ON DECODING INTERLEAVED REED-SOLOMON CODES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

OĞUZ YAYLA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
CRYPTOGRAPHY

SEPTEMBER 2011

Approval of the thesis:

ON DECODING INTERLEAVED REED-SOLOMON CODES

submitted by **OĞUZ YAYLA** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Department of Cryptography, Middle East Technical University** by,

Prof. Dr. Ersan Akyıldız ________________
Director, Graduate School of **Applied Mathematics**

Prof. Dr. Ferruh Özbudak ________________
Head of Department, **Cryptography**

Prof. Dr. Ferruh Özbudak ________________
Supervisor, **Mathematics Department, METU**

Examining Committee Members:

Prof. Dr. Ersan Akyıldız ________________
Mathematics Department, METU

Prof. Dr. Ferruh Özbudak ________________
Mathematics Department, METU

Assoc. Prof. Dr. Ali Doğanaksoy ________________
Mathematics Department, METU

Dr. Hamdi Murat Yıldırım ________________
Dept. of Computer Tech. and Information Systems, Bikent Uni.

Dr. Burcu Gülmez Temür ________________
Mathematics Department, Atılım University

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: OĞUZ YAYLA

Signature :

ABSTRACT

ON DECODING INTERLEAVED REED-SOLOMON CODES

Yayla, Oğuz
Ph.D., Department of Cryptography
Supervisor : Prof. Dr. Ferruh Özbudak

September 2011, 53 pages

Probabilistic simultaneous polynomial reconstruction algorithm of Bleichenbacher-Kiayias-Yung is extended to the polynomials whose degrees are allowed to be distinct. Furthermore, it is observed that probability of the algorithm can be increased. Specifically, for a finite field $\mathbb{F}$, we present a probabilistic algorithm which can recover polynomials $p_1, \ldots, p_r \in \mathbb{F}[x]$ of degree less than $k_1, k_2, \ldots, k_r$, respectively with given field evaluations $p_l(z_i) = y_{i,l}$ for all $i \in I$, $|I| = t$ and $l \in [r]$ with probability at least $1 - (n - t)/|\mathbb{F}|$ and with time complexity at most $O((nr)^3)$. Next, by using this algorithm, we present a probabilistic decoder for interleaved Reed-Solomon codes. It is observed that interleaved Reed-Solomon codes over $\mathbb{F}$ with rate R can be decoded up to burst error rate $\frac{r}{r+1}(1-R)$ probabilistically for an interleaving parameter r. It is proved that a Reed-Solomon code RS$(n; k)$ can be decoded up to error rate $\frac{r}{r+1}(1 - R')$ for $R' = \frac{(k-1)(r+1)+2}{2n}$ when probabilistic interleaved Reed-Solomon decoders are applied. Similarly, for a finite field $\mathbb{F}_{q^2}$, it is proved that q-folded Hermitian codes over $\mathbb{F}_{q^{2q}}$ with rate R can be decoded up to error rate $\frac{q}{q+1}(1 - R)$ probabilistically. On the other hand, it is observed that interleaved codes whose subcodes would have different minimum distances can be list decodable up to radius of minimum of list decoding radiuses of subcodes. Specifically, we present a list decoding algorithm for C, which is interleaving of $C_1, \ldots, C_b$ whose minimum distances would be different, decoding up to radius of minimum of list decoding radiuses of $C_1, \ldots, C_b$ with list size polynomial in the maximum of list sizes of $C_1, \ldots, C_b$ and with time complexity polynomial in list size of C and b. Next, by using this list decoding algorithm for interleaved codes, we obtained new list decoding algorithm for qh-folded Hermitian codes for q standing for field size the code defined and h is any positive integer. The decoding algorithm list decodes qh-folded Hermitian codes for radius that is generally better than radius of Guruswami-Sudan algorithm, with time complexity and list size polynomial in list size of h-folded Reed-Solomon codes defined over $\mathbb{F}_{q^2}$.

Keywords: interleaved codes, interleaved Reed-Solomon codes, simultaneous polynomial reconstruction, list decoding, folded Hermitian codes

ÖZ

GEÇMELİ REED-SOLOMON KODLARININ ÇÖZÜMLENMESİ ÜZERİNE

Yayla, Oğuz
Doktora, Kriptografi Bölümü
Tez Yöneticisi : Prof. Dr. Ferruh Özbudak

Eylül 2011, 53 sayfa

Bleichenbacher-Kiayias-Yung olasılıksal eşzamanlı polinom geriçatılması algoritması polinomların dereceleri farklı olduğu durumlara genelleştirilmiştir. Ayrıca, olasılığın artırılabileceği de gözlemlenmiştir. Belirgin şekliyle, $\mathbb{F}$ sonlu bir cisimi için, $p_l(z_i) = y_{i,l}$ $i \in I$, $|I| = t$ ve $l \in [r]$ polinom değerleri verildiğinde, dereceleri sırasıyla $k_1, k_2, \dots, k_r$ olan $p_1, \dots, p_r \in \mathbb{F}[x]$ polinomlarını en azından $1-(n-t)/|\mathbb{F}|$ olasılığı ve en çok $O((nr)^3)$ hesaplama karmaşıklığında eşzamanlı geriçatan olasılıksal bir algoritma sunulmuştur. Bu algoritmanın kullanımıyla geçmeli Reed-Solomon kodları için olasılıksal çözümleyici sunulmuştur. Bu çözümleyici, bigi oranı R olan $\mathbb{F}$ üzerinde tanımlı r geçmeli Reed-Solomon kodları $\frac{r}{r+1}(1-R)$ yığılma hata oranına kadar olasılıksal çözümleyebilir. Olasılıksal geçmeli Reed-Solomon kod çözümleyicisi kullanıldığında verilen bir Reed-Solomon kodu RS$(n; k)$ $R' = \frac{(k-1)(r+1)+2}{2n}$ için $\frac{r}{r+1}(1-R')$ hata oranına kadar çözümlenebileceği ispatlanmıştır. Benzer şekilde, bir $\mathbb{F}_{q^2}$ sonlu cismi için, bilgi oranı R olan ve $\mathbb{F}_{q^{2q}}$ üzerinde tanımlı q-katlı Hermityan kodlarının $\frac{q}{q+1}(1-R)$ hata oranına kadar olasılıksal çözümlenebileceği istalanmıştır. Diğer taraftan, altkodlarının en az aralıkları farklı olduğunda bile geçmeli kodların, altkodlarının en küçük liste çözümleme çapına kadar çözümlenebileceği gösterilmiştir. Diğer bir ifadeyle, en az aralıkları farklı olabilen $C_1, \dots, C_b$ kodlarının geçirilmesi ile oluşan C kodunu, $C_1, \dots, C_b$ kodlarının en küçük liste çözümleme çapına kadar, $C_1, \ldots, C_b$ kodlarının en büyük liste boyutunun polinom katı liste büyüklüğünde ve bu büyüklük ve b sabitinin bir polinom katı zamanda çözebilen bir algoritma sunulmuştur. Bu algoritmanın kullanımıyla tanımlandığı cisim büyüküğü q olan ve herhangibir h doğal sayısı için qh-katlı Hermityan kodlarını listeleme yöntemiyle çözümleyebilen yeni bir algoritma sunulmuştur. Bu algoritma, qh-katlı Hermityan kodlarını, Guruswami-Sudan algoritmasından daha iyi bir çapa kadar, $\mathbb{F}_{q^2}$ üzerinde tanımlı h-katlı Reed-Solomon kodlarının liste büyüklüğünün polinom katı hesap karmaşıklığında ve liste büyüklüğünde çözümler.

Anahtar Kelimeler: geçmeli kodlar, geçmeli Reed-Solomon kodlar, eşzamanlı polinom geriçatılması, listeleme yöntemiyle kod çözümlemesi, katlı Hermityan kodlar

Eşim Saniye'ye ve Kızım Duru'ya

ACKNOWLEDGMENTS

"No problem can be solved from the same level of consciousness that created it."

Albert Einstein

I would like to express my gratitude to all those who supported me in completing this thesis. I would like to express my deepest gratitude to my supervisor Prof. Dr. Ferruh Özbudak for his guidance and insight he provided throughout this research. His ideas and tremendous support had a major influence on this thesis.

I am indebted to all the members of the Institute of Applied Mathematics at Middle East Technical University for their friendship, understanding and help. I would also like to take this opportunity to thank to Prof. Dr. Ersan Akyıldız, director of IAM, for his support during my graduate study. Many thanks to all my colleagues and office mates at the Middle East Technical University.

I also would like to send my special thanks to all my close relatives and my parents-in-law, Gülersönmez family.

And, a huge thanks to my family: to Mum and Dad - thank you for all your support over the years; to Selim, for being there no matter what.

And finally, I would like to thank my wife Saniye and my daughter Duru for their extraordinary patience and continuous support during the preparation of this thesis and being with me all the way. Without their unending love, neither this thesis nor my life would be complete.

I also gratefully acknowledge the financial support of TÜBİTAK BİDEB 2211 Program during this thesis.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

CHAPTER 1

INTRODUCTION

Interleaved codes are defined to be interleaving finitely many subcodes column wise. Each column of any codeword of the interleaved codes is a codeword of the subcodes. And, each row is treated as a single symbol of the interleaved code. An illustrated codeword **c** of a interleaved code C which is interleaving of subcodes $C_1, C_2, \dots, C_r$ is presented below. Each $c_i = (c_{i,1}, c_{i,2}, \dots, c_{i,n})^T$ is a codeword in C_i for $i = 1, 2, \dots, r$.

$$\mathbf{c} = (c_1, c_2, \dots, c_r) = \begin{pmatrix} c_{1,1} & c_{2,1} & \dots & c_{r,1} \\ c_{1,2} & c_{2,2} & \dots & c_{r,2} \\ \vdots & \vdots & & \vdots \\ c_{1,n} & c_{2,n} & \dots & c_{r,n} \end{pmatrix}$$

In this thesis, we generally consider Reed-Solomon codes as subcodes. In a Reed-Solomon code RS(n, k) defined over a finite field, a message of length less than k is encoded into a polynomial of degree less than k as coefficients, and codewords are evaluations of those polynomials at n distinct elements of the finite field. The decoding problem of Reed-Solomon codes known as the Polynomial Reconstruction (PR) Problem is one of the challenging problems in coding theory and cryptography. Some well known RS decoding algorithms are briefly presented in Chapter 2.

An Interleaved Reed-Solomon (IRS) code for interleaving parameter r consists of r many Reed-Solomon codes of length n and dimensions $k_1, k_2, \dots, k_r$. If dimensions of subcodes are equal to each other, the code is called homogeneous IRS-code, otherwise it is called heterogeneous IRS. Therefore, decoding of IRS codes relies on a variation of PR called Simultaneous Polynomial Reconstruction (SPR). Bleichenbacher, Kiayias, and Yung [2] present a probabilistic algorithm that solves SPR problem for homogeneous case. Their algorithm is probabilistic and is employed in settings where instance of the algorithm is assumed to be random.

In this thesis, we mainly focus on heterogeneous IRS codes and their application. Schmidt, Sidorenko and Bossert [32] present an algorithm (SSB algorithm) using Berlekamp-Massey approach to decode heterogeneous IRS codes for burst errors, i.e. for errors occurring at each subcode (row) simultaneously. We present a probabilistic solution of SPR problem using Berlekamp-Welch approach which also decodes IRS codes for burst errors in Section 3.1.

Apart from decoding IRS codes, we also study list decoding of interleaved codes (possibly nonlinear) which is a more general frame work compared to decoding IRS codes. A list-decoding algorithm is an algorithm which tries to construct a list which includes codewords within a specified radius of an input codeword. The central problem of list decoding is to identify the radius up to which a list decoding algorithm exists in terms of the output list size and efficiency of the algorithm. Gopalan et al.[8] showed that any interleaved codes can be list decodable up to radius what subcode can be. Moreover, they presented an efficient algorithm for list decoding of interleaved codes. In Section 4.2, we present a list decoding algorithm for heterogeneous interleaved codes running in polynomial time and output list does not contain so many codewords.

Besides interleaving, Reed-Solomon codes are also generalized by choosing polynomials and evaluation elements from a generalized space. Algebraic-geometry (AG) codes are such generalization of Reed-Solomon codes. Polynomials are chosen from the Riemann-Roch space of an algebraic function field and evaluation elements are chosen as rational places of the function filed. For instance, Hermitian codes are defined on the function field of Hermitian curve. It is shown in [22] that Hermitian codes can be written as interleaving of heterogeneous Reed-Solomon codes. Hence, single and list decoders of heterogeneous IRS codes can be applied to decoding Hermitian codes and list decoding Hermitian codes, which is presented in Section 5.2 and Section 5.3, respectively.

CHAPTER 2

BERLEKAMP-WELCH ALGORITHM AND ITS GENERALIZATIONS

In this chapter, we first briefly mention about the decoding problem of Reed-Solomon codes. Then, a solution to this problem is presented in the point of view of Berlekamp and Welch [1]. Later, Sudan's algorithm extending Berlekamp-Welch approach for giving a list decoding algorithm of Reed-Solomon codes is presented. Finally, main idea of Guruswami-Sudan algorithm which improves the Sudan's algorithm is mentioned.

2.1 Decoding Problem

In Reed-Solomon (RS) codes $RS(n, k)$ of length n and dimension k defined over a finite field $\mathbb{F}_q$, message $m = (p_0, p_1, \ldots, p_{k-1})$ is embedded into the polynomials $p(x) = p_0 + p_1x + \ldots + p_{k-1}x^{k-1}$ of degree at most $k - 1$. Note that all $p_{k-1}, p_{k-2}, \ldots, p_{k-j}$ may be zero for some $j \leq k$. Next, the message is encoded as polynomial evaluations at distinct field elements $z_1, z_2, \ldots, z_n \in \mathbb{F}_q$ such that $c = (p(z_1), p(z_2), \ldots, p(z_n))$. This whole procedure is called RS encoding. After that, codeword c is sent.

At the receiver end, the receiver gets $r = (y_1, y_2, \ldots, y_n)$ and knows the evaluation values $z_1, z_2, \ldots, z_n$. She also assumes that at least t many of y_i satisfy $y_i = p(z_i)$ for $i = 1, 2, \ldots, n$. She aims to recover m or equivalently $p(x)$ from the pairs (z_i, y_i) for $i = 1, 2, \ldots, n$, known as decoding problem of RS codes.

2.2 Berlekamp-Welch Algorithm

In Berlekamp-Welch approach to decoding problem of RS codes, the receiver tries to find degree $n-t$ polynomial $E(x)$ and degree at most $n-t+k-1$ polinomial $m(x)$ satisfying the linear system

$$m(z_i) = y_i E(z_i), \text{ for } i = 1, 2, \ldots, n. \tag{2.1}$$

It can be shown that pair defined as

$$E(x) = x^{n-t-\Delta(y,p(z_i))_{i=1}^{n}} \prod_{j=1}^{e'} (x - z_{i_j}),\ m(x) = p(x)E(x)$$

satisfies (2.1) where z_{i_j} for $j = 1, 2, \ldots, e' \leq n-t$ are the error locations occurred during the sending operation of c. This is the reason why a linear system as (2.1) is constructed and to be solved. We note that we also handle the case that more than $\frac{n+k-1}{2}$ agreements occur as it is applied in the definition of $E(x)$.

In Berlekamp-Welch approach, the receiver tries to solve (2.1) having totally at most $2(n-t)+k$ unknowns which are coefficients of $E(x)$ and $m(x)$. We know the existence of a solution but not uniqueness. There may be many solutions. We only require that the ratio $\frac{m(x)}{E(x)}$ is unique, which is enough for uniqueness of the message. If number of agreement places satisfies $t > \frac{n+k-1}{2}$, then this requirement is achieved. The proof is as follow. Let $(E_1(x), m_1(x))$ and $(E_2(x), m_2(x))$ be two pairs satisfying (2.1). Then define $R(x) := E_1(x)m_2(x) - E_2(x)m_1(x)$ which has n roots $z_1, z_2, \ldots, z_n$. However, $R(x)$ has degree at most $2(n-t)+k-1 < n$ as $t > \frac{n+k-1}{2}$. Therefore, R is a zero polynomial, and we have the equalities $\frac{m_1(x)}{E_1(x)} = \frac{m_2(x)}{E_2(x)} = \frac{m(x)}{E(x)} = p(x)$.

We now present the Berlekamp-Welch algorithm below as Algorihtm 1.

We observed that Berlekamp-Welch algorithm returns the correct solution if the number of errors is less than $\frac{n-k+1}{2}$. Otherwise, the algorithm returns a failure, which means that number of errors is not in the responsible range. This is the desired decoding procedure.

We study extension of Berlekamp-Welch approach for Interleaved Reed-Solomon codes in Chapter 3. Apart from Berlekamp-Welch approach, we impose some extra conditions on the linear system so that it has unique solution, and then we try to find the unique solution, which is the one constructed as a proof of existence.

