

Direct Erasure-only Decoding for Generalized Reed-Solomon codes

Man Liang*, Qian Guo*, Bin Wang*

*School of Computer Science, Fudan University, Shanghai, China

{manliang, qianguo, bwang}@fudan.edu.cn

Abstract—The work presented in this paper focuses on an efficient erasure-only decoding algorithm for Generalized Reed-Solomon (in short, GRS) codes. The underlying principle of previous decoding algorithms for GRS is first to recover the codewords and further to obtain the original messages. Greatly different from the previous solutions, our novel algorithm, referred to as Direct Erasure-only Decoding, can directly obtain the original message in one step, which makes decoding process more efficient. In addition, we provide a circuit implementation for the algorithm over a finite field of characteristic 2.

Keywords: - Erasure decoding; Generalized Reed Solomon

I. INTRODUCTION

Reed-Solomon codes (RS) is a class of non-binary cyclic error-correcting codes invented by S. Reed and Solomon in 1960. Since found, RS codes have a wide variety of important applications in practice, from deep-space communication, data storage systems, mobile communications to consumer electronics. In their seminal work [1], they described a systematic way of building such a code that could detect and correct multiple random symbol errors, but also left a open problem that an efficient decoding algorithm for RS was not provided. Since then, the problem of decoding RS codes has attracted lots research efforts.

As the researches continue, the algorithms for hard decision decoding of RS codes are gradually classified into two sorts, namely syndrome-based decoding and interpolation-based decoding. The well-developed algorithms belonging to the former include the Peterson-Gorenstein-Zierler algorithm [2], Berlekamp-Massey algorithm [3], Euclidean algorithm [3], frequency domain algorithm [2], etc, while the algorithms belonging to the latter include the well-known Welch-Berlekamp algorithms [4] and list decoding algorithms [5][6], as Koetter-Vardy algorithm [7] is also a list decoding algorithm but with soft decision approaching.

The decoding processes of previous approaches involve two steps - first to recover the codewords and later to obtain the original messages. Our recent experiences show that the two steps can be combined into one, namely, our algorithm, referred to as Direct Erasure-only Decoding, is able to directly obtain the original message immediately. This remarkable change can significantly improve the efficiency of decoding process. In addition, we provide a circuit implementation for the algorithm over a finite field of characteristic 2.

This paper is organized as follows. Section II introduces some fundamental definitions with respect to erasure decoding

and GRS codes. Section III presents our decoding algorithm, referred to as Direct Erasure-only Decoding, as well as a simple example. In Section IV, a circuit implementation for the algorithm over a finite field of characteristic 2 is provided and corresponding workflow is described. Section V concludes this paper.

II. PRELIMINARIES

To facilitate our discussion, we introduce some fundamental definitions with respect to erasure decoding and Generalized Reed-Solomon codes (in short, GRS) in this section, which will be used throughout the paper.

An erasure code is a forward error correction (FEC) code [8] for the binary erasure channel, which transforms a message of k symbols into a longer message (code word) with n , ($n > k$) symbols such that the original message can be recovered from a subset of the n symbols. The fraction $r = k/n$ is called the *code rate* and the fraction k'/k , where k' denotes the number of symbols required for recovery, is called *reception efficiency*. The main task for erasure decoding is to recover the original message from the k' received symbols.

Optimal erasure codes have the property that any k out of the n code word symbols are sufficient to recover the original message, namely they have optimal reception efficiency. Optimal erasure codes are Maximum Distance Separable codes (in short, MDS). The GRS codes are a special type of MDS codes which is probably the most extensively-used codes in practice. We limit our discussion to GRS codes in this paper.

Definition 1: Let $F = GF(q)$ be a finite field and $w = (w_0, w_1, \dots, w_{n-1}) \in F_q^n$. We denote a $(n \times n)$ diagonal matrix $D(w)$ as

$$D(w) = \begin{pmatrix} w_0 & 0 & \dots & 0 \\ 0 & w_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_{n-1} \end{pmatrix}.$$

Definition 2: Let $F = GF(q)$ and $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$ be k distinct nonzero elements of F . We denote a $k \times k$ Vander-Monde matrix $V(\alpha)$ as

$$V(\alpha) = V(\alpha_1, \alpha_2, \dots, \alpha_k) = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_k \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_k^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \dots & \alpha_k^{k-1} \end{pmatrix}.$$

