

PAPER Special Section on Signal Design and Its Applications in Communications

# Block-Punctured Binary Simplex Codes for Local and Parallel Repair in Distributed Storage Systems\*\*

Jung-Hyun KIM<sup>†\*a)</sup>, Min Kyu SONG<sup>††b)</sup>, Nonmembers, and Hong-Yeop SONG<sup>††c)</sup>, Member

**SUMMARY** In this paper, we investigate how to obtain binary locally repairable codes (LRCs) with good locality and availability from binary Simplex codes. We first propose a Combination code having the generator matrix with all the columns of positive weights less than or equal to a given value. Such a code can be also obtained by puncturing all the columns of weights larger than a given value from a binary Simplex Code. We call by block-puncturing such puncturing method. Furthermore, we suggest a heuristic puncturing method, called subblock-puncturing, that punctures a few more columns of the largest weight from the Combination code. We determine the minimum distance, locality, availability, joint information locality, joint information availability of Combination codes in closed-form. We also demonstrate the optimality of the proposed codes with certain choices of parameters in terms of some well-known bounds.

**key words:** distributed storage systems, locally repairable codes, locality, availability, simplex codes

## 1. Introduction

Recently, due to the dramatically boost in data, distributed storage systems are becoming progressively more important. To guarantee the reliability against storage node failures, various coding techniques have been applied to the systems. The simplest and most commonly used way is replication, where every node is replicated several times. Such systems are very easy to implement, but severely inefficient in storage space, equipments, and devices. In contrast, maximum distance separable (MDS) codes [1] have minimal storage overhead for a given reliability requirement, but suffer from inefficiency in the repair process.

Motivated by the desire to reduce repair cost of codes for distributed storage systems, an interesting notion, locality, was introduced in [2]. More precisely, if a symbol of a code  $C$  can be expressed as a function of at most  $r_1$  other symbols in  $C$ , the symbol is said to have locality  $r_1$ . It is said that a code  $C$  has all symbol locality  $r_1$  if every symbol in  $C$  has the locality at most  $r_1$ . For a systematic code, it is said that the code has information symbol locality  $r_1$  if its every

information symbol has the locality at most  $r_1$ . After that, the authors of [3] proposed a generalized notion of locality for multiple node failures,  $(r_l, l)$ -cooperative locality: any set of  $l$  symbols are expressed as functions of at most  $r_l$  other symbols. In this paper, we call this  $l$ -locality  $r_l$  briefly. The authors in [4]–[6] suggested joint locality and joint information locality which consider multiple values of  $l$  instead of a single value of  $l$ . We call codes having the properties locally repairable codes (LRCs) [7]. Some bounds for LRCs have been reported in [2], [6], [8], [9], and some constructions of LRCs have also been proposed in [4], [5], [7], [10]–[12].

In addition to the locality, the availability was introduced in [13] as another important property of LRCs. A symbol of a code is said to have  $(r_1, t_1)$ -availability if it can be recovered from any single set of  $t_1$  disjoint repair sets of other symbols, each set of size at most  $r_1$ . We refer a systematic code to an LRC with  $(r_1, t_1)_i$ -availability if its every information symbol has the locality at most  $r$  and availability at least  $t_1$  and a code to an LRC with  $(r_1, t_1)_a$ -availability if its every symbol has the locality at most  $r$  and availability at least  $t_1$ . An LRC with  $(r_1, t_1)_a$ -availability also tolerates multiple node failures up to  $t_1$  failures in any local repair process. Moreover, such LRCs ensure parallel reads for each symbol, which is appealing in distributed storage systems containing so-called *hot data* that is frequently and simultaneously accessed by many users. Some bounds for LRCs with availability have been reported in [13]–[15], and some constructions of such LRCs have also been proposed in [13]–[20]. After that, the authors of [21] extended the availability for one symbol into the availability for multiple symbols, and defined joint availability.

In this paper, we investigate how to construct new binary LRCs with good locality and availability from binary Simplex codes. Binary Simplex codes (dual of Hamming codes or punctured Hadamard codes) [1] are well-known LRCs attaining existing upper bounds [9], [15] on the code dimension and minimum distance which take into account the field size, locality, and availability. An  $[n = 2^k - 1, k, d = 2^{k-1}]_2$  Simplex code has joint locality  $(r_1, r_2)_a = (2, 3)$  [5] and  $(2, d-1)_a$ -availability [15]. Even though the Simplex codes have good locality and availability properties, they have extremely low code rate  $\frac{k}{2^k-1}$ . To obtain high rate codes without destroying the minimum distance, locality, and availability properties of the Simplex codes as possible, we propose a new construction of LRCs by puncturing the Simplex codes. We consider two types of the puncturing, “block-puncturing” and “subblock-puncturing”, as follows:

\*Manuscript received January 6, 2018.

†Manuscript revised July 24, 2018.

<sup>a)</sup>The author is with Samsung Electronics, Seoul, Korea.

<sup>††</sup>The authors are with Yonsei University, Seoul, Korea.

<sup>\*</sup>He had contributed to this work while he had been with Yonsei University, Seoul, Korea.

<sup>\*\*</sup>This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2017R1A2B4011191).

a) E-mail: jh.kim06@yonsei.ac.kr

b) E-mail: mk.song@yonsei.ac.kr

c) E-mail: hysong@yonsei.ac.kr (Corresponding author)

DOI: 10.1587/transfun.E101.A.2374

*Block-puncturing:* We consider a puncturing method which drops all the columns having weights larger than a given value in the generator matrix. We note that the authors in [22] also considered a similar puncturing method, called spherically puncturing, to increase the minimum distance of the first order Reed-Muller codes and Hadamard codes. They derived the minimum distance of two families of single-layer spherically punctured codes in closed-form. However, our block-punctured binary Simplex codes are a special case of multilayer spherically punctured Hadamard codes. Thus, all the results of block-punctured binary Simplex codes are new.

*Subblock-puncturing:* We also consider a heuristic puncturing method which drops few more columns of the largest weight after block-puncturing. To minimize the loss of the minimum distance, locality, and availability after the puncturing, we propose a heuristic algorithm. We confirmed that the algorithm works well by a computer simulation.

In the remaining of this paper, we call by Combination codes block-punctured Simplex codes, and by punctured Combination codes the obtained codes by subblock-puncturing Combination codes. We determine the minimum distance, joint locality  $(r_1, r_2)_a$ , joint information locality, availability, and joint information availability of Combination codes in closed-form. We also discuss the optimality of the Combination codes and punctured Combination codes in terms of some well-known upper bounds on the code dimension, minimum distance, and code rate. The result shows that the proposed codes with certain choices of parameters attain bounds on the code dimension and minimum distance which take into account the field size, locality, and availability. Moreover, our codes have the best code rate among the existing LRCs with the same availability.

The rest of this paper is organized as follows. In Sect. 2, we propose new binary LRCs having good locality and availability properties. We provide the minimum distance, locality, and availability properties of the proposed codes. In Sect. 3, we show the optimality of the proposed codes in terms of well-known bounds on the code dimension, minimum distance, and code rate. In Sect. 4, we conclude the paper.

