Karimi, Negin; Darani, Ahmad Yousefian; Greferath, Marcus

# Correcting adversarial errors with generalized regenerating codes

*Published in:*
Advances in Mathematics of Communications

*DOI:*
10.3934/amc.2022005

Published: 01/02/2024

*Document Version*
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

*Please cite the original version:*
Karimi, N., Darani, A. Y., & Greferath, M. (2024). Correcting adversarial errors with generalized regenerating codes. *Advances in Mathematics of Communications*, *18*(1), 128-140. https://doi.org/10.3934/amc.2022005

Advances in Mathematics of Communications doi:10.3934/amc.2022005

# CORRECTING ADVERSARIAL ERRORS WITH GENERALIZED REGENERATING CODES

Negin Karimi

University of Mohaghegh Ardabili, 5619911367, Ardabil, Iran
Aalto University, 11000 Espoo, Finland

[2]Ahmad Yousefian Darani* and [3]Marcus Greferath

[2]University of Mohaghegh Ardabili, 5619911367, Ardabil, Iran
[3]University College Dublin, Dublin, Republic of Ireland

(Communicated by Philippe Gaborit)

Abstract. Traditional regenerating codes are efficient tools to optimize both storage and repair bandwidth in storing data across a distributed storage system, particularly in comparison to erasure codes and data replication. In traditional regenerating codes, the collection of any $k$ nodes can reconstruct all stored information and is called the reconstruction set, $\aleph_R$. A failed node can be regenerated from any $d$ surviving nodes. These collections of $d$ nodes are called the regeneration sets, $\aleph_H$. The number of reconstruction sets and the number of regeneration sets satisfy $|\aleph_R| = C_n^k$ and $|\aleph_H| = C_{n-1}^d$. In generalized regenerating codes, we will have, $1 \leq |\aleph_R| \leq C_n^k$ and $1 \leq |\aleph_H| \leq C_{n-1}^d$. In this paper, we address the problem of secure generalized regenerating codes and present a coding scheme by focusing on the features of the generalized regenerating codes that protects data in the distributed storage system in presence of an active omniscient adversary. This adversary can maliciously alter the data stored on the nodes under its control and send erroneous outgoing message when contacted for the repair of failed nodes. In our scheme notwithstanding the presence of an adversary in distributed storage system, a data collector can still obtain the original file using a classical minimum distance decoder.

## 1. Introduction

Network coding allows routers to mix information packets. It reduces the number of transfers needed to carry data, and it is indeed a powerful tool for distributing information in networks. It is, however, susceptible to packet transmission errors caused by an adversary. Different types of adversaries have been studied in pioneering papers. The aim of a distributed storage system (DSS) is to store data by using a distributed collection of nodes which may be individually unreliable. The availability problem of stored data in the collection of nodes has been studied by researchers for quite a while. The most trivial way to improve reliability is adding redundancy by applying replication. This way seems simple but has low storage efficiency. The other strategy is to utilize erasure coding which has better storage efficiency. In [1, 6] erasure codes have been used instead of replication.

2020 *Mathematics Subject Classification:* Primary: 58F15, 58F17; Secondary: 53C35.
*Key words and phrases:* Active omniscient adversary, generalized regenerating codes, fractional repetition code, resiliency capacity.

*Corresponding author: Ahmad Youosefian Darani.

Dimakis et al. [5] introduced *regenerating codes* which are defined based on a new technique that is termed the min-cut max-flow to optimize the repair bandwidth. In a nutshell, regenerating codes are efficient at storage and repair bandwidth. The main idea in this scenario is the MDS property i.e. the file can be retrieved by contacting any $k$ nodes and the collection of any $k$ nodes is called a reconstruction set. Upon failure the replacement node connects to $d$ existing nodes out of $n-1$ nodes and downloads $\beta$ symbols from each. The collection of these $d$ nodes is called a regeneration set. The set of reconstruction sets and that of regeneration sets is denoted by $\aleph_R$ and $\aleph_H$ respectively, $|\aleph_R| = C_n^k$ and $|\aleph_H| = C_{n-1}^d$.

In 2017 Jian Xu et al. [17] defined the concept of a generalized regenerating code. In this family of codes $1 \leq |\aleph_R| \leq C_n^k$ and $1 \leq |\aleph_H| \leq C_{n-1}^d$. This means, that accessing only specific sets of $k$ nodes, the original file can be recovered. Considering the DSS in presence of an intruder, the probability of revealing the original file to an intruder may be reduced.

In [17] the system security level has been discussed as the probability of revealing the original data file in the presence of the intruder. Due to the structure of a generalized regenerating code, an eavesdropper can obtain the entire data file only if he or she has obtained the data stored on $k$ nodes in a *reconstruction set* while any other collection of $k$ nodes may not be a reconstruction set. For this reason, generalized regenerating codes might exhibit a better security level.

Note that due to network coding a router can mix the information content before forwarding it to a destination. By design, it is robust to network failures, but if the network contains a malicious node under control of an adversary, then this adversary will not only be able to corrupt the data stored on his control but also send erroneous data in repair or reconstruction processes. Replacing the content of an affected node each time this node is asked in a repair or reconstruction process, will therefore allow a malicious adversary to compromise the entire system.

