On MacWilliams type identities for \( r \)-fold joint \( m \)-spotty weight enumerators

Anuradha Sharma *, Amit K. Sharma

Department of Mathematics, Indian Institute of Technology Delhi, New Delhi 110016, India

**Abstract**

Nowadays, high-density RAM chips with wide I/O data (e.g., 16, 32 and 64 bits) are used in computer memory systems. When exposed to high-energy particles, these chips become highly susceptible to \( m \)-spotty byte errors, which can be effectively detected or corrected using \( (m\)-spotty) byte error-control codes. In this paper, we introduce the \( r \)-fold joint \( m \)-spotty weight enumerator of byte error-control codes over the ring of integers modulo \( \ell \) \((\ell \geq 2 \text{ an integer})\) and over arbitrary finite fields. We also study some of its properties and establish some MacWilliams type identities for the same.

© 2012 Elsevier B.V. All rights reserved.

**1. Introduction**

For the past few years, there has been an increased usage of high-density RAM chips with wide I/O data, called a byte, in computer memory systems. These chips are highly vulnerable to multiple random bit errors when exposed to strong electromagnetic waves, radio-active particles or high-energy cosmic rays. To make these memory systems more reliable, spotty [17] and \( m \)-spotty [16] byte errors are introduced, which can be effectively detected or corrected using byte error-control codes. To determine the error-detecting and error-correcting capabilities of a code, some special types of polynomials, called weight enumerators, are studied.

In general, the weight enumerator of a code is a polynomial describing certain properties of the code, and an identity which relates the weight enumerator of a code with that of its dual code is called the MacWilliams identity. Various weight enumerators have been introduced and studied for various types of codes and different weight types. The most studied among them is the Hamming weight enumerator of a code, which measures error-correcting properties of the code and also determines the probability of undetected errors when the code is used purely for error detection [18].

In [5], the MacWilliams identity for Hamming weight enumerator of a linear code is derived. MacWilliams et al. [7] established a similar relation for non-linear codes. MacWilliams et al. [6] introduced the joint weight enumerator for two linear codes over arbitrary finite fields, which generalizes the notion of Hamming weight enumerator just as the joint probability density function generalizes the single probability density function. They also showed that it also satisfies the MacWilliams identity. Siap and Ray-Chaudhuri [13] further generalized the notion of joint weight enumerator for two linear codes to the \( r \)-fold case. That is, they defined the \( r \)-fold joint weight enumerator of \( r \) linear codes over arbitrary finite fields with respect to both Hamming and Lee weights, and established MacWilliams identities for these enumerators.

Dougherty et al. [3] extended the definition of \( r \)-fold joint weight enumerator for codes over the ring \( \mathbb{Z}_\ell \) of integers modulo \( \ell \) \((\ell \geq 2)\) and established MacWilliams identities for these enumerators. They also investigated the biweight...
enumerator (joint weight enumerator of a code with itself) of self-dual codes and derived Gleason-type theorems for the corresponding biweight enumerators using invariant theory. Choie et al. [1] defined the \(r\)-fold complete joint weight enumerator for codes over the ring \(\mathbb{Z}_{2^t}\) of integers modulo \(2^t\), which generalizes the \(r\)-fold joint weight enumerator defined in [3] and is used to study self-dual codes and their shadows. They also derived MacWilliams identities for the same. On the other hand, Yoshida [19] obtained the average joint weight enumerator of two binary linear codes in terms of their weight distributions and also obtained their average intersection number.

With respect to \(m\)-spotty Hamming weights, Suzuki et al. [14] defined the \(m\)-spotty Hamming weight enumerator for binary byte error-control codes, and proved a MacWilliams identity for it. Ozen and Siap [8] and Siap [12] extended this result to arbitrary finite fields and to the ring \(\mathbb{Z}_m\) of integers modulo \(m\) \((m \geq 2 \) is an integer). (The corresponding results for the Hamming weight enumerator of a linear code follow from these results.) Siap [11] defined \(m\)-spotty Lee weight and \(m\)-spotty Lee weight enumerator of byte error-control codes over \(\mathbb{Z}_4\) and derived a MacWilliams identity for the same.

Now let \(R\) be either the finite field \(\mathbb{F}_\ell\) \((\ell \) is a prime power) or the ring \(\mathbb{Z}_\ell\) of integers modulo \(\ell\) \((\ell \geq 2 \) is an integer).

In a previous work, we introduced joint \(m\)-spotty weight enumerators of two byte error-control codes over \(R\) with respect to both \(m\)-spotty Hamming weights [9] and \(m\)-spotty Lee weights [10]. We also discussed some of their applications and derived MacWilliams type identities for these enumerators.

In this paper, we generalize the notion of a joint \(m\)-spotty Hamming weight enumerator to the \(r\)-fold joint \(m\)-spotty (Hamming) weight enumerator for \(r\) byte error-control codes over \(R\). We study some of its properties and also derive some MacWilliams type identities for it.

This paper is organized as follows: In Section 2, we provide some basic definitions and results that we need to obtain our main results. In Section 3, we introduce and study the \(r\)-fold joint \(m\)-spotty weight enumerator of byte error-control codes over \(R\). In Section 4, we derive some MacWilliams type identities for this enumerator. In Section 5, we illustrate our results with an example. In Section 6, we mention a brief conclusion and a few interesting open problems.

2. Some preliminaries

Let \(R\) be a finite commutative ring with unity and \(N\) be a positive integer. Let \(R^N\) be the \(R\)-module of all \(N\)-tuples over \(R\). For a positive divisor \(b\) of \(N\), a byte error-control code of length \(N\) and byte length \(b\) over \(R\) is defined as an \(R\)-submodule of \(R^N\).

Let us write \(N = bn\) for some positive integer \(n\). Note that each vector \(v \in R^{bn}\) can be written as \(v = (v_1, v_2, \ldots, v_n)\), where for each \(i\), \(v_i \in R^b\) and is called the \(i\)th byte of \(v\). For \(u = (u_1, u_2, \ldots, u_n)\) and \(v = (v_1, v_2, \ldots, v_n)\) in \(R^{bn}\), the inner product of \(u\) and \(v\) is defined as

\[
\langle u, v \rangle = \sum_{i=1}^{n} \langle u_i, v_i \rangle = \sum_{i=1}^{n} \left( \sum_{j=1}^{b} u_{ij} v_{ij} \right),
\]

where \(u_i = (u_{i1}, u_{i2}, \ldots, u_{ib}) \in R^b\) and \(v_i = (v_{i1}, v_{i2}, \ldots, v_{ib}) \in R^b\) are the \(i\)th bytes of \(u\) and \(v\) respectively.

