# Efficient Detection of Corruption in a Replicated File Using a Reed Solomon Decoder

Ada Waichee Fu [1] and Siu-Cheung Chau [2]

[1] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong
Email: adafu@cs.cuhk.hk

[2] Department of Mathematics and Computer Science, University of Lethbridge, Canada
Email: chau@cs.uleth.ca

*Abstract* — **Techniques of error correcting codes is used to generate and manipulate combined signatures for detecting corruption in a replicated file. Based on a realistic assumption, we derived a distributed algorithm which achieves better performance than previously known results.**

## I. Introduction

When a data file is replicated at one or more sites in a distributed computer network, it is of interest to discover corruptions in the file copies by making comparison among the multiple physical copies of the file. An efficient method is as follows. For each page copy, a *page signature* is computed. From the page signatures, *combined signatures* are computed and transmitted among sites. The corruptions are detected by some computations on a set of combined signatures.

We consider the case where a file is replicated among a set of $M$ sites $SS = \{s_1, ..., s_M\}$. We assume that the file contains $N$ pages, $\{P_1, ..., P_N\}$, and the copy of page $P_i$ at site $s_j$ is $P_{i,j}$. We assume decentralized control so that any site can communicate with any other site. As in most previous papers, we assume that the total number of corrupted page copies will not exceed $f$.

One way to compute the combined signatures is based on the theory of an error-correcting code over some finite field and is introduced in [**?**, **?**]. In particular, a method using Reed-Solomon codes [**?**] has been suggested in [**?**] for problems with only 2 sites. [**?**] extended the method of Reed-Solomon codes and solves the problem for $M$ replicated sites, where $M \geq 3$.

Although [**?**] solves a more general problem for $M$ copies of the file, the following assumption is made.

**Assumption A**: page signatures for corrupted copies of a page are identical. □

We argue that the chance that page signatures for corrupted copies of a page are identical is very small. One can instead assume that the page signatures for any 2 corrupted copies of the same page will not be identical. Here we show that this can lead to a more efficient algorithm. Our assumption is the following:

**Assumption B**: The probability that 2 copies of the same page are corrupted and that their page signatures are identical is negligible. □

## II. Technique of Error Correcting Code

We shall adopt the combined signature techniques as in [**?**, **?**, **?**]. In this method, each site $s_A$ calculates a page signature $S(P_{i,A})$ for each of its page copies $P_{i,A}$. For this method to work, it must be the case that $N < 2^b$, where $b$ is the number of bits of a page signature. Since there are $2^b$ possible values of page signatures, each page signature can be considered as an element in a finite field (Galois field) $GF(2^b)$. From these page signatures, $s_A$ can further compute **combined signatures** of $b$ bits.

Suppose a site $s_A$ with a file copy $F_A = < P_{1,A}, ..., P_{N,A} >$ computes $k$ combined signatures. These combined signatures are given by

$$sig_{j,A} = \sum_{i=1}^{N} \alpha^{ij} S(P_{i,A}) \quad \text{for } j = 0, 1, ..., k-1,$$

where $\alpha$ is a primitive element in $GF(2^b)$ and the addition and multiplication operations are those of $GF(2^b)$. Hence the elements $\alpha^j$, for $1 \leq j \leq 2^b - 1$, are distinct. There are $2^b$ possible values for a combined signature. We refer to the signatures $sig_{j,A}$ for $j = 0, 1, 2, ...k-1$, where $k \geq 1$, as the **first** $k$ combined signatures.

