The Roadmap from Polynomials to Quantum-safe Cryptosystems

A perspective from discrete mathematics
Part 3/4: Reed-Muller codes and their decoding

Chunlei Li (University of Bergen, Norway)

INCP2-2024/10213 on
Mathematical Theory of Data Transmission and Data Encryption
Oct. 6-10, 2025, Tromsø
Fall school on Geometry in Cryptography and Communication

Outline[1]

1. Reed-Muller (RM) Codes

2. Encoding and Decoding

[1] The materials can be found in [1, Chapter 8]

Reed-Muller (RM) Codes

Overiew

- introduced by Muller in 1954
- Reed shortly proposed a decoding algorithm with error-correcting capability up to $\lfloor \frac{d-1}{2} \rfloor$
- had been used to transmit the black and white Mariner images (later replaced by Golay codes for transmitting color images)
- RM codes have a flavour of polarization, an idea adopted in Polar codes that are used in the 5G standard

Boolean functions

An m-variable Boolean function is a map from $\mathbb{F}_2^m$ to $\mathbb{F}_2$ given by

$$f(x_0, x_1, \cdots, x_{m-1}) = \sum_{s=0}^{m-1} \sum_{0 \le i_1 < i_2 \cdots, < i_s \le m-1} a_{i_1, i_2, \cdots, i_s} x_{i_1} x_{i_2} \cdots x_{i_s}$$

or by a truth table

$$[f(\mathbf{0}), f(\mathbf{1}), \ldots, f(\mathbf{2^m - 1})],$$

where $\mathbf{i}$ is a column vector of the binary representation of the integer i, $0 \le i \le 2^m - 1$.

The *algebraic degree* of f is

$$\deg(f) = \max\{\deg(x_{i_1} x_{i_2} \cdots x_{i_s}) \,|\, a_{i_1, i_2, \cdots, i_s} \neq 0 \le s < m\}$$

Example.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
x_0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Let $f(x_0, x_1, x_2, x_3) = 1 + x_0 + x_2 + x_0x_1 + x_1x_2x_3$. The truth table

$$c_f = (f(\mathbf{0}), f(\mathbf{1}), \ldots, f(\mathbf{15})) = (1100110100111101).$$

and the algebraic degree of f is $\deg(x_1x_2x_3) = 3$.

Definition

The binary r-th order Reed-Muller code of length $n = 2^m$ is:

$$RM(r, m) = \{c_f = (f(\mathbf{0}), f(\mathbf{1}), \ldots, f(\mathbf{2^m - 1})) \mid \deg(f) \leq r\}.$$

Example (m=4)

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x_0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
x_0x_1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
x_0x_2	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
x_0x_3	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
x_1x_2	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
x_1x_3	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1
x_2x_3	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
$x_0x_1x_2$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
$x_0x_1x_3$	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
$x_0x_2x_3$	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
$x_1x_2x_3$	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
$x_0x_1x_2x_3$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Generator Matrices

$RM(0, m)$: row 1; $RM(1, m)$: row 1-5; $RM(2, m)$: row 1-11; $RM(3, m)$: row 1-15; $RM(4, m)$: all 16 rows.

Properties of RM codes

The recursive relation:

$$RM(0,m) \subset RM(1,m) \subset RM(2,m) \subset \cdots \subset RM(m,m)$$

For any Boolean function $f(x_0,\ldots,x_{m-2},x_{m-1})$ with $\deg(f) \le r$,

- $f(x_0,\ldots,x_{m-1}) = f_1(x_1,\ldots,x_{m-2}) + x_{m-1}f_2(x_1,\ldots,x_{m-2})$.
- If $\deg(f) \le r$, then $\deg(f_1) \le r$ and $\deg(f_2) \le r-1$.
- $f(x_0,\ldots,x_{m-2},0) = f_1(x_0,\ldots,x_{m-2})$ and
  $f(x_0,\ldots,x_{m-2},1) = f_1(x_0,\ldots,x_{m-2}) + f_2(x_0,\ldots,x_{m-2})$

Recursive Relation

$$RM(r,m) = \{(u, u+v) \mid u \in RM(r,m-1), v \in RM(r-1,m-1)\}$$

Parameters

For the r-th order binary RM codes $RM(r, m)$, we have

- the dimension of RM(r,m) code: $k = \binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{r}$
- the minimum distance $d = 2^{m-r}$

Proof on Dimension.

Each monomial $x_{i_1} \dots x_{i_t}$ with $1 \leq t \leq r$ is a row in the generator matrix G of $RM(r, m)$. Together with the constant (all-one row), there are

$$\binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{r}$$

rows in G. This gives the dimension of $RM(r, m)$.