Algorithm 1 Berlekamp-Welch Algorithm

```
Require: $\mathbb{F}_q, n \leq q, 1 < k \leq n$, pairwise distinct $z_1, z_2, \ldots, z_n \in \mathbb{F}_q$, $y_1, y_2, \ldots, y_n \in \mathbb{F}_q$ and $t > \frac{n+k-1}{2}$.
Ensure: $p(x)$ of degree at most $k-1$ satisfying $p(z_i) = y_i$ for at least $t$ values of $i = 1, 2, \ldots, n$ or FAILURE.
Solve the linear system (2.1). If not possible,return FAILURE.
Compute a solution of the system: $m(x)$ and $E(x)$
Find $p(x) = m(x)/E(x)$. If not possible, return FAILURE.
return $p(x)$ if $p(z_i) = y_i$ for at least $t$ values of $i = 1, 2, \ldots, n$ else return FAILURE.
```

One can also consider Berlekamp-Welch approach as follows

- Interpolate $R(x, y) = m(x) - yE(x)$ where degree of m is at most $n - t + k - 1$ and degree of E is $n - t$ and

- Find the root $p(x)$ of $Q[x](y) := R(x, y)$.

Correctness follows form the fact that $R(x, p(x))$ has t roots $z_{i_1}, z_{i_2}, \ldots, z_{i_t}$ but its degree is $n - t + k - 1 < t$ if t is required to be greater than $\frac{n+k-1}{2}$. Therefore, $R(x, p(x))$ is a zero polynomial, i.e. $p(x)$ is a root of $R(x, y)$. On the other hand, we already know the existence of m and E.

Sudan extends this approach, and presents a crucial list decoding algorithm for RS codes in his milestone work [35]. This is presented in the next section.

2.3 Sudan's Algorithm

Sudan observed that Berlekamp-Welch algorithm has two stages interpolation and root finding as stated in the previous section. Besides that, Sudan used the crucial fact that the receiver has at least t agreements, hence interpolating a polynomial $Q(x, y) = Q_0(x) + yQ_1(x)$ having y-degree 1 from the pairs (z_i, y_i) for $i = 1, \ldots, n$ and satisfying $degQ_0 < t$ and $degQ_1 < t-(k-1)$ makes $Q(x, p(x))$ have more roots than its degree, that is, a zero polynomial. Therefore, $p(x)$ is a root of $Q(x, y)$.

Interpolation can be accomplished by solving the linear system whose variables are coefficients of $Q(x, y)$. Interpolation always finds a solution when number of coefficients $2t - k + 1$ of $Q(x, y)$ is greater than the number of equations n, i.e. when $t > \frac{n+k-1}{2}$, which was also the case in Berlekamp-Welch case.

Sudan extended this approach to the more general case: interpolate a $Q(x, y)$ whose y-degree is greater than 1 then find all its roots to constitute a list of possible message polynomials, one of which is the sent message $p(x)$. This is called as list decoding algorithm. In this case, the number of coefficients of $Q(x, y) = Q_0(x) + yQ_1(x) + \ldots + y^D Q_D(x)$ where $degQ_i < t - i(k - 1)$ for $i = 0, 1, \ldots, D$ and $D < t/(k - 1)$ is

$$\sum_{i=0}^{\lfloor \frac{t}{k-1} \rfloor} (t - i(k-1)) = (\left\lfloor \frac{t}{k-1} \right\rfloor + 1)(t - \left\lfloor \frac{t}{k-1} \right\rfloor \frac{k-1}{2}).$$

On the other hand, we have n linear equations. If number of unknowns exceeds the number of equations, than there always exist such a polynomial Q, that is when $t > \sqrt{2n(k - 1)}$.

Sudan's algorithm is presented below as Algorithm 2.

Algorithm 2 Sudan's Algorithm

```
Require: $\mathbb{F}_q, n \leq q, 1 < k \leq n$, pairwise distinct $z_1, z_2, \ldots, z_n \in \mathbb{F}_q$, $y_1, y_2, \ldots, y_n \in \mathbb{F}_q$ and
    $t > \sqrt{2n(k-1)}$.
Ensure: $L = \{p(x)$ of degree at most $k - 1$ satisfying $p(z_i) = y_i$ for at least $t$ values of
    $i = 1, 2, \ldots, n\}$
  Interpolate $Q(x, y) = Q_0(x) + yQ_1(x) + \ldots + y^D Q_D(x)$ where $degQ_i < t - i(k - 1)$ for
    $i = 0, 1, \ldots, D$ and $D < t/(k - 1)$
  find all degree at most $k - 1$ polynomials $p(x)$ such that $Q(x, p(x)) \cong 0$. If $p(z_i) = y_i$ for
    at least $t$ many of $i = 1 \ldots, n$, then include $p(x)$ into output list $L$
  return $L$.
```

In 1999, Guruswami and Sudan [11] further extended the above approach as follows. They also imposed on Q that it has multiplicity r at each pair (z_i, y_i) for $i = 1, \ldots, n$ so that number of roots of $Q(x, p(x))$ increases to rt.

In this case, the number of coefficients of $Q(x, y) = Q_0(x) + yQ_1(x) + \ldots + y^D Q_D(x)$ where $degQ_i < rt - i(k - 1)$ for $i = 0, 1, \ldots, D$ and $D < rt/(k - 1)$ is

$$\sum_{i=0}^{\lfloor \frac{rt}{k-1} \rfloor} (rt - i(k-1)) = (\left\lfloor \frac{rt}{k-1} \right\rfloor + 1)(rt - \left\lfloor \frac{rt}{k-1} \right\rfloor \frac{k-1}{2}).$$

On the other hand, we have $n\binom{r+1}{2}$ linear equations since we also impose new constraints on the coefficients of Q. If number of unknowns exceeds the number of equations, that is if $t > \sqrt{n(k-1)(1+1/r)}$, then there always exists such a polynomial Q.

We note that Guruswami-Sudan algorithm is also valid for more general codes than RS codes called as algebraic-geometric codes, and we study application of Guruswami-Sudan algorithm for a special algebraic-geometric code family Hermitian codes in Section 5.3.

CHAPTER 3

DECODING INTERLEAVED REED-SOLOMON CODES

We consider Interleaved Reed-Solomon (IRS) code C including r Reed-Solomon codes C_1, $C_2, \ldots, C_r$ of length n and dimensions $k_1, k_2, \ldots, k_r$, respectively. A codeword c of C consists of columns of codewords from each code C_i, $i = 1, \ldots, r$. If $k_1 = k_2 = \cdots = k_r$, the code is called homogeneous IRS code, otherwise it is called heterogeneous IRS code.

Interleaved Reed-Solomon (IRS) codes are investigated by different authors like Krachkovsky, Lee, and Garg [17, 18, 19], Bleichenbacher, Kiayias, and Yung [2], Brown, Minder and Shokrollahi [4, 5], Justesen, Thommesen, and Hoholdt [16], as well as Parvaresh and Vardy [21]. Then, Schmidt, Sidorenko and Bossert study IRS codes in their recent publications [27, 28, 29, 30, 31, 32]. IRS codes are mainly considered in applications where error bursts occur, since IRS codes are most effective if correlated errors affect all words of the interleaved scheme simultaneously.

Decoding heterogeneous IRS codes have first been considered in [19], and this is based on Berlekamp-Massey approach. Some of the properties of heterogeneous constructions have been investigated in [30]. Heterogeneous IRS codes are also considered in decoding of a single RS code beyond half the minimum distance like described in [31], where the problem of decoding a single low-rate RS code is transformed into the problem of decoding a heterogeneous IRS code.

In this chapter, we mainly focus on decoding heterogeneous IRS codes and their application. In Section 3.1, we first extend the study of Bleichenbacher, Kiayias, and Yung [2] that proposes a probabilistic algorithm based on Berlekamp-Welch Algorithm for simultaneous reconstruction of polynomials whose degrees are allowed to be distinct. Next, in Section 3.2, we decode heterogeneous IRS codes with the algorithm given in Section 3.1. Then, we apply this heterogeneous IRS decoder to RS codes and folded-Hermitian codes in Section 3.3 and Section 5.2, respectively.

3.1 Simultaneous Polynomial Reconstruction

The decoding problem of RS codes known as the Polynomial Reconstruction (PR) Problem is one of the challenging problems in coding theory. We look for a variation of PR called Simultaneous Polynomial Reconstruction (SPR). Firstly, we extend definition of SPR given in [2] to the heterogeneous case. We use the notation $[n]$ for the set $\{1, 2, \ldots, n\}$.

Definition 3.1.1 *(Simultaneous Polynomial Reconstruction-SPR). For parameters n, t, r, k_1, $k_2, \ldots, k_r \in \mathbb{N}$ and pairwise distinct $z_1, \ldots, z_n \in \mathbb{F}$, an instance of SPR is a set of tuples $\langle y_{i,1}, \ldots, y_{i,r} \rangle_{i=1}^{n}$ over a finite field $\mathbb{F}$ that satisfies the following: There exists an $I \subset [n]$ with $|I| = t$, and polynomials $p_1, \ldots, p_r \in \mathbb{F}[x]$ of degree less than $k_1, k_2 \ldots, k_r$, respectively such that $p_l(z_i) = y_{i,l}$ for all $i \in I$ and $l \in [r]$.*

Bleichenbacher, Kiayias, and Yung [2] present a probabilistic algorithm based on Berlekamp-Welch algorithm that solves SPR problem for $k_1 = k_2 = \cdots = k_r$. Their algorithm is probabilistic and is employed in settings where instance of the algorithm is assumed to be random. SPR problem for distinct k_i, $i = 1, \ldots, r$ can be solved by simply choosing $k = \max\{k_1, \ldots, k_r\}$ and running BKY algorithm for this k. But, this method solves SPR problem for larger t. In this section, we present a method with handling distinct degrees, which solves the problem probabilistically for smaller t under the random instance assumption, which means that polynomials $p_1, \ldots, p_r \in \mathbb{F}[x]$ and values $y_{i,l}$ for $i \in [n] - I$ and $l \in [r]$ are random.

We start with some observations for the SPR problem. Let $\langle y_{i1}, \ldots, y_{ir} \rangle_{i=1}^{n}$ be a given SPR instance. In the Berlekamp-Welch approach [1, 10], we know that error locator polynomial of a message polynomial $p_l(x)$ given by $E(x) = \Pi_{i \notin I}(x - z_i)$ of degree $n - t$ with leading term 1 and received message polynomial $m_l(x) = p_l(x)E(x)$ of degree $n - t + k_l - 1$ satisfy the linear system

$$m_l(z_i) = y_{il}E(z_i) \; i = 1, \ldots, n \tag{3.1}$$

for each $l = 1, \ldots, r$ where coefficients of polynomials m_l and E for $l = 1, \ldots, r$ are defined as variables.

Thus we know the existence of a solution to (3.1). If, in addition, it has a unique solution property, the solution will be $(m_1, m_2, \ldots, m_r, E)$ as constructed above. Then, $p_l(x)$ for $l = 1, 2, \ldots, r$ are easily recovered by divisions. It is easy to observe that (3.1) has a unique solution if it has full rank and the number of equations nr is as large as the number of unknowns $r(n-t) + \sum_{j=1}^{r} k_j + n - t$. The system does not have the full rank property for some instances. A necessary condition that (3.1) has full rank is $t \geq k_l$ for each $l = 1, 2, \ldots, r$. In other words, if $t < k_l$ for some $l \in [r]$, then (3.1) has never full rank. On the other hand, the number of equations is as large as the number of unknowns if and only if

$$nr \geq r(n-t) + \sum_{j=1}^{r} k_j + n - t$$

or equivalently,

$$t \geq \frac{n + \sum_{j=1}^{r} k_j}{r+1}. \tag{3.2}$$

Therefore, Algorithm 3 presented below is a generalization of BKY algorithm according to Definition 3.1.1 solving SPR problem probabilistically for t values satisfying

$$t \geq \max\{k_1, k_2, \ldots, k_r, \frac{n + \sum_{j=1}^{r} k_j}{r+1}\}. \tag{3.3}$$

Algorithm 3 Simultaneous Polynomial Reconstruction Probabilistically

```
Require: $\langle y_{i,1}, \ldots, y_{i,r} \rangle_{i=1}^{n} \in \mathbb{F}^{nr}$. Parameters $n, k_1, \ldots, k_r, t \in \mathbb{N}$ and distinct $z_1, \ldots, z_n \in \mathbb{F}$
    satisfying $\max\{k_1, k_2, \ldots, k_r, \frac{n+\sum_{j=1}^{r} k_j}{r+1}\} \leq t \leq n \leq |\mathbb{F}|$.
Ensure: $\{p_1, \ldots, p_r\}$ satisfying for each $l \in [r]$, $p_l(z_i) = y_{i,l}$ at least $t$ values of $i \in [n]$ or
    FAILURE.
Form the linear system (3.1). Let the matrix of the system be $A$.
if $A$ is full rank then
    Compute solution of the system: $m_1$,$m_2$,. . .,$m_r$ and $E$
    Find $p_1 = m_1/E, \ldots, p_r = m_r/E$.
    return $\{p_1, \ldots, p_r\}$.
else
    return FAILURE
end if
```

Our algorithm is an extension of BKY algorithm [2], but they have different sub-procedures, and this difference is explained below. Moreover, BKY algorithm is not designed to produce an output in the case $z_i = 0$ for any $i \in [n]$, but Algorithm 3 is capable to do that, which is a necessary case in Section 5.2.

Similar to [2], the matrix A constructed by the algorithm is full rank with high probability, assuming that the SPR input to the algorithm is distributed randomly.

Theorem 3.1.2 *Assuming the SPR instance is random, the system (3.1) constructed by Algorithm 3 accepts at most one solution with probability at least* $1 - (n-t)/|\mathbb{F}|$.

Proof. Let $\langle y_{i,1}, \ldots, y_{i,r} \rangle_{i=1}^{n}$ be an instance to the SPR problem and let A denote the matrix of the system of linear equations (3.1).

We start by investigating the structure of the matrix A. Consider the following matrices, for $l = 1, \ldots, r$:

$$M_l = \begin{bmatrix} 1 & z_1 & z_1^2 & \ldots & z_1^{n-t+k_l-1} \\ 1 & z_2 & z_2^2 & \ldots & z_2^{n-t+k_l-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & z_n & z_n^2 & \ldots & z_n^{n-t+k_l-1} \end{bmatrix}$$

$$E_l = \begin{bmatrix} y_{1,l} & y_{1,l}z_1^1 & \ldots & y_{1,l}z_1^{n-t-1} \\ y_{2,l} & y_{2,l}z_2^1 & \ldots & y_{2,l}z_2^{n-t-1} \\ \vdots & \vdots & & \vdots \\ y_{n,l} & y_{n,l}z_n^1 & \ldots & y_{n,l}z_n^{n-t-1} \end{bmatrix}$$

Given these definitions, it follows that the matrix of the system (3.1) can be written as follows:

$$A = \begin{bmatrix} M_1 & 0 & \ldots & 0 & -E_1 \\ 0 & M_2 & \ldots & 0 & -E_2 \\ \vdots & \vdots & & \vdots & \\ 0 & 0 & \ldots & M_r & -E_r \end{bmatrix}$$

We know that A is full rank if its any row eliminated minor $\hat{A}$ is nonsingular. Hence, probability that A is full rank is always higher than probability that $\hat{A}$ is nonsingular. In order to calculate the probability that $\hat{A}$ is nonsingular, we follow a way similar to [2] but we have to take care of distinct degrees.

We define a square submatrix $\hat{A}$ of A by eliminating $rt - \sum_{l=1}^{r} k_l - (n-t)$ rows. The main aim of the row elimination is removing rows having subindices in I from each block of $\hat{A}$. We fix $I = \{n-t+1, n-t+2, \ldots, n\}$ and do some row elimination according to I. But, similar elimination can be designed for different I.

Firstly, we remove last $t - k_l$ rows from the l-th block if $(r+1-l)t - \sum_{i=l}^{r} k_l$ is less than $rt - \sum_{l=1}^{r} k_l - (n-t)$ for $l = r, r-1, r-2, \ldots, c+2, c+1$, and next, we remove $rt - \sum_{l=1}^{r} k_l - (n-t) - ((r-c)t - \sum_{i=c+1}^{r} k_l)$ rows from the c-th block for some $1 \leq c \leq r$. Therefore, we totally removed $rt - \sum_{l=1}^{r} k_l - (n-t)$ rows which is the difference of the number of rows and columns of A. Thus, $\hat{A}$, which is the row-removed version of A, is a square matrix:

$$\hat{A} = \begin{bmatrix} \hat{M}_1 & 0 & \ldots & 0 & -\hat{E}_1 \\ 0 & \hat{M}_2 & \ldots & 0 & -\hat{E}_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \ldots & \hat{M}_r & -\hat{E}_r \end{bmatrix}. \tag{3.4}$$

Now, we observe that $\hat{A}$ is nonsingular with high probability. In order to calculate this probability easily, we do some row rearrangement according to I. The main aim of the row rearrangement is reducing each M_l to a square matrix by moving some rows of each block having subindices in $[n]-I$ to the bottom of $\hat{A}$. We reduce each M_l to a square matrix by the following rules: Define $s_l := t - k_l$ for $l = 1, 2, \ldots, c-1$, $s_c = t - k_l - (rt - \sum_{l=1}^{r} k_l - (n-t) - ((r-c)t - \sum_{i=c+1}^{r} k_l))$ and $s_l = 0$ for $l = c+1, c+2, \ldots, r$. We move first s_1 rows from the first block to the bottom of $\hat{A}$, so that $\hat{M}_1$ reduces to $\hat{M}_1^*$ and $\hat{E}_1$ reduces to $\hat{E}_1^*$. Then, we move s_2 rows numbered as $s_1+1, s_1+2, \ldots, s_1+s_2$ from the second block to the bottom of $\hat{A}$, so that $\hat{M}_2$ reduces to $\hat{M}_2^*$ and $\hat{E}_2$ reduces to $\hat{E}_2^*$. By continuing the same procedure for each block, we reach to a new matrix:

$$\hat{A}^* = \begin{bmatrix} \hat{M}_1^* & 0 & \ldots & 0 & -\hat{E}_1^* \\ 0 & \hat{M}_2^* & \ldots & 0 & -\hat{E}_2^* \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \ldots & \hat{M}_r^* & -\hat{E}_r^* \\ V_1 & V_2 & \ldots & V_r & M \end{bmatrix}$$

We know that $\hat{A}$ is nonsingular if and only if $\hat{A}^*$ is nonsingular. Hence, we first show that $\det\hat{A}^*$ is a non zero polynomial. Then, we obtain a probability for nonsingularity of $\hat{A}^*$.

