Partially Collaborative Storage Codes in the Presence of an Eavesdropper

Liu, Shiqiu; Oggier, Frédérique

2015

Liu, S., & Oggier, F. (2016). Partially collaborative storage codes in the presence of an eavesdropper. *International Journal of Information and Coding Theory*, 3(3), 177-196.

<https://hdl.handle.net/10356/103743>

© 2015 Inderscience. This is the author created version of a work that has been peer reviewed and accepted for publication by *International Journal of Information and Coding Theory*, Inderscience. It incorporates referee's comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at:
[<http://www.inderscience.com.ezlibproxy1.ntu.edu.sg/info/inarticle.php?artid=76955>].

Downloaded on 26 Apr 2023 15:54:24 SGT

Partially Collaborative Storage Codes in the Presence of an Eavesdropper

Shiqiu Liu and Frédérique Oggier

September 13, 2015

Abstract

We consider the study of codes designed for distributed storage systems, whose maintenance design criterion is to reduce the repair bandwidth, in a generic setting where t failures, $t \geq 1$, are repaired collaboratively, thus involving the participation of t storage nodes. Yet each of these t nodes is actually actively exchanging data with only a subset formed by $t - s$ of the $t - 1$ other nodes, for $1 \leq s \leq t$, explaining the term “partially collaborative”. To understand the role of the different code parameters, namely the threshold t of failures, the degree s of collaboration, but also the repair bandwidth, we adopt the general scenario where an eavesdropper may spy on some storage nodes, and their communication links. By doing so, we recover previous results on partially collaborative codes, and discuss some aspects of cooperation and security in terms of their influence on the repair bandwidth in the presence of an eavesdropper.

1 Introduction

Suppose that a data object $\mathbf{o}$ of size $\mathcal{M}$ has to be stored over n nodes belonging to an erasure coding based networked distributed storage system. We assume the alphabet to be the finite field $\mathbb{F}_q$ of cardinality q , q a prime power, that is $\mathbf{o} \in \mathbb{F}_q^{\mathcal{M}}$. The data object is encoded into n coefficients $s_1, \dots, s_n$, each to be stored at a different storage node, whose storage capacity is α (that is $s_i \in \mathbb{F}_q^\alpha$, $i = 1, \dots, n$). We assume that the encoding is such that any choice of k nodes allows to retrieve the data object, and we will refer to an (n, k) storage code. The

notation is reminiscent of a linear (n, k) erasure code, and a linear (n, k) storage code where $\alpha = 1$ is indeed a linear (n, k) erasure code, but not otherwise.

Suppose that some of these n nodes are failing. It is needed to reconstruct the lost data at some live nodes, to ensure that the fault tolerance of the system does not decrease. Let t be the number of failures (or threshold) after which a repair process is triggered: t nodes which do not store data from this object download data from (some of) these $n - t$ live nodes, each of the t nodes contacts d live nodes, and downloads from them β amount of data. This is referred to as the *download phase* of the repair. Next comes the *collaboration phase* of the repair process, where some of the t nodes exchange data with each other (see also Figure 1). When each of the t nodes collaborates with the other $t - 1$, we speak of *full collaboration*, which was introduced in [1, 2, 3].

The goal of an (n, k) storage code is thus two-fold: (1) to ensure a good trade-off between fault tolerance and storage overhead, and (2) to optimize the repair process, which may mean, reduce the repair bandwidth γ , minimize the number d of live nodes contacted per repair, or minimize the time taken to complete a repair. In this paper, we will focus on the repair bandwidth γ (see [4] for a survey of codes designed according to the other design criteria.)

Collaboration aims at reducing the repair bandwidth γ , which is given by

$$\gamma = d\beta + (t - 1)\beta',$$

since each node downloads $d\beta$ symbols and exchanges $(t - 1)\beta'$ of them. It was shown in [1, 2, 3] that there is a trade-off between the repair bandwidth γ , and the storage capacity α . Codes reaching this trade-off are called *(fully) collaborative regenerating codes*. Two extreme points have been identified:

- The *minimum storage repair (MSR) point*, where $\alpha = \mathcal{M}/k$. This minimum follows from the definition of k , since any k nodes should contain enough information to obtain $\mathbf{o} \in \mathbb{F}_q^{\mathcal{M}}$.
- The *minimum bandwidth repair (MBR) point*, where $\gamma = \alpha$. Indeed, this is the least amount of data to be transferred during repair, so as not to lose any information about $\mathbf{o}$.

The trade-off between these two points has been computed in [3].

The benefit in repair bandwidth observed with collaborative codes is inherited from network coding inspired techniques. Such codes are then expected to suffer from the security drawbacks observed in network coding: a passive eavesdropper

may listen to the communication between nodes, where information is now transmitted in the form of linear combinations of the packets, or an active adversary may insert bogus data, which is then contaminating other packets. For collaborative storage nodes, these threats become eavesdropping the data transmitted during repair [5, 6, 7, 8], or corrupting it [9], thus endangering the whole maintenance process. Furthermore, the more collaboration, the larger the amount of nodes involved in downloading and/or exchanging data, implying a likely increase in security risks. This suggests that a trade-off between gain in repair bandwidth and loss in security could be achieved. We suggest as one way to study this scenario to relax the extend of collaboration during the exchange phase of repair.

The case when only a subset of the t nodes collaborate among each other has been studied in [10], and some constructions at MSR and MBR points have been proposed in [11]. The amount of collaboration is modeled by introducing a degree s of collaboration, with $s = 1$ for full collaboration, and $s = t$ for no collaboration, in which case the repair bandwidth is given by

$$\gamma = d\beta + (t - s)\beta'.$$

See Figure 1 for an illustration of partial collaboration, where during the exchange phase, each node participating to the repair only contacts one other node.

The contributions of this paper are to study partially collaborative codes, in the presence of an eavesdropper, and to analyze their repair bandwidth as a function of different parameters. Note that we are not concerned with code design here.

1. We give (in Section 2) a bound on the secure file size for the minimum bandwidth case, for which we compute the optimal parameters α, β, β' , which in particular gives a generalization of [10]. We analyze what are the best strategies in terms of minimizing the repair bandwidth, as a function of different parameters. The number d of nodes contacted per repair plays an important role, and depending on the value of d , we derive when no collaboration, full collaboration, or partial collaboration is better as a function of the repair threshold t .
2. In Section 3, we compute a bound on the secure file size for the minimum storage point. We obtain a generalization of [10], and we discuss the security of stored files in terms of a bound on the size of their secure content, as a function of the collaboration involved during repair.

Notations are summarized in Table 1 for convenience.

Figure 1: On the upper left figure, a snapshot of a network of nodes is shown, where $n = 9$ out of the 13 nodes are storing an object $\mathbf{o}$. On the upper right figure, the system has been waiting for $t = 4$ failures to happen before triggering a repair process. On the lower left figure, the download phase is shown, where $t = 4$ nodes download data from $d = 2$ nodes each. On the lower right figure, the exchange phase completes the repair, after which, the object $\mathbf{o}$ is again stored across $n = 9$ nodes.