Proof of Min. Distance.

- Recall that for the $(u, u+v)$ construction C of codes C_1, C_2, the minimum distance of C satisfies

$$d = \min\{2d_1, d_2\}.$$

With the recursive relation,

$$RM(r,m) = \{(u, u{+}v) \mid u \in RM(r, m{-}1), v \in RM(r{-}1, m{-}1)\}$$

One has

$$d(RM(r,m)) = \min\{2d(RM(r, m-1)), d(RM(r-1, m-1))\}.$$

Note that for any integer m',

$$d(RM(m', m')) = 1 \text{ and } d(RM(0, m')) = 2^{m'}.$$

By induction the result can be obtained.

Dual of RM codes

The dual code of $RM(r, m)$ is $RM(m - r - 1, m)$.

- **Dimension**.

$$\sum_{i=0}^{r} \binom{m}{i} + \sum_{j=0}^{m-(r+1)} \binom{m}{j} = \sum_{i=0}^{r} \binom{m}{i} + \sum_{i=r+1}^{m} \binom{m}{i} = 2^m$$

- **Orthogonality**. For any $b \in \{0, 1, \ldots, 2^m - 1\}$,

$c_{f*g}(b) = 1 \iff (f*g)(b) = f(b)g(b) = 1 \iff c_f(b) = c_g(b) = 1.$

Therefore, one has $\langle c_f, c_g \rangle = wt(c_{f*g}) \pmod 2$. Because $\deg(f * g) \leq m - 1$, c_{f*g} is a codeword of $RM(m - 1, m)$, in which all codewords have even weight. This implies

$$\langle c_f, c_g \rangle = 0.$$

Encoding and Decoding

Encoding

Recall that the dimension of $RM(r, m)$ is

$$k = \sum_{i=0}^{r} \binom{m}{i}.$$

For any message $\mathbf{a}$ of length k, we can take each coordinate of $\mathbf{a}$ as the coefficient for a monomial in the Boolean function f and the truth table of f will be the corresponding codeword for $\mathbf{a}$.

Example. $r = 1$.

- $k = \binom{m}{0} + \binom{m}{1} = m + 1$
- the monomials are

$$1, x_0, x_1, \ldots, x_{m-1}$$

- for a message $\mathbf{a} = (a_0, a_1, \ldots, a_m)$, the corresponding Boolean functions is

$$f = a_0 + a_1 x_0 + a_2 x_1 + \cdots + a_m x_{m-1}$$

- the codeword is

$$c_f = (f(\mathbf{0}), f(\mathbf{1}), \ldots, f(\mathbf{2}^m - \mathbf{1})) = (1100110100111101).$$

Decoding of RM codes

- there are many decoding approaches for RM codes
- we will take a look at one approach by majority strategy to recover the Boolean function

$$f(x_0, x_1, \cdots, x_{m-1}) = \sum_{s=0}^{r} \sum_{0 \le i_1 < i_2 \cdots, < i_s \le m-1} a_{i_1, i_2, \cdots, i_1} x_{i_1} x_{i_2} \cdots x_{i_s}$$

- the decoding methods for $RM(r, m)$ are in general complex
- we start with the simpler case $RM(1, m)$

$$RM(1, m) = \{c_f \mid f = a_0 + a_1 x_0 + a_2 x_1 + \cdots + a_m x_{m-1},\ a_i \in \mathbb{F}_2\}.$$

Decoding $RM(1, m)$

For a codeword in

$$RM(1, m) = \{c_f \mid f = a_0 + a_1 x_0 + a_2 x_1 + \cdots + a_m x_{m-1},\ a_i \in \mathbb{F}_2\},$$

there are 2^m equations in $a_0, a_1, \ldots, a_m$.

If there is some errors in a received word $\mathbf{y} = (y_0, \ldots, y_{2^m-1})$.

One can determine the codeword based on the *majority decoding strategy*.

Example

Decode the $RM(1,3)$ code with majority strategy.

