Erasure statistics

Let there be m original chunks and k parity chunks, such that any m chunks out of the total $n = m + k$ ones are fully recoverable after the loss of any k of them. In the process of retrieving the n chunks, what is the likelihood of overall data corruption, given a per-chunk probability of error ϵ?

By "overall data corruption", we mean that more than k chunks are damaged in the data retrieval process. We assume that each chunk's probability of error is both equal to and independent of other chunks. In that case, the problem boils down to the independent drawing of n chunks, each of which undergo a Bernoulli trial of being faulty with probability ϵ. The total number of faulty chunks out of n independent Bernoulli trials is given by the binomial distribution:

$$B(i, n, \epsilon) = \binom{n}{i} \epsilon^k (1 - \epsilon)^{n-i}. \tag{1}$$

This expression is the probability mass function for the binomial distribution, yielding the probability that out of n chunks, exactly i will be faulty—assuming that the per-chunk probability of error is ϵ.

Since there are k parities out of the n chunks, the system can tolerate up to k chunk errors. The probability that no more than k errors accumulate can be expressed by summing Equation 1 over i up to k:

$$P(k, n, \epsilon) = \sum_{i=0}^{k} \binom{n}{i} \epsilon^k (1 - \epsilon)^{n-i}, \tag{2}$$

which is the cumulative distribution function of the binomial distribution.

The question we often want to answer is the following: given the number of chunks n and a security constant α such that that we want the overall probability of data corruption to be below this value, how many out of the n chunks should be parities? That is, in Equation 2 we are looking for the value of k which will make $P(k, n, \epsilon) = 1 - \alpha$ (Figure 1). This can be obtained by inverting the cumulative distribution function in k, resulting in the quantile function $Q(1 - \alpha, n, \epsilon)$. While this inverse has no convenient closed-form expression, it can be efficiently evaluated numerically for any set of input parameters.

Figures 2-4 summarize various aspects of the required parities and error rates.

Table 1 summarizes the number of chunks that are maintainable for a given number of parities k and per-chunk error rate ϵ such that the odds of data corruption stays below 1 in a million, i.e., $\alpha = 10^{-6}$. The first column of this table is the number of parities, followed by various per-chunk error rates. The entries in those subsequent columns are the number of chunks.

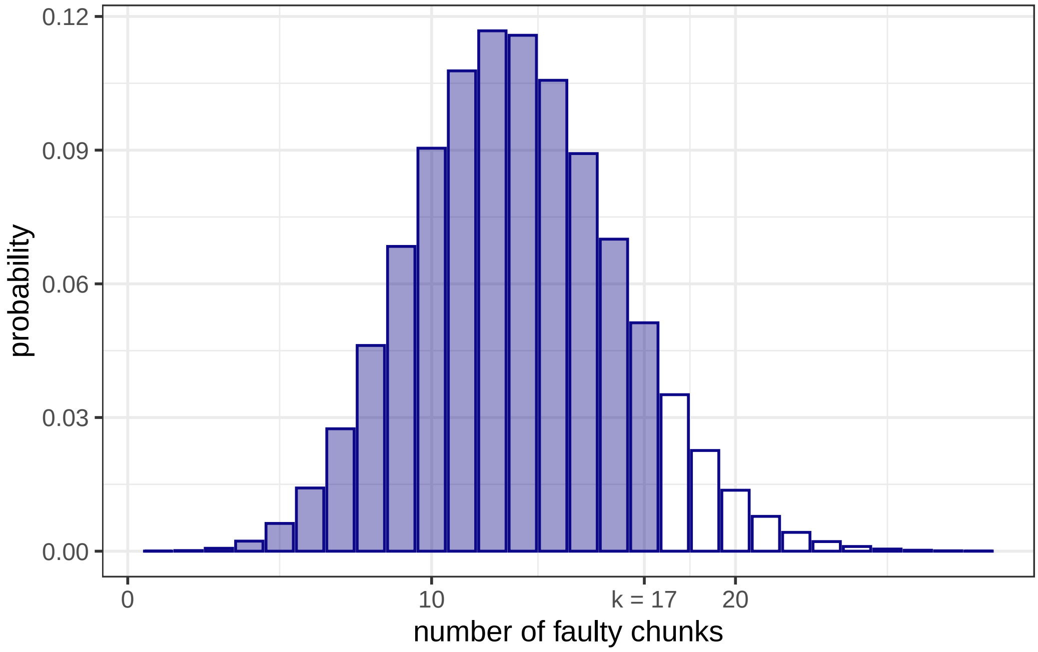

Figure 1: The point at $k = 17$ along the binomial distribution, where the probability of exceeding this many errors becomes less than $\alpha = 10\%$. Here the total number of chunks is $n = 128$, and the per-chunk error rate is $\epsilon = 0.1$.

