Capacity and Security of Heterogeneous Distributed Storage Systems

Toni Ernvall, Salim El Rouayheb, *Member, IEEE*,
Camilla Hollanti, *Member, IEEE*, and H. Vincent Poor, *Fellow, IEEE*

***Abstract*—The capacity of *heterogeneous* distributed storage systems under repair dynamics is studied. Examples of these systems include peer-to-peer storage clouds, wireless, and Internet caching systems. Nodes in a heterogeneous system can have different storage capacities and different repair bandwidths. Lower and upper bounds on the system capacity are given. These bounds depend on either the average resources per node, or on a detailed knowledge of the node characteristics. Moreover, the case in which nodes may be compromised by an adversary (passive or active) is addressed and bounds on the secure capacity of the system are derived. One implication of these new results is that symmetric repair maximizes the capacity of a homogeneous system, which justifies the model widely used in the literature.**

***Index Terms*—Distributed storage systems, information theoretic security, repair bandwidth, regenerating codes, network coding.**

I. Introduction

Cloud storage has emerged in recent years as an inexpensive and scalable solution for storing large amounts of data and making it pervasively available to users. The growing success of cloud storage has been accompanied by new advances in the theory of erasure codes for such systems, namely the application of network coding techniques for distributed data storage and the theory of regenerating codes introduced by Dimakis *et al.* [1], followed by a large body of further work in the literature.

Cloud storage systems are typically built using a large number of inexpensive commodity disks that fail frequently, making failures "the norm rather then the exception" [2]. Therefore, it is a prime concern to achieve fault-tolerance in these systems and minimize the probability of losing the stored data. The recent theoretical results uncovered fundamental tradeoffs among system resources (storage capacity, repair bandwidth, etc.) that are necessary to achieve fault-tolerance. They also provided novel erasure codes constructions that can achieve some of these tradeoff points; see for example [3], [4] and [5]. Recent work describes the implementation of erasure codes in distributed storage systems at the production level at Microsoft [6] and Facebook [7].

The majority of the results in the literature of this field focus on a homogeneous model when studying the information theoretic limits on the performance of distributed storage systems. In a homogeneous system all the nodes (hard disks or other storage devices) have the same parameters (storage capacity, repair bandwidth, etc.). This model encompasses many real-world storage systems such as clusters in a data center, and has been instrumental in forming the engineering intuition for understanding these systems. Recent development have included the emergence of *heterogeneous* systems that pool together nodes from different sources and with different characteristics to form one big reliable cloud storage system. Examples include peer-to-peer (p2p), or hybrid (p2p-assisted) cloud storage systems [8], [9], Internet caching systems for video-on-demand applications [10], [11], and caching systems in heterogeneous wireless networks [12]. Motivated by these applications, we study the capacity of heterogeneous distributed storage systems (DSS) here under reliability and secrecy constraints.

Contributions: The capacity of a DSS is the maximum amount of information that can be delivered to any user contacting k out of n nodes in the system. Intuitively, in a heterogeneous system, this capacity should be limited by the "weakest" nodes. However, nodes can have different storage capacities and different repair bandwidths. And the tension between these two set of parameters makes it challenging to identify which nodes are the "weakest".

Our first result establishes an upper bound on the capacity of a DSS that depends on the average resources in the system (average storage capacity and average repair bandwidth per node). We use this bound to prove that symmetric repair, *i.e.*, downloading equal amount of data from each helper node, maximizes the capacity of a homogeneous DSS. While the optimality of symmetric repair is known for the special case of MDS codes [13], our results assert that symmetric repair is always optimal for any choice of system parameters. Further, our proof avoids the combinatorial cut-based arguments typically used this context.

In addition, we give an expression for the capacity when we know the characteristics of all the nodes in the system (not just the averages). This expression may be hard to compute, but we use it to derive additional bounds that are easy to evaluate. Our techniques generalize to the scenario in which the system is compromised by an adversary. We consider two types of

T. Ernvall is with the Turku Centre for Computer Science (TUCS) & the Department of Mathematics and Statistics, University of Turku, Finland, e-mail: tmernv@utu.fi. T. Ernvall's reserach is supported by the TUCS and by the Academy of Finland under Grant #131745.

S. El Rouayheb and H. V. Poor are with the Department of Electrical Engineering, Princeton University, e-mails: {salim, poor}@princeton.edu. Their research is supported by the U. S. National Science Foundation under Grant CCF-1016671.

C. Hollanti is with the Department of Mathematics and Systems Analysis, Aalto University, Finland, e-mail: camilla.hollanti@aalto.fi. C. Hollanti's research is supported by the Academy of Finland under Grant #131745 and by the Emil Aaltonen Foundation, Finland.

Manuscript received December 3, 2012. Revised manuscript received May 28, 2013.

adversaries: a passive adversary who can only eavesdrop on certain nodes in the system, and an active malicious adversary who can also change the stored data on some nodes. For these cases, we give bounds on the secure capacity of the system and show that symmetric repair maximizes the secrecy capacity of a homogeneous system.

Parts of this paper have been presented at the 2013 Information Theory and Applications Workshop (invited presentation "Data security in heterogeneous distributed storage systems"), San Diego, California, USA, and at the ISIT 2013 [14].

Related work: Wu proved the optimality of symmetric repair in [13] for the special case of a DSS using Maximum Distance Separable (MDS) codes. Coding schemes for a non-homogeneous storage system with one super-node that is more reliable and has more storage capacity were studied in [15]. References [16] and [17] studied the problem of storage allocations in distributed systems under a total storage budget constraint where nodes can fail with different probabilities. Shah et al. studied in [18] constructions of "flexible" regenerating codes for systems that allow flexiblity in the amount of data downloaded for repair from each helper node as long as the total does not exceed a given budget. Pawar *et al.* [19], [20] studied the secure capacity of homogeneous distributed storage systems under eavesdropping and malicious attacks.

Organization: Our paper is organized as follows. In Section II, we describe our model for heterogeneous DSS and set up the notation. In Section III, we summarize our main results. In Section IV, we prove our bounds on the capacity of a heterogeneous DSS. In Section V, we study the secure capacity in the presence of an adversary. We conclude in Section VII and discuss some open problems.

II. Model

A. System Model

A heterogeneous distributed storage system is formed of n storage nodes $v_1, \ldots, v_n$ with storage capacities $\alpha_1, \ldots, \alpha_n$ respectively. Unless stated otherwise, we assume that the nodes are indexed in increasing order of capacity, *i.e.*, $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n$. In a homogeneous system all nodes have the same storage capacity α, *i.e.*, $\alpha_i = \alpha, \forall i$. As a reliability requirement, a user should be able to obtain a file by contacting any $k < n$ nodes in the DSS. The nodes forming the system are unreliable and can fail. The system is *repaired* from a failure by replacing the failed node with a new node. Upon joining the system, the new node downloads its data from d, $k \leq d \leq n-1$, helper nodes in the system.

