Research Article

A Novel Decoding Method for the Erasure Codes

Dan Tang ^{1,2} and **Hongliang Cai** ^{1,2}

¹Software Engineering Institute, Chengdu University of Information Technology, Chengdu 610225, China

²Sichuan Key Laboratory of Software Automatic Generation and Intelligent Service, Chengdu 610225, China

Correspondence should be addressed to Hongliang Cai; caihl@cuit.edu.cn

Received 15 May 2021; Accepted 6 July 2021; Published 16 July 2021

Academic Editor: Zhili Zhou

Copyright © 2021 Dan Tang and Hongliang Cai. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The erasure codes are widely used in the distributed storage with low redundancy compared to the replication method. However, the current research studies about the erasure codes mainly focus on the encoding methods, while there are few studies on the decoding methods. In this paper, a novel erasure decoding method is proposed; it is a general decoding method and can be used both over the multivariate finite field and the binary finite field. The decoding of the failures can be realized based on the transforming process of the decoding transformation matrix, and it is convenient to avoid the overburdened visiting problem by tiny modification of the method. The correctness of the method is proved by the theoretical analysis; the experiments about the comparison with the traditional methods show that the proposed method has better decoding efficiency and lower reconstruction bandwidth.

1. Introduction

Erasure coding is a widely used method in the distributed storage system to protect against the failures of the storage nodes; it can provide better failure tolerance and much lower storage redundancy. The original information is first divided into k blocks; then, the k blocks are encoded into n blocks using the encoding method; when there are less than $n-k$ blocks missed, the lost blocks can be recovered by the decoding method. There are a lot of research studies about the encoding methods, while only a few studies focus on the decoding methods.

The typical erasure codes are RS (Reed–Solomon) erasure code and the array codes [1, 2]. The most important property for the RS erasure code [3–5] is that it is MDS (Maximum Distance Separable) code, which means the RS code can provide the optimal failure recovery ability in a certain data redundancy. A storage system based on the RS erasure code can achieve the optimal tradeoff between the storage redundancy and the ability of the failure tolerance. Moreover, the tolerated failures are not restricted as the fixed number as the array codes.

However, the typical decoding method of the RS code is the equation solving method [6] over the finite field $GF(q)$,

and it is equivalent to the matrix inversion method and has very high computational complexity. Therefore, many research studies make efforts to optimize the decoding computation of the RS code. Some research studies improve the decoding efficiency by developing the improved decoding method, for example, the merge decoding method in [7] can be used for RS decoding, but it is in fact the modified method of the traditional equation solving method. Some other research studies improve the decoding efficiency by speeding up the operations over the finite field, for example, a finite field operation library GF-Complete in [8] is developed by Professor James S. Plank; it can speed up the multiplication operation on the finite field by optimizing the bottom SSE instruction, but it does not make any fundamental improvement to reduce the decoding complexity. In this paper, we will improve the decoding efficiency by developing a novel decoding method, and the method of optimizing the operations over the finite field is out of the scope of our research.

The array codes are another kind of the erasure codes widely used in the storage system. Since the two-dimensional coding structure of the array code is highly compatible with the disk array layout in some storage systems, the array codes

are often used in the RAID system. The array codes with MDS property can tolerate only 2 or 3 failures [9, 10], while the array codes tolerating more failures are usually non-MDS array codes [3, 11, 12].

The advantage of the array codes is that the computational efficiency of the encoding and decoding is high because they are performed over $GF(2)$, which means only the XOR operations are used in the encoding and the decoding.

The typical decoding method of the array codes is the loop iterative method [9–14]; most of the array codes use it as the decoding method, such as X code [10], EVENODD code [9], and RDP code [15]. The decoding is realized by continuously selecting the generating equation with only one missing data block, and the missing data block is recovered from the renewed generating equations of the check blocks. In general, the iterative method can be used for most of the array codes constructed from the perspective of the geometry, and it cannot apply for the array codes constructed from the perspective of the algebra [16, 17].

The equation solving method in [6] not only can be used for the RS code but it also can be used for other erasure codes; it is in fact a general decoding method over the field $GF(q)$. And, the loop iteration method can be regarded as the general decoding method for different array codes over the field $GF(2)$. There are only few research studies about the general decoding method for the different erasure codes over $GF(2)$ and $GF(q)$. The matrix decoding method in [18] over the binary field $GF(2)$ and the merging decoding method in [7] over the field $GF(2^m)$ are two typical general decoding methods. The advantage of the matrix decoding method in [18] is that the matrix inversion operation is not necessary in the decoding process, and the disadvantage is that it only can deal with the case that the invalid blocks are the original data blocks. When there are some original blocks and check blocks invalid, it needs firstly to recover the original blocks and then recalculate the lost check blocks again. It cannot decode directly to recover the check blocks directly in the decoding process. The merge decoding method in [7] can be regarded as the modified method of the traditional equation solving method in [6], which do not actually eliminate the matrix inversion operation. So the decoding performance of the method is gradually closer to the equation solving method while the failures increasingly reach the upper bound of the failure tolerance.

The organization of this paper is as follows: Section 2 is the preliminary for the proposed method. Section 3 proposes the detailed decoding process, and the proof and the analysis are also given out. And, Section 4 shows some experiments about the comparison between the proposed method and other decoding methods. At last, the conclusion is made in Section 5.

2. Preliminary

The finite field is an important concept in the coding theory. The basic concepts and important properties about the finite field can be found in [19]. In this paper, the finite field is

constructed as in [20]. It has the property that the inverse element of the addition operation is itself, so the finite field used in this paper is $GF(2^m)$, which is an important pre-mising constituent of the proposed method in this paper.

2.1. The Generator Matrix and the Check Matrix. In the (n, k) linear block code, the original information is divided into k symbols; then, the k symbols are encoded into n symbols which are called as the codeword.

The (n, k) linear code of dimension k and block length n over the $GF(q)$ is in fact a k -dimension subspace $GF(q^k)$ of the space $GF(q^n)$. And, the k -dimension subspace can be completely spanned by the base composed of k linear independent vectors.