## 2. Block-Punctured Binary Simplex Codes

We start with the definition of the proposed codes, Combination  $(k, w)$  codes. We use the notation  $[n, k, d]_q$  to refer to the parameters of a  $q$ -ary linear LRC of the code length  $n$ , dimension  $k$ , and minimum distance  $d$ . For a positive integer  $n$ , we denote by  $[n]$  the set of integers  $\{1, 2, \dots, n\}$ .

**Definition 1** Let two integers  $k$  and  $w$  satisfy  $k \geq 3$  and  $2 \leq w \leq k$ . For  $i \in [w]$ , let  $G_i$  be the  $k \times \binom{k}{i}$  matrix consisting of all the columns of weight  $i$ . Let  $G = [G_1 | G_2 | \dots | G_w]$ . Then, the binary linear code  $C$  generated by  $G$  has length  $\sum_{i=1}^w \binom{k}{i}$  and dimension  $k$ . We call  $C$  a Combination  $(k, w)$  code.  $\square$

In the remaining of this paper, we will fix  $G$  the generator matrix defined in Definition 1 for Combination  $(k, w)$

codes. We do not consider the case of  $k \leq 2$  and the case of  $w = 1$ , since they are trivial. We note that the Combination  $(k, 2)$  code and a Complete graph code [4] are permutation equivalent, and that the Combination  $(k, k)$  code and a Simplex code [1] are permutation equivalent. For  $w < k$ , the Combination  $(k, w)$  code is the result of block-puncturing of the Simplex code. That is,  $G$  can be also obtained by deleting all the columns of weight larger than  $w$  from the generator matrix of the Simplex code. Later, we will further consider a heuristic puncturing method, called subblock-puncturing, that punctures a few more columns of weight  $w$ .

Now, using the following Lemma 1 and Lemma 2, we derive the minimum distance of Combination  $(k, w)$  codes in Theorem 1.

**Lemma 1** Let  $C$  be a Combination  $(k, w)$  code with its generator matrix  $G$ . For  $s = 1, 2, \dots, k$ , a codeword obtained by adding any  $s$  rows of  $G$  has weight  $W(s)$ , which is given by the following:

$$W(s) = \sum_{i=1}^w \sum_{\substack{1 \leq j \leq i \\ j \text{ odd}}} \binom{s}{j} \binom{k-s}{i-j}. \quad (1)$$

**Proof** For every  $i \in [w]$ , let  $M_i$  be a  $s$ -by- $\binom{k}{i}$  submatrix of  $G_i$  corresponding to the selected  $s$  rows. Then,  $W(s)$  is the sum of the numbers of odd weight columns in  $M_i$  for all  $i \in [w]$ . Therefore, we obtain the above equation. ■

**Lemma 2** Let  $C$  be a Combination  $(k, w)$  code with its generator matrix  $G$ . For  $s = 2, 3, \dots, k$ , choose any  $s$  rows in  $G$ . We denote by  $\mathbf{x}$  any one of them, and consider a submatrix  $M'$  consisting of the remaining  $s - 1$  rows. We denote by  $W_1(s - 1)$  the number of odd weight columns in  $M'$  out of the columns corresponding to 1's positions of  $\mathbf{x}$ . We denote by  $W_0(s - 1)$ , similarly, those corresponding to 0's positions of  $\mathbf{x}$ . Then,  $W_1(s - 1) \leq W_0(s - 1)$ .

**Proof** Since each row of  $G$  is permutation equivalent, we can assume that any  $s$  rows are selected. It is also the same with  $\mathbf{x}$ . Then,  $W_0(s - 1)$  and  $W_1(s - 1)$  can be written as follows:

$$W_0(s - 1) = \sum_{i=1}^w \sum_{\substack{1 \leq j \leq i \\ j \text{ odd}}} \binom{s-1}{j} \binom{(k-1)-(s-1)}{i-j},$$

$$W_1(s - 1) = \sum_{i=2}^w \sum_{\substack{1 \leq j \leq i-1 \\ j \text{ odd}}} \binom{s-1}{j} \binom{(k-1)-(s-1)}{(i-1)-j}.$$

Therefore, we finally obtain  $W_1(s - 1) \leq W_0(s - 1)$ . ■

**Theorem 1** Let  $C$  be a Combination  $(k, w)$  code. Then, the minimum distance  $d$  of  $C$  is

$$d = \sum_{i=1}^w \binom{k-1}{i-1}.$$

**Proof** Without loss of generality, consider the top  $s$  rows in the generator matrix  $G$  of  $C$ . We can always suppose that the first row of  $G$  is of the form  $(1, \dots, 1, 0, \dots, 0)$  with column permutation. Then,  $W(s)$  in Lemma 1 can be written as follows:

$$W(s) = \begin{cases} W(1), & \text{for } s = 1, \\ W(1) - W_1(s-1) + W_0(s-1), & \text{for } 2 \leq s \leq k, \end{cases}$$

where  $W_1(s-1)$  and  $W_0(s-1)$  are the numbers of odd weight columns defined in Lemma 2. Since  $W_1(s-1) \leq W_0(s-1)$  by Lemma 2, we have  $W(1) \leq W(s)$ . Finally, using the result of Lemma 1, the minimum distance  $d$  is

$$d = \min_{s \in [k]} W(s) = W(1) = \sum_{i=1}^w \binom{k-1}{i-1}.$$

■

**Theorem 2** Let  $C$  be a Combination  $(k, w)$  code. Then,  $C$  has joint locality  $(r_1, r_2)_a = (2, 3)$ .

**Proof** Consider the generator matrix  $G$  of  $C$ . For  $r_1 = 2$ , we have to show that an erased column of  $G$  can be expressed as a linear combination of at most two other columns. For  $r_2 = 3$ , we have to show that two erased columns can be expressed as a linear combination of at most three other columns.

1. For  $r_1 = 2$ , let  $\mathbf{g}_{E_1}$  be the erased column in  $G$  and  $E_1$  be the set of non-zero row indices of  $\mathbf{g}_{E_1}$ . Then, as shown in Fig. 1, there always exist a column  $\mathbf{g}_{R_1}$  of weight 1,  $R_1 \neq E_1$ , and a corresponding column  $\mathbf{g}_{R_2} = \mathbf{g}_{E_1} + \mathbf{g}_{R_1}$ .
2. For  $r_2 = 3$ , let  $\mathbf{g}_{E_1}$  and  $\mathbf{g}_{E_2}$  be the two erased columns, and  $E_1$  and  $E_2$  be the sets of non-zero row indices of  $\mathbf{g}_{E_1}$  and  $\mathbf{g}_{E_2}$ , respectively. Without loss of generality, assume that  $|E_1| \geq |E_2|$ . Then, as shown in Fig. 2, there always exists a column  $\mathbf{g}_{R_1}$  and corresponding columns  $\mathbf{g}_{R_2} = \mathbf{g}_{E_1} + \mathbf{g}_{R_1}$  and  $\mathbf{g}_{R_3} = \mathbf{g}_{E_2} + \mathbf{g}_{R_1}$ .

■

Even though a bound of  $\ell$ -locality for the Simplex codes is introduced in [3], it is not easy to find the exact value of  $\ell$ -locality for Combination  $(k, w)$  codes. In this paper, as a first step, we provide the joint information locality of Combination  $(k, w)$  codes as the following.

