Research of Methods for Lost Data Reconstruction in Erasure Codes over Binary Fields

Dan Tang

Abstract—In the process of encoding and decoding, erasure codes over binary fields, which just need AND operations and XOR operations and therefore have a high computational efficiency, are widely used in various fields of information technology. A matrix decoding method is proposed in this paper. The method is a universal data reconstruction scheme for erasure codes over binary fields. Besides a pre-judgment that whether errors can be recovered, the method can rebuild sectors of loss data on a fault-tolerant storage system constructed by erasure codes for disk errors. Data reconstruction process of the new method has simple and clear steps, so it is beneficial for implementation of computer codes. And more, it can be applied to other non-binary fields easily, so it is expected that the method has an extensive application in the future.

Index Terms—Binary fields, data reconstruction, decoding, erasure codes.

1. Introduction

From the perspective of information theory, an erasure code is a forward error correction (FEC) code for the erasure channel, which transforms a message of k symbols into a longer message (codeword) with n symbols such that the original message can be recovered from a subset of the n symbols, where k and n are positive integers, and \( k \leq n \). The fraction \( r = k/n \) is called the code rate, the fraction \( k'/k \), where \( k' \) denotes the number of symbols required for recovery, and \( k' \geq k \) is called reception efficiency. Distinct from error correcting codes, erasure codes are designed to recover erased bits in a message word. The positions of all erased bits are known. In contrast, error correcting codes are intended to correct messages in which some of the bits may have been flipped, but the positions of those bits are unknown. As a simplest example, the parity check code is a typical single-erasure-correcting code, but it is not a single-error-correcting code. The basic principle of encoding and decoding on erasure codes is given in Fig. 1, some detailed knowledge can be referred to [1], and this article will no longer give more descriptions.

![Fig. 1. Encoding and decoding process on erasure codes.](image)

With the development of information technology, erasure codes are under a wide range of applications such as multichannel transmission, fault tolerance on storage systems, secret sharing schemes, information dispersal systems, and so on. Lots of codes can be seen as a kind of erasure codes, while the Reed-Solomon (RS) erasure code is the one which is most commonly used in various areas because of its powerful error correcting capability, sophisticated mathematical theories, and beautiful properties. According to information theory, the RS code meets the Shannon theorem with maximum distance separable (MDS) property, and it is perfect in code rate and fault tolerance ability. However, for RS codes, the encoding and decoding procedures are performed as operations over a finite field, so that systems based on RS codes will become very slow (even unacceptable) with the increase of operational data. Under this background, erasure codes over binary fields (just need AND operations and XOR operations) get more and more attention. So many erasure codes over binary fields were proposed in recent years, but most researchers have focused more on encoding processes, while the decoding methods get less attention. Almost all erasure codes over binary fields use recursive methods, Tanner graph methods, and geometrical methods (mainly in array codes) to complete decoding procedures. Reference [10] is one of few papers specializing in decoding method of erasure codes over binary fields, where a decoding method based on a generator matrix and pseudo-inverses was introduced. The method presented in [10] can be applied to almost all erasure codes over binary fields.

Nowadays, so many practical systems use an erasure code over binary fields as the core idea, and the fault tolerance scheme of storage systems (disk array) is one of...
the most typical representatives. While in the field of fault tolerance on storage systems, many kinds of erasure codes over binary fields have been proposed, such as EVENODD code\[^6\], X code\[^7\], STAR code\[^8\], Waver code\[^9\], and so on. So this article focuses on lost data reconstruction for storage systems (especially disk array). This strategy does not affect the universal property of the new method in other application areas certainly. Erasure codes for disk arrays model lose data most coarsely as loss of entire disks, but more precisely as loss of sectors (some terminologies like strip, stripe, and element in storage systems can be referred to [3]) of the code at present. For example, the X code can recover from two lost disks\[^7\] and the STAR code can recover from three lost disks\[^8\]. However, the most common types of failures are latent sector failures for disks, which only affect individual disk sectors. Then erasure errors can be seen as sector failures, and all sectors in a stripe are lost when the entire disk fails\[^3\]. Currently, increasing disk capacity together with a fairly stable bit-error rate implies that there is a significant probability of multiple uncorrelated or scattered sector errors within a given stripe, particularly in conjunction with one or more disk failures. Actually, two disk losses plus one sector loss or some sectors (in separate disks) loss may occur more often than two disks failures simultaneously. If all correlated or uncorrelated erasures occur within at most \( t \) disks where \( t \) is the (disk) fault tolerance of the code, then one method is to simulate the loss of all affected disks and rebuild according to the code-specific reconstruction algorithm\[^10\]. However, this scheme does not solve the more general problem when more than \( t \) disks have been affected with sector losses. In such a case, it is entirely possible that all of the lost data can be reconstructed, but we must to admit all data loss according to the preceding approach.