Assume that the base of the subspace is shown as (1), where $g_{i,j} \in GF(q)$, i and j are integers, $0 \leq i \leq k-1$ and $0 \leq j \leq n-1$:

$$\left. \begin{aligned} C_0 &= (g_{0,0}, g_{0,1}, \dots, g_{0,n-1})^T, \\ C_1 &= (g_{1,0}, g_{1,1}, \dots, g_{1,n-1})^T, \\ &\vdots \\ C_{k-1} &= (g_{k-1,0}, g_{k-1,1}, \dots, g_{k-1,n-1})^T. \end{aligned} \right\} \quad (1)$$

Joint the vectors into a matrix:

$$\mathbf{G} = \begin{pmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,k-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n-1,0} & g_{n-1,1} & \cdots & g_{n-1,k-1} \end{pmatrix}. \quad (2)$$

Suppose $\mathbf{d}$ is the vector about the original information symbols:

$$\mathbf{d} = (d_0, d_1, \dots, d_{k-1})^T. \quad (3)$$

Based on the property [21] that any codeword of the (n, k) linear block code can be generated by the linear combination of the base, so the codeword $C = (c_0, c_1, \dots, c_{n-1})$ can be generated by multiplying the matrix $\mathbf{G}$ by the vector $\mathbf{d}$:

$$\mathbf{C} = \mathbf{G} * \mathbf{d} = \begin{pmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,k-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n-1,0} & g_{n-1,1} & \cdots & g_{n-1,k-1} \end{pmatrix} \cdot \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ d_{k-1} \end{pmatrix}. \quad (4)$$

The $n \times k$ matrix $\mathbf{G}$ is called as the generator matrix. And, its rank is k , where $g_{i,j} \in GF(q)$. The generator matrix is not unique, and any $n \times k$ matrix $\mathbf{G}$ with rank k can be used as the generator matrix of the (n, k) linear block code.

There is a linear relationship between the elements of $C = (c_0, c_1, \dots, c_{n-1})$:

$$\left. \begin{array}{ccccccccc} h_{0,0}c_0 & + & h_{0,1}c_1 & + & \cdots & + & h_{0,k-1}c_{k-1} & = & c_k \\ h_{1,0}c_0 & + & h_{1,1}c_1 & + & \cdots & + & h_{1,k-1}c_{k-1} & = & c_{k+1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ h_{n-k-1,0}c_0 & + & h_{n-k-1,1}c_1 & + & \cdots & + & h_{n-k-1,k-1}c_{k-1} & = & c_{n-1} \end{array} \right\}. \quad (5)$$

The finite field has the property that the inverse element of the addition operation is itself [20]. So (5) can be turned into

$$\left. \begin{array}{ccccccccccc} h_{0,0}c_0 & + & h_{0,1}c_1 & + & \cdots & + & c_k & 0 & \cdots & 0 & = & 0 \\ h_{1,0}c_0 & + & h_{1,1}c_1 & + & \cdots & + & 0 & c_{k+1} & \cdots & 0 & = & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ h_{n-k-1,0}c_0 & + & h_{n-k-1,1}c_1 & + & \cdots & + & 0 & 0 & \cdots & c_{n-1} & = & 0 \end{array} \right\}. \quad (6)$$

Equation (6) can be expressed as $\mathbf{H} \cdot \mathbf{C} = 0$:

$$\begin{aligned} & \left(\begin{array}{cccccccc} h_{0,0} & h_{0,1} & \cdots & h_{0,k-1} & 1 & 0 & \cdots & 0 \\ h_{1,0} & h_{1,1} & \cdots & h_{1,k-1} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{n-k-1,0} & h_{n-k-1,1} & \cdots & h_{n-k-1,k-1} & 0 & 0 & \cdots & 1 \end{array} \right) \cdot \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \\ & H = \begin{pmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,k-1} & 1 & 0 & \cdots & 0 \\ h_{1,0} & h_{1,1} & \cdots & h_{1,k-1} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{n-k-1,0} & h_{n-k-1,1} & \cdots & h_{n-k-1,k-1} & 0 & 0 & \cdots & 1 \end{pmatrix}. \end{aligned} \quad (7)$$

The matrix $\mathbf{H}$ is called as the check matrix, and the i th row corresponds to the check equation for the i th check symbol independently, so the different rows can be swapped. The j th column is the coefficients related to the j th symbol, so the different columns cannot be swapped because it will corrupt the linear combination relationship.

The basic idea of the decoding method based on the equation solving is as follows. Suppose that there are $n - k$ symbols that are invalid in the codeword; extract the responding rows in matrix G to get the new matrix $\overline{\mathbf{G}}$, $\overline{\mathbf{C}} = \overline{\mathbf{G}} \cdot \mathbf{d}$, so $\overline{\mathbf{C}}$ is the remaining k valid symbols. As any k rows of the matrix G are linear independent, so the matrix $\overline{\mathbf{G}}$ is reversible, and the inverse matrix is denoted as $\overline{\mathbf{G}}'$. Then, multiply $\overline{\mathbf{G}}'$ in both sides of $\overline{\mathbf{C}} = \overline{\mathbf{G}} \cdot \mathbf{d}$, and we can get

$\overline{\mathbf{G}}' \cdot \overline{\mathbf{C}} = \overline{\mathbf{G}}' \cdot \overline{\mathbf{G}} \cdot \mathbf{d}$. As $\overline{\mathbf{G}}' \cdot \overline{\mathbf{G}} = I$, so the original data vector can be obtained by $\mathbf{d} = \overline{\mathbf{G}}' \cdot \overline{\mathbf{C}}$.

2.2. Linear Relation Matrix and Its Properties. The linear relation matrix is an important concept in the proposed decoding method; the definition of the linear relation matrix is shown as below.

Definition 1. Linear relation matrix.

From the generation of the n symbols in the codeword of the (n, k) linear block code, we can know that they can be represented as the linear combination of the n symbols, as shown in the following equation:

$$\left. \begin{array}{ccccccc} c_0 & = & a_{0,0}c_0 & + & a_{0,1}c_1 & + & \cdots & + & a_{0,n-1}c_{n-1} \\ c_1 & = & a_{1,0}c_0 & + & a_{1,1}c_1 & + & \cdots & + & a_{1,n-1}c_{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{n-1} & = & a_{n-1,0}c_0 & + & a_{n-1,1}c_1 & + & \cdots & + & a_{n-1,n-1}c_{n-1} \end{array} \right\}. \quad (8)$$

The nonhomogeneous linear equation system is called as the linear relation equation system, and the matrix of the coefficients can be called as the linear relation matrix.

Suppose that the linear relation matrix for the (n, k) linear block code is $\mathbf{W}$:

$$\mathbf{W} = \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,n-1} \end{pmatrix}, \quad (9)$$

where $a_{i,j} \in GF(q)$, i and j are the integers, $0 \leq i \leq n-1$ and $0 \leq j \leq n-1$. We can get the linear relationship of the i th symbol c_i by the i th row vector of $\mathbf{W}$.

