Reed-Solomon Coding: Variated Redundancy and Matrix Formalism

Aleksei V. Marov
RAIDIX
St. Petersburg, Russia
e-mail: Marov.A@raidix.com

Alexei Yu. Uteshev
Faculty of Applied Mathematics
St. Petersburg State University
St. Petersburg, Russia
e-mail: alexeiuteshev@gmail.com

ABSTRACT

We aim to construct a version of the Reed-Solomon coding procedure which admits an easy extension when the number of checksums has to be increased due to the rise of the expected error rate.

Keywords

Reed-Solomon codes, Berlekamp-Welch algorithm.

1. INTRODUCTION

Reed-Solomon codes are widely used in data transmission and storage systems. Many papers cover theoretical foundations and various aspects of applications of these codes. The present article is focused to answer the following particular question: assuming that the redundancy in the coding scheme is to be variated on the go, how to manage the efficient coding and error correcting procedures?

The proposed approach can be treated as the development of the Berlekamp-Welch algorithm [1]. However, compared with that algorithm reducing the error correction problem to that of rational interpolation, we suggest a different procedure for the computation of the error locator polynomial. This approach is based on our recent investigation on solving the interpolation problems using the appropriate Hankel polynomials [3].

We also discuss here the problem of efficient computation of the operations in finite fields, namely how the arithmetic in $\mathbf{GF}(2^s)$ can be implemented using the basic field $\mathbf{GF}(2)$. For this aim, we convert the operations from a polynomial to a matrix formalism and detail the structure of the obtained matrices.

Notation. We treat the Galois field $\mathbf{GF}(2^s)$ with the generating polynomial $f(x) := x^s + f_1x^{s-1} + \ldots + f_s \in \mathbf{GF}(2)[x]$ and with $a \in \mathbf{GF}(2^s)$ standing for a primitive element. For the sequence of distinct elements $\{x_1, \ldots, x_K\} \subset \mathbf{GF}(2^s)$, we set

$$W(x) := \prod_{\ell=1}^{K}(x - x_\ell)$$

and define the basic interpolation polynomials as follows

$$\widetilde{W}_j(x) := \frac{W_j(x)}{W_j(x_j)} \quad \text{where } W_j(x) := \frac{W(x)}{x - x_j},\ j = \overline{1, K}.$$

We will also use the version $\widetilde{W}_j(x; \{x_\ell\}_{\ell=1}^K)$ when the generating sequence is to be distinguished.

2. REED-SOLOMON ALGORITHM

Given the data blocks $\{D_{j-1}\}_{j=1}^n \subset \mathbf{GF}(2^s)$, we first compute the checksum blocks $\{C_{i-1}\}_{i=1}^m \subset \mathbf{GF}(2^s)$ via systematic coding. For this aim, compose first the polynomial

$$F(X) := D_0X^{n-1} + D_1X^{n-2} + \ldots + D_{n-1}$$

and compute the remainder on division of $X^mF(X)$ by $g(X) := \prod_{i=1}^m (X + a^{i-1})$; the coefficients of this remainder

$$C_0X^{m-1} + C_1X^{m-2} + \ldots + C_{m-1}$$

are taken as checksum blocks. The codeword is then considered in the form

$$(Y_0, Y_1, \ldots, Y_{N-1})$$

$$:= (D_0, D_1, \ldots, D_{n-1}, C_0, C_1, \ldots, C_{m-1})$$

with $N := n + m$. The division operation involved in syndrome computation can be replaced by the matrix multiplication

$$(C_0, C_1, \ldots, C_{m-1})^\top = \mathbf{K}_1\,(D_0, D_1, \ldots, D_{n-1})^\top \qquad (1)$$

using the so called *coding matrix* [2]

$$\mathbf{K}_1 := [\widetilde{W}_i(a^{N-j}; \{a^{m-\ell}\}_{\ell=1}^m)]_{i=\overline{1,m},\ j=\overline{1,n}} \qquad (2)$$

$$= \left(\begin{array}{cccc} \widetilde{W}_1(a^{N-1}) & \widetilde{W}_1(a^{N-2}) & \cdots & \widetilde{W}_1(a^m) \\ \widetilde{W}_2(a^{N-1}) & \widetilde{W}_2(a^{N-2}) & \cdots & \widetilde{W}_2(a^m) \\ \vdots & \vdots & & \vdots \\ \widetilde{W}_m(a^{N-1}) & \widetilde{W}_m(a^{N-2}) & \cdots & \widetilde{W}_m(a^m) \end{array}\right)$$