**Theorem 3** Let  $C$  be a Combination  $(k, w)$  code. Then,  $C$  has joint information locality

$$r_\ell = \begin{cases} \ell, & \text{for } \ell, w \geq 3, \\ \ell + 1, & \text{otherwise.} \end{cases}$$

**Proof** For  $\ell = 1$  and 2, we have  $r_\ell = \ell + 1$  from Theorem 2. For  $\ell \geq 3$ , we consider two cases: (1)  $w = 2$ . (2)  $w \geq 3$ . Consider the generator matrix  $G$  of  $C$ . Let  $\mathbf{g}_{E_1}, \mathbf{g}_{E_2}, \dots, \mathbf{g}_{E_\ell}$  be the erased columns.

Case (1): For  $\ell \geq 3$ ,  $w = 2$ , the minimum distance  $d$  is  $k$ , and thus  $\ell \leq k - 1$ . Then, we can always choose a column

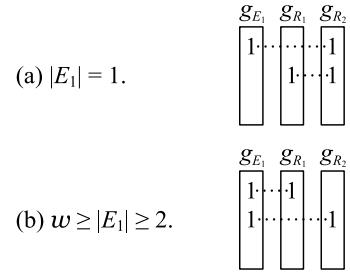


Fig. 1 Two cases for  $r_1 = 2$ .

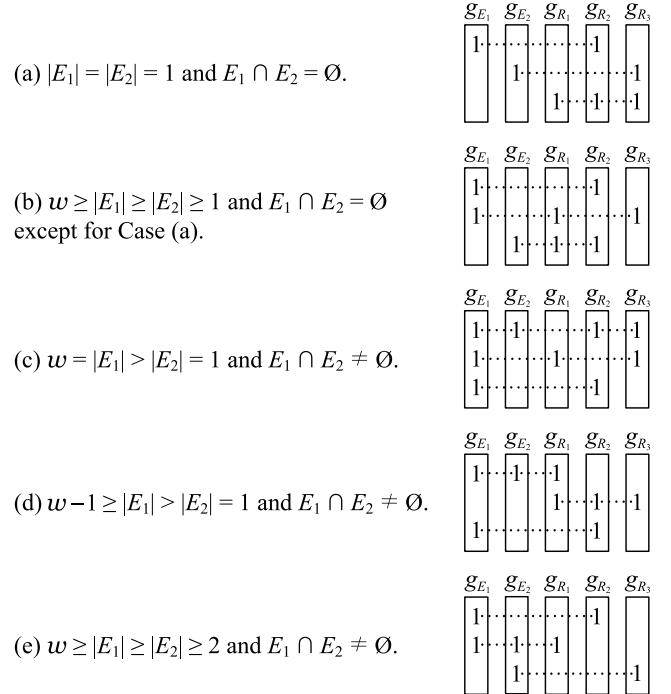


Fig. 2 Five cases for  $r_2 = 3$ .

$\mathbf{g}_A$  of weight 1 which is not erased. Using the column, we can reconstruct the erased columns from the following  $\ell + 1$  other columns.

$$\begin{aligned} \mathbf{g}_{R_1} &= \mathbf{g}_A, & \mathbf{g}_{R_2} &= \mathbf{g}_A + \mathbf{g}_{E_1}, \\ \mathbf{g}_{R_3} &= \mathbf{g}_A + \mathbf{g}_{E_2}, \dots, & \mathbf{g}_{R_{\ell+1}} &= \mathbf{g}_A + \mathbf{g}_{E_\ell}. \end{aligned}$$

Case (2): For  $\ell, w \geq 3$ , we can always reconstruct the erased columns from the following  $\ell$  other columns.

$$\begin{aligned} \mathbf{g}_{R_1} &= \mathbf{g}_{E_1} + \mathbf{g}_{E_2} + \mathbf{g}_{E_3}, & \mathbf{g}_{R_2} &= \mathbf{g}_{E_1} + \mathbf{g}_{E_2}, \\ \mathbf{g}_{R_3} &= \mathbf{g}_{E_1} + \mathbf{g}_{E_3}, \dots, & \mathbf{g}_{R_\ell} &= \mathbf{g}_{E_1} + \mathbf{g}_{E_\ell}. \end{aligned}$$

■

Now, using the following Lemma 3 and Lemma 4, we derive the availability of Combination  $(k, w)$  codes in Theorem 4.

**Lemma 3** Let  $C$  be a Combination  $(k, w)$  code with its generator matrix  $G$ . Then, the availability  $T(i)$  of a symbol

corresponding to a column of weight  $i$  in  $G$  is

$$T(i) = \frac{1}{2} \sum_{u=0}^i \binom{i}{u} \sum_{v=0}^{\min(w-u, w-(i-u))} \binom{k-i}{v} - 1. \quad (2)$$

**Proof** Recall that 1-locality  $r_1$  is 2 for Combination  $(k, w)$  codes by Theorem 2. The availability of a symbol is, therefore, the number of disjoint sets of two columns to repair the corresponding column in  $G$ . More precisely, assume that  $A$  is the set of non-zero row indices for an erased column of weight  $i$ . Then, the availability of the symbol corresponding to  $A$  is the number of disjoint choices for two distinct sets of integers,  $B$  and  $C$ , such that  $A = (B \cup C) - (B \cap C)$  where  $|A| = i$ , and  $0 < |C| \leq |B| \leq w$ . There are three cases for the choice:

1. If  $B \cap C = C$ , the number of choices is

$$\sum_{v=1}^{w-i} \binom{k-i}{v}.$$

2. If  $B \cap C = \emptyset$ , the number of choices is

$$\frac{1}{2} \sum_{u=1}^{i-1} \binom{i}{u}.$$

3. Otherwise, the number of choices is

$$\frac{1}{2} \sum_{u=1}^{i-1} \binom{i}{u} \sum_{v=1}^{\min(w-u, w-(i-u))} \binom{k-i}{v}.$$

The sum of the above three gives (2). ■

**Lemma 4** Let  $C$  be a Combination  $(k, w)$  code with its generator matrix  $G$ . For a possible positive integers  $i$ , let  $T(i)$  be the availability of a symbol corresponding to the column of weight  $i$  in  $G$ . Then,  $T(\cdot)$  satisfies the following:

$$T(2j-1) = T(2j) \geq T(2j+1), \quad (3)$$

i.e.  $T(1) = T(2) \geq T(3) = T(4) \geq T(5) = T(6) \cdots T(w)$ .

**Proof** Observe that there are  $\binom{k}{i}$  columns of weight  $i$  in  $G$ , and all the  $\binom{k}{i}$  symbols corresponding to these columns have the same availability, which is denoted by  $T(i)$ . Now, we rewrite (2) in Lemma 2 as follows

$$\begin{aligned} T(2j-1) &= \sum_{u=j}^{2j-1} \binom{2j-1}{u} \sum_{v=0}^{w-u} \binom{k-(2j-1)}{v} - 1, \\ T(2j) &= \sum_{u=j}^{2j} \binom{2j}{u} \sum_{v=0}^{w-u} \binom{k-2j}{v} - \frac{1}{2} \binom{2j}{j} \sum_{v=0}^{w-j} \binom{k-2j}{v} - 1. \end{aligned}$$

This gives  $T(2j-1) = T(2j)$ .

