New Regenerating Codes over Binary Cyclic Codes

Hanxu Hou[§‡], Yunghsiang S. Han[§], Patrick P. C. Lee[‡], and Qingfeng Zhou[§]
[§]School of Electrical Engineering & Intelligentization, Dongguan University of Technology
[‡]Department of Computer Science and Engineering, The Chinese University of Hong Kong

***Abstract*— Regenerating code is a class of erasure codes designed for distributed storage systems that can achieve optimal tradeoff between storage and repair bandwidth. Most existing constructions of regenerating codes are based on a finite field with large enough size. Recently, a new construction of regenerating codes over a binary cyclic code was proposed. It was shown that the new construction has lower computational complexity than the construction based on finite fields. This paper generalizes the construction by designing regenerating codes with binary cyclic codes that support more parameters. We show that the proposed coding method can achieve the fundamental tradeoff curve between the storage and repair bandwidth. We also give an example that an existing construction of regenerating codes can be transformed to a regenerating code over a binary cyclic code with less computational complexity. Furthermore, the proposed coding framework has more design space for exact repair constructions of minimum storage regenerating codes.**

I. Introduction

In a distributed storage system, a data file is stored in many connected storage nodes. As the storage nodes are unreliable, the data file should be stored in storage nodes redundantly. Erasure coding is one common method to provide redundancy. In an erasure-coding-based storage system, a data file is encoded and distributed to n storage nodes such that the file can be reconstructed from any $k < n$ of the storage nodes. When a storage node fails, we want to repair the failed node by downloading data from other surviving nodes. In the repair process, the total amount of data transferred in repairing a failed node is called *repair bandwidth*.

It is critical to minimize the repair bandwidth in practice. The repair problem was formulated in [1] and the *regenerating codes* (RGC) were also proposed to achieve the minimum repair bandwidth under a constraint. RGC is associated with parameters n, k, d and α, β, B. Specifically, a data file with B symbols over finite field $\mathbb{F}_q$ is encoded into $n\alpha$ symbols and distributed to n nodes. Each node stores α symbols with the requirement that any k nodes can reconstruct the data file. Whenever there is a failed node, it is replaced by a new node and the failed α symbols are recovered by downloading β symbols from each of d surviving nodes. After each repair, the requirement that any k nodes can reconstruct the data file must hold. There are two repair modes. The first one is called exact repair and the second one is functional repair. In exact repair, the data in the new node is the same as those in the failed node. In functional repair, the data in the new node may contain different data from that in the failed one, as long as any k nodes can reconstruct the original file. It is shown in [1] that there is a fundamental tradeoff between the storage capacity α and repair bandwidth $d\beta$ as

$$B \leq \sum_{i=1}^{k} \min\{(d-i+1)\beta, \alpha\}. \tag{1}$$

For exact repair, some recent results on the fundamental limit on repair bandwidth can be found in [2].

In the optimal tradeoff in (1), we have two extreme points. The first one is termed the minimum storage regeneration (MSR) point that corresponds to

$$\alpha = \frac{B}{k}, \beta = \frac{B}{k(d-k+1)}. \tag{2}$$

The second one is termed the minimum bandwidth regeneration (MBR) point that corresponds to

$$\alpha = \frac{2dB}{k(2d-k+1)}, \beta = \frac{2B}{k(2d-k+1)}. \tag{3}$$

Most existing exact repair constructions of RGC focus on the two extreme points over a finite field: MSR codes [3], [4], [5], [6] and MBR codes [3]. The paper [3] presented an exact repair product-matrix construction for MSR codes that can support the parameters $2k-2 \leq d \leq n-1$, and for MBR codes for all the values of parameters $k \leq d \leq n-1$. The constructions of exact repair MSR codes [4], [5], [6] with high coding rate ($k/n > 0.5$) are based on vector-linear codes, and the obtained MSR codes have exponential-level sub-packetization.

