New construction of ternary locally repairable codes that are also LCD codes

Ruipan Yang (yangruipan@aliyun.com)
Air Force Engineering University

Ruihu Li (liruihu@aliyun.com)
Air Force Engineering University

Qiang Fu (fuqiangkgd@163.com)
Air Force Engineering University

Research Article

Keywords: distributed storage system, locally repairable code, LCD code, CM bound

DOI: https://doi.org/10.21203/rs.3.rs-2415017/v2

License: ☕️ This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.
New construction of ternary locally repairable codes that are also LCD codes

Ruipan Yang1*, Ruihu Li1 and Qiang Fu1

1*Fundamentals Department, Air Force Engineering University, 1st Changledong Street, Xi’an, 710000, Shaanxi, China.

*Corresponding author(s). E-mail(s): yangruipan@aliyun.com; Contributing authors: liruihu@aliyun.com; fuqiangkgd@163.com;

Abstract

The concept of locally repairable code (LRC) is presented to improve the reliability and efficiency of distributed storage systems. Linear complementary dual (LCD) code is proved useful for preventing malicious attacks, and helps to secure the system. In this paper, we put LRC and LCD code together for consideration, to enhance repair efficiency and security. Firstly, a new method of constructing self-orthogonal code from arbitrary ternary linear code is proved. Then a new construction of ternary LRC that is also LCD code with small locality is proposed. As a result, plenty of ternary LCD-LRCs attaining the CM bound are obtained, which shows the effectiveness of our method.

Keywords: distributed storage system, locally repairable code, LCD code, CM bound

1 Introduction

Large data centers and peer-to-peer storage systems such as Ocean Store from Berkeley and BigTable from Google are famous examples of distributed storage systems[1]. While node failure of distributed storage systems seems inevitable due to hardware defects or communication errors. Thus, locally repairable codes (LRCs) are introduced, which can repair any lost node by accessing a small number of other available nodes[2]. Nowadays, LRCs are utilized in
some famous distributed storage systems, for example the Windows Azure and Facebook HDFS RAID[3].

In some respects, LRC is essentially a type of linear code with an additional parameter referred to as locality[1]. A linear code is called an LRC with locality $r$ if any coordinate value of a codeword can be recovered by other $r$ coordinates.

To verify the optimality of LRCs, many researchers have investigated on the bounds of LRCs. There are two frequently used bounds, one is Singleton-like bound[2] and the other is called CM bound[4] as follows:

$$k \leq \min_{t \in \mathbb{Z}^+} \left\{ tr + k_{opt}^{(q)}(n - t(r + 1), d) \right\},$$  \hspace{1cm} (1)

where $k_{opt}^{(q)}(n, d)$ is the largest possible dimension of a code for given length $n$, field size $q$ and minimum distance $d$. LRCs meeting bounds with equality are called optimal LRCs. For efficient and convenient hardware implementation, the constructions of LRCs over a small alphabet size is of particular interest[1]. Ref.[5-9] give constructions of binary LRCs, and some construction methods of ternary LRCs are designed in [10-12].

Linear complementary dual (LCD) code was first introduced by Massey in 1992[13]. LCD codes have been proved helpful for securing the communication against side-channel attacks (SCA) and fault injection attacks (FIA) by Carlet and Guilley[14]. Locally repairable codes that are also LCD codes may provide fast recovery when node failure occurs and protection against malicious attacks on distributed storage systems[15]. In recent years, Rajput et al. present several types of cyclic LRC-LCD codes, and determine an upper bound on the minimum distance of such codes[15]. In 2021, some cyclic LRC-LCD codes with $(r, \delta)$-locality and LCD codes with hierarchical-locality are also proposed[16].

In this paper, we are committed to constructing ternary LRCs that are also LCD codes. From now on, we use the term LCD-LRC to express an LRC that is also LCD for coherence, rather than LRC-LCD code. Firstly, a method of constructing ternary self-orthogonal code from arbitrary linear code is proposed. Then we present a construction of ternary LCD-LRC, and locality of the new code is upper bounded. All of our obtained LRCs are also LCD codes, and many of them are shown to be optimal in terms of the CM bound, which are instrumental in fast recovery of node failures, and also helpful for protection against some types of internet attacks. In addition, our resultant general linear codes are more flexible compared with previous work of [15, 16].