We consider det$\hat{A}^*$ as a multivariate polynomial of the variables $y_{i,l}$ and $a_{j,l}$ for $i \in [n] - I$, $l \in [r]$ where $a_{j,l}$ are coefficients of the polynomials p_l for $l \in [r]$. Hence, det$\hat{A}^*$ is a non zero polynomial if it is nonzero for some assigned values of the variables.

Define $S_{l-1} := \sum_{j=0}^{l-1} s_j$ and assign $y_{i,l} = 0$ for all $l \in [r]$ and $i \in \{S_{l-1}+1, \ldots, S_{l-1}+s_l\}$ so that M is a zero matrix.

Next, we assign $p_l(x) = 1$ for $l \in [r]$, and $y_{i,l} = 1$ for $i \in [n] - I - \{S_{l-1}+1, \ldots, S_{l-1}+s_l\}$ and $l \in [r]$ so that the remaining $y_{i,l} = 1$. This reduces $\hat{E_l}^*$ to

$$\hat{E_l}^* = \begin{bmatrix} 1 & z_1 & \ldots & z_1^{n-t} \\ \vdots & & \vdots & \\ 1 & z_{S_{l-1}} & \ldots & z_{S_{l-1}}^{n-t-1} \\ 1 & z_{S_l+1} & \ldots & z_{S_l+1}^{n-t-1} \\ \vdots & \vdots & & \vdots \\ 1 & z_{n-t+k_l} & \ldots & z_{n-t+k_l}^{n-t-1} \end{bmatrix}$$

Then, we row-reduce all rows of V_l with $\hat{M}_l^*$ for $l \in [r]$. This is possible since $\hat{M}_l^*$ is a Vandermonde matrix i.e. a full rank matrix. Moreover, it is easy to observe that while eliminating rows of $V_1, \ldots, V_r$, submatrix M is transformed to

$$M' = \begin{bmatrix} 1 & z_1 & z_1^2 & \ldots & z_1^{n-t-1} \\ 1 & z_2 & z_2^2 & \ldots & z_2^{n-t-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & z_{n-t} & z_{n-t}^2 & \ldots & z_{n-t}^{n-t-1} \end{bmatrix}$$

After row reduction, $\hat{A}^*$ transforms to

$$A' = \begin{bmatrix} M_1' & 0 & \ldots & 0 & -E_1' \\ 0 & M_2' & \ldots & 0 & -E_2' \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \ldots & M_r' & -E_r' \\ 0 & 0 & \ldots & 0 & M' \end{bmatrix}$$

Since A' is a triangular matrix with nonsingular diagonal block matrices, its determinant is nonzero, which implies that det$\hat{A}^*$ is a nonzero polynomial.

When det$\hat{A}^*$ is expanded, it is observed that it has combined degree $n - t$. By Schwarz's Lemma, the probability that det$\hat{A}^*$ takes zero values is deg(det$\hat{A}^*$)$/|\mathbb{F}| = (n-t)/|\mathbb{F}|$. Therefore, we proved the theorem. ■

Theorem 3.1.2 does not say anything about the existence of a solution of the system (3.1). However, we know that error locator polynomial $E(x)$ and message polynomials $m_1(x), m_2(x), \ldots, m_r(x)$ satisfy (3.1). Therefore, the system (3.1) has a unique solution and this solution can be found by solving the linear system corresponding to A if it is full rank.

Proposition 3.1.3 *If $t \geq \left\lceil \frac{n+min\{k_1,\ldots,k_r\}}{2} \right\rceil$, then Algorithm 3 always returns the solution of the instance.*

Proof. Assume that $k_1 = min\{k_1, \ldots, k_r\}$. Then, let $t \geq \left\lceil \frac{n+k_1}{2} \right\rceil$ and consider the following matrix

$$[M_1|E_1] = \left[\begin{array}{cccc|ccc}
1 & z_1 & \cdots & z_1^{n-t+k_1-1} & y_{1,1} & \cdots & y_{1,1}z_1^{n-t-1} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots \\
1 & z_{n-t+k_1} & \cdots & z_{n-t+k_1}^{n-t+k_1-1} & y_{n-t+k_1,1} & \cdots & y_{n-t+k_1,1}z_{n-t+k_1}^{n-t-1} \\
\hline
1 & z_{n-t+k_1+1} & \cdots & z_{n-t+k_1+1}^{n-t+k_1-1} & y_{n-t+k_1+1,1} & \cdots & y_{n-t+k_1+1,1}z_{n-t+k_1+1}^{n-t-1} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots \\
1 & z_t & \cdots & z_t^{n-t+k_1-1} & y_{t,1} & \cdots & y_{t,1}z_t^{n-t-1} \\
1 & z_{t+1} & \cdots & z_{t+1}^{n-t+k_1-1} & y_{t+1,1} & \cdots & y_{t+1,1}z_{t+1}^{n-t-1} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots \\
1 & z_n & \cdots & z_n^{n-t+k_1-1} & y_{n,1} & \cdots & y_{n,1}z_n^{n-t-1}
\end{array}\right]$$

We also assume that $I = \{1, 2, \ldots, n\}$, then it is known that $y_{i,1}z^j = p(z_i)z^j$ for $i \in I$ and $j = 0, \ldots, n-t-1$. Thus, the matrix constructed above is column equivalent to the following full rank matrix.

$$\left[\begin{array}{cccc|ccc}
1 & z_1 & \cdots & z_1^{n-t+k_1-1} & & & \\
\vdots & \vdots & & \vdots & & 0 & \\
1 & z_{n-t+k_1} & \cdots & z_{n-t+k_1}^{n-t+k_1-1} & & & \\
\hline
1 & z_{n-t+k_1+1} & \cdots & z_{n-t+k_1+1}^{n-t+k_1-1} & 0 & \cdots & 0 \\
\vdots & \vdots & & \vdots & \vdots & & \vdots \\
1 & z_t & \cdots & z_t^{n-t+k_1-1} & 0 & \cdots & 0 \\
1 & z_{t+1} & \cdots & z_{t+1}^{n-t+k_1-1} & y_{t+1,1} + p(z_{t+1}) & \cdots & (y_{t+1,1} + p(z_{t+1}))z_{t+1}^{n-t-1} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots \\
1 & z_n & \cdots & z_n^{n-t+k_1-1} & y_{n,1} + p(z_n) & \cdots & (y_{n,1} + p(z_n))z_n^{n-t-1}
\end{array}\right]$$

This shows that $[M_1|E_1]$ is a full rank matrix. Similarly, this is true for any I. Therefore, A is a full rank matrix, and (3.1) has at most one solution, but we know that (3.1) has a unique solution and this solution can be found by solving the linear system when A is a full rank matrix. ■

We need to point out the time complexity of the Algorithm 3. Most time consuming operation through the algorithm is solving the linear system corresponding to A, whose dimension is at most $nr \times nr$. Hence, time complexity of the algorithm is at most $O((nr)^3)$ with Gaussian elimination method. Therefore, we conclude the following result by combining Theorem 3.1.2 and Proposition 3.1.3.

Theorem 3.1.4 *Assuming the SPR instance is random, Algorithm 3 returns the solution of the instance with probability at least* $1 - (n-t)/|\mathbb{F}|$ *for t values satisfying* $max\{k_1, \ldots, k_r, \left\lceil \frac{n+\sum_{j=1}^{r} k_j}{r+1} \right\rceil\} \leq t < \left\lceil \frac{n+min\{k_1,\ldots,k_r\}}{2} \right\rceil$ *with time complexity at most* $O((nr)^3)$. *And, Algorithm 3 never fails for t values satisfying* $\left\lceil \frac{n+min\{k_1,\ldots,k_r\}}{2} \right\rceil \leq t \leq n$.

Apart from Algorithm 3, BKY algorithm first defines a square matrix $\hat{A}$ with row elimination from the end of each block of A without regarding I, then solves the eliminated linear system if $\hat{A}$ is nonsingular. $\hat{A}$ in BKY algorithm is defined as (3.4). We observe that the probability that $\hat{A}$ is nonsingular decreases with a high ratio when row elimination effects all blocks of $\hat{A}$ in BKY algorithm.

Proposition 3.1.5 *If* E_1 *has any row that is eliminated and consists of* y_{1i} *for some* $i \in [n] - I$, *then* $\hat{A}$ *is never full rank. Moreover, the probability of BKY Algorithm decreases with ratio approximately* $\frac{\binom{n-x_1}{n-t}}{\binom{n}{n-t}}$ *where* $x_1 = 2t - k_1 - n$ *is the number of eliminated rows from the first block.*

Proof. We first observe that if E_1 has an eliminated row, then $rt - \sum_{l=1}^{r} k_l - (n-t) > (r-1)t - \sum_{l=2}^{r} k_l$, i.e. $t > \frac{n+k_1}{2}$.

Then, let $[M_1|E_1]$ be given as follows

$$[M_1|E_1] = \left[\begin{array}{cccc|ccc} 1 & z_1 & \cdots & z_1^{n-t+k_1-1} & y_{1,1} & \cdots & y_{1,1}z_1^{n-t-1} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & z_n & \cdots & z_n^{n-t+k_1-1} & y_{n,1} & \cdots & y_{n,1}z_n^{n-t-1} \end{array}\right]$$

We assume that disagreement occurs at $\{t+1, t+2, \ldots, n\}$ and that the last row is eliminated. The proof can be generalized for other cases easily.

Now, row eliminated matrix $\left[\hat{M}_1 | \hat{E}_1\right]$ column reduces to

$$\left[\begin{array}{cccc|ccc}
1 & z_1 & \ldots & z_1^{n-t+k_1-1} & 0 & \ldots & 0 \\
\vdots & \vdots & & \vdots & \vdots & & \vdots \\
1 & z_t & \ldots & z_t^{n-t+k_1-1} & 0 & \ldots & 0 \\
1 & z_{t+1} & \ldots & z_{t+1}^{n-t+k_1-1} & y_{t+1,1} + p(z_{t+1}) & \ldots & (y_{t+1,1} + p(z_{t+1}))z_{t+1}^{n-t-1} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots \\
1 & z_{n-1} & \ldots & z_{n-1}^{n-t+k_1-1} & y_{n-1,1} + p(z_{n-1}) & \ldots & (y_{n-1,1} + p(z_{n-1}))z_{n-1}^{n-t-1}
\end{array}\right]$$

since we know that $y_{i,1}z^j = p(z_i)z^j$ for $i = 1, \ldots, t$ and $j = 0, \ldots, n-t-1$. The row $< 1, z_t, \ldots, z_t^{n-t+k_1-1} >$ in M_1 is row dependent by first $t-1$ rows since $n - t + k_1 < \frac{n+k_1}{2} < t$. Thus, we observed that $\hat{A}$ is equivalent to a matrix having a zero row. We conclude that $\det(\hat{A})$ is zero. Therefore, we proved the first part of the theorem. To prove the second part, we calculate the ratio of instances not satisfying the first part of the theorem. This is equivalent to $\frac{\binom{n-x_1}{n-t}}{\binom{n}{n-t}}$ where x_1 is the number of eliminated rows from first block. ■

However, probability decrease given in Proposition 3.1.5 occurs if one chooses $t > \frac{n+k_1}{2}$, in which interval non-probabilistic algorithms are valid, e.g Berlekamp-Welch algorithm [1]. This case would also be valid in the heterogeneous case if we eliminated some rows from A in Algorithm 3. We simulate this case with an example in the following pages. We also remark that the linear system (3.1) is never inconsistent without row elimination since there exists a solution of the system. Furthermore, failure probability of Algorithm 3 is less than the failure probability of the algorithm having row elimination step when any elimination occurs, i.e. when $t \neq \frac{n+\sum_{j=1}^{r} k_j}{r+1}$.

To verify the results presented in Theorem 3.1.4, we compare them with the failure probabilities obtained by Monte-Carlo simulations. In the simulations, we first generate random message polynomials. Then, corresponding codewords are obtained by the evaluations of message polynomials in distinct n field elements. Next, $n - t$ random places of the codewords are replaced with random field elements. Finally, these codewords with erroneous values are given as an input to the Algorithm 3. Simulation returns a failure if Algorithm 3 is failed. If Algorithm 3 returns a result equivalent to actual message values, then simulation returns a success. We simulated Algorithm 3 for $n = 15, r = 2, k = [2, 3], |\mathbb{F}| = 2^4$ with creating 10^7

Table 3.1: Simulated Failure Probability for the SPR problem $n = 15, k = [2, 3], |\mathbb{F}| = 2^4$, total trial = 10^7

t	$\frac{n-t}{\lvert\mathbb{F}\rvert}$	Algorithm 3	Algorithm 3′	$1 - \frac{\binom{n-x_1}{n-t}}{\binom{n}{n-t}}(1 - \frac{n-t}{\lvert\mathbb{F}\rvert})$
7	$5 \cdot 10^{-1}$	$3,91 \cdot 10^{-4}$	$6,20 \cdot 10^{-2}$	$5,00 \cdot 10^{-1}$
8	$4,37 \cdot 10^{-1}$	$9,00 \cdot 10^{-7}$	$6,12 \cdot 10^{-2}$	$4,37 \cdot 10^{-1}$
9	$3,75 \cdot 10^{-1}$	$< 10^{-7}$	$3,99 \cdot 10^{-1}$	$6,25 \cdot 10^{-1}$
10	$3,12 \cdot 10^{-1}$	$< 10^{-7}$	$7,37 \cdot 10^{-1}$	$8,18 \cdot 10^{-1}$
11	$2,50 \cdot 10^{-1}$	$< 10^{-7}$	$8,46 \cdot 10^{-1}$	$8,85 \cdot 10^{-1}$
12	$1,87 \cdot 10^{-1}$	$< 10^{-7}$	$8,77 \cdot 10^{-1}$	$9,00 \cdot 10^{-1}$
13	$1,25 \cdot 10^{-1}$	$< 10^{-7}$	$8,55 \cdot 10^{-1}$	$8,77 \cdot 10^{-1}$
14	$6,25 \cdot 10^{-2}$	$< 10^{-7}$	$7,34 \cdot 10^{-1}$	$7,50 \cdot 10^{-1}$

random error vectors of weight $n - t$ for $t = 7, 8, 9, \ldots, 14$. This is presented in Table 3.1. Simulation results show that number of failures decreases with increasing t values and they are zero for $t \geq 9$ as obtained in Theorem 3.1.4. Table 3.1 also indicates that the upper bound $\frac{n-t}{|\mathbb{F}|}$ given in Theorem 3.1.4 is a weak estimate compared to experimental results.

To verify results in Proposition 3.1.5, we first define a new algorithm called Algorithm 3′ which performs some row elimination from A as given in (3.4) then solves the corresponding system and returns the solution if it is nonsingular, otherwise returns a failure. We simulated Algorithm 3′ for $n = 14, r = 2, k = [2, 3], |\mathbb{F}| = 2^4$ with creating 10^7 random error vectors of weight $n - t$ for $t = 7, 8, 9, \ldots, 14$. Simulation results of Algorithm 3′ are also presented in Table 3.1. These results show that failure probability of Algorithm 3′ does not decrease with increasing t values. In fact, for values $t \geq 9$ there is a big increase in failure probabilities, and these are very near to the bound mentioned in Proposition 3.1.5.

We further simulated BKY algorithm [2] and Algorithm 3 for $n = 15, r = 2, k = [4, 4], |\mathbb{F}| = 2^4$ with creating 10^7 random error vectors of weight $n - t$ for $t = 8, 9, \ldots, 14$. This is presented in Table 3.2. In this case, BKY algorithm is equivalent to Algorithm 3′. The results show that failure probability of BKY algorithm does not decrease with increasing t values. In fact, for values $t \geq 10$ there is a big increase in failure probabilities, and these are very near to the bound mentioned in Proposition 3.1.5. Table 3.2 also indicates that Algorithm 3 has better decoding failure probabilities than BKY algorithm.

Table 3.2: Simulated Failure Probability for the SPR problem $n = 15, k = [4,4], |\mathbb{F}| = 2^4$, total trial $= 10^7$

t	$\frac{n-t}{\lvert\mathbb{F}\rvert}$	Algorithm 3	BKY algorithm [2]	$1 - \frac{\binom{n-x_1}{n-t}}{\binom{n}{n-t}}(1 - \frac{n-t}{\lvert\mathbb{F}\rvert})$
8	$4,37 \cdot 10^{-1}$	$9,79 \cdot 10^{-4}$	$6,17 \cdot 10^{-2}$	$4,37 \cdot 10^{-1}$
9	$3,75 \cdot 10^{-1}$	$7,00 \cdot 10^{-7}$	$6,33 \cdot 10^{-2}$	$3,75 \cdot 10^{-1}$
10	$3,12 \cdot 10^{-1}$	$< 10^{-7}$	$3,32 \cdot 10^{-1}$	$5,41 \cdot 10^{-1}$
11	$2,50 \cdot 10^{-1}$	$< 10^{-7}$	$6,35 \cdot 10^{-1}$	$7,28 \cdot 10^{-1}$
12	$1,87 \cdot 10^{-1}$	$< 10^{-7}$	$7,36 \cdot 10^{-1}$	$7,85 \cdot 10^{-1}$
13	$1,25 \cdot 10^{-1}$	$< 10^{-7}$	$7,33 \cdot 10^{-1}$	$7,66 \cdot 10^{-1}$
14	$6,25 \cdot 10^{-2}$	$< 10^{-7}$	$6,00 \cdot 10^{-1}$	$6,25 \cdot 10^{-1}$

3.2 Decoding Heterogeneous Interleaved Reed-Solomon Codes

We begin with the definition of heterogeneous interleaved Reed-Solomon (IRS) codes.