For the inequality in (3), we rewrite (2) as follows

$$T(2j) = \sum_{u=j+1}^{2j} \binom{2j}{u} \sum_{v=0}^{w-u} \binom{k-2j}{v} + \frac{1}{2} \binom{2j}{j} \sum_{v=0}^{w-j} \binom{k-2j}{v} - 1,$$

$$T(2j+1) = \sum_{u=j+1}^{2j+1} \binom{2j+1}{u} \sum_{v=0}^{w-u} \binom{k-(2j+1)}{v} - 1.$$

Now, it is easy to check that  $T(2j) - T(2j+1) \geq 0$ . ■

**Theorem 4** Let  $C$  be a Combination  $(k, w)$  code. Then, the availability  $t_1$  of  $C$  is

$$t_1 = \frac{1}{2} \sum_{u=0}^w \binom{w}{u} \sum_{v=0}^{\min(w-u, u)} \binom{k-w}{v} - 1.$$

In particular,  $t_1 = d-1$  when  $w = 2, k-1$  or  $k$ , and  $t_1 = d-2$  when  $w = k-2 > 2$ .

**Proof** By Lemma 3 and Lemma 4, the availability  $t_1$  of  $C$  becomes

$$t_1 = \min_{i \in [w]} T(i) = T(w) = \frac{1}{2} \sum_{u=0}^w \binom{w}{u} \sum_{v=0}^{\min(w-u, u)} \binom{k-w}{v} - 1.$$

We note that we can obtain the availabilities of a Complete graph code [4] and a binary Simplex code [1] using a Combination  $(k, 2)$  code and a Combination  $(k, k)$  code, respectively. The availability of a Complete graph code is also obtained from its graph representation [4], [5], and that of a binary Simplex code is also obtained from its one-step majority-logic decoding structure [15], [23].

Based on the proof of Theorem 3, we provide the joint information availability of Combination  $(k, w)$  codes.

**Theorem 5** Let  $C$  be a Combination  $(k, w)$  code. Then,  $C$  has joint information availability

$$(r_\ell, t_\ell)_i = \begin{cases} (\ell+1, \sum_{v=1}^{\min(w-1, k-\ell)} \binom{k-\ell}{v}), & \text{for } \ell = 1, 2, \\ (\ell+1, k-\ell), & \text{for } \ell \geq 3, w = 2, \\ (\ell, \min(\ell, w)-2), & \text{otherwise.} \end{cases}$$

**Proof** By Theorem 3,  $C$  has joint information locality  $(r_1, r_2)_i = (2, 3)$ . Consider the generator matrix  $G$  of  $C$ . To obtain the value of  $t_1$ , let  $\mathbf{g}_{E_1}$  be the erased column. Choose a column  $\mathbf{g}_A$  of weight  $a < w$ ,  $A \neq E_1$ . Then, we can reconstruct the erased column from the following two other columns.

$$\mathbf{g}_{R_1} = \mathbf{g}_A, \quad \mathbf{g}_{R_2} = \mathbf{g}_A + \mathbf{g}_{E_1}.$$

When  $a$  is fixed,  $\binom{k-1}{a}$  disjoint repairs sets exist, and there are  $w-1$  choices for  $a$ . Therefore,  $t_1 = \sum_{v=1}^{w-1} \binom{k-1}{v}$ .

Now, to obtain the value of  $t_2$ , let  $\mathbf{g}_{E_1}$  and  $\mathbf{g}_{E_2}$  be the erased columns. Choose a column  $\mathbf{g}_A$  of weight  $a < w$ ,  $A \neq E_1, E_2$ . Then, we can reconstruct the erased columns from the following three other columns.

$$\mathbf{g}_{R_1} = \mathbf{g}_A, \quad \mathbf{g}_{R_2} = \mathbf{g}_A + \mathbf{g}_{E_1}, \quad \mathbf{g}_{R_3} = \mathbf{g}_A + \mathbf{g}_{E_2}.$$

When  $a$  is fixed,  $\binom{k-2}{a}$  disjoint repairs sets exist. When

$w < k$ , there are  $w - 1$  choices for  $a$  and, when  $w = k$ , there are  $k - 2$  choices for  $a$ . Therefore,  $t_2 = \sum_{v=1}^{\min(w-1, k-2)} \binom{k-2}{v}$ .

Now, consider the case of  $\ell \geq 3$ . Since  $w \geq 2$  by the definition, we consider two cases: (1)  $\ell \geq 3$ ,  $w = 2$ . (2)  $\ell, w \geq 3$ . Consider the generator matrix  $G$  of  $C$ . Let  $\mathbf{g}_{E_1}, \mathbf{g}_{E_2}, \dots, \mathbf{g}_{E_\ell}$  be the erased columns.

Case (1): For  $\ell \geq 3$ ,  $w = 2$ , the minimum distance  $d$  is  $k$ , and thus  $\ell \leq k - 1$ . Then, we can always choose a column  $\mathbf{g}_A$  of weight 1 which is not erased. Using the column, we can reconstruct the erased columns from the following  $\ell + 1$  other columns.

$$\begin{aligned}\mathbf{g}_{R_1} &= \mathbf{g}_A, \quad \mathbf{g}_{R_2} = \mathbf{g}_A + \mathbf{g}_{E_1}, \\ \mathbf{g}_{R_3} &= \mathbf{g}_A + \mathbf{g}_{E_2}, \quad \dots, \quad \mathbf{g}_{R_{\ell+1}} = \mathbf{g}_A + \mathbf{g}_{E_\ell}.\end{aligned}$$

For a fixed column  $\mathbf{g}_A$ , there is only one disjoint repair set. The number of choices of such a column  $\mathbf{g}_A$  is  $k - \ell$ , and the repair sets are disjoint each other.

Case (2): For  $\ell, w \geq 3$ , we can always reconstruct the erased columns from the following  $\ell$  other columns. Here,  $i$  is a positive integer,  $3 \leq i \leq \min(\ell, w)$ .

$$\begin{aligned}\mathbf{g}_{R_1} &= \sum_{u=1}^i \mathbf{g}_{E_u}, \quad \mathbf{g}_{R_2} = \sum_{\substack{1 \leq u \leq i \\ u \neq 1}} \mathbf{g}_{E_u}, \quad \dots, \quad \mathbf{g}_{R_i} = \sum_{\substack{1 \leq u \leq i \\ u \neq i-1}} \mathbf{g}_{E_u}, \\ \mathbf{g}_{R_{i+1}} &= \sum_{u=1}^{i-2} \mathbf{g}_{E_u} + \mathbf{g}_{E_{i+1}}, \quad \dots, \quad \mathbf{g}_{R_\ell} = \sum_{u=1}^{i-2} \mathbf{g}_{E_u} + \mathbf{g}_{E_\ell}.\end{aligned}$$

For a fixed value  $i$ , there is only one disjoint repair set. For all the possible  $i$ , the repair sets are disjoint each other. ■

In addition, we propose another puncturing method, called subblock-puncturing, which deletes a few more columns of weight  $w$  from the generator matrix of a Combination  $(k, w)$  code. Let  $p$  be the number of punctured columns, called puncturing length. If  $p$  becomes  $\binom{k}{w}$ , the punctured code is the same with a Combination  $(k, w - 1)$  code. Thus, we only consider  $1 \leq p \leq \binom{k}{w} - 1$ .