The repair process can be either *exact* or *functional*. In the case of exact repair, the new node is required to store an exact copy of the data that was stored on the failed node. Whereas in the case of functional repair, the data stored on the new node does not have to be an exact copy of the lost data, but merely "functionally equivalent" in the sense that it preserves the property that contacting any k out of n nodes is sufficient to reconstruct a stored file. We focus on functional repair in this paper, although some of our results do generalize to the exact repair model (see the discussion in Appendix A).

An important system parameter is the *repair bandwidth* which refers to the total amount of data downloaded by the new node. In a homogeneous system, the repair bandwidth, denoted by γ, is the same for any new node joining the system. The typical model adopted in the literature assumes *symmetric repair* in which the total repair bandwidth γ is divided equally among the d helpers. Thus, the new node downloads $\beta = \gamma/d$ amount of information from each helper. In a heterogeneous system the repair bandwidth can vary depending on which node has failed and which nodes are helping in the repair process. We denote by β_{ijS} the amount of information that a new node replacing the failed node v_j is downloading from helper node v_i when the other helper node belong to the index set S ($i \in S, |S| = d$). An important special case is when the repair bandwidth per helper depends only on the identity of the helper node and not on the identity of the failed node or the other helpers. In this case, we say that helper node v_i has repair bandwidth β_i, *i.e.*, $\beta_{ijS} = \beta_i, \forall j, S$. In the case of a homogeneous system with symmetric repair, we have $\beta_{ijS} = \beta = \gamma/d, \forall i, j, S$.

We focus on repair from single node failures since it is the dominant failure pattern in DSSs [6]. Moreover, in the rare event of multiple simultaneous failures, the failed nodes can be repaired successively by invoking the repair scheme for single failures.

When a single node fails, there are $\binom{n-1}{d}$ possibilities for the set of helpers S. Therefore, the average repair bandwidth γ_j of node v_j is

$$\gamma_j = \binom{n-1}{d}^{-1} \sum_{\substack{S: j \notin S \\ |S|=d}} \sum_{i \in S} \beta_{ijS}. \tag{1}$$

We denote by $\overline{\gamma} = \frac{1}{n}\sum_{j=1}^{n} \gamma_j$ and $\overline{\alpha} = \frac{1}{n}\sum_{j=1}^{n} \alpha_j$ the average total repair bandwidth and average node capacity in the DSS, respectively.

We are interested in finding the capacity C of a heterogeneous DSS.

Definition: The capacity C of a DSS represents the maximum amount of information that can be downloaded by any user contacting k out of n nodes in the system.

It follows directly from the definition above that the capacity C^{ho} of a homogeneous DSS satisfies $C \leq k \cdot \alpha$. Recall from [1], that the capacity C^{ho} has the following expression for DSS with symmetric repair:

$$C^{ho}(\alpha, \gamma) = \sum_{i=1}^{k} \min\left\{\alpha, (d-i+1)\frac{\gamma}{d}\right\}. \tag{2}$$

B. Adversary Model

We are also interested in characterizing the secure capacity of the system when nodes are compromised by an adversary. The adversary can eavesdrop on some storage nodes, and possibly corrupt a subset of the stored data. We follow closely the adversary model in [19] and [20] and denote by ℓ the number of nodes that the intruder can eavesdrop on, and b the number of nodes it can control by maliciously corrupting its data. We study three types of adversaries:

a) Passive Eavesdropper: The eavesdropper can read the data downloaded during repair and stored on ℓ, $\ell < k$, compromised nodes, but cannot change the stored data ($b = 0$). We are interested here in information theoretic secrecy which characterizes the fundamental ability of the system to provide data confidentiality independently of cryptographic methods. The *secrecy capacity* of the system, denoted by C_s, is defined as the maximum amount of information that can be delivered to a user without revealing any information to the eavesdropper (perfect secrecy).

b) Active Omniscient Adversary: The active omniscient adversary knows the file stored in the system and can control b nodes in total, where $2b < k$. In this case, the adversary can maliciously corrupt the data stored on the nodes under his control and can send corrupted messages when contacted for repair or for file download.

c) Active Limited-knowledge Adversary: Here, the active adversary has *limited knowledge* about the data stored in the system. He can eavesdrop on $\ell < k$ nodes in the system, and among these nodes he can corrupt the data on $b \leq \ell$ of them. We assume that the number ℓ is not sufficient enough to let the adversary guess the stored file.

In the case of an active adversary, we are interested in computing the *resiliency capacity* of the system, *i.e.,* the maximum file size that can be stored on the DSS such that the user can still decode with no-errors despite the actions of the malicious adversary. When the type of the adversary is not specified, we use the term secure capacity to refer to the secrecy capacity or resiliency capacity of the system. Note that we always assume that the adversary has a complete knowledge of the code and the repair scheme implemented in the system.

III. Main Results

We start by summarizing our results. Theorem 1 gives a general upper bound on the storage capacity of a heterogeneous DSS as a function of the average resources per node.

Theorem 1: The capacity C of a heterogeneous distributed storage system, with node average capacity $\bar{\alpha}$ and average repair bandwidth $\bar{\gamma}$, is upper bounded by

$$C \leq \sum_{i=1}^{k} \min\left\{\bar{\alpha}, (d-i+1)\frac{\bar{\gamma}}{d}\right\} = C^{ho}(\bar{\alpha}, \bar{\gamma}). \qquad (3)$$

The right-hand side term in (3) is the capacity of a homogeneous DSS as in (2) in which all nodes have storage $\alpha = \bar{\alpha}$ and total repair bandwidth $\gamma = \bar{\gamma}$. Th. 1 states that the capacity of a DSS cannot exceed that of a homogeneous system where the total system resources are split equally among all the nodes. Also, Th. 1 implies that *symmetric repair is optimal* in homogeneous systems in the sense that it maximizes the system capacity. This justifies the repair model adopted in the literature. This result is stated formally in Cor. 2. While the optimality of symmetric repair is known for the special case of MDS codes [13], Cor. 2 asserts that symmetric repair is always optimal for any choice of system parameters. This result follows directly from Th. 1 and avoids the combinatorial cut-based arguments that may be needed in a direct proof [1].

Corollary 2: In a homogeneous DSS with node capacity α and total repair bandwidth γ, symmetric repair maximizes the system capacity.