2 The Case of Minimum Repair Bandwidth

Consider an (n, k) storage code made to minimize the repair bandwidth γ , that is data objects are encoded, and the encoded pieces are stored across n nodes, such that any k of them are enough to recover the data object. The repair bandwidth per node must at least be the content of the storage node α , so the minimum repair bandwidth is obtained when $\gamma = \alpha = d\beta + (t - s)\beta'$.

The encoded pieces $s_1, \dots, s_n$ are each stored at a distinct node, we label the nodes such that node i stores s_i , $i = 1, \dots, n$. We label the nodes participating to the repair process by $n + j$, $j \geq 1$.

In the presence of an eavesdropper, data objects are encrypted for ensuring their confidentiality. We therefore distinguish the secure file $\mathbf{o}^s$, of size $\mathcal{M}^s$, from its encrypted version, the file $\mathbf{o}$ of size $\mathcal{M}$. Encoding is then done on top of the encrypted file $\mathbf{o}$. Now an eavesdropper may spy on the content of some nodes, but also on the incoming links of the nodes. However for a storage code that minimizes the repair bandwidth, the contents of the node and that of the incoming

Table 1: Summary of Notations.

symbol	meaning
$\mathcal{M}$	size of data object
α	storage capacity per node
n	number of nodes storing data from one object
k	number of nodes contacted for object retrieval
t	number of failures which triggers a repair
d	number of nodes contacted per repair
β	amount of data downloaded during repair per node
β'	amount of data exchanged during repair per node
γ	total repair bandwidth per repair
$\mathbf{o}$	file (data object) to be stored
$\mathbf{s}_i$	encoded data stored at node i
s	degree of collaboration

links are the same.

Definition 1 Let $\mathbf{o}^s$ be a file of size $\mathcal{M}^s$, which is encrypted into a secure file $\mathbf{o}$, of size $\mathcal{M}$. A networked distributed storage system is said to achieve a secure file size of $\mathcal{M}^s$ against an l_1 -eavesdropper, $l_1 \leq k$, if, for any set ε_1 of size l_1

$$I(\mathbf{o}^s; \mathbf{e}) = 0,$$

where $\mathbf{e} = \{\mathbf{s}_i, i \in \varepsilon_1\}$ is the eavesdropper observation vector.

2.1 A Min-Cut Bound

To understand the effect of the presence of an eavesdropper, we use known techniques, in particular a max-flow min-cut bound in the storage network, and a standard upper bound on the leaked information [8, 5, 6], as detailed next.

Note that by definition of mutual information

$$I(\mathbf{o}^s; \mathbf{e}) = 0 \iff H(\mathbf{o}^s | \mathbf{e}) = H(\mathbf{o}^s).$$

Suppose that over time, k nodes have failed and been repaired at nodes $n + 1, \dots, n + k$ by g consecutive groups, each of size u_j , $j = 0, \dots, g - 1$, that is

$\sum_{j=0}^{g-1} u_j = k$, and $1 \leq u_j \leq t$. Suppose that $\varepsilon_1 \subseteq \{n+1, \dots, n+k\}$, then when a data collector contacts these k nodes, we have

$$\begin{aligned} H(\mathbf{o}^s | \mathbf{e}) &= H(\mathbf{o}^s | \mathbf{s}_i, i \in \varepsilon_1) \\ &= H(\mathbf{o}^s | \mathbf{s}_i, i \in \varepsilon_1) - H(\mathbf{o}^s | \mathbf{s}_j, j = n+1, \dots, n+k) \end{aligned} \quad (1)$$

$$= I(\mathbf{o}^s; \mathbf{s}_j, j \in \{n+1, \dots, n+k\} \setminus \varepsilon_1 | \mathbf{s}_i, i \in \varepsilon_1) \quad (2)$$

$$\leq H(\mathbf{s}_j, j \in \{n+1, \dots, n+k\} \setminus \varepsilon_1 | \mathbf{s}_i, i \in \varepsilon_1) \quad (3)$$

$$= \sum_{j \in \{n+1, \dots, n+k\} \setminus \varepsilon_1} H(\mathbf{s}_j | \mathbf{s}_{n+1}, \dots, \mathbf{s}_{j-1}, \mathbf{s}_i, i \in \varepsilon_1) \quad (4)$$

Indeed, (1) holds because $H(\mathbf{o}^s | \mathbf{s}_j, j = n+1, \dots, n+k) = 0$ since the knowledge of any k encoded pieces is enough to recover $\mathbf{o}$. Then (2) comes from the definition of conditional mutual information $I(X; Y | Z) = H(X | Z) - H(X | Y, Z)$. Alternatively, one may develop the conditional mutual information as $I(X; Y | Z) = H(X, Z) - H(Z) - H(X, Y, Z) + H(Y, Z) = H(X, Z) - H(X, Y, Z) + H(Y | Z) \leq H(Y | Z)$, which justifies (3). Finally, (4) holds because $H(X_1, \dots, X_n | Y) = \sum_{i=1}^n H(X_i | X_1, \dots, X_{i-1}, Y)$.

Proposition 1 *For an (n, k) partially collaborative storage code at the minimum repair bandwidth point (MBR), we have that the secure file size $\mathcal{M}^s$ is upper bounded by*

$$\mathcal{M}^s \leq (k - l_1)(2d + t - k - l_1 - s + 1) \frac{\beta}{2} \quad (5)$$

in the presence of an l_1 -eavesdropper spying the stored content of l_1 nodes with $0 \leq l_1 \leq k$. We further have $\beta = 2\beta'$ and the optimal file size is reached with equality in (5) and

$$\beta = \frac{\mathcal{M}^s}{k - l_1} \frac{2}{2d + t - s + 1 - k - l_1}, \quad \alpha = \frac{\mathcal{M}^s}{k - l_1} \frac{2d + t - s}{2d + t - s + 1 - k - l_1}.$$

Proof: We want $H(\mathbf{o}^s | \mathbf{e}) = H(\mathbf{o}^s)$, and we have the upper bound

$$H(\mathbf{o}^s) \leq \sum_{j \in \{n+1, \dots, n+k\} \setminus \varepsilon_1} H(\mathbf{s}_j | \mathbf{s}_{n+1}, \dots, \mathbf{s}_{j-1}, \mathbf{s}_i, i \in \varepsilon_1)$$

from (4). To upper bound $H(\mathbf{s}_j | \mathbf{s}_{n+1}, \dots, \mathbf{s}_{j-1}, \mathbf{s}_i, i \in \varepsilon_1)$, as is standardly done [8, 1, 2, 12], we consider a cut in the information flow graph that represents the

flow of information from a data owner which uploads the data, to the data collector which gathers it, via different repair phases. The cut involves the k nodes $n + j$, $j = 1, \dots, k$, that have been repaired, among which $l_1 = |\varepsilon_1|$ are observed by the eavesdropper. The k repaired nodes are grouped into g consecutive groups of size u_i , $i = 0, \dots, g - 1$. Each group of u_i nodes is partitioned into four groups:

- $l_1^{i,1}$ nodes have their content eavesdropped,
- $m_i - l_1^{i,1}$ nodes have their content that contribute $(m_i - l_1^{i,1})\alpha$ to the cut,
- $l_1^{i,2}$ nodes have their outgoing edges eavesdropped,
- $u_i - m_i - l_1^{i,2}$ nodes have their outgoing edges that contribute $(d - \sum_{j=0}^{i-1} u_j)\beta$ from the download phase, and $(t - s + 1 - u_i)\beta'$ from the collaborative phase (when $t - s + 1 \geq u_i$) to the cut,

with $l_1^i = l_1^{i,1} + l_1^{i,2}$, and $\sum_{i=0}^{g-1} (l_1^{i,1} + l_1^{i,2}) = \sum_{i=0}^{g-1} l_1^i = l_1$.

