Optimal Secure Partial-Repair in Distributed Storage Systems

† Majid Gerami, † Ming Xiao, ‡ Somayeh Salimi, † Mikael Skoglund, and † Panos Papadimitratos
† School of Electrical Engineering, KTH, Royal Institute of Technology, Sweden
E-mail: {gerami, mingx, skoglund, papadim}@kth.se
‡ Department of Engineering Sciences, Signal and Systems, Uppsala University, Sweden
Email:somayeh.salimi@angstrom.uu.se

***Abstract*—Consider a distributed storage system where parts of the source file fragments in storage nodes are lost. We denote a storage node that lost a part of its fragments as a *faulty storage node* and a storage node that lost non of its fragment as a *complete storage node*. In a process, termed as *partial repair*, a set of storage nodes (among faulty and complete storage nodes) transmit repairing fragments to other faulty storage nodes to recover the lost fragments. We first investigate the optimal partial repair in which the required bandwidth for recovering the lost fragments is minimal. Next, we assume that an eavesdropper wiretaps a subset of links connecting storage nodes, and overhears a number of repairing fragments. We then study optimal secure partial-repair in which the partial-repair bandwidth is minimal and the eavesdropper obtains no information about the source file by overhearing the repairing fragments. We propose optimal secure codes for exact partial-repair in a special scenario.**

I. Introduction

Distributed storage systems include cloud storage systems, peer-to-peer storage systems and private/public data centers. In these systems, particularly in cloud storage systems, users can store, archive, or back up their data on (geographically) distributed storage nodes; DropBox, Google File Systems, and AmazonS3 are examples. The availability of stored files, *anywhere, anytime* is one of the main advantages of these systems. Yet, there are two main concerns: reliability and security (privacy).

Storage at each computing system (machine, or set of collocated machines, termed here as node) can suffer failures and most likely those would be partial. For example, for a single disk, a partition could fails [1]; or for a file server a subset of its disks could fail. This implies that a subset of data would be lost at each node. Encoding fragments of a file and storing them in storage nodes make storage systems more robust against data loss, particularly when storage nodes are unreliable. Maximum distance separable (MDS) codes provide the highest reliability against data loss. A file when coded by an (n, k)-MDS code is divided into k equal-sized blocks[1] which are then coded to n blocks such that any set of k blocks can reconstruct the whole stored file. Most of the existing studies provide codes considering node failure, where all data in storage is lost. Then, in a process, termed as the *repair process*, a new node in generated by the help of surviving nodes. The optimal repair-bandwidth has been well studied in [2], where the use of network coding has been proposed in the repair problem of distributed storage systems. After repair, the new node might contain different data compared to the failed node. But, it keep the property that every k storage nodes can reconstruct the source file. This repair is termed as the *functional repair*. In contrast, in exact repair the content of the new node is the same as the failed node. Exact repair have been studied in [3]. To further decrease the repair bandwidth, cooperative repair has been proposed in [4] when multiple node fails. In all these works, the repair has been studied when a storage node completely fails. Applying the existing repair methods for the case of partial loss might be suboptimal, as we will show it later in a motivating example. Recently, in [1], partial-MDS codes have been studied over the systems that parts of data in storage node are lost. However, in [1] the number of transmissions (bandwidth) in partial-repair has not been considered. In this paper, we focus on designing partial-MDS codes that require the minimum partial-repair bandwidth. Our proposed codes efficiently exploit the available side information in faulty nodes to achieves the optimal partial-repair. We also propose codes for exact partial-repair in a special scenario.

We then investigate a method to secure the bandwidth-optimal partial-repair codes. Security in the repair problem, when a node completely fails, has been studied in [5]. In a recent work, the authors in [6] studied security in partial repair in wireless caching networks in which storage nodes use broadcast channels in repair. The study in this paper differ from [6] in the senses that i) the studied network is a wireline network and there is no broadcast channels, and ii) the eavesdropper wiretaps a subset of links instead of overhearing all the repairing fragments. To provide security, we encode the source file prior to MDS encoding and show that the secrecy capacity (the maximum amount of data that can be stored in the system while preserving security in partial repair) can be achieved.