When we know the parameters of the nodes in the system beyond the averages, we can obtain possibly tighter bounds as described in Th. 3. To simplify the notation, let us order the repair bandwidth per helper β_{ijS} into an increasing sequence $\beta'_1, \beta'_2, \ldots, \beta'_m$, such that $\beta'_l \leq \beta'_{l+1}$ and where $m = nd\binom{n-1}{d}$. Also, recall that $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n$.

Theorem 3: The capacity C of heterogeneous DSS is bounded by

$$C_{\min} \leq C \leq C_{\max}$$

where

$$C_{\min} = \min_{l=0,\ldots,k} \left(\sum_{j=1}^{l} \alpha_j + \sum_{j=1}^{h} \beta'_j \right),$$

$$C_{\max} = \min_{l=0,\ldots,k} \left(\sum_{j=1}^{l} \alpha_j + \sum_{j=1}^{h} \beta'_{m+1-j} \right),$$

and

$$h = \frac{(2d-k-l+1)(k-l)}{2}.$$

When the system is compromised by an adversary, whether passive or active, the system secure capacity can be upper bounded as stated in the next theorem.

Theorem 4: The secure capacity of a heterogeneous DSS under a passive or an active (omniscient or limited-knowledge) adversarial attack, is upper bounded by the secure capacity of a homogeneous system under the same attack and having node average capacity $\bar{\alpha}$ and average repair bandwidth $\bar{\gamma}$ (See more details in Table I).

This theorem implies that symmetric repair also maximizes the secure capacity of a homogeneous DSS.

IV. Capacity of Heterogeneous DSS

A. Example & Proof of Theorem 1

We illustrate the proof of Th. 1 through an example for the special case in which the bandwidths depend only on identity of the helper node. We compute the capacity of the DSS for this specific example, and show that it is strictly less than the upper bound of Th. 1. That is, it does not achieve the capacity of a homogenous system with the same average characteristics. More specifically, consider the heterogeneous DSS depicted in Fig. 1(a) with $(n, k, d) = (3, 2, 2)$ formed of 3 storage nodes v_1, v_2 and v_3 with storage capacities $(\alpha_1, \alpha_2, \alpha_3) = (1, 2, 2)$ and repair bandwidths $(\beta_1, \beta_2, \beta_3) = (1, 2, 2)$. The average node capacity $\overline{\alpha} = 5/3$ and repair bandwidth are $\overline{\beta} = 10/3$. Th. 1 gives that the capacity of this DSS $C \leq 10/3 = 3.33$.

For this example, it is easy to see that the DSS capacity is $C = 3 \leq 10/3$. In fact, a user contacting nodes v_1 and v_2 cannot download more information then their total storage

[1] The arguments are based on deriving the value of the min-cut in a graph, called *flow graph*, that represent DSSs [1]. For example, see the proof of Th. 8 in Appendix C.

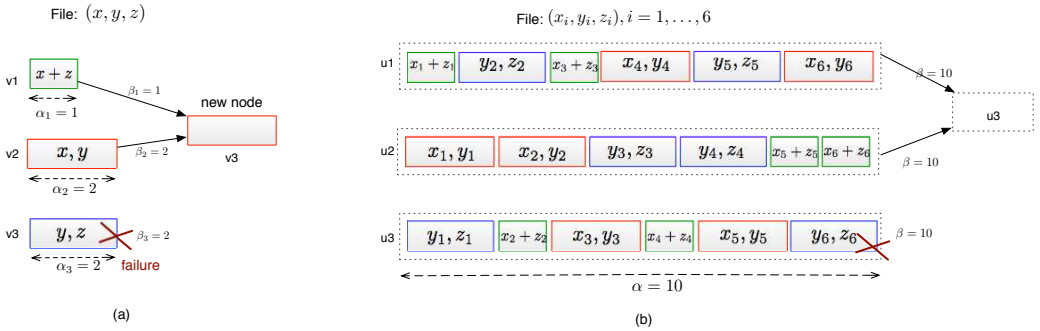

Fig. 1. An example that illustrates the proof of the upper bound (3) on the capacity of a heterogeneous system. (a) A heterogeneous distributed storage system (DSS) with $(n, k, d) = (3, 2, 2)$. The nodes have storage capacities $\alpha_1 = 1, \alpha_2 = \alpha_3 = 2$ and the repair bandwidth per helper are $\beta_1 = 1, \beta_2 = \beta_3 = 2$. (b) A DSS constructed by combining together $n! = 6$ copies of the original heterogeneous system corresponding to all possible node permutations. The obtained DSS is homogeneous with uniform storage per node $\alpha = 10$ and repair bandwidth per helper $\beta = 10$. The capacity of this system is 20 as given by (2) [1]. Any code that stores a file of size C ($C = 3$ here) on the original DSS can be transformed into a scheme that stores a file of size $n!C = 6C$ in the "bigger" system. This gives the upper bound in (3) $C \leq 20/6 = 10/3$.

$\alpha_1 + \alpha_2 = 3$. This upper bound is achieved by the code in Fig. 1(a). The code stores a file of 3 units (x, y, z) in the system. During repair the new node downloads the whole file and stores the lost piece of the data (note that the repair bandwidth constraints allow this trivial repair).

To obtain the upper bound in (3), we use the original heterogeneous DSS to construct a "bigger" homogeneous system. We obtain this new system by "glueing" together $n! = 3! = 6$ copies of the original DSS as shown in Fig. 1(b). Each copy corresponds to a different permutation of the nodes. In the figure, the i^{th} copy stores the file (x_i, y_i, z_i). For example in Fig. 1(b), the first copy is the original system itself, the second corresponds to node v_1 and node v_3 switching positions, and so on.

The "bigger" system is homogeneous because all its nodes have storage $\alpha = 10$ and repair bandwidth per helper $\beta = \gamma/d = 10$. The capacity C' of this system can be computed from (2):

$$\mathcal{C}' = \sum_{i=1}^{k} \min\left\{\alpha, (d - i + 1)\frac{\gamma}{d}\right\} = 20. \tag{4}$$

As seen in Fig. 1, any scheme that can store a file of size C in the original DSS can be transformed into a scheme that can store a file of size $n!C$ in the "bigger" DSS. Therefore, we get $n!C \leq C'$ and $C \leq 10/3$. This argument can be directly generalized to arbitrary heterogeneous systems. The general proof follows the same steps explained above and can be found in Appendix B.

