DOI:10.3969/J.1SSN.100-5137.2015.03.004

Using the division algorithm to decode Reed-Solomon Codes

CHRISTINA Eubanks-Turner[1], MATTHEW Lennon[2], EDUARDO Reynoso[3], BRANDY Thibodeaux[4], AMANDA Urquiza[5], ASHLEY Wheatley[6], DEREK Young[7]

(1.Department of Mathematics, Loyola Marymount University, Los Angeles, 90278, USA
2. Department of Mathematics, The Christ College, Cincinnati, 45219, USA
3.Department of Mathematics, California State University, Los Angeles, 90032, USA
4.Department of Mathematics, University of Louisiana at Lafayette, Lafayette, 70504, USA
5.Department of Mathematics, Arizona State University, Tempe, 85281, USA
6.Department of Mathematical Sciences, Tennessee State University, Nashville, 372091, USA
7.Department of Mathematics, Iowa State University, Ames, 500111, USA)

Abstract: Reed-Solomon codes are some of the most widely used error correcting codes. In this paper we introduce a decoding algorithm which utilizes the division algorithm. We develop theory and provide examples to support the algorithm. Lastly, we prove a theorem on probability related to this decoding algorithm and examine some probabilistic results on when this method is most effective.

Key words: Reed-Solomon; division algorithm; error-correcting codes
CLC number: O 236.2 **Document code:** A **Article ID:** 1000-5137(2015)03-0262-08
MR (2010): 68P30, 11A05, 11A07

1 Introduction

With the technological demands of today's society, digital security has been a topic of study for many researchers[1]. Reed-Solomon (RS) codes are non-binary linear cyclic codes, constructed over finite fields, that detect and correct multiple errors. They were created by Irving Reed and Gustave Solomon in 1960[2]. Reed-Solomon codes have a wide range of applications in data storage, bar codes, data transmission and satellite transmission. The well known Reed-Solomon code applications include CD, DVD and Blu-ray associated media players. The codes were also used in the transmission of photographs from Jupiter, Uranus, and Neptune in the NASA Voyager Program.

Reed Solomon codes are sets of algebraic curves defined by polynomials with a limited range of degrees. An $[n, k]$ Reed-Solomon code of length n and dimension k over a finite field $GF(q^m)$, for some prime number q, is a

Received date: 2014-07-20
Foundation item: This work is a part of the National Science Foundation funded Smooth Transition for the Advancement to Graduate Education (STAGE) for Underrepresented Groups in the Mathematical Sciences Pilot Project (DMS-1043223)
Corresponding author: CHRISTINA Eubanks-Turner, professor, E-mail: ceturuer@lmu.edu

cyclic code of minimum distance d, where $n = q^m - 1$ and $d = n - k + 1$,[3]. A major area of investigation of RS codes involves the development and study of various decoding algorithms of RS codes. There are unique algorithms, which decode received words that contain errors to a unique codeword [4–5], and list decoding algorithms, which output lists of possible codewords [6–8].

In this paper we develop a method of list decoding that utilizes the division algorithm and an array that is similar to the one constructed in standard array decoding. The array that we develop uses a correspondence between error polynomials and polynomials that are remainders of the quotient of received polynomials and the generator polynomial. We call this type of array a remainder correspondence array. In Section 2, we give background relevant to our work. Section 3 contains our main results that allows for decoding using a sub-array of the remainder correspondence array. Also, we give the Division Decoding algorithm and examples illustrating the algorithm. Probabilities and numerical examples related to the algorithm are given to show the practicality of the algorithm.

2 Background Information

In this section, we recall relevant definitions and results needed to develop our theory. For the remainder of the paper, let C be an $[n, k, d]$ RS code over a field $GF(q^m)$, where n is the code length, k is the message length, d is the minimum distance of the code, q is a prime number and $m \in \mathbb{N}$. There are two standard methods of encoding Reed-Solomon codes: the evaluation approach and the generator polynomial approach. In these methods codewords are represented as polynomials. We utilize the generator polynomial approach for our results.