Consider the generator matrix  $G = [G_1 | G_2 | \dots | G_w]$  of a Combination  $(k, w)$  code. To obtain a punctured Combination  $(k, w)$  code by the subblock-puncturing, we use the following generator matrix construction algorithm.

#### Algorithm 1 Construction of the generator matrix of a punctured Combination $(k, w)$ code by the subblock-puncturing

- 1: Set  $y = \binom{k}{w}$ .
- 2: **for** every column set of size  $p$  in  $G_w$  **do**
- 3:     Make a  $k \times p$  matrix  $\Phi$ .
- 4:     Calculate the difference  $x$  between maximum and minimum weights of rows in  $\Phi$ .
- 5:     **if**  $y \geq x$  **then**
- 6:         Set  $y = x$  and  $\Psi = \Phi$ .
- 7:     **end if**
- 8: **end for**
- 9: Delete columns of  $G$  which are the same with those of  $\Psi^\dagger$ .

<sup>†</sup>We note that  $\Psi$  is always defined since  $x$  is smaller than  $\binom{k}{w}$ .

Using the subblock-puncturing, we can obtain some optimal codes in terms of the following bounds. For LRCs with  $(r_1, t_1)_i$ -availability, an upper bound on the minimum distance was presented in [13] under the following condition. Let  $C$  be an  $[n, k, d]_q$  LRC. Every information symbol of  $C$  has  $t_1$  disjoint repair sets, each set of size at most  $r_1$ , such that any repair set contains only one parity symbol. Then, the minimum distance  $d$  of  $C$  is bounded by

$$d \leq n - k - \left\lceil \frac{kt_1}{r_1} \right\rceil + t_1 + 1. \quad (4)$$

Additionally, we note that the bound (4) implies

$$\frac{k}{n} \leq \frac{r_1}{r_1 + t_1}. \quad (5)$$

Now, consider a Combination  $(k, 2)$  code and puncture the code using Algorithm 1. For a positive integer  $\tau$  such that  $2|k\tau$ , when the puncturing length  $p = \frac{k}{2}(k - \tau - 1)$ , we obtain an  $[n = k + \frac{k\tau}{2}, k, d]_2$  punctured Combination  $(k, 2)$  code. Let  $C$  be the punctured code. Then, it is easy to see that  $C$  has the  $(2, \tau)_a$ -availability and minimum distance  $d = \tau + 1$ , and thus,  $C$  is optimal in terms of the bounds (4) and (5). In particular, for  $\tau = k - 1$ , the punctured code becomes a Complete graph code [4] and, for  $\tau = k - \frac{k}{\gamma}$  where  $\gamma$  is a positive integer such that  $2 \leq \gamma \leq k$  and  $\gamma|k$ , it becomes a Complete multipartite graph code [4]. Moreover, for  $\tau = 1$ , we have optimal binary LRCs with the code rate  $2/3$  and, for  $\tau = 2$ , we have optimal binary LRCs with the code rate  $1/2$ .

#### 3. Some Well-Known Upper Bounds and Optimality of the Proposed Codes

In this section, we review two bounds in [21] which take into account the field size, locality, and availability. We also review a code rate bound in [24] without the field size constraint. With respect to these bounds, we check the optimality of the proposed Combination codes and punctured Combination codes by the subblock-puncturing with certain choices of parameters.

The authors in [21] proved that, for an  $(n, k, d)_q$  code  $C$  which has joint information availability  $\{(r_l, t_l) : l \in [d-1]\}$ , the code dimension  $k$  satisfies

$$k \leq \min_{\substack{z \in \mathbb{Z}^+ \\ I = \{l_j \in [d-1], 1 \leq j \leq z\} \\ y = \{y_j \in [t_{l_j}], 1 \leq j \leq z\} \\ A(l, y) < k}} [A(l, y) + k_{opt}^{(q)}(n - B(l, y), d)], \quad (6)$$

and the minimum distance  $d$  satisfies

$$d \leq \min_{\substack{z \in \mathbb{Z}^+ \\ I = \{l_j \in [d-1], 1 \leq j \leq z\} \\ y = \{y_j \in [t_{l_j}], 1 \leq j \leq z\} \\ A(l, y) < k}} [d_{opt}^{(q)}(n - B(l, y), k - A(l, y))], \quad (7)$$

**Table 1** Optimality of combination codes and punctured combination codes with regard to well-known alphabet-dependent upper bounds.

$[n, k, d]_2$	$w$	$p$	$(r_1, r_2)_i$	$(t_1, t_2)_i$	$(r_1, r_2)_a$	$(t_1, t_2)_a$	$\Delta_k$	$\Delta_d$
[6, 3, 3]	2	0	(2, 3)	(2, 1)	(2, 3)	(2, 1)	0	0
[7, 3, 4]	3	0	(2, 3)	(3, 1)	(2, 3)	(3, 1)	0	0
[8, 4, 3]	2	2	(2, 3)	(2, 1)	(2, 4)	(1, 1)	0	0
[9, 4, 3]	2	1	(2, 3)	(2, 2)	(2, 3)	(2, 1)	0	1
[10, 4, 4]	2	0	(2, 3)	(3, 2)	(2, 3)	(3, 1)	0	0
[11, 4, 4]	3	3	(2, 3)	(3, 2)	(2, 3)	(3, 2)	1	1
[12, 4, 5]	3	2	(2, 3)	(4, 2)	(2, 3)	(4, 2)	0	1
[13, 4, 6]	3	1	(2, 3)	(5, 3)	(2, 3)	(5, 2)	0	0
[14, 4, 7]	3	0	(2, 3)	(6, 3)	(2, 3)	(6, 3)	0	0
[15, 4, 8]	4	0	(2, 3)	(7, 3)	(2, 3)	(7, 3)	0	0
[10, 5, 3]	2	5	(2, 3)	(2, 1)	(2, 4)	(1, 1)	0	0
[11, 5, 3]	2	4	(2, 3)	(2, 1)	(2, 4)	(1, 1)	1	1
[12, 5, 3]	2	3	(2, 3)	(2, 1)	(2, 4)	(1, 1)	1	1
[13, 5, 4]	2	2	(2, 3)	(3, 2)	(2, 4)	(2, 1)	1	1
[14, 5, 4]	2	1	(2, 3)	(3, 3)	(2, 3)	(3, 1)	2	2
[15, 5, 5]	2	0	(2, 3)	(4, 3)	(2, 3)	(4, 1)	1	1
[16, 5, 5]	3	9	(2, 3)	(4, 3)	(2, 4)	(3, 1)	1	2
[17, 5, 6]	3	8	(2, 3)	(5, 3)	(2, 3)	(3, 1)	1	2
[18, 5, 6]	3	7	(2, 3)	(5, 3)	(2, 3)	(3, 2)	2	2
[19, 5, 7]	3	6	(2, 3)	(6, 4)	(2, 3)	(4, 2)	1	1
[20, 5, 8]	3	5	(2, 3)	(7, 4)	(2, 3)	(5, 2)	1	1
[21, 5, 8]	3	4	(2, 3)	(7, 4)	(2, 3)	(6, 3)	2	2
[22, 5, 9]	3	3	(2, 3)	(8, 4)	(2, 3)	(6, 3)	1	1
[23, 5, 9]	3	2	(2, 3)	(8, 4)	(2, 3)	(7, 4)	1	2
[24, 5, 10]	3	1	(2, 3)	(9, 5)	(2, 3)	(8, 4)	1	2
[25, 5, 11]	3	0	(2, 3)	(10, 6)	(2, 3)	(9, 5)	1	1
[26, 5, 11]	4	4	(2, 3)	(10, 6)	(2, 3)	(10, 5)	1	1
[27, 5, 12]	4	3	(2, 3)	(11, 6)	(2, 3)	(11, 5)	1	1
[28, 5, 13]	4	2	(2, 3)	(12, 6)	(2, 3)	(12, 6)	0	1
[29, 5, 14]	4	1	(2, 3)	(13, 6)	(2, 3)	(13, 6)	0	0
[30, 5, 15]	4	0	(2, 3)	(14, 7)	(2, 3)	(14, 7)	0	0
[31, 5, 16]	5	0	(2, 3)	(15, 7)	(2, 3)	(15, 7)	0	0