The problem of securing DSS against adversaries was targeted in some pioneering earlier work. Desmedt [4] introduced the type of adversaries such as passive adversary, active adversary and jamming adversary. The Byzantine adversary was investigated in [9], [10] and [18]. Also, the problem of error correction in networks was investigated in [19] and [3]. Jian et al. [17] considered an intruder model in which an eavesdropper can access some nodes, and based on the generalized regenerating codes, they introduced a general upper bound on what is called secrecy capacity. Silva and Kschischang [16] investigated the problem of securing a network coding communication system against a wiretap adversary. In their scheme, the secure multicast communication is achieved using an MDS code over $\mathbf{F}_q$ in the first step and then designing a linear network code on top of it. Subsequently, Dialiotis et al. [7] studied the problem of maintaining encoded distributed storage system when some nodes contain adversarial errors.

Pawar et al. [12] studied the problem of securing DSS against *the active omniscient adversary*, *the active limited-knowledge adversary*, and *the passive eavesdropper*. They defined the resiliency capacity of a distributed storage system as the maximum amount of information that it can store safely on a distributed storage system in presence of a malicious adversary. They derived general upper bounds on this resiliency capacity. Their scheme consists of an MDS code, and in the second layer, the output of the MDS code is stored by a RSKR-repetition structure. In the scheme presented, the classical minimum distance decoder fails and the data collector cannot be used for a general adversarial scenario.

MDS codes are linear codes which meet the Singleton bound i.e. linear codes with $[n, k, n-k+1]$ over $\mathbf{F}_q$. Regenerating codes and subsequently generalized regenerating codes use MDS codes in their structure. In this work the structure of generalized regenerating codes is a concatenation of an MDS code with a fractional repetition code. Generality, MDS codes form an important class of linear codes which are discussed extensively in the practical applications such as security aspects of the networks. However MDS codes have been used frequently to store data on the DSS but we know the nontrivial MDS codes with parameters $[n, k, d]$ over $\mathbf{F}_q$ there exist whenever $2 \leq k \leq min\{n-2, q-1\}$ and $n \leq q+k-1 \leq 2q-2$. These conditions on the parameters $n$ and $k$ implies some constraints to construction of the nontrivial MDS codes over $\mathbf{F}_q$. On the other hand in a DSS a data collector recovers the original file by contacting any $k$ nodes out of $n$ nodes. This property is called the maximum distance separable (MDS) property which suggests to use this class of codes.

In this work as an application of generalized regenerating codes, we show that an information file will be stored on a DSS using a linear code and then coded symbols will be stored by a *Fractional Repetition* code. Indeed in our scheme we can use a linear code instead of the MDS code. Especially, when the information file is over $\mathbf{F}_q$, and $q$ is a small prime power, this feature can be useful to get rid of the constraints on parameters posed by the use of MDS codes.

In the present paper, we will address the problem of securing distributed storage systems against an active omniscient adversary. Especially, in a dynamic setting, an important security problem is to safeguard the system against a malicious adversary who may be able to compromise a large part of the system.

Regarding the properties of generalized regenerating codes and features of the active omniscient adversary, we introduce a scheme for storing data in a distributed storage system in presence of an adversary, such that a user, which is referred to as a data collector, can obtain the original file using a classical minimum distance decoder. Our code construction consists of the concatenation of a linear code and a *Fractional Repetition* code as illustrated in Table 1. Our scheme is designed as a special case of the the so-called MBR point, where $\alpha = d\beta$.

The rest of this paper will be divided into a few sections. In Section 2 we start with a description of generalized regenerating codes and discuss the distributed storage system and the adversary model. Also, we describe the construction of a design for storing data on the DSS that are used in the design of generalized regenerating codes. Subsequently, in Section 3 we provide the construction of generalized regenerating codes in presence of Calvin (the active omniscient adversary). An upper bound for the resiliency capacity of a DSS in generalized regenerating codes framework in presence of Calvin are introduced in Section 4 and we compare this upper bound with introduced upper bound for the resiliency capacity in [12]. The last section describes an analysis of the structure presented for the generalized regenerating code.

## 2. Generalized regenerating codes

In this section, we focus on distributing the original file based on generalized regenerating codes in a distributed storage system and repair a single failed node. As pointed out earlier, in a DSS, a data collector must be able to reconstruct the information file by downloading data stored in $k$ nodes of $n$ nodes. The collection of these $k$ nodes is termed a *reconstruction set* and this means, we have $|\aleph_R| = C_n^k$.

Under the operation of the regenerating codes, the amount of downloading during the repair process can be reduced such that upon a failure a replacement node connects to *any* $d$ nodes out of $n-1$ nodes. These $d$ nodes are called *regeneration set*, and hence, we have $|\aleph_H| = C_{n-1}^d$. We will consider the *exact repair* case for the replacement node. Jian et al. [17] introduced the generalized reconstruction and generalized regeneration properties and—based on these—they defined what is called a generalized regenerating code.

**Definition 2.1.** Let the number of reconstruction sets, $\aleph_R$ satisfies $1 \leq |\aleph_R| \leq C_n^k$, meaning, there exist at least one reconstruction set consisting of $k$ nodes for recovering the original file by data collector (DC). We will refer to this property as the *generalized reconstruction property*.