Motivating Example: Consider a partial repair process illustrated in Fig. 1, where a distributed storage system stores a file containing four fragments a_1, a_2, b_1, b_2 by an $(4, 2)-$MDS code over the finite field $\mathbb{F}_3$. Suppose that, fragments a_1+a_2 in node 3 and b_2 in node 4 are lost. To recover the lost fragments, we can consider the faulty nodes as completely failed node. Then by a method in [2], we must transmit the entire file (four fragments) to each faulty node. This requires eight fragment transmissions in total. A better approach, proposed in [4], is to allow the completely failed nodes to cooperate. In this case, six fragment transmissions in total are required (for details, please

[1] Here, a block may contain a number of equal-sized fragments of information.

978-1-5090-4780-2/17/$31.00 ©2017 IEEE

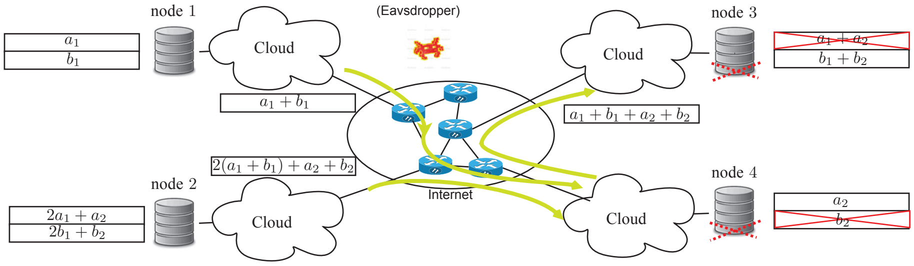

Fig. 1: A distributed cloud storage system. Parts of stored data are lost. The lost data is recovered in a partial repair process. Here, an eavesdropper is overhearing the communicated information between storage nodes during the partial repair process.

see [4]). Alternatively, one can consider each fragment in the system as a (virtual) node; then each lost fragment requires two fragment transmissions (e.g., for recovering b_2 in node 4, node 1 sends b_1 and node 2 sends $2b_1 + b_2$ to node 4). Thus, in total four fragment transmissions are required to recover all lost fragments. In our proposed approach, we efficiently exploit the available side information in the faulty nodes. Then, node 1 and 2 respectively send $a_1 + b_1$ and $2(a_1 + b_1) + a_2 + b_2$ toward node 4. Then, node 4 can recover its lost fragments by the received fragments and its side information, after performing simple operations over $\mathbb{F}_3$ as $2(a_1+b_1)+a_2+b_2-2(a_1+b_1)-a_2 = b_2$. Node 4 then sends coded fragment $a_1+b_1+a_2+b_2$ to the node 3. Then, node 3 recovers its lost fragment by an operation in $\mathbb{F}_3$ as $a_1 + b_1 + a_2 + b_2 - (b_1 + b_2) = a_1 + a_2$. Therefore, the lost fragments are recovered by three fragment transmissions. We show later that the proposed approach is optimal in terms of partial-repair bandwidth. To make the partial-repair secure, instead of b_1 and b_2, source file fragments, we substitute two symbols z_1 and z_2, which are randomly and uniformly selected from $\mathbb{F}_3$. Then the eavesdropper cannot obtain any information about the source by overhearing the repairing fragments. Thus, security is provided by the cost in reducing the storage space (here, to two fragments) for storing the source file.

The organization of the paper is as follows. In Section II, we formulate the optimal secure partial-repair problem. In Section III, we provide our main results. In Section IV, we present explicit code construction for exact optimal-secure partial-repair. Finally, we conclude the paper in Section V.

II. Problem Formulation

Notation: We use a bold lowercase letter to denote a column vector, and a bold uppercase letter to denote a matrix. Superscript T denotes matrix transpose. The set $[n]$ denotes $\{1, 2, \cdots, n\}$. $|\mathcal{P}|$ denotes the cardinality of set $\mathcal{P}$. For a random variable X, we denote $H(X)$ as the entropy of X. For a set $\mathcal{X} = \{X_1, X_2, \ldots, X_i\}$, we have $H(\mathcal{X}) = H(X_1, X_2, \ldots, X_i)$.