To construct a t-error correcting, RS code C, we use the generator polynomial $g(y) = (y - a)(y - a^2) \dots (y - a^{2t}) \in GF(q^m)[y]$, where a is the primitive element of $GF(q^m)$, and y is an indeterminate over $GF(q^m)$. The codewords over $GF(q^m)$ can be represented by the coefficients of the code polynomials which are obtained by multiplying a message polynomial, $m(y) \in GF(q^m)[y]$, where $\deg(m(y)) < k$. For more details, see

Remark 1 Since Reed-Solomon codes are polynomial codes we use the terms ″word″ and ″polynomial″ interchangeably.

Example 1 Consider the 2-error correcting, $[7, 3, 5]$ RS code C generated by $g(y) = (y - a)(y - a^2)(y - a^3)(y - a^4) = y^4 + (a + 1)y^3 + y^2 + ay + a + 1$, where a is the primitive element of $GF(2^3)$. To encode a message polynomial we use the encoding method described above. For this example note that $k = 3$. Therefore, the message polynomials are of degree less than 3. Let $m(y) = a^2y + 1$. To encode $m(y)$, use the formula, $c(y) = m(y)g(y)$. Then $c(y) = a^2y^5 + (a^2 + a)y^4 + (a^2 + a + 1)y^3 + (a)y^2 + (a^2 + 1)y + (a + 1)$. The corresponding codeword is $c = (0, a^2, a^4, a^5, a, a^6, a^3)$. In binary, $c = (000, 001, 011, 111, 010, 101, 110)$. This is called the block representation of the codeword.

Arrays are used in coset decoding. In this type of decoding a minimum weight codeword is chosen to be the coset leader. We give the definition below.

Definition 1 Let C be a $[n, k, d]$ RS code over $GF(q^m)$. A standard array for C is a $q^{n-k} \times q^k$ array of elements of $GF(q^m)^n$ defined as follows:

1. The first row in the array consists of the codewords of C, starting with the all-zero codeword.

2. Each subsequent row starts with a word $e \in GF(q^m)^n$ of a smallest Hamming weight that has not yet appeared in previous rows, followed by the words $e + c$, where c ranges over all the nonzero codewords in C in their order of appearance in the first row.

Example 2 Consider the $[4, 2]$ binary, linear code $C = \{0000, 1011, 0101, 1110\}$. Then the standard array for C is

$$\begin{pmatrix} 0000 & 1011 & 0101 & 1110 \\ 1000 & 0011 & 1101 & 0110 \\ 0100 & 1111 & 0001 & 1010 \\ 0010 & 1001 & 0111 & 1100 \end{pmatrix}.$$

The following lemma gives a way to relate elements of the array we use which is similar to the standard array. We write each received polynomial $r(y) = c(y) + e(y)$, where $c(y)$ is a code polynomial and $e(y)$ is an error polynomial. Note when $e(y) = 0$, the codeword has been transmitted correctly.

Lemma 1 Let C be a $[n, k]$ RS code over $GF(q^m)$ generated by $g(y)$. Let the received polynomial $r(y) = c(y) + e(y)$, where $c(y)$ is a code polynomial and $e(y)$ is a an error polynomial. Then $\mathcal{R}(y) \equiv e(y)(\mathrm{mod}\, g(y))$, where $\mathcal{R}(y)$ is the remainder of $r(y)/g(y)$ in $GF(q^m)[y]$.