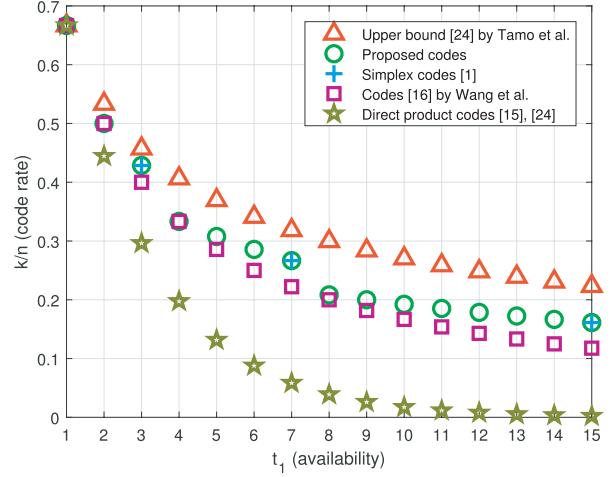
where

$$A(\mathbf{l}, \mathbf{y}) = \sum_{j=1}^z ((r_{l_j} - 1)y_j + 1), \quad B(\mathbf{l}, \mathbf{y}) = \sum_{j=1}^z (r_{l_j} y_j + l_j).$$

We note that  $k_{opt}^{(q)}(n', d')$  is the largest possible dimension of a  $q$ -ary linear code of length  $n'$  and minimum distance  $d'$ , and  $d_{opt}^{(q)}(n', k')$  is the largest possible minimum distance of a  $q$ -ary linear code of length  $n'$  and dimension  $k'$ .

In Table 1, we demonstrate the optimality of the proposed codes with some parameters in terms of the above bounds (6) and (7). To obtain the locality and availability properties of punctured Combination codes, we carried out a computer simulation. We also use the online table [25] for  $k_{opt}^{(2)}(\cdot, \cdot)$  and  $d_{opt}^{(2)}(\cdot, \cdot)$  in the bounds. The  $\Delta k$  represents the difference between the code dimension of the proposed code and the optimal value in terms of the bound (6). The  $\Delta d$  represents the difference between the minimum distance of the proposed code and the optimal value in terms of the bound (7). We note that the codes with parameters  $[7, 3, 4]_2$ ,  $[15, 4, 8]_2$ , and  $[31, 5, 16]_2$  are binary Simplex codes.

In [24], for  $[n, k, d]_q$  code with  $(r_1, t_1)_a$ -availability, an



**Fig. 3** Comparison of the code rates for  $r_1 = 2, 1 \leq t_1 \leq 15$ : For  $t_1 \geq 2$ , all the proposed codes have joint locality  $(r_1, r_2)_a = (2, 3)$ .

upper bound on the code rate is given by

$$\frac{k}{n} \leq \frac{1}{\prod_{j=1}^{t_1} (1 + \frac{1}{j \cdot r_1})}. \quad (8)$$

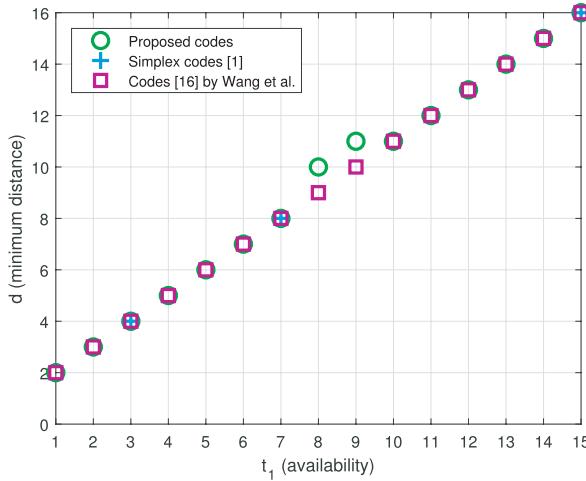
Unfortunately, for  $t_1 \geq 2$ , it is not known whether the bound (8) is achievable. A construction of binary linear codes achieving any given  $(r_1, t_1)$ -availability was introduced in [16]. To the best of our knowledge, in terms of the code rate, this is the best known construction of codes achieving arbitrary  $(r_1, t_1)$ -availability. Since  $r_1 \geq 2$  except for the repetition codes, we are more interested in  $(2, t_1)$ -availability case. There are also some existing codes achieving this availability: the codes from [16], Direct product codes [14], [24], and Simplex codes [1] (only for those values of  $t_1 = 2^{k-1} - 1$ ).

In Fig. 3, we compare the code rates of the above mentioned codes and the proposed codes, both Combination codes and punctured Combination codes. Figure 3 shows the code rates of various codes achieving  $(2, t_1)$ -availability versus the value  $t_1$  from 1 to 15, together with the bound (8). From the figure, we confirm that the proposed codes have higher code rates than the codes from [16] (best known codes) as well as Direct product codes [14], [24] when the same  $(2, t_1)$ -availability is maintained. In Fig. 4 and Fig. 5, we also compare the minimum distances and code lengths of the three families of the codes, the proposed codes, Simplex codes, and codes in [16], respectively. From these results, we confirmed that the proposed codes are attractive not only for code rate but also for other code parameters such as the minimum distance and code length.

We also compare our codes to some of previously proposed LRCs in Table 2. The table summarizes parameters of existing  $[n, k, d]_q$  codes and locality  $r_1$  and availability  $t_1$  properties of them. We restrict our attention to binary LRCs ( $q = 2$ ) with locality  $r_1 = 2$  because of their low encoding, decoding, and repair complexity. All of our codes have such property and their parameters are presented in Table 1 and Figs. 3–5. We can check that almost existing LRCs have the