We consider a distributed storage system with n storage nodes; the source file encoded by an $(n, k)-$MDS code, that is, any set of k storage nodes can reconstruct the source file. Suppose that the source file contains M fragments, elements of $\mathbb{F}_q$, where q denotes the code alphabet size. Let us denote the source file by a column vector $\mathbf{s}$ of M elements. Then, a coded fragment, $\mathbf{x}$, is computed by $\mathbf{x} = \mathbf{g}^T\mathbf{s}$, where $\mathbf{g}$ is the global encoding vector [7] of $M \times 1$ dimension with elements from $\mathbb{F}_q$. A fragment, $\mathbf{x_1} = \mathbf{g_1}^T\mathbf{s}$, is innovative to a storage node if $\mathbf{g_1}$ is not in the span of the global encoding vectors of the fragments that already exist in the node. Two fragments, $\mathbf{x_1} = \mathbf{g_1}^T\mathbf{s}$ and $\mathbf{x_2} = \mathbf{g_2}^T\mathbf{s}$, are independent when their global encoding vectors $\mathbf{g_1}$ and $\mathbf{g_2}$ are independent.

When parts of the stored fragments in one or more storage nodes are lost and the faulty storage nodes have no access to the source file, storage nodes (including faulty and complete storage nodes) exchange information to recover the lost fragments. Note that in our model, storage nodes transmit coded fragments over error-free channels[2] to other storage nodes. The process of recovering the lost fragments is termed as the partial repair. Suppose that node i, $i \in [n]$, has access to set $\mathcal{P}_i = \{X_1, X_2, \ldots, X_{|\mathcal{P}_i|}\}$ of independent coded fragments; $|\mathcal{P}_i|$ is the amount of information that node i has access as side information. For a given $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n\}$, a fault, i.e., partial loss of fragments, results in $M/k - |\mathcal{P}_i|$ lost fragments in node i, for $i \in [n]$.

Clearly, to recover all the lost fragments in partial-repair process, the available information in the system must not be less than M. We formally state the necessary condition over a given loss pattern $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n\}$ as

$$H(\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n) \geq M. \tag{1}$$

Otherwise some information is permanently lost and thus the repair is not possible. In the rest of the paper, we assume (1) holds.

In partial repair, node i transmits β_{ij} fragments to storage node j for $i, j \in [n]$ and $i \neq j$. The transmission schedule in partial repair can be explicitly stated in the following two steps:

1) Complete storage nodes transmit fragments (functions of their stored fragments) to the faulty storage nodes;
2) Faulty storage nodes transmit fragments (functions of their stored fragments and their received fragments) to other faulty storage nodes.

We formally define the partial-repair bandwidth, Γ:

$$\Gamma \triangleq \sum_{i,j=1}^{n} \beta_{ij}. \tag{2}$$

[2] Error-free transmission can be achieved by complete channel coding or re-transmission. The impact of transmission errors to partial repair is beyond the scope of this paper.

We can characterize the necessary and sufficient conditions over β_{ij}s such that partial-repair is done successfully.

Definition 1: Consider a distributed storage system that stores a file coded by an (n, k)-MDS code. Suppose some fragments are lost and the available data on storage nodes is given by set $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n\}$. Define the region $R(\mathcal{P}) \subset \mathbb{R}^{n\times 1}$ to be set of all $\{\beta_{ij}\}$ satisfying

$$\sum_{j\in\mathcal{Q}} \sum_{i\in\mathcal{Q}^c} \beta_{ij} \geq M - \sum_{j\in\mathcal{Q}} |\mathcal{P}_j|, \tag{3}$$

for every $\mathcal{Q} \subset [n]$ and $|\mathcal{Q}| = k$ storage nodes. In next section, we formally state that $R(\mathcal{P})$ is the feasible region for successful partial-repair.