Here the basic interpolation polynomials $\{\widetilde{W}_\ell(X)\}_{\ell=1}^m$ are generated by the sequence $\{a^{m-1}, a^{m-2}, \ldots, 1\}$. Having the matrix $\mathbf{K}_1$ precomputed for storage, one can organize the syndrome evaluation for any information block vector. In [2] a procedure is suggested for parallelization of computing the polynomials $\left\{\widetilde{W}_i(X)\right\}_{i=1}^m$. We now mention just only one property of the matrix (2): sum of the entries of any of its column equals 1.

A sequence

$$\left(\widehat{Y}_0, \widehat{Y}_1, \ldots, \widehat{Y}_{N-1}\right) \qquad (3)$$

is taken as a true codeword iff the values of the polynomial

$$\widehat{G}(X) := \widehat{Y}_0X^{N-1} + \widehat{Y}_1X^{N-2} + \ldots + \widehat{Y}_{N-1}$$

at $\{a^{j-1}\}_{j=1}^{m}$:

$$\widehat{G}(1), \widehat{G}(a), \dots, \widehat{G}(a^{m-1})$$

(named syndromes) are all zero. If any of these syndromes is not zero then an error is detected. Assuming that the number E of errors in (3) does not exceed $\lfloor m/2 \rfloor$, the error locator polynomial

$$\left| \begin{array}{llll} \widehat{G}(1) & \widehat{G}(a) & \dots & \widehat{G}(a^E) \\ \widehat{G}(a) & \widehat{G}(a^2) & \dots & \widehat{G}(a^{E+1}) \\ \vdots & \vdots & & \vdots \\ \widehat{G}(a^{E-1}) & \widehat{G}(a^E) & \dots & \widehat{G}(a^{2E-1}) \\ 1 & X & \dots & X^E \end{array} \right| \tag{4}$$

possesses the zeros

$$a^{N-j_1}, a^{N-j_2}, \dots, a^{N-j_E}$$

with $j_1, j_2, \dots, j_E$ standing for the positions of corrupted blocks in the sequence (3).

3. ALTERNATIVE CODING SCHEME

Generate the basic interpolation polynomials $\{\widetilde{W}_k(X)\}_{k=1}^{n}$ by the sequence $a_1, a_2, \dots, a_n$ of arbitrary distinct elements of $\mathbf{GF}(2^s)$ and compose the polynomial

$$L(X) := D_0 \widetilde{W}_1(X) + D_1 \widetilde{W}_2(X) + \dots + D_{n-1} \widetilde{W}_n(X)$$

which is just the Lagrange interpolation polynomial of a degree $\le n-1$ satisfying the conditions $\{L(a_j) = D_{j-1}\}_{j=1}^{n}$. This time we define the checksums as the values of $L(X)$ at m extra distinct elements $a_{n+1}, \dots, a_N$ of $\mathbf{GF}(2^s)$:

$$C_{m-1} = L(a_{n+1}), C_{m-2} = L(a_{n+2}), \dots, C_0 = L(a_N) \,. \tag{5}$$

This means the coding redundancy is organized by extending the interpolation table: the number of polynomial values exceeds that of its coefficients. The formulas (5) can be rewritten into the matrix form as

$$(C_0, C_1, \dots, C_{m-1})^\top = \mathbf{K}_2 \, (D_0, D_1, \dots, D_{n-1})^\top \tag{6}$$

using the coding matrix

$$\mathbf{K}_2 := [\widetilde{W}_j(a_{n+i}; \{a_\ell\}_{\ell=1}^{n})]_{i=\overline{1,m},\ j=\overline{1,n}} \tag{7}$$

$$= \left(\begin{array}{llll} \widetilde{W}_1(a_{n+1}) & \widetilde{W}_2(a_{n+1}) & \cdots & \widetilde{W}_n(a_{n+1}) \\ \widetilde{W}_1(a_{n+2}) & \widetilde{W}_2(a_{n+2}) & \cdots & \widetilde{W}_n(a_{n+2}) \\ \vdots & \vdots & & \vdots \\ \widetilde{W}_1(a_N) & \widetilde{W}_2(a_N) & \cdots & \widetilde{W}_n(a_N) \end{array} \right) .$$

Compared with the coding matrix (2), the matrix (7) possesses the property with regard to the entries of its rows: sum of them equals 1 for any row.