To help explain the algorithm in the next section, we first describe a few properties of the combined signatures. In the algorithm we shall compare two sets of combined signatures from two different sites $s_A$ and $s_B$. In a special case, suppose we compute the first $N$ combined signatures in each site, and we know that one of the sites, say $s_A$, is correct. The two sets of signatures are given by

$$sig_{j,A} = \sum_{i=1}^{N} \alpha^{ij} S(P_{i,A}) \quad \text{for } j = 0, 1, ..., N-1$$
$$sig_{j,B} = \sum_{i=1}^{N} \alpha^{ij} S(P_{i,B}) \quad \text{for } j = 0, 1, ..., N-1$$

If site $s_A$ sends its first $N$ combined signatures to site $s_B$, then $s_B$ can compute the difference between $sig_{j,A}$ and $sig_{j,B}$ for $j = 0, 1, ..., N-1$. The difference between the two sets is given by

$$sig_{j,A} - sig_{j,B} = \sum_{i=1}^{N} \alpha^{ij} (S(P_{i,A}) - S(P_{i,B})) \qquad (1)$$

for $j = 0, 1, ..., N-1$.

Equation (1) can be written in a matrix form:

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha^1 & \alpha^2 & \cdots & \alpha^N \\ \alpha^2 & \alpha^4 & \cdots & \alpha^{2N} \\ \vdots & \vdots & & \vdots \\ \alpha^{(N-1)} & \alpha^{2(N-1)} & \cdots & \alpha^{N(N-1)} \end{pmatrix} \begin{pmatrix} S(P_{1,A}) - S(P_{1,B}) \\ S(P_{2,A}) - S(P_{2,B}) \\ \vdots \\ S(P_{N,A}) - S(P_{N,B}) \end{pmatrix}$$
$$= \begin{pmatrix} sig_{0,A} - sig_{0,B} \\ sig_{1,A} - sig_{1,B} \\ \vdots \\ sig_{N-1,A} - sig_{N-1,B} \end{pmatrix}$$

In the above, the left most matrix is a **Vandermonde matrix**. Theorem 7.2.1 of [?] says that the above Vandermonde matrix has a nonzero determinant.

**Observation 1**: Given the first $N$ combined signatures $sig_{i,C}$ of a correct sites $s_C$, for $i = 0, ...N-1$, and the first $N$ combined signatures $sig_{i,A}$ of a site $s_A$, one can solve the following set of equations to locate all corruptions in $s_A$.

$$\sum_{n=1}^{N} e_n \alpha^{nj} = sig_{j,C} - sig_{j,A}, \quad for j = 0, ...N-1 \qquad (2)$$

This has also been remarked in [?]. Next we have the following remark.

**Remark 2** [?]: Suppose that $(e_1, ..., e_N)$ has at most $v$ non-zero entries. If $\sum_{n=1}^{N} e_n \alpha^{jn} = 0$, where $j = 1, ..., v$, then $e_n = 0$ for $n = 1, ..., N$. □

The left hand side of the equations in Remark 2 can be written as

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha^1 & \alpha^2 & \cdots & \alpha^N \\ \vdots & \vdots & & \vdots \\ \alpha^{v-1} & \alpha^{2(v-1)} & \cdots & \alpha^{N(v-1)} \end{pmatrix} \begin{pmatrix} S(P_{1,A}) - S(P_{1,B}) \\ S(P_{2,A}) - S(P_{2,B}) \\ \vdots \\ S(P_{N,A}) - S(P_{N,B}) \end{pmatrix}$$

Therefore, if we compare the first $t$ combined signatures for $s_A$ and $s_B$ and find that $sig_{j,A} = sig_{j,B}$ for $1 \leq j \leq t$, then the signatures $S(P_{i,A})$ and $S(P_{i,B})$ are identical for all $i$'s, and therefore the page copies $P_{i,A}$ and $P_{i,B}$ are also identical. Since we assume (Assumption B) that two page copies of the same page that are corrupted are not identical, it can only be the case that sites $s_A$ and $s_B$ are both correct.

**Observation 2**: If there can be at most $v$ errors at 2 sites $s_A$ and $s_B$, and the first $v$ combined signatures for $s_A$ and $s_B$ are identical, then sites $s_A$ and $s_B$ are both correct, otherwise, at least one of the sites is incorrect. □

Given that we have found a correct site, from Observation 1, one can identify the corrupted page copies if the first $N$ combined signatures are computed. However, if we know that the maximum number of corrupted page copies is not large, we can compute and transmit less combined signature. In order to do this, we note the following remark.