Definition 3.2.1 *Let $z_1, \ldots, z_n \in \mathbb{F}$ be pairwise distinct and $C_l = RS(n, k_l) = \{(p_l(z_1), \ldots, p_l(z_n))^T | p_l(x) \in \mathbb{F}[x]$ of degree less than $k_l\}$, $l = 1, \ldots, r$ be r Reed-Solomon codes of length n over $\mathbb{F}$. Then, an interleaved Reed-Solomon code $C = IRS(n; k_1, \ldots, k_r)$ is the set of matrices*

$$C = \{(c_1, c_2, \ldots, c_r), c_l = (c_{l,1}, \ldots, c_{l,n})^T \in C_l, l = 1, \ldots, r\}.$$

Schmidt, Sidorenko and Bossert [32] present a probabilistic algorithm using Berlekamp-Massey approach to decode heterogeneous IRS codes for burst errors, i.e. for errors occurring at the same place of the subcodes simultaneously. In this section, we use probabilistic solution of SPR problem based on Berlekamp-Welch approach presented in the previous section which also decodes IRS codes for burst errors.

IRS Decoder:

Parameters: $n, k_1, \ldots, k_r, t \in \mathbb{N}$ and distinct $z_1, \ldots, z_n \in \mathbb{F}$ satisfying
$\max\{k_1, k_2, \ldots, k_r, \left\lceil \frac{n+\sum_{j=1}^{r} k_j}{r+1} \right\rceil\} \leq t \leq n \leq |\mathbb{F}|$.
Input: the received codeword $\langle c_{i,1}, \ldots, c_{i,r} \rangle_{i=1}^{n}$
Assumption: messages and error values are random.
Algorithm: Algorithm 3 with input $\langle c_{i,1}, \ldots, c_{i,r} \rangle_{i=1}^{n}$, $n, k_1, \ldots, k_r$ and $z_1, \ldots, z_n \in \mathbb{F}$, and by trying t from $\max\{k_1, k_2, \ldots, k_r, \left\lceil \frac{n+\sum_{j=1}^{r} k_j}{r+1} \right\rceil\}$ to n until simulating on the correct one decodes IRS$(n; k_1, \ldots, k_r)$ code probabilistically having at most $n - t$ burst errors.

Theorem 3.2.2 *Assume that messages and error values are random.*

i. Interleaved Reed-Solomon codes $IRS(n; k_1, \dots, k_r)$ *over* $\mathbb{F}$ *satisfying*

$$\left\lceil \frac{n + \sum_{j=1}^{r} k_j}{r+1} \right\rceil \geq max\{k_1, k_2, \dots, k_r\}$$

and having rate $R = \frac{\sum_{l=1}^{r} k_l}{nr}$ *can be corrected with the IRS Decoder up to burst error rate* ϵ*, where*

$$\epsilon < \frac{r}{r+1}(1 - R) \tag{3.5}$$

with probability at least $1 - \frac{r}{r+1}\frac{(1-R)n}{|\mathbb{F}|}$.

ii. Interleaved Reed-Solomon codes $IRS(n; k_1, \dots, k_r)$ *over* $\mathbb{F}$ *satisfying*

$$\left\lceil \frac{n + \sum_{j=1}^{r} k_j}{r+1} \right\rceil < max\{k_1, k_2, \dots, k_r\}$$

with $R_{max} = max\{k_1, k_2, \dots, k_r\}/n$ *can be corrected with the IRS Decoder up to burst error rate* ϵ*, where*

$$\epsilon \leq (1 - R_{max}) \tag{3.6}$$

with probability at least $1 - \frac{(1-R_{max})n}{|\mathbb{F}|}$.

Complexity of the IRS Decoder is at most $O(n(nr)^3)$ *field operations.*

Proof. For the first part of the theorem, we observe that

$$\epsilon n \quad = \quad n - t < n - \frac{n + \sum_{l=1}^{r} k_l}{r+1} = n\frac{r}{r+1}(1 - R)$$

and

$$\frac{n-t}{|\mathbb{F}|} < n\frac{r}{r+1}\frac{(1-R)}{|\mathbb{F}|}.$$

For the second part, we similarly observe that

$$\epsilon n \quad = \quad n - t \leq n - \max\{k_1, k_2, \dots, k_r\} = n(1 - R_{max})$$

and

$$\frac{n-t}{|\mathbb{F}|} \leq \frac{(1 - R_{max})n}{|\mathbb{F}|}.$$

Hence, theorem directly follows from Theorem 3.1.4. ■

Remark 3.2.3 *We need to remark some cases that are possible during the IRS decoder. Let weight of an error vector be* $n - a_0$*. Then, the corresponding linear system has never full rank for* $t < a_0$*. We proved that the system does not have full rank for* $t = a_0$ *with a low probability. Assume that the system does not have full rank for* a_0*. Then, the linear system may be full rank for some* $t > a_0$ *but it is easy to see that the linear system can never be consistent. In theoretical calculation of the failure probability, we count all these cases as failure in the worst case. If one uses Algorithm 3′ in IRS decoder, then in the case that the corresponding linear system does not have full rank for* a_0*, the linear system may be full rank for some* $t > a_0$ *but* m_l *may not be divisible by* p_l *for some* $l \in [r]$ *or the linear system may be full rank for some* $t > a_0$ *and* m_l *may be divisible by* p_l *for all* $l \in [r]$ *but* $\hat{p}_l$ *may be different than* p_l *for some* $l \in [r]$*. Also, it is possible that at the end it may be the case that* $\hat{p}_l = p_l$ *for all* $l \in [r]$*. Therefore, it is possible to get decoding error when Algorithm 3′ is used in IRS decoder instead of Algorithm 3.*

In this section, we generally consider the IRS$(n; k_1, \dots, k_r)$ codes over $\mathbb{F}$ satisfying $\left\lceil \frac{n+\sum_{j=1}^{r} k_j}{r+1} \right\rceil \geq \max\{k_1, k_2, \dots, k_r\}$. In this case, one can decode IRS codes using IRS decoder for any error rate arbitrarily close to the bound $1 - R$ with Algorithm 3 for a given R and for a desired failure probability at most η as follows: first choose $\epsilon < 1 - R$ and choose r so that (3.5) is satisfied, that is choose $r \geq \left\lceil \frac{\epsilon}{1-R-\epsilon} \right\rceil$. Then select $\mathbb{F}$ and $n < \frac{\eta|\mathbb{F}|(r+1)}{(1-R)r}$ such that nR is an integer so that the failure probability of Theorem 3.2.2 becomes at most η and random selection of $k_1, \dots, k_r \leq \left\lceil \frac{n+\sum_{j=1}^{r} k_j}{r+1} \right\rceil$ satisfying $R = \frac{\sum_{l=1}^{r} k_l}{nr}$ is possible.

Now, we give some examples presenting the results obtained in Theorem 3.2.2. In addition, we present experimental results which are obtained by simulating the corresponding examples. These results are also compared in Table 3.3.

Example 3.2.4 *IRS Decoder decodes IRS*$(15; 6, 5, 4)$ *code over* $\mathbb{F}_{2^4}$ *with failure probability at most* $\eta = 4{,}375 \cdot 10^{-1}$ *up to 7 errors where rate is* $R = \frac{1}{3}$*. If we were over* $\mathbb{F}_{2^{10}}$*, then we would reach to same error correction capability with the failure probability at most* $6{,}8 \cdot 10^{-3}$*. On the other hand, it is experimentally obtained that failure probability of decoding IRS*$(15; 6, 5, 4)$ *code over* $\mathbb{F}_{2^4}$ *for at most 7 errors is approximately* $2{,}61 \cdot 10^{-3}$.

Example 3.2.5 *IRS Decoder decodes homogeneous IRS*$(255; 223, 223, 233)$ *code over* $\mathbb{F}_{2^8}$ *with failure probability at most* $\eta = 9{,}375 \cdot 10^{-2}$ *up to 24 errors. On the other hand, it* *is experimentally obtained that failure probability of decoding IRS*$(255; 223, 223, 223)$ *code over* $\mathbb{F}_{2^8}$ *for at most 24 errors is approximately* $3,70 \cdot 10^{-3}$.

Example 3.2.6 *One can decode for errors close to bound* $1 - R$ *with Algorithm 3 for a given* $R = 0,7$ *and for a desired fail probability at most* $\eta = 10^{-3}$ *as follows: first choose* $\epsilon = 0,2$ *and choose* $r = \left\lceil \frac{\epsilon}{1-R-\epsilon} \right\rceil = 2$. *Then select* $\mathbb{F} = \mathbb{F}_{2^{12}}$ *and* $n = 20 < \frac{\eta|\mathbb{F}|(r+1)}{(1-R)r}$. *Finally, choose randomly* $k_1 = 13, k_2 = 15 \leq 16$ *satisfying* $R = \frac{\sum_{l=1}^{r} k_l}{nr}$. *In other words, IRS*$(20; 13, 15)$ *over* $\mathbb{F}_{2^{12}}$ *can be decoded up to 4 errors for a failure probability at most* 10^{-3}. *Experimental results show that IRS*$(20; 13, 15)$ *code over* $\mathbb{F}_{2^{12}}$ *for at most 4 errors is decoded with a failure probability approximately* $2,35 \cdot 10^{-4}$

Remark 3.2.7 *Experimental results given in the above examples are obtained by simulating IRS Decoder for* 10^7 *random message polynomials with random errors in the random places. All errors in the experimental results have the most possible weight in order to compare efficiently with theoretical results and to calculate the worst case probability. For instance, all errors have weight 7 for IRS*$(15; 6, 5, 4)$ *code during the simulation. On the other hand, we obtain that failure probability decreases when errors have random weight. For instance, the failure probability of decoding IRS*$(15; 6, 5, 4)$ *code defined over* $\mathbb{F}_{2^4}$ *decreases to* $3,3 \cdot 10^{-5}$ *if errors have weight between 0 and 7 randomly.*

Comparison of results of decoding IRS$(n; k_1, k_2, \ldots, k_r)$ codes over $\mathbb{F}$ up to $n - t$ burst errors with an estimated failure probability $\eta = \frac{n-t}{|\mathbb{F}|}$ according to Theorem 3.2.2 and experimental failure probability P_f of IRS decoder based on Algorithm 3 is presented in Table 3.3.

Table 3.3: Simulated Failure Probability for Interleaved Reed-Solomon Codes

Code	$\lvert\mathbb{F}\rvert$	$n - t$	η	P_f
IRS$(15; 6, 5, 4)$	2^4	7	$4,375 \cdot 10^{-1}$	$2,61 \cdot 10^{-3}$
IRS$(15; 6, 5, 4)$	2^{10}	7	$6,8 \cdot 10^{-3}$	$< 10^{-7}$
IRS$(255; 223, 223, 223)$	2^8	24	$9,375 \cdot 10^{-2}$	$3,70 \cdot 10^{-3}$
IRS$(20; 13, 15)$	2^{12}	4	10^{-3}	$2,35 \cdot 10^{-4}$

In Section 3.3 and Section 5.2, we will use the decoder of heterogeneous IRS codes presented in this section for decoding RS codes and folded Hermitian codes, respectively.

3.3 Decoding RS Codes with Heterogeneous IRS Decoder

Schmidt, Sidorenko and Bossert [31] observed that decoding RS codes by using decoding heterogeneous IRS codes is possible. In this section, we start with this observation, and then we apply the IRS decoder presented in Section 3.2 for decoding RS codes.

Let $p(x) = p_0 + p_1 x + \cdots + p_{n-1}x^{n-1}$ be a polynomial over $\mathbb{F}$. Denote by $p^{<i>}(x)$ the polynomial $p^{<i>}(x) = p_0^i + p_1^i x + \cdots + p_{n-1}^i x^{n-1}$. Consider the RS Code $\mathbf{C} = \mathrm{RS}(n; k)$ over $\mathbb{F}$, and define the codes

$$C^{<i>} = \{c^{<i>}(x) : c(x) \in \mathbf{C}\}, \; i = 1, \ldots, r.$$

Lemma 3.3.1 *[31] If $i(k-1)+1 \leq n$, then $C^{<i>} \subset C_i$, where $C_i = RS(n; i(k-1)+1)$ is a Reed-Solomon code of dimension $i(k-1)+1$ over $\mathbb{F}$.*

Now we select an integer r, such that $r(k-1)+1 \leq n$. For every codeword $c(x) \in \mathbf{C}$, we create r codewords $c^{<i>}(x) \in C_i$, and the corresponding vectors $c^{<i>} = (c_0^i, c_1^i, \ldots, c_{n-1}^i)$ for $i = 1, \ldots, r$. From these vectors, we create the matrix

$$\mathbf{c} = \begin{pmatrix} c^{<1>} \\ c^{<2>} \\ \vdots \\ c^{<r>} \end{pmatrix}^T = \begin{pmatrix} c_0 & c_1 & \cdots & c_{n-1} \\ c_0^2 & c_1^2 & \cdots & c_{n-1}^2 \\ \vdots & \vdots & & \vdots \\ c_0^r & c_1^r & \cdots & c_{n-1}^r \end{pmatrix}^T .$$

This matrix can be considered as a codeword of an IRS code, obtained by interleaving the codes $C_1, \ldots, C_r$ with varying redundancy.

Now, assume that we transmit $c(x) \in \mathbf{C}$ over a channel, and observe the corrupted word $y(x) = c(x) + e(x)$ at the channel output. From this received word, we calculate the polynomials $y^{<i>}(x) = y_0^i + y_1^i x + \cdots + y_{n-1}^i x^{n-1}$, $i = 1, \ldots, r$, where

$$y_j^i = (c_j + e_j)^i = c_j^i + e_j^{[i]}.$$

We observe that if $e_j = 0$, then $e_j^{[i]} = 0$ for all i. From this we conclude that the locations of the erroneous symbols in $y(x)$ are not modified by raising the coefficients to the i-th power. Therefore, we observed that errors of $\mathbf{c}$ are burst errors. Hence, one can decode RS codes by using the heterogeneous IRS Decoder given in the previous section.

If the IRS Decoder is applied to decode RS codes, then one gets a probabilistic decoding algorithm with the result:

Theorem 3.3.2 *Assume that messages and error values are random. Reed-Solomon codes $RS(n;k)$ over $\mathbb{F}$ can be corrected up to error rate ϵ, where*

$$\epsilon < \frac{r}{r+1}(1-R) \tag{3.7}$$

with probability at least $1 - \frac{r}{r+1}\frac{(1-R)n}{|\mathbb{F}|}$ for parameters r satisfying

$$r \leq \left\lfloor \sqrt{2\frac{n-1}{k-1} + \frac{1}{4}} - \frac{1}{2} \right\rfloor$$

and $R = \frac{(k-1)(r+1)+2}{2n}$. Moreover, $RS(n;k)$ over $\mathbb{F}$ can be corrected up to error rate ϵ, where

$$\epsilon \leq 1 - \frac{r(k-1)+1}{n} \tag{3.8}$$

with probability at least $1 - \frac{n-r(k-1)-1}{|\mathbb{F}|}$ for r values satisfying

$$\left\lfloor \sqrt{2\frac{n-1}{k-1} + \frac{1}{4}} - \frac{1}{2} \right\rfloor < r \leq \left\lfloor \frac{n-1}{k-1} \right\rfloor. \tag{3.9}$$

Complexity of the decoding procedure is at most $O(n(nr)^3)$ field operations.

Proof. We first calculate the rate of the interleaved code $\mathrm{IRS}(n;k,2(k-1)+1,\ldots,r(k-1)+1)$ reduced from $\mathrm{RS}(n;k)$ as follows

$$R = \frac{\sum_{l=1}^{r}(l(k-1)+1)}{nr} = \frac{(k-1)(r+1)+2}{2n}.$$

Then, we compare the maximum dimension

$$\max\{k, 2(k-1)+1, \ldots, r(k-1)+1\} = r(k-1)+1 \tag{3.10}$$

with the integer

$$\left\lceil \frac{n + \sum_{i=1}^{r}(i(k-1)+1)}{r+1} \right\rceil. \tag{3.11}$$

We observe that (3.11) is greater than (3.10) for $r \leq \left\lfloor \sqrt{2\frac{n-1}{k-1} + \frac{1}{4}} - \frac{1}{2} \right\rfloor$. Therefore, first part of the theorem is a consequence of Lemma 3.3.1 and Theorem 3.2.2.(i).