Now let \(C\) be a byte error-control code of length \(bn\) and byte length \(b\) over \(R\). Then the dual code of \(C\), denoted by \(C^\perp\), is defined as

\[C^\perp = \{ v \in R^{bn} : \langle u, v \rangle = 0 \text{ for all } u \in C \}.\]

It is easy to see that \(C^\perp\) is also a byte error-control code in \(R^{bn}\) having the same byte length \(b\).

To define the \(m\)-spotty weight enumerator of a byte error-control code over \(R\), we fix an integer \(t\) \((1 \leq t \leq b)\) throughout this paper, and we define the notions of a spotty byte error and an \(m\)-spotty byte error as follows:

**Definition 2.1** ([17]). A byte error is said to be a spotty byte error or \(t/b\)-error if \(t\) or fewer bit errors occur in a \(b\)-bit byte.

**Definition 2.2** ([15]). If more than \(t\) bit errors occur in a \(b\)-bit byte, then the byte error is called an \(m\)-spotty or multiple spotty byte error.

Throughout this paper, let \([x]\) denote the ceiling of \(x\) for any real number \(x\), i.e., the number \([x]\) is equal to the smallest integer greater than or equal to \(x\).

**Definition 2.3** ([15]). Let \(v = (v_1, v_2, \ldots, v_n)\) be any vector in \(R^{bn}\) with \(v_i \in R^b\) as its \(i\)th byte. Then the \(m\)-spotty weight of \(v\), denoted by \(w_M(v)\), is defined as

\[w_M(v) = \sum_{i=1}^{n} \left\lfloor \frac{w_H(v_i)}{t} \right\rfloor,
\]

where \(w_H(v_i)\) equals the number of non-zero components of \(v_i\) and is called the Hamming weight of \(v_i\) over \(R\).

For \(t = 1\), we have \(w_M(v) = w_H(v)\). For \(t = b\), \(w_M(v)\) equals the Hamming weight of \(v\) over \(R^b\).
Definition 2.4 ([17]). Let \( u \) and \( v \) be any two vectors in \( \mathbb{R}^{2m} \) with their \( i \)-th coordinate of \( u_i \) and \( v_i \) respectively. Then the \( m \)-spotty distance between \( u \) and \( v \), denoted by \( d_M(u, v) \), is defined as
\[
d_M(u, v) = \sum_{i=1}^{n} \left\lfloor \frac{d_H(u_i, v_i)}{t} \right\rfloor = \sum_{i=1}^{n} \left\lfloor \frac{w_H(u_i - v_i)}{t} \right\rfloor = w_M(u - v),
\]
where \( d_H(u_i, v_i) = w_H(u_i - v_i) \) denotes the Hamming distance between the \( i \)-th bytes \( u_i \) and \( v_i \) for each \( i \).

It is easy to see that \( d_M \) is a metric on \( \mathbb{R}^{2m} \).

Definition 2.5 ([14]). The \( m \)-spotty (Hamming) weight enumerator \( W_c(z) \) of a byte error-control code \( C \) over \( \mathbb{R} \) is defined as
\[
W_c(z) = \sum_{u \in C} z^{w_M(u)}.
\]

3. \( r \)-fold joint \( m \)-spotty weight enumerator

In this section, we will define the \( r \)-fold joint \( m \)-spotty weight enumerator of byte error-control codes over \( \mathbb{R} \), where \( r \) is any positive integer. For this, we need the following:

Let \( \mathbb{F}_2 \) denote the vector space of all \( r \)-tuples over \( \mathbb{F}_2 = \{0, 1\} \). Let \( (\mathbb{F}_2)^* = \mathbb{F}_2 \setminus \{0\} \). For each \( a \in \mathbb{F}_2^* \), let \( [a]_i \) \((1 \leq i \leq r)\) denote the \( i \)-th coordinate of \( a \). Then for \( 1 \leq i \leq r \), let \( S_i = \{a \in \mathbb{F}_2^*: [a]_i = 1\} \).

For positive integers \( k \) and \( m \), let \( (\mathbb{R}^{m})^k = \mathbb{R}^{m} \times \mathbb{R}^{m} \times \cdots \times \mathbb{R}^{m} \) 

Now to define the \( r \)-fold joint \( m \)-spotty weight enumerator of byte error-control codes over \( \mathbb{R} \), we prove the following proposition:

Proposition 3.1. For each \( a \in (\mathbb{F}_2)^* \), there exists a function \( N_a : (\mathbb{R}^{m})^r \rightarrow \mathbb{Z} \) satisfying the following:
\[
N_a(u_i) = w_M(u_i) \quad (1 \leq i \leq r)
\]
for all \( u = (u_1, u_2, \ldots, u_r) \in (\mathbb{R}^{m})^r \) with each \( u_i = (u_i^{(1)}, u_i^{(2)}, \ldots, u_i^{(n)}) \in \mathbb{R}^{m} \) and \( u_i^{(j)} \in \mathbb{R}^{k} \) \((1 \leq j \leq n)\), where
\[
N_a(u) = \sum_{j=1}^{n} N_a(u_i^{(j)}, u_i^{(2)}, \ldots, u_i^{(j)}).
\]

To prove this proposition, we need the following two lemmas:

Lemma 3.2. Let \( g, t \) be positive integers and \( \alpha_1, \alpha_2, \ldots, \alpha_g \) be arbitrary non-negative integers. Then we have
\[
\left\lfloor \frac{\alpha_1 + \alpha_2 + \cdots + \alpha_g}{t} \right\rfloor = \left\lfloor \frac{\alpha_1}{t} \right\rfloor + \left\lfloor \frac{\alpha_2}{t} \right\rfloor + \cdots + \left\lfloor \frac{\alpha_g}{t} \right\rfloor + s,
\]
where \( s \) \((0 \leq s \leq g)\) is an integer satisfying \((s - 1)t < \alpha_1 + \cdots + \alpha_g \leq st \) and \( \bar{a}_i \) \((1 \leq i \leq g)\) denotes the least non-negative residue of \( \alpha_i \) modulo \( t \).
In other words, we have
\[\left\lceil \frac{\alpha_1 + \alpha_2 + \cdots + \alpha_k}{t} \right\rceil = \left\lfloor \frac{\alpha_1}{t} \right\rfloor + \left\lfloor \frac{\alpha_2}{t} \right\rfloor + \cdots + \left\lfloor \frac{\alpha_k}{t} \right\rfloor + \left\lceil \frac{\tilde{\alpha}_1 + \cdots + \tilde{\alpha}_k}{t} \right\rceil.\]

(Here \([x]\) denotes the ceiling of \(x\) for any real number \(x\), i.e., the smallest integer greater than or equal to \(x\); and \([x]\) denotes the floor of \(x\) for any real number \(x\), i.e., the greatest integer less than or equal to \(x\). For any integer \(p\), \(\overline{p}\) denotes the least non-negative residue of \(p\) modulo \(t\), i.e., \(\overline{p}\) is the remainder obtained upon dividing \(p\) by \(t\).)