Definition 3: Let $F = GF(q)$ and $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ be n distinct nonzero elements of F . We denote a $k \times n$ Vander-Monde matrix $V_k(\alpha_1, \alpha_2, \dots, \alpha_n)$ as

$$V_k(\alpha_1, \alpha_2, \dots, \alpha_n) = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \dots & \alpha_n^{k-1} \end{pmatrix}.$$

Definition 4: Let $F = GF(q)$ and $n \in \mathbb{N}$ such that $1 \leq n \leq q$, let $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ be n distinct nonzero elements of F , let $v = (v_1, v_2, \dots, v_n)$ be n nonzero elements taken from F , not being necessarily distinct, and let $k \in \mathbb{N}$ such that $1 \leq k \leq n$. Then a GRS code over F can be defined as

$$GRS_{n,k}(\alpha, v) = \{(v_1 f(\alpha_1), \dots, v_n f(\alpha_n)) | f(x) \in F_q[x]_k\},$$

where $F_q[x]_k$ represents the collection of all polynomials over F_q of degree less than k .

Based upon the above definitions, we obtain the generator matrix of GRS codes as follow:

$$G_{GRS} = \begin{pmatrix} v_1 & v_2 & \dots & v_n \\ v_1 \alpha_1 & v_2 \alpha_2 & \dots & v_n \alpha_n \\ v_1 \alpha_1^2 & v_2 \alpha_2^2 & \dots & v_n \alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ v_1 \alpha_1^{k-1} & v_2 \alpha_2^{k-1} & \dots & v_n \alpha_n^{k-1} \end{pmatrix}.$$

According to Definition 1 and Definition 3, the generator matrix may be represented with following representation:

$$G_{GRS} = V_k(\alpha_1, \alpha_2, \dots, \alpha_n) D(v).$$

We denote $V_k(\alpha_1, \alpha_2, \dots, \alpha_n)$ as G_{RS} , and we have

$$G_{GRS} = G_{RS} D(v).$$

Definition 5: Let $F = GF(q)$, let $\alpha_1, \alpha_2, \dots, \alpha_n$ be n distinct nonzero elements of F , and let $v_1, v_2, \dots, v_n$ be n nonzero elements taken from F , not being necessarily distinct. A GRS code over F is a linear $[n, k, d]$ code over F with a parity-check matrix

$$H_{GRS} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \dots & \alpha_n^{n-k-1} \end{pmatrix} D(v).$$

The GRS codes have two important properties, which are stated by following two propositions.

Proposition 1: Every $[n, k, d]$ GRS codes is MDS, namely, $d = n - k + 1$.

Proposition 2: The dual of a GRS code is a GRS code. In particular, $GRS_{n,k}(\alpha, u)^\perp = GRS_{n,n-k}(\alpha, w)$, where $w = (w_1, w_2, \dots, w_n)$ and $w_i = u_i \prod_{j \neq i} (\alpha_i - \alpha_j)$.

The representation of GRS codes can be in the form of polynomial, in which we can write a parity-check matrix as follow:

$$H_{GRS} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \dots & \alpha_n^{n-k-1} \end{pmatrix} D(w),$$

where $w = (w_1, w_2, \dots, w_n)$ and $w_i = u_i \prod_{j \neq i} (\alpha_i - \alpha_j)$.

III. DIRECT ERASURE-ONLY DECODING

In this section, we propose our novel decoding algorithm for GRS codes, which is referred to as Direct Erasure-only Decoding (in sort, DED). The DED is based on Lagrange interpolation and is able to obtain original message in one step. We firstly elaborate the DED and then present an example to explain it.

A. Algorithm Description

If a $[n, k, d]$ GRS code has a generator matrix being

$$G_{GRS} = \begin{pmatrix} v_1 & v_2 & \dots & v_n \\ v_1 \alpha_1 & v_2 \alpha_2 & \dots & v_n \alpha_n \\ v_1 \alpha_1^2 & v_2 \alpha_2^2 & \dots & v_n \alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ v_1 \alpha_1^{k-1} & v_2 \alpha_2^{k-1} & \dots & v_n \alpha_n^{k-1} \end{pmatrix},$$

then the encoding process is a function

$$\psi: F_q^k \rightarrow F_q^n,$$

namely,

$$(t_0, t_1, \dots, t_{k-1}) \mapsto (t_0, t_1, \dots, t_{k-1}) G_{GRS}.$$

We suppose the k messages $(t_0, t_1, \dots, t_{k-1})$ have been encoded into n codewords $(s_0, s_1, \dots, s_{n-1})$. After a transmission process over an erasure channel, we suppose there are up to $n - k$ erasures in the received codewords. Now, the question is how to recover k messages $(t_0, t_1, \dots, t_{k-1})$ from the received codewords. We may denote k error free symbols as $(s_{\rho_1}, s_{\rho_2}, \dots, s_{\rho_k})$ and define the polynomials as