Theorem 1 implies that symmetric repair, *i.e.*, downloading equal numbers of bits from each of the helpers, is optimal in a homogeneous system. To see this, consider a DSS with node storage capacity α, and a total repair bandwidth budget γ. A new node joining the system has the flexibility to arbitrarily split its repair bandwidth among the d helpers as long as the total amount of downloaded information does not exceed γ. In other words, we have $\sum_{i \in S} \beta_{ijS} = \gamma, \forall j, S$. Now, irrespective of how each new node splits its bandwidth budget, the average repair bandwidth in the system is the same, $\overline{\gamma} = \gamma$. If we apply Th. 1, we get an upper bound that matches exactly the capacity in (2) of a homogeneous DSS with symmetric repair. Hence, we obtain the result in Cor. 2.

B. Proof of Theorem 3

To avoid heavy notation, we focus on the case in which the repair bandwidth depends only on the helper node ($\beta_{ijS} = \beta_i$). We give in Th. 5 lower and upper bounds specific to this case. These bounds are similar to the ones in Th. 3, but can be tighter. The proof of Th. 3 follows the exact steps of the proof below and will be omitted here. Again, we assume that the nodes are indexed in increasing order of node capacity, $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n$. We also order the values of the repair bandwidths β to obtain the increasing sequence $\beta'_1 \leq \beta'_2 \leq \cdots \leq \beta'_n$.

Theorem 5: The capacity C of a heterogeneous DSS, in which the repair bandwidth depends only on the identity of the helper node, is bounded as $C'_{\min} \leq C \leq C'_{\max}$, where

$$\begin{aligned} C'_{\min} &= \sum_{i=1}^{k} \min(\alpha_i, \beta'_1 + \beta'_2 + \cdots + \beta'_{d-i+1}) \\ &= \min_{l=0,\ldots,k} \left(\sum_{i=1}^{l} \alpha_i + \sum_{j=0}^{k-l-1} \sum_{i=1}^{d-l-j} \beta'_i \right), \end{aligned} \tag{5}$$

and

$$\begin{aligned} C'_{\max} &= \sum_{i=1}^{k} \min(\alpha_i, \beta'_{i+1} + \beta'_{i+2} + \cdots + \beta'_{d+1}) \\ &= \min_{l=0,\ldots,k} \left(\sum_{i=1}^{l} \alpha_i + \sum_{j=1}^{k-l} \sum_{i=l+1+j}^{d+1} \beta'_i \right). \end{aligned} \tag{6}$$

The second expressions for $C'_{\min}$ and $C'_{\max}$ highlight the analogy with the bounds in Th. 3. Before proving Th. 5, we give a couple of illustrative examples and discuss some special cases.

Example 6: Consider again the example in the previous section where $(n, k, d) = (3, 2, 2)$ and where the nodes parameters are $(\alpha_1, \beta_1) = (1, 1), (\alpha_2, \beta_2) = (\alpha_3, \beta_3) = (2, 2)$. Here, $C'_{\min} = 2$ and $C'_{\max} = 3$. Note that here $C'_{\max}$ is tighter then

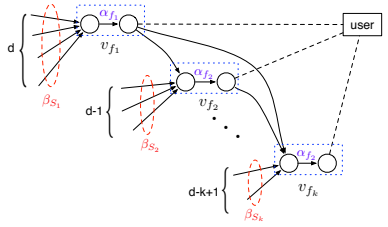

Fig. 2. A series of k failures and repairs in the DSS that explains the capacity expression in (7). Nodes $v_{f_1},\dots,v_{f_k}$ fail successively and are repaired as depicted above. The amount of "new" information that node v_{f_i} can give the user is the minimum between his storage capacity α_{f_i} and downloaded data β_{S_i}.

the average-based upper bound of Th. 1 which gives $C \leq 3.33$. Recall that the capacity for this system is $C = 3 = C'_{\min}$.

Example 7: Consider now a second DSS with $(n,k,d) = (3,2,2)$ and $(\alpha_1,\beta_1) = (5,3),(\alpha_2,\beta_2) = (6,4)$ and $(\alpha_3,\beta_3) = (7,5)$. Here, $C'_{\min} = 9$ and $C'_{\max} = 11$, and Th. 1 gives $C \leq 10 < C'_{\max}$.

The upper and lower bounds can coincide ($C'_{\min} = C'_{\max}$) in certain cases, which gives the exact expression of the capacity. For example:

1) A homogeneous DSS, where we recover the capacity expression in (2).
2) A DSS with uniform repair bandwidth, *i.e.*, $\beta_i = \beta, \forall i$. The capacity is $C = \sum_{i=1}^{k} \min(\alpha_i, (d-i+1)\beta)$.
3) Whenever $\alpha_i \leq \beta'_1, \forall i$. In this case the capacity $C = \sum_{i=1}^{k} \alpha_i$.

To prove the upper and lower bounds in Th. 5, we first establish the following expression of the DSS capacity.

Theorem 8: The capacity C of a heterogeneous DSS is given by

$$C = \min_{\substack{(f_1,\dots,f_k) \\ f_i \neq f_j \text{ for } i \neq j}} \sum_{i=1}^{k} \min\left(\alpha_{f_i}, \min_{\substack{|S_i|=d+1-i \\ S_i \cap \{f_1,\dots,f_i\}=\emptyset}} \beta_{S_i}\right), \quad (7)$$

where for any $S \subseteq \{1,\dots,n\}, \beta_S = \sum_{i\in S} \beta_i$.

The proof of Th. 8 is a generalization of the proof in [1] of the capacity of a homogeneous system (2). We defer this proof to Appendix C and explain here the intuition behind it. Consider the scenario depicted in Fig. 2 where nodes $v_{f_1},\dots,v_{f_k}$ fail and are repaired successively such that node v_{f_i} is repaired by downloading data from the previously repaired nodes $v_{f_1},\dots,v_{f_{i-1}}$ and $d-(i-1)$ other helper nodes in the system. Consider now a user contacting nodes $v_{f_1},\dots,v_{f_k}$. The amount of "non-redundant" information that node v_{f_i} can give to the user is evidently limited by its storage capacity α_i on one hand, and on the other hand, by the amount of information β_{S_i} downloaded from the $d-i+1$ helper nodes that are not connected to the user. Minimizing over all the choices of $f_1,\dots,f_k$ gives the expression in (7).