**Proof.** The proof is trivial. □

**Lemma 3.3.** For each \(a \in (\mathbb{F}_2^*)^r\), there exists an integer-valued function \(N_a\) defined on \((R^b)^r\) and satisfying the following:

\[\sum_{a \in S_i} N_a(u) = w_H(u_i) \quad (1 \leq i \leq r)\]

for all \(u = (u_1, u_2, \ldots, u_r) \in (R^b)^r\) with each \(u_i \in R^b\).

**Proof.** For this, let \(u = (u_1, u_2, \ldots, u_r) \in (R^b)^r\) be fixed, where \(u_i = (u_{i1}, u_{i2}, \ldots, u_{ib}) \in R^b\) for \(1 \leq i \leq r\).

Now for each \(a \in (\mathbb{F}_2^*)^r\), define the number \(n_a(u)\) as follows:

\[n_a(u) = |\{j : 1 \leq j \leq b, (\widehat{u}_{ij}, u_{ij}, \ldots, \widehat{u}_{ij}) = a\}|,
\]

where for each \(i\), \(\widehat{u}_{ij}(1 \leq j \leq b)\) is given by

\[\widehat{u}_{ij} = \begin{cases} 1 & \text{if } u_{ij} \neq 0; \\ 0 & \text{if } u_{ij} = 0. \end{cases}\]

(Here \(|A|\) denotes the cardinality of the set \(A\).)

It is easy to see that

\[\sum_{a \in S_i} n_a(u) = w_H(u_i) \quad \text{for each } i,\]

(1)

where \(w_H(u_i)\) denotes the Hamming weight of \(u_i\).

Let \(\{e_1, e_2, \ldots, e_r\}\) be the standard basis of \(\mathbb{F}_2^r\) over \(\mathbb{F}_2\). To define the functions \(N_a\)'s of the desired type, we first define the functions \(\Omega_a\)'s for each \(a \in (F_2^r)^r\), as follows:

\[\Omega_a(u) = \begin{cases} \frac{\sum_{\beta \in S_i} n_{\beta}(u)}{t} & \text{if } a = e_i \ (1 \leq i \leq r); \\ 0 & \text{otherwise}. \end{cases}\]

(2)

Now for each \(a \in (F_2^r)^r\), let the functions \(N_a : (R^b)^r \rightarrow \mathbb{Z}\) be defined by

\[N_a(u) = \left\lfloor \frac{n_a(u)}{t} \right\rfloor + \Omega_a(u)\]

(3)

for all \(u \in (R^b)^r\).

For \(1 \leq i \leq r\), consider

\[\sum_{a \in S_i} N_a(u) = \sum_{a \in S_i} \left\lfloor \frac{n_a(u)}{t} \right\rfloor + \Omega_a(u) = \sum_{a \in S_i} \left\lfloor \frac{n_a(u)}{t} \right\rfloor + \Omega_{e_i}(u) = \sum_{a \in S_i} \left\lfloor \frac{n_a(u)}{t} \right\rfloor + \left\lfloor \frac{\sum_{\beta \in S_i} n_{\beta}(u)}{t} \right\rfloor,
\]

which by Lemma 3.2, gives

\[\sum_{a \in S_i} N_a(u) = \left\lfloor \frac{\sum_{a \in S_i} n_a(u)}{t} \right\rfloor.
\]
This, by (1), yields
\[ \sum_{a \in S_t} N_a(u) = \left\lceil \frac{w_M(u_i)}{t} \right\rceil = w_M(u_i). \]
This proves the lemma. \( \square \)

**Proof of Proposition 3.1.** Let \( u = (u_1, u_2, \ldots, u_t) \in (R^n)^t \) be fixed, where each \( u_i = (u_i^{(1)}, u_i^{(2)}, \ldots, u_i^{(n)}) \in R^n \) with \( u_i^{(j)} \in R^n \) for \( 1 \leq j \leq n \).

Now for each \( a \in (F_2^r)^* \), let
\[ N_a(u) = \sum_{j=1}^n N_a(u_i^{(j)}). \]
Then for \( 1 \leq i \leq r \), we have
\[ \sum_{a \in S_t} N_a(u) = \sum_{a \in S_t} \sum_{j=1}^n N_a(u_i^{(j)}). \]
By Lemma 3.3, we have \( \sum_{a \in S_t} N_a(u_i^{(j)}) = w_M(u_i^{(j)}) \) for each \( i \) and \( j \). This implies that
\[ \sum_{a \in S_t} N_a(u) = \sum_{j=1}^n w_M(u_i^{(j)}) = w_M(u_i) \quad \text{for each } i. \]
This proves the proposition. \( \square \)

We are now ready to define the \( r \)-fold joint \( m \)-spotty weight enumerator of byte error-control codes over \( R \).

**Definition 3.4.** Let \( C_1, C_2, \ldots, C_r \) be byte error-control codes of length \( bn \) and byte length \( b \) over \( R \). Then the \( r \)-fold joint \( m \)-spotty weight enumerator of the codes \( C_1, C_2, \ldots, C_r \) is defined as
\[ J_{C_1, C_2, \ldots, C_r}(x_a : a \in (F_2^r)^*) = \sum_{a \in (F_2^r)^*} \prod_{c \in (F_2^r)^*} x_a^{Q_c(C_1, C_2, \ldots, C_r)}, \]
where \( Q_a \)'s are functions from \( (R^n)^t \) into \( \mathbb{Z} \), as defined by (2)-(4).