$$f(x) = \prod_{j=1}^k (x - \alpha_{\rho_j}). \quad (1)$$

Then we have

$$\begin{aligned} f_i(x) &= \frac{f(x)}{(x - \alpha_{\rho_i})} \\ &= \prod_{j=1, j \neq i}^k (x - \alpha_{\rho_j}) \\ &= \sum_{j=0}^{k-1} f_i^j x^j, \end{aligned} \quad (2)$$

where $f_i^j, j = 0, 1, \dots, k-1$ are the coefficients of the polynomial $f_i(x) = \prod_{j=1, j \neq i}^k (x - \alpha_{\rho_j})$.

Theorem 1: If the vector $(s_{\rho_1}, s_{\rho_2}, \dots, s_{\rho_k})$ formed by k error-free symbols is received, then the k source symbols can be reconstructed by

$$t_j = \sum_{i=1}^k \frac{s_{\rho_i}}{v_{\rho_i}} \frac{f_i^j}{f'(\alpha_{\rho_i})},$$

where $j = 0, 1, \dots, k-1$ and $f'(x)$ is the formal derivative of $f(x)$.

Proof: The vector $t = (t_0, t_1, \dots, t_{k-1})$ can be viewed as a polynomial $t(x) = \sum_{j=0}^{k-1} t_j x^j$ by the following mapping

$$\varphi : F_q^k \rightarrow F_q[x],$$

namely,

$$(t_0, t_1, \dots, t_{k-1}) \mapsto t(x) = \sum_{j=0}^{k-1} t_j x^j.$$

Thus, encoded vector is $(v_1 t(\alpha_1), v_2 t(\alpha_2), \dots, v_n t(\alpha_n))$, namely

$$v_i t(\alpha_{\rho_i}) = s_{\rho_i}.$$

The problem to recover k source symbols $(t_0, t_1, \dots, t_{k-1})$ can be changed to determine a polynomial of degree less than k in the presence of the values at k points $\{\rho_i, i = 1, 2, \dots, k\}$ is known, which is actually a typical Lagrange interpolation problem. Based on this observation, we have

$$\begin{aligned} t(x) &= \sum_{i=1}^k \frac{s_{\rho_i} \prod_{j=1, j \neq i}^k (x - \alpha_{\rho_j})}{v_{\rho_i} \prod_{j=1, j \neq i}^k (\alpha_{\rho_i} - \alpha_{\rho_j})} \\ &= \sum_{i=1}^k \frac{s_{\rho_i} f_i(x)}{v_{\rho_i} f_i(\alpha_{\rho_i})}. \end{aligned}$$

We compare the coefficients on both sides

$$t_j = \sum_{i=1}^k \frac{s_{\rho_i}}{v_{\rho_i}} \frac{f_i^j}{f_i(\alpha_{\rho_i})}.$$

The formal derivative of $f(x)$ is

$$f'(x) = (x - \alpha_{\rho_i}) f_i'(x) + f_i(x),$$

and we have

$$f'(\alpha_{\rho_i}) = f_i(\alpha_{\rho_i}).$$

We substitute it into the formula (2) and obtain

$$t_j = \sum_{i=1}^k \frac{s_{\rho_i}}{v_{\rho_i}} \frac{f_i^j}{f'(\alpha_{\rho_i})}.$$

The decoding complexity of our algorithms is $O(k^2)$, which means our scheme is suitable for a low-rate code. ■

B. A Example

For a concrete understanding, consider an example in which we assume $G_{RS} = G_{GRS}(\alpha, w)$ and $n = q - 1$, where $w = (1, 1, \dots, 1)$. Let α be a root of an irreducible polynomial $x^3 + x + 1 = 0$. Then α is a generator of the finite field $GF(2^3)$ defined on that polynomial, which means the first $2^3 - 1$ powers of α can generate all the nonzero elements of $GF(2^3)$. Below are all nonzero elements of $GF(2^3)$ using the power representation and corresponding polynomial representation.

$$\left(\begin{array}{c|c|c|c|c|c|c} 1 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ \hline (0, 0, 1) & (0, 1, 0) & (1, 0, 0) & (0, 1, 1) & (1, 1, 0) & (1, 1, 1) & (1, 0, 1) \end{array} \right).$$