The rest of the paper is organized as follows. Section 2 gives some preliminaries about LRC and LCD codes. In Section 3, we derive a method of constructing ternary self-orthogonal codes from arbitrary linear code. Section 4 gives a construction of ternary LCD-LRC with small locality. Many optimal ternary LCD-LRCs attaining the CM bound are also presented. Section 5 concludes this paper.
2 Preliminaries

Let $F_3$ be a finite field with three elements and let $F_3^n$ be the $n$-dimensional row vector space over $F_3$. For two vectors $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n) \in F_3^n$, the Hamming distance between $x$ and $y$ is $d(x, y) = |\{i \mid x_i \neq y_i\}|$. A linear $[n, k]$ code $C$ is a $k$-dimensional subspace of $F_3^n$, and if the minimum Hamming distance $d = \min\{d(x, y) \mid x \neq y, x \in C, y \in C\}$, then $C$ is denoted as $C = [n, k, d]$. A code symbol with locality $r$ means its coordinate value can be presented as linear combinations of other $r$ coordinates. An $[n, k, d; r]$ code is an $[n, k, d]$ code with locality $r$ for all its symbols.

A generator matrix $G$ of code $C = [n, k, d]$ is a $k \times n$ matrix whose $k$ rows form a basis of $C$. The dual code of $C$ is defined as $C^\perp = \{x \in F_q^n \mid x \cdot c = 0, \forall c \in C\}$, where $x \cdot c$ is an Euclidean inner product. A generator matrix of $C^\perp$ is called a parity-check matrix of $C$. In this paper, all codes we discussed are ternary codes and we use $L(A)$ to indicate the number of columns of a matrix $A$.

A linear code $C$ is called a linear complementary dual (LCD) code if $C \cap C^\perp = \{0\}$, and is called a self-orthogonal code if $C \subset C^\perp$.

**Theorem 1** [13] Let $G$ and $H$ be a generator matrix and parity-check matrix of $C$ respectively, the following properties are equivalent:

(i) $C$ is a LCD code;
(ii) $C^\perp$ is a LCD code;
(iii) $G \cdot G^T$ is nonsingular;
(iv) $H \cdot H^T$ is nonsingular.

A linear code $C$ with generator matrix $G$ is self-orthogonal if and only if the Gram matrix $G \cdot G^T = 0_{k \times k}[17]$. To set forth our construction method, the following propositions are known:

**Proposition 2** (Wedderburn rank-one reduction formula)[18] Let $A$ be a $m \times n$ matrix over $F_q$. If there exist column vector $x \in F_q^n$ and $y \in F_q^m$ such that $z = y^T A x \neq 0$, then $\text{Rank}(A - z^{-1} A x y^T A) = \text{Rank}(A) - 1$.

**Proposition 3** Denote $G_{k,n_1+n_2} = (G_{k,n_1} \mid G_{k,n_2})$. If $G_{k,n_1}$ generates an $[n_1, k, d_1]$ self-orthogonal code and $G_{k,n_2}$ generates an $[n_2, k, d_2]$ LCD code, then $G_{k,n_1+n_2}$ generates an $[n_1 + n_2, k, d_1 + d_2]$ LCD code.

**Proposition 4** Denote $G_{k,n_1+n_2} = (G_{k,n_1} \mid G_{k,n_2})$. If $G_{k,n_1}$ generates an LRC with locality $r_1$ and and $G_{k,n_2}$ generates an LRC with locality $r_2$, then $G_{k,n_1+n_2}$ generates an LRC with locality $r \leq \max\{r_1, r_2\}$. 


3 New construction of self-orthogonal codes

Let \( C \) be an \([n, k, d]\) linear code with generator matrix \( P(0) \). Suppose \( A(0) = P(0)P(0)^T = (a_{p,q})_{k	imes k} \) is a nonzero \( k \times k \) symmetric matrix and \( I_k \) is the \( k \times k \) identity matrix.

Obviously, there are three cases for \( A(0) \):

Case 1: There exists a diagonal element 2.

Case 2: Every diagonal element not equals to 2, while there exists a diagonal element equals to 1.

Case 3: Every diagonal element equals to 0, while there exists an element not equals to 0.

Now we show how to build new matrices \( P(1) \) and \( A(1) = P(1)P(1)^T \), with \( \text{Rank}(A(1)) = \text{Rank}(A(0)) - 1 \).