**Table 2** Parameters of existing LRCs and properties of them

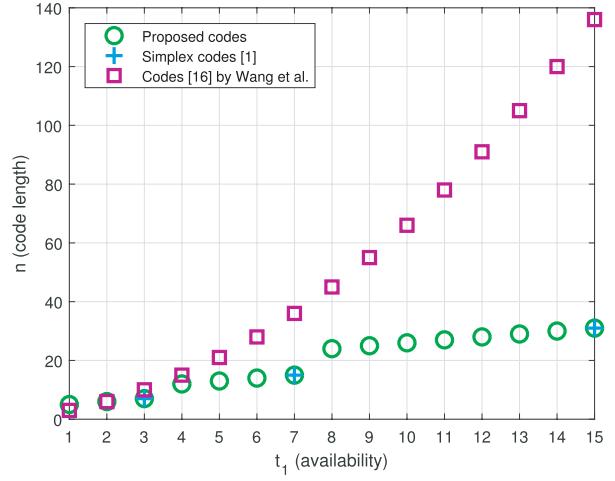
$q$	$n$	$k$	$d$	$r_1$	$t_1$	$R = k/n$	Ref.
$q$ (prime power) $\geq n$	$r+1 n$	$r k$	$n-k-\frac{k}{r}+2$	$(r_1)_a=r$	1	$\leq \frac{r}{r+1}$	[11]
$q$ (prime power) $\geq 2$	$N+t\frac{k}{r}$	$r k$	$n-k-\frac{kt}{r}+t+1$	$(r_1)_t=r$	$t$	$\leq \frac{r}{r+t}$	[13]
$q$ (prime power) $\geq 2$	$N+\frac{N}{r}+(t+1)\frac{k}{r}$	$r k$	$n-k-\frac{kt}{r}+t+1$	$(r_1)_a=r$	$t$	$\leq \frac{r}{r+t}$	[13]
2	$2^{m-1}l+m-1$	$(2^{m-1}-1)l$	3	$(r_1)_i=2^{m-1}-1$	1	$\leq \frac{r}{r+1}$	[15]
2	$(2^m-1-m)l+m+1$	$(2^m-2-m)l$	4	$(r_1)_i=2^m-2-m$	1	$\leq \frac{r}{r+1}$	[15]
2	$(2^m-m)l+m+1$	$(2^m-1-m)l$	4	$(r_1)_i=2^m-1-m$	1	$\leq \frac{r}{r+1}$	[15]
2	$8l+12$ ( $l \leq 15$ )	$7l$	5	$(r_1)_i=7$	1	$\leq \frac{r}{r+1}$	[15]
2	$22l+16$ ( $l \leq 63$ )	$21l$	5	$(r_1)_i=21$	1	$\leq \frac{r}{r+1}$	[15]
2	$(2^m-1-m)l$	$(2^m-2-m)l-1-m$	4	$(r_1)_a=2^m-2-m$	1	$\leq \frac{r}{r+1}$	[15]
2	$(2^m-m)l$	$(2^m-1-m)l-1-m$	4	$(r_1)_a=2^m-1-m$	1	$\leq \frac{r}{r+1}$	[15]
2	$(2^m-1)l$	$(2^m-2)l-m\sum_{i=2}^{\mu}(\lceil\frac{\mu}{i-1}-1\rceil)$	$2\mu$	$(r_1)_a=2^m-2$	1	$\leq \frac{r}{r+1}$	[15]
2	$2^ml$	$(2^m-1)l-m\sum_{i=2}^{\mu}(\lceil\frac{\mu}{i-1}-1\rceil)$	$2\mu$	$(r_1)_a=2^m-2$	1	$\leq \frac{r}{r+1}$	[15]
2	$2^{ms}-1$	$\frac{(2^m-2)(2^{ms}-1)}{2^{m-1}}-(s+1)m$	6	$(r_1)_a=2^m-2$	1	$\leq \frac{r}{r+1}$	[15]
2	$(2^{2m}+1)(2^m-1)$	$(2^{2m}+1)(2^m-2)-6m$	8	$(r_1)_a=2^m-2$	1	$\leq \frac{r}{r+1}$	[15]
2	$(2^{ms}-1)(2^m-1)$	$(2^{ms}-1)(2^m-2)-ms\sum_{i=2}^{\mu}(\lceil\frac{\mu}{i-1}-1\rceil)$	$2\mu$	$(r_1)_a=2^m-2$	1	$\leq \frac{r}{r+1}$	[15]
2	$2^{2s}+2^s+1$	$2^{2s}+2^s-3^s$	$2^s+2$	$(r_1)_a=2^s$	$2^s+1$	$\frac{2^{2s}+2^s-3^s}{2^{2s}+2^s+1}$	[17]
2	$2^{2s}-1$	$2^{2s}-3^s$	$2^s+2$	$(r_1)_a=2^s-1$	$2^s+1$	$\frac{2^{2s}-3^s}{2^{2s}+1}$	[17]
$p$ (prime)	$p^{em}-1$	$p^{e(m-1)}-1$	$\geq \max(p^e+1, em+1)$	$(r_1)_a=m-1$	$em$	$\frac{p^{e(m-1)}-1}{p^{em-1}}$	[18]
$p$ (prime)	$\frac{p^{3e}-1}{p^{e}-1}$	$p^{2e}+p^e+\binom{p+1}{2}^e$	$\geq p^e+2$	$(r_1)_a=p^e$	$p^e+1$	$\frac{p^{2e}+p^e-\binom{p+1}{2}^e}{p^{2e}+p^e+1}$	[18]
2	$\frac{1}{6}(2^m-1)(2^m-2)$	$n-(2^m-1-m)$	4	$(r_1)_a=2^{m-1}-2$	3	$1-\frac{6(2^m-1-m)}{(2^m-1)(2^m-2)}$	[18]
2	$\frac{(r+1)(r+2)m}{2}$	$(r+1)m$	$\geq 3$	$(r_1)_a=r$	2	$\frac{r}{r+2}$	[19]
2	$7m$	$3m$	4	$(r_1)_a=2$	3	$\frac{r}{7}$	[19]

**Fig. 4** Comparison of the minimum distances for  $r_1 = 2, 1 \leq t_1 \leq 15$ : For  $t_1 \geq 2$ , all the proposed codes have joint locality  $(r_1, r_2)_a = (2, 3)$ .

code rate  $R$  less than or equal to  $\frac{r_1}{r_1+r_1}$ , that is, also less than ours, except the first type of codes in [17], three types of codes in [18], and the second type of codes in [19]. However, for the case of  $q = 2$  and  $r_1 = 2$ , the first construction in [17], the second and the third constructions in [18], and the second construction in [19] provide only a  $[7, 3, 4]_2$  Simplex code. Compared to the first construction in [18], our construction provides codes having higher code rate with some parameters. For example, as an LRC with  $(r_1 = 2, t_1 = 9)$ -availability, our construction provides a  $[25, 5, 11]_2$  code in Table 1 and its code rate is  $\frac{5}{25}$  while the construction in [18] provides a code having the code rate  $\frac{63}{511}$ .

#### 4. Concluding Remarks

In this paper, we investigated block-punctured binary Simplex codes, named Combination  $(k, w)$  codes, with good

**Fig. 5** Comparison of the code lengths for  $r_1 = 2, 1 \leq t_1 \leq 15$ : For  $t_1 \geq 2$ , all the proposed codes have joint locality  $(r_1, r_2)_a = (2, 3)$ .

locality and availability properties. The proposed codes are simply obtained by a process called block-puncturing that punctures all the columns of weights from  $k$  down to  $w+1$ , in the generator matrix of a binary Simplex code of dimension  $k$ . The minimum distance, joint locality  $(r_1, r_2)_a$ , joint information locality, availability, and joint information availability of the proposed codes are determined in closed-form expressions. Moreover, we suggested another puncturing method, called subblock puncturing, that punctures few more columns of weight  $w$  from the Combination  $(k, w)$  code. As we have expected, the punctured Combination codes also have good locality and availability properties. Both of Combination codes and punctured Combination codes with certain choices of parameters attain well-known upper bounds on the code dimension and minimum distance which take into account the field size, locality, and availability. Moreover, our codes have the best code rate among the existing

LRCs with the same availability.

