Repairable Fountain Codes

Megasthenis Asteris and Alexandros G. Dimakis
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089
Email: {asteris, dimakis}@usc.edu

***Abstract*—We introduce a new family of Fountain codes that are systematic and also have sparse parities. Although this is impossible if we require the code to be MDS, we show it can be achieved if we relax our requirement into a near-MDS property. More concretely, for any ϵ we construct codes that guarantee that a random subset of $(1+\epsilon)k$ symbols suffices to recover the original symbols with high probability. Our codes produce an unbounded number of output symbols, creating each parity independently by linearly combining a logarithmic number of input symbols.**

This structure has the additional benefit of logarithmic locality: a single symbol loss can be repaired by accessing only $O(\log k)$ other coded symbols. This is a desired property for distributed storage systems where symbols are spread over a network of storage nodes. Our mathematical contribution involves analyzing the rank of sparse random matrices over finite fields. We rely on establishing that a new family of sparse random bipartite graphs have large matchings with high probability.

I. Introduction

Fountain codes [1], [2], [3] form a new family of linear erasure codes with several attractive properties. For a given set of k input symbols, a Fountain code produces a potentially limitless stream of output symbols, each created independently of others as a random combination of input symbols according to a given distribution. Ideally, given a randomly selected subset of $(1+\epsilon)k$ encoded symbols, a decoder should be able to efficiently recover the original k input symbols with high probability (w.h.p.) for some small overhead ϵ. Independent and random construction of encoded symbols allows decentralized encoding and dynamic adjusting in the number of coded symbols. Further, Fountain codes can be used with efficient encoding and decoding algorithms.

Current cloud storage systems are starting to use erasure coding techniques, typically Reed-Solomon codes [4]. In this paper we investigate and design Fountain codes for such distributed storage applications.

One important property of distributed storage codes is efficient repair [5]: when a single encoded symbol is lost it should be possible to reconstruct it without communicating too much information from other coded symbols. A related property is that of locality of each symbol: the number of other code symbols that need to be accessed to reconstruct that symbol [4], [6], [7], [8].

In addition, it is highly desired that distributed storage codes are systematic, *i.e.*, the original information symbols appear in the encoded sequence. This enables the reading of symbols without decoding and is a practical requirement for most storage applications. Raptor codes, a class of Fountain codes, can be transformed into a systematic form [3] by introducing a change of variables. Unfortunately, due to this two layer encoding, the parity symbols are no longer sparse in the input symbols.

Our Contribution: We introduce a new family of Fountain codes that are systematic and also have parity symbols with logarithmic sparsity. We show that this is impossible if we require the code to be MDS, but is possible if we allow a near-MDS property similar to the probabilistic guarantees provided by LT and Raptor codes.

More concretely, for any $\epsilon > 0$ we construct codes that guarantee that a random subset of $(1+\epsilon)k$ symbols suffices to recover the original symbols w.h.p. Our codes produce an unbounded number of output symbols, creating each parity independently by linearly combining a logarithmic number of input symbols.

This structure provides also logarithmic locality: each symbol in our codes is repairable by accessing only $O(\log k)$ other coded symbols. The disadvantage of our codes is less efficient decoding complexity, since the peeling decoder cannot be used for our construction.

Technically, we rely on a novel random matrix result: we show that systematic matrices with independent parity columns and logarithmic density have full rank submatrices of near optimal size. Our analysis builds on the connections of matrix determinants to flows on random bipartite graphs, using techniques from [9], [10]. Our key technical result is showing that a new family of sparse random graphs have matchings w.h.p.

II. Problem Description

Given k input symbols, elements of a finite field $\mathbb{F}_q$, we want to encode them into n symbols using a linear code. Linear codes are described by a $k \times n$ *generator* matrix $\mathbf{G}$ over $\mathbb{F}_q$, which when multiplied by an input vector $\mathbf{u} \in \mathbb{F}_q^{1\times k}$ produces a codeword $\mathbf{v} \doteq \mathbf{uG} \in \mathbb{F}_q^{1\times n}$. We want $\mathbf{G}$ to have the following properties:

- *Systematic form*, *i.e.*, a subset of the columns of $\mathbf{G}$ forms the identity matrix, $\mathbf{I}$, which implies that the input symbols are reproduced in the encoded sequence.
- *Rateless property*, *i.e.*, each column is created independently. The number n of columns does not have to be

This research was supported in part by NSF Career Grant CCF-1055099 and research gifts by Intel and Microsoft Research.

specified for the encoder a priori.