**Definition 2.2.** Let the number of regeneration sets, $\aleph_H$ satisfies $1 \leq |\aleph_H| \leq C_{n-1}^d$, meaning, that there exists at least one regeneration set consisting of $d$ nodes out $n-1$ nodes. This is called the *generalized regenerating property*.

Based on 2.1 and 2.2 it can be deduced, that not every group of $k$ nodes can serve as a reconstruction set and not every group containing $d$ nodes can serve as a regeneration set. In (classical) regenerating codes any $k$ nodes out of $n$ nodes can be a reconstruction set and any $d$ nodes out of $n-1$ nodes can be a regeneration set. Using these two properties the generalized regenerating codes are defined in the following way.

**Definition 2.3.** A regenerating code is called *a generalized regenerating code* if it has the generalized reconstruction property and the generalized regenerating property.

**Distributed storage system model:** We consider a DSS as a graph on $n$ vertices (nodes) and a certain number of edges. These nodes include a source node $S$ which contain the source file. This source node is connected to $n$ storage node by directed edges with infinite capacity and each node with $\alpha$ capacity. These storage nodes are individually unreliable and some of them may fail over time. To guarantee reliability when a node failed, a replacement node of the same capacity will join to DSS and connect to a regeneration set including $d$ nodes. We consider the exact repair case that allows the recovery of an exact replica of the lost data. Also, the DSS should allow a data collector to reconstruct the entire message by downloading the data stored in one of the reconstruction sets.

We start with a single source node S. This contains encoded information over $\mathbf{F}_q$. Any storage node $v_i$ is represented by a pair ${}^{\text{in}}v_i$ and ${}^{\text{out}}v_i$ for all $i = 1, 2, \ldots, n$. These two parts of any node are connected by a directed edge with $\alpha$ capacity (capacity equal to the amount of data stored at the $i$-th node). Upon node failure, a replacement node connects to $d$ helper node and downloads $\beta$ symbols from each for a repair process. Due to exact repair ${}^{\text{out}}v_i$ connects to ${}^{\text{in}}v_{n+1}$ replacement node and downloads $\beta$ symbols from ${}^{\text{out}}v_i$ helper node. Under this explanation, the repair bandwidth is $\gamma = d\beta$.

In regenerating codes, a data collector can retrieve original file by connecting any $k$ nodes. Upon a failure of a single node, any $d$ nodes out of $n-1$ nodes (this would allow the choice of up to $C_{n-1}^d$ collectios) can be used as helper nodes to regenerate the failed data. But in generalized regenerating codes not all collection of $k$ nodes can realize the original file. Due to the presented structure to store data on the DSS, only selected choices of $k$ nodes can serve as a reconstruction set. It is

clear that any regenerating code is a generalized regenerating code. The generalized regenerating codes have an advantage compared with regenerating codes that we refer to in the following:

The probability of revealing the original file in presence of an intruder refers to the system security level. In regenerating codes, if the intruder obtains the stored content on any $k$ nodes, the whole data file will leak to the intruder. In a generalized regenerating code-based system, this can only happen, when the intruder obtained the data stored on $k$ nodes in one reconstruction set, which can be collectively used for data reconstruction. It means that the generalized regenerating codes have better security level compared with traditional regenerating codes in the presence of an intruder.

**Adversarial Model:** According to pioneer works different adversaries have been studied literature in cryptography and coding theory. We assume the DSS in presence of an active omniscient adversary named Calvin. Let Calvin be assumed to be omniscient i.e. he knows the content of the source file. Moreover, he has complete knowledge of the storage and repair schemes. He knows the content stored on all nodes in the system. However, he is in control of merely $b$ nodes where $2b < k$. These $b$ nodes can include some of the original nodes and/or some of the replacement nodes. He can maliciously alter the data stored on the nodes under his control, and send the erroneous outgoing message when contacted for repair or reconstruction.

In this paper, we focus on the *dynamic errors* model i.e. an omniscient adversary may replace the content of a node under his control, each time the node is used for a repair or reconstruction process. Also, in the dynamic behavior, an adversary send erroneous data when contacted by a new node in the repair process.

We present a scheme for DSS which provides resilience against this active omniscient adversary using generalized regenerating codes. In our scheme, the original file will first be encoded using a linear code with minimum distance bigger than $2b\alpha$ and then the output of this process is stored on the DSS following a special issue of *Fractional Repetition* code.

## 3. Generalized regenerating codes for the presence of Calvin

In this section, we will discuss the construction of generalized regenerating codes for a DSS that has to cope with the presence of Calvin, active omniscient adversary. We design the code construction at the minimum bandwidth repair (MBR) point that is characterized by the parameters $\alpha = d$ and $\beta = 1$.

Assume that the DSS has $n$ nodes. These $n$ nodes are divided to $\varrho$ groups any group includes $k$ nodes and a group includes $n$ modulo $k$ with $\alpha$ capacity for each node. Any group will be considered as a reconstruction set. The maximum amount of a data that can be stored on the DSS and then delivered reliably to a data collector that contacts a reconstruction set has been called *the resiliency capacity* of a DSS. The pioneer papers [10], [19] and [3] are the first approach for investigation on the reliably store data on the DSS. It shown there that the resiliency capacity of a DSS is $C - 2t$ where $C$ is the capacity of the multicast network in the absence of the adversary and $t$ is the number of edges under control of Calvin with unit capacity.