The optimization problem underlying the capacity expression in (7) is over a space that is exponential in k and d. It is not clear whether there exists a polynomial time algorithm that solves this problem and computes the DSS capacity. For this reason we give upper and lower bounds that are easy to compute. To get the lower bound in (5), let $(f_1,\dots,f_k) = (f_1^*,\dots,f_k^*)$ be the minimizer of (7). We have

$$\begin{aligned} C &= \sum_{i=1}^{k} \min\left(\alpha_{f_i^*}, \min_{\substack{|S_i|=d+1-i \\ \{f_1^*,\dots,f_i^*\}\cap S_i=\emptyset}} \beta_{S_i}\right) \\ &\geq \sum_{i=1}^{k} \min\left(\alpha_{f_i^*}, \beta'_1+\beta'_2+\cdots+\beta'_{d-i+1}\right) \\ &\geq \sum_{i=1}^{l^*} \alpha_i + \sum_{i=1}^{d-l^*} \beta'_i + \sum_{i=1}^{d-l^*-1} \beta'_i + \cdots + \sum_{i=1}^{d-k+1} \beta'_i \\ &= \min_{l=0,\dots,k} \left(\sum_{i=1}^{l} \alpha_i + \sum_{j=0}^{k-l-1} \sum_{i=1}^{d-l-j} \beta'_i\right), \end{aligned} \quad (8)$$

where $l^*, 0 \leq l^* \leq k$ is the number of those cases where $\alpha_{f_i^*}$ is smaller or equal than the corresponding sum of β's.

The upper bound $C'_{\max}$ is obtained by taking $(f_1,\dots,f_k) = (1,\dots,k)$ in (7) and following similar steps as above.

V. Security

We now consider the case of a passive adversary that eavesdrops on ℓ nodes in the system, and elaborate on the definition of secrecy capacity. The secrecy capacity C_s of the system is the maximum amount of information that can be delivered to any user without revealing any information to the eavesdropper (perfect secrecy).

Formally, let S be the information source that represents the file that is stored on the DSS. A user contacts the nodes in any set $B \subset \{v_1,\dots,v_n\}$ of size k and downloads their stored data denoted by C_B. The user should be able to decode the file, which implies that the relative entropy $H(S|C_B) = 0$. Let E be the set of the ℓ compromised nodes, and D_E be the data observed by the eavesdropper. The perfect secrecy condition implies that $H(S|D_E) = H(S)$. Following the definition in [20], we write the secrecy capacity as

$$C_s(\alpha,\gamma) = \sup_{\substack{H(S|C_B)=0 \forall B \\ H(S|D_E)=H(S) \forall E}} H(S). \quad (9)$$

Recall that in the case of an active adversary, the resiliency capacity of the system is the maximum amount of information that can be stored on the DSS such that any user contacting k nodes can still decode with no-errors despite the errors introduced by the malicious adversary.

Finding the secrecy and resiliency capacities of a DSS is a hard problem and is still open in general, even for the class of homogeneous systems. Following the same steps in the proof of Th. 1, we can show that the secrecy and resilience capacities of a heterogeneous DSS cannot exceed that of a homogeneous DSS having the same average resources. This result is stated in Th. 4.

In [20], the secure capacity of a homogeneous system with symmetric repair was studied. Moreover, upper bounds on the secrecy capacity in the presence of an eavesdropper and resiliency capacity in the presence of an omniscient and limited-knowledge active adversary were derived. We use these upper bounds in conjunction with Th. 4, to give the

Adversary Model	Upper Bound: Secure Capacity of a Homogenous DSS
Passive eavesdropper ($\ell < k$, $b = 0$)	$C_s \leq \sum_{i=\ell+1}^{k} \min\left\{\bar{\alpha}, (d-i+1)\frac{\bar{\gamma}}{d}\right\}$
Active omniscient adversary ($\ell = k$, $2b < k$)	$C_r \leq \sum_{i=2b+1}^{k} \min\left\{\bar{\alpha}, (d-i+1)\frac{\bar{\gamma}}{d}\right\}$
Active limited-knowledge adversary(ℓ, $b \leq \ell$)	$C_r \leq \sum_{i=b+1}^{k} \min\left\{\bar{\alpha}, (d-i+1)\frac{\bar{\gamma}}{d}\right\}$

TABLE I

UPPER BOUNDS ON THE SECRECY CAPACITY C_s AND THE RESILIENCY CAPACITY C_r OF A HETEROGENEOUS DISTRIBUTED STORAGE SYSTEM WITH AVERAGE NODE CAPACITY $\bar{\alpha}$ AND AVERAGE TOTAL REPAIR BANDWIDTH $\bar{\gamma}$. THE ADVERSARY IS CHARACTERIZED BY TWO PARAMETERS: ℓ, THE NUMBER OF NODES IT CAN EAVESDROP ON, AND b, THE NUMBER OF NODES IT CAN CONTROL. NOTE THAT IF THE CONDITIONS ON ℓ, b SPECIFIED IN THE FIRST COLUMN ARE NOT SATISFIED, THEN C_s, C_r ARE EQUAL TO ZERO.

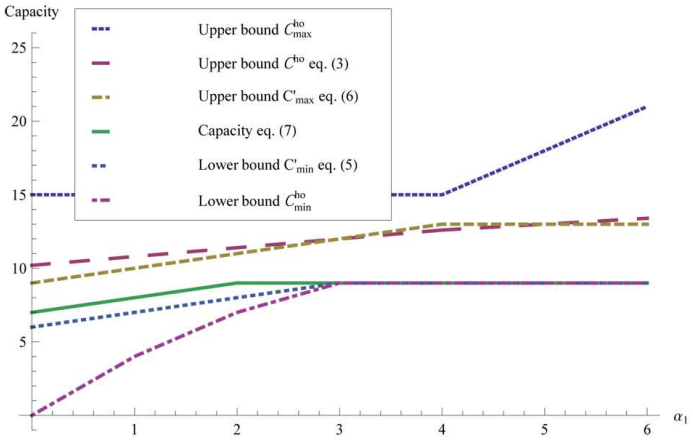

Fig. 3. The results of the first simulation. The different bounds and capacity are plotted for a heterogeneous DSS with $(n, k, d) = (5, 4, 4)$ in which we vary the storage capacity α_1 of the first node.

upper bounds on the secrecy and resiliency capacity of a heterogeneous DSS which can be found in Table I.

Using Th. 4, we easily deduce that symmetric repair is also optimal in terms of maximizing the secure capacity of a compromised DSS.

Corollary 9: Symmetric repair maximizes the secrecy capacity of a homogeneous system with a given budget on total repair bandwidth.

VI. SIMULATIONS

In this section, we compare our different bounds to the actual value of the capacity for two examples of heterogeneous DSSs. The examples we consider have small values for the parameters (n, k, d) which makes it possible to compute the capacity in (7) by brute force. In general, the tightness of our bounds depends on the system parameters and the distribution of the nodes' storage capacities α_i and repair bandwidths β_i. This behavior can be seen in the simulation plots in Fig. 3 and Fig. 4.