If Case 1 occurs for \( A(0) \), denote \( a_{i,i} = 2 \) and \( I_k = (e_1, e_2, \ldots, e_k) \). Define column vectors \( x = y = e_i \). Calculate \( z = y^TA(0)x = a_{i,i} \), and \( z^{-1}A(0)xy^TA(0) = a_{i,i}^{-1}u_iu_i^T \), where \( u_i \) is the \( i \)-th column of \( A(0) \). As \( a_{i,i} = 2 \), \( A(0) - z^{-1}A(0)xy^TA(0) = P(0)P(0)^T + u_iu_i^T \). Suppose \( P(1) = (P(0) \mid u_i) \) and \( A(1) = P(1)P(1)^T \), then we can check that \( \text{Rank}(A(1)) = \text{Rank}(A(0)) - 1 \).

If Case 2 occurs for \( A(0) \), denote \( a_{i,i} = 1 \). Construct \( P(0)^\prime = (P(0) \mid u_i) \), and calculate \( A(0)^\prime = P(0)^\prime P(0)^\prime^T = A(0) + u_iu_i^T \). Denote \( A(0)^\prime = (a_{p,q}^\prime)_{k	imes k} \), then it’s easy to obtain that \( a_{i,i}^\prime = 2a_{i,i} = 2 \). Similar to Case 1, suppose \( P(1) = (P(0)^\prime \mid u_i^\prime) \) and \( A(1) = P(1)P(1)^T \), where \( u_i^\prime \) is the \( i \)-th column of \( A(0)^\prime \). Then we have \( \text{Rank}(A(1)) = \text{Rank}(A(0)) - 1 \).

If Case 3 occurs for \( A(0) \), denote \( a_{i,i} = 0 \) and \( a_{i,j} \neq 0 \), where \( 1 \leq i < j \leq k \). Define \( \lambda = a_{i,j} \) and \( E_{i,j} \) is a \( k \times k \) matrix with all zero elements except for 1 in \((i, j)\)-position. Construct \( S = I_k + \lambda E_{i,j} \) and \( P(0)^\prime = SP(0) \). Calculate \( A(0)^\prime = P(0)^\prime P(0)^\prime^T = (a_{p,q}^\prime)_{k	imes k} \). Notice that \( P(0)^\prime P(0)^\prime^T = SP(0)P(0)^TST = SA(0)S^T \), and \( a_{i,i}^\prime = \lambda a_{i,i} + \lambda a_{j,j} = 2\lambda a_{i,j} = 2 \). Similar to Case 1, suppose \( P(1) = (P(0)^\prime \mid u_i^\prime) \) and \( A(1) = P(1)P(1)^T \), where \( u_i^\prime \) is the \( i \)-th column of \( A(0)^\prime \), then we have \( \text{Rank}(A(1)) = \text{Rank}(A(0)) - 1 \).

From above discussion, we have constructed matrix \( P(1) \) and \( A(1) \) such that \( \text{Rank}(A(1)) = \text{Rank}(A(0)) - 1 \). Using mathematical induction on \( s \), assume that \( \text{Rank}(A(s)) = t \) and \( P(s) \) is obtained, where \( 1 \leq s \leq t - 1 \).

Calculate \( A(s) = P(s)P(s)^T = (a_{i,j}^{(s)})_{k	imes k} \). For the convenience of statement, we still use \( \text{Case 1}, \text{Case 2} \) or \( \text{Case 3} \) to describe which situation the square matrix \( A(s) \) in. Define

\[
P(s)^\prime = \begin{cases} 
P(s), & \text{Case 1} \\
(P(s) \mid u_i^{(s)}), & \text{Case 2} \\
(I_k + a_{i,j}^{(s)}E_{i,j})P(s), & \text{Case 3} 
\end{cases}
\]
New construction of ternary locally repairable codes that are also LCD codes

and calculate \( A^{(s)′} = P^{(s)′} P^{(s)′T} \). Then construct \( P^{(s+1)} = (P^{(s)′} \mid u_i^{(s)′}) \), where \( u_i^{(s)′} \) is the \( i \)-th column of \( A^{(s)′} \). Let \( A^{(s+1)} = P^{(s+1)} P^{(s+1)T} \), and easy to obtain \( \text{Rank}(A^{(s+1)}) = \text{Rank}(A^{(s)}) - 1 \).