Set $J = \{i, t - s + 1 - u_i \geq 0\}$ and $\bar{J} = \{i, t - s + 1 - u_i < 0\}$ then

$$\begin{aligned}
H(\mathbf{o}^s) &\leq \sum_{j \in R} H(\mathbf{s}_j | \mathbf{s}_{n+1}, \dots, \mathbf{s}_{j-1}, \mathbf{s}_i, i \in \varepsilon_1) \\
&\leq \sum_{i \in J} \{ (m_i - l_1^{i,1})\alpha + (u_i - m_i - l_1^{i,2})[(d - \sum_{j=0}^{i-1} u_j)\beta + (t - s + 1 - u_i)\beta'] \} \\
&\quad + \sum_{i \in \bar{J}} \{ (m_i - l_1^{i,1})\alpha + (u_i - m_i - l_1^{i,2})(d - \sum_{j=0}^{i-1} u_j)\beta \} \\
&\leq \sum_{i \in J} (u_i - l_1^i) \min\{ \alpha, (d - \sum_{j=0}^{i-1} u_j)\beta + (t - s + 1 - u_i)\beta' \} \\
&\quad + \sum_{i \in \bar{J}} (u_i - l_1^i) \min\{ \alpha, (d - \sum_{j=0}^{i-1} u_j)\beta \}
\end{aligned}$$

using Jensen's inequality. Since at MBR point $\alpha = \gamma = d\beta + (t - s)\beta'$

$$H(\mathbf{o}^s) \leq \sum_{i \in J} (u_i - l_1^i) [(d - \sum_{j=0}^{i-1} u_j)\beta + (t - s + 1 - u_i)\beta'] + \sum_{i \in \bar{J}} (u_i - l_1^i) (d - \sum_{j=0}^{i-1} u_j)\beta. \quad (6)$$

To evaluate β , we note from (6) that the smallest contribution in β' occur when $u_i = t - s + 1, t - s + 1 | k$, in which case (6) becomes

$$\begin{aligned}
H(\mathbf{o}^s) &\leq \sum_{i=0}^{g-1} (u_i - l_1^i) (d - \sum_{j=0}^{i-1} u_j) \beta \\
&= [d(k - l_1) - (t - s + 1) \sum_{i=0}^{g-1} i(u_i - l_1^i)] \beta \\
&= [d(k - l_1) - \frac{k(k - t + s - 1)}{2} + (t - s + 1) \sum_{i=0}^{g-1} i l_1^i] \beta
\end{aligned}$$

since $\sum_{i=0}^{g-1} u_i = \sum_{i=0}^{g-1} (t - s + 1) = k$ means that $g = \frac{k}{t-s+1}$. The minimum is obtained for $t - s + 1 | l_1$ and for the values of l_1^i such that $l_1^i = u_i$ for $i = 0, \dots, \frac{l_1}{t-s+1} - 1$, and $l_1^i = 0$ for $i \geq \frac{l_1}{t-s+1}$. We then get

$$\begin{aligned}
H(\mathbf{o}^s) &\leq [d(k - l_1) - \frac{k(k - t + s - 1)}{2} + (t - s + 1)^2 \sum_{i=0}^{\frac{l_1}{t-s+1} - 1} i] \beta \\
&= [d(k - l_1) - \frac{k(k - t + s - 1)}{2} + \frac{l_1(l_1 - (t - s + 1))}{2}] \beta \\
&= [d(k - l_1) + \frac{(t - s + 1)(k - l_1) - (k - l_1)(k + l_1)}{2}] \beta \\
&= \frac{(k - l_1)}{2} [2d + t - s + 1 - k - l_1] \beta.
\end{aligned}$$

Conversely, the smallest contribution in β happens when $u_i = 1$, thus combining (6) with the minimal value of β , namely

$$\beta = \frac{H(\mathbf{o}^s)}{k - l_1} \frac{2}{2d + t - s + 1 - k - l_1}$$

yields

$$H(\mathbf{o}^s) \leq \frac{(2d + 1 - k - l_1) H(\mathbf{o}^s)}{2d + t - s + 1 - k - l_1} + (k - l_1)(t - s) \beta'$$

that is

$$\beta' \geq \frac{H(\mathbf{o}^s)}{k - l_1} \frac{1}{2d + t - s + 1 - k - l_1}$$

and the minimal values of β and β' satisfy $\beta = 2\beta'$.

When $g = k$, $u_i = 1, i = 0, \dots, g - 1$, and we have

$$H(\mathbf{o}^s) \leq \frac{(k - l_1)(2d - k - l_1 + 1)\beta}{2} + (k - l_1)(t - s)\beta'.$$

When $g = 1$, $t \geq k$, then $u_0 = k$, $l_1^0 = l_1$, if $t \leq k + s - 1$, the bound will be

$$H(\mathbf{o}^s) \leq (k - l_1)[d\beta + (t - s + 1 - k)\beta'];$$

if $t < k + s - 1$, the bound will be

$$H(\mathbf{o}^s) \leq (k - l_1)d\beta.$$

When $1 < g < k$, since we want minimize the bound, we need to minimize the size of J , or alternatively to maximize u_i . Let $t < k$, $u_i = t, i = 0, \dots, g - 2$, $g = \lfloor k/t \rfloor + 1$, then $u_{g-1} = k - \lfloor k/t \rfloor t$, we get the bound

$$H(\mathbf{o}^s) \leq \min_{l_1^i \leq t} \left\{ \sum_{i=0}^{g-2} (t - l_1^i)(d - it)\beta + (k - \lfloor k/t \rfloor t - l_1^{g-1})[(d - \lfloor k/t \rfloor t)\beta + (t - s + 1 - k + \lfloor k/t \rfloor t)\beta'] \right\}.$$

The tightest bound is obtained when $g = k$, $u_i = 1, i = 0, \dots, g - 1$.

The relation $\beta = 2\beta'$ in (6) gives the bound (5).

Recalling that $\alpha = \gamma = d\beta + (t - s)\beta'$ finally gives

$$\alpha = 2d\beta' + (t - s)\beta' = \beta'(2d + t - s).$$

□

The upper bound on the secure file size (5) of the above proposition is illustrated in Figure 2, as a function of s , $1 \leq s \leq 5$, for fixed $l_1, d = k = t = 5$, and $\beta = 2$:

$$\mathcal{M}^s \leq (5 - l_1)(10 - l_1 - s + 1). \quad (7)$$

The upper bound is monotonously decreasing, with extremes on the boundary points, that is maximum when $s = 1$ and minimum when $s = t = 5$. This shows that no collaboration ($s = t$) has the smallest secure file size, see Figure 2. Increasing the number of eavesdropper decreases the secure file size.