Take the (5,3) linear code over $GF(2^3)$ for example; suppose that the linear relation matrix of the code is $\mathbf{W}_1$:

$$\mathbf{W}_1 = \begin{pmatrix} 0 & 7 & 6 & 2 & 0 \\ 2 & 0 & 7 & 3 & 0 \\ 3 & 2 & 0 & 4 & 0 \\ 7 & 6 & 5 & 0 & 0 \\ 5 & 2 & 7 & 0 & 0 \end{pmatrix}. \quad (10)$$

So, we can get $c_0 = 7c_1 + 6c_2 + 2c_3$, $c_1 = 2c_0 + 7c_2 + 3c_3$, and so on.

There are two properties about the transformation of the linear relation matrix, which are important for the proposed decoding method.

Property 1. The result of any elementary row transformation on the linear relation matrix is no longer a linear relation matrix.

Proof. In this paper, we propose a novel method for the erasure codes for the storage system which is different from the replication method, so the erasure codes do not include the repetition code which is in fact the replication method; there are three cases for the elementary row transformation over $GF(q)$ [21]: $\square$

Case 1. Multiply one row of the matrix by t , where $t \in GF(q)$ and $t \neq 0$;

Case 2. Multiply one row of the matrix by t and add the result to another row, where $t \in GF(q)$ and $t \neq 0$;

Case 3. Exchange any two rows in the matrix.

For the three row transformation cases, if the transformed matrix is still the relation matrix or not is discussed, respectively.

For Case 1, from the definition of the linear relation matrix, the i th row in the linear relation matrix represents the linear combination about c_i , i is an integer and $0 \leq i \leq n-1$:

$$c_i = a_{i,0}c_0 + a_{i,1}c_1 + \cdots + a_{i,n-1}c_{n-1}. \quad (11)$$

Multiply the i th row by nonzero t ; the i th row is changed into $(t \cdot a_{i,0}, t \cdot a_{i,1}, \dots, t \cdot a_{i,n-1})$. Multiply t in the both sides of (11):

$$t \cdot c_i = t \cdot (a_{i,0}c_0 + a_{i,1}c_1 + \cdots + a_{i,n-1}c_{n-1}). \quad (12)$$

Since t is nonzero, obviously,

$$c_i \neq t \cdot (a_{i,0}c_0 + a_{i,1}c_1 + \cdots + a_{i,n-1}c_{n-1}). \quad (13)$$

It is clear that the i th symbol cannot be represented by the renewed i th row vector $(t \cdot a_{i,0}, t \cdot a_{i,1}, \dots, t \cdot a_{i,n-1})$, so we can derive that multiplying the row of the matrix by nonzero t makes the matrix no longer a linear relation matrix.

For Case 2, multiply the j th row of the matrix by nonzero t and add the result to the i th row, and the i th row is changed into $(t \cdot a_{j,0} + a_{i,0}, \dots, t \cdot a_{j,n-1} + a_{i,n-1})$. As $c_j = a_{j,0}c_0 + a_{j,1}c_1 + \cdots + a_{j,n-1}c_{n-1}$, multiply the both sides by nonzero t and add (11), and we can get

$$t \cdot c_j + c_i = (t \cdot a_{j,0} + a_{i,0})c_0 + \cdots + (t \cdot a_{j,n-1} + a_{i,n-1})c_{n-1}, \quad (14)$$

where i and j are integers and $0 \leq i$ and $j \leq n-1$.

Since $t \neq 0$, thus $t \cdot c_j + c_i \neq c_i$. c_i obviously cannot be represented by the row vector $(t \cdot a_{j,0} + a_{i,0}, \dots, t \cdot a_{j,n-1} + a_{i,n-1})$, so the transformed matrix in Case 2 will not be a linear relation matrix.

For Case 3, from the definition of the linear relation matrix, the i th row of the matrix represents the i th symbol c_i , and it cannot be used to represent the j th symbol c_j , and it is obvious that exchanging any two rows of the matrix makes it no longer a linear relation matrix.

Property 2. For the vector $\mathbf{v} = (v_0, v_1, \dots, v_{n-1})$, if $v_0c_0 + v_1c_1 + \cdots + v_{n-1}c_{n-1} = 0$, c_i, v_i , and $t \in GF(q)$. Multiply the vector $\mathbf{v}$ by t and add the result to i th row of the linear relation matrix $\mathbf{W}$, and it is still a linear relation matrix.

Proof. From the definition of the linear relation matrix, the i th row vector $(a_{i,0}, a_{i,1}, \dots, a_{i,n-1})$ is the linear combination of c_i , $c_i = a_{i,0}c_0 + a_{i,1}c_1 + \cdots + a_{i,n-1}c_{n-1}$.

After multiplying the vector $\mathbf{v}$ by t and adding the result to i th row of $\mathbf{W}$, the i th row of the renewed linear matrix becomes $(a_{i,0} + tv_0, \dots, a_{i,n-1} + tv_{n-1})$.

As $v_0c_0 + v_1c_1 + \cdots + v_{n-1}c_{n-1} = 0$, so we can know that

$$tv_0c_0 + tv_1c_1 + \cdots + tv_{n-1}c_{n-1} = t \cdot 0 = 0. \quad (15)$$

Add equation (16) to equation (11), and we can get the result:

$$c_i = (a_{i,0}c_0 + \cdots + a_{i,n-1}c_{n-1}) + (tv_0c_0 + \cdots + tv_{n-1}c_{n-1}). \quad (16)$$

Merge the similar items:

$$c_i = (a_{i,0} + tv_0)c_0 + \cdots + (a_{i,n-1} + tv_{n-1})c_{n-1}. \quad (17)$$

Obviously, the row vector $(a_{i,0} + tv_0, \dots, a_{i,n-1} + tv_{n-1})$ is still the linear representation of the i th symbol c_i . So, the renewed matrix after the row transformation is still a linear relation matrix. $\square$

3. The Proposed Decoding Method for the Erasure Codes

In this paper, we will propose a novel decoding method mainly for the erasure codes in the storage system, such as RS code and array codes. The proposed decoding method is based on the property: for any element $e_i \in GF(q)$, $e_i + e_i = 0$, where $q = 2^m$, m is a positive integer.

3.1. The Decoding Process. Suppose that the original information is divided into k symbols, and they are encoded into n symbols $\{c_0, c_1, \dots, c_{n-1}\}$ by the (n, k) erasure code. There are no more than $n - k$ failures, and the set of failure symbols is marked as F , and the decoding can be achieved by the following steps.