In the first simulation, we consider a DSS with $(n, k, d) = (5, 4, 4)$ and $\beta_1 = 1, \beta_2 = 3, \beta_3 = 2, \beta_4 = 1, \beta_5 = 2$ and $\alpha_2 = 3, \alpha_3 = 3, \alpha_4 = 4, \alpha_5 = 4$. We vary the storage α_1 of the first node from 0 to 6 continuously and plot the corresponding bounds and capacity in Fig. 3. The bounds we plot are the two upper bounds C^{ho} in (3) and C'_{max} in (6), and the lower bound C'_{min} in (5). Moreover, we plot the trivial upper and lower bounds, denoted respectively by C^{ho}_{min} and C^{ho}_{max}, which are obtained by using the minimum (respectively maximum) values of the α_i's and β_i's in (2).

In the second simulation, we consider a DSS with two types of nodes and vary the number of nodes of each type while keeping the total number of nodes n constant. We choose $(n, k, d) = (6, 4, 5)$ and nodes of type 1 have $(\alpha, \beta) = (6, 6)$ while nodes of type 2 have $(\alpha', \beta') = (5, 1)$. The results of this simulation are presented in Fig. 4.

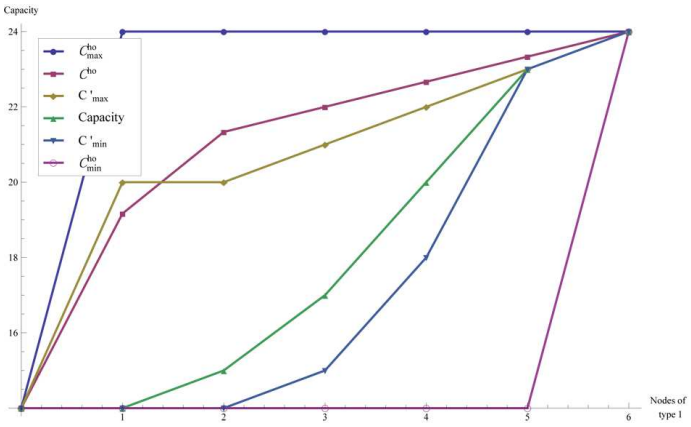

Fig. 4. The results of the first simulation. The different bounds and capacity are plotted for a heterogeneous DSS with $(n, k, d) = (6, 4, 5)$ and nodes of type 1 having $(\alpha, \beta) = (6, 6)$ and nodes of type 2 having $(\alpha', \beta') = (5, 1)$.

VII. CONCLUSION

We have studied distributed storage systems that are heterogeneous. Nodes in these systems can have different storage capacities and different repair bandwidths. We have focused on determining the information theoretic capacity of these systems, *i.e.*, the maximum amount of information they can store, to achieve a required level of reliability (any k out of the n nodes should be able to give a stored file to a user). We have

proved an upper bound on the capacity that depends on the average resources available per node. Moreover, we have given an expression for the system capacity when we know all the nodes' parameters. This expression may be hard to compute, but we use it to derive additional upper and lower bounds that are easy to evaluate. We have also studied the case in which the system is compromised by an active or passive adversary, and have provided bounds on the system secure capacity. Our results imply that symmetric repair maximizes the capacity of a homogeneous system, which justifies the repair model used in the literature. Problems that remain open include finding an efficient algorithm to compute the capacity of a heterogeneous distributed storage system, as well as efficient code constructions.

APPENDIX

A. Functional vs. Exact Repair

All of our results so far assumed a functional repair model. However, Theorems 1 and 4 can be directly extended to the exact repair case. For instance, Th. 1 becomes:

Theorem 10: The capacity C of a heterogeneous distributed storage system under exact repair, with node average capacity $\bar{\alpha}$ and average repair bandwidth $\bar{\gamma}$, is upper bounded by

$$C \leq C^{ho}_{exact}(\bar{\alpha}, \bar{\gamma}), \tag{10}$$

where $C^{ho}_{exact}(\bar{\alpha}, \bar{\gamma})$ is the capacity of a homogeneous DSS under exact repair.

In the proofs of Theorems 1 and 4 we construct a new "big" storage system using the original one as a building block. Hence, if we had exact repair in the original system to start with, we will have exact repair in the new "big" system. The results can thus be straightforwardly generalized to the case of exact repair. Moreover, under an exact repair constraint, a homogeneous DSS with symmetric repair maximizes capacity under given average node storage and repair bandwidth budgets.

The other results, namely Theorems 3, 5, and 8, are proved using the analysis of the information flow graph. Therefore, It is not clear if there is an obvious extension of these results to the case of exact repair.

B. Proof of Theorem 1

We prove Th. 1 by making formal the argument of the example in Section IV-A. We start by describing the operation of adding, or combining, together multiple storage systems having same number of nodes. Let $\mathcal{DSS}_1, \mathcal{DSS}_2$ be two storage systems with nodes $v_1^1, \ldots, v_n^1$ and $v_1^2, \ldots, v_n^2$, respectively. The new system that we refer to as $\mathcal{DSS}$ obtained by combining $\mathcal{DSS}_1$ and $\mathcal{DSS}_2$ is comprised of n nodes, say $u_1, \ldots, u_n$. Node u_i has storage capacity $\alpha_i = \alpha_i^1 + \alpha_i^2$ (superscript $j, j = 1, 2$, denotes a parameter of system S_j). Moreover, when node u_j fails in $\mathcal{DSS}$, the new node downloads $\beta_{ijS} = \beta_{ijS}^1 + \beta_{ijS}^2$ amount of information from helper node u_i (recall that S is the set of indices of the d helper nodes). We write $\mathcal{DSS} = \mathcal{DSS}_1 + \mathcal{DSS}_2$.