- *MDS property*, *i.e.*, any k columns of $\mathbf{G}$ have rank k, implying that any subset of k encoded symbols suffices to retrieve the input.
- *Good locality*. $\mathbf{G}$ has locality l if each column can be written as a linear combination of at most l other columns. If the code is systematic, then sparse parities suffice to obtain good locality [8].

Any sufficiently large subset of encoded symbols should allow recovery of the original data. In the case of MDS codes, an information theoretically minimum subset of k encoded symbols suffices to decode. It can be easily shown, however, that the generator matrix of a systematic MDS code affords no zero coefficient in the parity columns. Consider a parity column with a zero coefficient in the i-th position: this parity column along with any $k-1$ systematic columns excluding the one corresponding to the i-th input symbol, form a singular matrix. Therefore, if parities are deliberately sparse in the input symbols, seeking to improve the code's *locality*, the *"any k"* property has to be relaxed.

III. Prior Work

In LT codes, the first practical realizations of Fountain codes invented by Luby [2], the average degree of the output symbols, *i.e.*, the number of input symbols combined into an output symbol, is $O(\log k)$. Note, however, that sparsity in this case does not imply good locality, since LT codes lack systematic form.

Shokrollahi in [3] introduced Raptor codes, a different class of Fountain codes. Building on LT, Raptor codes decreased the per symbol encoding and decoding cost (which corresponds to average degree of encoded symbols) to a constant. A systematic flavour of Raptor codes is provided in [3], but the parity symbols are no longer sparse in the input symbols.

Gummadi in his thesis [11] also considers the use of Fountain codes for storage applications and suggests sparse systematic variants of LT and Raptor codes. His constructions have the disadvantage of requiring an overhead ϵ that cannot be made arbitrarily small but rather is bounded by $1+\delta$, $(\delta > 0)$ and 0.25 respectively. On the other hand, the advantage of this construction is that efficient decoding is still possible.

IV. Repairable Fountain Codes

We introduce a new family of Fountain codes that are systematic and also have sparse parities. Each parity symbol is a random linear combination of up to $d(k)$ randomly chosen input symbols. Due to their randomized nature, our codes provide a probabilistic guarantee on successful decoding of $k^{'} = (1+\epsilon)k$ randomly selected encoded symbols, for arbitrarily small $\epsilon > 0$ (near-MDS). Requiring decoding of a random set of $k^{'}$ encoded symbols to be successful w.h.p. (that is with vanishingly small probability of failure as k grows), we show that a sparsity level of $O(\log k)$ is achievable. Our main result, which is asymptotic in k, is established in Theorem 1, at the end of this section.

It is useful to describe our randomized construction through a correspondence to a bipartite graph $\mathcal{G}(U, V, E)$, depicted in Fig. 1. The set of nodes U on the left side corresponds to the input symbols and the set V on the right corresponds to the encoded symbols. An edge $(i, j) \in E$ if the input symbol $i \in U$ is one of the symbols combined into the encoded symbol $j \in V$. Evidently, each of the k first encoded symbols is connected to a single, distinct input symbol (systematic part). The remaining nodes in V correspond to parity symbols. Each parity node $j \in V$ randomly and *independently* selects (with repetition) a subset $N(j)$ of the input symbols as follows: an input symbol is selected uniformly and independently from U and added in $N(j)$ and this procedure is repeated $d(k)$ times. Therefore, $|N(j)|$ will be smaller than $d(k)$ if the same data node is selected twice. In fact, the size of the set $N(j)$ is exactly the number of coupons a coupon collector would have after purchasing $d(k)$ coupons from a set of k coupons. It is not hard to see that when $d(k) \ll k$, $|N(j)|$ will be approximately equal to $d(k)$ w.h.p. Each parity symbol is a random linear combination of the input symbols it is connected to:

$$v_j = f_{ij} u_i \quad \forall i \in N(j), \tag{1}$$

where the coefficients f_{ij} are selected uniformly and independently over $\mathbb{F}_q$.