We suppose a $[7, 3, 5]$ RS code is

$$C = \{c(x) = \sum_{i=0}^6 c_i x^i \in F_8[x] | c(\alpha) = c(\alpha^2) = \dots = c(\alpha^4) = 0\}$$

with a generator matrix being

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{12} \end{pmatrix}.$$

After a transmission process over an erasure channel, we receive the codeword $(?, \alpha^3, ?, \alpha^6, ?, ?, 1)$. Hence, the subscripts of three received error-free symbols are $\rho_1 = 2, \rho_2 = 4, \rho_3 = 7$. We also know that the corresponding error-free values are $\alpha_{\rho_1} = \alpha, \alpha_{\rho_2} = \alpha^3, \alpha_{\rho_3} = \alpha^6$. Then we have

$$f(x) = \prod_{j=1}^3 (x - \alpha_{\rho_j}) = x^3 + \alpha^2 x^2 + \alpha^3 x + \alpha^3$$

and the formal derivative of $f(x)$ is

$$f'(x) = x^2 + \alpha^3.$$

It follows that $f'(\alpha) = \alpha^5, f'(\alpha^3) = \alpha^4, f'(\alpha^6) = \alpha^2$. Now, we calculate $f_i(x)$ as follows:

$$f_1(x) = (x - \alpha^3)(x - \alpha^6) = x^2 - \alpha^4 x + \alpha^2.$$

In a similar fashion,

$$f_2(x) = x^2 - \alpha^5 x + 1,$$

$$f_3(x) = x^2 - x + \alpha^4.$$

By the Theorem 1, we can calculate source symbols by

$$t_j = \sum_{i=1}^3 \frac{s_{\rho_i} f_i^j}{f'(\alpha_{\rho_i})},$$

and we obtain $t_0 = 1$, $t_1 = \alpha$, $t_2 = \alpha^2$. Thus the original symbols sent by source are $(1, \alpha, \alpha^2)$.

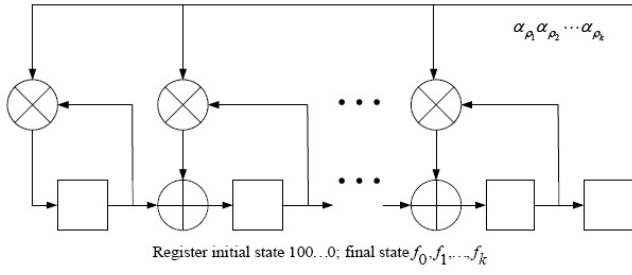


Figure 1. Circuit for calculating $f(x)$

IV. CIRCUIT IMPLEMENTATION

In this section, we provide a circuit implementation for the algorithm. All components such as flip-flops, adders and multipliers are all defined over a finite field of characteristic 2. This suggest that $x = -x$ for every element in the field. Figure 1 [9] illustrates a shift register implementation for the sum representation of the polynomial $f(x)$. We denote $\eta_i = \frac{1}{v_{p_i} f'(\alpha_{p_i})}$ and calculate the terms of $f'(\alpha_{p_i})$ of degrees in odd numbers. That is,

$$\begin{aligned} (f_{2m} x^{2m})' &= 2m f_{2m} x^{2m-1} \\ &= 0 \\ (f_{2m+1} x^{2m+1})' &= (2m+1) f_{2m+1} x^{2m} \\ &= f_{2m+1} x^{2m}. \end{aligned}$$

A circuit implementation depicted in Figure 2 below can be used to efficiently calculate η_i . We elaborate on how the circuit works. In Figure 2, the shift register at the left hand is initialized by $f_t, f_{t-2}, \dots, f_3, f_1$, where $t = 2\lfloor \frac{k+1}{2} \rfloor - 1$. Meanwhile, SR1 and SR2, two shift registers of one stage are initialized by 1. The α_{p_i} will be fed into the input with the switch K_1 on and K_2 off. Consequently, the $\alpha_{p_i}^2$ will be stored in the memory and calculation process begins. As the elapse of time, f_t is the output of the shift register after t clocks, which means the calculation process of $f'(\alpha_{p_i})$ is done. We turn the switch K_2 on and obtain the final output η_i .