Let $N$ denote the maximum number of information that can be stored on the DSS in absence of Calvin.

Assume that $R = N - 2b\alpha$ where b denotes the number of nodes under control of Calvin. A generalized regenerating code can be a $[N, R]$ linear code $\phi$ over $\mathbf{F}_q$

with minimum distance $2b\alpha + 1$. These $N$ symbols observed by the data collector form a codeword of the $[N, R]$ MDS code $\phi$. The minimum distance of $\phi$ is

$$d_{min} \leq N - R + 1 = 2b\alpha + 1.$$

In the rest of this paper, assume that $G$ is the generator matrix for the MDS $\phi$ and $V = (v_1, v_2, \ldots, v_R)$ is the information file. The output of multiplying the information file $(v_1, v_2, \ldots, v_R)$ by the generator matrix $G$ will be $(x_1, x_2, \ldots, x_N)$. These $N$ coded symbols are stored on DSS by using the following new *Fractional Repetition* code. Rouayheb and Ramchandran [8] introduced a new design for a DSS which is called *Fractional Repetition Code*. They defined it as follows.

**Definition 3.1.** A *Fractional Repetition Code* $C$ with repetition degree $r$ for an $(n, k, d)$ DSS is a collection of $n$ subsets $v_1, \ldots, v_n$ of a set $\Omega = \{1, 2, \ldots, \theta\}$ and cardinality $d$ each, satisfying the condition that each element of $\Omega$ belongs to exactly $r$ sets in the collection.

In the mentioned paper, the design of *Fractional Repetition Codes* Codes was studied that can achieve the DSS capacity for the MBR point. To store the $N$ coded symbols we will use this design, however we will assume that each element of $\Omega$ belongs to at least $\varrho$ sets in collection. Here we consider a DSS with parameters $(n, k, d)$, $n$ subsets $\xi_1, \ldots, \xi_n$ of a set $\Omega = \{x_1 \ldots, x_N\}$ and cardinality $\alpha$ each, satisfying the condition that each element of $\Omega$ belongs to at least $\varrho$ sets in the collection. Regarding to this situation we can consider the following design to store the encoded information.

As already mentioned $n = \varrho k + \varrho'$ where $0 \leq \varrho' < k$ and $N$ coded symbols as a $N$-tuple $(x_1, \ldots, x_N)$. Any reconstruction set includes $k$ nodes each with $\alpha$ capacity, meaning $N = k\alpha$. In other words, the number of $n$ nodes divided to $\varrho$ groups of $k$ nodes and a further group include of $n$ modulo $k$ nodes.

The first node is filled by coded symbols $x_1, \ldots, x_\alpha$. The second node is filled by $x_{\alpha+1}, \ldots, x_{2\alpha}$. Continuing this way $x_{(k-1)\alpha+1}, \ldots, x_{k\alpha}$ are stored on $k$-th node in the first group. Now perform a single cyclic right shift of components in the $N$-tuple $(x_1, \ldots, x_N)$ for the second group and then, distribute the shifted components on $k$ nodes at second group similar to what we performed for the first group. The following sketch may illustrate the scheme.

$$\begin{array}{cccc} x_{k\alpha} & x_1 & \ldots & x_{\alpha-1} \\ x_\alpha & x_{\alpha+1} & \ldots & x_{2\alpha-1} \\ \vdots & \vdots & \vdots & \vdots \\ x_{(k-1)\alpha} & x_{(k-1)\alpha+1} & \ldots & x_{k\alpha-1} \end{array}$$

Repeating this process and using $\varrho$ times a single right cyclic shift for $\varrho$ groups forms $n\alpha - \varrho'\alpha$ locations will be filled as shown in Table 1. The remaining locations in the last group including $\varrho'$ nodes will be filled randomly by any of the $x_i$s.

Table 1

| | | | | | |
|---|---|---|---|---|---|
| $v_1$ | : | $x_1$ | $x_2$ | $\dots x_\alpha$ | |
| $v_2$ | : | $x_{\alpha+1}$ | $x_{\alpha+2}$ | $\dots x_{2\alpha}$ | |
| | | $\vdots$ | $\vdots$ | $\vdots$ | |
| $v_k$ | : | $x_{(k-1)\alpha+1}$ | $\dots$ | $x_{k\alpha}$ | |
| $v_{k+1}$ | : | $x_{k\alpha}$ | $x_1$ | $\dots$ | $x_{\alpha-1}$ |
| $v_{k+2}$ | : | $x_\alpha$ | $x_{\alpha+1}$ | $\dots$ | $x_{2\alpha-1}$ |
| | | $\vdots$ | $\vdots$ | | |
| $v_{2k}$ | : | $x_{(k-1)\alpha}$ | $x_{(k-1)\alpha+1}$ | $\dots$ | $x_{k\alpha-1}$ |
| | | $\vdots$ | $\vdots$ | $\vdots$ | |