The generator matrix of $RM(1,3)$ is given by

$$\begin{array}{c|cccccccc} & \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} & \mathbf{7} \\ g_0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ g_1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ g_2 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ g_3 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}$$

The vector g_0, g_1, g_2, g_3 are the basis of the generator matrix. The function of any codeword is given by

$$\mathbf{c} = a_0 g_0 + a_1 g_1 + a_2 g_2 + a_3 g_3$$

This gives the codeword as

$$\begin{aligned}(\quad & a_0, a_0 + a_3, a_0 + a_2, a_0 + a_2 + a_3, \\ & a_0 + a_1, a_0 + a_1 + a_3, a_0 + a_1 + a_2, a_0 + a_1 + a_2 + a_3).\end{aligned}$$

If no error occurs in received word

$$\begin{aligned}\mathbf{y} &= (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)\\ &= (a_0, a_0 + a_3, a_0 + a_2, a_0 + a_2 + a_3,\\ &\qquad a_0 + a_1, a_0 + a_1 + a_3, a_0 + a_1 + a_2, a_0 + a_1 + a_2 + a_3)\end{aligned}$$

one has

$$\begin{aligned}a_1 &= y_0 + y_4 = y_1 + y_5 = y_2 + y_6 = y_3 + y_7\\ a_2 &= y_0 + y_2 = y_1 + y_3 = y_4 + y_6 = y_5 + y_7\\ a_3 &= y_0 + y_1 = y_2 + y_3 = y_4 + y_5 = y_6 + y_7\end{aligned}$$

- If one error has occurred in $\mathbf{y}$, then all the calculations above are made, 3 of 4 values will agree for each a_i, so the correct valued will be obtained by majority decoding.
- Finally a_0 can be determined by the majority of the components of $\mathbf{y} + a_1 g_1 + a_2 g_2 + a_3 g_3$

Fast Decoding of the 1-st order Reed-Muller codes

First order RM codes can be efficiently decoded using a fast Hadamard transform. This can be efficiently done in 3 steps:

1. Build the 2^m-order Hadamard matrix.
2. Apply Binary Phase Shift Keying on the received word r.
3. Compute its Walsh coefficients.

The Hadamard matrix of order 2^m is defined as

$$H_{2^m} = H_2 \otimes H_{2^{m-1}} = \begin{bmatrix} H_{2^{m-1}} & H_{2^{m-1}} \\ H_{2^{m-1}} & -H_{2^{m-1}} \end{bmatrix} \text{ with } H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

Actually this recursion helps achieve **fast transform** and drop the complexity from $O(2^m \times 2^m)$ to $O(m2^m)$.

Fast Decoding of the 1-st order Reed-Muller codes

First order RM codes can be efficiently decoded using a fast Hadamard transform. This can be efficiently done in 3 steps:

1. Build the 2^m-order Hadamard matrix.
2. Apply Binary Phase Shift Keying on the received word r.
3. Compute its Walsh coefficients.

The Hadamard matrix of order 2^m is defined as

$$H_{2^m} = H_2 \otimes H_{2^{m-1}} = \begin{bmatrix} H_{2^{m-1}} & H_{2^{m-1}} \\ H_{2^{m-1}} & -H_{2^{m-1}} \end{bmatrix} \text{ with } H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

Actually this recursion helps achieve **fast transform** and drop the complexity from $O(2^m \times 2^m)$ to $O(m2^m)$.

Fast Decoding of the 1-st order Reed-Muller codes

First order RM codes can be efficiently decoded using a fast Hadamard transform. This can be efficiently done in 3 steps:

1. Build the 2^m-order Hadamard matrix.
2. Apply Binary Phase Shift Keying on the received word r.
3. Compute its Walsh coefficients.

The Hadamard matrix of order 2^m is defined as

$$H_{2^m} = H_2 \otimes H_{2^{m-1}} = \begin{bmatrix} H_{2^{m-1}} & H_{2^{m-1}} \\ H_{2^{m-1}} & -H_{2^{m-1}} \end{bmatrix} \text{ with } H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

Actually this recursion helps achieve **fast transform** and drop the complexity from $O(2^m \times 2^m)$ to $O(m2^m)$.

Example for $m = 3$**.** The generator matrix for the $RM(1,3)$ code is

$$G = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{array}{c} 1 \\ x_1 \\ x_2 \\ x_3 \end{array}$$

and the 8-order Hadamard matrix is

$$H_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & - & 1 & - & 1 & - & 1 & - \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & 1 & 1 & 1 & - & - & - & - \\ 1 & - & 1 & - & - & 1 & - & 1 \\ 1 & 1 & - & - & - & - & 1 & 1 \\ 1 & - & - & 1 & - & 1 & 1 & - \end{bmatrix}$$

Example for $m = 3$**.** The generator matrix for the $RM(1,3)$ code is

$$\begin{array}{cc} G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} & \begin{array}{c} 1 \\ x_1 \\ x_2 \\ x_3 \end{array} \end{array}$$

and the 8-order Hadamard matrix is

$$H_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & - & 1 & - & 1 & - & 1 & - \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & 1 & 1 & 1 & - & - & - & - \\ 1 & - & 1 & - & - & 1 & - & 1 \\ 1 & 1 & - & - & - & - & 1 & 1 \\ 1 & - & - & 1 & - & 1 & 1 & - \end{bmatrix}$$

Example for $m = 3$.