The work [7] proposed a construction of functional repair RGC that can achieve all the optimal points on the fundamental tradeoff curve with the underlying finite field being sufficiently large. However, multiplication and division in large finite fields are costly to implement in practice. Recently, BASIC RGC was proposed by replacing the underlying finite fields by binary cyclic codes such that only XOR operations and bit-wise cyclic-shifts are involved in the coding and repair processes [8], [9] with limited parameters. Another class of RGC using XOR and bit-wise shifts were given in [10]. In this paper, we introduce a new class of RGC over binary cyclic codes. We first show that the proposed new RGC over binary cyclic codes can achieve the optimal trade-off curve between storage and repair bandwidth, and then demonstrate that the

This work was partially supported by the National Natural Science Foundation of China (No. 61701115, 61671007, 61871136, 61471156), Start Fund of Dongguan University of Technology (No. GB200902-19, KCYXM2017025, G200906-49), Research Grants Council of Hong Kong (GRF 14216316 and CRF C7036-15G), Provincial-level major scientific research projects in Guangdong Province (No. 2017KZDXM028) and the Science and Technology Planning Project of Guangdong Province (No. 2016B010108002).

existing exact repair construction of RGC can be converted into the RGC over binary cyclic codes. BASIC RGC [8], [9] can be viewed as a special case of the proposed codes.

II. Introduction to Binary Cyclic Codes

In this section, we introduce some known results of binary cyclic codes. Let $\mathbb{F}_2[z]$ denote the set of polynomials in the indeterminate z with coefficients in a binary field $\mathbb{F}_2$. Let $\mathbb{F}_2[z]/(1+z^m)$ denote the quotient ring with addition and multiplication being performed with modulo $1+z^m$. The code generated by a divisor of $1+z^m$ is called *binary cyclic code*. If m is an even number, the binary cyclic code is also called *binary repeated-root cyclic code* [11]. In this paper, we consider that m is an integer except a power of 2.

Let $m=p\tau$, where p is an odd number and τ is a power of a positive integer. An element in the ring $\mathbb{F}_2[z]/(1+z^{p\tau})$ can be represented by a polynomial $a(z)=a_0+a_1z+\ldots+a_{p\tau-1}z^{p\tau-1}$ with coefficients from binary field $\mathbb{F}_2$. In $\mathbb{F}_2[z]/(1+z^{p\tau})$, multiplication by z can be implemented as a cyclic shift, as we have

$$za(z) \bmod (1+z^{p\tau}) = a_{p\tau-1}+a_0z+a_1z^2+\ldots+a_{p\tau-2}z^{p\tau-1}.$$

Consider a sub-ring $C_{p\tau}$ of $\mathbb{F}_2[z]/(1+z^{p\tau})$ of which the polynomial in $C_{p\tau}$ is a multiple of $1+z^\tau$,

$$C_{p\tau}=\{a(z)(1+z^\tau) \bmod (1+z^{p\tau}) | a(z)\in \mathbb{F}_2[z]/(1+z^{p\tau})\}.$$

A sufficient and necessary condition of a polynomial in $C_{p\tau}$ is shown in [12, Theorem 1]. We summarize the result as follows.

Theorem 1. *[12, Theorem 1] A polynomial* $b(x)=\sum_{i=0}^{p\tau-1} b_iz^i \in \mathbb{F}_2[z]/(1+z^{p\tau})$ *is in* $C_{p\tau}$ *if and only if*

$$b_{(p-1)\tau+j}=\sum_{\ell=0}^{p-2} b_{\ell\tau+j}, \tag{4}$$

where $j=0,1,\ldots,\tau-1$.

For example, when $p=3,\tau=2$, a polynomial

$$b(z)=b_0+b_1z+b_2z^2+b_3z^3+b_4z^4+b_5z^5$$

is in $C_{2\cdot 3}$ if and only if

$$b_4=b_0+b_2, b_5=b_1+b_3,$$

according to Theorem 1. It is shown in [12, Lemma 3] that the ring $C_{p\tau}$ is isomorphic to $\mathbb{F}_2[z]/(h(z))$, where

$$h(z)=1+z^\tau+\cdots+z^{(p-1)\tau}.$$

The isomorphism

$$\theta: C_{p\tau}\to \mathbb{F}_2[z]/(h(z))$$

is defined as

$$\theta(f(z))=f(z) \bmod h(z),$$

and the inverse function $\phi(b(z))$ is given by

$$\phi(b(z))=b(z)\cdot(1+h(z)) \bmod 1+z^{p\tau}.$$

Note that the ring $C_{p\tau}$ is a sub-ring of $\mathbb{F}_2[z]/(1+z^{p\tau})$ and is a binary cyclic code. It can be seen that the multiplication identity in $C_{p\tau}$ is $1+h(z)$. The RGC constructed in this paper are indeed codes over the ring $C_{p\tau}$. BASIC RGC [8], [9] can be viewed as RGC over $C_{p\tau}$ with $\tau=1$.