Now, let $\mathcal{DSS}$ be the given heterogeneous system for which we wish to compute its capacity C. For each permutation $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$, we denote by $\mathcal{DSS}_\sigma$ the storage system with nodes $v_1^\sigma, \ldots, v_n^\sigma$ such that $v_i^\sigma = v_{\sigma(i)}$. Let $\mathcal{P}_n$ denote the set of all $n!$ permutations on the set $\{1, \ldots, n\}$. We define a new "big" system by

$$\mathcal{DSS}_b = \sum_{\sigma \in \mathcal{P}_n} \mathcal{DSS}_\sigma.$$

The new system $\mathcal{DSS}_b$ is homogeneous with symmetric repair where the storage capacity per node α_b is given by

$$\alpha_b = (n-1)! \sum_{i=1}^{n} \alpha_i = n! \overline{\alpha}.$$

The repair bandwidth $\beta_{b,ijS}$ in $\mathcal{DSS}_b$ can be expressed as:

$$\begin{aligned}
\beta_{b,ijS} &\overset{(a)}{=} \sum_{\sigma \in \mathcal{P}_n} \beta_{i'j'S'} \\
&\overset{(b)}{=} \sum_{i',j',S'} \sum_{\substack{\sigma \in \mathcal{P}_n: \\ \sigma(i')=i, \sigma(j')=j, \sigma(S')=S}} \beta_{i'j'S'} \\
&\overset{(c)}{=} (n-d-1)!(d-1)! \sum_{j'=1}^{n} \sum_{\substack{i'=1 \\ i' \neq j'}}^{n} \sum_{S'} \beta_{i'j'S'} \qquad (11) \\
&\overset{(d)}{=} (n-d-1)!(d-1)! \sum_{j'=1}^{n} \binom{n-1}{d} \gamma_{j'} \\
&= \frac{(n-1)!}{d} \sum_{j'=1}^{n} \gamma_{j'} = n! \frac{\bar{\gamma}}{d}.
\end{aligned}$$

(a) Here, $i' = \sigma^{-1}(i), j' = \sigma^{-1}(j)$ and $S' = \sigma^{-1}(S)$. (b) $i', j' = 1, \ldots, n$, and $i' \neq j'$. $|S'| = d$ with $i' \in S'$ and $j' \notin S'$. (c) The number of permutations σ that satisfy $\sigma(i') = i, \sigma(j') = j, \sigma(S') = S$ is $(n-d-1)!(d-1)!$. (d) This follows from (1).

Therefore, the capacity C_b of $\mathcal{DSS}_b$ as given by (2) is

$$C_b = n! \sum_{i=1}^{k} \min\left\{\bar{\alpha}, (d-i+1)\frac{\bar{\gamma}}{d}\right\}. \tag{12}$$

Any scheme achieving the capacity C of the original system can be naturally extended to store a file of size $n!C$ in $\mathcal{DSS}_b$ (see Fig. 1). Therefore, $C_b \geq n!C$. This inequality combined with (12) gives the result of the Th. 1.

C. Proof of Theorem 8

We use the definition of the flow graph in [1] to represent the DSS. The flow graph is a multicast network in which the multiple destinations correspond to the users requesting files from the DSS by contacting any k out of the n nodes. Therefore, the capacity of the DSS is the capacity of this multicast network which is equal to the minimum value of the min-cuts to the users, by the fundamental theorem of network coding. Note that in the flow graph, a storage node v_i is represented by two vertices x_{in}^i and x_{out}^i connected by an edge of capacity α_i (see Fig. 2).

Let C be the capacity of the DSS and define F to be

$$F \triangleq \min_{\substack{(f_1,\ldots,f_k) \\ f_i \neq f_j \text{ for } i \neq j}} \sum_{i=1}^{k} \min \left(\alpha_{f_i}, \min_{\substack{|S_i|=d+1-i \\ \{f_1,\ldots,f_i\} \cap S_i = \emptyset}} \beta_{S_i} \right).$$

We want to show that $C = F$.

Let $(f_1, \ldots, f_k)$ be fixed and consider the successive failures and repairs of nodes $v_{f_1}, \ldots, v_{f_n}$ as seen in Fig. 2. Suppose node v_{f_1} is repaired by contacting the helper nodes that minimize the sum β_{S_1} with $|S_1| = d$ and $\{f_1\} \cap S_1 = \emptyset$, and node v_{f_2} is repaired by contacting node v_{f_1} and the $d-1$ helper nodes that minimize the sum β_{S_2} with $|S_2| = d-1$ and $\{f_1, f_2\} \cap S_2 = \emptyset$. We continue in this fashion and finish with node v_{f_k} being repaired by contacting nodes $v_{f_1}, \ldots, v_{f_{k-1}}$ and the $d - k + 1$ helper nodes that minimize β_{S_k} with $|S_k| = d + 1 - k$ and $\{f_1, \ldots, f_k\} \cap S_k = \emptyset$. Now consider a user contacting nodes $v_{f_1}, \ldots, v_{f_n}$ there is a cut to the user of value F. By the max-flow min-cut theorem, we get $C \leq F$.

To prove the other direction, consider a user in the system and let E denote the edges in the min-cut that separates this user from the source in the flow graph. Also, let V be the set of vertices in the flow graph that have a path to the user. Since the flow graph is acyclic, we have a topological ordering of the vertices in V, which means that they can be indexed such that an edge from v_i to v_j implies $i < j$.

Let x^1_{out} be the first "out-node" in V (with respect to the ordering). If $x^1_{in} \notin V$, then $x^1_{in}x^1_{out} \in E$. On the other hand, if $x^1_{in} \in V$, then the set of incoming edges S'_1, $|S'_1| = d$, of x^1_{in} must be in E.

Now similarly let x^2_{out} be the second "out-node" in V with respect to the ordering. If $x^2_{in} \notin V$, then $x^1_{in}x^2_{out} \in E$. If $x^2_{in} \in V$, then the set $S'_2, |S'_2| \geq d-1$, of edges incoming to x^2_{in}, not including a possible edge from x^1_{out}, must be in E. All k nodes adjacent to the user must be in V so continuing in the same fashion gives that the min-cut is at least

$$\sum_{i=1}^{k} \min(\alpha_{f_i}, \beta_{S'_i}),$$

where $f_i \neq f_j$ for $i \neq j$, $|S'_i| \geq d + 1 - i$, and $\{f_1, \ldots, f_i\} \cap S'_i = \emptyset$.

Let $S''_i \subseteq S'_i$ be a subset such that $|S''_i| = d + 1 - i$. Now the min-cut is at least

$$\sum_{i=1}^{k} \min(\alpha_{f_i}, \beta_{S''_i}) \geq \sum_{i=1}^{k} \min \left(\alpha_{f_i}, \min_{\substack{|S_i|=d+1-i \\ \{f_1,\ldots,f_i\} \cap S_i = \emptyset}} \beta_{S_i} \right)$$

giving that $C \geq F$.

REFERENCES

[1] A. Dimakis, P. Godfrey, Y. Wu, M. Wainright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Transactions on Information Theory*, 56(9):4539–4551, Sep. 2010.

[2] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The Google file system. In *Proc. 19th ACM Symposium on Operating Systems Principles*, 2003.

[3] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh. A survey on network codes for distributed storage. *arXiv:1004.4438*, 2010.

[4] K. V. Rashmi, Nihar B. Shah, and P. Vijay Kumar. Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction. *IEEE Transactions on Information Theory*, 57(8), 2011.