We investigate security in partial repair. Let $Y_1, Y_2, \ldots, Y_\Gamma$ denote the random variables representing Γ transmitted repairing-fragments. Now assume that there is an eavesdropper who overhears a subset $\mathcal{E}$ of links and has access to $\mu \leq |\mathcal{E}|$ repairing fragments with independent encoding vectors. We aim to design bandwidth-optimal codes in the repair problem at hand such that there is no leakage of information to the eavesdropper. This is formally defined, as follows.

Definition 2: Consider a distributed storage system in which a source file is distributed among n storage nodes. Let the source file be denoted by a set $\mathcal{S}$ which contains $|\mathcal{S}|$ fragments, i.e., $\mathcal{S} = \{s_1, s_2, \ldots, s_{|\mathcal{S}|}\}$. Assume an eavesdropper has access to a subset of links carrying fragments $\mathcal{E} = \{e_1, \ldots, e_{|\mathcal{E}|}\}$. The code is secure, if

$$H(\mathcal{S}|\mathcal{E}) = H(\mathcal{S}) \tag{4}$$

A fundamental question is that how much is the maximum amount of information that can be stored in the storage system such that an eavesdropper obtains no information about the source file by overhearing the repairing fragments in a partial-repair process. More formally, suppose that we use an (n, k)-MDS code for storing the source file in the storage nodes, then a set of k nodes, which is denoted by $\mathcal{D}$, contains M independent fragments. For security in partial repair, we may store some random symbols taking uniformly from $\mathbb{F}_q$. This random variables occupy some space in storage nodes and thereby the space for storing the source file is reduced. Hence, every k storage nodes may have smaller than M fragments of data from the source. That is $|\mathcal{S}| \leq M$, and $M - |\mathcal{S}|$ random symbols is stored in every set of k storage nodes for providing security in partial repair. The eavesdropper overhearing fragments $Y_1, \ldots, Y_{|\mathcal{E}|}$ obtains no information about the source if

$$H(\mathcal{S}|Y_1, \ldots, Y_{|\mathcal{E}|}) = H(\mathcal{S}). \tag{5}$$

Since every k nodes should be able to reconstruct the source file, we have

$$H(\mathcal{S}|\mathcal{D}) = 0, \text{ for } \forall\mathcal{D} \subset [n], |\mathcal{D}| = k. \tag{6}$$

We may refer to this as the complete reconstruction condition. We formally define the secrecy capacity (which is here denoted as C_{ss}) as

$$\begin{aligned} C_{ss} \triangleq \max \quad & H(\mathcal{S}), \\ \text{subject to:} \quad & H(\mathcal{S}|Y_1, \ldots, Y_{|\mathcal{E}|}) = H(\mathcal{S}), \\ & H(\mathcal{S}|\mathcal{D}) = 0, \text{ for } \forall\mathcal{D} \subset [n], |\mathcal{D}| = k. \end{aligned} \tag{7}$$

We shall derive the secrecy capacity in the next section.

III. Main Results

We first derive the minimum required bandwidth in partial-repair. Then, we derive the secrecy capacity for the bandwidth-optimal partial-repair. We note that the results in this section is valid for functional partial-repair. Since, exact partial-repair has more constraints than functional partial-repair, then the minimum bandwidth for functional partial-repair serves as a lower bound for exact partial-repair. Based on the same argument, the secrecy capacity derived for functional repair serves as an upper bound for exact partial-repair. For the ease of notation, we drop the term functional whenever it is clear that we are analysing functional partial-repair. The following theorem states the necessary and sufficient conditions for partial repair.

Theorem 1: Consider a distributed storage system using an (n, k)-MDS code. Suppose that the storage nodes have access to parts of their stored data based on $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n\}$. For given β_{ij}'s, then all the lost fragments can be recovered if and only if $\{\beta_{ij}\} \in R(\mathcal{P})$.

The proof is based on formulating the partial-repair problem into a multicast problem. The detail proof is provided in Appendix A of the extended version [8]. Next, we will apply this theorem on our motivating example in Fig. 2.