We recover the following particular cases:

Figure 2: The secure file size $\mathcal{M}^s$ as a function of $1 \leq s \leq 5$ with $d = k = t = 5$, the lines from top to bottom represent the eavesdropper number l_1 from 1 to 4 respectively.

Corollary 2.1 [8] *For a fully collaborative storage code operating at the MBR point, the secure file size $\mathcal{M}^s$ is upper bounded by*

$$\mathcal{M}^s \leq (k - l_1)(2d + t - k - l_1) \frac{\beta}{2}.$$

Proof: Full collaboration corresponds to $s = 1$. Thus set $s = 1$ into $\mathcal{M}^s \leq (k - l_1)(2d + t - k - l_1 - s + 1) \frac{\beta}{2}$. $\square$

Corollary 2.2 *For a storage code with no collaboration operating at the MBR point, the secure file size $\mathcal{M}^s$ is upper bounded by*

$$\mathcal{M}^s \leq (k - l_1)(2d - k - l_1 + 1) \frac{\beta}{2}.$$

Proof: No collaboration corresponds to $s = t$. Then set $s = t$ into $\mathcal{M}^s \leq (k - l_1)(2d + t - k - l_1 - s + 1) \frac{\beta}{2}$. $\square$

Corollary 2.3 [10] *For a partially collaborative code with no eavesdropper at the minimum bandwidth repair (MBR) point, we have*

$$\alpha = \frac{\mathcal{M}}{k} \frac{2d+t-s}{2d-k+t-s+1}, \beta = \frac{\mathcal{M}}{k} \frac{2}{2d-k+t-s+1}, \beta' = \frac{\mathcal{M}}{k} \frac{1}{2d-k+t-s+1}$$

for a total repair bandwidth of

$$\gamma = \frac{\mathcal{M}}{k} \frac{2d+t-s}{2d+t-k-s+1}.$$

Proof: From the above proposition, we have $\beta = 2\beta'$ and

$$\beta = \frac{\mathcal{M}^s}{k-l_1} \frac{2}{2d+t-s+1-k-l_1}, \alpha = \frac{\mathcal{M}^s}{k-l_1} \frac{2d+t-s}{2d+t-s+1-k-l_1}.$$

Just set $l_1 = 0$. $\square$

Observation 2.1 *It is also possible to use the result of [10] to claim that $\beta = 2\beta'$ at MBR, in which case the proof of the above proposition can be simplified by using $\beta = 2\beta'$ inside. Then Corollary 2.3 is not a corollary anymore, rather, the proposition becomes a corollary of the results from [10]. We preferred the more general approach which gives the result of [10] at MBR as a corollary of the present analysis.*

We given an example of partial collaboration with no eavesdropper, which is illustrated on Figure 3.

Example 1 [11] Consider for example an object $\mathbf{o} = (o_1, o_2, \dots, o_8) \in \mathbb{F}_q^8$ of size $\mathcal{M} = 8$. Encode this object using a single parity check code, that is, $\mathbf{o}$ is encoded into the length 9 codeword $\mathbf{x}$ given by $\mathbf{x} = (o_1, o_2, \dots, o_8, o_1 + o_2 + \dots + o_8)$. The codefficients x_i are allocated into the nodes 1, ..., 9, as follows:

node 1	x_1, x_6, x_7, x_8, x_9	node 6	x_2, x_3, x_4, x_5, x_6
node 2	x_1, x_2, x_7, x_8, x_9	node 7	x_3, x_4, x_5, x_6, x_7
node 3	x_1, x_2, x_3, x_8, x_9	node 8	x_4, x_5, x_6, x_7, x_8
node 4	x_1, x_2, x_3, x_4, x_9	node 9	x_5, x_6, x_7, x_8, x_9
node 5	x_1, x_2, x_3, x_4, x_5		

This is a particular instantiation of a more general algorithm described in [11]. Suppose that $t = 4$ failures need to be repaired, with repair degree $d = 2$, and cooperative repair degree $t - s = 1$. By abuse of notation, we will call node i below the node that is actually repairing node i . Here is an example of repair. Suppose that nodes 1, 2, 3 and 4 fail. During the downloading phase:

Figure 3: A snapshot of a network of nodes is shown, where $n = 9$ out of the 13 nodes are storing an object $\mathbf{o}$ and $t = 4$ failures happened. The download phase is shown on the right, where $t = 4$ nodes download data from $d = 2$ nodes each. The exchange phase completes the repair on the left, after which, the object $\mathbf{o}$ is again stored across $n = 9$ nodes.

- node 1 gets x_6, x_7 from node 8, and x_8, x_9 from node 9,
- node 2 gets x_1, x_2 from node 5, and x_7, x_8 from node 8,
- node 3 gets x_1, x_3 from node 5, and x_8, x_9 from node 9,
- node 4 gets x_1, x_4 from node 5, and x_2, x_3 from node 6

and during the collaboration phase: node 1 sends x_9 to node 4, node 4 sends x_2 to node 3, node 3 sends x_9 to node 2, and node 2 sends x_1 to node 1. We note that this strategy is not unique.

This example has for parameters $\mathcal{M} = 8$, $d = 2$, $k = t = 4$, $s = 3$. The amount of data stored per node is $\alpha = 5$, every node downloads $\beta = 2$ symbols from $d = 2$ nodes, and exchanges $\beta' = 1$ symbol, thus the repair bandwidth is $\gamma = 5$.

We can check that this code is at MBR, since

$$\alpha = \frac{\mathcal{M}}{k} \frac{2d + t - s}{2d - k + t - s + 1} = 5 = \gamma$$

and

$$\beta = \frac{\mathcal{M}}{k} \frac{2}{2d - k + t - s + 1} = 2, \quad \beta' = \frac{\mathcal{M}}{k} \frac{1}{2d - k + t - s + 1} = 1.$$

2.2 A Bandwidth Analysis

To understand the effect of the different parameters on the repair bandwidth, we consider the normalized repair bandwidth $\frac{\gamma}{\mathcal{M}^s}$, introduced in [8]. Since we have computed upper bounds on $\mathcal{M}^s$, we will be able to provide a lower bound $\bar{\gamma}$ on the normalized repair bandwidth:

$$\frac{\gamma}{\mathcal{M}^s} \geq \bar{\gamma}(s, t) = \frac{2d + t - s}{(k - l_1)(2d + t - s - k - l_1 + 1)}.$$

Note that the regime of no collaboration holds only for $d \geq k$, and that $d \leq n - t$ (once t nodes have failed, only $n - t$ nodes are still live and at most all of them can be contacted). We thus have $k \leq d \leq n - t$.

A first observation is that $\bar{\gamma}(s, t)$ is always greater when $l_1 > 0$ compared to $l_1 = 0$, that is having an eavesdropper does increase $\bar{\gamma}(s, t)$.