Example 1. There is a 5×5 disk array which uses the X code to tolerant faults. Its data layout is shown in Fig. 2. According to the fault-tolerant ability of the X code, the storage system can tolerate any 2 disks faults.

![Disk array based on X code](image)

As shown in the disk array above, Disk 1 fails (loss of all data on Disk 1) and all other disks have one sector of data loss. The loss of all data on the disk array depends upon its own data recovery mechanism. Unfortunately, to disks which have data loss in sectors, all loss data cannot be recovered totally by the decoding method of X code even if we did not label them as loss of entire disks. At this point, the storage system is breakdown completely, thus all information which is stored on this storage system is lost. However, all lost data under the above conditions can be reconstructed if there is a better decoding method actually, like the method presented in this paper.

In this paper, a new decoding method will be proposed, a matrix method for lost data reconstruction in erasure codes over binary fields. The new method can be applied to any kind of erasure codes, and it has simple steps although a relatively heavy computation and is easy for implementation of computer codes.

The rest of this paper is organized as follows. The next section contains a few remarks on elementary theories of our work. Section 3 describes the new decoding method for lost data reconstruction in erasure codes over binary fields amply. Section 4 proves correctness of the new method and gives an analysis of performance and complexity briefly. Section 5 is the summary of the paper.

2. Basic Theory

In order to describe the new method better, some basic theories are introduced in this section\[^10\].

First, we review some important basic notions of linear algebra over a binary field, and recall all operations used in this paper over the binary field unless a specified explanation.

In binary fields, addition is equal to the XOR operation and multiplication is equal to the AND operation.

Definition 1. Linearly independent over binary fields: A set of binary vectors is linearly independent if no subset sums to the zero vector.

Suppose that, vector \( \mathbf{v}_1=(0\ 1\ 1\ 0) \), vector \( \mathbf{v}_2=(0\ 1\ 0\ 1) \), vector \( \mathbf{v}_3=(0\ 0\ 1\ 1) \), and vector \( \mathbf{v}_4=(1\ 0\ 0\ 0) \). By Definition 1, a conclusion can be drawn easily: The set of binary vectors \( \{\mathbf{v}_1,\ \mathbf{v}_2,\ \mathbf{v}_3\} \) is linearly dependent, because of \( \mathbf{v}_1+\mathbf{v}_2+\mathbf{v}_3=\mathbf{0} \). Similarly, the set of binary vectors \( \{\mathbf{v}_1,\ \mathbf{v}_2,\ \mathbf{v}_4\} \) is linearly independent, because the following inequalities hold:\( \mathbf{v}_1+\mathbf{v}_2\neq\mathbf{0},\ \mathbf{v}_1+\mathbf{v}_3\neq\mathbf{0},\ \mathbf{v}_2+\mathbf{v}_4\neq\mathbf{0}, \) and \( \mathbf{v}_1+\mathbf{v}_2+\mathbf{v}_3\neq\mathbf{0} \).

Definition 2. Row rank: Assume that \( \mathbf{M} \) is a rectangular matrix of size \( m\times n \) with \( m\leq n \). The row rank of \( \mathbf{M} \) is the maximum number of linearly independent row vectors.

For example, two matrices \( \mathbf{M}_1 \) and \( \mathbf{M}_2 \) of size 4×5 are defined as follows:

\[
\mathbf{M}_1 = \begin{bmatrix}
1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1
\end{bmatrix}, \quad \mathbf{M}_2 = \begin{bmatrix}
1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1
\end{bmatrix}.
\]

The row rank of \( \mathbf{M}_1 \) is 4, because no subset sums of \( \mathbf{M}_1 \) is the zero vector, while the row rank of \( \mathbf{M}_2 \) is 3, because the sum of the first 3 row vectors of \( \mathbf{M}_2 \) is a zero vector.