With respect to this type of *Fractional Repetition Code* each of $\varrho$ groups forms a reconstruction set i.e. there exist $\varrho$ reconstruction sets. Therefore they satisfy the generalized reconstruction property $1 \leq |\aleph_R| \leq C_n^k$. Assume that the data collector connects to the $\ell$-th group to recover the original file. Let node $x_{\ell k+j}$ connect by $x_{\ell k+i}$ at $\ell$- group and there be one input edge to each ${}^{\text{in}}x_{\ell k+i}$ and one output edge ${}^{\text{out}}x_{\ell k+j}$. We assume without loss of generality that the symbols sent on ${}^{\text{out}}x_{\ell k+j}$ is the same symbols received by ${}^{\text{in}}x_{\ell k+i}$.

Note, that Calvin controls $b$ nodes and due to our scheme, Calvin can introduce at most $b\alpha$ errors among $N$ symbols on a reconstruction set. Since based on the introduced structure to store information on a DSS any coded symbol $x_i$ can appear at most once in any group, so upon a failure in the $\ell-$th group, the replacement node has to join the $\ell-$th group and connect to other groups except the $\ell$-th group to download the failed symbols.

In the following, we explain the construction of the regeneration and reconstruction process based on our scheme.

**Reconstruction and Regeneration process:** In the following we will discuss the decoding process due to introduced structure for generalized regenerating codes. First, we introduce the description of reconstruction and regeneration process. Without loss of generality assume that a data collector accessing $k$ nodes in the first group as a reconstruction set. Thereupon the data collector will observe a total symbols $k\alpha$ which all of them have distinct indices. Assume that none of nodes are under control of Calvin and have not been failed. It follows that data collector can obtain the oroginal file without errors created by Calvin.

Now assume that the first node in the first group is failed. Under introduced structure for storing data on DSS , any group include $k\alpha$ symbols which have distinct indices. This means corresponding any failure the replacement node can only connect to other groups to download failed symbols i.e. $1 \leq |\aleph_H| < C_{n-1}^d$. Since our codes construction is designed for $\beta = 1$, so the number of errors introduced by Calvin in any group can be at most $b$ nodes equivalently, $b\alpha$ symbols. Irregardless the number of failures and by a simple manipulation

$$t < [\frac{d_{min}(\phi) - 1}{2}] \tag{1}$$

where $t$ denotes the number of errors introduced by Calvin. Whereas $2b < k$, a simple manipulation shows that, the data collector will be able to recover the file by the classical minimum distance decoder when the information file has been encoded by a MDS with minimum distance in range $d = 2b\alpha$.

The following example of generalized regenerating codes in the presence of Calvin may help to understand what we have shown before.

**Example 1.** Assume that $D(n = 16, k = 3, d = 4)$ is a DSS in presence of Calvin such that Calvin controls $b = 1$ node (recall that Calvin can control up to $b$ nodes where $2b < k$). Due to the presented scheme as the generalized regenerating code in the previous section the information file will be encoded by an $[N, R]$ linear code such that $N = k\alpha = 12$ and $R = N - 2b\alpha = 4$ with minimum distance 9. In second layer 12 encoded symbols will be stored on the DSS by the following Table.

| | | | | | |
|---|---|---|---|---|---|
| $v_1$ | : | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
| $v_2$ | : | $x_5$ | $x_6$ | $x_7$ | $x_8$ |
| $v_3$ | : | $x_9$ | $x_{10}$ | $x_{11}$ | $x_{12}$ |
| $v_4$ | : | $x_{12}$ | $x_1$ | $x_2$ | $x_3$ |
| $v_5$ | : | $x_4$ | $x_5$ | $x_6$ | $x_7$ |
| $v_6$ | : | $x_8$ | $x_9$ | $x_{10}$ | $x_{11}$ |
| $v_7$ | : | $x_{11}$ | $x_{12}$ | $x_1$ | $x_2$ |
| $v_8$ | : | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
| $v_9$ | : | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ |
| $v_{10}$ | : | $x_{10}$ | $x_{11}$ | $x_{12}$ | $x_1$ |
| $v_{11}$ | : | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
| $v_{12}$ | : | $x_6$ | $x_7$ | $x_8$ | $x_9$ |
| $v_{13}$ | : | $x_9$ | $x_{10}$ | $x_{11}$ | $x_{12}$ |
| $v_{14}$ | : | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
| $v_{15}$ | : | $x_5$ | $x_6$ | $x_7$ | $x_8$ |
| $v_{16}$ | : | $x_1$ | $x_2$ | $x_3$ | $x_4$ |

Assume that $v_1$ is under control of Calvin. The data collector can connect to one of reconstruction sets $R_1 = \{v_1, v_2, v_3\}$, $R_2 = \{v_4, v_5, v_6\}$, $R_3 = \{v_7, v_8, v_9\}$, $R_4 = \{v_{10}, v_{11}, v_{12}\}$ and $R_5 = \{v_{13}, v_{14}, v_{15}\}$ in order to obtain the information file i.e. $|\aleph_R| = 5$ and generalized reconstruction property is satisfied. Assume that the data collector connects to the second group.