According to Lemma 3.3.1, we can also choose r between $\left\lfloor \sqrt{2\frac{n-1}{k-1} + \frac{1}{4}} - \frac{1}{2} \right\rfloor$ and $\left\lfloor \frac{n-1}{k-1} \right\rfloor$ in which case even RS codes can be reduced to IRS codes. In this case, (3.10) is greater than

(3.11). Thus, Theorem 3.2.2.(ii) implies that IRS decoder corrects RS(n,k) codes up to error rate $\frac{n-r(k-1)-1}{n}$ with probability at least $1 - \frac{n-r(k-1)-1}{|\mathbb{F}|}$. And, this proves the later part of the theorem. ■

Now, we compare decoding radii (3.7) and (3.8) with the radius of bounded minimum distance (BMD) decoders such as Berlekamp-Welch algorithm [1]. Radius of a BMD decoder is equal to

$$(1 - R)/2. \tag{3.12}$$

Proposition 3.3.3 *For a given interleaving parameter r, decoding radius (3.7) is better than radius* $(1 - R)/2$ *for RS codes having rates* $R \leq 1/\binom{r+1}{2}$.

Proof. (3.7) is better than $(1 - R)/2$ if

$$\frac{r}{r+1}(1 - \frac{(k-1)(r+1)+2}{2n}) > \frac{1}{2}(1 - \frac{k}{n})$$

is satisfied. So, r satisfies

$$(k-1)r^2 - (n-1)r + (n-k) < 0$$

that is

$$r \in (1, \frac{n-k}{k-1}).$$

On the other hand, we observe that

$$\left\lfloor \sqrt{2\frac{n-1}{k-1} + \frac{1}{4}} - \frac{1}{2} \right\rfloor \leq \left\lfloor \frac{n-k}{k-1} \right\rfloor$$

for all n and $k \leq n$. Hence, we have a better decoding radius than (3.12) for each r satisfying

$$r \leq \left\lfloor \sqrt{2\frac{n-1}{k-1} + \frac{1}{4}} - \frac{1}{2} \right\rfloor.$$

This implies that the rate of RS code is required to satisfy

$$R \leq \frac{2k}{(k-1)(r^2+r)+2}$$

for a given r, or in the worst case we say that $R \leq 1/\binom{r+1}{2}$. ■

For instance, for $r = 2$:

$$\frac{2}{2+1}(1 - \frac{(k-1)(2+1)+2}{2n}) = \frac{2}{3} - \frac{k}{n} + \frac{1}{3n}$$

is better than

$$\frac{1}{2} - \frac{k}{2n}$$

for rates at most $\frac{1}{3}$.

Similarly, for $r = 3$ we have better decoding radius than (3.12) if the rate is at most $\frac{1}{6}$, and for $r = 4$ if rate is at most $\frac{1}{10}$.

Proposition 3.3.4 *For a given interleaving parameter r, decoding radius (3.8) is better than radius* $(1 - R)/2$ *for RS codes having rates* $1/\binom{r+1}{2} \leq R \leq 1/(2r - 1)$.

Proof. (3.8) is better than $(1 - R)/2$ if

$$(1 - \frac{r(k-1)+1}{n}) > \frac{1}{2}(1 - \frac{k}{n})$$

is satisfied. So, r satisfies

$$r < \frac{n+k-2}{2(k-1)}.$$

Thus, for a given r, rate of the code is required to satisfy

$$r < \frac{1 + R - \frac{2}{n}}{2(R-1)}$$

or equivalently, $R \leq 1/(2r - 1)$. On the other hand, we know that r is also required to satisfy

$$r \geq \left\lfloor \sqrt{2\frac{n-1}{k-1} + \frac{1}{4}} - \frac{1}{2} \right\rfloor.$$

Then, by Proposition 3.3.3 rate is to be at least $1/\binom{r+1}{2}$. Therefore, $1/\binom{r+1}{2} \leq R \leq 1/(2r - 1)$. As a final step, it is easy to check that $\left\lfloor \frac{n+k-2}{2(k-1)} \right\rfloor$ is in the range (3.9). This ends proof of the theorem. ■

For instance, for $r = 2$:

$$1 - \frac{2(k-1)+1}{n}$$

is better than

$$\frac{1}{2} - \frac{k}{2n}$$

for rates at most $\frac{1}{3}$.

Similarly, for $r = 3$ we have better decoding radius than (3.12) if rate is at most $\frac{1}{5}$, and for $r = 4$ if rate is at most $\frac{1}{7}$.

Now, we consider the case $r \leq \left\lfloor \sqrt{2\frac{n-1}{k-1} + \frac{1}{4}} - \frac{1}{2} \right\rfloor$, and observe some results. Similar observations can be done for other r values.

A given RS$(n; k)$ code over $\mathbb{F}$ can be corrected better than bounded minimum distance decoder with the decoder based on Algorithm 3. To maximize the difference choose $r = \left\lceil \frac{1-k+\sqrt{2}\sqrt{1-k-n+kn}}{-1+k} \right\rceil$ which is the positive zero of derivative of the function defined as the difference of radius (3.7) and unique decoding radius, then errors up to rate $\epsilon \leq \frac{r}{r+1}(1 - R)$ can be corrected with probability at least $1 - \frac{r}{r+1}\frac{(1-R)n}{|\mathbb{F}|}$ where $R = \frac{(k-1)(r+1)+2}{2n}$.

On the other hand, one can decode RS codes with Algorithm 3 for a given rate β and fail probability at most η as follows: first choose r satisfying $\beta \leq \frac{1}{\binom{r+1}{2}}$ and set $R = \frac{\beta(r+1)}{2}$. Next, choose error rate ϵ as $\epsilon \leq \frac{r}{r+1}(1-R)$. Then, choose $\mathbb{F}$ and $n < \frac{\eta|\mathbb{F}|(r+1)}{(1-R)r}$ such that $n\beta$ is an integer so that fail probability is at most η and dimension of the Reed-Solomon code is meaningful.

To support results obtained in this section, an example with experimental results observed by simulating IRS decoder for RS codes is given below. These results are also compared in Table 3.4.

Example 3.3.5 *IRS Decoder decodes RS*$(31; 6)$ *code over* $\mathbb{F}_{2^8}$ *with failure probability at most* $\eta = 5,86 \cdot 10^{-2}$ *up to error rate* $\epsilon \leq 15/31$ *by choosing* $r = 2$*. If we were over* $\mathbb{F}_{2^5}$*, then failure probability increases to* $4,688 \cdot 10^{-1}$*. However, experimentally we obtained that RS*$(31; 6)$ *over* F_{2^5} *is decoded for up to 15 errors with failure probability approximately* $3,08 \cdot 10^{-2}$*. Similarly, experimental results show that RS*$(31; 4)$ *over* $\mathbb{F}_{2^5}$ *is decoded for up to 18 errors with failure probability approximately* $3,20 \cdot 10^{-2}$*. These experimental results are very near to ones given in [31]. On the other hand, for a given rate* $\beta = 1/5$ *and failure probability at most* 10^{-2}*, IRS Decoder decodes for any error rate* $\epsilon \leq 9/20$ *by choosing* $r = 2$, $\mathbb{F} = \mathbb{F}_{2^{10}}$ *and* $n = 20$*. In other words, RS*$(20; 4)$ *over* $\mathbb{F}_{2^{10}}$ *can be decoded for at most 9 errors with failure probability at most* 10^{-2}*. However, experimental results show that it is less than* 10^{-7}.

Comparison of results of decoding RS$(n; k)$ over $\mathbb{F}$ up to $n - t$ errors with an estimated failure probability $\eta \approx \frac{n-t}{|\mathbb{F}|}$ according to Theorem 3.3.2 and experimental failure probability P_f of RS decoder based on Algorithm 3 is presented in Table 3.4. Experimental results are obtained by simulating IRS Decoder for 10^7 random message polynomials with random errors in the random places. All errors in the simulation results have the most possible weight in order to compare efficiently with theoretical results and to observe the worst case probability. Capability of number of errors corrected by bounded minimum distance (BMD) decoders are also presented in Table 3.4. We note that BMD decoders are capable of decoding RS$(n; k)$ code for all error patterns up to weight $\left\lfloor \frac{n-k}{2} \right\rfloor$.

Table 3.4: Simulated Failure Probability for Reed-Solomon Codes

Code	$\lvert\mathbb{F}\rvert$	$n-t$	$\left\lfloor \frac{n-k}{2} \right\rfloor$	η	P_f
RS(31; 6)	2^5	15	12	$4,688 \cdot 10^{-1}$	$3,08 \cdot 10^{-2}$
RS(31; 6)	2^8	15	12	$5,86 \cdot 10^{-2}$	$3,92 \cdot 10^{-3}$
RS(31; 4)	2^5	18	13	$5,625 \cdot 10^{-1}$	$3,20 \cdot 10^{-2}$
RS(31; 4)	2^8	18	13	$7,03 \cdot 10^{-2}$	$3,97 \cdot 10^{-3}$
RS(20; 4)	2^{10}	9	8	$8,8 \cdot 10^{-3}$	$< 10^{-7}$

CHAPTER 4

LIST DECODING INTERLEAVED CODES

A list-decoding algorithm is an algorithm which tries to construct a list which includes codewords within a specified radius of an input codeword. The central problem of list decoding is to identify the radius up to which a list decoding algorithm exists in terms of the output list size and efficiency of the algorithm.

List decoding was first introduced by Elias [6] and Wozencraft [37]. In [35], Sudan proposed first non-exponential time list-decoding algorithm for Reed-Solomon (RS) codes. The list-decoding algorithm in [35] works only for codes of low rates. Guruswami and Sudan [11] later proposed improved list-decoding algorithm (GS algorithm) for RS-codes. The algorithm of Guruswami and Sudan has the largest list decoding radius for RS-codes up to date, and are applicable to codes of any rates.

Recently, by specializing the ideas of Pararesh and Vardy [20] of construction of correlated RS-codes, Guruswami and Rudra [12] constructed folded RS-codes, which are obtained as folding each b symbols of RS-code together representing these consecutive b symbols as a new symbol. Guruswami and Rudra showed that folded RS-codes achieve the information theoretically best possible trade off between the rate and error-correction radius. They presented a list decoding algorithm (GR algorithm) for folded RS-codes that decodes up to information theoretically best possible radius (called capacity) and runs in polynomial time, and also outputs a list of codewords whose size is also polynomial.

It is known that folded RS-codes are subcodes of interleaved RS-codes, and there is also a study on interleaved RS-codes. Gopalan et al.[8] showed that homogeneous interleaved codes (possibly nonlinear) can be list decodable up to radius what subcode can be. In particular, this implies that interleaved folded RS-codes can also be list decodable up to the information theoretically best possible radius. Moreover, they presented an efficient algorithm for list decoding interleaved codes.

In this chapter, we will continue in this direction, and we show that heterogeneous interleaved codes, whose minimum distances would be distinct, can be list decodable up to radius of minimum of list decoding radii of subcodes in Section 4.2.

4.1 Previous Results

Guruswami and Sudan [11] presented a list decoding algorithm for RS-codes with the following result. See also Corollary 4.9 in [14].

Theorem 4.1.1 *[11] For every $0 < \epsilon$, a Reed-Solomon code of rate R and length n can be list decoded in polynomial time $O(n^3)$ up to a fraction $1 - \sqrt{(1+\epsilon)R}$ of errors using lists of size $O(\epsilon^{-1}/\sqrt{R})$.*

Later, Guruswami and Rudra [12] constructed folded Reed-Solomon codes having the following result.

Theorem 4.1.2 *[12] For every $0 < \epsilon$ and $0 < R < 1$, there is a family of folded Reed-Solomon codes that have rate at least R and which can be list decoded up to a fraction $1-R-\epsilon$ of errors in time (and outputs a list of size at most) $(N/\epsilon^2)^{O(\epsilon^{-1}log_2(1/R))}$ where N is the block length of the code. The alphabet size of the code as a function of the block length N is $(N/\epsilon^2)^{O(1/\epsilon^2)}$ and the folding parameter of the code is approximately $O(1/\epsilon^2)$.*

For interleaved codes, Gopalan et al.[8] obtained the following result.

Theorem 4.1.3 *[8] Let C be interleaving of b-wise c of length n having minimum distance δ. For $\eta < \delta$, let $l(\eta)$ be the list size of c when list decoded up to radius η with time complexity T. And, let $x = \left\lceil \frac{\eta}{\delta-\eta} \right\rceil$ and $y = \left\lceil log_2 \frac{\delta}{\delta-\eta} \right\rceil$. Then C can be list decodable up to radius η with list size $L(\eta) < \binom{x+y}{y} l(\eta)^y$ and time complexity at most $O(bT + b^2 nl(\eta)L(\eta))$.*

4.2 List Decoding Interleaved Codes

We formally define interleaved codes as follows.

Definition 4.2.1 *The interleaving C of the codes $C_1, \dots, C_b \subset \mathbb{F}_q^n$ consists of $n \times b$ matrices over $\mathbb{F}_q$ whose i-th column is a codeword in C_i for $i = 1, \dots, b$. Each row is treated as a single symbol, thus $C \subset \mathbb{F}_{q^b}^n$.*

Throughout the section, $C_i \subset \mathbb{F}_q^n$ will be arbitrary codes (possibly non-linear) over an alphabet $\mathbb{F}_q$ with relative distance δ_i for $i = 1, \dots, b$. And, let C be interleaving of $C_1, \dots, C_b$. We will show that ideas of [8] are also applicable if each C_i also has distinct relative distance (heterogeneous case). But, list decoding radius weakens a little bit. In this section we follow the proof technique of [8] with little changes.

Let $d_q(r, c)$ denote the Hamming distance between strings in $\mathbb{F}_q^n$ and $\Delta_q(r, c) = d_q(r, c)/n$ denote the normalized Hamming distance.

By using algorithms DecodeC_i that can list decode C_i up to radius η with a list L_i whose size is $l_i(\eta)$ for $i = 1, \dots, b$, DecodeC given in Algorithm 4 list decodes C up to radius η in time polynomial in the list-size and b. Thus it is required to bound the list-size. In order to do this, an inefficient algorithm (Algorithm 5) will be given. Let $r_{\leq i} = (r_1, ..., r_i)$, $L_{\leq i}$ direct product of $L_1, L_2 \dots, L_i$ and $l(\eta) = max_i\{l_i(\eta)\}$ for $i \leq b$.

Proposition 4.2.2 *Assume that DecodeC$_i(r_i, \eta)$ runs in time T_i for $i = 1, 2, \dots, b$. Then Algorithm 4:DecodeC(r, η) returns a list of codewords within distance η of r in time at most $O(\sum_i T_i + b^2 n l(\eta) L(\eta))$.*

Proof. If $\Delta_{q^b}(c, r) \leq \eta$ for any c, then for every $i \leq b$, we have $\Delta_{q^i}(c_{\leq i}, r_{\leq i}) \leq \eta$, that is, $c_{\leq i} \in L_{\leq i}$. Conversely, if $c \in L_{\leq b}$, then we have $\Delta_{q^b}(c, r) \leq \eta$. Therefore, $c \in L_{\leq b}$ if and only if $\Delta_{q^b}(c, r) \leq \eta$, which shows that Algorithm 4 outputs the correct codewords.

Steps 1-3 take time at most $O(\sum_i T_i)$. And, each iteration of the loop in Step 7 requires computing the distance between r and c at most $l_i(\eta) L_{\leq i}(\eta) < l(\eta) L(\eta)$ candidates for c. And, computing the distance takes at most $O(bn)$ operations over $\mathbb{F}_q$. Therefore, Steps 5-9 take at most $O(b^2 n l(\eta) L(\eta))$ operations. ■

Algorithm 4 DecodeC: An Efficient List Decoding Algorithm for C

```
Require: $r = (r_1, ..., r_b) \in (\mathbb{F}_q^n)^b$, $\eta$.
Ensure: List of all $c \in C$ so that $\Delta_{q^b}(r, c) \le \eta$.
for $i = 1, \dots, b$ do
    Set $L_i = DecodeC_i(r_i, \eta)$.
end for
Set $L_{\le 1} = L_1$.
for $i = 2, \dots, b$ do
    for $c \in L_{\le i-1} \times L_i$ do
        Add $c$ to $L_{\le i}$ if $\Delta_{q^i}(c, r_{\le i}) \le \eta$.
    end for
end for
return $L_{\le b}$.
```

Now, we give Algorithm 5 in the following pages to bound the output list size of the Algorithm 4.

Definition 4.2.3 *Given a code $C \subset \mathbb{F}_q^n$, erasing the indices corresponding to $S \subset [n]$ gives the code $C^{-S} = \{c^{-S} : c \in C\} \subseteq \mathbb{F}_q^{n-|S|}$.*

Let $|S| = \mu n$. We will only consider the case that $\mu < \delta := min_i\{\delta_i\}$. It is easy to see that the code C_i^{-S} has distance $d(C_i^{-S}) > (\delta_i - \mu)n$, and there is a 1-1 correspondence between codewords in C_i and their projections in C_i^{-S}. For $\eta < 1 - \mu$, let $l_i^{-S}(\eta)$ be the maximum number of codewords of C_i^{-S} that lie in a Hamming ball of radius ηn in $(\mathbb{F}_q^b)^{n(1-\mu)}$. Then following result shows the relation between l_i^{-S} and l_i.