Suppose first that all live nodes participate to the repair process, that is $d = n - t$:

$$\frac{\gamma}{\mathcal{M}^s} \geq \bar{\gamma}(s, t) = \frac{(2n - t - s)}{(k - l_1)(2n - t - s - k - l_1 + 1)}.$$

The two extreme cases for s are no collaboration ($t = 1$ and $s = t$), for which

$$\bar{\gamma}(s = t = 1) = \frac{1}{k - l_1} \frac{2n - 2}{2n - k - l_1 - 1},$$

and full collaboration ($t = t_0 \geq 2$ and $s = 1$):

$$\bar{\gamma}(s = 1, t = t_0) = \frac{1}{k - l_1} \frac{2n - t_0 - 1}{2n - k - l_1 - t_0}.$$

We have

$$\begin{aligned} \bar{\gamma}(s = t = 1) &\leq \bar{\gamma}(s, t) \\ \iff (2n - k - l_1 - t - s + 1)(2n - 2) &\leq (2n - t - s)(2n - k - l_1 - 1) \\ \iff (t + s - 2)(k + l_1 - 1) &\geq 0 \end{aligned}$$

which always holds for $t \geq 2$, corresponding to the case where collaboration does not reduce the normalized repair bandwidth with respect to no collaboration, no matter the degree of collaboration $1 \leq s \leq t$. If we compare instead partial collaboration and full collaboration, then

$$\bar{\gamma}(s, t) \leq \bar{\gamma}(s = 1, t = t_0) \iff t_0 \geq t + s - 1.$$

The interpretation is that the best strategy is to repair one failure at a time. In case of correlated failures, the system is forced to repair several failures in parallel. If there is a coding strategy that allows full collaboration for a small threshold t_0 which can take care of the correlated failures, then this will give the least normalized repair bandwidth (if $t_0 = t$, then $t_0 \geq t + s - 1$ exactly when $s = 1$ which is full collaboration). If however there exists a partial collaboration for a choice of s, t such that $t_0 \geq t + s - 1$, which can happen when the full collaborative strategy exists for a large threshold t_0 , then partial collaboration is better. This holds irrespectively of the value of l_1 .

Consider next the case $n > d + t \iff d < n - t$. Comparing no collaboration with partial collaboration (and in particular full collaboration), the reverse conclusion is obtained in that partial collaboration reduces the normalized repair bandwidth with respect to no collaboration:

$$\bar{\gamma}(s = t = 1) = \frac{1}{k - l_1} \frac{2d}{2d - k - l_1 + 1} > \bar{\gamma}(s, t) = \frac{1}{k - l_1} \frac{2d + t - s}{2d - k - l_1 + t - s + 1}$$

for $t - s \geq 1$. Next comparing partial collaboration and full collaboration, we have

$$\bar{\gamma}(s, t) \leq \bar{\gamma}(s = 1, t = t_0) \iff t - s + 1 \geq t_0.$$

The interpretation in this case is that it is best to delay to repair until several failures happen. If there is a coding strategy that allows full collaboration for a large threshold t_0 , then this will give the least normalized repair bandwidth (if $t_0 = t$, then $t - s \geq t_0 - 1$ exactly when $s = 1$ which is full collaboration). If however there exists a partial collaboration for a choice of s, t such that $t - s + 1 \geq t_0$, which can happen when the full collaborative strategy exists for a small threshold t_0 , then it is best to delay further the repair, and partial collaboration is better in terms of normalized bandwidth. This holds irrespectively of the value of l_1 .

3 The Case of Minimum Storage

Consider a storage code that minimizes the storage capacity α . The storage capacity must be at least $\mathcal{M}/k$, since the object $\mathbf{o}$ has size $\mathcal{M}$, and any k nodes allow the retrieval of $\mathbf{o}$. We thus have $\alpha = \mathcal{M}/k$.

The encoded pieces $s_1, \dots, s_n$ are each stored at a distinct node and as before, we label the nodes such that node i stores s_i , $i = 1, \dots, n$, and the nodes participating to the repair process by $n + j$, $j \geq 1$.

Now an eavesdropper may spy on the content of some nodes, but also on the incoming links of the nodes. It is important to distinguish both cases, because the amount of information that is transmitted to a node during the repair process could be more than what is actually stored, especially at the minimum storage capacity $\alpha = \mathcal{M}/k$. In fact, it was observed in [7], in the case of $t = 1$ failure, that at MSR point, the node that performs the repair has actually access to more data ($\mathcal{M}/k(d-k+1)$) than what it can store ($\mathcal{M}/k$). This is not the case at MBR point, as seen in the above computations. This has consequences in the presence of an eavesdropper, who can access the data that was downloaded during the repair: the eavesdropper gains more information.

The same is true in the partially collaboration setting. We know from [10] that every node in the set of t nodes that participate in the repair of t failures at MSR point access to more data ($\mathcal{M}(d+t-s)/k(d-k+t-s+1)$) than what it can store ($\mathcal{M}/k$), whenever $k \geq 2$.

Suppose that we use a linear code to encode the data file $\mathbf{o}$, so that $\mathbf{s}_i \in \mathbb{F}_q^\alpha$ may then be written as

$$\mathbf{s}_i = (\mathbf{o}^T \mathbf{g}_i^1, \dots, \mathbf{o}^T \mathbf{g}_i^\alpha)$$

for some column vectors $\mathbf{g}_i^1, \dots, \mathbf{g}_i^\alpha$.

Let $\mathcal{S}_i$ be the subspace spanned by the vectors $\{\mathbf{g}_i^1, \dots, \mathbf{g}_i^\alpha\}$, $i = 1, \dots, n$, so that the content of each node may be seen as a vector space of dimension α . Let $\mathbf{d}_{i,j} \in \mathbb{F}_q^\beta$ be the downloaded data from node i to node j , which may be written as

$$\mathbf{d}_{i,j} = (\mathbf{o}^T \mathbf{g}_{i,j}^1, \dots, \mathbf{o}^T \mathbf{g}_{i,j}^\beta),$$

where $\mathbf{g}_{i,j}^r \in \mathcal{S}_i$, $r = 1, \dots, \beta$, and we refer to $\mathcal{D}_{i,j}$ as the subspace spanned by the column vectors $\{\mathbf{g}_{i,j}^1, \dots, \mathbf{g}_{i,j}^\beta\}$.

We collect all the downloaded data to the node j in $\mathbf{d}_j \in \mathbb{F}_q^{d\beta}$. Similarly, let $\mathbf{d}_{A,B}$ be the set of all the downloaded data from a set of nodes indexed by A to a set of nodes indexed by B , and let $\mathcal{D}_{A,B}$ be the subspace spanned by the vectors $\mathbf{g}_{i,j}$, $i \in A, j \in B$.

Let $\mathbf{d}'_j \in \mathbb{F}_q^{\beta'(t-s)}$ be all the data obtained during the exchange phase at node j , and $\mathbf{d}'_{A,B}$ be all the data exchanged from the set of nodes indexed by A to the set of nodes indexed by B .

Definition 2 Let $\mathbf{o}^s$ be a secret file of size $\mathcal{M}^s$, which is encrypted into a secure file $\mathbf{o}$, of size $\mathcal{M}$. A networked distributed storage system is said to achieve a secure file size of $\mathcal{M}^s$ against an (l_1, l_2) eavesdropper, if, for any disjoint sets ε_1

and ε_2 of size l_1 and l_2 , respectively, $|l_1 \cup l_2| \leq k$,

$$I(\mathbf{o}^s; \mathbf{e}) = 0,$$

where $\mathbf{e} = \{\mathbf{s}_i, i \in \varepsilon_1, \mathbf{d}_j, \mathbf{d}'_j, j \in \varepsilon_2\}$ is the eavesdropper observation vector.