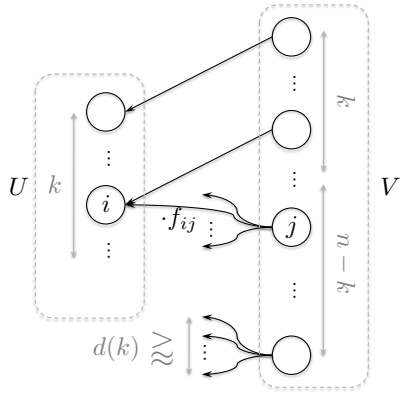

Fig. 1. Bipartite graph corresponding to our randomized code construction.

In matrix notation, the construction can be summarized into

$$\mathbf{v} \doteq \mathbf{u}\mathbf{G}, \tag{2}$$

where $\mathbf{v}$ is a $1 \times n$ vector of encoded symbols, $\mathbf{u}$ is a $1 \times k$ input symbol vector and $\mathbf{G}$ is a $k \times n$ matrix of the form $\mathbf{G} = [\,\mathbf{I} \mid \mathbf{P}\,]$, *i.e.*, the k first columns correspond to the identity matrix (systematic part). For $j \in \{k+1, \ldots, n\}$, the j-th column of $\mathbf{G}$ (*i.e.*, each column of the parity matrix, $\mathbf{P}$) has $|N(j)| \leq d(k)$ nonzero entries. To retrieve the k input symbols based on $k^{'}$ encoded symbols, we need k out of the latter to be linearly independent combinations of the former. Therefore, the key property required for successful decoding is that a randomly selected matrix $\mathbf{G}_{\mathcal{S}}$, consisting of $k^{'}$ columns of $\mathbf{G}$ (including any combination of systematic and parity parts) has full rank w.h.p.

So far, we have seen that our construction is systematic and has the rateless property: parity symbols can be created randomly and independently and hence any number of symbols can be produced dynamically. Its locality is directly related to $d(k)$, which is clearly a measure of the sparsity of $\mathbf{G}$: the repair of any erased parity symbol involves as few as $d(k)$ systematic symbols, while similarly, the repair of a systematic symbol involves a parity symbol covering the erased symbol and up to $d(k)-1$ other systematic symbols contributing to the same parity.

Our main contribution is identifying how *small* $d(k)$ can be to ensure that a randomly selected submatrix $\mathbf{G}_{\mathcal{S}}$ is full rank w.h.p.

Theorem 1. *Let* $\mathbf{G} = [\,\mathbf{I} \mid \mathbf{P}\,]$ *be a random matrix with independent columns constructed as described. Then,* $d(k) = c\log k$ *is sufficient for a random* $k \times k^{'}$ *submatrix* $\mathbf{G}_{\mathcal{S}}$ *of* $\mathbf{G}$ *to be full rank, i.e. contain an invertible* $k \times k$ *submatrix* $\mathbf{G}_{\mathcal{S}_k}$*, with probability* $1 - \frac{k}{q} - o(1)$*, for some positive constant* c*.*

Theorem 2. *(Converse) If each parity column of* $\mathbf{G}$ *is generated independently, then* $d(k) = \Omega(\log k)$ *is necessary for a random* $k \times k^{'}$ *submatrix* $\mathbf{G}_{\mathcal{S}}$ *of* $\mathbf{G}$ *to be full rank w.h.p.*

From the two theorems, it follows that our codes achieve optimal locality with a logarithmic degree for every parity symbol. Original data is reconstructed in $O(k^3)$ using Maximum Likelihood (ML) decoding, which corresponds to solving a linear system of k equations in $GF(q)$. Note, however, that the Wiedemann algorithm [12] can reduce complexity to $O\left(k^2 \log k\right)$ on average, exploiting the sparsity of the linear equations, with negligible extra memory requirement. Finally, note that in order to achieve vanishingly small probability of failure as k grows, the size of the field must grow accordingly.