Lemma 4.2.4 *For any $\eta < 1 - \mu$ and $1 \le i \le b$, $l_i^{-S}(\eta) \le l_i(\eta + \mu)$.*

Proof. For any received word $r_i^{-S} \in \mathbb{F}_q^{n(1-\mu)}$, let $c_{i1}^{-S}, ..., c_{il_i^{-S}(\eta)}^{-S}$ be codewords satisfying $d(r_i^{-S}, c_{ij}^{-S}) \le \eta n$. One can define $r_i \in \mathbb{F}_q^n$ by fixing values of r_i^{-S} at the set S arbitrarily. By the triangle inequality, it is known that $d(r_i, c_{ij}) \le d(r_i^{-S}, c_{ij}^{-S}) + |S| \le \eta n + \mu n$, which implies that $l_i(\eta + \mu) > l_i^{-S}(\eta)$. ■

Algorithm 5 is a list decoding algorithm having inputs $S \in [n]$, $r_i \in \mathbb{F}_q^n$ and an error parameter e_i, and returning all codewords $c_i \in C_i$ so that $d(c_i^{-S}, r_i^{-S}) \le e_i$. This algorithm is not efficient, but it is not important since we only use it for estimating the upper bound of the list size of Algorithm 4. Let $c_{\leq i} = (c_1, ..., c_i)$. Algorithm 5 erases the set of positions S where $c_{\leq i} \neq r_{\leq i}$ and then runs a list decoder for C_{i+1}^{-S} on r_{i+1}.

Algorithm 5 Erase-Decode

```
Require: $r \in \mathbb{F}_{q^b}^n, \eta$.
Ensure: List $L$ of all $c \in C$ so that $\Delta_{q^b}(r, c) \leq \eta$.
Set $S_1 = \emptyset, \mu_1 = 0$.
for $i = 1, \ldots, b$ do
    Set $L_i$ = List-Decode($S_i, r_i, (\eta - \mu_i)n$).
    Choose $c_i \in L_i$.
    Set $S_{i+1} = \{j \in [n]$ s.t. $c_{\leq i}[j] \neq r_{\leq i}[j]\}$ and $\mu_{i+1} = |S_{i+1}|/n$.
end for
return $c = (c_1, \ldots, c_b)$.
```

During Erase-Decode algorithm, for each $c_1 \in L_1$ distinct places S_2 of c_1 and r_1 are removed from r_2 and obtained $r_2^{-S_2}$. Then it is list decoded and we obtain L_2. The same procedure is applied for each $c_2 \in L_2$. At the end, if we reach to the b-th stage on some path, then the codeword $(c_1, c_2, \ldots, c_b)$ of this path is included in L. Each $c_i \in L_i$ leads to a different path after itself if the list decoding at the next stage produces any output. So the whole procedure of Erase-Decode can be realized as a tree. As a remark, it is known that there is a 1-1 correspondence between C_i and C_i^{-S}. Thus the size of L gives a bound for the size of list produced by Algorithm 4.

For a received word r, Tree(r) is a tree with $b + 1$ levels. The root is at level 0. A node v at level i is labeled by $c(v) = (c_1, ..., c_i)$. It is associated with a set $S_{i+1}(v) \subseteq [n]$ of erasures accumulated so far which has size $\mu_{i+1}(v)n$. The resulting code $C_{i+1}^{-S_{i+1}(v)}$ has minimum distance $\delta_{i+1}(v)n > (\delta - \mu_{i+1}(v))n$. List decoding algorithm in Step 3 finds all codewords in $C_{i+1}^{-S_{i+1}(v)}$ that are within distance $(\eta - \mu_{i+1}(v))n$ of the modified received word $r_{i+1}^{-S_{i+1}(v)}$, call this list $L(v)$. By Lemma 4.2.4, $L(v)$ contains at most $l_{i+1}(\eta)$ codewords which is smaller than $l(\eta) := max_i\{l_i(\eta)\}$. Each edge leaving v is labeled by a distinct codeword c_{i+1} from $L(v)$; it is assigned a weight $w(c_{i+1}) = d(c_{i+1}^{-S(v)}, r_{i+1}^{-S(v)})/n$. The weight $w(c_{i+1}) \in [0, 1]$ of an edge indicates how many new erasures that edge contributes. Thus $\mu_{i+1}(v) = w(c_1) + \ldots + w(c_i)$. The leaves at level b correspond to codewords in the list L. There might be no out-edges from v if the list $L(v)$ is empty. This could result in a leaf node at a level $i < b$ which does not correspond to codewords. Thus the number of leaves in Tree(r) is an upper bound on the list-size for r.

In order to bound the number of leaves, we assign colors to the various edges based on their weights. Let c_i be an edge leaving the vertex v. We color it White if $w(c_i) < \delta - \eta$, Blue if $w(c_i) > \delta - \eta$ but $w(c_i) < \delta_i(v)/2$, and Red if $w(c_i) > \delta_i(v)/2$. White edges correspond to codewords that are very close to the received word, Blue edges to codewords that are within the unique-decoding radius, and Red edges to codewords beyond the unique decoding radius.

We begin by observing that White edges can only occur if the list is of size 1.

Lemma 4.2.5 *If a vertex v has a White out-edge, then it has no other out-edges.*

Proof. Assume that the edge labeled with $c_i \in L(v)$ is colored White, so that $d(c_i, r_i^{-S_i(v)}) < (\delta - \eta)n$. Let c_i' be another codeword in $L(v)$, so that $d(c_i', r_i^{-S_i(v)}) \leq (\eta - \mu_i(v))n$. Then by the triangle inequality,

$$d(c_i, c_i') < (\delta - \eta)n + (\eta - \mu_i(v))n = (\delta - \mu_i(v))n \leq \delta_i(v)n$$

But this is a contradiction since $d(c_i, c_i') > \delta_i(v)n$. ■

We observe that Blue edges do not cause much branching and cannot result in very deep paths.

Lemma 4.2.6 *A vertex can have at most one Blue edge leaving it. A path from the root to a leaf can have no more than* $\left\lceil \frac{\eta}{\delta - \eta} \right\rceil$ *Blue edges.*

Proof. The first part holds as there can be at most one codeword within the unique decoding radius. After $\left\lceil \frac{\eta}{\delta - \eta} \right\rceil$ Blue edges, all ηn errors have been identified, so all remaining edges have to be White. ■

Lastly, we show that Red edges do not give deep paths either.

Lemma 4.2.7 *A path from the root to a leaf can have no more than* $\left\lceil log_2 \frac{\delta}{\delta - \eta} \right\rceil$ *Red edges.*

Proof. Since $w(c_i) > \delta_i(v)/2$, and the relative distance $\delta_i(v)$ at node v satisfies $\delta_i(v) > (\delta - \mu_i(v))$, we obtain that every Red edge leaving vertex v has weight at least $(\delta - \mu_i(v))/2$.

Assume now for contradiction that some path from the root to a leaf contains k red edges for $k > \left\lceil log_2 \frac{\delta}{\delta-\eta} \right\rceil$. Suppose that the edges have weights $\rho_1, ..., \rho_k$, respectively. Contract the Blue and White edges between successive Red edges into a single edge, whose weight is the sum of weights of the contracted edges. We also do this for the edges before the first Red edge and those after the last Red edge. This gives a path containing $2k + 1$ edges, where the even edges are Red, and the weight of the edges along the path are $\beta_1, \rho_1, \beta_2, ..., \rho_k, \beta_{k+1}$, respectively. Let v_j be the parent vertex of the j-th Red edge for $j \in [1, k]$. Then we have $\mu(v_1) = \beta_1$ and $\mu(v_j) = \beta_j + \rho_{j-1} + \mu(v_{j-1})$ for $j > 1$. But since $\rho_{j-1} > (\delta - \mu(v_{j-1}))/2$ and $\beta_j > 0$, we get

$$\mu(v_j) > \frac{\delta + \mu(v_{j-1})}{2}.$$

Now by induction one obtains $\mu(v_j) \geq \delta(1 - 2^{1-j})$ for all $j \geq 2$. If we take $j = \left\lceil log_2 \frac{\delta}{\delta-\eta} \right\rceil + 1$, then

$$\mu(v_j) > \delta(1 - \frac{\delta - \eta}{\delta}) = \eta.$$

So when we decode at vertex v_j, all the error locations have been identified and erased. Hence we are now decoding from $\eta < \delta$ erasures and no errors, so the decoding is unique and error-free. So vertex v_j will have a single White edge leaving it and no Red edges, which is a contradiction. ■

Theorem 4.2.8 *Assume $\eta < \delta$ and let $x = \left\lceil \frac{\eta}{\delta-\eta} \right\rceil$, $y = \left\lceil log_2 \frac{\delta}{\delta-\eta} \right\rceil$. Then Tree(r) has at most $\binom{x+y}{y} l(\eta)^y$ leaves (and hence $L(\eta) < \binom{x+y}{y} l(\eta)^y$).*

Proof. By Lemma 4.2.5, we know that white edges are the only out-edges leaving their parent nodes, hence they do not contribute to the number of leaves. Thus we eliminate White edges from the tree. Therefore, it is enough to count the number of leaves of a tree consisting only of Red and Blue edges. By Lemma 4.2.6 and 4.2.7, each path of this tree has at most x Blue and y Red edges. Denote this number by $t(x, y)$. We also know that each node of this tree has at most one Blue edge and $l(\eta)$ Red edges leaving it. So, $t(x, y)$ satisfies the recursion $t(x, y) \leq t(x - 1, y) + l(\eta) t(x, y - 1)$ with the initial conditions $t(a, 0) = 1$ for $a = 0, 1, 2, \ldots, x$. Then, $t(x, y) \leq \binom{x+y}{y} l(\eta)^y$. ■

Hence, by using Proposition 4.2.2 and Theorem 4.2.8, one obtains that result of [8] is applicable to the more general interleaved codes. Specifically, we proved that

Theorem 4.2.9 *Let C be interleaving of $C_1, \ldots, C_b$ of length n and minimum distance δ_i. Let $\delta = min_i \delta_i$ and, for $\eta < \delta$, $l(\eta)$ be the maximum of list sizes of $C_1, \ldots, C_b$ when list decoded up to radius η with time complexity T_i. And, let $x = \left\lceil \frac{\eta}{\delta - \eta} \right\rceil$ and $y = \left\lceil log_2 \frac{\delta}{\delta - \eta} \right\rceil$. Then C can be list decodable up to radius η with list size $L(\eta) < \binom{x+y}{y} l(\eta)^y$ and time complexity at most $O(\sum_i T_i + b^2 n l(\eta) L(\eta))$.*

In Section 5.3, by using Algorithm 4 and Theorem 4.2.9, we present a new list decoding algorithm for folded Hermitian codes.

CHAPTER 5

DECODING FOLDED HERMITIAN CODES

Hermitian codes are one of the families of Algebraic-geometric (AG). AG codes were first introduced by Goppa [9]. In 1982, Tsfasman, Vladut and Zink [36] showed the existence of a sequence of codes that exceeds the Gilbert-Varshamov bound. AG codes are also thought as a generalization of RS codes. The length of the RS codes is limited by the size of the finite field, but AG codes enable us to break this limitation, and so to reduce field operations.

Feng and Rao [7] presented an algorithm based on Gaussian elimination which corrects all error patterns of weight less than half the minimum distance. Later, Sakata, Justesen, Madelung, Elbrond Jensen and Hoholdt [23, 24, 25] extended the idea of Feng and Rao by using extended Berlekamp-Massey approach so that all error patterns of weight less than half the minimum distance were corrected with low complexity.

In 1999, Shokrollahi and Wasserman generalized Sudan's algorithm [35] and derived a list-decoding algorithm for AG codes [26]. The list-decoding algorithm in [26] work only for codes of low rates. Guruswami and Sudan [11] later proposed improved list-decoding algorithm (GS algorithm) for AG codes. The algorithm of Guruswami and Sudan have the largest list decoding radius for AG codes up to date, and are applicable to codes of any rates.

Guruswami and Rudra [12] constructed folded RS codes that achieve the information theoretically best possible trade off between the rate and error-correction radius. They presented a list decoding algorithm (GR algorithm) for folded RS codes that decodes up to information theoretically best possible radius (called capacity) and runs in polynomial time, and also outputs a list of codewords whose size is also polynomial.

One of the generalization of GR-algorithm would be list decoding AG codes up to capacity.

Such a generalization is studied in [13] and [15]. Guruswami [13] explains how capacity achieving list decoding schemes for RS codes arise out of the Artin-Frobenius automorphism at primes in Galois extensions. Technically, Guruswami constructs new list-decodable capacity achieving AG codes (called folded AG codes) based on cyclotomic function fields with a cyclic Galois group.

Huang and Narayanan [15] also consider AG codes constructed from Galois extensions, and observe how automorphisms of large order can be used for folding such codes.

Main motivations of these works are to gain a deeper understanding of the general algebraic principles underlying the above folding and extending it to more general AG codes. The latter is important for potentially improving the alphabet size of the codes, as well as the decoding complexity and output list size of the decoding algorithm. Huang and Narayanan [15] defined codes on cyclotomic function fields that do not improve folded Reed Solomon codes in terms of alphabet size. But, Guruswami [13] overcame this obstacle by considering certain special subfields of the cyclotomic fields thereby achieving an alphabet size that is logarithmic in the block length, which is an improvement on folded RS case whose alphabet size is polynomial in the block length.

In this chapter, decoding of a special family of AG codes called Hermitian code is studied. Hermitian codes are defined on the function field of Hermitian curve. It is shown in [22] that decoding Hermitian codes can be reduced to decoding interleaving of heterogeneous Reed-Solomon codes, which is also presented in Section 5.1. Then, single and list decoders of heterogeneous IRS codes presented in previous chapters can be applied to decoding Hermitian codes and list decoding Hermitian codes, which are presented in Section 5.2 and Section 5.3, respectively.

5.1 Folded Hermitian Codes

A Hermitian curve $H(q)$ over $\mathbb{F}_{q^2}$ in affine coordinates is defined by

$$H(q) : y^q + y = x^{q+1}. \tag{5.1}$$

There are $q^3 + 1$ rational points on $H(q)$, the q^3 points that satisfy (5.1) are denoted by $R_1, R_2, \ldots, R_n$, where $n = q^3$, and the point at infinity will be denoted by Q.

The following proposition is from [34], [38].

Proposition 5.1.1 *For each $m \geq 0$, the following set is a basis of $L(mQ)$*

$$\{x^i y^j | 0 \leq i, 0 \leq j \leq q-1, iq + j(q+1) \leq m\}.$$

Then, Hermitian code is defined as

$$H_m = C(D, mQ) = \{(g(R_1), g(R_2), \dots, g(R_n)) | g \in L(mQ)\}$$

where $D = R_1 + R_2 + \dots + R_n$ with $n = q^3$.

From [38], we know that $x^q + x = 0$ has q solutions in $\mathbb{F}_{q^2}$. We use $B = \beta_1, \beta_2, \dots, \beta_q$ to denote the set of solutions to the equation $x^q + x = 0$. Let $(1, y_0)$ be a solution to (5.1), then according to [38],

$$(\eta, \eta^{q+1} y_0 + \beta_i)$$

are all the q^3 rational points on Hermitian curve $H(q)$, where $\eta \in \mathbb{F}_{q^2}$.

Suppose α is a primitive element in $\mathbb{F}_{q^2}$, then the q^3 rational points on Hermitian curve can be expressed as

$$(\alpha^i, \alpha^{i(q+1)} y_0 + \beta_j)$$

where $i = -\infty, 0, 1, 2, \dots, q^2 - 2$, $j = 1, 2, \dots, q$, and $\alpha^{-\infty} := 0$.

Now, we will continue with the decoding of Hermitian code as given in [22], and then we will apply the IRS Decoder presented in Section 3.2 to obtain a new decoding algorithm for folded Hermitian codes. In order to describe the decoding algorithm, we first arrange the q^3 rational points as in the following list.

$$\begin{aligned}
P_{1,1} &= (0,\beta_1)\\
P_{1,2} &= (0,\beta_2)\\
&\vdots\\
P_{1,q} &= (0,\beta_q)\\
P_{2,1} &= (\alpha^0,\alpha^0 y_0+\beta_1)\\
P_{2,2} &= (\alpha^0,\alpha^0 y_0+\beta_2)\\
&\vdots\\
P_{2,q} &= (\alpha^0,\alpha^0 y_0+\beta_q)\\
&\vdots\\
P_{q^2,1} &= (\alpha^{q^2-2},\alpha^{(q^2-2)(q+1)} y_0+\beta_1)\\
P_{q^2,2} &= (\alpha^{q^2-2},\alpha^{(q^2-2)(q+1)} y_0+\beta_2)\\
&\vdots\\
P_{q^2,q} &= (\alpha^{q^2-2},\alpha^{(q^2-2)(q+1)} y_0+\beta_q)
\end{aligned}$$

Now we express the Hermitian code H_m over $\mathbb{F}_{q^2}$ as

$$\{(g(P_{1,1}),\dots,g(P_{1,q}),\dots,g(P_{q^2,1}),\dots,g(P_{q^2,q}))|g \in L(mQ)\}.$$

Suppose $r = (g(P_{1,1}),\dots,g(P_{1,q}),\dots,g(P_{q^2,1}),\dots,g(P_{q^2,q}))$ is a codeword that is transmitted for some $g \in L(mQ)$. We will only consider the case $m \geq q^2-1$. Then, for any $g \in L(mQ)$, we may assume

$$g(P_{i,l}) = f_0(P_{i,l}) + y(P_{i,l})f_1(P_{i,l}) + \ldots + y^{q-1}(P_{i,l})f_{q-1}(P_{i,l})$$

where $degf_j < k(j)$ for $j = 0,1,2,\dots,q-1$ and

$$k(j) = max\{i|iq + j(q+1) \leq m\} + 1 = \left\lfloor \frac{m - j(q+1)}{q} \right\rfloor + 1$$

according to Proposition 2 in [22].