- (1) Generate $n \times n$ unit matrix $\mathbf{W}$ as the linear relation matrix:

$$\mathbf{W}_{n \times n} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}. \quad (18)$$

- (2) The $n \times n$ linear relation matrix $\mathbf{W}$ and the $(n - k) \times n$ check matrix $\mathbf{H}$ are combined into the $(2n - k) \times n$ decoding transformation matrix $\mathbf{S}$:

$$\mathbf{S}_{(2n-k) \times n} = \begin{pmatrix} \mathbf{W}_{n \times n} \\ \mathbf{H}_{(n-k) \times n} \end{pmatrix} = \begin{pmatrix} \mathbf{S}_{n \times n}^{\wedge} \\ \mathbf{S}_{(n-k) \times n}^{\vee} \end{pmatrix} \quad (19)$$

$$= \begin{pmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,n-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{2n-k-1,0} & a_{2n-k-1,1} & \dots & a_{2n-k-1,n-1} \end{pmatrix}.$$

Based on the linear relation of the matrix $\mathbf{H}$ and $\mathbf{W}$, we can know that the i th column of $\mathbf{H}$ and $\mathbf{W}$ both corresponds to c_i , so the i th column of $\mathbf{S}$ also corresponds to c_i .

- (3) Select the failed symbol c_j from F , j is an integer and $0 \leq j \leq n - 1$. Find the nonzero elements in the j th column in $\mathbf{S}_{(2n-k) \times n}$, and suppose that there are t nonzero elements, and the row indexes of the nonzero elements are denoted as the set NR^j , $NR^j = \{nr_0^j, nr_1^j, \dots, nr_{t-1}^j\}$.
- (4) If there exists the elements greater than or equal to n in the set NR^j , the first element is nr_a^j ,

$a \in \{0, 1, 2, \dots, t - 1\}$, and go to the Step 5; otherwise, it indicates that the symbol c_j cannot be recovered.

- (5) For each element nr_b^j in NR^j except nr_a^j , $b \neq a$, the nr_b^j row of $\mathbf{S}_{(2n-k) \times n}$ is renewed as following:

$$\mathbf{S}(nr_b^j, :) = \mathbf{S}(nr_a^j, :) \cdot \left(\frac{\mathbf{S}(nr_b^j, j)}{\mathbf{S}(nr_a^j, j)} \right). \quad (20)$$

The operation of " $A = B$ " is the shorthand expression of " $A = A + B$," which means the value of " A " is reassigned by the result of " $A + B$."

Then, delete nr_b^j from the set NR^j .

- (6) After all the nr_b^j rows are renewed, set all the elements in the nr_a^j th row of $\mathbf{S}$ to be zero. Then, delete c_j from the set F .
- (7) Now, check whether $\mathbf{S}_{(n-k) \times n}^{\vee} = 0$ or not; if yes, it means that the decoding process is terminated. If not, continue to execute the decoding operation.
- (8) Repeat Step 3 to Step 7 until $\mathbf{S}_{(n-k) \times n}^{\vee} = 0$; if all the elements in the set F are deleted, it means all the failure symbols can be recovered successfully. If there are still some elements in the set F , it means that these symbols cannot be recovered.
- (9) In the last step, the upper $n \times n$ submatrix of the renewed $\mathbf{S}_{(2n-k) \times n}$ is still a linear relation matrix, and it can be used as the decoding matrix $\mathbf{D}$. So, the k th row of the matrix $\mathbf{D}$ is the coefficients about the linear combination of the lost symbol c_k by the remaining symbols. Thus, c_k can be recovered:

$$c_k = \sum_{j=0}^n \mathbf{D}(k, j) \cdot c_j, \quad (21)$$

where $\mathbf{D}(i, j)$ is the element in the i th row and the j th column of matrix $\mathbf{D}$, i and j are integers and $0 \leq i$ and $j \leq n - 1$.

Based on the matrix transforming process about the decoding transforming matrix, the lost symbols deleted from the set F in (6) can be recovered over $GF(q)$ no matter they are the original data symbols or the check symbols. If all the symbols are deleted from F , all the lost symbols can be recovered; if not, we also can know which part of symbols can be recovered.

3.2. Example of the Decoding. A $(5, 3)$ erasure code over $GF(2^3)$ is constructed based on the Cauchy matrix, the original information $\{3, 4, 1\}$, and encoded into 5 symbols $\{3, 4, 1, 1, 3\}$, and the generator matrix $\mathbf{G}$ and the check matrix $\mathbf{H}$ are shown as follows:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 7 & 6 & 5 \\ 5 & 2 & 7 \end{pmatrix}, \quad (22)$$

$$\mathbf{H} = \begin{pmatrix} 7 & 6 & 5 & 1 & 0 \\ 5 & 2 & 7 & 0 & 1 \end{pmatrix}. \quad (23)$$

Suppose that c_0 and c_1 are lost, $F = \{c_0, c_1\}$, and the decoding process is as follows:

- (1) Generate the 5×5 unit matrix $\mathbf{I}$ as the initial linear relation matrix $\mathbf{W}$.
- (2) Combine the matrix $\mathbf{W}_{5 \times 5}$ and $\mathbf{H}_{2 \times 5}$ to get the initial decoding transformation matrix $\mathbf{S}$,

$$\mathbf{S}_{7 \times 5} = \begin{pmatrix} \mathbf{W}_{5 \times 5} \\ \mathbf{H}_{2 \times 5} \end{pmatrix},$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 7 & 6 & 5 & 1 & 0 \\ 5 & 2 & 7 & 0 & 1 \end{pmatrix}. \quad (24)$$

- (3) Select the first element c_0 in the set F . Find all the elements in the first column of $\mathbf{S}$, and the row indexes of the nonzero elements are $NR^0 = \{0, 5, 6\}$.
- (4) There are two elements $\{5, 6\}$ are greater than or equal to n , $n = 5$, and it satisfies the condition for the next decoding step.
- (5) Select the first element in $\{5, 6\}$, and $nr_a^j = 5$, so for other elements $\{0, 6\}$ except 5 in NR^0 , execute the operations on the 0th and 6th row of $\mathbf{S}$ as follows:

$$\mathbf{S}(0, :) + = \mathbf{S}(5, :) \cdot \left(\frac{\mathbf{S}(0, 0)}{\mathbf{S}(5, 0)} \right),$$

$$\mathbf{S}(6, :) + = \mathbf{S}(5, :) \cdot \left(\frac{\mathbf{S}(6, 0)}{\mathbf{S}(5, 0)} \right). \quad (25)$$

Then, the element 0 and 6 are deleted from $\{0, 6\}$.

- (6) After the two rows are renewed, set all elements in the 5th row to be zero in the matrix $\mathbf{S}$:

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_{5 \times 5}^{\wedge} \\ \mathbf{S}_{2 \times 5}^{\vee} \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 7 & 6 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 6 & 1 \end{pmatrix}. \quad (26)$$

The element c_0 can be deleted from $F = \{c_0, c_1\}$; now, $F = \{c_1\}$.