V. Analysis and Proofs

Proof of Theorem 1. The main body of this section is dedicated to the proof of Theorem 1, stating that when $\mathbf{G}$ is constructed as described in section IV, a randomly selected $k \times k^{'}$ submatrix $\mathbf{G}_{\mathcal{S}}$ is full rank w.h.p. More formally,

$$\Pr\left(\exists\ \mathbf{G}_{\mathcal{S}_k} : \det(\mathbf{G}_{\mathcal{S}_k}) \neq 0\right) = 1 - \left(\frac{k}{q} + o(1)\right). \tag{3}$$

In the following, we exploit a connection between determinants and perfect matchings (P.M.'s) in bipartite graphs. In section IV, we showed the correspondence of the randomly constructed matrix $\mathbf{G}$ to an unbalanced bipartite graph $\mathcal{G} = (U, V, E)$. The submatrix $\mathbf{G}_{\mathcal{S}}$ corresponds to a subgraph $\mathcal{G}_{\mathcal{S}} = (U_{\mathcal{S}} = U, V_{\mathcal{S}}, E_{\mathcal{S}})$, depicted in Fig. 2, containing all k nodes of U on the left side, $k^{'}$ out of the n nodes of V on the right side, and the subset $E_{\mathcal{S}}$ of the edges incident only to nodes in these sets. Similarly, any $k \times k$ submatrix $\mathbf{G}_{\mathcal{S}_k}$ of $\mathbf{G}_{\mathcal{S}}$ corresponds to a smaller, balanced bipartite graph, $\mathcal{G}_{\mathcal{S}_k} = (U, V_{\mathcal{S}_k}, E_{\mathcal{S}_k})$, with k nodes on each side. $\mathbf{G}_{\mathcal{S}_k}$ can be regarded as the Edmond's matrix of the corresponding bipartite graph $\mathcal{G}_{\mathcal{S}_k}$:

$$G_{\mathcal{S}_k}(i,j) = \begin{cases} a_{i,j}, & \text{if } (u_i, v_j) \in E_{\mathcal{S}_k} \\ 0, & \text{if } (u_i, v_j) \notin E_{\mathcal{S}_k} \end{cases}, \tag{4}$$

where $u_i \in U, v_j \in V_{\mathcal{S}_k}$ and $a_{i,j}$'s are indeterminates. Then:

Lemma 1. *The determinant of* $\mathbf{G}_{\mathcal{S}_k}$ *is nonzero if and only if there exists a perfect matching in* $\mathcal{G}_{\mathcal{S}_k}$*, i.e.*

$$\det(\mathbf{G}_{\mathcal{S}_k}) \neq 0 \quad \Leftrightarrow \quad \exists \text{ perfect matching } M \text{ in } \mathcal{G}_{\mathcal{S}_k}. \tag{5}$$

Proof: We use the following expression for the determinant:

$$\det(\mathbf{G}_{\mathcal{S}_k}) = \sum_{\pi \in S_n} \operatorname{sgn}(\pi) \prod_{i=1}^{n} \mathbf{G}_{\mathcal{S}_k}(i, \pi(i)), \tag{6}$$

where S_n is the set of all permutations on $\{1, \ldots, n\}$ and $\operatorname{sgn}(\pi)$ is the sign of permutation π. There is a one to one correspondence between a permutation $\pi \in S_n$ and a candidate P.M. $\left\{(u_1, v_{\pi(1)}), \ldots, (u_n, v_{\pi(n)})\right\}$ in $\mathcal{G}_{\mathcal{S}_k}$. Note that if the candidate P.M. does not exist in $\mathcal{G}_{\mathcal{S}_k}$, i.e. some edge $(u_i, v_{\pi(i)}) \notin E_{\mathcal{S}_k}$ then the term corresponding to π in the summation is 0. Therefore, we have:

$$\det(\mathbf{G}_{\mathcal{S}_k}) = \sum_{\pi \in \mathcal{P}} \operatorname{sgn}(\pi) \prod_{i=1}^{n} a_{i,\pi(i)}, \tag{7}$$

where $\mathcal{P}$ is the set of perfect matchings in $\mathcal{G}_{\mathcal{S}_k}$. This is clearly zero if $\mathcal{P} = \emptyset$, *i.e.*, if $\mathcal{G}_{\mathcal{S}_k}$ has no P.M. If $G_{\mathcal{S}_k}$ has a P.M., there exists a $\pi \in \mathcal{P}$ and the term corresponding to π is $\prod_{i=1}^{n} a_{i,\pi(i)} \neq 0$. Additionally, there is no other term in the summation containing the exact same set of variables and this term cannot be cancelled out. In this case $\det(\mathbf{G}_{\mathcal{S}_k}) \neq 0$, which concludes the proof of the lemma. ∎

However, $\mathbf{G}_{\mathcal{S}_k}$ is not an actual Edmond's matrix; its entries are (randomly selected) elements of a finite field $\mathbb{F}_q$, not indeterminates. There are two substantially different cases in which $\det(\mathbf{G}_{\mathcal{S}_k}) = 0$:

- The determinant polynomial is identically zero which occurs if and only if $\mathcal{G}_{\mathcal{S}}$ has no P.M., or
- it is not identically zero (*i.e.*, $\mathcal{G}_{\mathcal{S}_k}$ has a P.M.), but the selected coefficients correspond to a root of the polynomial.