For each i, the q rational points $P_{i,1},\dots,P_{i,q}$ have the same first coordinate. Therefore, $f_j(P_{i,l})$ does not depend on l. In fact, $f_j(P_{i,l}) = f_j(\alpha^{i-2})$ for $i = 2,\dots,q^2$ and $f_j(P_{1,l}) = f_j(0)$. Let $u = (u_{1,1},\dots,u_{1,q},\dots,u_{q^2,1},\dots,u_{q^2,q})$ be the received word.

Replace $f_j(P_{i,l})$ and $g(P_{i,l})$ by $x_{j,i}$ and $u_{i,l}$, respectively, and regard $x_{j,i}$ as variables. From the

q rational points in the i-th row, $P_{i,1}, P_{i,2}, \ldots, P_{i,q}$, we get the following equation-system

$$\begin{aligned} x_{0,i} + y(P_{i,1})x_{1,i} + \ldots + (y(P_{i,1}))^{q-1}x_{q-1,i} &= u_{i,1}, \\ x_{0,i} + y(P_{i,2})x_{1,i} + \ldots + (y(P_{i,1}))^{q-1}x_{q-1,i} &= u_{i,2}, \\ &\vdots \\ x_{0,i} + y(P_{i,q})x_{1,i} + \ldots + (y(P_{i,1}))^{q-1}x_{q-1,i} &= u_{i,q}. \end{aligned} \tag{5.2}$$

Since the coefficient matrix for (5.2) is a Vandermonde matrix, there is a unique solution to (5.2). Obviously, when $u = (u_{1,1}, \ldots, u_{1,q}, \ldots, u_{q^2,1}, \ldots, u_{q^2,q})$ is a codeword, that is u has no errors, then we can solve $(x_{0,i}, x_{1,i}, \ldots, x_{q-1,i})$ from (5.2) for all $i = 1, \ldots, q^2$. By solving all the q^2 equation-systems, we get q^2 solutions:

$$\begin{array}{cccc} (f_0(0), & f_1(0), & \ldots, & f_{q-1}(0)) \\ (f_0(\alpha^0), & f_1(\alpha^0), & \ldots, & f_{q-1}(\alpha^0)) \\ (f_0(\alpha^1), & f_1(\alpha^1), & \ldots, & f_{q-1}(\alpha^1)) \\ \vdots & \vdots & & \vdots \\ (f_0(\alpha^{q^2-2}), & f_1(\alpha^{q^2-2}), & \ldots, & f_{q-1}(\alpha^{q^2-2})) \end{array} \tag{5.3}$$

Let $r_j = (f_j(0), f_j(\alpha^0), f_j(\alpha^1), \ldots, f_j(\alpha^{q^2-2}))$, $j = 0, 1, \ldots, q-1$.

Whenever $u = (u_{1,1}, \ldots, u_{1,q}, \ldots, u_{q^2,1}, \ldots, u_{q^2,q})$ has errors, there exists at least one j such that r_j has errors.

Suppose that we bundle each consecutive q symbols of H_m together and obtain a new code called as q-folded Hermitian code H'_m of block length q^2, over the alphabet $\mathbb{F}_{q^{2q}}$. Then, if H'_m has agreement at least t, then at least t many of $(u_{1,1}, u_{1,2}, \ldots, u_{1,q}), \ldots, (u_{q^2,1}, u_{q^2,2}, \ldots, u_{q^2,q})$ have no errors, and at least t many of solutions as obtained in (5.3) are correct solutions. Then, each r_j has agreement at least t, and agreement places are same for all $j = 0, 1, \ldots, q-1$.

Now, we define RS codes by using sub-polynomials used in H'_m as

$$C_{j+1} := \{(f_j(0), f_j(\alpha^0), \ldots, f_j(\alpha^{q^2-2})) | deg(f_j) < \left\lfloor \frac{m - j(q+1)}{q} \right\rfloor + 1\}$$

for each $j = 0, 1, \ldots, q-1$. So, decoding H'_m is reduced to decoding heterogeneous interleaved RS codes, C_{j+1} for $j = 0, 1, \ldots, q-1$. Hence, we proved:

Theorem 5.1.2 *Decoding q-folded Hermitian code over $\mathbb{F}_{q^{2q}}$ can be reduced to decoding heterogeneous interleaved Reed-Solomon codes over $\mathbb{F}_{q^2}$. As a generalization, for any positive* *integer h, qh-folded Hermitian code over $\mathbb{F}_{q^{2qh}}$ can be reduced to decoding heterogeneous interleaved h-folded Reed-Solomon codes over $\mathbb{F}_{q^{2h}}$.*

Remark 5.1.3 *Through the reduction steps of q-folded Hermitian code to interleaved RS codes, most time consuming operation is solving (5.2) for q^2 times, whose time complexity is at most $q^2O(q^3) = O(q^5)$. Next, after decoding RS codes, recovering the Hermitian codeword again requires to calculate (5.2) for q^2 times by replacing with the output of the decoding algorithm of interleaved RS codes, which takes at most $q^2 \cdot q^2O(logq) = O(q^4logq)$. Overall, reduction and recovery have time complexity at most $O(q^5)$.*

5.2 Decoding Folded Hermitian Codes with Heterogeneous IRS Decoder

We studied decoding of interleaved RS codes in Section 3.2. When the IRS Decoder is used to decode q-folded Hermitian codes, Theorem 3.2.2, Theorem 5.1.2 and Remark 5.1.3 result in the following theorem.

Theorem 5.2.1 *Assume that messages and error values are random.*

i. q-folded Hermitian codes H'_m over $\mathbb{F}_{q^{2q}}$ satisfying

$$\left\lfloor \frac{m}{q} \right\rfloor + 1 \leq \left\lceil \frac{q^2 + \sum_{j=0}^{q-1} (\lfloor \frac{m-j(q+1)}{q} \rfloor + 1)}{q+1} \right\rceil$$

and having rate $R = \frac{\sum_{j=0}^{q-1} k(j)}{q^3}$ can be corrected with the IRS Decoder up to error rate ϵ, where

$$\epsilon < \frac{q}{q+1}(1 - R) \tag{5.4}$$

with probability at least $1 - \frac{q}{q+1}(1 - R)$

ii. q-folded Hermitian codes H'_m over $\mathbb{F}_{q^{2q}}$ satisfying

$$\left\lfloor \frac{m}{q} + 1 \right\rfloor > \left\lceil \frac{q^2 + \sum_{j=0}^{q-1} (\lfloor \frac{m-j(q+1)}{q} \rfloor + 1)}{q+1} \right\rceil$$

and can be corrected with the IRS Decoder up to error rate ϵ, where

$$\epsilon \leq 1 - \frac{\lfloor \frac{m}{q} \rfloor + 1}{q^2} \tag{5.5}$$

with probability at least $\eta = \frac{\lfloor \frac{m}{q} \rfloor + 1}{q^2}$.

Complexity of the decoding procedure is at most $O(q^{11})$ *field operations.*

Remark 5.2.2 *There are better decoding algorithms for number of burst errors less than*

$$q^2 - \frac{q^2 + min_{j=1}^{q-1}\{k_j\}}{2} = q^2 - \frac{q^2 - \left\lfloor \frac{(q-1)(q+1)}{q} \right\rfloor + 1}{2}.$$

It is seen that (5.4) is better than this bound. Similarly, we observe that (5.5) is better than this bound for $m < q^3 - q^2$.

Example 5.2.3 *Let q=4 and m=37. Then, Hermitian curve is defined by* $y^4 + y = x^5$ *over* $\mathbb{F}_{4^2}$. *Let D be sum of all finite points of the Hermitian curve* $D = R_1 + R_2 + \ldots + R_{64}$ *and Q is the point at infinity. We try to decode the Hermitian code* $H_{37} := C(D, 37Q)$ *over* $\mathbb{F}_{4^2}$ *with the IRS Decoder. Firstly, we calculate* $k(0) = 10$, $k(1) = 9$, $k(2) = 7$, $k(3) = 6$. *Then, according to Theorem 5.1.2, decoding 4-folded* H_{37} *code can be reduced to decoding* $IRS(16; 10, 9, 7, 6)$ *code over* $\mathbb{F}_{4^2}$. *Therefore, 4-folded* H_{37} *code can be decoded up to 6 errors with IRS decoder with failure probability* $3,75 \cdot 10^{-1}$. *However, when IRS Decoder is simulated for this case, it is experimentally obtained that the failure probability is approximately* $1,72 \cdot 10^{-3}$. *Experimental failure probabilities* P_f *of Hermitian code decoder based on Algorithm 3 is presented in Table 5.1.*

Table 5.1: Simulated Failure Probability for Hermitian Codes

Code	corrected errors	η	P_f
4-folded H_{37}	6	$3,75 \cdot 10^{-1}$	$1,72 \cdot 10^{-3}$

In [22], an example of decoding H_{37} over $\mathbb{F}_{4^2}$ for some burst error of weight 24 is demonstrated. Theorem 5.2.1 says that at least 62,5 percent of burst errors of weight 24 can be corrected while experimental results say that approximately 99,828 percent of burst errors of weight 24 can be corrected. Therefore, results obtained in this section are an extension of results in [22].

5.3 List Decoding Folded Hermitian Codes

We studied list decoding of interleaved codes in Chapter 4. And, it is observed that decoding Hermitian codes can be reduced to decoding heterogeneous interleaved Reed-Solomon codes in Section 5.1. Hence, following result is a direct consequence of Theorem 4.2.9, Theorem 5.1.2, and Remark 5.1.3.

Proposition 5.3.1 *Let q-folded Hermitian code H'_m be reduced to interleaving of RS codes $C_1, \ldots, C_q$ of length q^2 and minimum distance δ_i. Let $\delta = min_i \delta_i$ and, for $\eta < \delta$, $l(\eta)$ be the maximum of list sizes of $C_1, \ldots, C_q$ when list decoded up to radius η with time complexity T_i. And, let $x = \left\lceil \frac{\eta}{\delta-\eta} \right\rceil$ and $y = \left\lceil log_2 \frac{\delta}{\delta-\eta} \right\rceil$. Then H'_m can be list decodable up to radius η with list size $L(\eta) < \binom{x+y}{y} l(\eta)^y$ and time complexity at most $O(q^5 + \sum_i T_i + q^4(log_2 q + l(\eta))L(\eta))$.*

Since any RS code can be list decodable up to the largest decoding radius by GS algorithm up to date, hence we obtain the following result by using Proposition 5.3.1 and Theorem 4.1.1.

Corollary 5.3.2 *For every $0 < \epsilon$, let $r_1 = \frac{\lfloor \frac{m}{q} \rfloor + 1}{q^2}$, $r_2 = \frac{\lfloor \frac{m-q^2+1}{q} \rfloor + 1}{q^2}$, $x = \left\lceil \frac{1-\sqrt{(1+\epsilon)r_1}}{\sqrt{(1+\epsilon)r_1} - r_1} \right\rceil$ and $y = \left\lceil log_2 \frac{1-r_1}{\epsilon} \right\rceil$. q-folded Hermitian code H'_m can be list decoded up to a fraction $1 - \sqrt{(1+\epsilon)r_1}$ of errors in time $O(q^5 + q^4 \binom{x+y}{y}(log_2 q + \epsilon^{-1}/\sqrt{r_2})(\epsilon^{-1}/\sqrt{r_2})^y)$ and outputs a list of size at most $\binom{x+y}{y}(\epsilon^{-1}/\sqrt{r_2})^y$ where alphabet size of the code is q^{2q}.*

Proof. Parameters $k(j)$ are calculted explicitly for $j = 0, 1, \ldots, q-1$ as

$$\begin{aligned} k(j) &= max\{i | iq + j(q+1) \leq m\} + 1 \\ &= max\{i | i \leq \tfrac{m-j(q+1)}{q}\} + 1 \\ &= \left\lfloor \tfrac{m-j(q+1)}{q} \right\rfloor + 1. \end{aligned}$$

Hence,

$$max_j\{k(j)\} = k(0) = \left\lfloor \frac{m}{q} \right\rfloor + 1 \tag{5.6}$$

and

$$min_j\{k(j)\} = k(q-1) = \left\lfloor \frac{m - q^2 + 1}{q} \right\rfloor + 1. \tag{5.7}$$

The remaining parts of the theorem follows from Proposition 5.3.1 and Theorem 4.1.1. ■

We give the decoding algorithm steps of q-folded Hermitian code described in Algorithm 6.

Example 5.3.3 *We try to decode the Hermitian code $H_{37} := C(D, 37Q)$ over $\mathbb{F}_{4^2}$ given in Example 5.2.3 with Algorithm 6. We calculate $r_1 = 10/16, r_2 = 6/16, x = 2$ and $y = 3$ for a chosen $\epsilon = 0,05$. Then, 4-folded H_{37} can be list decodable up to 3 errors (totally 12 errors) in time approximately $O(4^{10})$ with list size at most $O(4^7)$.*

Moreover, in order to increase the decoding radius, we can think each sub-RS codes as folded RS codes. But, for this, we need to fold Hermitian code extra (approximately $O(1/\epsilon^2)$ times for some ϵ). By folding Hermitian code extra, each sub-RS code becomes a folded RS code, whose list decoding by GR algorithm reaches to capacity, and so, we obtain by using Theorem 5.1.2, Proposition 5.3.1 and Theorem 4.1.2 a folded Hermitian codes that has the following results.

Corollary 5.3.4 *For every $\frac{1}{q} < \epsilon < 1 - r_1$ where $r_1 = \frac{\lfloor \frac{m}{q} \rfloor + 1}{q^2}$, let $r_2 = \frac{\lfloor \frac{m - q^2 + 1}{q} \rfloor + 1}{q^2}$, $x = \lceil \frac{1 - r_1 - \epsilon}{\epsilon} \rceil$, $y = \lceil log_2 \frac{1 - r_1}{\epsilon} \rceil$ and $T = (q^2)^{O(\epsilon^{-1} log_2(1/r_2))}$, then q/ϵ^2-folded Hermitian codes of block length $(\epsilon q)^2$ can be list decoded up to a fraction $1 - r_1 - \epsilon$ of errors in time $O(q^5 + q^4 \binom{x+y}{y}(log_2 q + T)T^y)$ and outputs a list of size at most $\binom{x+y}{y} T^y$ where alphabet size of the code is $q^{2O(q/\epsilon^2)}$.*

Similar to Algorithm 6, a list decoding algorithm of qh-folded Hermitian code can be described.

It is not easy to compare fairly our list decoding algorithm for folded Hermitian codes with previously known algorithms. We briefly mention similarities and differences with GS algorithm [11]. Similar to GS algorithm, our decoding algorithm is in polynomial time and output list size is polynomial. However, our decoding radius given in Corollary 5.3.4 is generally better than GS algorithm (see Theorem 4.1.1), which is observed below, but our alphabet size is very big. In order to reduce alphabet size, one can use code concatenation as it is done in [12].