We select a binary cyclic code $C_{p\tau}$ of length $m=p\tau$, and treat it as the alphabet set. The coding framework based on $C_{p\tau}$ is stated as follows. A data file is assumed to have $\kappa(p-1)\tau$ bits and divided into κ groups. Each group contains $(p-1)\tau$ bits. For each group, say $b_0,b_1,\ldots,b_{(p-1)\tau-1}$, we append τ bits and the appended bits $b_{(p-1)\tau+j}$ are computed by (4) for $j=0,1,\ldots,\tau-1$. The polynomial $b(x)=\sum_{i=0}^{p\tau-1}b_iz^i$ is in $C_{p\tau}$ by Theorem 1. Therefore, we can represent the data file by κ polynomials in $C_{p\tau}$. A polynomial in $C_{p\tau}$ is called a *symbol*, a *data symbol* or a *coded symbol*. Each symbol carries $(p-1)\tau$ information bits. A (ν,κ) linear code over $C_{p\tau}$ with length ν and dimension κ is defined by a $\kappa\times\nu$ generator matrix $\mathbf{G}$, where $\nu>\kappa$. Each information symbol is in $C_{p\tau}$, and each entry of $\mathbf{G}$ is a polynomial in $\mathbb{F}_2[z]/(1+z^{p\tau})$. The encoding is performed by multiplying a row vector $\mathbf{w}$ of length κ containing κ data symbols, and the generator matrix $\mathbf{G}$. An entry in $\mathbf{wG}$ is called a *coded symbol*. A coded symbol is thus a linear combination of the κ data symbols, with elements from $\mathbb{F}_2[z]/(1+z^{p\tau})$ as the coefficients. Since $C_{p\tau}$ is an ideal in $\mathbb{F}_2[z]/(1+z^{p\tau})$, all coded symbols in each codeword of the code is also in $C_{p\tau}$. We only store the coefficients of degree from 0 to $(p-1)\tau-1$ of a symbol or a polynomial in a storage node, as the last τ coefficients can be computed when it is needed.

A code is said to be *maximum distance separable* (MDS) if we can recover the κ data symbols in $\mathbf{w}$ from any κ out of ν coded symbols. Let $\mathcal{I}=\{i_1,i_2,\ldots,i_\kappa\}$ be the column index set of the generator matrix $\mathbf{G}$, where $i_1,i_2,\ldots,i_\kappa$ are different integers from 1 to ν. Denote $\mathbf{G}_\mathcal{I}$ as the sub-matrix of $\mathbf{G}$ obtained by retaining the columns indexed by $\mathcal{I}$. The code is MDS if and only if all the sub-matrices of $\mathbf{G}$ are invertible over $C_{p\tau}$, i.e., there exists a square matrix $\tilde{\mathbf{G}}_\mathcal{I}$ over $\mathbb{F}_2[z]/(1+z^{p\tau})$ for each $\mathbf{G}_\mathcal{I}$ such that $\mathbf{G}_\mathcal{I}\tilde{\mathbf{G}}_\mathcal{I}$ is the $\kappa\times\kappa$ identity matrix over $C_{p\tau}$.

The MDS condition is given in the next theorem.

Theorem 2. *[12, Theorem 6] Suppose that* $s_1(z)$ *to* $s_\kappa(z)$ *are data symbols, and* $p_1(z)$ *to* $p_\nu(z)$ *are coded symbols with generator matrix* $\mathbf{G}$. *We can compute* $s_1(z)$ *to* $s_\kappa(z)$ *from any* κ *out of* ν *coded symbols if and only if the determinant of any* $\kappa\times\kappa$ *sub-matrix of* $\mathbf{G}$ *is a non-zero polynomial and is relatively prime to* $h(z)$.

Note that $C_{p\tau}$ is isomorphic to a finite field $\mathbb{F}_{2^{(p-1)\tau}}$ if and only if 2 is a primitive element in $\mathbb{Z}_p$ and $\tau=p^i$ for some non-negative integer i [13]. When $\tau=1$ and p is a prime number such that 2 is a primitive element in $\mathbb{Z}_p$, the ring C_p is a finite field.