The procedure of testing the sequence (3) to be a coding one goes as follows: we compose the polynomial $W(X) := \prod_{\ell=1}^{N} (X + a_\ell)$ and compute the values

$$\tau_k := \frac{\widehat{Y}_0 a_1^k}{W'(a_1)} + \frac{\widehat{Y}_1 a_2^k}{W'(a_2)} + \dots + \frac{\widehat{Y}_{N-1} a_N^k}{W'(a_N)} \tag{8}$$

for $k = \overline{0, m-1}$.

Theorem 1. *If any of the equalities*

$$\tau_0 = 0, \tau_1 = 0, \dots, \tau_{m-1} = 0$$

fails then the error is detected.

Proof. Let

$$\widehat{L}(x) := L_0 X^{N-1} + L_1 X^{N-2} + \dots + L_{N-1}$$

be the polynomial such that $\{\widehat{L}(a_j) = \widehat{Y}_{j-1}\}_{j=1}^{N}$. Then

$$\tau_k = \sum_{j=1}^{N} \frac{\widehat{L}(a_j) a_j^k}{W'(a_j)} = L_0 \sigma_{N+k-1} + \dots + L_{N-1} \sigma_k$$

where

$$\sigma_i := \sum_{j=1}^{N} a_j^i / W'(a_j), \ i = \overline{0, N+m-1} \,.$$

The Euler-Lagrange equalities

$$\sigma_i = \left\{ \begin{array}{lll} 0 & \text{if} & i = \overline{0, N-2}, \\ 1 & \text{if} & i = N-1 \end{array} \right. \tag{9}$$

lead to the chain of relations

$$\tau_0 = L_0, \ \tau_1 = L_0 \sigma_N + L_1, \ \tau_2 = L_0 \sigma_{N+1} + L_1 \sigma_N + L_2, \dots$$

wherefrom it follows that $\deg \widehat{L}(x) \le n-1$ (i.e., a degree of the new interpolation polynomial does not exceed the original estimation) iff the condition of the theorem is fulfilled.

Theorem 2. *Assuming that the number E of errors in (3) does not exceed $\lfloor m/2 \rfloor$, the error locator polynomial*

$$\left| \begin{array}{llll} \tau_0 & \tau_1 & \dots & \tau_E \\ \tau_1 & \tau_2 & \dots & \tau_{E+1} \\ \vdots & \vdots & & \vdots \\ \tau_{E-1} & \tau_E & \dots & \tau_{2E-1} \\ 1 & X & \dots & X^E \end{array} \right| \tag{10}$$

possesses zeros

$$a_{j_1}, a_{j_2}, \dots, a_{j_E}$$

with $j_1 - 1, j_2 - 1, \dots, j_E - 1$ standing for the positions of corrupted blocks in the sequence (3).

Proof of the general statement is contained in [3]. The underlying idea is outlined here with the case $E = 1$. Let $n < N-2$ and the polynomial $\widehat{L}(x)$ be such that $\widehat{L}(a_j) = L(a_j) = Y_{j-1}$ for $j = \overline{2, N}$ but $\widehat{L}(a_1) = \widehat{Y}_0 \ne Y_0$. Set $\varepsilon := Y_0 - \widehat{Y}_0$. Then

$$\tau_k = \left(\frac{\varepsilon a_1^k}{W'(a_1)} + \frac{\widehat{Y}_0 a_1^k}{W'(a_1)} \right) + \frac{\widehat{Y}_1 a_2^k}{W'(a_2)} + \dots + \frac{\widehat{Y}_{N-1} a_N^k}{W'(a_N)}$$

$$= \frac{\varepsilon a_1^k}{W'(a_1)} + \sum_{j=1}^{N} \frac{L(a_j) a_j^k}{W'(a_j)} = \frac{\varepsilon a_1^k}{W'(a_1)}$$

for $k = 0$ and $k = 1$ due to the equalities (9). Therefore, one has

$$\left| \begin{array}{ll} \tau_0 & \tau_1 \\ 1 & X \end{array} \right| = \left| \begin{array}{cc} \dfrac{\varepsilon}{W'(a_1)} & \dfrac{\varepsilon a_1}{W'(a_1)} \\ 1 & X \end{array} \right| = \frac{\varepsilon}{W'(a_1)} (X - a_1) \,.$$