Consider the case, where $v_5$ fails, and replacement node $v_{17}$ joins to the second group. Our scheme prescribes, that the replacement node has to connect to other groups and download the failed symbols $x_4, x_5, x_6$ and $x_7$ which were stored on node $v_5$. To download $x_4$, the replacement node connects to $v_1, v_8, v_{12}$ or $v_{14}$. Obviously, $v_1$ sent an erroneous symbol $x_4$. By assumption $\beta = 1$ and our assumption that any coded symbol is present at most once in any group, Calvin can send at most 4 erroneous symbols in any group. In other words, Calvin can only modify and alter $x_1, x_2, x_3$ and $x_4$ and any group contain only one $x_1, x_2, x_3$ and $x_4$. It is easy to see that by classical minimum distance decoder the original file can be obtained.

## 4. The resiliency capacity and the number of failures

Pawar et al. [12] have defined the concept resiliency capacity of distributed storage systems as the maximum amount of information that can be stored on DSS in presence of an active omniscient adversary. They have derived an upper bound on the resiliency capacity. In this section, we will discuss on the upper bound for resiliency capacity while the information file will be stored on DSS in presence of Calvin and using generalized regenerating codes framework. We will compare the upper bound of the resiliency capacity in [12] and upper bound of the resiliency

capacity in the considered scenario based on the generalized regenerating codes at the MBR point.

This upper bound interpret by *min-cut* technique. As shown by the pioneer works, *min-cut* separates the source node and data collector such that the *min-cut* between the source node and data collector is larger than the initial size of information file.

**Proposition 1.** *Let $D(n,k,d)$ be a DSS such that $n$ nodes are divided to $k$ groups and each group is a reconstruction set. If an active omniscient adversary controls $b \geq 1$ nodes, with $2b < k$, then the resiliency capacity $C_r(\alpha,\gamma)$ is upper bounded as follows:*

$$C_r(\alpha,\gamma) \ \leq \ (n-\omega-d-2b)\alpha + \omega(d-t')\beta, \tag{2}$$

*where $t'$ is the number of nodes under control of Calvin that any newcoming node may contact in the repair process and $\omega$ is the number of failures at the $\ell$-th group.*

*Proof.* Assume that the data collector DC connects to the $\ell$-th group to obtain the original file. According to [12], the redundancy of $2b$ nodes is needed in order to correct the adversarial errors on $b$ nodes. Let $\aleph'$ denote the set of $2b$ nodes considered as redundancy to correct the adversarial errors on $b$ nodes in presence of Calvin. Consider $U^*$ be the set of ${}^{\text{out}}x_t$ in DSS except those of the $\ell$-th group. For any ${}^{\text{out}}x_t \in U^*$, $w_t$ denotes the sum of the capacities of edges from ${}^{\text{out}}x_t$ to replacement nodes in the $\ell$-th group. We consider a cut $C(U,\overline{U})$ defined in the following way:

$$\begin{cases} {}^{\text{out}}x_t \in \overline{U} & w_t = 0 \\ {}^{\text{out}}x_t \in U & w_t \geq \beta \end{cases}$$

This means that ${}^{\text{out}}x_t \in \overline{U}$ whenever ${}^{\text{out}}x_t$ does not connect to any replacement nodes in the $\ell$-th group and otherwise ${}^{\text{out}}x_t \in U$. With respect to these notations, $U$ is defined as:

$$U \ = \ \{{}^{\text{in}}x_i\}_{i\in I} \cup \{{}^{\text{out}}x_t\}_{w_t\geq\beta},$$

where $\{{}^{\text{in}}x_i\}_{i\in I}$ denote the set of all nodes in DSS include of original nodes and replacement nodes except replacement nodes in the $\ell$-th group i.e. $|I| = n-\omega$. Also, $\{{}^{\text{out}}x_t\}_{w_t\geq\beta}$ denote the set of helper nodes outside of the $\ell$-th group. On the other hand, we consider

$$\overline{U} = \{{}^{\text{out}}x_i\}_{k-\omega} \cup \{{}^{\text{in}}x_i - {}^{\text{out}}\ x_i\}_{\omega} \cup \{{}^{\text{out}}x_t\}_{w_t=0}$$

where $\{{}^{\text{out}}x_i\}_{k-\omega}$ denotes the set of original nodes in the $\ell$-th group, $\{{}^{\text{in}}x_i - {}^{\text{out}}x_i\}_{\omega}$ corresponding to replacement nodes in the $\ell$-th group and $\{{}^{\text{out}}x_t\}_{w_t<\beta}$ is the set of nodes outside of the $\ell$-th group which connect to replacement nodes within the $\ell$-th group.