Proof Suppose $m(y)$ is the message that corresponds to $c(y)$, the codeword. Then $c(y) = m(y)g(y)$ and also $c(y) = r(y) - e(y)$. Thus $m(y)g(y) = r(y) - e(y)$, which implies $0 \equiv r(y) - e(y)(\mathrm{mod}g(y))$. Hence, $e(y) \equiv r(y)(\mathrm{mod}\, g(y))$. Also, by the division algorithm over $GF(q^m)$, we may write $r(y) = n(y)g(y) + \mathcal{R}(y)$, for unique $\mathcal{R}(y), n(y) \in GF(q^m)[y]$, where either $\mathcal{R}(y) = 0$ or $\deg(\mathcal{R}(y)) < \deg(g(y))$. Thus, $r(y) \equiv \mathcal{R}(y)(\mathrm{mod}g(y))$. Therefore, $e(y) \equiv r(y) \equiv \mathcal{R}(y)(\mathrm{mod}\, g(y))$ giving the result.

Now we give a theorem that relates remainder polynomials and error polynomials. This is instrumental in our development of creating a decoding array.

Theorem 1 Let C be a $[n, k, d]$ RS code generated by $g(y)$ over $GF(q^m)$. Then there is a bijective correspondence between V and U, where $V = \{\mathcal{R}(y) \mid \mathcal{R}(y)$ is the remainder of $\dfrac{r(y)}{g(y)}$, where $r(y)$ is a received polynomial$\}$ and $U = \{e(y) \mid e(y)$ is an error polynomial of $r(y)$, where $r(y) = c(y) + e(y)$ is a received polynomial$\}$.

Proof For a code C with generator polynomial $g(y)$, consider the mapping

$$T : V \to U$$
$$T : \mathcal{R}(y) \mapsto e(y)(\mathrm{mod}g(y)),$$

where $\mathcal{R}(y) \equiv e(y)(\mathrm{mod}\, g(y))$. From Lemma 1 it follows that T is well-defined. We will show that T is bijective. To show T is injective, suppose $e_1(y)(\mathrm{mod}g(y)) \equiv e_2(y)(\mathrm{mod}g(y))$. By Lemma 1, $\mathcal{R}_1(y) \equiv e_1(y)(\mathrm{mod}g(y))$ and $\mathcal{R}_2(y) \equiv e_2(y)(\mathrm{mod}g(y))$, for some $\mathcal{R}_1(y), \mathcal{R}_2(y) \in V$. Since $e_1(y)(\mathrm{mod}g(y)) \equiv e_2(y)(\mathrm{mod}g(y))$, by transitivity, we have $\mathcal{R}_1(y) \equiv \mathcal{R}_2(y)(\mathrm{mod}g(y))$. Since $\mathcal{R}_1(y), \mathcal{R}_2(y) \in V$ are remainders, by the uniqueness of the Division Algorithm over $GF(q^m)$, $\mathcal{R}_1(y) = \mathcal{R}_2(y)$.

To see that T is surjective, note that, by Lemma 1, for every error polynomial $e(y)$ there is a remainder polynomial of $\dfrac{r(y)}{g(y)}$ is equivalent to $e(y)$, where $r(y)$ is a received polynomial. Therefore T is surjective.

We now introduce a lemma that we use to count the number of possible errors in any received word. This number gives the size of the array we create, which we use to decode. Later, we introduce a theorem that will allow us to decrease the size of the array.

Lemma 2 Given a $[n, k]$ t- error correcting RS code over $GF(q^m)$ the order of the set of error polynomials is given by

$$\phi = n\binom{n}{1} + n^2\binom{n}{2} + \cdots + n^t\binom{n}{t} = \sum_{i=0}^{t} n^i\binom{n}{i}. \tag{1}$$

Proof Fix i such that $0 \leqslant i \leqslant t$. There are $\binom{n}{i}$ ways to pick i positions out of n. Also, there are n^i possible errors that occur in exactly i positions. Since there are n ways to receive a symbol in error, for a received word with errors in exactly i positions, there are n^i choices. Then the total number of errors possible in the code is $\phi = n\binom{n}{1} + n^2\binom{n}{2} + \cdots + n^t\binom{n}{t} = \sum_{i=0}^{t} n^i\binom{n}{i}$.

Next, we define the remainder correspondence array which we use for decoding. This array makes use of the correspondence given in Theorem 1.