At first glance, the proposed scheme does not have any advantage over the classical one recalled in Section 2 since both matrices (2) and (7) are of the same order. However, the utility of using the coding matrix (7) versus (2) appears when the number m of checksums should be enlarged because of the increasing failure rate, i.e., when it happens that $E > \lfloor m/2 \rfloor$. While the increase in the number of checksums by 1 causes, besides the appearance of an extra row in both matrices, the modification of any entry of the matrix (2), the entries of the matrix (7) remain unchanged. Indeed, we just compute an extra value for the polynomial $L(X)$. For recalculation of the values (8) when the interpolation table is extended by one element, we suggest the following result:

Theorem 3. Let

$$W(x) := \prod_{\ell=1}^{K}(x - x_\ell), \breve{W}(X) := \prod_{\ell=1}^{K+1}(x - x_\ell)$$

and

$$\tau_j := \sum_{\ell=1}^{K} \frac{y_\ell x_\ell^j}{W'(x_\ell)}, \ \breve{\tau}_j := \sum_{\ell=1}^{K+1} \frac{y_\ell x_\ell^j}{\breve{W}'(x_\ell)} .$$

Then the following relationship is valid

$$\breve{\tau}_j = \sum_{\ell=1}^{j} \tau_{j-\ell} x_{K+1}^{\ell-1} + x_{K+1}^{j} \breve{\tau}_0 .$$

Proof follows from the equality:

$$\tau_j = \breve{\tau}_{j+1} - x_{K+1}\breve{\tau}_j .$$

4. MULTIPLICATION IN GF: POLYNOMIALS VS MATRICES

Our next aim is to utilize the ambiguity in the choice of the elements $\{a_j\}_{j=1}^N$ for optimizing the structure of the matrix (7). To do this, we first intend to reduce the arithmetic in $\mathbf{GF}(2^s)$ to that in $\mathbf{GF}(2)$.

Multiplication of the elements $(c_0, c_1, \dots, c_{s-1})$ and $(b_0, b_1, \dots, b_{s-1})$ of $\mathbf{GF}(2^s)$ is traditionally introduced with the aid of polynomial multiplication, i.e., the product $(p_0, p_1, \dots, p_{s-1})$ is obtained as a result of the modular 2 remainder computation

$$\sum_{j=1}^{s} p_{j-1} x^{s-j}$$

$$\equiv \underbrace{\left(\sum_{j=1}^{s} c_{j-1} x^{s-j}\right)}_{:=C(x)} \underbrace{\left(\sum_{j=1}^{s} b_{j-1} x^{s-j}\right)}_{:=B(x)} \pmod{f(x)}$$

with $f(x)$ standing for the generating polynomial of the field. In order to translate this operation into the matrix multiplication over $\mathbf{GF}(2)$ we compute successively the remainder polynomials on division of

$$B(x), xB(x), x^2B(x), \dots, x^{s-1}B(x)$$

by $f(x)$:

$$\Big\{ b_{k0}x^{s-1} + b_{k1}x^{s-2} + \dots + b_{k,s-1}$$

$$\equiv x^k B(x) \pmod{f(x)} \Big\}_{k=0}^{s-1} .$$

Compose the matrix from the rows of coefficients of these remainders

$$\mathbf{B} := [b_{s-1-k,\ell}]_{k,\ell=0}^{s-1} \tag{11}$$

i.e., with the order of the rows being bottom-up.

Theorem 4. One has:

$$(p_0, p_1, \dots, p_{s-1})^\top = \mathbf{B}^\top (c_0, c_1, \dots, c_{s-1})^\top .$$

Proof follows from the congruences

$$\sum_{j=1}^{s} p_{j-1} x^{s-j} \equiv C(x)B(x) \pmod{f(x)}$$

$$\equiv \left[\sum_{j=1}^{s} c_{j-1} x^{s-j} B(x)\right] \pmod{f(x)}$$

$$\equiv \sum_{j=1}^{s} c_{j-1} \left[x^{s-j}B(x) \pmod{f(x)}\right] .$$

Remark. Matrix (11) computed for polynomials $B(x)$ and $C(x)$ over arbitrary (not necessarily finite) field is known as a matrix of *Bézout's representation* of the *resultant* of these polynomials [4].

Now the coding process given by (6) can be carried into the $\mathbf{GF}(2)$ both with the the data blocks $\{D_j\}_{j=0}^{n-1}$ treated as vectors, and the entries of the coding matrix (7) treated as matrices.