In other words, in contrast to the use of indeterminates, an unfortunate selection of the random coefficients of $\mathbf{G}_{\mathcal{S}_k}$ can lead to zero determinant even when $\mathcal{G}_{\mathcal{S}_k}$ has a P.M.

Taking into account that a P.M. in some subgraph $\mathcal{G}_{\mathcal{S}_k}$ of $\mathcal{G}_{\mathcal{S}}$ is a P.M. M in $\mathcal{G}_{\mathcal{S}}$ and vice versa, the probability we are interested in can be written as

$$\begin{aligned} &\Pr(\exists \mathbf{G}_{\mathcal{S}_k} : \det(\mathbf{G}_{\mathcal{S}_k}) \neq 0) \\ =&1 - \Pr(\nexists \mathbf{G}_{\mathcal{S}_k} : \det(\mathbf{G}_{\mathcal{S}_k}) \neq 0) \\ =&1 - \Bigg[\underbrace{\Pr(\nexists \mathbf{G}_{\mathcal{S}_k} : \det(\mathbf{G}_{\mathcal{S}_k}) \neq 0 | \exists M)}_{\alpha} \cdot \Pr(\exists M) \\ &\quad + \underbrace{\Pr(\nexists\ \mathbf{G}_{\mathcal{S}_k} : \det(\mathbf{G}_{\mathcal{S}_k}) \neq 0 | \nexists M)}_{\beta} \cdot \Pr(\nexists M)\Bigg]. \end{aligned} \tag{8}$$

Nonexistence of a P.M. in $G_{\mathcal{S}}$, implies that we cannot find k nodes in $V_{\mathcal{S}}$ that can be perfectly matched with the k nodes of U. In that case, no submatrix $\mathbf{G}_{\mathcal{S}_k}$ can have nonzero determinant, *i.e.* $\beta = 1$. On the other hand, existence of a P.M. M in $\mathcal{G}_{\mathcal{S}}$, implies the existence of a submatrix $\mathbf{G}_{\mathcal{S}_k}$ that will most probably have a nonzero determinant, depending on the randomly selected coefficients of $\mathbf{G}_{\mathcal{S}_k}$. The determinant of $\mathbf{G}_{\mathcal{S}_k}$ corresponding to M, is a polynomial of degree exactly k and the probability that it equals zero can be bounded using the Schwartz-Zippel Theorem [13], by $\frac{k}{q}$, where q is the number of elements in the finite field from which the coefficients of $\mathbf{G}_{\mathcal{S}_k}$ are drawn. A step further, the probability $\Pr(\nexists\ \mathbf{G}_{\mathcal{S}_k} : \det(\mathbf{G}_{\mathcal{S}_k}) \neq 0 \mid \exists M)$, *i.e.*, that no invertible matrix $\mathbf{G}_{\mathcal{S}_k}$ exists despite the existence of a P.M., can be upper bounded by the same quantity. Hence $\alpha \leq \frac{k}{q}$. Continuing from (8), we have:

$$\Pr\left(\exists \mathbf{G}_{\mathcal{S}_k} : \det\left(\mathbf{G}_{\mathcal{S}_k}\right) \neq 0\right) \geq 1 - \left[\frac{k}{q}\Pr\left(\exists M\right) + \Pr\left(\nexists M\right)\right]$$
$$= 1 - \frac{k}{q}\left(1 - \Pr\left(\nexists M\right)\right) - \Pr\left(\nexists M\right) = 1 - \frac{k}{q} - o(1), \qquad (9)$$

where the fact that $\Pr\left(\nexists M\right) = o(1)$ is based on Lemma 2, provided in section V-A. This completes the proof of theorem 1. $\square$