Let $T$ be the set of edges outgoing from ${}^{\text{in}}x_i$ such that ${}^{\text{in}}x_i$'s do not belong $\aleph'$ and edges belonging to the cut $C(U,\overline{U})$ that are incoming to replacement nodes in the $\ell$-th group such that these edges do not come from nodes in $\aleph'$. Hence, the capacity of the DSS is upper bounded by the total capacity of the edges in the set $T$. With respect to the number of failures and $C(U,\overline{U})$, we consider the following conditions:

1. If all of nodes in $\aleph'$ are outside of the $\ell$-th group, then

$$C_r(\alpha,\gamma) \leq (k-\omega)\alpha+(n-k-d-2b)\alpha+\omega(d-t')\beta = (n-\omega-d-2b)\alpha+\omega(d-t')\beta,$$

   where $t'$ is the number of nodes in the set $\aleph'$ that newcomers in the $l$-th group connect to them.
2. If some of nodes in $\aleph'$ are in the $\ell$-th group, then

$$C_r(\alpha,\gamma) \leq (n-k-\xi')\alpha+(k-\omega-\xi)\alpha+\omega(d-t')\beta = (n-\omega-\xi'-\xi)\alpha+\omega(d-t')\beta$$

   where $\xi$ is the number of nodes in $\aleph'$ which have not failed and $\xi'$ is the number of nodes in $\aleph'$ at other groups.
3. If all of nodes in $\aleph'$ are in the $l$-th group, then

$$C_r(\alpha,\gamma) \leq (n-k)\alpha+(k-2b-\omega)\alpha+\omega(d-t')\beta = (n-\xi-\omega)\alpha+\omega d\beta$$

The bound will be obtained by taking the minimum of all above upper bounds, so

$$C_r(\alpha,\gamma) \;\leq\; (n-\omega-d-2b)\alpha+\omega(d-t')\beta.$$

□

Now we compare the upper bound of the resiliency capacity in [12] and introduced upper bound in this work. In our scenario and due to Proposition 1 any group can tolerate $k-2b$ failures. As pointed out in [12], and according to the $MBR$ point with parameters $\alpha = d\beta$ and $\beta = 1$, the upper bound of resiliency capacity in [12] can be written by the following

$$C_r(\alpha,\beta) \leq \sum_{i=2b+1}^{k} min\{(d-i+1)\beta,\alpha\} \tag{3}$$

$$= \sum_{i=2b+1}^{k} min\{(d-i+1),d\}$$

$$= (d-2b)+(d-2b-1)+\ldots+(d-k+1)$$

There should be a

$$\frac{1}{2}(k-2b)(2d-k-2b+1) \tag{4}$$

Any group can tolerate $k-2b$ failures i.e. $\omega \leq k-2b$. Consider $\omega = k-2b$, then at MBR point (2) can be written as

$$(n-k-d)d+(k-2b)(d-t') \tag{5}$$

A simple comparison between (4) and (5) shows that at MBR point the upper bound in (2) is bigger than (3).

## 5. Analysis of structure presented for generalized regenerating codes

Our analysis is based on the structure of the information flow graph $G$ which presents a the structure of DSS. The structure presented for generalized regenerating codes in this work possess some desirable properties which we refer them in the following.

**Express the DSS through an information flow graph:** As pointed out in Section 2, in this work we explain a DSS as an information flow graph which is a directed acyclic graph consisting of a source node S, $n$ storage nodes which are represented by ${}^{\text{in}}x_i, {}^{\text{out}}x_i$ and data collectors. The single source nodes contains the original data and two parts ${}^{\text{in}}x_i$ and ${}^{\text{out}}x_i$ are connected by an edge with capacity equal to $\alpha$. Note that $\alpha$ is the amount of data stored at any storage node. At initial step, source node S connects to input parts $({}^{\text{in}}x_i)$ with directed edges of infinite capacity.

Frequent node failures and upon any failure a replacement node join to DSS. When the replacement node $v_j$ choose to connect with an available node $v_i$, then a directed edge from ${}^{\text{out}}x_i$ to ${}^{\text{in}}x_j$. Regard as introduced FractionalRepetition in Section (3), $n$ storage nodes are divided to $\rho$ groups each group include $k$ nodes. Finally a data collector corresponds to request for retrieving data where the data collector obtain the stored data by connecting to output parts of subsets of available nodes through edges with infinite capacity.

**Encoding Information:** Note we presented our scenario when construction of generalized regenerating codes meet the upper bound on resiliency capacity of a DSS. Albeit our codes construction consist of the concatenation of a linear code with a FR code. Notice that one advantage of utilizing generalized regenerating codes is that we are not only limited to use MDS codes to encoding information file and we have open hand to choose linear codes with more general minimum distance. It can be useful to overcome constraints for nontrivial MDS codes on $\mathbf{F}_q$ when $q$ is a small integer.

**Better system security:** As discussed in Section 1 using generalized regenerating codes a DSS has better security in the presence an intruder. If an intruder obtains the whole data file just when obtained the data stored on $k$ nodes in a reconstruction set. On the other hand in generalized regenerating codes, not every collection of $k$ nodes will be a reconstruction set.

**Minimum distance decoder:** We know that a minimum distance decoder is one of the simplest ways to retrieve the information file through a data collector. In [12], the minimum distance decoder fails. Utilizing generalized regenerating codes, a data collector will be capable to obtain the information file through minimum distance decoder.

**Resiliency capacity upper bound:** Notice in (2), $t'$ can be at most $b$ and according to the MBR point $\alpha = d\beta$ and $\beta = 1$. Also, $\omega$ can be at most $k - 2b$. Under these situations a simple computation show that at MBR point the upper bound in (2) is bigger than (3). This means utilizing generalized regenerating codes the resiliency capacity has a bigger upper bound.