5. SPARSE CODING MATRIX

We have experimented with the choice of $\{a_j\}_{j=1}^N$ in order to generate maximally sparse coding matrices (7), i.e., matrices with maximal number of zeros when represented in $\mathbf{GF}(2)$. Some results are given below.

Example 1. $\mathbf{GF}(2^6), f(x) = x^6 + x + 1, n = 16, m = 4$. *Sample size* 20000 *matrices.*

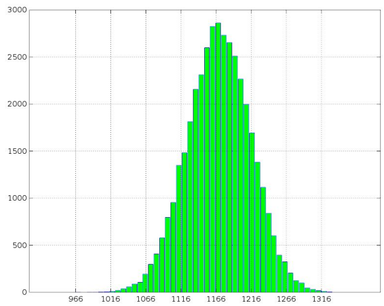

Figure 1. Histogram for the units distribution: Example 1

The maximally sparse coding matrix is selected for the following values of a_j represented in decimals

j	1	2	3	4	5	6	7	8	9	10
a_j	32	39	53	46	59	30	52	41	25	7
j	11	12	13	14	15	16	17	18	19	20
a_j	26	56	5	38	45	48	43	1	49	13

If first represented in decimals

$$\begin{pmatrix} 33 & 56 & 33 & 43 & 40 & 53 & 8 & 10 & 57 & 3 & 24 & 51 & 36 & 56 & 55 & 55 \\ 24 & 62 & 32 & 57 & 20 & 34 & 35 & 22 & 12 & 48 & 3 & 41 & 49 & 16 & 52 & 62 \\ 3 & 57 & 5 & 8 & 4 & 1 & 55 & 2 & 17 & 1 & 40 & 48 & 11 & 30 & 15 & 20 \\ 9 & 7 & 37 & 37 & 53 & 6 & 29 & 53 & 12 & 24 & 63 & 8 & 4 & 32 & 17 & 2 \end{pmatrix}$$

and then with every entry replaced by the corresponding binary matrix (11) of the order 6*, the resulting coding* *matrix over* $\mathbf{GF}(2)$ *contains* 966 *units. Compared with the median at* 1152 *units, the economy is more than* 16%.

Example 2. $\mathbf{GF}(2^7), f(x) = x^7 + x^3 + 1, n = 16, m = 4$. *Sample size* 40000 *matrices.*

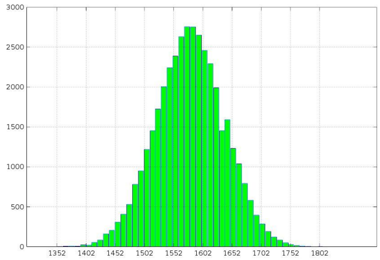

Figure 2. Histogram for the units distribution: Example 2

The maximally sparse coding matrix is detected that contains 1352 *units. Compared with the median at* 1568 *units, the economy is about* 14%.

6. CONCLUSIONS

We have developed a coding scheme for Reed-Solomon code which effectively allows to raise the redundancy in the number of checksums without modifying the already precalculated values.

Both approaches, i.e., the traditional one from Section 2 and a new one presented in 3, result in construction of the error locator polynomial in the form of Hankel polynomial (formulas (4) and (10)). For further investigation, it is necessary to compose an efficient algorithm for recomputing these polynomials when an additional checksum is involved into the error correction process.

The structure of the coding matrix (7) is also subject to optimization in order to make it maximally sparse by a suitable choice of parameters.

7. ACKNOWLEDGEMENTS

While working on this research, the second author was supported by the RFBR according to the research project No 17-29-04288.

REFERENCES

[1] L.R. Welch, E.R. Berlekamp "Error correction for algebraic block codes", US Patent No 4 633 47, Dec. 30, 1986.

[2] A.V. Marov, A.Yu. Uteshev "Matrix formalism of the Reed-Solomon codes", *Vestnik of St.Petersburg State University. Series 10.*, issue 4., pp. 4-17, 2016.

[3] A.Yu. Uteshev, I. Baravy "Solution of interpolation problems via the Hankel polynomial construction". *arXiv:* cs.SC/1603.08752. 2016. 56 p.

[4] P. Bikker,A.Yu. Uteshev "On the Bézout construction of the resultant", *J.Symb.Comput.*, V. 28, No 1, pp.45-88, 1999.