3.1 A Min-Cut Bound

Suppose that over time, k nodes have failed and been repaired at nodes $n + 1, \dots, n + k$ by g consecutive groups, each of size $u_j, j = 0, \dots, g - 1$, that is $\sum_{j=0}^{g-1} u_j = k$, and $1 \leq u_j \leq t$. Suppose that $(\varepsilon_1 \cup \varepsilon_2) \subseteq \{n + 1, \dots, n + k\}$, let $R = \{n + 1, \dots, n + k\} \setminus (\varepsilon_1 \cup \varepsilon_2)$, then when a data collector contacts these k nodes, we have

$$\begin{aligned} H(\mathbf{o}^s | \mathbf{e}) &= H(\mathbf{o}^s | \mathbf{s}_i, i \in \varepsilon_1, \mathbf{d}_j, \mathbf{d}'_j, j \in \varepsilon_2) \\ &= H(\mathbf{o}^s | \mathbf{s}_i, i \in \varepsilon_1, \mathbf{d}_j, \mathbf{d}'_j, j \in \varepsilon_2) - H(\mathbf{o}^s | \mathbf{s}_i, i \in \varepsilon_1, \mathbf{s}_R, \mathbf{d}_j, \mathbf{d}'_j, j \in \varepsilon_2) \\ &= I(\mathbf{o}^s; \mathbf{s}_R | \mathbf{s}_i, i \in \varepsilon_1, \mathbf{d}_j, \mathbf{d}'_j, j \in \varepsilon_2) \\ &\leq H(\mathbf{s}_R | \mathbf{s}_i, i \in \varepsilon_1, \mathbf{d}_j, \mathbf{d}'_j, j \in \varepsilon_2) \\ &= \sum_{j \in R} H(\mathbf{s}_j | \mathbf{s}_{n+1}, \dots, \mathbf{s}_{j-1}, \mathbf{s}_i, i \in \varepsilon_1, \mathbf{d}_j, \mathbf{d}'_j, j \in \varepsilon_2) \end{aligned}$$

using the same arguments as in the previous section, with the slight variation that $H(\mathbf{o}^s | \mathbf{s}_i, i \in \varepsilon_1, \mathbf{s}_R, \mathbf{d}_j, \mathbf{d}'_j, j \in \varepsilon_2) = 0$ because the knowledge of $\mathbf{d}_j, \mathbf{d}'_j, j \in \varepsilon_2$ gives that of $\mathbf{s}_j, j \in \varepsilon_2$.

Proposition 2 *For an (n, k) partially collaborative storage code at the minimum storage repair point (MSR), we have that the secure file size $\mathcal{M}^s$ is upper bounded by*

$$\mathcal{M}^s \leq \sum_{i=1}^{k-l_1-l_2} (\alpha - I(\mathbf{s}_i; \mathbf{d}_{i, \varepsilon_2})) \quad (8)$$

If $I(\mathbf{s}_i; \mathbf{d}_{i, \varepsilon_2}) \geq \beta' = \beta$, then (8) becomes

$$\mathcal{M}^s \leq (k - l_1 - l_2)(\alpha - \beta), \quad (9)$$

in the presence of an (l_1, l_2) eavesdropper spying the stored content of l_1 nodes and the downloaded and exchanged data of l_2 nodes, where $l_1 + l_2 \leq k$. Furthermore $\beta = \beta'$ and the optimal secure file size is reached with equality in (9), with

$$\beta = \beta' = \frac{\alpha}{d - k + t - s + 1}.$$

Proof: Recall that nodes in ε_1 are eavesdropped on what they store only, while nodes in ε_2 are eavesdropped not only on what they store, but also on what they download and exchange. Each group of u_i nodes is partitioned into the following two groups:

- $l_1^{i,1}, l_2^{i,1}$ nodes are having what they store eavesdropped, among which, $m_i - l_1^{i,1} - l_2^{i,1}$ of them belong to R , which are denoted by h_i . They may leak information to the nodes in ε_2 , so they contribute $(m_i - l_1^{i,1} - l_2^{i,1})\alpha - \dim \mathcal{D}_{h_i; \varepsilon_2}$ to the cut.
- $l_1^{i,2}, l_2^{i,2}$ nodes are being eavesdropped during the download phase, among which, $u_i - m_i - l_1^{i,2} - l_2^{i,2}$ of them are in R , contributing either $(d - \sum_{j=0}^{i-1} u_j)\beta + (t - s + 1 - u_i)\beta'$ or $(d - \sum_{j=0}^{i-1} u_j)\beta$ to the cut,

with $l_1^i = l_1^{i,1} + l_1^{i,2}$, $l_2^i = l_2^{i,1} + l_2^{i,2}$ and $\sum_{i=0}^{g-1} l_1^i = l_1$, $\sum_{i=0}^{g-1} l_2^i = l_2$.

By Jensen's inequality, we have

$$\begin{aligned} \mathcal{M}^s \leq & \sum_{i \in J} (u_i - l_1^i - l_2^i) \min\left\{\alpha - \frac{\dim \mathcal{D}_{h_i, \varepsilon_2}}{u_i - l_1^i - l_2^i}, (d - \sum_{j=0}^{i-1} u_j)\beta + (t - s + 1 - u_i)\beta'\right\} \\ & + \sum_{i \in \bar{J}} (u_i - l_1^i - l_2^i) \min\left\{\alpha - \frac{\dim \mathcal{D}_{h_i, \varepsilon_2}}{u_i - l_1^i - l_2^i}, (d - \sum_{j=0}^{i-1} u_j)\beta\right\}. \end{aligned} \quad (10)$$

The highest contribution from β comes from no contribution in β' , that is when $u_i = t - s + 1$, for $i = 0, \dots, k/(t - s + 1) - 1$. Let $u_i = t - s + 1$, for $i = 0, \dots, \lfloor k/(t - s + 1) \rfloor - 1$, and $u_i = k - \lfloor k/(t - s + 1) \rfloor (t - s + 1)$, for $i = \lfloor k/(t - s + 1) \rfloor$. Then $g = \lfloor k/(t - s + 1) \rfloor + 1$, $\bar{J}$ is empty, and (10) becomes

$$\begin{aligned} \mathcal{M}^s \leq & (u_{g-1} - l_1^{g-1} - l_2^{g-1}) \min\left\{\alpha - \frac{\dim \mathcal{D}_{h_{g-1}, \varepsilon_2}}{u_{g-1} - l_1^{g-1} - l_2^{g-1}}, \right. \\ & \left. (d - \sum_{j=0}^{g-2} u_j)\beta + (t - s + 1 - u_{g-1})\beta'\right\} \\ & + \sum_{i=0}^{g-2} (u_i - l_1^i - l_2^i) \min\left\{\alpha - \frac{\dim \mathcal{D}_{h_i, \varepsilon_2}}{u_i - l_1^i - l_2^i}, (d - i(t - s + 1))\beta\right\}. \end{aligned}$$

Each term in the sum is at most

$$(u_i - l_1^i - l_2^i) \left(\alpha - \frac{\dim \mathcal{D}_{h_i, \varepsilon_2}}{u_i - l_1^i - l_2^i} \right) = (u_i - l_1^i - l_2^i) \alpha - \dim \mathcal{D}_{h_i, \varepsilon_2},$$

and equality can be obtained only when

$$(k - l_1 - l_2)\alpha - \dim \mathcal{D}_{R, \varepsilon_2} = \mathcal{M}^s.$$

Thus each term which contains β should be no less than $\alpha - \frac{\dim \mathcal{D}_{h_i, \varepsilon_2}}{u_i - l_1^i - l_2^i}$.