## 6. Conclusion

In this paper, the generalized regenerating codes are proposed for securing distributed storage systems against an active adversary. The structure of a generalized

regenerating code provides the conditions that in spite of the existence of the adversary, data collector and replacement nodes can obtain desired data by reconstruction sets and regeneration sets that have less modified data. Moreover, the structure of generalized regenerating codes presents a bigger upper bound for resilience capacity under the specific conditions compared with the resiliency capacity in [12]. The details are presented in the proposition 1.

In the pioneering work [12], a data collector was not able to catch the original file using a minimum distance decoder when a distributed storage system contains an adversary. The authors had to presents more complicated decoding algorithms while the structure of regenerating codes provides the minimum distance decoder for a data collector. In our work due to the structure of generalized regenerating codes, a data collector can catch information files by minimum distance decoder.

Silberstein et al. [15] have presented a coding scheme for distributed storage systems in presence of an adversarial error. They present a coding scheme and an upper bound on resilience capacity for an adversary that can replace the content of an affected node only once. While in our work we consider dynamic error model i.e. an adversary may replace the content of a node under his control each time the node is used for a repair or reconstruction process.

## ACKNOWLEDGMENTS

We would like to thank you for **following the instructions above** very closely in advance. It will definitely save us lot of time and expedite the process of your paper's publication.

## REFERENCES

[1] R. Bhagwan, K. Tati, Y.-C. Cheng, S. Savage and G. M. Voelker, Total recall: System support for automated availability management, *Proc. of the Symposium on Networked Systems Design and Implementation*, **NSDI** (2004).

[2] N. Cai and R. Yeung, Secure network coding on a wiretap network, *IEEE Trans. Inform. Theory*, **57** (2010), 424–435.

[3] N. Cai and R. W. Yeung, Network error correction, II: Lower bounds, *Commun. Inf. Syst.*, **6** (2006), 37–54.

[4] Y. Desmedt, Unconditionally private and reliable communication in an untrusted network, *Theory and Practice in Information-Theoretic Security, Information Theory Workshop on.*, **IEEE** (2005) 38–41.

[5] A. Dimakis, P. Godfrey, Y. Wu, M. Wainright and K. Ramchandran, Network coding for distributed storage systems, *IEEE Trans. Inform. Theory*, **56** (2007), 4539–4551.

[6] A. G. Dimakis, V. Prabhakaran and K. Ramchandran, Decentralized erasure codes for distributed networked storage, *IEEE Trans. Inform. Theory*, **52** (2006), 2809–2816.

[7] T. Dikaliotis, A. G. Dimakis and T. Ho, Security in distributed storage systems by communicating a logarithmic number of bits, *International Symposium on Information Theory, IEEE*, (2010), 1948–1952.

[8] S. El Rouayheb and K. Ramchandran, Fractional repetition codes for repair in distributed storage systems, *Communication, Control, and Computing (Allerton), 48th Annual Allerton Conference,IEEE*, (2010), 1510–1517.

[9] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros and D. R. Karger, Byzantine modification detection in multicast networks using randomized network coding, *International Symposium on Information Theory*, **ISIT** (2004).

[10] S. Jaggi and M. Langberg, Resilient network codes in the presence of eavesdropping Byzantine adversaries, *Information Theory*, **IEEE** (2007), 541–545.

[11] A. Kshevetskiy and E. Gabidulin, The new construction of rank codes, *Proceedings. International Symposium on Information Theory*, 2005.

[12] S. Pawar, S. Rouayheb and K. Ramchandran, Security dynamic distributed storage systems against eavesdropping and adversarial attacks, *IEEE Trans. Inform. Theory*, **57** (2011), 6734–6753.
[13] K. Rashmi, N. B. Shah, P. V. Kumar and K. Ramchandran, Exact regenerating codes for distributed storage, *Arithmetic of Finite Fields*, **6087** (2010), 215–223.
[14] S. Rouayheb and E. Soljanin, On wiretap networks II, *IEEE International Symposium on Information Theory*, (ISIT) (2007).
[15] N. Silberstein, A. S. Rawat and S. Vishwanath, Error resilience in distributed storage via rank-metric codes, *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE*, 2012.
[16] D. Silva and F. R. Kschischang, Security for wiretap network via rank-metrik codes, *IEEE Internat. Symp. Inform. Th.*, **ISIT** (2004), 616–624.
[17] J. Xu, Y. Cao and D. Wang, Generalised regenerating codes for securing distributed storage systems against eavesdropping, *Journal of Information Security and Applications*, **34** (2017), 225–232.
[18] H. Yao, D. Silva, S. Jaggi and M. Langberg, Network codes resilient to jamming and eavesdropping, *IEEE International Symposium on Network Coding*, 2010.
[19] R. W. Yeung and N. Cai, Network error correction, I: Basic concepts and upper bounds, *Commun. Inf. Syst.*, **6** (2006), 19–35.

Received December 2019; revised October 2021; early access February 2022.

*E-mail address*: negin.karimi@aalto.fi
*E-mail address*: yousefian@uma.ac.ir
*E-mail address*: marcus.greferath@aalto.fi