Definition 3. Full row rank: The matrix \( \mathbf{M} \) has a full row rank if its row rank equals the number of rows.

Obviously, the matrix \( \mathbf{M}_1 \) has a full row rank.

Definition 4. Null space: A null space for a matrix \( \mathbf{M} \) is
the set of all vectors that are orthogonal with every row vector of \( M \).

This is a vector space closed under vector addition.

Definition 5. Null space basis: A null space basis is a maximal set of linearly independent vectors from the null space.

For column vectors, there are the same definitions and concepts above, not repeated in this section.

And next, we recall the erasure code notions of the generator matrix (usually use the label \( G \) to denote) and parity check matrix (usually use the label \( H \) to denote) in this section. These are used frequently in our methodology.

In simple terms, the generator matrix \( G \) of an erasure code converts the word (original data or incoming data) into a codeword (data and parity). While the parity check matrix verifies that the codeword contains consistent data and parity.

If there are \( k \) data elements input into the code and \( r \) parity elements computed by the code, then the generator matrix has dimensions \((k + r) \times k\). In coding theory, the generator matrix is a basis for a linear code, generating all possible codewords. If the linear code is marked as \( C \), then an equation \( c = Gd \) holds, where \( c \) is a codeword of the linear code \( C \), \( d \) is a column vector, and a bijection exists between \( c \) and \( d \). A generator matrix can be used to construct the parity check matrix for the same code, and vice versa.

The parity check matrix \( H \) corresponding to the generator matrix \( G \) above has dimensions \( r \times (k + r) \), and every element in a codeword can establish a bijection relationship to a column vector of matrix \( H \). Communication channels use the parity check matrix to detect errors. Each row corresponds to a parity equation. After the data and parity are read off the channel, the parity is performed the XOR operation with the data as indicated by its corresponding row to produce a syndrome. If a syndrome is not zero, an error has occurred. For erasure codes, this is a parity consistency check. With \( c = Gd \) as above, the test \( Hc = 0 \) is a parity consistency check, in other words, the equation \( HGd = 0 \) holds when the codeword \( c \) is correct. From description above, the equation \( HG = 0 \) holds, so that each vector in \( H \) is in the null space of \( G \). A simple dimensionality argument shows that in fact \( H \) is a basis of the null space of \( G \).

3. Decoding Method for Erasure Codes over Binary Fields

For erasure codes over binary fields, various decoding methods are almost for different application scenarios. For example, the recursive decoding method is mainly for codes constructed by the algebraic way; the geometrical decoding method is mainly for array codes constructed by the geometrical way. Those methods do not have a universal property. And more, those methods lack a judgment whether errors can be recovered. This section will describe a matrix method to reconstruct lost data. The new method can prejudge whether lost data can be recovered or not, and also be applicable to any kind of erasure codes.

As noted earlier, for the code \( C \), let the matrix \( H \) be a parity check matrix, and the matrix \( G \) a generator matrix, so each element in any codeword generated by the matrix \( G \) has a bijection relationship to one column vector of the matrix \( H \). And more, let the codeword which has some erasure errors be \( c = (c_1, c_2, \ldots, c_r)^T \). Now we can describe the new method, steps are described as follows:

Step 1. According to the hypothesis above, the codeword \( c \) has some erasure errors, then all elements in \( c \) is rearranged so that all erasure errors are on the right and the correct elements are on the left, thus \( c \) can be marked as \( c = (d|x)^T \), where \( d \) is a vector containing all errorless elements and \( x \) is a vector containing all erased elements.

Step 2. Rearrange column vectors of the matrix \( H \) according to the new positions of elements in \( c \), then \( H \) can be labeled as \( H = (A|B) \), and we assume that the submatrix \( A \) has dimensions \( m \times k \), and the submatrix \( B \) has dimensions \( m \times r \).

Step 3. If \( m > r \), all lost data cannot be rebuilded, the reconstruction process terminates. Otherwise, turn to Step 4.