**Remark 3.5.** The \( r \)-fold joint \( m \)-spotty weight enumerator coincides with

(i) the \( r \)-fold joint Hamming weight enumerator [13] when \( t = 1 \);
(ii) the \( m \)-spotty Hamming weight enumerator [8] when \( r = 1 \);
(iii) the joint \( m \)-spotty weight enumerator [9] when \( r = 2 \).

In the following theorem, we show that the \( r \)-fold joint \( m \)-spotty weight enumerator generalizes \( m \)-spotty weight enumerator just like the joint probability density function generalizes single probability density function.

**Theorem 3.6.** Let \( J_{C_1, C_2, \ldots, C_r}(x_a : a \in (F_2^r)^*) \) be the \( r \)-fold joint \( m \)-spotty weight enumerator of byte error-control codes \( C_1, C_2, \ldots, C_r \) over \( R \). Then we have the following:

(i) \( J_{C_1, C_2, \ldots, C_r}(1, 1, \ldots, 1) = |C_1| |C_2| \cdots |C_r| \).
(ii) Let \( i, j \) be the integers satisfying \( 1 \leq i < j \leq r \). For each \( a \in (F_2^r)^* \), define \( \tilde{a} \in (F_2^r)^* \) by
\[ [\tilde{a}]_k = \begin{cases} [a] & \text{if } k = i; \\ [a] & \text{if } k = j; \\ [a] & \text{otherwise} \end{cases} \]
for each \( k, 1 \leq k \leq r \). Then the \( r \)-fold joint \( m \)-spotty weight enumerator of the codes \( \tilde{C}_i, \tilde{C}_j, \cdots, \tilde{C}_r \) (i.e., the same sequence of codes except for \( C_i \) and \( C_j \) interchanged) is \( J_{\tilde{C}_1, \tilde{C}_2, \ldots, \tilde{C}_r}(x_a : a \in (F_2^r)^*) \).
(iii) For each \( i (1 \leq i \leq r) \), the \( m \)-spotty weight enumerator of the code \( C_i \) is given by
\[ W_{C_i}(z) = \frac{1}{|C_i|} J_{C_1, C_2, \ldots, C_r}(x_a : a \in (F_2^r)^*) \quad \text{with } x_a = \begin{cases} z & \text{if } a \in S_i; \\ 1 & \text{otherwise}, \end{cases} \]
where the product \( \prod_j \) is extended over all integers \( j \) satisfying \( 1 \leq j \leq r \) and \( j \neq i \).
Proof. (i) It is easy to see that
\[ J_{e_1, e_2, \ldots, e_r}(1, 1, \ldots, 1) = \sum_{e_1 \in e_1} \sum_{e_2 \in e_2} \cdots \sum_{e_r \in e_r} 1 = |e_1||e_2| \cdots |e_r|, \]
which proves (i).

(ii) By Definition 3.4, we have
\[ J_{e_1, e_2, \ldots, e_r}(x_a : a \in (F^r_2)^*) = \sum_{e_1 \in e_1} \sum_{e_2 \in e_2} \cdots \sum_{e_r \in e_r} \prod_{a \in (F^r_2)^*} x_a^{N_a(e_1, e_2, \ldots, e_r)}. \]

Since for each \( a \in (F^r_2)^* \), we have
\[ [\tilde{a}]_k = \begin{cases} |a| & \text{if } k = i; \\ |a| & \text{if } k = j; \\ a & \text{otherwise}, \end{cases} \]
for each \( c_k \in C_k (1 \leq k \leq r) \), we get
\[ N_a(c_1, \ldots, c_i, \ldots, c_j, \ldots, c_r) = N_a(c_1, \ldots, c_i, \ldots, c_j, \ldots, c_r), \]
from which the desired result follows immediately.

(iii) By Definition 3.4, we have
\[ J_{e_1, e_2, \ldots, e_r}(x_a : a \in (F^r_2)^*) = \sum_{e_1 \in e_1} \sum_{e_2 \in e_2} \cdots \sum_{e_r \in e_r} \prod_{a \in (F^r_2)^*} x_a^{N_a(e_1, e_2, \ldots, e_r)}. \]

As \( x_a = z \) when \( a \in S_i \) and \( x_a = 1 \) otherwise, we get
\[ J_{e_1, e_2, \ldots, e_r}(x_a : a \in (F^r_2)^*) = \sum_{e_1 \in e_1} \sum_{e_2 \in e_2} \cdots \sum_{e_r \in e_r} \prod_{a \in (F^r_2)^*} x_a^{N_a(e_1, e_2, \ldots, e_r)}. \]

By Proposition 3.1, we have \(\sum_{a \in S_i} N_a(c_1, \ldots, c_i, \ldots, c_r) = w_M(c_i) \) for each \( i \). This yields
\[ J_{e_1, e_2, \ldots, e_r}(x_a : a \in (F^r_2)^*) = \sum_{e_1 \in e_1} \sum_{e_2 \in e_2} \cdots \sum_{e_r \in e_r} \prod_{a \in (F^r_2)^*} x_a^{N_a(e_1, e_2, \ldots, e_r)} = \prod_{j=1}^r |C_j| W_{C_j}(x), \]
and proves (iii).

This completes the proof of the theorem. \( \square \)

4. MacWilliams type identities

To derive MacWilliams type identities for the \( r \)-fold joint \( m \)-spotty weight enumerator of byte error-control codes over \( R \), we need the following notations:

For each integer \( i \) \((1 \leq i \leq r)\), we define the vectors \( \sigma_i(a), \mu_i(a) \in F^{r+1}_2 \) for \( a \in F^r_2 \), as follows:

\[ [\sigma_i(a)]_j = \begin{cases} |a| & \text{if } 1 \leq j \leq i - 1; \\ 1 & \text{if } j = i; \\ |a|_{r+1} & \text{if } i + 1 \leq j \leq r + 1, \end{cases} \]

\[ [\mu_i(a)]_j = \begin{cases} |a| & \text{if } 1 \leq j \leq i - 1; \\ 0 & \text{if } j = i; \\ |a|_{r+1} & \text{if } i + 1 \leq j \leq r + 1. \end{cases} \]

Note that \( F^{r+1}_2 = \bigcup_{a \in F^r_2} \{\sigma_i(a), \mu_i(a)\} \) for each \( i \).