III. Functional Repair RGC over Binary Cyclic Codes

In the rest of this paper, we show that we can achieve the optimal tradeoff of functional repair RGC over the binary cyclic code $C_{p\tau}$ if the length m is large enough and satisfies a condition.

In the encoding process of RGC over the binary cyclic code, the data file with $B(p-1)\tau$ bits is first divided into B groups, with each group containing $(p-1)\tau$ bits. We append τ bits for each group according to (4), and represent the $(p-1)\tau$ bits and the τ appended bits as a polynomial in $\mathbb{F}_2[z]/(1+z^{p\tau})$. Therefore, each group of $(p-1)\tau$ bits is then encoded to a codeword of $C_{p\tau}$. We let $s_1(z), s_2(z), \ldots, s_B(z) \in C_{p\tau}$ be the resulting codewords. We call these B symbols the *source data symbols*. Then, we generate $n\alpha$ coded symbols over $C_{p\tau}$ by multiplying the B source data symbols and a $B \times n\alpha$ generator matrix. We then store α coded symbols (only the first $(p-1)\tau$ coefficients of each symbol are stored in node) in each node. Each coded symbol is a linear combination of the B source data symbols. The coefficients of the linear combination form the *global encoding vector* of the corresponding coded symbol. The generator matrix is to be chosen to satisfy the (n,k) *recovery property* that any k nodes can recover the data file.

Suppose that node f fails. We replace the failed node by a new node which contacts d surviving nodes. The storage nodes which participate in the repair process are also called the *helpers*. Each of the helper nodes transmits β symbols to the new node, and each of these symbols is computed by a linear combination of the α coded symbols in the node, with local encoding coefficients being chosen from $\mathbb{F}_2[z]/(1+z^{p\tau})$. Upon receiving the $d\beta$ symbols from the d helpers, the new node computes α symbols and stores them. Each of the α symbols stored in the new node is obtained by a linear combination of the received $d\beta$ symbols. Note that the local encoding coefficients are from $\mathbb{F}_2[z]/(1+z^{p\tau})$. We want to show that by choosing the proper local encoding coefficients from $\mathbb{F}_2[z]/(1+z^{p\tau})$, we can always satisfy the (n,k) recovery property after infinite rounds of failure/repair.

Similar to the method in [7], [8], we need the following lemma in the proof of functional repair RGC over $C_{p\tau}$.

Lemma 3 (DeMillo-Lipton-Schwartz-Zippel [14])**.** *Let $\mathbb{F}$ be a finite field and $\mathcal{S}$ be a subset of elements in $\mathbb{F}$. Let f be a non-zero multivariate polynomial in $\mathbb{F}[X_1, X_2, \ldots, X_N]$ of degree e. Then the polynomial f has at most $e|\mathcal{S}|^{N-1}$ roots in $\mathcal{S}^N$.*

Before giving the main result, we need to introduce more notations. For $i = 1, 2, \ldots, k$, let

$$s_i = \min\{(d-i+1)\beta, \alpha\}.$$

For $i = k+1, k+2, \ldots, n$, let $s_i = 0$. Denote $\mathcal{H}$ as the set of vectors of length n whose components are non-negative integers. Let $\mathcal{H}$ be majorized by the vector $\mathbf{s} = (s_1, s_2, \ldots, s_n)$. In other words, if we sort the components of a vector $\mathbf{h} \in \mathbb{Z}_+^n$ in non-increasing order as $h_{[1]} \geq h_{[2]} \geq \cdots \geq h_{[n]}$, then $\mathbf{h}$ is in $\mathcal{H}$ if and only if

$$\sum_{i=1}^{\mu} h_{[i]} \begin{cases} \leq \sum_{i=1}^{\mu} s_i & \text{for } m = 1, 2, \ldots, n-1, \\ = B & \text{for } \mu = n. \end{cases}$$

We refer the readers to [15] for more details on majorization theory.

In [8], the existence of RGC over a binary cyclic code was given by showing that we can choose the local encoding coefficients to be powers of z such that a collection of the determinants of all sub-matrices of the generator matrix are all evaluated to be invertible in C_p. In this paper, the local encoding coefficients are chosen from any non-zero polynomials in $C_{p\tau}$, and the collection of the determinants of all sub-matrices of the generator matrix are evaluated to be invertible in $C_{p\tau}$. The local encoding coefficient has p different selections in [8] while the local encoding coefficient has $2^{p\tau}-1$ different choices in this paper. Consider that τ is a power of 2, we have that

$$h(z) = 1 + z^{\tau} + \cdots + z^{(p-1)\tau} = (1 + z + \cdots + z^{p-1})^{\tau}.$$

Then, a polynomial is invertible in $C_{p\tau}$ is equivalent to that the polynomial is also invertible in C_p. The difference between the proof of the work in [8] and this paper is that the local encoding coefficients in this paper have more choices. We thus have more design space for the construction of RGC and the value of p can be reduced in functional repair RGC.