Therefore, after \( t \) steps of iterations, one can obtain matrix \( A^{(t)} \) that generates a self-orthogonal code.

**Theorem 5** Let \( P^{(0)} \) be a generator matrix of \( C = [n, k, d] \), \( A^{(0)} = P^{(0)} P^{(0)T} \). If \( \text{Rank}(A^{(0)}) = t \) and \( l = \sum_{i=1}^{t} \{ L(A^{(i)}) - L(A^{(i-1)}) - 1 \} \), then there exists a self-orthogonal \([n + t + l, k, \geq d]\) code.

4 New construction of LCD-LRC

Assume \( C \) be an \([n, k, d]\) linear code with generator matrix \( P^{(0)} = (I_k \mid Q^{(0)}) \), and \( A^{(0)} = Q^{(0)} Q^{(0)T} \) is a nonzero \( k \times k \) matrix. On the basis of previous discussion, suppose \( \text{Rank}(A^{(0)}) = t \), we can obtain a matrix \( Q^{(t)} \) such that \( \text{Rank}(Q^{(t)} Q^{(t)T}) = 0 \) after \( t \) steps of iterations.

Suppose \( A^{(s)} = Q^{(s)} Q^{(s)T} \) for \( 0 \leq s \leq t \), if \( l = \sum_{i=1}^{t} \{ L(A^{(i)}) - L(A^{(i-1)}) - 1 \} \), then \( Q^{(t)} \) generates a \([n - k + t + l, k]\) self-orthogonal code. Moreover, let \( B^{(k)} \) be a \( k \times k \) matrix such that \( B^{(k)} B^{(k)T} \) is nonsingular, then \( G = (Q^{(t)} \mid B^{(k)}) \) generates an \([n + t + l, k]\) LCD code. In this section we still use Case 1, Case 2 and Case 3 to describe which circumstance the corresponding \( A^{(s)} \) in during the \((s + 1)\)-th step of iterations. The following description shows the construction of matrix \( B^{(k)} \).

To begin with, in the first step of iterations \( Q^{(0)′} \) and \( A^{(0)′} = Q^{(0)′} Q^{(0)′T} \) have been obtain. Then construct

\[
B^{(1)} = \begin{cases} 
(u_i), & \text{Case 1} \\
(u_i′), & \text{Case 2 or Case 3}
\end{cases}
\]

where \( u_i \) and \( u_i′ \) is the \( i \)-th column of \( A^{(0)} \) and \( A^{(0)′} \) respectively.

Next, assume that matrices \( Q^{(s)}, A^{(s)} = Q^{(s)} Q^{(s)T} \) and the corresponding \( B^{(s)} \) have been obtained, where \( 1 \leq s \leq t - 1 \). As shown in Section 3, in the \((s + 1)\)-th step of iterations, \( Q^{(s)′} \) and \( A^{(s)′} \) are presented. Then construct

\[
B^{(s+1)} = \begin{cases} 
(B^{(s)} \mid u_i^{(s)′}), & \text{Case 1 or Case 2} \\
((I_k + a_i^{(s) E_i,j}) B^{(s)} \mid u_i^{(s)′}), & \text{Case 3}
\end{cases}
\]

where \( u_i^{(s)′} \) is the \( i \)-th column of \( A^{(s)′} \), and \( a_i^{(s)} \) is the \((i, j)\)-position of \( A^{(s)} \).
Therefore, after \( t \) steps of iterations, we can obtain a \( k \times t \) matrix \( B^{(t)} \). Assume that in the constructing process: \( B^{(1)} \to B^{(2)} \to \cdots \to B^{(t)} \), the set of coordinates for chosen columns \( u_{i_1}', u_{i_2}', \cdots, u_{i_t}' \) is \( J = \{ i_1, i_2, \cdots, i_t \} \). Define a matrix \( M_{k \times (k-t)} \) by puncturing the \( m \)-th coordinate of \( I_k \), for every \( m \in J \). Then we can construct \( B^{(k)} = (B^{(t)} | M_{k \times (k-t)}) \).