Definition 4.1. Let \( t \) and \( q \) be fixed integers satisfying \( 1 \leq t \leq b \) and \( 1 \leq q \leq l \). Let \( \delta = (\delta_a : a \in F^r_2) \) be a \( 2^r \)-tuple over \( \{0, 1, 2, \ldots, b\} \) satisfying \( \sum_{a \in F^r_2} \delta_a = b \). For an integer \( p \) \((0 \leq p \leq b)\), let \( A_p \) be the set of all tuples \( \alpha = (\alpha_g : g \in F^{r+1}_2) \) of non-negative integers \( \alpha_g \)'s satisfying the following:

\[ \sum_{a \in F^r_2} \alpha_{\sigma_a+1}(a) = p \quad \text{and} \quad \alpha_{\sigma_a+1}(a) + \alpha_{\mu_a+1}(a) = \delta_a \quad \text{for each } a \in F^r_2. \]
Then we define the polynomial $G_3(x_a : a \in (\mathbb{F}_2)^r)$ as
\[
G_3(x_a : a \in (\mathbb{F}_2)^r) = \prod_{i=1}^{n} G_{\delta_i}(x_a : a \in (\mathbb{F}_2)^r). \tag{7}
\]

Definition 4.2. Let $\delta = (\delta_1, \delta_2, \ldots, \delta_n)$, where each $\delta_i = (\delta_{a_i} : a \in \mathbb{F}_2^r)$ is a $2^r$-tuple over $\{0, 1, 2, \ldots, b\}$ satisfying $\sum_{a \in \mathbb{F}_2^r} \delta_{a_i} = b$. Then we define the polynomial
\[
G_3(x_a : a \in (\mathbb{F}_2)^r) = \prod_{i=1}^{n} G_{\delta_i}(x_a : a \in (\mathbb{F}_2)^r). \tag{7}
\]

Definition 4.3. Let $c_1, c_2, \ldots, c_r$ be vectors in $\mathbb{F}_2^b$. The joint composition vector of the $r$-tuple $(c_1, c_2, \ldots, c_r)$, denoted by $j(c_1, c_2, \ldots, c_r)$, is defined as the tuple $\delta = (\delta_{a_i} : a \in \mathbb{F}_2^r)$, where for each $a \in \mathbb{F}_2^r$, $\delta_a$ is given by
\[
\delta_a = |\{k : 1 \leq k \leq b, (\hat{c}_{1k}, \hat{c}_{2k}, \ldots, \hat{c}_{rk}) = a\}| \quad \text{with} \quad \hat{c}_{ik} = \begin{cases} 1 & \text{if} \ c_{ik} \neq 0; \\ 0 & \text{otherwise.} \end{cases}
\]

It is easy to see that $\sum_{a \in \mathbb{F}_2^r} \delta_a = b$.

For $1 \leq i \leq r$, let $c(i) = (c_{1i}, c_{2i}, \ldots, c_{mi})$ be a vector in $\mathbb{F}_2^b$ with each $c_{ik} \in \mathbb{F}_2$. The joint composition vector of the $r$-tuple $(c^{(1)}, c^{(2)}, \ldots, c^{(r)})$ is defined as
\[
j(c^{(1)}, c^{(2)}, \ldots, c^{(r)}) = \delta = (\delta_1, \delta_2, \ldots, \delta_n),
\]
where $\delta_k = j(c_{1k}, c_{2k}, \ldots, c_{rk})$ for each $k$, which is as defined above.

In the following theorem, we derive some MacWilliams type identities for the $r$-fold joint $m$-spotty weight enumerator of byte error-control codes over $R$.

Theorem 4.4. Let $C_1, C_2, \ldots, C_r$ be byte error-control codes of length $bn$ and byte length $b$ over $R$. Let $P(\delta)$ be the number of $r$-tuples $(c_1, c_2, \ldots, c_r)$ of codewords $c_i \in C_i$ ($1 \leq i \leq r$) having joint composition vector as $\delta$. Then for $1 \leq q \leq r$, we have
\[
\mathcal{J}_{c_1, c_2, \ldots, c_r} = \frac{1}{|C_q|} \sum_{i} P(\delta) G_3(x_a : a \in (\mathbb{F}_2)^r),
\]
where the summation runs over all $n$-tuples $\delta = (\delta_1, \delta_2, \ldots, \delta_n)$ such that each $\delta_i = (\delta_{a_i} : a \in \mathbb{F}_2^r)$ is a $2^r$-tuple over $\{0, 1, 2, \ldots, b\}$ satisfying $\sum_{a \in \mathbb{F}_2^r} \delta_{a_i} = b$, and the polynomial $G_3(x_a : a \in (\mathbb{F}_2)^r)$ is as defined by (7).

Remark 4.5. In general, it is very hard to compute the $r$-fold joint $m$-spotty weight enumerator of the codes $C_1, C_2, \ldots, C_r$ when one of the codes, say $C_q$, is of large size. However, applying Theorem 4.4, it is comparatively easier to determine the same from the list of numbers $P(\delta)$’s for the codes $C_1, \ldots, C_{q-1}, C_q^\perp, C_{q+1}, \ldots, C_r$.

To prove Theorem 4.4, we need to prove the following lemma:

**Lemma 4.7.** Let $c_1, c_2, \ldots, c_r$ be fixed vectors in $R^b$. For $1 \leq q \leq r$, let the joint composition vector of the $r$-tuple $(c_1, c_2, \ldots, c_r)$ be $\delta = (\delta_a : a \in \mathbb{F}_2^q)$. Then we have

$$
\sum_{v \in \mathbb{R}^b} \chi((c_q, v)) \prod_{a \in (\mathbb{F}_2^q)^*} X_a^{N_a(c_1, \ldots, c_{q-1}, v, c_{q+1}, \ldots, c_r)} = G_\delta(x_a : a \in (\mathbb{F}_2^q)^*).
$$

**Proof.** First of all, we see that the sum

$$
\sum_{v \in \mathbb{R}^b} \chi((c_q, v)) \prod_{a \in (\mathbb{F}_2^q)^*} X_a^{N_a(c_1, \ldots, c_{q-1}, v, c_{q+1}, \ldots, c_r)}
$$
equals \sum_{p=0}^{b} \sum_{v \in \mathbb{R}^b} \chi((c_q, v)) \prod_{a \in (\mathbb{F}_2^q)^*} X_a^{N_a(c_1, \ldots, c_{q-1}, v, c_{q+1}, \ldots, c_r)}
.$$ We assert that this sum depends upon the joint composition vector $\delta = (\delta_a : a \in \mathbb{F}_2^q)$ of the tuple $(c_1, c_2, \ldots, c_r)$.