Definition 2 Let C be a $[n, k, d]$ RS code over $GF(q^m)$. A remainder correspondence array for C is a $\phi \times 2$ array such that the first column consists of all possible remainders $\mathcal{R}(y)$ of $r(y)/g(y)$, where $r(y)$ is a received word and the second column consists of corresponding errors $e(y)$ such that the elements in the ith row are equivalent modulo $g(y)$, for all i, $1 \leqslant i \leqslant \phi$.

3 Results

A drawback to utilizing decoding methods that use arrays is a decrease in decoding efficiency. Since this type of decoding requires a search of a list, the time to decode depends on the computing system used. To make our decoding algorithm more efficient, we utilize the following theoretical finding.

Theorem 2 Suppose C is an $[n, k]$ RS code over the finite field $GF(q^m)$. If $r(y) = \sum_{i=0}^{n-1} r_i y_i$ is received message with at most t errors occurring in coefficients r_i, for some $i, 0 \leqslant i \leqslant 2t-1$, where $t = \dfrac{n-k}{2}$, then $\mathcal{R}(y) = e(y)$, where $\mathcal{R}(y)$ is the remainder of $r(y)/g(y)$ and $e(y)$ is the error polynomial of $r(y)$.

Proof By the Division Algorithm, $r(y) = \mathcal{Q}(y)g(y) + \mathcal{R}(y)$, for unique $\mathcal{Q}(y), \mathcal{R}(y) \in GF(q^m)[y]$, with $\mathcal{R}(y) = 0$ or $\deg(\mathcal{R}(y)) < \deg(g(y)) = 2t$. Also, since $e(y)$ is the error polynomial associated to $r(y)$, we have, $r(y) = c(y) + e(y) = m(y)g(y) + e(y)$, where $m(y)$ is the message being transmitted and $c(y)$ is the codeword. So this implies $(m(y) - \mathcal{Q}(y))g(y) = \mathcal{R}(y) - e(y)$. Thus, $g(y) \mid \mathcal{R}(y) - e(y)$. By assumption since at most t errors in $r(y)$ occur in coefficients of terms y^i, for some $i, 0 \leqslant i \leqslant 2t-1$, $\deg(e(y)) < 2t$. Also by the Division algorithm, $\deg(\mathcal{R}(y)) < 2t$ and so we have $\deg(\mathcal{R}(y) - e(y)) < 2t$. Therefore, $\mathcal{R}(y) - e(y) = 0$ and so $\mathcal{R}(y) = e(y)$.

Remark 2 The previous theorem can be stated as: Suppose C is an $[n, k]$ RS code over the finite field $GF(q^m)$. If $r(y)$ is a received word with at most t errors in the first $2t$ positions, then $\mathcal{R}(y) = e(y)$, where $\mathcal{R}(y)$ is the remainder of $r(y)/g(y)$ and $e(y)$ is the error polynomial of $r(y)$.

The previous theorem allows us to decode some words without utilizing the remainder correspondence array. To see how this impacts efficiency, we have the following lemma.

Lemma 3 Given a t- error correcting, $[n, k]$ RS code over $GF(q^m)$ the order of the set of error polynomials that occur in the first $2t$ positions is given by

$$\theta = n\binom{2t}{1} + n^2\binom{2t}{2} + \cdots + n^t\binom{2t}{t} = \sum_{i=0}^{t} n^i\binom{2t}{i}. \tag{2}$$

Proof The proof is similar to the proof of Lemma 2.

Remark 3 1. Notice that if the received word's errors occur in the first $2t$ position, by Theorem 2 $\mathcal{R}(y)$ has at most t terms since there are at most t errors.

2. By Theorem 2 we can adjust the array defined in Definition 2.1 to only consider the $(\phi - \theta) \times 2$ array whose first column consists of all remainders which have greater than t terms and whose second column has the corresponding errors.