- (7) Notice that $\mathbf{S}_{2 \times 5}^{\vee} \neq 0$, and go to the next decoding step.
- (8) Select the next element c_1 from F , and find out the nonzero elements in 1st column, $NR^1 = \{0, 1, 6\}$.

There is one element 6 which is greater than n , $n = 5$, so for other elements $\{0, 1\}$, perform the operation on the 0th and 1st row of $\mathbf{S}$:

$$\mathbf{S}(0, :) + = \mathbf{S}(6, :) \cdot \left(\frac{\mathbf{S}(0, 1)}{\mathbf{S}(6, 1)} \right),$$

$$\mathbf{S}(1, :) + = \mathbf{S}(6, :) \cdot \left(\frac{\mathbf{S}(1, 1)}{\mathbf{S}(6, 1)} \right). \quad (27)$$

Then, the element 0 and 1 are deleted from $\{0, 1\}$. And, all the elements in the 6th row are set to be zero. The element c_1 can be deleted from $F = \{c_1\}$; now, $F = \emptyset$:

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_{5 \times 5}^{\wedge} \\ \mathbf{S}_{2 \times 5}^{\vee} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 2 & 3 & 7 \\ 0 & 0 & 1 & 6 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (28)$$

Notice that $\mathbf{S}_{2 \times 5}^{\vee} = 0$; it means that the maximum failure tolerance of the code has been reached. $F = \emptyset$ shows that all the failures $\{c_0, c_1\}$ can be recovered successfully.

- (9) Now the upper 5×5 submatrix of the renewed matrix $\mathbf{S}$ can be used as the decoding matrix $\mathbf{D}$:

$$\mathbf{D} = \begin{pmatrix} 0 & 0 & 2 & 3 & 7 \\ 0 & 0 & 1 & 6 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (29)$$

The two failure symbols $\{c_0, c_1\}$ can be easily decoded:

$$\begin{aligned} c_0 &= 2c_2 + 3c_3 + 7c_4 \\ &= 2 \cdot 1 + 3 \cdot 1 + 7 \cdot 3 \\ &= 3, \\ c_1 &= 1c_2 + 6c_3 + 1c_4 \\ &= 1 \cdot 1 + 6 \cdot 1 + 1 \cdot 3 \\ &= 4. \end{aligned} \quad (30)$$

The addition and the multiplication over $GF(2^3)$ are operated based on the polynomial operations. At last, all the failure symbols are recovered successfully.

3.3. Proof

3.3.1. Proof of the Method. In this section, we will give out two basic important theorems, and the correctness of the proposed method can be proved based on the theorems.

Theorem 1. Suppose that S is the decoding transformation matrix of the (n, k) linear block code over the finite field $GF(q)$; after multiplying the j th row of S by t , the renewed matrix S is still the decoding transformation matrix, and the upper submatrix $\mathbf{S}_{n \times n}^\wedge$ of the matrix S is still a linear relation matrix where $q = 2^m$, $t \neq 0$, $t \in GF(q)$, j is an integer, and $n \leq j \leq 2n - k - 1$.

Proof. Suppose that the j th row of S is $\{a_{j,0}, \dots, a_{j,n-1}\}$; after the transformation, the j th row is renewed as $\{ta_{j,0}, \dots, ta_{j,n-1}\}$.

From the construction of the decoding transformation matrix S , since the lower part $\mathbf{S}_{(n-k) \times n}^\vee$ in the beginning is the check matrix $\mathbf{H}_{(n-k) \times n}$, so we can get

$$a_{j,0}c_0 + a_{j,1}c_1 + \dots + a_{j,n-1}c_{n-1} = 0, \quad (31)$$

where j is an integer, $n \leq j \leq 2n - k - 1$.

Then, multiply the j th row of S by t :

$$\begin{aligned} t \cdot (a_{j,0}c_0 + a_{j,1}c_1 + \dots + a_{j,n-1}c_{n-1}) &= t \cdot a_{j,0} \cdot c_0 + t \cdot a_{j,1} \cdot c_1 + \dots + t \cdot a_{j,n-1} \cdot c_{n-1} \\ &= 0. \end{aligned} \quad (32)$$

where j is an integer, $n \leq j \leq 2n - k - 1$.

It is obvious that the renewed j th row $\{ta_{j,0}, \dots, ta_{j,n-1}\}$ is still can be used to check $\{c_0, c_1, \dots, c_{n-1}\}$. So, the renewed matrix S is still the decoding transformation matrix. And, the upper submatrix $\mathbf{S}_{n \times n}^\wedge$ of the matrix S is still a linear relation matrix. $\square$

Theorem 2. Suppose that S is the decoding transformation matrix of the (n, k) linear block code over the finite field $GF(q)$. After multiplying the j th row in the lower part by t and adding to the i th row, the upper submatrix $\mathbf{S}_{n \times n}^\wedge$ of the renewed S is still a linear relation matrix, where $q = 2^m$, $t \neq 0$, $t \in GF(q)$, i and j are integers, $n \leq j \leq 2n - k - 1$ and $0 \leq i \leq 2n - k - 1$.

Proof. Suppose that the i th row is $\{a_{i,0}, \dots, a_{i,n-1}\}$; the j th row is $\{a_{j,0}, \dots, a_{j,n-1}\}$ after the transformation the i th row is renewed as $\{ta_{j,0} + a_{i,0}, \dots, ta_{j,n-1} + a_{i,n-1}\}$. We can discuss it in two cases. $\square$

Case 4. If $n \leq i \leq 2n - k - 1$, the i th row is in the lower part of the matrix S , as the lower part $\mathbf{S}_{(n-k) \times n}^\vee$ in the beginning is the check matrix $\mathbf{H}_{(n-k) \times n}$:

$$\begin{aligned} a_{i,0}c_0 + a_{i,1}c_1 + \dots + a_{i,n-1}c_{n-1} &= 0, \\ a_{j,0}c_0 + a_{j,1}c_1 + \dots + a_{j,n-1}c_{n-1} &= 0. \end{aligned} \quad (33)$$

Multiply the j th row of the matrix S by t and add to the i th row; we can get

$$(ta_{j,0} + a_{i,0})c_0 + \dots + (ta_{j,n-1} + a_{i,n-1})c_{n-1} = 0. \quad (34)$$

The renewed i th row $\{ta_{j,0} + a_{i,0}, \dots, ta_{j,n-1} + a_{i,n-1}\}$ still can be used as the check relationship for $\{c_0, c_1, \dots, c_{n-1}\}$.