**Remark 3**: [?] For any given $E_0, ..., E_{J-1} \in GF(2^b)$, where $1 \leq J \leq N$, the system of equations $\sum_{n=1}^{N} e_n \alpha^{jn} = E_j$ for $j = 0, ..., J-1$, has at most one solution for $(e_1, ..., e_N)$ with less than or equal to $\lfloor J/2 \rfloor$ non-zero entries. This solution can be obtained using a Reed-Solomon decoder [?]. □

In the above, only part of a Reed-Solomon decoder need to be utilized since we are interested in identifying the locations of errors rather than in correcting the combined signatures for corrupted copies.

**Observation 3**: If we have the first $J$ combined signatures of a correct site $s_A$ and another site $s_B$, and we know that for $s_B$ there can be at most $\lfloor J/2 \rfloor$ corrupted page copies, then there is a unique solution for the system of $J$ equations as described in Remark 3, corresponding to the 2 set of combined signatures. The solution identifies the locations of all the corruptions in $s_B$.

## III. Distributed Algorithm

We consider how to reduce the amount of message savings. We shall follow assumptions of [?] except for Assumption A, which is replaced by Assumption B. We derive Algorithm A. We assume $M > 2f$. There are 2 cases:

**Case 1:** For $N \leq f$, we compare the $N$ page signatures between 2 sites. Any two page signatures that do not agree indicates one or two corrupted copies. At the end of the comparison of all groups, one must be able to locate at least one correct site since there must be at least one group that contain two correct sites. The amount of signature transmission is given by $\lceil M/2 \rceil N$. Let $FG$ be the set of groups that are found to contain some corruption. Suppose there are $F$ groups in $FG$. In the second phase, the correct site sends its page copies to locate the the corruptions in each of the groups in $FG$. The maximum number of signatures sent for this purpose is bounded by by $FN$. Hence the total number of signature transmission is $(\lceil M/2 \rceil + F)N$.

**Case 2:** For $N > f$, we have a procedure as shown in Figure 1. Most of the computation in Figure 1 are based on the local view of a site $s_{i1}$. The lines marked with ** in Figure 1 are where global computation or global consideration of some kind is made. In Case 2, Algorithm A is made up of 2 phases.

1. Phase 1: partition the set of sites into groups of size 2 each, find a correct site by doing comparison between the 2 sites in each group.

**ALGORITHM A**:

**Given**: a set of sites SS.

**Prep** Partition SS into groups of of size 2.

Let the partitioning be $\{G_1, \ldots G_k\}$. ($G_i = \{ s_{i1}, s_{i2} \}$ for $1 \leq i \leq k$)

The followings are operations at site $s_{i1}$ of $G_i$.

**Phase 1: Find a correct site**

(1) SEND( REQUEST ( $sig_{1,i2}, \ldots sig_{f,i2}$ ) ) to $s_{i2}$.

(3) RECEIVE ($sig_{1,i2}, \ldots sig_{f,i2}$) from $s_{i2}$ ................(R1')

(4) **If** $sig_{k,i1} = sig_{k,i2}$ $for$ $1 \leq k \leq f$ **Then** $msg_i$ = "Correct"
**Else** $msg_i$ = "Incorrect"

(5) SEND( $msg_i$ ) to $s_{j1}$ of all other groups $G_j$.

(6) RECEIVE( $msg_j$ ) from $s_{j1}$ of all other groups $G_j$.

The followings are operations at site $s_{i1}$ of $G_i$.

**Phase 2: Find corrupted page copies**

(1) **....Let $FG$ be the set of groups $G_j$ with $msg_j$ = 'Incorrect'

(2) **.....Let $F$ be the number of groups in $FG$.