Binary Phase Shift Keying: we assign a phase to each bit r_i of the received word. For the binary case this is a map $F : \{0,1\} \to \{-1,1\}$ as

$$F(r_i) = (-1)^{r_i}.$$

The vector $\mathbf{w}$ of its Walsh coefficients are computed by $\mathbf{w} = \mathbf{r}H_8$.

Example for $m = 3$.

Binary Phase Shift Keying: we assign a phase to each bit r_i of the received word. For the binary case this is a map $F : \{0, 1\} \to \{-1, 1\}$ as

$$F(r_i) = (-1)^{r_i}.$$

The vector $\mathbf{w}$ of its Walsh coefficients are computed by $\mathbf{w} = \mathbf{r}H_8$.

Let's consider r is a valid codeword associated to the polynomial x_1. Then $\mathbf{r} = (0, 0, 0, 0, 1, 1, 1, 1)$.

Its BPSK representation is $(1, 1, 1, 1, -1, -1, -1, -1)$.

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}^\top \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & - & 1 & - & 1 & - & 1 & - \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & 1 & 1 & 1 & - & - & - & - \\ 1 & - & 1 & - & - & 1 & - & 1 \\ 1 & 1 & - & - & - & - & 1 & 1 \\ 1 & - & - & 1 & - & 1 & 1 & - \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 8 \\ 0 \\ 0 \\ 0 \end{bmatrix}^\top$$

Let's consider r is a valid codeword associated to the polynomial x_1. Then $\mathbf{r} = (0,0,0,0,1,1,1,1)$.

Its BPSK representation is $(1,1,1,1,-1,-1,-1,-1)$.

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}^\top \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & - & 1 & - & 1 & - & 1 & - \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & 1 & 1 & 1 & - & - & - & - \\ 1 & - & 1 & - & - & 1 & - & 1 \\ 1 & 1 & - & - & - & - & 1 & 1 \\ 1 & - & - & 1 & - & 1 & 1 & - \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 8 \\ 0 \\ 0 \\ 0 \end{bmatrix}^\top$$

Fast Decoding of the 1-st order Reed-Muller codes

Let's consider one error in $\mathbf{r} = (0,0,0,0,1,1,1,0)$.

Its BPSK representation is $(1,1,1,1,-1,-1,-1,1)$.

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ 1 \end{bmatrix}^{\top} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & - & 1 & - & 1 & - & 1 & - \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & 1 & 1 & 1 & - & - & - & - \\ 1 & - & 1 & - & - & 1 & - & 1 \\ 1 & 1 & - & - & - & - & 1 & 1 \\ 1 & - & - & 1 & - & 1 & 1 & - \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ 7 \\ 1 \\ 1 \\ -1 \end{bmatrix}^{\top}$$

This strategy is again the **majority decoding**.

Fast Decoding of the 1-st order Reed-Muller codes

Let's consider one error in $\mathbf{r} = (0, 0, 0, 0, 1, 1, 1, 0)$.

Its BPSK representation is $(1, 1, 1, 1, -1, -1, -1, 1)$.

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ 1 \end{bmatrix}^\top \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & - & 1 & - & 1 & - & 1 & - \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & 1 & 1 & 1 & - & - & - & - \\ 1 & - & 1 & - & - & 1 & - & 1 \\ 1 & 1 & - & - & - & - & 1 & 1 \\ 1 & - & - & 1 & - & 1 & 1 & - \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ 7 \\ 1 \\ 1 \\ -1 \end{bmatrix}^\top$$

This strategy is again the **majority decoding**.

Fast Decoding of the 1-st order Reed-Muller codes

Let's consider one error in $\mathbf{r} = (0, 0, 0, 0, 1, 1, 1, {\color{red}0})$.

Its BPSK representation is $(1, 1, 1, 1, -1, -1, -1, {\color{red}1})$.

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ {\color{red}1} \end{bmatrix}^\top \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & - & 1 & - & 1 & - & 1 & - \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & 1 & 1 & 1 & - & - & - & - \\ 1 & - & 1 & - & - & 1 & - & 1 \\ 1 & 1 & - & - & - & - & 1 & 1 \\ 1 & - & - & 1 & - & 1 & 1 & - \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ 7 \\ 1 \\ 1 \\ -1 \end{bmatrix}^\top$$

This strategy is again the **majority decoding**.