Let $v = (v_1, v_2, \ldots, v_b)$ be a vector in $R^b$ having Hamming weight $w_P(v) = p$. Let us write each vector $c_i \in R^b$ as $c_i = (c_{i1}, c_{i2}, \ldots, c_{i, p})$. For each $g \in \mathbb{F}_2^{b+1}$, suppose that $\alpha_g$ is the number of positions $j$ $(1 \leq j \leq b)$ satisfying

$$(\hat{c}_{ij}, \ldots, \hat{c}_{ij}, \hat{v}_j, \hat{c}_{q+1,j}, \ldots, \hat{c}_{ij}) = g.$$ where $\hat{d}(d \in R)$ is given by

$$\hat{d} = \begin{cases} 1 & \text{if } d \neq 0; \\ 0 & \text{if } d = 0. \end{cases}$$

As the joint composition vector of the tuple $(c_1, c_2, \ldots, c_r)$ is $\delta = (\delta_a : a \in \mathbb{F}_2^q)$, it is easy to see that the numbers $\alpha_g$'s are the non-negative integers satisfying the following:

$$\sum_{a \in \mathbb{F}_2^q} \alpha_{\alpha_g(a)} = p \quad \text{and} \quad \alpha_{\alpha_g(a)} + \alpha_{\mu\cdot \alpha_g(a)} = \delta_a \quad \text{for each } a \in \mathbb{F}_2^q.$$ Thus the tuple $\alpha = (\alpha_g : g \in \mathbb{F}_2^{b+1})$ lies in $A_p$.

Now for each $a \in (\mathbb{F}_2^q)^*$, by (3), we have

$$N_a(c_1, \ldots, c_{q-1}, v, c_{q+1}, \ldots, c_r) = \left[ \frac{\alpha_{\alpha_g(a)} + \alpha_{\mu\cdot \alpha_g(a)}}{t} \right] + \theta_a^{(\alpha)},$$

where $\theta_a^{(\alpha)}$ is as defined by (6).

Let $\alpha = (\alpha_g : g \in \mathbb{F}_2^{b+1}) \in A_p$ be fixed. Consider the sum $\sum_{v \in \mathbb{R}^b} \chi((c_q, v))$, where the summation $\sum_{1}$ runs over all $v \in R^b$ with the positions of $v$ fixed such that for each $g \in \mathbb{F}_2^{b+1}$,

$$|\{ j : 1 \leq j \leq b, (\hat{c}_{ij}, \ldots, \hat{c}_{ij}, \hat{v}_j, \hat{c}_{q+1,j}, \ldots, \hat{c}_{ij}) = g \}| = \alpha_g.$$ Since the Hamming weight of $v$ is $p = \sum_{a \in \mathbb{F}_2^q} 1$, the non-zero entries of $v$ appear at $p$ positions, say $i_1, i_2, \ldots, i_p$.

If we let $R^* = R \setminus \{0\}$, then we have

$$\sum_{1} \chi((c_q, v)) = \sum_{v_1, v_2, \ldots, v_p \in R^*} \chi \left( \sum_{k=1}^p c_{qk}v_{ik} \right).$$

It is easy to see that for all such $v$'s, the vectors $c_q$ and $v$ have simultaneous non-zero bits at $h_P(\alpha) = \sum_{a \in \mathbb{F}_2^q} \alpha_{\alpha_g(a)}$ positions. Again without any loss of generality, we suppose that the vectors $c_q$ and $v$ have simultaneous non-zero bits at positions $i_1, i_2, \ldots, h_P(\alpha)$. This gives

$$\sum_{1} \chi((c_q, v)) = \prod_{k=1}^{h_P(\alpha)} \sum_{v_k \in R^*} \chi(c_{qk}v_{ik}) \prod_{k=(h_P(\alpha)+1)}^p \sum_{v_k \in R^*} \chi(0v_{ik}).$$

As $\chi$ is a non-trivial additive character of $R$, we have $\sum_{d \in R^*} \chi(d) = -1$. Also using $\chi(0) = 1$ and $|R^*| = \ell - 1$, we get

$$\sum_{1} \chi((c_q, v)) = (-1)^h_P(\alpha)(\ell - 1)^{p-h_P(\alpha)}.$$
Proof of Theorem 4.4.

Let $f(v) = \sum_{x \in \mathbb{F}_2^q} x_a^{N_a(c_1, \ldots, c_{q-1}, v, c_{q+1}, \ldots, c_r)}$ for $v \in \mathbb{F}_2^n$. Then for $u = (u_1, u_2, \ldots, u_n) \in \mathbb{F}_2^n$, applying Lemma 2.6, we have

$$\tilde{f}(u) = \sum_{v \in \mathbb{F}_2^n} \chi(\langle v, u \rangle) \sum_{x \in \mathbb{F}_2^n} x_a^{N_a(c_1, \ldots, c_{q-1}, v, c_{q+1}, \ldots, c_r)}$$

$$= \sum_{v \in (c_1, v_2, \ldots, v_n) \in \mathbb{F}_2^n} \chi(\sum_{i=1}^n \langle v_i, u_i \rangle) \prod_{a \in \mathbb{F}_2^q} x_a^{N_a(c_1, \ldots, c_{q-1}, v, c_{q+1}, \ldots, c_r)}$$

$$= \sum_{v \in \mathbb{F}_2^n} \chi(\langle v, u \rangle) \prod_{a \in \mathbb{F}_2^q} x_a^{N_a(c_1, \ldots, c_{q-1}, v, c_{q+1}, \ldots, c_r)}$$

where each $c_k = (c_{k1}, c_{k2}, \ldots, c_{kn}) \in C_k$ for $1 \leq k \leq r$ and $k \neq q$. 

Proof of Theorem 4.4. To prove this theorem, we will apply Lemma 2.6. By the definition of $r$-fold joint $m$-spotty weight enumerator for the codes $C_1, \ldots, C_{q-1}, C_q, C_{q+1}, \ldots, C_r$, we have

$$\tilde{g}_{C_1, \ldots, C_{q-1}, C_q, C_{q+1}, \ldots, C_r}(x : a \in (\mathbb{F}_2^q)^*) = \sum_{v \in \mathbb{F}_2^n} \sum_{x \in \mathbb{F}_2^n} x_a^{N_a(c_1, \ldots, c_{q-1}, v, c_{q+1}, \ldots, c_r)}$$

where the summation $\sum_{q}$ runs over all the codewords $c_q \in C_q$ for $1 \leq k \leq r$ and $k \neq q$. 