(3) **If** $msg_i$ = 'Correct' (The sites in this group are correct);

(4) **Then**

(5) If chosen by Protocol A, SEND the first $f$ combined signatures to some sites

(6) Wait for REQUEST messages and SEND signatures according to requests.

(7) **Else**

(8) There is a correct site $s_c$ chosen by some protocol for $G_i$.

(9) RECEIVE ($sig_{1,c}, \ldots sig_{f,c}$) from site $s_c$ ...............(R2')

(10) **If** $sig_{k,i2} = sig_{k,c}$ $for$ $1 \leq k \leq f$
**Then** $d = i2$ (Site $s_{i2}$ is correct)

(11) **If** $sig_{k,i1} = sig_{k,c}$ $for$ $1 \leq k \leq f$
**Then** $d = i1$ (Site $s_{i1}$ is correct)

(12) **Else** $d = c$ (both sites $s_{i1}$ and $s_{i2}$ are incorrect)

(13) **For** $m = i1$ and $m = i2$

(14) **If** it is not true that $sig_{k,m} = sig_{k,d}$ $for$ $1 \leq k \leq f$

(15) **Then** (The site is corrupted)

(16) Compute a vector $(e_{1,m}, \ldots, e_{N,m})$ of at most $min\{N, f/2\}$ non-zero elements

(17) as a solution to the system of equations

(18) $\sum_{n=1}^{N} e_{n,m} \alpha^{jn} = sig_{j,m} - sig_{j,d}$,
where $j = 1, .., f$ (*)

(19) **....If $f = N$ or (each site has $\leq f/2$ errors),
then all corrupted pages are found

(20) **....Otherwise, at most one site $s_w$ has more than $f/2$ errors, $s_w$ has no solution in the above.

(21) **....Let $f'' = \min\{N, 2(f - G)\}$ where $G$ is the number of corrupted page copies found so far

(22) **If** $s_w$ exists and is in this group

(23) **Then**

(24) **If** $s_{i1}$ is correct **Then** $s_d = s_{i1}$ **Else** $s_d = s_c$

(25) **If** $s_w = s_{i2}$

(26) **Then**

(27) SEND( REQUEST ($sig_{f+1,i2}, \ldots sig_{f'',i2}$)) to site $s_{i2}$

(28) RECEIVE ($sig_{f+1,i2}, \ldots sig_{f'',i2}$) from site $s_{i2}$ ......(R3')

(29) **If** $s_d \neq s_{i1}$

(30) **Then**

(31) SEND( REQUEST ($sig_{f+1,d}, \ldots sig_{f'',d}$) to site $s_d$

(32) RECEIVE ($sig_{f+1,d}, \ldots sig_{f'',d}$) from site $s_d$ ..........(R4')

(33) Compute a vector $(e_{1,w}, \ldots, e_{N,w})$ with at most $f$ non-zero

(34) elements as a solution to the system of equations

(35) $\sum_{n=1}^{N} e_{n,w} \alpha^{jn} = sig_{j,w} - sig_{j,d}$,
where $j = 1, .., f''$ (*)

(36) **....**For** m = 1 ,..., M

(37) **.... $C_m = \{P_{n,m} | 1 \leq n \leq N$ $and$ $e_{n,m} \neq 0\}$

(38) **....( Note: $C_m$ is the set of corrupted pages in site $s_m$.)

Fig. 1: Algorithm A

2. Phase 2: A set of groups $FG$ will be found to contain corruptions. It will receive some signatures from a correct site found in Phase 1 to detect corruptions.

In the first phase, we try to locate one or more correct site, as well as discover a number of sites which are correct and hence need not be considered in phase 2. We do this by partitioning the set of sites into groups of 2 sites each. The reason for comparing 2 sites is that by Observation 2, the comparison of 2 sites can discover correct sites if both the sites are correct, and that a set of 2 sites is the smallest possible set of sites to detect a correct file copy. More messaging is needed if we compare signatures in groups of more than 2 sites. Suppose we have groups of 3 sites, the number of sets of signatures to be sent in total is $\lceil \frac{2}{3}M \rceil$, which is greater than $\lceil \frac{1}{2}M \rceil$ for the case with groups of size 2.

