deep holes.

Theorem (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 of the form

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

where $a \in \mathbb{F}_q^*$, $b, c \in \mathbb{F}_q$, and $\deg v(x) \leq k - 1$, do not represent deep holes.

The variety of these results makes the deep hole problem for $D = \mathbb{F}_q^*$ very uncertain.

6 Our Results

6.1 Extensions to Zhu-Wan

Zhu and Wan's original work applied for Reed-Solomon codes over $\mathbb{F}_q$ with evaluation set $D = \mathbb{F}_q$. In new work with Zhu and Wan, we were able to produce new statements for codes with slightly smaller evaluation sets.

Theorem. Let $\mathcal{C}$ be a Reed-Solomon code over $\mathbb{F}_q$ using the evaluation set D with $|D| > q/2$. 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, 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

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

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

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

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

We have a more familiar setting by taking $D = \mathbb{F}_q^*$ and $h(x) = 1$.

Corollary. Take a 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 $r \leq d := \deg(u(x)) - k \leq q - 2 - k$. There are positive constants c_1 and c_2 such that if

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

then $d(u, \mathcal{C}) \leq q - 1 - k - r$.

To get an idea of what these statements mean, we can look at some examples.

Example. Take a Reed-Solomon code over $\mathbb{F}_{2^8}$ with evaluation set $D = \mathbb{F}_{2^8}^*$. 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

$$d < 16c_1 \ \text{ and } \ 8.045\left(\frac{d+r}{2} + 1\right) < k < 256c_2$$

then $d(u, \mathcal{C}) \leq 255 - k - r$.

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 of the theorem, we can explicitly compute c_1 and c_2 using the formulas

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

To obtain a wide range of k, fix $c_1 = .0626$. Then we have the condition

$$16.09 < k < 256\left(255 \cdot 256^{-\frac{k+2}{k+1}} - \frac{1}{2} - .0626\right)$$

The inequality is satisfied when $17 \leq k \leq 96$. 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 $d(u, \mathcal{C}) \leq 254 - k$.

Similarly, 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 $30 \leq k \leq 59$. Such words u satisfy the estimate $d(u, \mathcal{C}) \leq 254 - k$.

Here is a table with a few examples of words covered by our bounds. We will denote α to 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	17	$x^{254} + x^{17} + 1$	$(x^{18} + x + 1)/x$
2	30	$x^{254} + x^{253} + x^{30}$	$(x^{32} + x + 1)/x^2$
2	30	$(\alpha^4 + 1)x^{254} + \ldots + (\alpha^6 + \alpha^3 + \alpha^2 + 1)$	$(x^{32} + x^2 + \alpha)/(x^2 - \alpha^2 x + \alpha + 1)$

6.2 Prerequisites for a Proof

Multiplicative Character

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 $(\mathbb{F}_q[x]/(h(x)))^*$. It can be extended to the entire group $\mathbb{F}_q[x]/(h(x))$ by setting $\chi(f(x)) = 0$ if $\gcd(f(x), h(x)) \neq 1$.

Weil's Character Sum Bound

As stated in [17]:

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

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

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 $\mathbb{F}_q^*$, these bounds need to be slightly modified:

Lemma 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 $\chi : (\mathbb{F}[x]/(h_1(x)))^* \to \mathbb{C}^*$ 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. □

Li-Wan's New Sieve

We also state Li-Wan's new sieve (as in [13] and [21]): 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. Suppose $q \geq d$. If $t_i = q$ for $d|i$ and $t_i = s$ for $d \nmid i$, then we have

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

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. If f is normal on X, then

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

A Rephrasing of Error Distance

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

Lemma 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 $u(x)$ its interpolated polynomial with $\deg u(x) = k + d$, where $k + 1 \leq k + d \leq q - 1$. The error distance $d(u, \mathcal{C}) \leq |D| - k - r$ 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 result follows.

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

6.3 The Proof

Our idea is to phrase the problem in terms of finding solutions to a polynomial congruence. In particular, our results will be proved if we can guarantee that there is at least one solution to our congruence. To do this, we will use a character sum argument combined with estimations from Li-Wan's new sieve.