Theorem 4. *Given the parameters n, k, d, α and β. Let $m = p\tau$ be an even prime, p be an odd number, τ be a power of 2, and $ord_p(2)$ be the multiplicative order of 2 mod p. If 2^m is larger than*

$$\frac{p-1}{ord_p(2)}\tau \cdot B \cdot \max\left\{\binom{n\alpha}{B}, 2|\mathcal{H}|\right\}, \tag{5}$$

where B is defined in (1)*, then there exists a functional repair RGC over the binary cyclic code of length m, which achieves the optimal tradeoff between storage and repair bandwidth.*

Sketch of proof. The proof is similar to the proof in [8]. The main difference is that the local encoding coefficients have more choices in this paper. The local encoding coefficients in each repair process are from $\mathbb{F}_2[z]/(1+z^{p\tau})$ and the (n,k) recovery property is maintained. Note that the polynomial $1 + z + \ldots + z^{p-1}$ has $\frac{p-1}{ord_p(2)}$ factors with degree $ord_p(2)$ and

$$1 + z^{\tau} + \ldots + z^{(p-1)\tau} = (1 + z + \cdots + z^{p-1})^{\tau}$$

has $\frac{p-1}{ord_p(2)}\tau$ factors with degree $ord_p(2)$. In the application of Lemma 3, we take the set $\mathcal{S}$ to be non-zero polynomials in $\mathbb{F}_2[z]/(1+z^{p\tau})$. Each entry of the local encoding coefficients has $2^{p\tau}-1$ choices. □

If we choose the parameter p to be a prime number such that the multiplicative order of 2 mod p is equal to $p-1$,

$1+z+\ldots+z^{p-1}$ is irreducible polynomial, and the value of $ord_p(2)$ is equal to $p-1$. Then the value in (5) becomes

$$\tau \cdot B \cdot \max\left\{\binom{n\alpha}{B}, 2|\mathcal{H}|\right\}.$$

Note that there are infinitely many such prime number p under the Artin's conjecture on primitive roots [16].

IV. Exact Repair Construction

In this section, we show that the existing exact repair construction of RGC can be converted to RGC over the binary cyclic codes, via the product-matrix RGC [3] as an example.

A. Product-Matrix MSR Codes

In the following, we construct the product-matrix MSR codes for $d = 2k-2$. The construction can be extended to $d \geq 2k-2$, as like the construction in [3].

Let $\beta = 1$, as $d = 2k-2$, we have that $B = k(k-1)$ and $\alpha = k-1$ according to (2). The data file consists of B groups, each of $(p-1)\tau$ bits. We generate B data symbols in $C_{p\tau}$ by appending τ bits for each group by (4). Let $s_1(z), s_2(z), \ldots, s_B(z)$ be the B data symbols. Create two $(k-1)\times(k-1)$ symmetric matrices $\mathbf{S}_1, \mathbf{S}_2$ by filling the upper-triangular part of each of $\mathbf{S}_1, \mathbf{S}_2$ by $k(k-1)$ data symbols. We thus obtain the $d \times (k-1)$ data matrix

$$\mathbf{M} = \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \end{bmatrix}.$$

The generator matrix $\mathbf{\Psi}$ can be chosen to be the $n \times d$ Vandermonde matrix or Cauchy matrix. In the following, we choose the generator matrix to be Vandermonde matrix, i.e., the i-th row of the generator matrix $\mathbf{\Psi}$ is defined as

$$\boldsymbol{\psi}_i^T := \begin{bmatrix} 1 & z^{i-1} & z^{2(i-1)} & \ldots & z^{(d-1)(i-1)} \end{bmatrix}, \quad (6)$$

for $i = 1, 2, \ldots, n$. The i-th node stores $\alpha = k-1$ coded symbols in $\boldsymbol{\psi}_i^T \mathbf{M}$.