Step 4. Search for \( r \)-row vectors in the matrix \( B \), and these \( r \)-row vectors must be linearly independent over binary fields. If there was no such row vectors then lost data cannot be recovered, the reconstruction process terminates. And if \( r \)-row vectors with linear independence exist, these vectors can form a matrix \( B \) of size \( r \times r \), then turn to Step 5.

Step 5. Take the \( r \) rows corresponding to \( B \) from the matrix \( A \), and form a new matrix \( A' \).

Step 6. From the description in Step 4, the matrix \( B \) is invertible, and we label its inverse element as \( B^{-1} \).

Step 7. Calculate \( B^{-1}A \), the result is the vector \( x \), all lost data are reconstructed completely.

In order to understand all above steps better, we will recover all lost data of Example 1 step by step. About the detailed description of the problem, please refer to Example 1, and this section does not repeat it.

The parity check matrix \( H \) is a key, and \( H \) can be gotten easily through the encoding process of \( X \) code as follows and its data layout (shown in Fig. 2).

\[
H = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 
\end{bmatrix}
\]
Rearrange column vectors of the matrix $H$ according to Step 2 in Section 3, and then $H$ is divided to two sub matrices $A$ and $B$.

\[
A = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\]

\[
B = \begin{pmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

Then the lost data can be calculated as follows: $\overline{B}^{-1}\overline{A}d$, where $\overline{B}^{-1}$, $\overline{A}$, and $d$ are known, all lost data can be recovered completely.

Besides some applications of fault tolerance for the disk array, the new method is also applied to analyze and correct erasure errors for all coding systems over binary fields.

Example 2. The Hamming code is a kind of classical coding system which reaches the Hamming bound. The code distance of a $(7, 4)$ Hamming code is 3, so it can correct one random error or three erasure errors at most by information theory. There are 7 bits for each codeword in $(7, 4)$ Hamming codes over a binary field, and we label them as $c_1, c_2, \ldots, c_7$.

Now assuming that the matrix $H$ is a parity check matrix of $(7, 4)$ Hamming codes.

\[
H = \begin{pmatrix}
1 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 1
\end{pmatrix}
\]

Every column in the matrix $H$ is corresponding to a bit in a codeword. If the original data is $(0 1 0 1 1 0 0)$, then the codeword is $(0 1 0 1 1 0 1)^T$ according the parity check matrix $H$ above, namely $c_1 = 0$, $c_2 = 1$, $c_3 = 0$, $c_4 = 1$, $c_5 = 1$, $c_6 = 0$, and $c_7 = 0$.

Supposing three bits $c_1$, $c_2$, and $c_3$ were lost during the transmission process, then we can recover them from steps as follows:

Rearrange bits in the codeword: $(1 1 0 1 1 0 0)$, and $d = (1 1 0 1 1 0 1)^T$.

Rearrange column vectors of the matrix $H$ according to Step 2 in Section 3, and then $H$ is divided to two sub matrices $A$ and $B$.

\[
A = \begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0
\end{pmatrix}
\]

\[
B = \begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & 0 \\
1 & 0 & 1
\end{pmatrix}
\]

The rank of matrix $B$ is 3 over a binary field, so the matrix $B$ is invertible. Thus $\overline{B} = B$, and then $\overline{A} = A$.

Because of matrix $\overline{B}$ is invertible, $\overline{B}^{-1}$ is got as

\[
\overline{B}^{-1} = \begin{pmatrix}
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]
Inversing the matrix $\mathbf{B}$:

$$\mathbf{B}^{-1} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. $$

Then all lost bits can be calculated as the following expression: $\mathbf{B}^{-1}\mathbf{A}\mathbf{d}$, where $\mathbf{B}^{-1}$, $\mathbf{A}$, and $\mathbf{d}$ are known.

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}. $$

4. Analysis and Discussions

The correctness of the new method should be proved firstly.