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 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

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

where Λ denotes the von Mangoldt function. 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 $X = \mathbb{F}_q^{k+r}$, $\bar{X} = \{(x_1, x_2, \ldots, x_{k+r}) \in \mathbb{F}_q^{k+r} \mid x_i \neq x_j, i \neq j\}$, $f(x) = \chi((1 - x_1x)(1 - x_2x)\cdots(1 - x_{k+r}x))$, 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 is over $x_{st_s} \in D$, $1 \le s \le k+r$, and $1 \le t_s \le c_s$. We will use the Li-Wan sieve estimate and the bounds

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

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

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

Therefore, it is sufficient to prove that

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

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

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

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

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

which means that it is enough to find

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

By some rearrangement, we have the condition when $|D| > \frac{q}{2}$

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

A simpler condition can be prescribed. Since $r \le d \le K < q^{1/2}$, we can just replace $\log_2 d - r\left(\log_2|D| + 1 - \log_2 q\right)$ with $\frac{1}{2}\log_2 q$ to receive:

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

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

$$N_u = \frac{1}{|\hat{B}|} \sum_{\substack{x_i \in D \\ \text{distinct}}} 1 + \frac{1}{|\hat{B}|} \sum_{\substack{x_i \in D \\ \text{distinct}}} \sum_{\substack{\chi \in \hat{B} \\ \chi \neq 1}} \chi\left(\frac{(1 - x_1 x)(1 - x_2 x) \cdots (1 - x_{k+r} x)}{\tilde{w}(x)} \right)$$

$$= \frac{1}{|\hat{B}|} (|D|)_{k+d} + \frac{1}{|\hat{B}|} \sum_{\substack{x_i \in D \\ \text{distinct}}} \sum_{\substack{\chi \in \hat{B} \\ \chi \neq 1}} \chi\left(\frac{(1 - x_1 x)(1 - x_2 x) \cdots (1 - x_{k+d} x)}{\tilde{w}(x)} \right)$$

Applying the same method as the previous case gives the estimate

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

and then it is enough to have

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

which gives a similar estimate as the previous case. This concludes the proof. □

The Corollary

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

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

over all characters χ from $\hat{B}$ in the main proof. This is where we can use Lemma 1, noting that $\deg \bar{h}(x) = d+1$ and taking $\ell = q - 1$:

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

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

6.4 Summary and Further Work

In certain Reed-Solomon codes with large evaluation sets satisfying certain character sum bounds, we have found families of received words that are not deep holes. We can apply these results specifically for $D = \mathbb{F}_q$ or $D = \mathbb{F}_q^*$. (In the former case, we recover Zhu-Wan's original result.) These results raise a few questions:

1. Our techniques only work when $|D| > q/2$, which prevents us from making any statements when D is relatively small. Can we come up with better estimates to allow for smaller D, particularly when D is a subgroup of $\mathbb{F}_q^*$?

2. We were also required to assume $d < c_1 q^{1/2}$. If it were possible to relax the condition to $d < c_1 q^{1/2+\varepsilon}$, we would be able to choose much smaller values of c_1 to get better information rates on our codes. Is it possible to do this?

In the general scope of deep holes, researchers have approached the problem from two ways:

1. Find families of words that are not deep holes.

2. Find families of words that are deep holes.

Most current results only apply to families of very low degree (slightly higher than k) or very high degree (slightly less than q). Our goal is to try to fill up the gap and complete the classification of deep holes. Doing this would help us gain further insight on the nature of the Reed-Solomon decoding problem.

References

[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] 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.

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

[6] 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).

[7] 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.

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

[9] 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.

[10] 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.

[11] Q. Liao. On Reed-Solomon Codes. Chinese Annals of Mathematics, 32B(1), 8998. Springer-Verlag Berlin Heidelberg, 2011.

[12] 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

[13] 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.

[14] 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.

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

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

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

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

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

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

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