With the same discussion in [3], we can show that the proposed product-matrix MSR code satisfies the (n,k) recovery property, if the square Vandermonde matrix is invertible in $C_{p\tau}$ or the determinant of the square Vandermonde matrix is relatively prime to $h(z)$ by Theorem 2. The above condition is satisfied if we choose p to be a prime number such that $p \geq n$. The method of repairing a failed node is the same as in [3], except that we are now working over $C_{p\tau}$ instead of over a finite field. Therefore, we can repair one node by downloading one symbol from each of arbitrary $d = 2k-2$ surviving nodes. Please refer to [3] for the detailed repair method.

B. Product-Matrix MBR Codes

Consider the product-matrix construction of MBR codes. Let $\beta = 1$, then we have

$$B = \frac{k(k+1)}{2} + k(d-k)$$

and $\alpha = d$. A data file contains $B(p-1)\tau$ bits is divided into B groups, each with $(p-1)\tau$ bits. We generate B data symbols $s_1(z), s_2(z), \ldots, s_B(z)$ in $C_{p\tau}$ for the data file. We create $n\alpha$ coded symbols by multiplying the $n \times \alpha$ generator matrix $\mathbf{\Psi}$ and the $\alpha \times \alpha$ data matrix $\mathbf{M}$. The data matrix $\mathbf{M}$ is of the form

$$\mathbf{M} := \begin{bmatrix} \mathbf{S} & \mathbf{T} \\ \mathbf{T}^T & \mathbf{0} \end{bmatrix},$$

where the matrix $\mathbf{S}$ is a symmetric $k \times k$ matrix obtained by first filling the upper-triangular part by data symbols $s_j(z)$, for $j = 1, 2, \ldots, k(k+1)/2$, and then obtain the lower-triangular part by reflection along the diagonal. The rectangular matrix $\mathbf{T}$ has size $k \times (d-k)$, and the entries in $\mathbf{T}$ are data symbols $s_j(z)$, $j = k(k+1)/2, \ldots, B$, listed in some fixed but arbitrary order. The matrix $\mathbf{0}$ is a $(d-k) \times (d-k)$ all-zero matrix.

The $n \times \alpha$ generator matrix $\mathbf{\Psi}$ is composed of the $d \times d$ identity matrix $\mathbf{I}_{d\times d}$ and the encoding matrix $\mathbf{\Psi}_{(n-d)\times d}$. The encoding matrix $\mathbf{\Psi}_{(n-d)\times d}$ is an $(n-d) \times d$ Vandermonde matrix, with the i-th row defined in (6) with $i = 1, 2, \ldots, n-d$. For $i = 1, 2, \ldots, n$, node i stores the d symbols in the i-th row of $\mathbf{\Psi M}$. We choose p to be an odd number such that the determinant of any $d \times d$ sub-matrix of the generator matrix $\mathbf{\Psi}$ is relatively prime to $h(z)$, and by Theorem 2, the constructed codes satisfy the (n,k) recovery property.

In the following, we give an example for $n = 7$, $k = 3$, $d = 4$. This example contains all the essential features of product-matrix MBR codes over the binary cyclic codes.

There are $B = 9$ data symbols $s_1(z)$ to $s_9(z)$. The data matrix is

$$\mathbf{M} = \left[\begin{array}{ccc|c} s_1(z) & s_2(z) & s_3(z) & s_7(z) \\ s_2(z) & s_4(z) & s_5(z) & s_8(z) \\ s_3(z) & s_5(z) & s_6(z) & s_9(z) \\ \hline s_7(z) & s_8(z) & s_9(z) & 0 \end{array}\right].$$

The generator matrix $\mathbf{\Psi}$ is composed of the 4×4 identity matrix $\mathbf{I}_{4\times 4}$ and a 3×4 Vandermonde matrix $\mathbf{\Psi}_{3\times 4}$. Specifically, for $i = 1, 2, 3$, the encoding vector of node $4+i$ is

$$\boldsymbol{\psi}_i^T = \begin{bmatrix} 1 & z^{i-1} & z^{2(i-1)} & z^{3(i-1)} \end{bmatrix}.$$

For any four distinct node indices i_1, i_2, i_3 and i_4 between 1 to 7, the determinant of the sub-matrix of the generator matrix $\mathbf{\Psi}$ consisting by rows i_1, i_2 i_3 and i_4 is invertible in $C_{p\tau}$. Thus, the code satisfies the (n,k) recovery property.