Example 1: Examining Theorem 1 on the four-node storage network in motivating example provides us the following conditions for partial repair:

$\mathcal{S}$	induced constraint
$\{1, 3\}$	$\beta_{23} + \beta_{43} \geq 1$
$\{1, 4\}$	$\beta_{24} + \beta_{34} \geq 1$
$\{3, 4\}$	$\beta_{13} + \beta_{23} + \beta_{14} + \beta_{24} \geq 2$
$\{2, 3\}$	$\beta_{43} + \beta_{13} \geq 1$
$\{2, 4\}$	$\beta_{34} + \beta_{14} \geq 1$

Summing both sides of the above inequalities gives $2(\beta_{43} + \beta_{34} + \beta_{23} + \beta_{13} + \beta_{14} + \beta_{24}) \geq 6$. This turns out that $\Gamma = \beta_{43} + \beta_{34} + \beta_{23} + \beta_{13} + \beta_{14} + \beta_{24} \geq 3$, and thus the minimum total number of fragment transmissions for partial repair is 3. Thus the code presented in Fig. 1 is optimal.

Example 2: In a more general case than the previous example, consider a distributed storage system where a file of size M is encoded by an $(n, k)-$MDS code. Suppose that $n - k$ storage nodes have equally lost ξ number of fragments of their stored fragments, where $0 \leq \xi \leq M/k$. We assume that there are always k complete storage nodes in the system. This assumption assures us that the file availability condition in (1) is always satisfied, for any value of ξ, e.g., for $\xi = M/k$. Without loss of generality, we assume that nodes $1, 2, \ldots, k$ are complete storage nodes and the other $(n-k)$ storage nodes are faulty storage nodes. That is, $|\mathcal{P}_i| = M/k$, for $i \in \{1, \ldots, k\}$, and $|\mathcal{P}_i| = M/k - \xi$, for $i \in \{k + 1, \ldots, n\}$. The following corollary states a lower bound on the required total number of fragment transmissions (Γ) for the partial-repair.

Corollary 1: A lower bound on the required total number of fragment transmissions for partial-repair in the above problem is

$$\Gamma \geq (n - 1)\xi \tag{8}$$

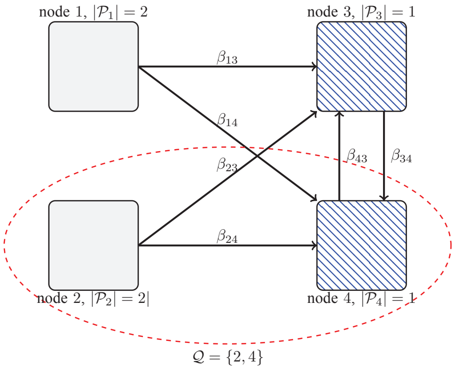
node 1, $|\mathcal{P}_1| = 2$
node 3, $|\mathcal{P}_3| = 1$
β_{13}
β_{14}
β_{23}
β_{43}
β_{34}
β_{24}
node 2, $|\mathcal{P}_2| = 2|$
node 4, $|\mathcal{P}_4| = 1$
$\mathcal{Q} = \{2, 4\}$

Fig. 2: Optimal partial repair.

Proof: The proof is provided in Appendix B of the extended version [8]. ■

In general, we aim to minimize the total number of fragment transmissions, $\Gamma = \sum_{i,j=1}^{n} \beta_{ij}$, for the points in the feasible region. Formally, we aim to

$$\begin{array}{ll} \min\limits_{\beta_{ij}} & \Gamma = \sum_{i,j=1}^{n} \beta_{ij} \\ \text{subject to:} & \sum_{j \in Q} \sum_{i \in Q^c} \beta_{ij} \geq M - \sum_{i \in Q} |\mathcal{P}_i|, \\ & \mathcal{Q} \subset [n], |\mathcal{Q}| = k, \\ & \beta_{ij} \in \mathbb{R}^+, \end{array} \tag{9}$$

where $\mathbb{R}^+$ is the set of non-negative real numbers. This problem is a linear programming problem and can be efficiently solved [9].