**Proposition 6** Matrix \( B^{(k)} \) is nonsingular, and \( B^{(k)}B^{(k)T} \) is also nonsingular.

**Proof** Since elementary row operations do not change the rank of a matrix. Without loss of generality, let’s assume that in the constructing process: \( B^{(1)} \to B^{(2)} \to \cdots \to B^{(t)} \), the coordinates of chosen columns \( u_{i_1}', u_{i_2}', \cdots, u_{i_t}' \) follow the increasing order \( i_1 = 1, i_2 = 2, \cdots, i_t = t \).

If Case 3 never occurs for every \( A^{(s)} \), where \( 1 \leq s \leq t-1 \), matrix \( B^{(k)} \) is a \( k \times k \) lower triangular matrix, which is nonsingular obviously.

Supposing that in a certain step of iterations Case 3 happens, considering that \( 1 \leq i < j \leq k \), it is easy to see matrix \( B^{(k)} \) is still a \( k \times k \) lower triangular matrix. In addition, matrix \( B^{(k)}B^{(k)T} \) is also nonsingular, and \( B^{(k)} \) generates an LCD code.

As \( B^{(k)}B^{(k)T} \) is nonsingular, then \( G^{(k)} = (Q^{(t)} | B^{(k)}) \) generates an LCD code according to Proposition 3. Moreover, if \( t = k \), locality of the new code is upper bounded by locality of the one that \( Q^{(0)} \) generates.

**Proposition 7** If \( Q^{(0)} \) generates a code with locality \( r \) and \( \text{Rank}(A^{(0)}) = k \), then \( G^{(k)} = (Q^{(k)} | B^{(k)}) \) generates an LRC with locality \( r' \leq r \).

**Proof** Firstly, assume that in every step of the above iteration procedure, Case 2 or Case 3 never happens for \( A^{(0)}, A^{(1)}, \cdots, A^{(k-1)} \), then we can construct \( G^{(k)} = \)

\[
(Q^{(k)} | B^{(k)}) = (Q^{(0)} | u_{i_1}', u_{i_2}', \cdots, u_{i_k}'^{(k-1)'}) = (Q^{(0)} | u_{i_1}', u_{i_2}', \cdots, u_{i_k}'^{(k-1)'}).
\]

Clearly, \( (u_{i_1}', u_{i_2}', \cdots, u_{i_k}'^{(k-1)'}) \) generates a code with locality 1. Thereby, it can be shown by using Proposition 4 that \( G^{(k)} \) generates an LRC with locality \( r' \leq r \).

Next, consider other two specific situations.

1. Suppose that in the second step of the constructing procedure, Case 2 happens for \( A^{(1)} \), then we can construct \( G^{(k)} = \)

\[
G = (Q^{(0)} | u_{i_1}', 2u_{i_2}', u_{i_2}', \cdots, u_{i_k}'^{(k-1)'}) = (Q^{(0)} | u_{i_1}', u_{i_2}', \cdots, u_{i_k}'^{(k-1)'}).
\]
Note that \( (u_{i_1}', 2u_{i_2}'(1)', u_{i_2}'(1'), \ldots, u_{i_k}'(k-1)'), u_{i_1}', u_{i_2}'(1'), \ldots, u_{i_k}'(k-1)') \) also gives a code with locality 1. Then \( G^{(k)} \) generates an LRC with \( r' \leq r \).

(2) Suppose that in the second step of the constructing procedure, Case 3 happens for \( A^{(1)} \), then we can construct

\[
G^{(k)} = (SQ^{(0)} | S_{u_{i_1}}, u_{i_2}'(1'), \ldots, u_{i_k}'(k-1)'), S_{u_{i_1}}, u_{i_2}'(1'), \ldots, u_{i_k}'(k-1)')
\]

Similarly, \( (S_{u_{i_1}}, u_{i_2}'(1'), \ldots, u_{i_k}'(k-1')), S_{u_{i_1}}, u_{i_2}'(1'), \ldots, u_{i_k}'(k-1)') \) still gives a code with locality 1, and matrix \( SQ^{(0)} \) generates the same code as \( Q^{(0)} \) does. Then \( G^{(k)} \) generates an LRC with \( r' \leq r \).

If \( t < k \), matrix \( G^{(k)} = (Q^{(t)} | B^{(t)} | M_{k \times (k-t)}) \) still gives an LRC with small locality since every column vector in \( M_{k \times (k-t)} \) can be expressed by small number of other columns in \( G^{(k)} \). Therefore, we can construct LCD-LRCs from arbitrary linear codes.