Case 5. If $0 \leq i \leq n - 1$, the i th row is in the upper part of the matrix S , as the upper part of S is the linear relation matrix, so

$$a_{i,0}c_0 + a_{i,1}c_1 + \dots + a_{i,n-1}c_{n-1} = c_i. \quad (35)$$

The j th row is in the lower part of the matrix S , as the lower part of the matrix S is the check matrix, so

$$a_{j,0}c_0 + a_{j,1}c_1 + \dots + a_{j,n-1}c_{n-1} = 0. \quad (36)$$

Multiply the j th row in the lower part by t and add to the i th row in the upper part:

$$(ta_{j,0} + a_{i,0})c_0 + \dots + (ta_{j,n-1} + a_{i,n-1})c_{n-1} = c_i. \quad (37)$$

Obviously, the renewed i th row $\{ta_{j,0} + a_{i,0}, \dots, ta_{j,n-1} + a_{i,n-1}\}$ is still the linear relationship of c_i . So, Theorem 2 is valid.

Based on the two basic theorems, the proof can be described as follows.

Proof. In the proposed decoding method, the initial decoding transformation matrix is generated by joining the linear relation matrix W with the check matrix H ; in Step 9, the upper part of S is the decoding matrix which is used to decode to recover the lost symbols.

So the key point to prove the correctness of the method is that, how to make sure the matrix transformation will not corrupt the property of the renewed upper matrix $S_{n \times n}^\wedge$; it is still the relation matrix and can be used to show the linear combination about the symbols.

The key transformation in the decoding process is Step 5. As in Step 4, the selected element $nr_a^j \geq n$; then, the nr_a^j th row of S is multiplied by $S_{n \times n}^\wedge$ and added to the nr_b^j th row ($b \neq a$).

Since $nr_a^j \geq n$, it satisfies $n \leq nr_a^j \leq 2n - k - 1$, so Step 5 is the case that the nr_a^j th row in the lower part of S is multiplied by t and added to the nr_b^j th row, $t = S_{n \times n}^\wedge$. From Theorem 2, the upper submatrix $S_{n \times n}^\wedge$ of the renewed S is still a linear relation matrix, and $S_{n \times n}^\wedge$ still can be used to show the linear relationship about the symbols, so in Step 9, we can decode to get the lost symbols by using the decoding matrix $D = S_{n \times n}^\wedge$. $\square$

3.3.2. Analysis of the Algorithm Complexity. The complexity of the proposed method can be derived from the analysis about the decoding process. There are addition and multiplication operations over the finite field in the decoding because the multiplication over the finite field is much more time-consuming than the addition operation, so the complexity is generally evaluated by the multiplication operations.

Most of the multiplication operations are executed in Step 5; the nr_b^j th row of S is renewed by multiplying the nr_a^j th rows by t and adding to the nr_b^j th row. It is necessary to perform at most n multiplication operations over the finite field; suppose that there are p symbols missing, so the total multiplication operations need to be performed at most $p \cdot n$ times in the decoding. So, the complexity of this method is about $O(p \cdot n)$.

The computational complexity of the equation solving method used in [6] is about $O(n^3)$, and the merge decoding method used in [7] performs decoding by the inversion of the check matrix, so the complexity is also about $O(n^3)$. The complex of the matrix method in [18] is $O(M^2)$, where M is the number of the ones in the row, $M < n$, so the complexity of the method is $O(n^2)$. The decoding method in EVENODD code [9], X code [10], and RDP [15] is the cyclic iteration method, and the decoding complexity is $O(n \log n)$, and the decoding method in [3] is also the cyclic iteration method, so the decoding complexity is $O(n \log n)$.

From the complexity analysis and the comparison between the different methods, we can see that the complexity of the proposed method is nearly linear, and it is lower than other methods.

Furthermore, there is another point in the proposed decoding method that is beneficial to improve the decoding efficiency and not as the decoding methods in [11, 15]; when the lost symbols contain both the original data symbols and the check symbols, it is necessary to recover the original

symbols first and then perform the multiply operation by using the check matrix to get the lost check symbols. In contrast, the proposed method can decode to get the original data symbols and the check symbols directly, and it is not necessary to recalculate to get the lost check symbols after the decoding process, so the decoding efficiency can be further improved.

4. Experiments and Analysis

In this section, some experiments about the evaluation of the decoding performance of different decoding methods are given out by simulation.

As we all know, there are two typical kinds of erasure codes, RS code, and array codes, so the proposed decoding method is compared with the decoding methods for different kinds of erasure codes.

For the RS code, the proposed decoding method is compared with the equation solving method in [6] and the merge decoding method in [7]. For the array codes, the proposed decoding method is compared with the loop iteration method in X code [10], EVENODD code [9], RDP code [15], and iteration decoding method in the slope code [3]. The decoding method in [18] is also compared with the proposed method by the typical array code.

The reconstruct bandwidth is another advantage of the method by tiny modification. To evaluate the reconstruction bandwidth, the comparison of the performance of different array codes is given out.

At last, the strategy of involved nodes in the decoding to reduce the burden is proposed.

4.1. Time Efficiency of Failure Recovery. For RS erasure codes, we will compare our method with the equation solving method in [6] and the merge decoding method in [7]. The Cauchy RS code is used as the failure tolerance method in the simulation.

Experiment 1. Suppose that the erasure code is CRS (4,8) erasure code over $GF(2^8)$; there are 8 independent storage nodes, including 4 original data nodes and 4 check nodes, and it is clear that it can tolerate at most 4 failures.

Suppose that the storage block size in each storage node is 14,376B. The network bandwidth is enough for the data transmission in the distributed storage system. The field $GF(2^8)$ is constructed by using the method in [20].

Assume that there are 1, 2, 3, and 4 nodes' failures in the storage system, respectively. We will decode by using the equation solving method in [6], the merge decoding method in [7], and the proposed decoding method. The comparison of time consumption is shown in Table 1.

From the table, we can see that the decoding performance of the proposed method is much better than both the merge decoding method and the equation solving method. And, the decoding time of the merge decoding method in [7] is lower than the equation solving method in [6] when there is one node failure, but the decoding time is higher in the case of multiple nodes' failure.

TABLE 1: The decoding time of different nodes' failure.

	Method in [6]	Method in [7]	The proposed method
1 node' failure	191.07	64.11 s	15.06 s
2 nodes' failure	856.83	1357.3 s	29.78 s
3 nodes' failure	1438.6	2619.89 s	47.11 s
4 nodes' failure	1499.13	3077.47 s	62.93 s

For the array codes, the most common decoding method is the cyclic iteration method, so the proposed method is compared with the cyclic iteration method in Experiments 2 and 3.