For a given loss pattern $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n\}$, we can derive the secrecy capacity, as follows.

Theorem 2: Suppose an (n, k)-MDS-coded distributed storage system has the capacity of storing M fragments. Suppose that some fragments in storage nodes are lost and storage node i, for $i \in [n]$, has access to $|\mathcal{P}_i|$ $(0 \leq |\mathcal{P}_i| \leq M/k)$ fragments. If an eavesdropper overhears a subset $\mathcal{E}$ of links with rank $\mu \leq |\mathcal{E}|$ repairing fragments (that is, the eavesdropper overhears μ fragments with independent encoding vectors). Then the secrecy capacity is

$$C_{ss} = \max\{M - \mu, 0\}. \tag{10}$$

Proof: The proof is provided in Appendix C of the extended version [8]. ■

Examining Theorem 2 over Example 1 where an eavesdropper overhears three repairing fragments gives that $C_{ss} = 4 - 2 = 2$, noting that the number of independent fragments overheard by the eavesdropper is $\mu = 2$. In the next section, we propose an explicit code construction for the exact partial repair in which the repair bandwidth is minimal and the partial repair is secure.

IV. Optimal and Secure Codes for Exact Partial-Repair

In this section, we provide an explicit code construction for the exact partial-repair that achieves the lower bound in Corollary 1. We note that the lower bound derived in Corollary 1 is valid for functional and exact partial-repair, due to the fact that exact partial-repair is a specific case of functional partial-repair. This implies that the proposed code, which achieves the lower bound, is optimal in partial-repair bandwidth. Next, we construct the secure code by encoding the source file fragments prior to this bandwidth-optimal code.

We first divide the source file into $M = k^2$ fragments[3], meaning that each node stores $M/k = k$ fragments before data loss. We construct a $k \times k$-matrix $\mathbf{S}$ which contain the source file fragments as

$$\mathbf{S} = \begin{pmatrix} s_{11} & \dots & s_{1k} \\ s_{21} & \dots & s_{2k} \\ \vdots & \ddots & \vdots \\ s_{k1} & \dots & s_{kk} \end{pmatrix}. \tag{11}$$

We assume there are k nodes that store the uncoded fragments. These nodes are termed as *systematic nodes*. In addition, there are $(n-k)$ *parity nodes* that store coded fragments. Without loss of generality, assume that nodes labeled as $1, \dots, k$ are systematic nodes and nodes $k+1, \dots, n$ are parity nodes. We store in k systematic nodes the symbols in rows of matrix $\mathbf{S}$. That is, the i-th systematic node stores k fragments in the i-th row of matrix $\mathbf{S}$. Next, we store coded fragments in $n-k$ parity nodes. To get the coded fragments in parity nodes, we construct matrix $\mathbf{P}$ as

$$\mathbf{P} = \Phi \mathbf{S}, \tag{12}$$

where $\mathbf{\Phi}$ is a $(n-k) \times k$-dimensional Cauchy matrix [10], with elements from a finite field $\mathbb{F}_q$, for $q > n$. We store in node i, for $i = k+1, \dots, n$ the symbols in the $(i-k)$-th row of matrix $\mathbf{P}$. If $\mathbf{P_i}$ denotes the vector in the i-th row of matrix $\mathbf{P}$, then the coded fragments in node i are elements of vector $\mathbf{P}_{i-k}$ as

$$\mathbf{P_{i-k}} = \Phi_{i-k} \mathbf{S}, \text{ for } i = k+1, \dots, n. \tag{13}$$

Here, Φ_i denotes the $i-$th row of matrix Φ. By this construction, the code on the storage nodes is an (n, k)-MDS code.

Proposition 1: The above code is an (n, k)-MDS code.