Proof of Theorem 2 An input symbol is *covered* by an encoded symbol, if it participates with a nonzero coefficient in the formation of the latter. In order to be able to retrieve the original k symbols from a subset of $k^{'}$ encoded symbols, it is imperative that the subset covers all input symbols. It is a standard result in balls and bins analysis that covering k bins w.h.p. requires throwing $\Omega\left(k \log k\right)$ balls. Here, each of the $k^{'}$ encoded symbol "throws $d(k)$ balls". Therefore, we need $k^{'} \cdot d(k) = (1+\epsilon)k \cdot d(k) = \Omega\left(k \log k\right)$ from which the theorem becomes obvious. $\square$

A. Existence of Perfect Matching in the random subgraph

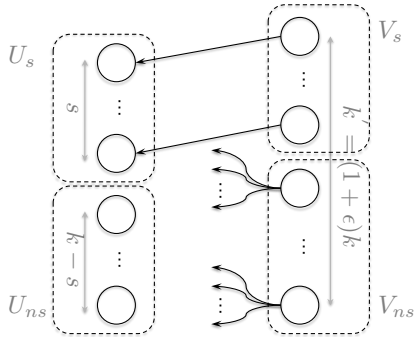

Fig. 2. Bipartite Graph $\mathcal{G}_{\mathcal{S}} = (U, V_{\mathcal{S}}, E_{\mathcal{S}})$ corresponding to $\mathbf{G}_{\mathcal{S}}$.

Lemma 2. *The bipartite graph $\mathcal{G}_{\mathcal{S}} = (U, V_{\mathcal{S}}, E_{\mathcal{S}})$ corresponding to the submatrix $\mathbf{G}_{\mathcal{S}}$ of $\mathbf{G}$ has a perfect matching with probability $1 - o(1)$ as $k \to \infty$.*

Proof: We establish an upper bound on the probability that a *perfect matching (P.M.)* between U and $V_{\mathcal{S}}$ does not exist. (Recall that, since $|U| < |V_{\mathcal{S}}|$, a P.M. is a matching that saturates all nodes in U). In particular, we require the probability to vanish asymptotically with a rate $1/\text{poly}(k)$ and show that the value $d(k) = O(\log k)$ used in constructing the bipartite graph suffices to achieve that.

Let V_s denote the subset of $V_{\mathcal{S}}$ corresponding to systematic encoded symbols ($0 \leq |V_s| \leq k$). We can assume that all nodes in V_s are used in the P.M.: each saturates a distinct node in U, leaving us with more options for saturating the remaining nodes in U. Since $k^{'} = (1+\epsilon)k > k$, $V_{\mathcal{S}}$ will definitely have a nonempty subset corresponding to parity (nonsystematic) symbols, denoted by $V_{ns} = V_{\mathcal{S}} \backslash V_s$. Let U_s denote the subset of U that is saturated by V_s and $U_{ns} = U \backslash U_s$. Since U_s and V_s are matched, *a P.M. between U and $V_{\mathcal{S}}$ exists if and only if a P.M. exists between U_{ns} and V_{ns}.*

The probability that a P.M. does not exist equals the probability that there exists a *contracting* set of nodes in U_{ns}. For simplicity, let $s \triangleq |V_s|$ and consequently, $|V_{ns}| = k^{'} - s$. Let E_i be the event that there exists a set of i nodes in U_{ns} that *contracts*, *i.e.*, has at most $i-1$ neighbours in V_{ns}. This is equivalent to at least $k^{'} - s - (i-1)$ nodes in V_{ns} being only adjacent to nodes in U_{ns} other than the i nodes of interest. Then,

$$\Pr\left(\nexists P.M. \text{ in } G_{\mathcal{S}}\right) = \Pr(\bigcup_{i=1}^{|U_{ns}|} E_i) \leq \sum_{i=1}^{|U_{ns}|} \Pr(E_i)$$
$$\leq \sum_{i=1}^{|U_{ns}|} \binom{|U_{ns}|}{i} \binom{|V_{ns}|}{|V_{ns}| - (i-1)} \left(\frac{k-i}{k}\right)^{d(k)(|V_{ns}|-(i-1))}$$
$$= \sum_{i=1}^{k-s} \binom{k-s}{i} \binom{k^{'}-s}{k^{'}-s-i+1} \left(\frac{k-i}{k}\right)^{d(k)\left(k^{'}-s-i+1\right)}$$
$$= \underbrace{\sum_{i=1}^{k-s} \binom{k-s}{k-s-i} \binom{k^{'}-s}{k^{'}-s-i+1} \left(\frac{k-i}{k}\right)^{d(k)\left(k^{'}-s-i+1\right)}}_{A}. \qquad (10)$$