As mentioned in the previous section, for a t-error correcting $[n, k]$ Reed-Solomon code, we have a code polynomial $c(y)$, which has the form $c(y) = m(y)g(y)$, where $m(y)$ is the message polynomial of degree less than k and $g(y)$ is the generator polynomial of the code C. We use Theorem 1 and Theorem 2 to decode a received word $r(y)$. We now outline the steps of the Division Algorithm Decoding method. **Division Algorithm Decoding**

Input: Received word $r(y) \in GF(q^m)[y]$.

1. Divide $r(y)$ by $g(y)$:

2. Receive the remainder $\mathcal{R}(y)$ and the quotient, $\mathcal{Q}(y)$.

3. (a) If the number of nonzero terms in $\mathcal{R}(y) \leqslant t$, then

Output: $\mathcal{Q}(y)$, which is the message.

(b) Otherwise do 4

4. Scan the given $(\phi - \theta) \times 2$ remainder correspondence array to find $\mathcal{R}(y)$ and the corresponding $e(y)$ such that $e(y) \equiv \mathcal{R}(y) (\mathrm{mod}\, g(y))$.

5. Compute $r(y) - e(y)$

Output: Code polynomial $c(y) \in GF(q^m)[y]$.

3.1 Numerical Examples

Since a ″by hand″ computation can be very time consuming, we utilize GAP computer programming system. For more information on GAP, see [9]. We apply the quotient remainder command in GAP to find the remainder and quotient of the received word when divided by the generator polynomial. Now we give some examples to illustrate Division Algorithm Decoding.

Example 3 Let C be the $[7, 3, 5]$ RS code. This code can correct up to two errors. Also from Example 1, we have, $g(y) = y^4 + a^3y^3 + y^2 + ay + a^3$. Consider the received word $r = (0, a^2, a^4, a^5, a, a, a^4)$, which corresponds to the received polynomial $r(y) = a^2y^5 + a^4y^4 + a^5y^3 + ay^2 + ay + a^4$. Then for $\dfrac{r(y)}{g(y)}$, we have remainder $\mathcal{R}(y) = a^5y + a^6$ and quotient $\mathcal{Q}(y) = \alpha^2y + 1$. Since $\mathcal{R}(y)$ has two nonzero terms and C is a 2-error correcting

code, we utilize Step 3(a) of Division Algorithm Decoding. Therefore the message is $m(y) = \alpha^2 y + 1$ from Example 1.

As the previous example did not require a remainder correspondence array, we now give examples that require the use of a remainder correspondence array.

Example 4 Let C be the 1-error correcting, $[3, 1]$ RS code over $GF(2^2)$. Let α be the primitive element of $GF(2^2)$. Since C is 1-error correcting we set $g(y) = (y - \alpha)(y - \alpha^2) = y^2 + y + 1$. Say we receive the word $r(y) = y^2 + \alpha y + \alpha$. Using the Decoding Algorithm we compute the remainder $\mathcal{R}(y)$ of $\dfrac{r(y)}{g(y)}$, which is $\mathcal{R}(y) = \alpha^2 y + \alpha^2$. Since the remainder has 2 terms, we use the remainder correspondence array given in Table 1 to decode. Therefore, from the array we have $e(y) = \alpha^2 y^2$ and so the codeword is $c(y) = r(y) - e(y) = \alpha y^2 + \alpha y + \alpha$.

Table 1 Error Correspondence Array for the $[3, 1]$ code C

$\mathcal{R}[y]$	$e(y)$
$y + 1$	y^2
$\alpha y + \alpha$	αy^2
$\alpha^2 y + \alpha^2$	$\alpha^2 y^2$