Decoding of $RM(r, m)$-Overview

Let $f \in RM(r, m)$ and a codeword $c_f = (f(x))_{x \in \mathbb{F}_2^m}$ is transmitted. Suppose there are at most $\left\lfloor \frac{2^{m-r}-1}{2} \right\rfloor$ errors that occurs in the received word c_g.

- First, determine the coefficients of highest degree c_f in f.
  - It is possible to find 2^{m-r} equations to determine each of these coefficients by majority decoding.
- Next, determine the coefficients of next highest degree $r - 1$ in f.
  - It is possible to find 2^{m-r+1} equations to determine each of these coefficients by majority decoding.
- Continue this way to find all coefficients of f.

Finding degree-r terms in $f \in RM(r, m)$

$$f\left(x_0, x_1, \cdots, x_{m-1}\right) = \sum_{s=0}^{r} \sum_{0 \le i_1 < i_2 \cdots, < i_s \le m-1} a_{i_1, i_2, \cdots, i_1} x_{i_1} x_{i_2} \cdots x_{i_s}$$

For the term $a_{m-r, m-r+1, \cdots, m-1} x_{m-r} x_{m-r+1} \cdots x_{m-1}$, its coefficient can be determined from the following lemma.

Lemma

There are 2^{m-r} equations to determine $a_{m-r, m-r+1, \cdots, m-1}$ given by

$$a_{m-r, m-r+1, \cdots, m-1} = c_f \cdot c_{(x_0+u_0)(x_1+u_1)\cdots(x_{m-r-1}+u_{m-r-1})}$$

for $u_0, u_1, \cdots, u_{m-r-1} \in \{0, 1\}$.

Proof. Given $u = (u_0, \cdots, u_{m-r-1}) \in GF(2)^r$, define

$$g_u(x) = (x_0 + u_0)(x_1 + u_1) \cdots (x_{r-1} + u_{m-r-1}).$$

Write f as $f(x) = a_{m-r,m-r+1,\cdots,m-1} x_{m-r} x_{m-r+1} \cdots x_{m-1} + f_1(x)$ and consider

$$f(x)g_u(x) = a_{m-r,m-r+1,\cdots,m-1} x_{m-r} x_{m-r+1} \cdots x_{m-1} g_u(x) + f_1(x)g_u(x)$$

where $\deg(f_1 g_u) < m$. Observe that

$$g_u(x) x_{m-r} x_{m-r+1} \cdots x_{m-1} = 1 \text{ iff } x = (u_0+1, \ldots, u_{m-r-1}+1, 1, \ldots, 1)$$

and $wt(f_1 g_u) \equiv 0 \bmod 2$ since $\deg(f_1 g_u) < m$. This implies

$$a_{m-r,m-r+1,\cdots,m-1} \equiv wt(fg_u) \bmod 2 = c_f \cdot c_{g_u} \bmod 2$$

Let $f \in RM(r,m)$ be transmitted and let c_f be its characteristic vector. Suppose at most $t = 2^{m-r-1} - 1 = \left\lfloor \frac{d-1}{2} \right\rfloor$ errors and we receive $r = c_f + e$, where e is the error pattern of weight at most t.

Let

$$g_u(x) = (x_0 + u_0) \cdots (x_{m-r-1} + u_{m-r-1}), u \in \{0,1\}^r.$$

We will find 2^{m-r} equations in $a_{m-r,m-r+1,\cdots,m-1}$.

Step 1: Compute $r \cdot g_u$ for $u_0, \cdots, u_{m-r-1} \in \{0,1\}$. If no errors these 2^{m-r} checks are all equal to $a_{m-r,m-r+1,\cdots,m-1}$. The errors means we only get an estimate of $a_{m-r,m-r+1,\cdots,m-1}$.

Step 2: Compute $a_{m-r,m-r+1,\cdots,m-1}$ as majority of the values of the 2^{m-r} parity checks.

Remark

The parity checks $g_u(x)$ checks disjoint positions. (Because $g_u(x)$ checks positions where $g_u(x) = 1$ and these positions are disjoint for different values of u).

Each error therefore changes only one parity-check. Since there are 2^{m-r} parity-checks a majority will give the value $a_{m-r,m-r+1,\cdots,m-1}$ since there are at most $\left\lfloor \frac{2^{m-r}-1}{2} \right\rfloor$ errors.

References i

T. K. Moon.
Error Correction Coding: Mathematical Methods and Algorithms.
John Wiley & Sons, 2nd edition, 2020.