Using the fact that $\binom{n}{k} \leq 2^{\left[nH\left(\frac{k}{n}\right)+\log_2(n)\right]}$, where $H(\cdot)$ is the binary entropy function, we have: $\binom{k-s}{k-s-i} \leq 2^{B^{(1)}_{k,s}(i)}$ and $\binom{k^{'}-s}{k^{'}-s-i+1} \leq 2^{B^{(2)}_{k,s}(i)}$ where

- $B^{(1)}_{k,s}(i) \triangleq (k-s)H\left(\frac{k-s-i}{k-s}\right) + \log_2(k-s)$
- $B^{(2)}_{k,s}(i) \triangleq (k^{'}-s)H\left(\frac{k^{'}-s-i+1}{k^{'}-s}\right) + \log_2(k^{'}-s)$

The entire sum can be bounded by $k-s$ times the largest among these bounds. In other words,

$$A \leq (k-s)\max_i \left(2^{\left[B^{(1)}_{k,s}(i)+B^{(2)}_{k,s}(i)+B^{(3)}_{k,s}(i)\right]}\right) \qquad (11)$$

where $B^{(3)}_{k,s}(i) \triangleq d(k)\left(k^{'}-s-i+1\right)\log_2\left(\frac{k-i}{k}\right)$. We require $\Pr\left(\nexists P.M. \text{ in } \mathcal{G}_{\mathcal{S}}\right)$ to vanish asymptotically faster than $\frac{1}{k^{\rho}}, \rho > 0$, for each value of $s \in \{0, \ldots, k\}$. It suffices to

require

$$(k-s)2^{\sum_{j=1}^{3} B_{k,s}^{(j)}(i)} \leq k^{-\rho}, \quad \forall \begin{array}{l} s \in \{0,\ldots,k\} \\ i \in \{1,\ldots,k-s\} \end{array}, \quad (12)$$

and equivalently,

$$\log_2(k-s) + \sum_{j=1}^{3} B_{k,s}^{(j)}(i) \leq -\rho \log_2(k)$$

which by expanding and rearranging terms leads to

$$d(k) \geq \frac{\left[\begin{array}{l} \rho\log_2(k) + 2\log_2(k-s) + \log_2\left(k'-s\right) \\ +(k-s)H\left(\frac{k-s-i}{k-s}\right) + (k'-s)H\left(\frac{k'-s-i+1}{k'-s}\right) \end{array}\right]}{-(k'-s-i+1)\log_2\left(\frac{k-i}{k}\right)}. \quad (13)$$

Let N and D denote the numerator and denumerator of the right hand side of inequality (13). In the following, we show that $\frac{N}{D} = O(\ln(k))$ and hence choosing $d(k) = c\ln(k)$ for some constant c suffices to satisfy (13) and achieve the vanishing probability.

- For the numerator, N, we have the following upper bound:

$$N \leq (3+\rho)\log_2(k) + \log_2(1+\epsilon) + kH\left(\frac{k-i}{k}\right) + (1+\epsilon)kH\left(\frac{(1+\epsilon)k-i+1}{(1+\epsilon)k}\right),$$

  where the inequality is due to the monotonicity of the logarithm and the fact that $f(x) = xH\left(\frac{x-y}{x}\right)$ is increasing with respect to x when $0 \leq y \leq x$.
- For the denumerator, D, we have:

$$D = \left(\underbrace{k-s-i}_{\geq 0} + \epsilon k + 1\right) \underbrace{\log_2\left(\frac{k}{k-i}\right)}_{\geq 0}$$

$$\geq \epsilon k \log_2\left(\frac{k}{k-i}\right) \overset{(\gamma)}{\geq} \epsilon k \frac{\epsilon}{\ln(2)} i, \quad (14)$$

  where (γ) is due to $\ln(1+x) > \frac{x}{x+1}$ for $x > -1, x \neq 0$ (here, $x = \frac{i}{k-i} > 0$).

We examine the ratio $\frac{N}{D}$ in parts:

(i) For the first part:

$$\frac{(3+\rho)\log_2(k) + \log_2(1+\epsilon)}{\epsilon k \log_2\left(\frac{k}{k-i}\right)} \overset{(14)}{\leq} \frac{\ln(2)}{\epsilon}(4+\rho)\log_2(k), \quad (15)$$

where the last inequality holds for $k > (1+\epsilon)$.