Example 5 Let C be the $[7, 3]$ Reed-Solomon code then we use the generator polynomial $g(y) = y^4 + \alpha^3 y^3 + y^2 + \alpha y + \alpha^3$. Let $c(y) = \alpha^2 y^5 + \alpha^4 y^4 + \alpha^5 y^3 + \alpha y^2 + (\alpha^6) y + \alpha^3$ represent the transmitted code polynomial. Suppose $r(y) = \alpha y^5 + \alpha^4 y^4 + \alpha^5 y^3 + \alpha y^2 + \alpha^3 y + \alpha^3$ is the received polynomial. Then the remainder of $\dfrac{r(y)}{g(y)}$ is $\mathcal{R}(y) = \alpha^6 y^3 + \alpha^4 y^2 + \alpha^6 y + \alpha^3$. Since $\mathcal{R}(y)$ has four nonzero terms, we utilize Step 4 of Division Algorithm Decoding. Now we can look at the $(\phi - \theta) \times 2 = 756 \times 2$ array and find the remainder listed. From the array we find $\mathcal{R}(y) \equiv \alpha^4 y^5 + \alpha^4 y \pmod{g(y)}$ and so $r(y) - (\alpha^4 y^5 + \alpha^4 y) = c(y)$.

3.2 Probabilities Related to Division Algorithm Decoding

As we saw in Examples 4 and 5 demonstrate the efficiency of Division Algorithm decoding depends on the number of nonzero terms in the remainder. That is, we can more efficiently correct any received word that has up to t errors in the first $2t$ positions. Here we discuss the probability of up to t errors occurring in the first $2t$ positions. To do this we examine certain transmission channels commonly used when transmitting RS codes over finite fields see [10].

Definition 3 Consider an alphabet with b symbols. A b-ary symmetric channel is a transmission where each symbol transmitted has the same probability p of being received in error; meaning $1 - p$ is the probability the symbol is received correctly. Also each of the $b - 1$ possible errors are equally likely.

We provide the following theorem to show the probability of up to t errors occurring in the first $2t$ positions, which by Theorem 2 will allow us to utilize Step 3(a) of Division Algorithm Decoding.

Theorem 3 For a $[n, k]$ RS Code, let p be the probability that a symbol was transmitted incorrectly. The probability that up to t errors occur in the first $2t$ positions of a received word is

$$(1 - p)^n + \frac{1}{\binom{n}{2t}} \sum_{i=1}^{t} \binom{2t}{i} p^i (1 - p)^{n-i}. \tag{3}$$

Proof Out of n positions, there are $\binom{n}{2t}$ different ways to choose $2t$ positions from n positions. However, the likelihood of having the first $2t$ positions occur is $\frac{1}{\binom{n}{2t}}$. There are $\binom{2t}{i}$ ways to choose i errors in $2t$ positions, where $1 \leqslant i \leqslant t$. If no errors occur the probability is $(1-p)^n$. In addition, the probability of i errors being received is $p^i(1-p)^{n-i}$. Therefore, the probability of correcting i errors that satisfy Theorem 2 is: $\frac{1}{\binom{n}{2t}}\binom{2t}{i}p^i(1-p)^{n-i}$ for each i, $1 \leqslant i \leqslant t$. As a result, the probability that t errors occur in the first $2t$ positions is $(1-p)^n + \frac{1}{\binom{n}{2t}}\sum_{i=1}^{t}\binom{2t}{i}p^i(1-p)^{n-i}$.

Table 2 gives us some numerical probabilities of receiving words that satisfy the properties from Theorem 3 for three finite fields. The table shows that as the field grows and the length of the codewords increase, the probability of having t errors occur in the first $2t$ positions decreases. This implies that the Division Algorithm Decoding becomes less efficient as n and t increase. Thus when using this decoding algorithm, one is more dependent on the remainder correspondence array for decoding.

Table 2 Probability of receiving a word satisfying Theorem 3

Field	Errors(t)	Probability
$GF(2^3)$	1	$7.94728597005e-07$
	2	$4.86373901367e-06$
	3	0.000520706176758
$GF(2^4)$	1	$1.11517937741e-18$
	2	$1.76331781899e-18$
	3	$1.31655287183e-17$
$GF(2^5)$	1	$2.48146603058e-47$
	2	$2.59939139395e-47$
	3	$4.00477926503e-47$

4 Conclusion

In the future, we plan to further investigate the decoding method in order to further develop or generalize the Division Decoding Algorithm. We also would like to investigate the use of Division Algorithm decoding over error and erasure channels. Error-erasure channels model situations where both an error and an erasure occurs[11,12]. We plan to look at some error and erasure channels, study current decoding algorithms over these channels and see how we may implement the Division Decoding Algorithm over these channels. We hope to develop an algorithm for decoding error-erasure channels that is heavily based on the Division Algorithm.