To finish the decoding process, we should gain the value of f_i^j in advance. From $f(x) = (x - \alpha_{p_i}) f_i(x)$, we can obtain

$$\sum_{i=0}^k f_i x^i = (x - \alpha_{p_i}) \left(\sum_{j=0}^{k-1} f_i^j x^j \right).$$

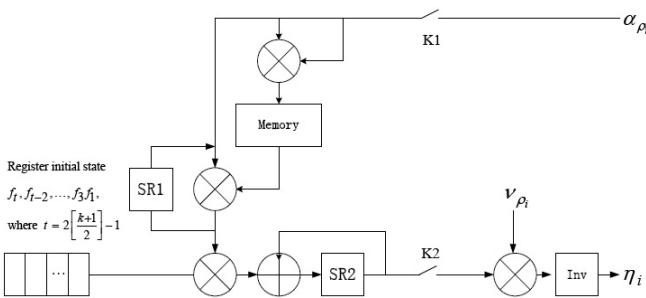


Figure 2. Circuit for calculating η_i

Thus,

$$f_j = \begin{cases} \alpha_{p_i} f_i^0 & j = 0 \\ f_i^{j-1} - \alpha_{p_i} f_i^j & 1 \leq j \leq k-1 \\ f_i^{k-1} & j = k \end{cases}$$

It is followed that

$$\begin{aligned} f_i^{k-1} &= f_k \\ f_i^{k-2} &= f_{k-1} + \alpha_{p_i} f_i^{k-1} \\ &\dots \\ f_i^0 &= f_1 + \alpha_{p_i} f_i^1. \end{aligned}$$

Figure 3 below describes a circuit implementation for the final step processing. The shift register of $k-1$ stages is initialized by the vector $(f_1, f_2, \dots, f_{k-1})$. All the shift registers of one stage are initialized by $f_k = 1$. The source symbols t_i are produced in sequence for $i = k-1, k-2, \dots, 1, 0$.

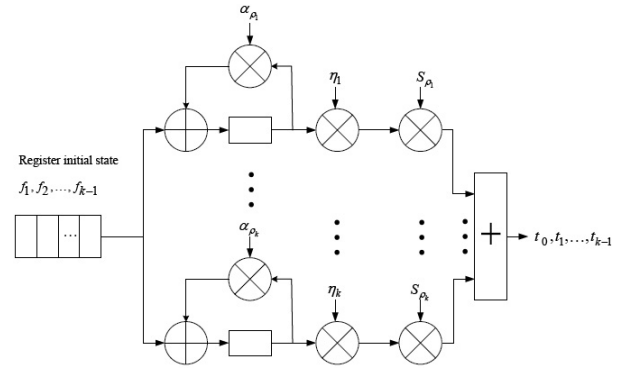


Figure 3. Circuit for erasure-only decoding

V. CONCLUSION

Previous solutions for decoding GRS codes need two steps to obtain original messages - first to recover the codewords and then to obtain the original messages. In this paper, we propose a novel decoding algorithm for GRS codes based on lagrange interpolation, referred to as Direct Erasure-only Decoding. Different from the traditional decoding approaches, DED combines two steps into one, hence significantly improves the efficiency of decoding process for GRS codes. To elaborate the algorithm, we also provide a simply example to explain it. Finally, a detailed circuit implementation for the algorithm over a finite field of characteristic 2 is designed and corresponding workflow is described.

ACKNOWLEDGMENT

The work described in this paper was partially supported by a grant from the Development Program for the Creative Talents in Key Disciplinary Areas of Fudan University.

REFERENCES

- [1] I. Reed and G. Solomon, "Polynomial codes over certain finite fields," *SIAM J. Appl. Math.*, vol. 8, pp. 300-304, 1960.
- [2] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, 1st ed., Prentice Hall, New York, NY, USA, 1995.
- [3] S. Lin and D. J. Costello Jr., *Error Control Coding*, 2nd ed., Prentice Hall, New York, NY, USA, 2004.
- [4] S. V. Fedorenko, "A simple algorithm for decoding Reed-Solomon codes and its relation to the Welch-Berlekamp algorithm," *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 1196-1198, 2005.
- [5] M. Sudan, "Decoding of Reed Solomon codes beyond the error-correction bound," *Journal of Complexity*, vol. 13, no. 1, pp. 180-193, 1997.
- [6] V. Guruswami and M. Sudan, "Improved decoding of reedsolomon and algebraic-geometry codes," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1757-1767, 1999.
- [7] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Transactions on Information Theory*, vol. 49, no. 11, pp. 2809-2825, 2003.
- [8] J.H. Van Lint, *Introduction to Coding Theory*, 3rd ed., Springer-Verlag, 1999.
- [9] Y. Xu and T. Zhang, "Variable shortened-and-punctured Reed-Solomon codes for packet loss protection," *IEEE Transactions on Broadcast*, vol. 48, no. 3, pp. 237-245, 2002.