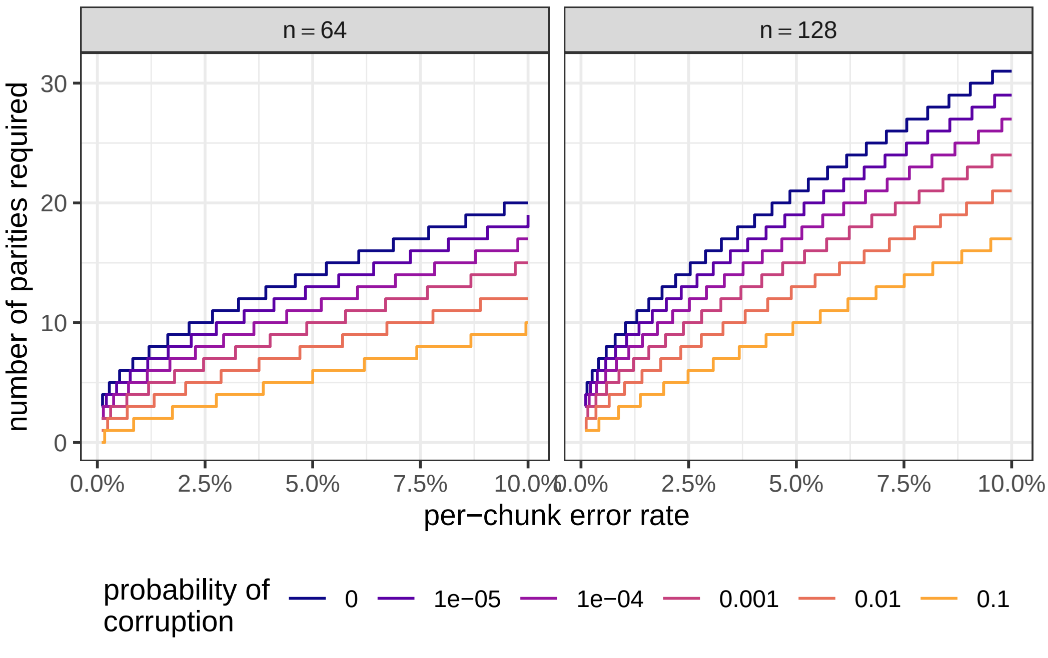

Figure 2: The number of parities needed (ordinate) as a function of the per-chunk error rate ϵ (abscissa), for keeping the probability of overall data corruption below given limits (colors).

Table 1: For a given number of parities (first column) and per-chunk error rate (subsequent columns), how many chunks can be supported to still have an overall data corruption probability less than $\alpha = 10^{-6}$? The number of chunks is the raw number, without the supporting parities.

	uniform and independent error rate of chunk retrieval

parities	1%	5%	10%	50%
1				
2				
3	1			
4	5			
5	14	1		
6	28	3	1	
7	46	6	2	
8	68	10	3	
9	94	15	5	
10		20	8	
11		26	10	
12		32	13	
13		39	16	
14		46	19	
15		53	22	
16		61	26	
17		69	29	
18		77	33	
19		86	37	
20		95	41	1
21		104	45	1
22			50	1
23			54	1
24			59	2
25			63	2
26			68	2
27			73	3
28			77	3
29			82	3
30			87	4
31			92	4
32				5
33				5
34				5
35				6
36				6
37				7
38				7
39				8
40				8
41				9
42				9
43				9
44				10
45				10
46				11
47				11
48				12
49				13
50				13

51	14
52	14
53	15
54	15
55	16
56	16
57	17
58	17
59	18
60	19
61	19
62	20
63	20
64	21
65	21
66	22
67	23
68	23
69	24
70	24
71	25
72	26
73	26
74	27
75	27
76	28
77	29
78	29
79	30
80	30
81	31
82	32
83	32
84	33
85	34
86	34
87	35
88	36
89	36
90	37
91	37

Table 2 is structured similarly, but for encrypted chunks. Each encrypted chunk takes up 2 slots, but the parity chunks still only use a single one. Thus, the number of effective chunk slots used is obtained as twice the number of chunks plus the number of parities.