For $i = 1, 2, 3, 4$, node i is data node and stores i-th row of matrix $\mathbf{M}$. For $i = 5, 6, 7$, node i is coded node and stores the four coded symbols in the i-th row of $\mathbf{\Psi M}$. Each of the coded symbols can be obtained by right-cyclic-shifting and adding the source packets appropriately.

Suppose that a data collector connects to nodes 5, 6 and 7. We can solve for $s_7(z)$, $s_8(z)$ and $s_9(z)$ from

$$\begin{bmatrix} s_7(z)+s_8(z)+s_9(z) \\ s_7(z)+zs_8(z)+z^2s_9(z) \\ s_7(z)+z^2s_8(z)+z^4s_9(z) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & z & z^2 \\ 1 & z^2 & z^4 \end{bmatrix} \begin{bmatrix} s_7(z) \\ s_8(z) \\ s_9(z) \end{bmatrix}.$$

It can be solved by Theorem 2. Then, $s_1(z)$ to $s_6(z)$ can be decoded from

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & z & z^2 \\ 1 & z^2 & z^4 \end{bmatrix} \begin{bmatrix} s_1(z) & s_2(z) & s_3(z) \\ s_2(z) & s_4(z) & s_5(z) \\ s_3(z) & s_5(z) & s_6(z) \end{bmatrix}.$$

Suppose the node 5 fails and we want to repair it from nodes 1, 2, 6, and 7. The coded symbol sent from helper node i to the newcomer is the multiplication of the symbols in node i and the encoding vector of node 5. If we arrange the symbols received by the newcomer as a column vector, then the column vector can be written as

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & z & z^2 & z^3 \\ 1 & z^2 & z^4 & z^6 \end{bmatrix} \cdot \mathbf{M} \cdot \boldsymbol{\psi}_1.$$

Since the matrix on the left has non-zero determinant, according to Theorem 2, the newcomer can compute $\mathbf{M} \cdot \boldsymbol{\psi}_1$. As $\mathbf{M}$ is symmetric, this is exactly equal to the content of the failed node. The repair of other nodes can be done similarly.

With the same discussion, we can show that the computational complexity of product-matrix RGC over the binary cyclic codes is the same as that of RGC over binary cyclic codes in [9], both codes have less computational complexity than the product-matrix RGC over finite fields in [3]. Similarly, we may convert other existing constructions of exact repair RGC such as [4], [5], [6] to RGC over the binary cyclic codes with less computational complexity.

V. Discussion and Conclusion

In this paper, we propose a coding framework of designing RGC over a binary cyclic code. In the coding framework, only the XOR and bit-wise cyclic shifts are involved in the coding and repair processes. The previous constructions of RGC over binary cyclic codes [8], [9] can be viewed as a special case of the construction in this paper. We show that the fundamental tradeoff curve between storage and repair bandwidth of functional repair RGC can be achieved when the parameter m is large enough. For exact repair, we show that the product-matrix construction of RGC can be converted into the RGC over the binary cyclic codes with less computational complexity. More importantly, the proposed binary cyclic codes have more design space of exact repair constructions of MSR codes, by choosing some special generator matrix and parameters p, τ.

Given k data symbols $s_j(x)$ in $C_{p\tau}$, for $j = 1, 2, \ldots, k$. We can generate $k + r$ coded symbols $p_j(x)$ in $C_{p\tau}$, for $j = 1, 2, \ldots, k + r$ by computing the product

$$[p_1(x), p_2(x), \cdots, p_{k+r}(x)] = [s_1(x), s_2(x), \cdots, s_k(x)] \cdot \mathbf{G}$$

over $\mathbb{F}_2[x]/(1+x^{p\tau})$, where the generator matrix $\mathbf{G}$ is of size $k \times (k + r)$. If $\mathbf{G}$ is composed of the $k \times k$ identity matrix $\mathbf{I}_{k\times k}$ and a $k \times r$ *encoding matrix* $\mathbf{P}_{k\times r}$, then resulting code is systematic. A systematic code is determined by the encoding matrix $\mathbf{P}_{k\times r}$. By choosing some well-designed encoding matrix, we can obtain the systematic code that have optimal repair bandwidth or asymptotically for the k data symbols. For example, when the encoding matrix is

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ x & x^2 & x^4 & \cdots & x^{2^{k-2}} & 1 \\ 1 & x^{2^{k-2}} & x^{2^{k-3}} & \cdots & x^2 & x \end{bmatrix}^T,$$

it is shown in [17] that the corresponding codes with $r = 3$ and $d = k + 1$ have asymptotically optimal repair bandwidth for any data symbol. The detailed construction can be found in [18]. More constructions with general parameters k, r and d that achieve asymptotically optimal repair bandwidth is one of the future work.