Again for each $a \in \mathbb{F}_2^n$, note that the vector $\sigma_{q+1}(a)$ matches with $\mu_{q+1}(a)$ at all positions except at the $(q+1)$th position, so the $\alpha_{\sigma_{q+1}(a)}$ non-zero positions of $v$ can be chosen from $\alpha_{\sigma_{q+1}(a)} + \alpha_{\mu_{q+1}(a)} = \delta_a$ positions in $\binom{\delta_a}{\delta_{q+1}(a)}$ ways (as shown in Fig. 1).

Let $\tilde{g}_{C_1, \ldots, C_{q-1}, C_q, C_{q+1}, \ldots, C_r}(x : a \in (\mathbb{F}_2^q)^*) = \sum_{v \in \mathbb{F}_2^n} \sum_{x \in \mathbb{F}_2^n} x_a^{N_a(c_1, \ldots, c_{q-1}, v, c_{q+1}, \ldots, c_r)}$ for $v \in \mathbb{F}_2^n$. Then for $u = (u_1, u_2, \ldots, u_n) \in \mathbb{F}_2^n$, applying Lemma 2.6, we have

$$\tilde{g}_{C_1, \ldots, C_{q-1}, C_q, C_{q+1}, \ldots, C_r}(x : a \in (\mathbb{F}_2^q)^*) = \sum_{v \in \mathbb{F}_2^n} \sum_{x \in \mathbb{F}_2^n} x_a^{N_a(c_1, \ldots, c_{q-1}, v, c_{q+1}, \ldots, c_r)}$$

$$= \sum_{v \in \mathbb{F}_2^n} \sum_{x \in \mathbb{F}_2^n} x_a^{N_a(c_1, \ldots, c_{q-1}, v, c_{q+1}, \ldots, c_r)}$$

$$= \sum_{v \in \mathbb{F}_2^n} \sum_{x \in \mathbb{F}_2^n} x_a^{N_a(c_1, \ldots, c_{q-1}, v, c_{q+1}, \ldots, c_r)}$$

where each $c_k = (c_{k1}, c_{k2}, \ldots, c_{kn}) \in C_k$ for $1 \leq k \leq r$ and $k \neq q$. 

This proves the lemma. 

\hspace{1cm} \square
Now suppose that for each $i$, the joint composition vector of the $r$-tuple $(c_{1i}, c_{2i}, \ldots, c_{ni})$ is $\delta_i$. Then by Lemma 4.7, we get

$$\sum_{v \in \mathbb{F}_2^n} \chi((u_i, v_i)) \prod_{v \in (\mathbb{F}_2^r)^*} x_a^{N(c_{1i}, \ldots, c_{q-1}; v_i, c_{q+1} ; \ldots ; c_{ni})} = G_{\delta_i}(x_a : a \in (\mathbb{F}_2^r)^*)$$

for each $i$.

This gives

$$\sum_{u \in C_q} \bar{f}(u) = \sum \prod_{i=1}^n G_{\delta_i}(x_a : a \in (\mathbb{F}_2^r)^*),$$

where the summation $\sum$ runs over all codewords $c_k = (c_{k1}, c_{k2}, \ldots, c_{kn}) \in C_k$ for $1 \leq k \leq r$, satisfying $j(c_{1i}, c_{2i}, \ldots, c_{ni}) = \delta_i$ for each $i$.

If $P(\delta)$ is the number of $r$-tuples $(c_1, c_2, \ldots, c_r)$ of codewords $c_k \in C_k$ ($1 \leq k \leq r$) such that $j(c_1, c_2, \ldots, c_r) = \delta$, then using (7), we get

$$\sum_{u \in C_q} \bar{f}(u) = \sum P(\delta) G_{\delta}(x_a : a \in (\mathbb{F}_2^r)^*),$$

(9)

where the summation runs over all $n$-tuples $\delta = (\delta_1, \delta_2, \ldots, \delta_n)$ such that each $\delta_i = (\delta_a^{(i)} : a \in \mathbb{F}_2^n)$ is a $2^t$-tuple over $\{0, 1, 2, \ldots, b\}$ satisfying $\sum_{u \in \mathbb{F}_2^n} \delta_a^{(i)} = b$. Again applying Lemma 2.6 and using (9), we get

$$\mathcal{G}_{1...e_{q-1}, e_q, e_{q+1}...e_r}(x_a : a \in (\mathbb{F}_2^r)^*) = \frac{1}{|C_q|} \sum_{v \in C_q} f(v) = \frac{1}{|C_q|} \sum_{u \in C_q} \bar{f}(u)$$

$$= \frac{1}{|C_q|} \sum P(\delta) G_\delta(x_a : a \in (\mathbb{F}_2^r)^*),$$

which proves the theorem. □

5. An example

To illustrate our results, we obtain the $r$-fold joint $m$-spotty weight enumerator with $t = 2$ for ternary byte error-control codes $C_1$, $C_2$, $C_3$ of length 9, byte length 3 and having generator matrices $G_1$, $G_2$, $G_3$ respectively, where

$$G_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix},$$

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 2 & 1 & 1 \end{bmatrix},$$

and

$$G_3 = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 2 & 0 & 20 & 0 \end{bmatrix}. $$

Since $|C_1| = (6561)$ is large, we apply Theorem 4.4 to obtain the 3-fold joint $m$-spotty weight enumerator of the codes $C_1$, $C_2$ and $C_3$.

It is easy to see that the generator matrix for the dual code $C_1^\perp$ of $C_1$ is given by $G_1^\perp = \begin{bmatrix} 1 & 0 & 0 & 2 & 1 & 0 & 0 & 1 & 2 \end{bmatrix}$. Now to compute the 3-fold joint $m$-spotty weight enumerator, we need to compute the joint composition vectors of the codewords in $C_1^\perp$, $C_2$ and $C_3$.