According to the hypothesis above, the codeword $\mathbf{c}$ is generated by a generator matrix $\mathbf{G}$ of the code $\mathbf{C}$, so we have

$$\mathbf{H}\mathbf{c} = \mathbf{0}. \quad (1)$$

After rearranging all column vectors of the matrix $\mathbf{H}$ and all elements of the vector $\mathbf{c}$, $\mathbf{H}$ is marked as $\mathbf{H} = (\mathbf{A}\mathbf{B})$ and $\mathbf{c}$ is marked as $\mathbf{c} = (\mathbf{d}\mathbf{x})^T$, equation (1) can be marked as the following form:

$$(\mathbf{A}\mathbf{B})(\mathbf{d}\mathbf{x})^T = \mathbf{0}. \quad (2)$$

From (2) and the calculation rules of partitioned matrices, the following equation holds:

$$\mathbf{A}\mathbf{d} + \mathbf{B}\mathbf{x} = \mathbf{0}. \quad (3)$$

As mentioned previously, all calculations are carried out over the binary field, and all add operations are equal to XOR operations. And two basic rules for XOR operations over a binary field would be described as follows: $n_1 + n_2 = 1$ when $n_1 \neq n_2$, and $n_1 + n_2 = 0$ when $n_1 = n_2$, where $n_1, n_2 \in \{0,1\}$ and the plus sign ‘+’ represents XOR operation. From the two rules of XOR operations, another deduction can be obtained: if $n_1 + n_2 + \cdots + n_{j-1} + n_m = a$ exists, $n_1 + n_2 + \cdots a + \cdots n_{m-1} + n_m = n_i$ holds, where $n_j, a \in \{0,1\}$, $1 \leq i, j \leq m$. The proof of the deduction is simple: the equation $n_1 + n_2 + \cdots + n_{j-1} + n_{m} = a$ is the given information, then add $n_j + a$ to both sides of the equation above, namely $n_1 + n_2 + \cdots + n_{j-1} + n_{m} + n_j + a = a + n_j + a$, and then equation $n_1 + n_2 + \cdots + n_{j-1} + n_{m} = a$ holds from basic rules of XOR operations.

Equation (4) holds according to (3) with the deduction above.

$$\mathbf{A}\mathbf{d} = \mathbf{B}\mathbf{x}. \quad (4)$$

$\mathbf{A}$ and $\mathbf{B}$ are sub-matrices of $\mathbf{A}$ and $\mathbf{B}$, respectively, and all row vectors of $\mathbf{A}$ and $\mathbf{B}$ have the same corresponding relationship to $\mathbf{A}$ and $\mathbf{B}$, then we have

$$\mathbf{A}\mathbf{d} = \mathbf{B}\mathbf{x}. \quad (5)$$

Because matrix $\mathbf{B}$ is invertible, erased data can be recovered from (5). End of the proof.

The computational complexity is a point we need to focus on, and in the new method, the number of erasure errors is the biggest factor affecting the amount of calculations. Fig. 3 shows this relationship between them. From this figure, reconstructing 500 erasure data, about $2.5 \times 10^8$ XOR operations are needed, namely, all core operations can be completed within 2 seconds by an ordinary personal computer nowadays.

However, another thing we have to mention is that, compared with traditional decoding methods, the new method requires a greater amount of computation though it has more advantages, such as a pre-judgment whether errors can be recovered, suitable for all erasure codes, applicable to other finite fields with few changes, and so on. A computational complexity comparison of the new method with some classical methods is shown in Fig. 4.

5. Summary

A new decoding method for erasure codes has been proposed in this paper, which can be applied to all kinds of erasure codes over the binary field. Moreover, it is not
difficult to extend the method to any non-binary fields, because the method is always tenable as long as any additional inverse element is itself over a field, so this is a universal decoding scheme of erasure codes. Besides a pre-judgment whether errors can be recovered, the method can rebuild sectors of loss data on a fault-tolerant storage system constructed by erasure codes without any changes of encoding algorithms. Great arithmetic demand is a principal disadvantage of the method, so improving the computational efficiency is our next step of research.

Acknowledgment

The author would like to take his deepest gratitude to Prof. Xiao-Jing Wang for his constant encouragement and guidance in preparing this paper. Without Prof. Xiao-Jing Wang’s illuminating instruction, this paper could not reach its present form.

References


Dan Tang received his Ph.D. degree from Graduate University of Chinese Academy of Sciences (CAS), Beijing, China in 2010. He is currently an associate professor with Chengdu University of Information Technology, Chengdu, China. His research interests include coding theory and secret sharing scheme.