(ii) For the second part, expanding the entropy we obtain:

$$\frac{kH\left(\frac{k-i}{k}\right)}{\epsilon k \log_2\left(\frac{k}{k-i}\right)} = \frac{k-i}{\epsilon k} - \frac{\frac{i}{k}\log_2\left(\frac{i}{k}\right)}{\epsilon \log_2\left(\frac{k}{k-i}\right)}$$

$$\leq \frac{1}{\epsilon} + \frac{i\log_2\left(\frac{k}{i}\right)}{\epsilon k \frac{i}{k}} \leq \frac{2}{\epsilon}\log_2(k), \quad (16)$$

where the last inequality holds for $k \geq 2$.

(iii) For the third and final part, since for $i = 1$ the entropy in this term is zero, we need only consider $i \geq 2$:

$$\frac{(1+\epsilon)kH\left(\frac{(1+\epsilon)k-i+1}{(1+\epsilon)k}\right)}{\epsilon k \log_2\left(\frac{k}{k-i}\right)} \leq \frac{(1+\epsilon)}{\epsilon}\left(1 + \frac{i-1}{(1+\epsilon)k}\frac{\log_2\frac{(1+\epsilon)k}{i-1}}{\frac{i}{\ln(2)(1+\epsilon)k}}\right) \leq \frac{2\ln(2)(1+\epsilon)}{\epsilon}\log_2(k), \quad (17)$$

where the last inequality holds when $\log_2(k) \geq 1 + \ln(2)\log_2(1+\epsilon)$.

Combining (15), (16) and (17), it is evident that using $d(k) = c\log(k)$, where $c \propto 1/\epsilon$, suffices to make $\Pr(\nexists \text{P.M. in } \mathcal{G}_{\mathcal{S}}) = o(1)$ which completes the proof. ■

REFERENCES

[1] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *SIGCOMM Comput. Commun. Rev.*, vol. 28, pp. 56–67, October 1998. [Online]. Available: http://doi.acm.org/10.1145/285243.285258

[2] M. Luby, "Lt codes," in *Proceedings of the 43rd Symposium on Foundations of Computer Science*, ser. FOCS '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 271–. [Online]. Available: http://dl.acm.org/citation.cfm?id=645413.652135

[3] A. Shokrollahi, "Raptor codes," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 2551–2567, June 2006. [Online]. Available: http://dx.doi.org/10.1109/TIT.2006.874390

[4] O. Khan, R. Burns, J. Plank, W. Pierce, and C. Huang, "Rethinking erasure codes for cloud file systems: Minimizing i/o for recovery and degraded reads," in *Usenix Conference on File and Storage Technologies (FAST)*, 2012.

[5] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *Information Theory, IEEE Transactions on*, vol. 56, no. 9, pp. 4539 –4551, sept. 2010.

[6] F. Oggier and A. Datta, "Self-repairing homomorphic codes for distributed storage systems," in *INFOCOM, 2011 Proceedings IEEE*, april 2011, pp. 1215 –1223.

[7] D. Papailiopoulos, J. Luo, A. Dimakis, C. Huang, and J. Li, "Simple regenerating codes: Network coding for cloud storage," *Arxiv preprint arXiv:1109.0264*, 2011.

[8] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *Arxiv preprint arXiv:1106.3625*, 2011.

[9] P. Erdős and A. Rényi, "On random matrices," *Publ. Math. Inst. Hungar. Acad. of Sciences. 8*, 1964.

[10] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decentralized erasure codes for distributed networked storage," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 2809–2816, June 2006. [Online]. Available: http://dx.doi.org/10.1109/TIT.2006.874535

[11] R. Gummadi, "Coding and scheduling in networks for erasures and broadcast," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2011.

[12] D. Wiedemann, "Solving sparse linear equations over finite fields," *Information Theory, IEEE Transactions on*, vol. 32, no. 1, pp. 54 – 62, jan 1986.

[13] R. Motwani and P. Raghavan, "Algorithms and theory of computation handbook," M. J. Atallah and M. Blanton, Eds. Chapman & Hall/CRC, 2010, ch. Randomized algorithms, pp. 12–12. [Online]. Available: http://dl.acm.org/citation.cfm?id=1882757.1882769