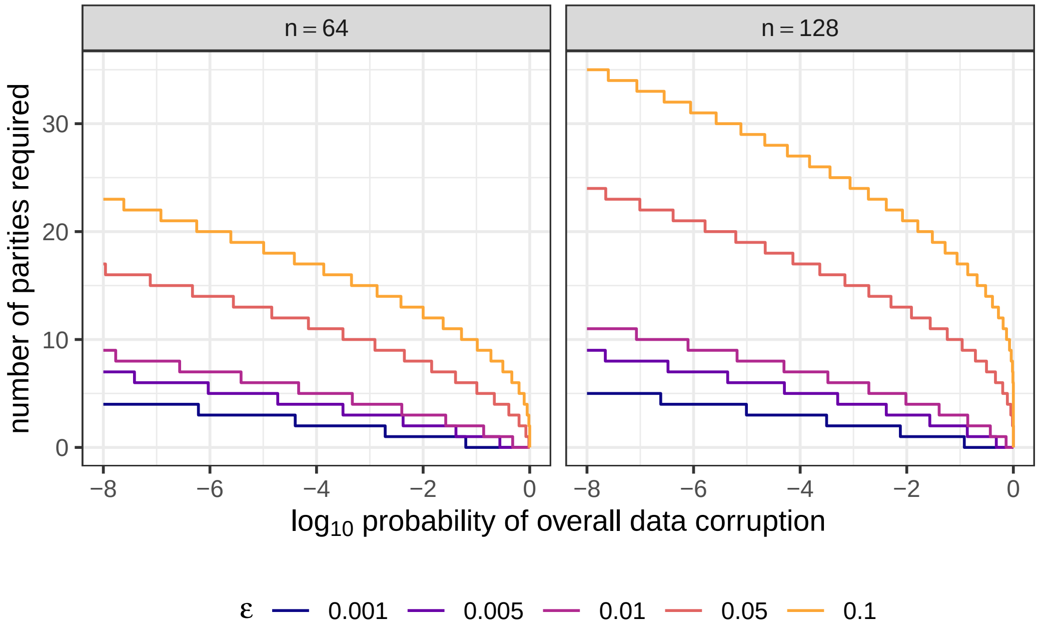

Figure 3: The number of parities required (ordinate) to keep the probability of overall data corruption at a given level (abscissa), for various values of the per-chunk error rate ϵ (colours) and for $n = 64$ chunks (left panel) and $n = 128$ chunks (right panel).