References:

[1] NYAMORADI N, JAVIDI M. Qualitative and bifurcation analysis using a computer virus model with a saturated recovery function[J]. Journal of Applied Analysis and Computation, 2012, 2(3):305-313.

[2] REED G, SOLOMON I. Polynomial codes over certain finite[J]. SIAM Journal of Applied Math, 1960, 8:300-304.

[3] PLESS V. Introduction to the Theory of Error-Correcting Codes[M]. New York: John Wiley & Son Inc, 1998.

[4] BERLEKAMP E R. Algebraic Coding Theory[M]. Revised ed. CA: Aegean Park Press, 1984.

[5] MASSEY J L. Shift-register synthesis and BCH decoding[J]. IEEE Transaction on Information Theory, 1969, T-15(1):122-127.

[6] SUDAN M. Decoding Reed-Solomon codes beyond the error-correction bound[J]. Journal of Complexity, 1997, 13(1):180-193.

[7] GURUSWAMI V, RUDRA A. Explicit Codes Achieving List Decoding Capacity: Error-Correction With Optimal Redundancy[J]. IEEE Transaction on Information Theory, 2008, 54(1):135-150.

[8] GURUSWAMI V. List decoding of error-correcting codes, Lecture Notes in Computer Science, no. 3282[M]. New York: Springer, 2004.

[9] The GAP Group. GAP Manuals Online[EB/OL]. Retrieved from http://www.gap-system.org/Doc/manuals.html, 2012.

[10] RYAN W E, LIN S. Channel Codes: Classical and Modern[M]. New York: Cambridge University Press, 2009.

[11] DANA A F, GOWAIKAR R, PALANKI R, et al. Capacity of Wireless Erasure Networks[J/OL]. IEEE Transactions on Information Theory, 2006, 52(3):789-804. http://authors.library.caltech.edu/4122/1/DANieeetit06a.pdf.

[12] SUNDARAVARADHAN S P. Complexity-feedback Tradeoffs and Capacity Results for Packet Erasure Networks[D/OL]. Notre Dame: University of Notre Dame, 2010, http://etd.nd.edu/ETD-db/theses/available/etd-04152010-204112/unrestricted/PuducheriS042010D.pdf.

使用除法算法对里德 - 所罗门编码进行解码

CHRISTINA Eubanks-Turner[1], MATTHEW Lennon[2], EDUARDO Reynoso[3]

BRANDY Thibodeaux[4], AMANDA Urquiza[5], ASHLEY Wheatley[6], DEREK Young[7]

(1. 洛约拉 · 玛丽蒙特大学，数学系，洛杉矶 90278;
2. 克莱斯特学院，数学系，辛辛那提 45219;
3. 加利福尼亚州立大学洛杉矶分校，数学系，洛杉矶 90032;
4. 路易斯安那大学拉斐特分校，数学系，拉菲特，美国 70504;
5. 亚利桑那州立大学，数学系，滕比谷，美国 85281;
6. 田纳西州立大学，数学科学系，那什维尔，美国 372091;
7. 爱荷华州立大学，数学系，埃姆斯，美国 500111)

摘　要：里德 - 所罗门编码是最广泛使用的纠错码之一. 介绍一种使用除法算法的解码方法，发展该算法的理论并讨论支持该算法的例子. 最后，证明与该解码算法有关的一个概率上的定理，关于何时该方法是最有效的，得出一些概率上的结果.

关键词：里德 - 所罗门编码；除法算法；纠错码

(责任编辑：冯珍珍，顾浩然)