In Experiment 2, the proposed decoding method is compared with the iteration decoding method for EVEN-ODD code [9], X code [10], and RDP code [15].

Experiment 2. For EVENODD code [9], X code [10], and RDP code [15], use the proposed method and the cyclic iteration method as the decoding method, respectively. Assume that the file size is 1,228,800B and the maximum storage block size of each storage node is 10,240 bits. The decoding performance about the different methods is shown in Table 2.

From the table, we can see that the time efficiency in the decoding using our method is better than the method in [9, 10, 15].

Experiment 3. For the slope code [3], suppose that the strip size is 4 and the maximum failure tolerance is 4. From the construction of the slope code, there are 26 storage nodes including 13 data information nodes and 13 check information nodes. Suppose that the file size is 1,261,568B and the block in each storage node is 10,240B. Assume that the failure is from 1 to 4; use the cyclic iteration method and the proposed method as the decoding method for the different failure cases; the comparison of decoding efficiency is shown in Figure 1.

From this figure, we can know that the decoding efficiency of the proposed method in this paper is better than the traditional cyclic iteration method for the slope code.

Experiment 4. In this experiment, the decoding efficiency performance between the method in [18] and the proposed method is compared.

Take EVENODD code as the failure tolerance method. Suppose that the data size of each storage node is 10800B, and the decoding efficiency about one node failed is shown as Table 3.

The decoding efficiency when there are two nodes failed is shown in Table 4.

From the tables, we can see that the decoding efficiency of the proposed method is little better than the method in [18] when there is one node failed, but the decoding efficiency is much better when there are two nodes failed.

From the comparisons among the different decoding methods for RS codes and the array codes, we can draw the

conclusion that the decoding efficiency of the proposed method is better than the other methods.

4.2. Reconstruction Bandwidth. In the distributed storage system based on the (n, k) erasure code, there will be a lot of data need to be transmitted even only one node is failed, which is called as the reconstruction bandwidth problem. So how to effectively reduce the amount of data transmitted is a hot topic in the distributed storage. Based on the basic idea of the regenerating code [22], our method can reduce the reconstruction bandwidth only by tiny modification.

It only needs to add a judgment condition in Step 4 of the proposed method. When we select the nr_a^j in Step 4, the element with the highest repetition rate of data blocks involved in the decoding will be selected. In other words, for every selection of nr_a^j in the decoding, the row with the lowest hamming weight is selected.

Experiment 5. For the array codes including the EVENODD code [9], the X code [10], the RDP code [15], and the slope code [3], use the proposed method and the iteration method in [3, 9, 10, 15] as the decoding method. Assume that the file size is 1,261,568B, and the block size of each storage node is 10,240 bits. The comparison of the amount of data transmitted is shown in Figure 2, where the EVENODD code, the X code, and the RDP code are constructed based on the prime number 7.

From the figure, we can see that the number of the data blocks need to transmit in the proposed method for different erasure codes is lower than the iteration method in [3, 9, 10, 15].

The comparison of the decoding time is shown as Figure 3, as the decoding complexity of the proposed method is lower than the cyclic method, and the amount of data transmitted in the decoding is less than the cyclic method, so from the figure we can see that the decoding efficiency of the proposed method is also better than the other methods.

Experiment 6. In the experiment, we will evaluate the reconstruction bandwidth and the recovery performance while the failed nodes are increased.

Suppose the distributed storage system is based on the slope code, the strip size is 5, and the maximum failure tolerance is 5. Assume the file size is 1,261,568B and the block size in each storage node is 10,240 bits; use the cyclic iteration method in [3] and the proposed method as the decoding method, and the transmitted data blocks required for the data reconstruction from 1 to 5 failures are shown in Table 5.

Obviously, the proposed method can reduce the amount of transmitting data in the decoding, which can also improve the decoding efficiency. The comparison of the decoding time from 1 to 5 failure nodes is shown in Table 6.

4.3. Design of Involved Nodes for Data Recovery. In the distributed storage system, it is possible that some nodes are

TABLE 2: The decoding time of the different methods.

	EVENODD decode [9]	X decode [10]	RDP decode [15]
Original decode method	1.49	1.51 s	1.62 s
The proposed decode method	1.38	1.42 s	1.48 s

FIGURE 1: Time comparison of data recovery from multiple failure nodes.

TABLE 3: The decoding time of different file sizes when one node failed.

(KB)	Method in [18]	The proposed method
200	0.51 s	0.58 s
400	1.06 s	0.97 s
600	1.55 s	1.50 s
800	2.11 s	1.98 s
1000	2.53 s	2.46 s

TABLE 4: The decoding time of different file sizes when two nodes failed.

(KB)	Method in [18]	The proposed method
200	2.65 s	1.98 s
400	4.31 s	3.71 s
600	6.75 s	5.03 s
800	8.17 s	5.21 s
1000	10.25 s	8.94 s

FIGURE 2: Network transmission blocks required for a single failure node reconstruction.

FIGURE 3: Time comparison for a single failure node reconstruction.

TABLE 5: The data blocks required for multiple failure nodes reconstruction.

	Cyclic iteration method	The proposed method
1 node failed	25	20
2 node failed	45	36
3 node failed	65	50
4 node failed	85	67
5 node failed	105	74

visited frequently in a period of time, and it will undoubtedly increase the access burden of these overheated nodes and lead to the low efficiency. In the proposed method, it can dynamically select the involved nodes in the decoding to avoid the excessively visiting overheated nodes.

Take the slope code as the example, assume that the strip size is 2 and the maximum failure tolerance is 2. The stripe of the slope code is shown in Figure 4; in each stripe, the first 3 blocks are the data blocks and the last 3 blocks are the parity blocks, and the layout of all the blocks is shown in Figure 4.

TABLE 6: The reconstruction time of multiple nodes failed.

	Cyclic iteration method (s)	The proposed method (s)
1 node failed	1.18	1.13
2 node failed	2.07	1.95
3 node failed	2.79	2.29
4 node failed	3.88	2.94
5 node failed	4.62	3.61

FIGURE 4: A stripe based on the slope code.

According to the definition of the slope code, the check matrix H of the slope code is shown as (38). Assume the node 1 is failed, so the blocks $\{0, 3\}$ in node 1 are invalid, and the decoding matrix D based on the decoding method in this paper is shown as (39):

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad (38)$$

$$D = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (39)$$

From the decoding method in this paper, the failure blocks can be recovered by using the decoding matrix D , $E_0 = E_4 + E_6$. It shows that the nodes $\{1, 2, 3, 4\}$ will participate in the decoding of the failure blocks $\{0, 3\}$ based on the blocks' layout, as shown in Figure 4.