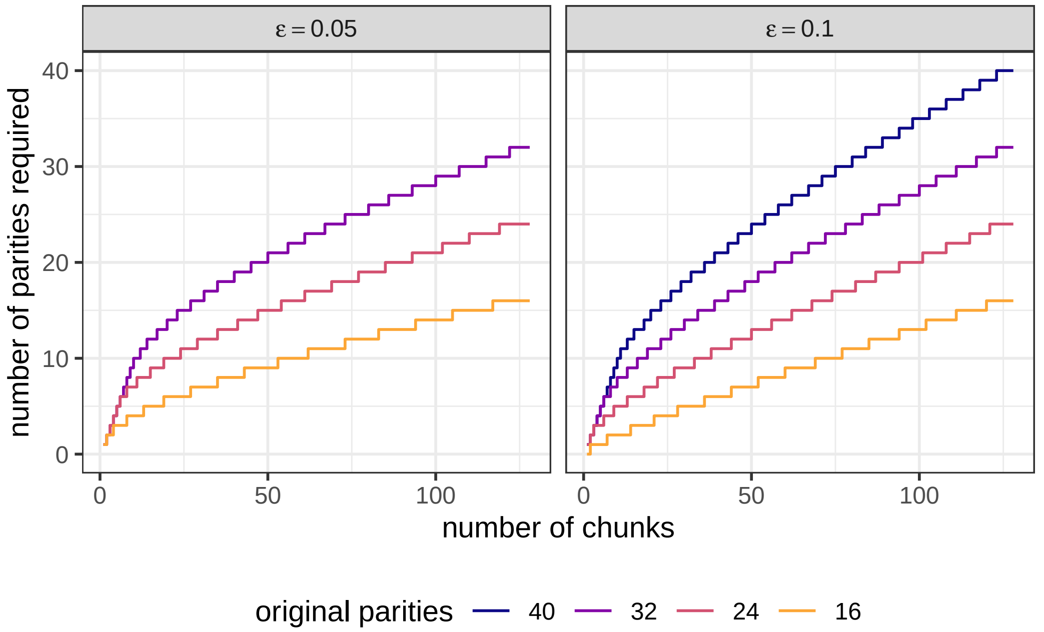

Figure 4: Number of chunks (abscissa) and the corresponding required number of parities (ordinate) such that will maintain the same overall probability of no data corruption as would be the case with 128 chunks, an original number of parities indicated by the colors, and a likelihood ϵ of an erroneous retrieval of a single chunk indicated in the panel headers. (These probabilities are listed in Table 1.)

Table 2: As Table 1, but for encrypted chunks. The maximum number of encrypted chunks is 64, so each encrypted chunks reference is a double segment (a hash reference, plus the segment-sized decryption key. The parity chunks on the other hand and parities extended the encrypted chunks, so they themselves do not need to b encrypted, so their reference is a single segment long Consequently, the number of effective chunk slots used is obtained as twice the number of chunks in any one of columns 2-5, plus the number of parities in column 1.

	uniform and independent error rate of chunk retrieval			
parities	1%	5%	10%	50%
4	2			
5	7			
6	14	1		
7	23	3	1	
8	34	5	1	
9	47	7	2	
10		10	4	
11		13	5	
12		16	6	
13		19	8	
14		23	9	
15		26	11	
16		30	13	
17		34	14	
18		38	16	
19		43	18	
20		47	20	
21		52	22	
22			25	
23			27	
24			29	1
25			31	1
26			34	1
27			36	1
28			38	1
29			41	1
30			43	2
31			46	2
32				2
33				2
34				2
35				3
36				3
37				3
38				3
39				4
40				4
41				4
42				4
43				4
44				5

45	5
46	5
47	5
48	6
49	6
50	6
51	7
52	7
53	7
54	7
55	8
56	8
57	8
58	8
59	9
60	9
61	9
62	10
63	10
64	10
65	10
66	11
67	11
68	11
69	12
70	12
71	12
72	13
73	13
74	13
75	13
76	14
77	14
78	14
79	15
80	15
81	15
82	16
83	16
84	16
85	17
86	17
87	17
88	18
89	18
90	18
91	18

Finally, an important special case is the required number of parities for just a single chunk (in which case the "parities" may as well be thought of as simple duplicates) to squeeze the probability of data corruption below a certain level α. For this special case, an explicit formula can be given.

If the probability of one of these parities being faulty is ϵ, then assuming independence, the probability that n parities are faulty is ϵ^n. Here we can write $n = k + 1$; that is, we have one "original" chunk and the rest of them are the k parities. Keeping the overall error probability below α then means that

$$\epsilon^{k+1} = \alpha \tag{3}$$

must be satisfied. Taking logarithms on both sides and rearranging, we get

$$k = \frac{\log(\alpha)}{\log(\epsilon)} - 1. \tag{4}$$

This is the number of replicas that a chunk without sisters (and without parent context) is required to have to keep the overall data corruption probability below α.

The base of the log in Equation 4 is arbitrary. This means that if we use base-10 logarithms and assume that $\alpha = 10^{-6}$, we get the simpler

$$k = \frac{6}{|\log_{10}(\epsilon)|} - 1. \tag{5}$$

For example, if the per-chunk error rate is ten percent ($\epsilon = 0.1$), then $|\log_{10}(\epsilon)| = |\log_{10}(1/10)| = 1$, and so $k = 6/1 - 1 = 5$ replicas are needed. If instead the per-chunk error rate is just one percent ($\epsilon = 0.01$), then only $k = 6/2 - 1 = 2$ parities are necessary. Overall, for the same per-chunk error rates as in Table 1, we get:

Table 3: For a given per-chunk error rate (first column), how many replicas (second column) are required of a single chunk to keep the overall data corruption probability below $\alpha = 10^{-6}$?

error rate	parities required
1%	2
5%	4
10%	5
50%	19