For $i = 0, \dots, g-2$,

$$(d - i(t - s + 1))\beta \geq \alpha - \frac{\dim \mathcal{D}_{h_i, \varepsilon_2}}{u_i - l_1^i - l_2^i},$$

and since $\dim \mathcal{D}_{h_i, \varepsilon_2} \geq 0$,

$$\beta \geq \frac{\alpha}{d - i(t - s + 1)}.$$

When $i = g-2 = \lfloor k/(t-s+1) \rfloor - 1$,

$$\beta = \frac{\alpha}{d + t - s + 1 - \lfloor k/(t-s+1) \rfloor (t-s+1)},$$

and since $\lfloor k/(t-s+1) \rfloor (t-s+1) \leq k$, we have that the minimum value of β is

$$\beta = \frac{\alpha}{d - k + t - s + 1}. \quad (11)$$

For $i = g-1$,

$$(d - (g-1)(t-s+1))\beta + (t-s+1 - u_{g-1})\beta' \geq \alpha - \frac{\dim \mathcal{D}_{h_{g-1}, \varepsilon_2}}{u_{g-1} - l_1^{g-1} - l_2^{g-1}},$$

then taking into account that $\dim \mathcal{D}_{h_{g-1}, \varepsilon_2} \geq 0$ and (11),

$$\begin{aligned} \beta' &\geq \frac{\alpha - (d - (g-1)(t-s+1))\beta}{t-s+1 - u_{g-1}} \\ &= \frac{\alpha(t-s+1 - k + u_{g-1})}{(d - k + t - s + 1)(t-s+1 - u_{g-1})} \\ &= \frac{\alpha}{d - k + t - s + 1}. \end{aligned} \quad (12)$$

Conversely, the highest contribution from β' comes when $u_i = 1$ for all $i = 0, \dots, g-1$, so that $g = k$ and $i = 0, \dots, k-1$, then (10) becomes

$$\mathcal{M}^s \leq \sum_{i=0}^{k-1} (1 - l_1^i - l_2^i) \min\left\{\alpha - \frac{\dim \mathcal{D}_{h_i, \varepsilon_2}}{u_i - l_1^i - l_2^i}, (d-i)\beta + (t-s)\beta'\right\}.$$

Here we still have that each term is at most

$$(1 - l_1^i - l_2^i)(\alpha - \frac{\dim \mathcal{D}_{h_i, \varepsilon_2}}{1 - l_1^i - l_2^i}),$$

and equality is actually obtained when each term is equal to it.

Thus, each term that contains β and β' should be no less than $\alpha - \frac{\dim \mathcal{D}_{h_i, \varepsilon_2}}{1 - l_1^i - l_2^i}$, that is for $i = 0, \dots, k - 1$,

$$(d - i)\beta + (t - s)\beta' \geq \alpha - \frac{\dim \mathcal{D}_{h_i, \varepsilon_2}}{1 - l_1^i - l_2^i}.$$

From (11) and $\dim \mathcal{D}_{h_i, \varepsilon_2} \geq 0$, we get

$$\begin{aligned} \beta' &\geq \frac{\alpha - (d - i)\beta}{t - s} \\ &= \frac{\alpha(t - s + 1 - k + i)}{(d - k + t - s + 1)(t - s)}, \end{aligned}$$

so take $i = k - 1$, and

$$\beta' \geq \frac{\alpha}{d - k + t - s + 1}. \quad (13)$$

Combining (12) and (13), we get the minimum value of β' is that

$$\beta' = \frac{\alpha}{d - k + t - s + 1},$$

considering the minimum value β from (11), we get $\beta = \beta'$.

When $g = k$, $u_i = 1, i = 1, \dots, g - 1$, we thus the the upper bound

$$\mathcal{M}^s \leq \sum_{i=0}^{g-1} (1 - l_1^i - l_2^i)(\alpha - \frac{\dim \mathcal{D}_{n+i+1, \varepsilon_2}}{1 - l_1^i - l_2^i}).$$

Suppose R is composed of the set of nodes $\{n + j, j = 1, \dots, k - l_1 - l_2\}$, then we have

$$\mathcal{M}^s \leq \sum_{i=0}^{k-l_1-l_2-1} (\alpha - I(\mathbf{s}_i; d_{i, \varepsilon_2}))$$

thus, we get the bound (8).

At the minimum storage point, we can bound $\dim \mathcal{D}_{n+i+1, \varepsilon_2} \geq \beta' = \beta$, then the above bound becomes that of [8], that is

$$\mathcal{M}^s \leq (k - l_1^i - l_2^i)(\alpha - \beta).$$

When $g = 1, t \geq k$, then $u_0 = k, l_1^0 = l_1$ and $l_2^0 = l_2$. If $t - s + 1 < k$, then $J = \emptyset$ and $\alpha < d\beta$, the minimum value is $\alpha - \frac{\dim \mathcal{D}_{h_i, d_{\varepsilon_2}}}{k - l_1^i - l_2^i}$. If $t - s + 1 \geq k, \bar{J} = \emptyset$, the minimum value will still be $\alpha - \frac{\dim \mathcal{D}_{h_i, d_{\varepsilon_2}}}{k - l_1^i - l_2^i}$, and the bound is

$$\mathcal{M}^s \leq (k - l_1 - l_2)(\alpha - \frac{\dim \mathcal{D}_{h_i, d_{\varepsilon_2}}}{k - l_1^i - l_2^i}) = (k - l_1 - l_2)\alpha - \dim \mathcal{D}_{R, \varepsilon_2}.$$

We can also use the bound $\dim \mathcal{D}_{R, \varepsilon_2} \geq \beta' = \beta$, yielding

$$\mathcal{M}^s \leq (k - l_1 - l_2)\alpha - \beta.$$

When $1 < g < k$, since we want minimize the upper bound, we need to minimize the size of J . Then set $u_i = t, i = 0, \dots, g - 2, g = \lfloor k/t \rfloor + 1$, so that $u_{g-1} = k - \lfloor k/t \rfloor t$. For $u_i = t, t - s + 1 - u_i \leq 0$, the minimum value comes from α ; for u_{g-1} , no matter whether $t - s + 1 - k + \lfloor k/t \rfloor t \geq 0$, or $t - s + 1 - k + \lfloor k/t \rfloor t < 0$, the minimum value always comes from α . We then get the upper bound

$$\begin{aligned} H(\mathbf{o}^s) \leq \min_{l_1^i \leq t} \{ & \sum_{i=0}^{g-2} (t - l_1^i - l_2^i)(\alpha - \frac{\dim \mathcal{D}_{h_i, d_{\varepsilon_2}}}{t - l_1^i - l_2^i}) \\ & + (k - \lfloor k/t \rfloor t - l_1^{g-1} - l_2^{g-1})(\alpha - \frac{\dim \mathcal{D}_{h_{g-1}, d_{\varepsilon_2}}}{k - \lfloor k/t \rfloor t - l_1^{g-1} - l_2^{g-1}}) \}. \end{aligned}$$