It is clear that the codewords $u_1 = (1, 0, 0, 2, 1, 0, 0, 1, 2) \in C_1^\perp$, $u_2 = (0, 1, 0, 0, 1, 1, 2, 1, 1) \in C_2$ and $u_3 = (0, 0, 1, 1, 0, 2, 0, 2, 0) \in C_3$, and have joint composition vector $j(u_1, u_2, u_3)$ as $\delta = (\delta_1, \delta_2, \delta_3)$, where $\delta_1 = (0, 1, 1, 0, 0, 0, 0, 0, 0)$, $\delta_2 = (0, 0, 0, 0, 0, 0, 0, 1, 0)$ and $\delta_3 = (0, 0, 0, 0, 0, 0, 1, 1, 1)$. Also note that this $\delta$ contributes

$$G_{\delta_1}(x_a : a \in (\mathbb{F}_2^3)^*) = G_{\delta_2}(x_a : a \in (\mathbb{F}_2^3)^*) = G_{\delta_3}(x_a : a \in (\mathbb{F}_2^3)^*)$$

to the 3-fold joint $m$-spotty weight enumerator of $C_1$, $C_2$, $C_3$, where by (5), we have

$$G_{\delta_1}(x_a : a \in (\mathbb{F}_2^3)^*) = x_1^2 x_2 + 3x_1 x_2 x_4 - 4x_1 x_2^2 x_3,$$

$$G_{\delta_2}(x_a : a \in (\mathbb{F}_2^3)^*) = x_1 x_2 - 3x_1 x_2 x_4 + 2x_1 x_2 x_3$$

and

$$G_{\delta_3}(x_a : a \in (\mathbb{F}_2^3)^*) = x_1^2 x_2^2 - 2x_1 x_2 x_6 - x_1 x_2^2 x_4 + 2x_1 x_2 x_4 x_6.$$
Table 1

<table>
<thead>
<tr>
<th>$\delta_1$</th>
<th>$\delta_2$</th>
<th>$\delta_3$</th>
<th>$P(\delta)$</th>
<th>$G_4(x_{a} : a \in \mathbb{F}_2^3)^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>1</td>
<td>$a^3$</td>
</tr>
<tr>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>2</td>
<td>$b^2c$</td>
</tr>
<tr>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>2</td>
<td>$d_{ek}$</td>
</tr>
<tr>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>4</td>
<td>$g_{oi}$</td>
</tr>
<tr>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>2</td>
<td>$j^2f^2$</td>
</tr>
<tr>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>4</td>
<td>$l_{mn}$</td>
</tr>
<tr>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>4</td>
<td>$h_{pr}$</td>
</tr>
<tr>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>${0, 0, 0, 0, 0, 0, 0, 0}$</td>
<td>8</td>
<td>$q_{ts}$</td>
</tr>
</tbody>
</table>

Working in a similar way, we obtain the contributing polynomials for the remaining choices for joint composition vectors, which are given by Table 1.

Now by Theorem 4.4, the 3-fold joint $m$-spotty weight enumerator of $C_1, C_2, C_3$ is given by

$$f_{C_1,C_2,C_3}(x_{a} : a \in \mathbb{F}_2^3)^* = \frac{1}{|C_1|^3} \sum P(\delta)G_4(x_{a} : a \in \mathbb{F}_2^3)^*$$

$$= 1 + 6808 x_1^3 x_2^4 x_3^5 + 208 x_1^3 x_2^2 x_3 x_4 x_5 + 3552 x_1^3 x_2^3 x_4^4 x_5 + 320 x_1^2 x_2^4 x_5^6 + 2256 x_1^2 x_2 x_4^3 x_6 + 5008 x_1^2 x_2^3 x_4 x_5^6 + 1120 x_1^2 x_2^4 x_4 x_5^6 + 640 x_1^2 x_2^3 x_4 x_5^6 + 1192 x_1^2 x_2 x_4^3 x_5^6 + 2000 x_1^2 x_2^3 x_4 x_5^6 + 2600 x_4^2 x_4^3 x_5^6 + 2504 x_1^2 x_2 x_4^3 x_5^6 + 1096 x_1^2 x_2^3 x_4 x_5^6 + 2 x_3^3 x_4^3 x_5^6 + 152 x_2^3 x_4^2 x_5^6 + 28 x_2^3 x_4 x_5^6 + 4 x_3^3 x_5^6 + 2 x_4^2 x_4 x_5^6 + 4 x_1^3 x_2 x_4^3 x_5^6 + 3404 x_1^3 x_4^3 x_5^6 + 2696 x_1^3 x_4^2 x_5^6 + 608 x_1^3 x_4 x_5^6 + 1128 x_1^3 x_4^2 x_5^6 + 40 x_2^3 x_4^2 x_5^6 + 104 x_1^2 x_4 x_5^6 + 320 x_2^3 x_4 x_5^6 + 64 x_1^2 x_4^2 x_5^6 + 176 x_2^2 x_4^2 x_5^6 + 2740 x_2 x_4 x_5^6 + 5392 x_1^3 x_2^4 x_4^4 + 672 x_2^2 x_4 x_5^6 + 320 x_2^2 x_4^2 x_5^6 + 16 x_4^2 + 2984 x_2^3 x_4^3 x_5^6 + 864 x_1^2 x_4^3 x_5^6 + 1216 x_1^2 x_2 x_4^4 + 1776 x_1^2 x_4^3 x_5^6 + 2 x_2^4 + 448 x_2 x_4^3 x_5^6 + 326 x_2^4 x_5^6 + 2266 x_4^3.$$

6. Conclusions and future work

Let $R$ be either the finite field $\mathbb{F}_t$ or the ring $\mathbb{Z}_t$ of integers modulo $t$. In this paper, the $r$-fold joint $m$-spotty weight enumerator is introduced for byte error-control codes over the ring $R$, and some MacWilliams type identities are also derived.

As an application of the $r$-fold joint $m$-spotty weight enumerator, it would be interesting to propose and study a generalization of the secret sharing scheme studied by Cruz et al. [2].

Another application of the $r$-fold joint $m$-spotty weight enumerator will lie in studying self-dual codes, which form an important class of codes due to the following reasons: (i) they have a rich mathematical structure, (ii) many good codes belong to this class, and (iii) they have many interesting applications. It would be interesting to derive Gleason’s Theorem and its generalizations for self-dual ($m$-spotty) byte error-control codes over $R$ using our $r$-fold joint $m$-spotty weight enumerators and invariant theory. These results will enable one to classify self-dual byte error-control codes over $R$.

References