Note that the constructions in [18], [12] can only achieve asymptotically optimal repair bandwidth for any data symbol. How to combine the existing constructions of exact repair RGC over finite field and the encoding matrix construction over $C_{p\tau}$ such that all n symbols can achieve (asymptotically) optimal repair bandwidth is an interesting future work. Another interesting future work is the decoding algorithm. When more than two data symbols fail, how to design the decoding algorithm to recover the failed data symbols is also an important and practical problem.

References

[1] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," *IEEE Trans. on Information Theory*, vol. 56, no. 9, pp. 4539–4551, September 2010.

[2] C. Tian, "Rate Region of the (4, 3, 3) Exact-Repair Regenerating Codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, July 2013, pp. 1426–1430.

[3] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction," *IEEE Trans. on Information Theory*, vol. 57, no. 8, pp. 5227–5239, August 2011.

[4] I. Tamo, Z. Wang, and J. Bruck, "Zigzag Codes: MDS Array Codes with Optimal Rebuilding," *IEEE Transactions on Information Theory*, vol. 59, no. 3, pp. 1597–1616, 2013.

[5] M. Ye and A. Barg, "Explicit Constructions of High-Rate MDS Array Codes With Optimal Repair Bandwidth," *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 2001–2014, 2016.

[6] J. Li, X. Tang, and C. Tian, "A Generic Transformation to Enable Optimal Repair in MDS Codes for Distributed Storage Systems," *accepted in IEEE Trans. on Information Theory*, vol. PP, no. 99, 2018.

[7] Y. Wu, "Existence and Construction of Capacity-Achieving Network Codes for Distributed Storage," *IEEE J. Selected Areas in Communications*, vol. 28, no. 2, pp. 277–288, February 2010.

[8] K. W. Shum, H. Hou, M. Chen, and H. Xu, "Regenerating Codes over a Binary Cyclic Code," in *IEEE International Symposium on Information Theory*, 2014, pp. 1046–1050.

[9] H. Hou, K. W. Shum, M. Chen, and H. Li, "BASIC Codes: Low-Complexity Regenerating Codes for Distributed Storage Systems," *IEEE Trans. on Information Theory*, vol. 62, no. 6, pp. 3053–3069, 2016.

[10] ——, "BASIC Regenerating Code: Binary Addition and Shift for Exact Repair," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, July 2013, pp. 1621–1625.

[11] G. Castagnoli, J. L. Massey, P. A. Schoeller, and N. V. Seemann, "On Repeated-Root Cyclic Codes," *IEEE Trans. on Information Theory*, vol. 37, no. 2, pp. 337–342, 1991.

[12] H. Hou, Y. S. Han, P. P. C. Lee, Y. Hu, and H. Li, "A New Design of Binary MDS Array Codes with Asymptotically Weak-Optimal Repair," *arXiv preprint https://arxiv.org/pdf/1802.07891.pdf*, 2018.

[13] T. Itoh, "Characterization for a Family of Infinitely Many Irreducible Equally Spaced Polynomials," *Information Processing Letters*, vol. 37, no. 5, pp. 273–277, 1991.

[14] S. Jukna, *Extremal Combinatorics with Applications in Computer Science*, 2nd ed. Berlin: Springer-Verlag, 2011.

[15] A. W. Marshall and I. Olkin, "Theory of Majorization and its Applications," *Academic, New York*, 1979.

[16] M. R. Murty, "Artin's Conjecture for Primitive Roots," *Math. Intelligencer*, vol. 10, no. 4, pp. 59–67, 1988.

[17] H. Hou, P. P. C. Lee, Y. S. Han, and Y. Hu, "Triple-Fault-Tolerant Binary MDS Array Codes with Asymptotically Optimal Repair," in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 839–843.

[18] H. Hou and Y. S. Han, "A Class of Binary MDS Array Codes with Asymptotically Weak-Optimal Repair," *Science China Information Sciences*, 2018.