**Theorem 8** Let \( P^{(0)} = (I_k | Q^{(0)}) \) be a generator matrix of \( C = [n, k, d] \), \( A^{(0)} = Q^{(0)}Q^{(0)^T} \). If \( \text{Rank}(A^{(0)}) = t \) and \( l = \sum_{i=1}^t \{L(A^{(i)}) - L(A^{(i-1)}) - 1\} \), then there exists an \( [n + t + l, k, \geq d] \) LCD-LRC with small locality.

Now we use generator matrices of best known ternary linear codes from codetable[19]. For a ternary distance-optimal code with length \( n \) and dimension \( k \), denote its generator matrix as \( P^{(0)} = (I_k | Q^{(0)}) \), then we can derive an LCD-LRC with length \( n + t + l \) and dimension \( k \) from Theorem 8.

**Example 1** From [19], let \( G_{3,26} = (I_3 | Q^{(0)}) \) be the generator matrix of \([26, 3, 18]\) code. After 3 iterations, we have

\[
Q^{(3)} = (Q^{(0)} | u_1', u_2'(1'), u_3'(2')) = \begin{pmatrix} 2 & 1 & 2 & 0 & 1 & 2 & 0 & 2 & 1 & 2 & 0 & 2 & 0 & 1 & 2 & 0 & 2 & 0 & 1 & 2 & 0 & 2 & 0 & 1 & 2 & 0 & 2 & 0 & 1 & 2 & 0 & 2 & 0 & 1 & 2 & 0 & 2 & 0 & 1 & 2 & 0 & 2 & 0 & 1 & 2 & 0 & 2 & 0 & 1\end{pmatrix}
\]

and

\[
B^{(3)} = \begin{pmatrix} 2 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 2 \end{pmatrix}
\]

Thus, \( G^{(3)} = (Q^{(3)} | B^{(3)}) \) generates an optimal [29, 3, 19; 2] LCD-LRC in terms of the CM bound, which is also distance-optimal due to the codetable[19].

Subsequently, a lot of LCD-LRCs can be obtained. Moreover, many of them are optimal according to the CM bound, as shown in Table 1.

**5 Conclusion**

In this paper, we combine LRC and LCD code by integration of enhancing repair efficiency and security of distributed storage systems. Following a
New construction of ternary locally repairable codes that are also LCD codes