## References

- [1] W.C. Huffman and V. Pless, *Fundamentals of Error Correcting Codes*, Cambridge University Press, 2003.
- [2] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, “On the locality of codeword symbols,” *IEEE Trans. Inf. Theory*, vol.58, no.11, pp.6925–6934, Nov. 2012.
- [3] A.S. Rawat, A. Mazumdar, and S. Vishwanath, “Cooperative local repair in distributed storage,” *EURASIP J. Adv. Signal Process.*, vol.2015, no.107, pp.1–17, Dec. 2015.
- [4] J.-H. Kim, M.-Y. Nam, and H.-Y. Song, “Binary locally repairable codes from complete multipartite graphs,” *Proc. Int. Conf. on ICT Convergence (ICTC)*, pp.1093–1095, Oct. 2015.
- [5] J.-H. Kim, M.-Y. Nam, K.-H. Park, and H.-Y. Song, “Optimal binary locally repairable codes with joint information locality,” *Proc. IEEE Information Theory Workshop (ITW)*, pp.54–58, Oct. 2015.
- [6] J.-H. Kim, M.-Y. Nam, and H.-Y. Song, “Alphabet-dependent upper bounds for locally repairable codes with joint locality,” *Proc. Int. Symp. Inf. Theory and Its Applications (ISITA)*, pp.36–40, Oct. 2016.
- [7] D.S. Papailiopoulos and A.G. Dimakis, “Locally repairable codes,” *IEEE Trans. Inf. Theory*, vol.60, no.10, pp.5843–5855, Oct. 2014.
- [8] M. Forbes and S. Yekhanin, “On the locality of codeword symbols in non-linear codes,” *Discrete Math.*, vol.324, pp.78–84, June 2013.
- [9] V. Cadambe and A. Mazumdar, “Bounds on the size of locally recoverable codes,” *IEEE Trans. Inf. Theory*, vol.61, no.11, pp.5787–5794, Nov. 2015.
- [10] C. Huang, M. Chen, and J. Li, “Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems,” *Proc. IEEE Int. Symp. Netw. Comput. Appl.*, pp.79–86, July 2007.
- [11] I. Tamo and A. Barg, “A family of optimal locally recoverable codes,” *IEEE Trans. Inf. Theory*, vol.60, no.8, pp.4661–4676, Aug. 2014.
- [12] I. Tamo, D.S. Papailiopoulos, and A.G. Dimakis, “Optimal locally repairable codes and connections to matroid theory,” *IEEE Trans. Inf. Theory*, vol.62, no.12, pp.6661–6671, Dec. 2016.
- [13] A.S. Rawat, D.S. Papailiopoulos, A.G. Dimakis, and S. Vishwanath, “Locality and availability in distributed storage,” *IEEE Trans. Inf. Theory*, vol.62, no.8, pp.4481–4493, Aug. 2016.
- [14] A. Wang and Z. Zhang, “Repair locality from a combinatorial perspective,” *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, pp.1972–1976, June 2014.
- [15] P. Huang, E. Yaakobi, H. Uchikawa, and P.H. Siegel, “Binary linear locally repairable codes,” *IEEE Trans. Inf. Theory*, vol.62, no.11, pp.6268–6283, Nov. 2016.
- [16] A. Wang, Z. Zhang, and M. Liu, “Achieving arbitrary locality and availability in binary codes,” *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, pp.1866–1870, June 2015.
- [17] L. Shen, F. Fu, and X. Guang, “On the locality and availability of linear codes based on finite geometry,” *IEICE Trans. Fundamentals*, vol.E98-A, no.11, pp.2354–2355, Nov. 2015.
- [18] A. Wang, Z. Zhang, and D. Lin, “Two classes of  $(r, t)$ -locally repairable codes,” *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, pp.445–449, June 2017.
- [19] S. Kadhe and R. Calderbank, “Rate optimal binary linear locally repairable codes with small availability,” *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, pp.166–170, June 2017.
- [20] J.-H. Kim and H.-Y. Song, “Hypergraph-based binary locally repairable codes with availability,” *IEEE Commun. Lett.*, vol.21, no.11, pp.2332–2335, Nov. 2017.
- [21] J.-H. Kim and H.-Y. Song, “Alphabet-dependent bounds for locally repairable codes with joint information availability,” *IEEE Commun. Lett.*, vol.21, no.8, pp.1687–1690, Aug. 2017.
- [22] I. Dumer and O. Kapralova, “Spherically punctured biorthogonal codes,” *IEEE Trans. Inf. Theory*, vol.59, no.9, pp.6010–6017, Sept. 2013.
- [23] S. Lin and D.J. Costello, *Error Control Coding*, Prentice Hall, 2004.
- [24] I. Tamo, A. Barg, and A. Frolov, “Bounds on the parameters of locally recoverable codes,” *IEEE Trans. Inf. Theory*, vol.62, no.6, pp.3070–3083, June 2016.
- [25] M. Grassl, “Code tables: Bounds on the parameters of various types of codes,” [Online]. Available: <http://www.codetables.de/>, 2017.



**Jung-Hyun Kim** received the BS and the MS degree in Electrical and Electronics Engineering from Yonsei University of Seoul, Korea, in 2006 and 2008, respectively. He spent 3 years as a member of Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, from 2010 to 2012, contributed to a team developing a fundamental technology on next-generation wireless mesh communication system. He received the Ph.D. degree from in Electrical and Electronics Engineering from Yonsei University of Seoul, Korea, in 2017. He is currently working at Samsung Electronics, Seoul, Korea. His area of research interest includes artificial intelligence, deep learning, distributed storage systems, coding theory, and applications of wireless communications.



**Min Kyu Song** received his B.S. degree in Electronic Engineering from Konkuk University, Seoul, Korea, and M.S. degree in Electrical and Electronic Engineering from Yonsei University, Seoul, Korea, in 2011 and 2013, respectively. He is currently a Ph.D. candidate working in Channel Coding and Crypto Lab. at Yonsei University. His area of research interest includes PN sequences, cryptography, and coding theory.



**Hong-Yeop Song** received his BS degree in Electronic Engineering from Yonsei University in 1984, MSEE and Ph.D. degrees from the University of Southern California, Los Angeles, California, in 1986 and 1991, respectively. He spent 2 years as a research associate at USC and then 2 years as a senior engineer at the standard team of Qualcomm Inc., San Diego, California. Since Sept. 1995, he has been with Dept. of electrical and electronic engineering, Yonsei University, Seoul, Korea. He had been serving IEEE IT society Seoul Chapter as a chair from 2009 to 2016, and served as a general co-chair of IEEE ITW 2015 in Jeju, Korea. He was awarded the 2017 Special Contribution Award from Korean Mathematical Society for his contribution to the global wide-spread of the fact that Choi (1646-1715) from Korea had discovered a pair of orthogonal Latin squares of order 9 much earlier than Euler. His area of research interest includes digital communications and channel coding, design and analysis of various pseudo-random sequences for communications and cryptography. He is a member of IEEE, MAA (Mathematical Association of America) and domestic societies KICS, IEIE, KIISC and KMS.