Phase one is where Assumption B leads to a significant gain in minimizing the signature transmission compared to Assumption A. With Assumption A, in order to locate a correct site, one must find a majority of sites whose first $k$ combined signatures are identical, where $k$ is the number of possible errors at a site. In the worst case one would have to collect the combined signatures from every site before the majority is found. The number of signature transmissions is given by $(M-1)k$. With Assumption B, one can find a correct site by comparing 2 sites at a time, and hence in Phase one of Algorithm A, only $\frac{M}{2}k$ signature transmissions are necessary.

Observations 1, 2 and 3 ensure that the corrupted copies will be located.

## IV. Performance of Algorithm A

The total number of combined signature transmissions is bounded by

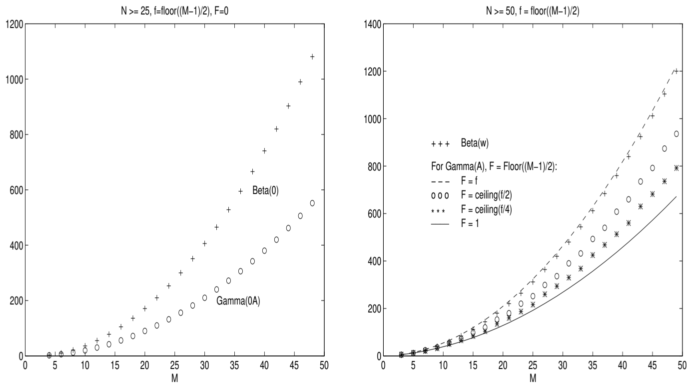
$$\gamma^A(M, N, f, F) = \left(\lceil \frac{M}{2} \rceil + F\right) \min\{N, f\} + 2 \min\{\max\{N - f, 0\}, f\}$$

If there is no failure, then the number of combined signature transmissions is given by

$$\gamma_0^A(M, N, f) = \left\lceil \frac{M}{2} \right\rceil \min\{N, f\}$$

Let us compare the performance of Algorithm A with the bounds achieved by [?] (**Algorithm Z**). The number of signature transmission it requires is given by $\beta_w(M, N, f)$ for cases with failure and $\beta_0(M, N, f)$ for cases with no failure.

The comparisons are shown in Figures 2 and 3. In these figures, $\gamma_0^A(M, N, f)$ is denoted by $Gamma(0A)$, $\gamma^A(M, N, f, F)$ is denoted by $Gamma(A)$, $\beta_w(M, N, f)$ is denoted by $Beta(w)$, and $\beta_0(M, N, f)$ is denoted by $Beta(0)$. The figures show that Algorithm A has better performance.




(a) (b)

Fig. 2: Signature transmission of Algorithm A and Z

## References


[1] K.A.S. Abdel-Ghaffar and A. El Abbadi. Efficient detection of corrupted pages in a replicated file. In *12th ACM Symposium on Principles on Distributed Computing*, pages 219–227, 1993.

[2] K.A.S. Abdel-Ghaffar and A. El Abbadi. An optimal strategy for comparing file copies. *IEEE Transaction on Parallel and Distributed Systems*, pages 87–93, Jan 1994.

[3] R.E. Blahut. *Theory and Practice of Error Control Codes*. Addison Wesley, Reading, MA, 1984.

[4] J.J. Metzner. Reliable and efficient broadcast of files to a group of locally interconnected stations. *Proc. GLOBECOM '86*, pages 1762–1767, Dec 1986.

[5] J.J. Metzner and Kapturowski. A general decoding technique applicable to replicated file disagreement location and concatenated code decoding. *IEEE Transaction on Inform. Theory*, pages IT–36:911–917, July 1990.