Table 1 Some new optimal ternary LCD-LRCs

<table>
<thead>
<tr>
<th>No.</th>
<th>Initial code</th>
<th>Obtained code</th>
<th>No.</th>
<th>Initial code</th>
<th>Obtained code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>n=20, k=3</td>
<td>[23, 3, 13; 2]</td>
<td>43</td>
<td>n=26, k=6</td>
<td>[32, 6, 16; 3]</td>
</tr>
<tr>
<td>2</td>
<td>n=22, k=3</td>
<td>[25, 3, 16; 2]</td>
<td>44</td>
<td>n=31, k=6</td>
<td>[37, 6, 19; 3]</td>
</tr>
<tr>
<td>3</td>
<td>n=23, k=3</td>
<td>[26, 3, 16; 2]</td>
<td>45</td>
<td>n=38, k=6</td>
<td>[42, 6, 22; 2]</td>
</tr>
<tr>
<td>4</td>
<td>n=26, k=3</td>
<td>[29, 3, 19; 2]</td>
<td>46</td>
<td>n=49, k=6</td>
<td>[55, 6, 31; 2]</td>
</tr>
<tr>
<td>5</td>
<td>n=27, k=3</td>
<td>[30, 3, 19; 2]</td>
<td>47</td>
<td>n=56, k=6</td>
<td>[62, 6, 37; 2]</td>
</tr>
<tr>
<td>6</td>
<td>n=28, k=3</td>
<td>[31, 3, 19; 2]</td>
<td>48</td>
<td>n=57, k=6</td>
<td>[63, 6, 37; 3]</td>
</tr>
<tr>
<td>7</td>
<td>n=31, k=3</td>
<td>[34, 3, 22; 2]</td>
<td>49</td>
<td>n=58, k=6</td>
<td>[64, 6, 37; 3]</td>
</tr>
<tr>
<td>8</td>
<td>n=32, k=3</td>
<td>[35, 3, 22; 2]</td>
<td>50</td>
<td>n=63, k=6</td>
<td>[69, 6, 40; 2]</td>
</tr>
<tr>
<td>9</td>
<td>n=35, k=3</td>
<td>[38, 3, 25; 2]</td>
<td>51</td>
<td>n=64, k=6</td>
<td>[70, 6, 40; 2]</td>
</tr>
<tr>
<td>10</td>
<td>n=36, k=3</td>
<td>[39, 3, 25; 2]</td>
<td>52</td>
<td>n=67, k=6</td>
<td>[73, 6, 43; 2]</td>
</tr>
<tr>
<td>11</td>
<td>n=39, k=3</td>
<td>[42, 3, 28; 2]</td>
<td>53</td>
<td>n=68, k=6</td>
<td>[74, 6, 43; 2]</td>
</tr>
<tr>
<td>12</td>
<td>n=40, k=3</td>
<td>[43, 3, 28; 2]</td>
<td>54</td>
<td>n=84, k=6</td>
<td>[90, 6, 55; 2]</td>
</tr>
<tr>
<td>13</td>
<td>n=42, k=3</td>
<td>[46, 3, 29; 2]</td>
<td>55</td>
<td>n=85, k=6</td>
<td>[91, 6, 55; 2]</td>
</tr>
<tr>
<td>14</td>
<td>n=43, k=3</td>
<td>[46, 3, 29; 2]</td>
<td>56</td>
<td>n=86, k=6</td>
<td>[92, 6, 55; 2]</td>
</tr>
<tr>
<td>15</td>
<td>n=44, k=3</td>
<td>[47, 3, 31; 2]</td>
<td>57</td>
<td>n=90, k=6</td>
<td>[96, 6, 58; 2]</td>
</tr>
<tr>
<td>16</td>
<td>n=46, k=3</td>
<td>[49, 3, 31; 2]</td>
<td>58</td>
<td>n=98, k=6</td>
<td>[104, 6, 64; 2]</td>
</tr>
<tr>
<td>17</td>
<td>n=48, k=3</td>
<td>[51, 3, 34; 2]</td>
<td>59</td>
<td>n=85, k=7</td>
<td>[92, 7, 55; 4]</td>
</tr>
<tr>
<td>18</td>
<td>n=50, k=3</td>
<td>[54, 3, 34; 2]</td>
<td>60</td>
<td>n=86, k=7</td>
<td>[93, 7, 55; 3]</td>
</tr>
<tr>
<td>19</td>
<td>n=20, k=4</td>
<td>[24, 4, 13; 2]</td>
<td>61</td>
<td>n=128, k=7</td>
<td>[135, 7, 82; 2]</td>
</tr>
<tr>
<td>20</td>
<td>n=21, k=4</td>
<td>[25, 4, 13; 2]</td>
<td>62</td>
<td>n=129, k=7</td>
<td>[136, 7, 82; 2]</td>
</tr>
<tr>
<td>21</td>
<td>n=24, k=4</td>
<td>[28, 4, 16; 2]</td>
<td>63</td>
<td>n=140, k=7</td>
<td>[147, 7, 91; 2]</td>
</tr>
<tr>
<td>22</td>
<td>n=25, k=4</td>
<td>[29, 4, 16; 2]</td>
<td>64</td>
<td>n=141, k=7</td>
<td>[148, 7, 91; 2]</td>
</tr>
<tr>
<td>23</td>
<td>n=32, k=4</td>
<td>[36, 4, 22; 2]</td>
<td>65</td>
<td>n=146, k=7</td>
<td>[153, 7, 94; 2]</td>
</tr>
<tr>
<td>24</td>
<td>n=33, k=4</td>
<td>[37, 4, 22; 2]</td>
<td>66</td>
<td>n=147, k=7</td>