Proof: The proof is provided in Appendix D of the extended version [8]. ■

Now, we describe the process of exact partial-repair. There are $n-k$ faulty storage nodes, each of which has lost ξ fragments. Let us first assume $n-k$ faulty storage nodes are parity nodes, and thereby, k complete storage nodes are systematic nodes. Let $\mathbf{P}_{ij}$ denotes the element in row i and column j of matrix $\mathbf{P}$. The partial repair proceeds as the following steps:

Step 1) Systematic node i transmits fragments $v^1_{i,k+1}, \dots, v^u_{i,k+1}, \dots v^\xi_{i,k+1}$ to node $k+1$, where

$$v^u_{(i,k+1)} = \mathbf{S}_i \mathbf{b}_u \text{ for } u = 1, \dots, \xi. \tag{14}$$

Here, $\mathbf{b}_1, \dots, \mathbf{b}_\xi$ are rows of a $\xi \times k$ Cauchy matrix, where each element is taken from $\mathbb{F}_q$. This Cauchy matrix requires $q \geq k + \xi$. This step runs for all $i \in \{1, \dots, k\}$.

[3] For this, we should properly design the fragment size. For example if $M = 1$ mega-bits, and $q = 2, k = 4$, the fragment size=$\lceil 2^{20}/4^2 = \rceil = 2^{16}$ bits.

Step 2) In node $k+1$, if fragments $\mathbf{P}_{1j_1}, \ldots, \mathbf{P}_{1j_\xi}$ are lost, then the node calculates

$$\mathbf{\Phi_1} \underbrace{\begin{pmatrix} v^1_{(1,k+1)} & v^2_{(1,k+1)} & \cdots & v^\xi_{(1,k+1)} \\ v^1_{(2,k+1)} & v^2_{(2,k+1)} & \cdots & v^\xi_{(2,k+1)} \\ \vdots & \vdots & \cdots & \vdots \\ v^1_{(k,k+1)} & v^2_{(k,k+1)} & \cdots & v^\xi_{(k,k+1)} \end{pmatrix}}_{\mathbf{Z}} \tag{15}$$

and obtains

$$\mathbf{\Phi_1 Z} =$$

$$\underbrace{\left(\mathbf{\Phi_1 S}^T_{j_1} \ldots \mathbf{\Phi_1 S}^T_{j_\xi}\right) \underbrace{\begin{pmatrix} b^{(1)}_{j_1} & \cdots & b^{(1)}_{j_\xi} \\ \vdots & \ddots & \vdots \\ b^{(\xi)}_{j_1} & \cdots & b^{(\xi)}_{j_\xi} \end{pmatrix}}_{\mathbf{B}}}_{\text{the desired term}}$$

$$\underbrace{+ \left(\mathbf{\Phi_1 S}^T_{r_1} \ldots \mathbf{\Phi_1 S}^T_{r_{k-\xi}}\right)_{r_i \notin 1,\ldots,\xi} \begin{pmatrix} [b^{(1)}_{j_l}]_{l \notin 1,\ldots,\xi} \\ \vdots \\ [b^{(\xi)}_{j_l}]_{l \notin 1,\ldots,\xi} \end{pmatrix}}_{\text{the interfering terms}}. \tag{16}$$

Node $k+1$ cancels the interfering terms in (16) by using its side information. Then the lost fragments can be retrieved since matrix $\mathbf{B}$ is invertible (due to the fact that any square submatrix of a Cauchy matrix is invertible).

Step 3) In the next step, node $k+1$ calculates

$$\mathbf{\Phi_{i-k} Z} \tag{17}$$

and sends ξ elements of the above vector to node i, for all $i \in \{k+2, \ldots, n\}$.

Step 4) Suppose fragments j^i_r for $r \in \{1, \ldots, \xi\}$ are lost in node i. Parity node i recovers its lost fragments similar to node $k+1$ by removing the interfering terms. This operation is performed for all $i \in \{k+2, \ldots, n\}$.

Step 5) If there are systematic nodes among faulty nodes, then we first change the variables such that again we have k systematic nodes and $n-k$ parity nodes. Then we proceed through steps (1)-(4).

Proposition 2: The proposed code is optimal in the bandwidth for partial repair.