[5] S. El Rouayheb and K. Ramchandran. Fractional repetition codes for repair in distributed storage systems. In *Proc. 48th Annual Allerton*, Monticello, IL, 2010.

[6] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin. Erasure coding in windows azure storage. In *Proc. 2012 USENIX Annual Technical Conference (ATC)*, 2012.

[7] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur. XORing Elephants: Novel Erasure Codes for Big Data. In *arXiv:1301.3791*, 2013.

[8] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Shea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. Oceanstore: An architecture for global-scale persistent storage. In *Proc. 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 190–201, November 2000.

[9] Anthony Ha. P2P startup Space Monkey raises 2.25m led by Google Ventures and Venture51, http://techcrunch.com/2012/07/11/space-monkey-seed-round, July 2012.

[10] H. Zhang, M. Chen, A. Parek, and K. Ramchandran. A distributed multichannel demand-adaptive p2p vod system with optimized caching and neighbor-selection. In *Proceedings of SPIE*, San Diego, CA, 2011.

[11] S. Pawar, S. El Rouayheb, H. Zhang, K. Lee, and K. Ramchandran. Codes for a distributed caching based video-on-demand system. In *Proc. Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, November 2011.

[12] N. Golrezaei, A. G. Dimakis, and A. F. Molisch. Wireless device-to-device communications with distributed caching. In *Proc. IEEE International Symposium on Information Theory*, Cambridge, MA, 2012.

[13] Y. Wu. A construction of systematic MDS codes with minimum repair bandwidth. *arXiv:0910.2486*, 2009.

[14] T. Ernvall, S. El Rouayheb, C. Hollanti, and H. V. Poor. Capacity and security of heterogeneous distributed storage systems. In *Proceedings of 2013 IEEE Int. Symp. on Inf. Theory (ISIT 2013)*, Istanbul, Turkey, July 2013.

[15] V. T. Van, C. Yuen, and J. Li. Non-homogeneous distributed storage systems. In *arXiv:1208.2078*, 2012.

[16] D. Leong, A. G. Dimakis, and T. Ho. Distributed storage allocations. In *arXiv:1011.5287v3*, 2012.

[17] V. Ntranos, G. Caire, and A. G. Dimakis. Allocations for heterogeneous distributed storage. In *Proc. IEEE International Symposium on Information Theory*, Cambridge, MA, 2012.

[18] N. B. Shah, K. V. Rashmi, and V. Kumar. A flexible class of regenerating codes for distributed storage. In *IEEE International Symposium on Information Theory (ISIT)*, 2010.

[19] S. Pawar, S. El Rouayheb, and K. Ramchandran. On secure distributed data storage under repair dynamics. In *Proc. IEEE International Symposium on Information Theory*, Austin, TX, 2010.

[20] S. Pawar, S. El Rouayheb, and K. Ramchandran. Securing dynamic distributed storage systems against eavesdropping and adversarial attacks. *IEEE Transactions on Information Theory*, 58(3):6734–6753, March 2012.

Toni Ernvall received the M. Sc. degree from the University of Turku, Finland, in 2011 in mathematics.

Since February 2011, he has been with the Department of Mathematics and Statistics at the University of Turku. His research interests include coding theory and networks.

Salim El Rouayheb (S'07, M'09) received the Diploma degree in electrical engineering from the Lebanese University, Roumieh, Lebanon, in 2002, and the M.S. degree in computer and communications engineering from the American University of Beirut, Lebanon, in 2004. He received the Ph.D. degree in electrical engineering from Texas A&M University, College Station, in 2009.

Dr. El Rouayheb is an associate research scholar at the Electrical Engineering Department at Princeton University. From 2010 to 2011, he was a postdoctoral research fellow with the Wireless Foundations (WiFo) Lab at the Electrical Engineering and Computer Sciences (EECS) Department, University of California, Berkeley (2010-2011). His research interests lie in the broad area of communications with a focus on distributed storage systems, network coding, and information-theoretic security.

Camilla Hollanti (M'09) received the M.Sc. and Ph.D. degrees from the University of Turku, Finland, in 2003 and 2009, respectively, both in pure mathematics. Her research interests include applications of algebraic number theory to wireless communications with an emphasis on MIMO lattice code design, and distributed storage systems.

For 2004-2013 Hollanti was with the Department of Mathematics, University of Turku, Finland. In 2005, she visited the Department of Algebra at Charles' University, Prague, Czech Republic for six months. She joined the Department of Mathematics, University of Tampere, Finland, as a Lecturer for the academic year 2009-2010. Since November 2011 she has been with the Department of Mathematics and Systems Analysis at Aalto University, Finland, as an Assistant Professor.

Hollanti has received several grants and scholarships, including the Finnish Cultural Foundation research grant in 2007, the Finnish Academy of Science research grant in 2008, the Young Researcher's Project grant from the Emil Aaltonen Foundation in 2009, and the Academy of Finland project grant in 2010.

H. Vincent Poor (S'72, M'77, SM'82, F'87) received the Ph.D. degree in EECS from Princeton University in 1977. From 1977 until 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990 he has been on the faculty at Princeton, where he is the Michael Henry Strater University Professor of Electrical Engineering and Dean of the School of Engineering and Applied Science. He has also held visiting appointments at several other institutions, including most recently Imperial College and Stanford. Dr. Poor's research interests are in the areas of stochastic analysis, statistical signal processing and information theory, and their applications in wireless networks and related fields including social networks and smart grid. Among his publications in these areas are the recent books Smart Grid Communications and Networking (Cambridge University Press, 2012) and Principles of Cognitive Radio (Cambridge University Press, 2013).

Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences, a Fellow of the American Academy of Arts and Sciences, an International Fellow of the Royal Academy of Engineering (U. K), and a Corresponding Fellow of the Royal Society of Edinburgh. He is also a Fellow of the IET, the Optical Society of America, and other scientific and technical organizations. In 1990, he served as President of the IEEE Information Theory Society, and in 2004-07 he served as the Editor-in-Chief of the IEEE Transactions on Information Theory. He received a Guggenheim Fellowship in 2002, the IEEE Education Medal in 2005, and the Marconi and Armstrong Awards of the IEEE Communications Society in 2007 and 2009, respectively. Recent recognition of his work includes the 2010 IET Ambrose Fleming Medal for Achievement in Communications, the 2011 IEEE Eric E. Sumner Award, a Royal Academy Distinguished Visiting Fellowship (2012), and honorary doctorates from Aalborg University, the Hong Kong University of Science and Technology, and the University of Edinburgh.