<td>[154, 7, 96; 2]</td>
</tr>
<tr>
<td>25</td>
<td>n=37, k=4</td>
<td>[41, 4, 25; 2]</td>
<td>67</td>
<td>n=150, k=7</td>
<td>[157, 7, 97; 2]</td>
</tr>
<tr>
<td>26</td>
<td>n=40, k=4</td>
<td>[44, 4, 28; 2]</td>
<td>68</td>
<td>n=154, k=7</td>
<td>[161, 7, 100; 2]</td>
</tr>
<tr>
<td>27</td>
<td>n=41, k=4</td>
<td>[45, 4, 28; 2]</td>
<td>69</td>
<td>n=194, k=7</td>
<td>[201, 7, 127; 3]</td>
</tr>
<tr>
<td>28</td>
<td>n=42, k=4</td>
<td>[46, 4, 28; 2]</td>
<td>70</td>
<td>n=140, k=8</td>
<td>[148, 8, 89; 2]</td>
</tr>
<tr>
<td>29</td>
<td>n=43, k=4</td>
<td>[47, 4, 28; 2]</td>
<td>71</td>
<td>n=144, k=8</td>
<td>[152, 8, 92; 2]</td>
</tr>
<tr>
<td>30</td>
<td>n=47, k=4</td>
<td>[51, 4, 31; 2]</td>
<td>72</td>
<td>n=145, k=8</td>
<td>[153, 8, 92; 2]</td>
</tr>
<tr>
<td>31</td>
<td>n=50, k=4</td>
<td>[54, 4, 34; 2]</td>
<td>73</td>
<td>n=146, k=8</td>
<td>[154, 8, 92; 2]</td>
</tr>
<tr>
<td>32</td>
<td>n=20, k=5</td>
<td>[25, 5, 13; 3]</td>
<td>74</td>
<td>n=150, k=8</td>
<td>[158, 8, 94; 2]</td>
</tr>
<tr>
<td>33</td>
<td>n=21, k=5</td>
<td>[26, 5, 13; 3]</td>
<td>75</td>
<td>n=164, k=8</td>
<td>[172, 8, 103; 2]</td>
</tr>
<tr>
<td>34</td>
<td>n=25, k=5</td>
<td>[30, 5, 16; 3]</td>
<td>76</td>
<td>n=172, k=8</td>
<td>[180, 8, 109; 2]</td>
</tr>
<tr>
<td>35</td>
<td>n=30, k=5</td>
<td>[35, 5, 19; 2]</td>
<td>77</td>
<td>n=178, k=8</td>
<td>[186, 8, 112; 2]</td>
</tr>
<tr>
<td>36</td>
<td>n=31, k=5</td>
<td>[36, 5, 19; 2]</td>
<td>78</td>
<td>n=184, k=8</td>
<td>[192, 8, 116; 2]</td>
</tr>
<tr>
<td>37</td>
<td>n=34, k=5</td>
<td>[39, 5, 22; 2]</td>
<td>79</td>
<td>n=188, k=8</td>
<td>[196, 8, 119; 2]</td>
</tr>
<tr>
<td>38</td>
<td>n=35, k=5</td>
<td>[36, 5, 19; 2]</td>
<td>80</td>
<td>n=192, k=8</td>
<td>[200, 8, 122; 2]</td>
</tr>
<tr>
<td>39</td>
<td>n=38, k=5</td>
<td>[43, 5, 25; 2]</td>
<td>81</td>
<td>n=193, k=8</td>
<td>[201, 8, 122; 2]</td>
</tr>
<tr>
<td>40</td>
<td>n=39, k=5</td>
<td>[40, 5, 22; 2]</td>
<td>82</td>
<td>n=210, k=8</td>
<td>[218, 8, 133; 2]</td>
</tr>
<tr>
<td>41</td>
<td>n=47, k=5</td>
<td>[52, 5, 31; 2]</td>
<td>83</td>
<td>n=177, k=9</td>
<td>[186, 9, 109; 2]</td>
</tr>
<tr>
<td>42</td>
<td>n=48, k=5</td>
<td>[53, 5, 31; 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

detailed mathematical derivation, we give a method of constructing ternary self-orthogonal code. Based on this, we propose a new construction of ternary LCD-LRC. As a result, many optimal ternary LCD-LRCs are presented therewith. In view of the fact that self-orthogonal code can be used to construct quantum error-correcting codes, which protect quantum information in quantum computations and quantum communications, that could be another application scenario of this method.
Acknowledgments. This work is supported by National Natural Science Foundation of China under Grant No.U21A20428, 11901579, Natural Science Foundation of Shaanxi under Grant No.2022JQ-046, 2023-JC-QN-0033.

References


New construction of ternary locally repairable codes that are also LCD codes

Intelligence and Design (ISCID), (2020) 155-158.