Proof: For the recovery of the lost fragments, we transmit $k\xi$ fragments in Step 1 and $(n-k-1)\xi$ fragments in Step 3. Thus, in total, we transmit $k\xi + (n-k-1)\xi = (n-1)\xi$ fragments, meaning that the proposed code achieves the lower bound in Corollary 1. Thereby, it is optimal. ■

In the distributed storage system studied in this section, assume that the eavesdropper overhears $\Gamma = (n-1)\xi$ transmitted repairing fragments. When the above code is used in partial repair, then the number of independent fragments that the eavesdropper can access is $\mu = k\xi$ (note that the transmitted fragments in partial repair are linear combination of $k\xi$ transmitted fragments in Step 1). Hence, for the exact partial-repair, we have the upper bound as $C_{ss} \leq \max\{M - k\xi, 0\}$. We present the codes that achieve the upper bound. For that, we can precode the source code prior to MDS encoding. A simple precoding can be applied for this specific case by substituting $k\xi$ random symbol taking uniformly from $\mathbb{F}_q$ in the source matrix as the following

$$\mathbf{S} =$$

$$\begin{pmatrix} s_{11} & \cdots & s_{1(k-\xi)} & z_1 & \cdots & z_{k\xi-k+1} \\ s_{21} & \cdots & s_{2(k-\xi)} & z_2 & \cdots & z_{k\xi-k+2} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ s_{k1} & \cdots & s_{k(k-\xi)} & z_k & \cdots & z_{k\xi} \end{pmatrix}. \tag{18}$$

The partial repair process remains the same as before.

Proposition 3: The above code is optimal secure code, achieving secrecy capacity $C_{ss} = \max\{M - k\xi, 0\}$.

Proof: The proof is provided in Appendix E of the extended version [8]. ■

The proposed code requires the code alphabet size to be greater than $\max\{k+\xi, n\}$. This implies that the code is quite simple for implementation.

V. Conclusion

We studied optimal bandwidth partial-repair in distributed storage systems. We investigated the security of partial repair where an eavesdropper has access to a subset of repairing fragments, and derived the secrecy capacity of the system. We derived the minimum required bandwidth for partial repair. In a scenario, we showed that the optimal bandwidth is achievable for exact partial-repair, and then we made this exact partial repair secure. In future, we study security in partial repair where the storage nodes store the file by non-MDS codes. We also studty weakly secure codes for partial repair.

References

[1] M. Blaum, J. L. Hafner, and S. Hetzler, “Partial-mds codes and their application to raid type of architectures,” *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4510–4519, 2013.

[2] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[3] K. Rashmi, N. B. Shah, K. Ramchandran, and P. Kumar, “Regenerating codes for errors and erasures in distributed storage,” in *Proc. IEEE Symp. Inf. Theory*, 2012, pp. 1202–1206.

[4] Y. Hu, Y. Xu, X. Wang, C. Zhan, and P. Li, “Cooperative recovery of distributed storage systems from multiple losses with network coding,” *IEEE J. Sel. Area*, vol. 28, no. 2, pp. 268–276, 2010.

[5] S. Pawar, S. El Rouayheb, and K. Ramchandran, “Securing dynamic distributed storage systems against eavesdropping and adversarial attacks,” *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6734–6753, 2011.

[6] M. Gerami, M. Xiao, S. Salimi, and M. Skoglund, “Secure partial repair in wireless caching networks with broadcast channels,” in *Proc. IEEE Conf. on Communications and Network Security (CNS)*, 2015, pp. 353–360.

[7] R. W. Yeung, *Information theory and network coding*. Springer, 2008.

[8] M. Gerami, M. Xiao, S. Salimi, M. Skoglund, and P. Papadimitratos, “Optimal secure partial repair in distributed storage systems,” *Extended version, Available at https://www.kth.se/profile/gerami/.*

[9] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2009.

[10] G. Heinig, “Inversion of generalized cauchy matrices and other classes of structured matrices,” in *Linear algebra for signal processing*. Springer, 1995, pp. 63–81.