We will now compare our list decoding radius $1 - r_1 - \epsilon$ with one in [11]. The decoding radius can be at most

$$\eta < 1 - r_1 - \epsilon = 1 - \frac{max_j\{k(j)\}}{q^2} - \epsilon$$

for $\epsilon > 1/q$, so by (5.6),

$$\eta < 1 - \frac{\lfloor \frac{m}{q} \rfloor + 1}{q^2} - \frac{1}{q}.$$

Algorithm 6 List Decoding q-folded Hermitian Codes

```
Require:
    • Let $\alpha$ be a primitive element in $GF(q^2)$
    • Let $P_{1,1} = (0,\beta_1)$, $P_{2,1} = (\alpha^0, \alpha^0 y_0 + \beta_1)$,...,$P_{q^2,1} = (\alpha^{q^2-2}, \alpha^{(q^2-2)(q+1)} y_0 + \beta_1)$,..., $P_{1,q} = (0,\beta_q)$,
      $P_{2,q} = (\alpha^0, \alpha^0 y_0 + \beta_q)$,...,$P_{q^2,q} = (\alpha^{q^2-2}, \alpha^{(q^2-2)(q+1)} y_0 + \beta_q)$.
    • Suppose $r = (G_1, G_2, \ldots, G_{q^2})$ is a codeword that is transmitted, where $G_i = (g(P_{i,1}),$
      $\ldots, g(P_{i,q}))$ and $g \in L(mQ)$.
    • Let $U = (U_1, \ldots, U_{q^2})$ be the received word where $U_i = (u_{i,1}, \ldots, u_{i,q})$
Ensure: List $L$ of all $c \in C$ so that $\Delta_{q^{2q}}(r, c) \leq \eta$.
1: Solve the following equation-system
       $x_{0,i} + y(P_{i,1})x_{1,i} + \ldots + (y(P_{i,1}))^{q-1} x_{q-1,i} = u_{i,1},$
       $x_{0,i} + y(P_{i,2})x_{1,i} + \ldots + (y(P_{i,1}))^{q-1} x_{q-1,i} = u_{i,2},$
       $\vdots$
       $x_{0,i} + y(P_{i,q})x_{1,i} + \ldots + (y(P_{i,1}))^{q-1} x_{q-1,i} = u_{i,q}.$
2: Get $q^2$ solutions:
       $(f_0(0)$              ,  $f_1(0)$              , ... ,  $f_{q-1}(0))$
       $(f_0(\alpha^0)$       ,  $f_1(\alpha^0)$       , ... ,  $f_{q-1}(\alpha^0))$
       $(f_0(\alpha^1)$       ,  $f_1(\alpha^1)$       , ... ,  $f_{q-1}(\alpha^1))$
       $\vdots$                  $\vdots$                          $\vdots$
       $(f_0(\alpha^{q^2-2})$ ,  $f_1(\alpha^{q^2-2})$ , ... ,  $f_{q-1}(\alpha^{q^2-2}))$
3: $C_{j+1} := \{(f_j(0), f_j(1), \ldots, f_j(\alpha^{q^2-2})) | deg(f_j) < k(j)\}$ for each $j = 0, 1, \ldots, q-1$.
4: Define $H'_m$ as interleaving of $C_1, \ldots, C_q$.
5: Decode $H'_m$ with Algorithm 4.
6: Let $L$ be the output list of Step 5 with size $l$.
7: Calculate the following equations by replacing $x_{i,j,k}$ with the components of each element of $L$ for
   $i = 1, \ldots, q^2$; $j = 0, 1, \ldots, q-1$; and $k = 1, \ldots, l$
       $\bar{u}_{i,1,k} = y(P_{i,1})^0 x_{0,i,k} + \ldots + (y(P_{i,1}))^{q-1} x_{q-1,i,k},$
       $\bar{u}_{i,2,k} = y(P_{i,2})^0 x_{0,i,k} + \ldots + (y(P_{i,1}))^{q-1} x_{q-1,i,k},$
       $\vdots$
       $\bar{u}_{i,q,k} = y(P_{i,q})^0 x_{0,i,k} + \ldots + (y(P_{i,1}))^{q-1} x_{q-1,i,k}.$
8: Let $\bar{U}_{i,k} = (\bar{u}_{i,1,k}, \bar{u}_{i,2,k}, \ldots, \bar{u}_{i,q,k})$ for $i = 1, \ldots, q^2$ and $k = 1, \ldots, l$.
9: return $\bar{U}_k = (\bar{U}_{1,k}, \ldots, \bar{U}_{q^2,k})$ for $k = 1, \ldots, l$.
```

At worst, η can be chosen as

$$1 - \frac{\frac{m}{q} + 1}{q^2} - \frac{1}{q} = 1 - \frac{m + q + q^2}{q^3} \tag{5.8}$$

On the other hand, Guruswami-Sudan list decoding radius is at most $1 - \sqrt{R}$, where R is the rate of Hermitian code. By using Corollary 1 of [22],

$$dimH_m = \sum_{j=0}^{q-1} k(j),$$

and so

$$R = \frac{\sum_{j=0}^{q-1} k(j)}{q^3}.$$

Then,

$$\begin{aligned} 1 - \sqrt{R} &\leq 1 - \sqrt{\frac{\sum_{j=0}^{q-1} k(j)}{q^3}} \\ &= 1 - \sqrt{\frac{\sum_{j=0}^{q-1} \left\lfloor \frac{m-j(q+1)}{q} \right\rfloor + 1}{q^3}} \\ &\leq 1 - \sqrt{\frac{\sum_{j=0}^{q-1} \frac{m-j(q+1)}{q} - 1 + 1}{q^3}} \\ &= 1 - \sqrt{\frac{\sum_{j=0}^{q-1} \frac{m}{q} - j\frac{q+1}{q}}{q^3}} \\ &= 1 - \sqrt{\frac{m - \frac{(q-1)q}{2}\frac{q+1}{q}}{q^3}} \\ &= 1 - \sqrt{\frac{2m - q^2 + 1}{2q^3}} \end{aligned} \tag{5.9}$$

It is time to investigate for which values of m, (5.8) is better than (5.9).

$$1 - \frac{m + q + q^2}{q^3} > 1 - \sqrt{\frac{2m - q^2 + 1}{2q^3}}$$

$$\frac{m + q + q^2}{q^3} < \sqrt{\frac{2m - q^2 + 1}{2q^3}}$$

$$2m^2 + (4q + 4q^2 - 2q^3)m + q^5 + 2q^4 + 3q^3 + 2q^2 < 0 \tag{5.10}$$

If $q \geq 7$ and m takes values between $1/2(-2q - 2q^2 + q^3 - \sqrt{2q^3 - 4q^4 - 6q^5 + q^6})$ and $1/2(-2q - 2q^2 + q^3 + \sqrt{2q^3 - 4q^4 - 6q^5 + q^6})$, then (5.10) is satisfied. Since it is preassumed that $m \geq q^2 - 1$, we obtain that m must be at least $\max\{q^2 - 1, 1/2(-2q - 2q^2 + q^3 - \sqrt{2q^3 - 4q^4 - 6q^5 + q^6})\}$ in order to have better decoding radius. If one chooses $m = \left[\frac{(q^3-2q^2-2q)}{2}\right]$, then the difference maximizes.

We present he following example to illustrate the results.

Example 5.3.5 *We try to decode the Hermitian code* $H_{1776} := C(D, 1776Q)$ *over* $\mathbb{F}_{16^2}$ *where m was chosen as* $m = \left[\frac{(q^3-2q^2-2q)}{2}\right]$. *We calculate* $r_1 = 0,44, r_2 = 0,38, x = 4$ *and* $y = 3$ *for a given* $\epsilon = 0,125$. *Then, 1024-folded* H_{1776} *code can be list decodable up to 111 errors (totally 1776 errors) in time approximately* $O(16^{54})$ *with list size at most* $O(16^{50})$ *whose alphabet size is* 16^{2048} *and block length is 4. On the other hand, GS Algorithm decodes* H_{1776} *up to 1405 errors in time approximately* $O(16^6)$ *with list size at most* $O(16^6)$ *whose alphabet size is* 16^2 *and block length is* 16^3.

CHAPTER 6

CONCLUSION

In this thesis decoding algorithms for interleaved Reed-Solomon codes and their applications are presented.

Firstly, a probabilistic algorithm solving simultaneous polynomial reconstruction problem for polynomials allowed to have distinct degrees is obtained. And, it is extended in order to increase the probability of the algorithm. Then, this algorithm is used for decoding heterogeneous interleaved Reed-Solomon codes, Reed Solomon codes and folded-Hermitian codes probabilistically. All probabilistic decoders have better decoding radius than half the minimum distance and they run in polynomial time.

Next, it is shown that interleaved codes whose minimum distances would be different can be list decodable up to radius of minimum of list decoding radii of subcodes, and a list decoding algorithm for those codes using sub-list decoding algorithms is presented. Then, by using decoding algorithm of interleaved different codes, a new basic decoding algorithm for folded Hermitian codes with significant list decoding radius is obtained.

REFERENCES

[1] E. R. Berlekamp, L. Welch, *Error correction of algebraic block codes*, US Patent, Number 4, 633, 470, 1986.

[2] D. Bleichenbacher, A. Kiayias, and M. Yung, *Decoding of interleaved Reed Solomon codes over noisy data*, in Springer Lecture Notes in Computer Science, vol. 2719, pp. 97-108, Jan. 2003.

[3] E. L. Blokh and V. V. Zyablov, *Coding of generalized concatenated codes*, Transl. from Russian, original in Problemy Peredachi Informatsii, pp. 45-50 Probl. Inf. Transm., vol. 10, pp. 218-222, Jul.-Sep. 1974.

[4] A. Brown, L. Minder, and A. Shokrollahi, *Probabilistic decoding of interleaved RS-codes on the q-ary symmetric channel*, in Proc. IEEE Int. Symp. Information Theory, Chicago, IL, p. 327, 2004.

[5] A. Brown, L. Minder, and A. Shokrollahi, *Improved decoding of interleaved AG codes*, in Cryptography and Coding . Berlin, Germany: Springer Verlag, vol. 3796 of Lecture Notes in Computer Science, pp. 37-46, Dec. 2005.

[6] P. Elias, *List decoding for noisy channel*, Res. Lab. Electron., MIT, Cambridge, MA, Tech. Rep. 335, 1957.

[7] G. L. Feng and T. R. N. Rao, *Decoding algebraic geometric codes up to the designed minimum distance*, IEEE Trans. Inform. Theory, vol. 39, pp. 3746, Jan 1993.

[8] P. Gopalan, V. Guruswami, P. Raghavendra, *List decoding tensor products and interleaved codes*, STOC'09: Proceedings Of The 2009 Acm Symposium on Theory of Computing Pages: 13-22, 2009.

[9] V.D. Goppa, *Codes associated with divisors*, Probl. Peredachi Inform. vol. 13 (1), pp. 33-39, 1977. Translation: Probl. Inform. Transmission, vol. 13, pp. 22-26, 1977.

[10] P. Gemmell and M. Sudan. *Highly resilient correctors for polynomials*, Information processing letters, 43(4):169-174, Sep. 1992.

[11] V. Guruswami and M. Sudan, *Improved decoding of Reed-Solomon and algebraic-geometry codes*, IEEE Trans. Inf. Theory, vol. 45, pp. 1757-1767, Nov. 1999.

[12] V. Guruswami and A. Rudra *Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy*, IEEE Trans. Inf. Theory, vol. 54, pp. 135-150, 2008.

[13] V. Guruswami, *Artin automorphisms, cyclotomic function fields, and folded list decodable codes*, STOC'09: Proceedings Of The 2009 Acm Symposium on Theory of Computing Pages: 23-32, 2009.

[14] V. Guruswami *Algorithmic Results in List Decoding*, volume 2 of Foundations and Trends in Theoretical Computer Science. NOW publishers, No 2, 107-195, 2007.

[15] M.D. Huang and A.K. Narayanan, *Folded algebraic geometric codes from Galois extensions*, http://arxiv.org/abs/0901.1162v1

[16] J. Justesen, C. Thommesen, and T. Hoholdt, *Decoding of concatenated codes with interleaved outer codes*, in Proc. IEEE Int. Symp. Inf. Theory, Chicago, IL, p. 329, 2004.

[17] V. Y. Krachkovsky and Y. X. Lee, *Decoding for interleaved Reed-Solomon schemes*, IEEE Trans. Magn., vol. 33, pp. 2740-2743, Sep. 1997.

[18] V. Y. Krachkovsky, *Reed-Solomon codes for correcting phased error busts*, IEEE Trans. Inf. Theory, vol. 49, pp. 2975-2984, Nov. 2003.

[19] V. Y. Krachkovsky, Y. X. Lee, and H. K. Garg, *Decoding of parallel RS codes with applications to product and concatenated codes*, in Proc. IEEE Int. Symp. Inf. Theory, Boston, MA, p. 55, 1998.

[20] F. Parvaresh and A. Vardy. *Correcting errors beyond the Guruswami-Sudan radius in polynomial time*'. In Proc. 46th IEEE Symp. Foundations of Comp. Science, pages 285-294, 2005.

[21] F. Parvaresh and A. Vardy, *Multivariate interpolation decoding beyond the Gurswami-Sudan radiu*s, in Proc. 42nd Annu. Allerton Conf. Commun., Contr. Comput., Urbana, IL, 2004.

[22] J. Ren, *On the structure of Hermitian codes and decoding for burst errors*, IEEE Trans. Inf. Theory, vol. 50, pp. 2850-2854, 2004.

[23] S. Sakata, J. Justesen, Y. Madelung, H. Elbrond Jensen and T. Hoholdt, *Fast decoding of algebraic geometric codes up to the designed minimum distance*, IEEE Trans. Inform. Theory, vol. 41, pp. 1672-1677, Nov. 1995.

[24] S. Sakata, H. Elbrond Jensen and T. Hoholdt, *Generalized Berlekamp-Massey decoding of algebraic geometric codes up to half the Feng-Rao bound*, IEEE Trans. Inform. Theory, vol. 41, pp. 1762-1768, Nov. 1995.

[25] S. Sakata, J. Justesen, Y. Madelung, H. Elbrond Jensen and T. Hoholdt, *A fast decoding method of AG codes from Miura-Kamiya curves Cab up to Half the Feng-Rao bound*, Finite Fields and their Applications vol. 11, pp. 83-101, 1995.

[26] M. Shokrollahi and H. Wasserman, *List decoding of algebraic-geometric codes*, IEEE Trans. Inf. Theory, vol. 45, pp. 432-437, Mar. 1999.

[27] G. Schmidt, V. R. Sidorenko, and M. Bossert, *Error and erasure correction of interleaved Reed-Solomon codes*, in Proc. Int. Workshop Coding Cryptogr., Bergen, Norway, pp. 20-29, Mar. 2005.

[28] G. Schmidt, V. R. Sidorenko, and M. Bossert, *Interleaved Reed-Solomon codes in concatenated code designs*, in Proc. IEEE Information Theory Workshop, Rotorua, New Zealand, pp. 187-191, Aug. 2005.

[29] G. Schmidt, V. R. Sidorenko, and M. Bossert, *Error and erasure correction of interleaved Reed-Solomon codes*, in Coding and Cryptography. Berlin, Germany: Springer-Verlag, vol. 3969 of Lecture Notes in Computer Science, pp. 22-35, 2006.

[30] G. Schmidt, V. R. Sidorenko, and M. Bossert, *Heterogeneous interleaved Reed-Solomon code designs*, in Proc. 10th Int. Workshop Algebraic Combin. Coding Theory (ACCT-10), Zvenigorod, Russia, pp. 230-233, Sep. 2006.

[31] G. Schmidt, V. R. Sidorenko, and M. Bossert, *Decoding Reed-Solomon codes beyond half the minimum distance using shift-register synthesis*, in Proc. IEEE Int. Symp. Inf. Theory, Seattle, WA, pp. 459-463, Jul. 2006.

[32] G. Schmidt, V. R. Sidorenko, and M. Bossert, *Collaborative decoding of interleaved Reed-Solomon codes and concatenated code designs*, IEEE Trans. Inform. Theory 55, no. 7, 2991-3012, 2009.

[33] C. Senger, V. Sidorenko, M. Bossert, and V. Zyablov, *Decoding generalized concatenated codes using interleaved Reed-Solomon codes*, in Proc. IEEE Int. Symp. Information Theory, Toronto, ON, Canada, 2008

[34] H. Stichtenoch, *A note on Hermitian codes over GF(q^2)*, IEEE Trans. Inf. Theory, vol. 34, pp. 1345-1348, Sept. 1988.

[35] M. Sudan, *Decoding of Reed-Solomon codes beyond the error-correction bound*, J. Complexity, vol. 13, pp. 180-193, 1997.

[36] M. A. Tsfasman, S. G. Vladut and T. Zink, *Modular curves, Shimura curves and Goppa codes, better than Varshamove-Gilbert bound*, Math.Nachr., vol. 104, pp. 13-28, 1982.

[37] J. M. Wozencraft, *List decoding*, in Quarterly Progress Report. Cambridge, MA: Res. Lab. Electronics, MIT, vol. 48, pp. 90-95, 1958.

[38] T. Yaghoobian and I. F. Blake, *Hermitian codes as generalized Reed-Solomon codes*, Design, Codes and Cryptography, vol. 2, pp, 5-17, 1992.

[39] V. A. Zinoviev, *Generalized cascade codes*, Transl.: from Russian, original in Problemy Peredachi Informatsii, pp. 5-15 Probl. Inf. Transm., vol. 12, pp. 2-9, Jan.-Mar. 1976.

VITA

PERSONAL INFORMATION

Surname, Name: Yayla, Oğuz

Date and Place of Birth: 1981 - Ankara

Marital Status: Married with one daughter

email: oguzyayla@gmail.com

ACADEMIC DEGREES

Ph.D.	METU, Department of Cryptography Graduate School of Applied Mathematics Middle East Technical University-Ankara Supervisor: Prof. Dr. Ferruh Özbudak Thesis Title: On Decoding Interleaved Reed-Solomon Codes	2011
M.Sc.	METU, Department of Cryptography Graduate School of Applied Mathematics Middle East Technical University-Ankara Supervisor: Prof. Dr. Ersan Akyıldız Thesis Title: Scalar Multiplication on Elliptic Curves	2006
Minor Degree	METU, Dept. of Electrical and Electronics Eng.(Telecommunications)	2005
B.S.	METU, Department of Mathematics	2004
High School	Bursa Boys High School	1999

WORK EXPERIENCE

2008 - 2011	METU, Institute of Applied Mathematics	Research Assistant

PUBLICATIONS

A. Papers in Progress:

A1. with F. Özbudak, List Decoding Interleaved Codes and Folded Hermitian Codes, submitted to AAECC.

A2. with F. Özbudak, Decoding Interleaved Reed-Solomon Codes and Folded Hermitian Codes, in preperation.

B. Papers published in National Conference Proceedings:

B1. with M. Cenk, Ayrık Logaritma Problemini Kullanan E-İmza, Proceedings of Information Security and Cryptology Conference, (ISCTURKEY 2006), Ankara, 2006 pp. 1-6.

B2. with S. Akleylek, PKI-Lite: A PKI System with Limited Resources, Proceedings of Information Security and Cryptology Conference, (ISCTURKEY 2007), Ankara, 2007, pp. 59-62.

B3. DSA Sisteminin Çalıştırılması ve Test Edilmesi, Proceedings of Information Security and Cryptology Conference, (ISCTURKEY 2007), Ankara, 2007, pp. 290-297.

B4. with H. Özadam, On Algebraic Attacks Using Groebner Basis, Proceedings of Information Security and Cryptology Conference, (ISCTURKEY 2007), Ankara, 2007, pp. 312-318.

B5. Kriptografik Modüllerin Güvenlik Gereksinimleri, Proceedings of Information Security and Cryptology Conference, (ISCTURKEY 2008), Ankara, 2008, pp. 253-256.