If the node 1 is visited frequently, to avoid visiting the node 1, we can assume that the node 1 is also failed, so the blocks $\{1, 4\}$ in the node 1 are failed. Now, there are four blocks $\{0, 3, 1, 4\}$ that failed, so the decoding matrix D is changed into the following equation:

$$D = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (40)$$

So, the failure blocks can be decoded based on the decoding matrix, and $E_0 = E_2 + E_6 + E_{11}$, $E_3 = E_2 + E_8$. We can find that the nodes $\{2, 3, 4, 5\}$ will participate in the decoding of the blocks $\{0, 3\}$, and the node 1 will not be involved in the decoding.

Furthermore, if the node 1 and 2 are overheated, we continue to assume that the node 1 and 2 are failed, so the six blocks $\{0, 1, 2, 3, 4, 5\}$ are failed. In this situation, there is no valid decoding matrix that can be found. Therefore, we can know that the node 2 is the essential node for the decoding, and the decoding cannot be realized without the node 2.

5. Conclusion

In this paper, the novel decoding method for the erasure codes is proposed, and it is a general decoding method to improve the decoding efficiency for different kinds of erasure codes over different finite fields. It is easy to get the linear combination of the failure symbols by other valid symbols by the simple matrix transformation of the decoding transformation matrix. For the RS codes over the binary extension field, the proposed method can eliminate the matrix inversion in the decoding, so the decoding efficiency is greatly improved. For the array codes, the proposed method is still useful for the XOR-based array codes over $GF(2)$ to improve the decoding efficiency. Furthermore, the proposed method can effectively reduce the repair bandwidth for the array codes in the data reconstruction compared with the classical iteration method by tiny modification. Another improved strategy can be realized by dynamically planning the valid nodes that are involved in the decoding to lighten the overburdened storage nodes.

Data Availability

No data were used to support the findings of the study.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by Sichuan Province Science and Technology Support Program (China), nos. 2020YF0230, 2020YFG0150, and 2020YFG0294.

References

- [1] J. S. Plank, "Erasure codes for storage systems: a brief primer," *Log in: the magazine of USENIX & SAGE*, vol. 38, no. 6, pp. 44–50, 2013.
- [2] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos et al., "XORing elephants: novel erasure codes for big data," *Very Large Data Bases*, vol. 6, no. 5, pp. 325–336, 2013.
- [3] D. Tang and H. P. Shu, "A class of array erasure codes with high fault-tolerance," *Science China Information Sciences*, vol. 46, no. 4, pp. 523–538, 2016.
- [4] J. S. Plank and Y. Ding, "Note: Correction to the 1997 tutorial on reed-solomon coding," *Software: Practice and Experience*, vol. 35, no. 2, pp. 189–194, 2005.
- [5] J. S. Plank and L. Xu, "Optimizing cauchy reed-solomon codes for fault-tolerant network storage applications," in *Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications*, pp. 173–180, Cambridge, MA, USA, July 2006.
- [6] J. S. Plank, "A tutorial on Reed-Solomon coding for fault-tolerance in RAID-like systems," *Software: Practice and Experience*, vol. 27, no. 9, pp. 995–1012, 1997.
- [7] D. Tang, "Research of methods for lost data reconstruction in erasure codes over binary fields," *Journal of electronic science and technology*, vol. 14, no. 1, pp. 43–48, 2016.
- [8] J. S. Plank, K. M. Greenan, and E. L. Miller, "Screaming fast galois field arithmetic using intel SIMD instructions," in *Proceedings of the FAST: USENIX Conference on File and Storage Technologies*, pp. 299–306, Allerton, IL, USA, October 2013.
- [9] M. Blaum, J. Brady, J. Bruck et al., "EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Transactions on Computers*, vol. 45, no. 2, pp. 192–202, 1995.
- [10] L. Xu and J. Bruck, "X-code: MDS array codes with optimal encoding," *IEEE Transactions on Information Theory*, vol. 45, no. 1, pp. 272–276, 1999.
- [11] J. L. Hafner, "WEAVER codes: highly fault tolerant erasure codes for storage systems," in *Proceedings of the FAST '05 Conference On File And Storage Technologies*, San Francisco, California, USA, December 2005.
- [12] J. L. Hafner, "HoVer erasure codes for disk arrays," in *Proceedings of the International Conference on Dependable Systems & Networks*, pp. 217–226, Philadelphia, PA, USA, June 2006.
- [13] J. S. Plank and M. Blaum, "Sector-Disk (S. D.) Erasure Codes for Mixed Failure Modes in RAID Systems," *ACM Transactions on Storage*, vol. 10, no. 1, pp. 1–18, 2014.
- [14] L. Chen, J. Z. Zhang, P. G. Teng, and X. J. Wang, "Random binary extensive code (RBEC): an efficient code for distributed storage system," *Chinese Journal of Computers*, vol. 40, no. 9, pp. 1–17, 2016.
- [15] P. Corbett, B. English, A. Goel et al., "Row-diagonal parity for double disk failure correction," in *Proceedings of the Usenix Conference on File and Storage Technologies*, San Francisco, CA, USA, March 2004.
- [16] L. Xu and J. Bruck, "Highly available distributed storage systems," in *Proceedings of the international Conference on High Performance Computing Data and Analytics*, pp. 307–330, Calcutta, India, December 1999.
- [17] J. Blomer, M. Kalfane, R. Karp et al., *An XOR-Based Erasure-Resilient Coding Scheme*, International Computer Science Institute, CA, USA, 1995.
- [18] J. L. Hafner, V. Deenadhayalan, K. K. Rao et al., "Matrix methods for lost data reconstruction in erasure codes," in *Proceedings of the Conference On Usenix Conference On File & Storage Technologies*, San Francisco, CA, USA, December 2005.
- [19] C. C. Pinter, *A Book of Abstract Algebra*, McGraw-Hill, NY, USA, 1982.
- [20] W. P. Wardlaw, "Matrix representation of finite fields," *Mathematics Magazine*, vol. 67, no. 4, pp. 289–293, 1994.
- [21] P. Vera, R. A. Brualdi, and W. C. Huffman, *Handbook of Coding Theory*, Elsevier, Amsterdam, Netherlands, 1998.
- [22] D. S. Papailiopoulos, J. Luo, A. G. Dimakis et al., "Simple regenerating codes: network coding for cloud storage," in *Proceedings of the International Conference on Computer Communications*, pp. 2801–2805, Ottawa, Canada, September 2012.