Also, $\dim \mathcal{D}_{h_i, d_{\varepsilon_2}}, i = 0, \dots, g - 1$, can be bounded by $\beta' = \beta$, yielding

$$\mathcal{M}^s \leq (k - l_1 - l_2)\alpha - g\beta = (k - l_1 - l_2)\alpha - (\lfloor k/t \rfloor + 1)\beta.$$

Comparing the above three conditions, we get the tightest bound when $g = k, u_i = 1, i = 0, \dots, 1$, which gives the result, matching the bound of (9). $\square$

Note that for the fully collaborative case $s = 1$, and the values of β becomes

$$\beta = \beta' = \frac{\alpha}{d - k + t}$$

as proven in [8].

The total repair bandwidth, based on the above proposition is then

$$d\beta + (t - s)\beta' = (d + t - s)\beta = \frac{\alpha(d + t - s)}{d - k + t - s + 1}.$$

Corollary 3.1 [10] *For a partially collaborative code with no eavesdropper at the minimum storage point, we have*

$$\alpha = \frac{\mathcal{M}}{k}, \beta = \beta' = \frac{\mathcal{M}}{k} \frac{1}{d - k + t - s + 1},$$

for a total repair bandwidth of

$$\gamma = \frac{\mathcal{M}}{k} \frac{d + t - s}{d + t - k - s + 1}.$$

3.2 Discussion

From (8) in Proposition 2, we get that the upper bound on the secure file size $\mathcal{M}^s$ depends on $\sum_{i=1}^{k-l_1-l_2} I(\mathbf{s}_i; \mathbf{d}_{i,\varepsilon_2})$. This sum is involving each node whose content $\mathbf{s}_i$ is not eavesdropped ($k - l_1 - l_2$ of them), and data originating from each of these nodes but being eavesdropped when it is downloaded or exchanged in the set ε_2 of nodes. Therefore, for each $\mathbf{s}_i$, at most $(t - s)\beta'$ amount of data is involved in the collaboration process and at most $l_2\beta$ amount of data is downloaded later. For the same value t , and since $\beta = \beta'$, we thus have

$$I(\mathbf{s}_i; \mathbf{d}_{i,\varepsilon_2}) \leq (t - s)\beta' + l_2\beta = (t - s + l_2)\beta'.$$

Note that since $1 \leq s \leq t$, $t - s + l_2 \geq 0$, so apart if $t = s$ and $l_2 = 0$, we always have $t - s + l_2 \geq 1$ and thus for the range

$$\beta' \leq I(\mathbf{s}_i; \mathbf{d}_{i,\varepsilon_2}) \leq (t - s + l_2)\beta',$$

we get from Proposition 2 the bound (9), and the secure file size $\mathcal{M}^s$ is upper bounded by

$$\mathcal{M}^s \leq (k - l_1 - l_2)(\alpha - \beta) \tag{14}$$

with

$$\beta = \frac{\alpha}{d - k + t - s + 1}.$$

We have

$$\alpha - \beta = \alpha \left(1 - \frac{\alpha}{d - k + t - s + 1} \right) = \alpha \left(\frac{d - k + t - s}{d - k + t - s + 1} \right).$$

We consider fixed $\alpha, k - l_1 - l_2$. Then, since both denominators are positive:

$$\frac{d - k + t - s}{d - k + t - s + 1} \geq \frac{d - k + t - s_0}{d - k + t - s_0 + 1} \iff t - s \geq t - s_0.$$

Thus the secure file size is the largest when $t - s = t - 1$, that is, $s = 1$ (full collaboration), while it is the smallest when $t - s = 0$, that is $t = s$, and there is no collaboration.

4 Acknowledgements

This work was supported by the MoE Tier-2 grant eCode: erasure codes for data center environments (MOE2013-T2-1- 068).

References

- [1] K. W. Shum, “Cooperative Regenerating Codes for Distributed Storage Systems”, in the proceedings of the *2011 International Conference on Communications (ICC 2011)*.
- [2] A.-M. Kermarrec, N. Le Scouarnec, G. Straub, “Repairing Multiple Failures with Coordinated and Adaptive Regenerating Codes”, in the proceedings of the *2011 International Symposium on Network Coding (NetCod 2011)*.
- [3] K. W. Shum, Y. Hu, “Cooperative Regenerating Codes”, *IEEE Trans. on Information Theory*, vol. 59, no 11, July 2013.
- [4] F. Oggier, A. Datta. “Coding Techniques for Repairability in Networked Distributed Storage Systems”, in the series *Foundations and Trends in Communications and Information Theory*, vol. 9, no. 4, June 2013.
- [5] S. Pawar, S. El Rouayheb, K. Ramchandran, “On Secure Distributed Data Storage Under Repair Dynamics”, in the proceedings of the *2010 IEEE International Symposium on Information Theory (ISIT 2010)*, Austin.
- [6] N.B. Shah, K.V. Rashmi, P.V. Kumar, “Information Theoretically Secure Regenerating Codes for Distributed Storage”, in the proceedings of the *2011 IEEE Global Telecommunications Conference (GLOBECOM 2011)*, Houston.

- [7] Ankit Singh Rawat, O. Ozan Koyluoglu, Natalia Silberstein, and Sriram Vishwanath “Optimal Locally Repairable and Secure Codes for Distributed Storage Systems”, preprint, <http://arxiv.org/abs/1210.3664>.
- [8] Ankit Singh Rawat, O. Ozan Koyluoglu, Sriram Vishwanath “Secure Cooperative Regenerating Codes for Distributed Storage Systems”, *IEEE Transactions on Information Theory*, vol. 60, no 9, Sept. 2014.
- [9] F. Oggier, A. Datta, “Byzantine Fault Tolerance of Regenerating Codes”, in the proceedings of the *2011 IEEE International Conference on Peer-to-Peer Computing (P2P 2011)*.
- [10] S. Liu, F. Oggier, “On Storage Codes Allowing Partially Collaborative Repairs”, in the proceedings of the *2014 IEEE International Symposium on Information Theory (ISIT 2014)*.
- [11] S. Liu, F. Oggier, “Two Storage Code Constructions Allowing Partially Collaborative Repairs”, in the proceedings of the *2014 International Symposium on Information Theory and Applications (ISITA 2014)*.
- [12] A.G.Dimakis, P.B.Godfrey, Y.Wu, M.Wainwright, and K.Ramchandran, “Network Coding for Distributed Storage Systems”, *IEEE Transactions on Information Theory*, Vol. 56, Issue 9, Sept. 2010.