Моделирование и анализ информационных систем. Т. 22, № 4 (2015), с. 464–482
Modeling and Analysis of Information Systems. Vol. 22, No 4 (2015), pp. 464–482

DOI: 10.18255/1818-1015-2015-4-464-482

УДК 517.9

Алгоритмы для мажоритарного декодирования групповых кодов

Деундяк В. М., Косолапов Ю. В.

получена 29 апреля 2015

Решается задача конструктивного описания и обоснования алгоритмов, необходимых при практической реализации мажоритарного декодера для групповых кодов, заданных как левые идеалы групповых алгебр. Кроме алгоритмов, необходимых для реализации классического декодера Дж. Мэсси, построено обобщение классического декодера для кодов с неравной защитой символов, который в ряде случаев может быть лучше классического. Для применения как классического декодера Дж. Мэсси, так и его обобщения к групповым кодам разработан алгоритм построения декодирующих деревьев, которые лежат в основе этих алгоритмов мажоритарного декодирования. В силу того, что групповые коды определяются как левые идеалы групповых алгебр, алгоритм построения декодирующих деревьев позволяет по одному дереву построить все декодирующие деревья. На основе обобщенного алгоритма декодирования разработан алгоритм декодирования групповых кодов, индуцированных кодами на подгруппе. Применение разработанных декодеров проиллюстрировано на примере кодов Рида–Маллера–Бермана и индуцированных ими групповых кодах на неабелевой группе аффинных преобразований. В частности, для кода Рида–Маллера–Бермана приводится описание и обоснование алгоритма построения одного декодирующего дерева, по которому с использованием алгоритма построения всех декодирующих деревьев строится мажоритарный декодер кода Рида–Маллера–Бермана и индуцированных им кодов.

Ключевые слова: мажоритарный декодер, групповые алгебры, групповые коды, коды Рида–Маллера–Бермана

Для цитирования: Деундяк В. М., Косолапов Ю. В., "Алгоритмы для мажоритарного декодирования групповых кодов", *Моделирование и анализ информационных систем*, **22**:4 (2015), 464–482.

Об авторах: Деундяк Владимир Михайлович, orcid.org/0000-0001-8258-2419, канд. физ.-мат. наук, доцент,
ФГНУ НИИ "Спецвузавтоматика",
пер. Газетный, 51, г. Ростов-на-Дону, 344002, Россия, e-mail: vlade@math.rsu.ru,

Косолапов Юрий Владимирович, orcid.org/0000-0002-1491-524X, канд. техн. наук,
Южный Федеральный Университет,
ул. Большая Садовая, 105/42, г. Ростов-на-Дону, 344006, Россия, e-mail: itaim@mail.ru

Введение

Среди комбинаторных методов декодирования линейных кодов, таких как декодирование по синдромам, декодирование по информационным совокупностям [1], мажоритарное декодирование Дж. Мэсси [2], особо выделяется последний метод. Особенность этого метода состоит в том, что скорость декодирования остается высокой при росте длины кода, в то время как первые два метода применимы для небольших длин кодов [3]. Однако для применения мажоритарного декодера необходимо, чтобы линейный код обладал некоторой специальной структурой. К таким кодам относятся введенные Дж. Мэсси MLD-коды и, в частности, классические коды Рида–Маллера, которые применяются как в теории связи, так и в криптографии [4]. Коды Рида–Маллера, как показано С.Д. Берманом в [5], могут быть описаны на основе использования аппарата групповой алгебры. В дальнейшем алгебраический подход Бермана к описанию кодов был развит в ряде работ. В частности, в [6] описываются p-ичные коды Рида–Маллера (p – простое число) и строится соответствующий мажоритарный декодер; в [7] исследуются коды типа Рида–Маллера на конечной абелевой группе, однако при этом декодер не описывается. В [8] показано, что используя алгебраический подход С.Д. Бермана, можно по известным кодам (для которых применим декодер Месси) построить класс других групповых кодов, пригодных для мажоритарного декодирования. Представляется, что новые коды, кроме применения в теории связи, могут быть использованы для усиления теоретико-кодовых методов защиты данных, таких, как кодовое зашумление [9], [10], широковещательное шифрование [11], кодовые криптосистемы [12], [13], [14]. Поэтому актуальной является задача конструктивного описания и обоснования алгоритмов, необходимых для практической реализации процедуры декодирования групповых MLD-кодов.

Настоящая статья состоит из трех разделов. В первом разделе рассматривается мажоритарный декодер Дж. Мэсси; в удобном виде приводится ряд известных фактов и с использованием декодирующих деревьев приводится конструктивное описание этого алгоритма, а также ряда вспомогательных алгоритмов. Далее строится обобщение декодера Дж. Мэсси на случай неравномерной защиты кодовых координат. Неравномерная защита является актуальной в случаях, когда номера координат в кодовых словах наделены разной по важности смысловой нагрузкой, и поэтому представляют интерес коды, позволяющие правильно декодировать важные координаты, даже если остальные координаты при этом декодируются с ошибкой (см., например, [15]).

Во втором разделе статьи рассматривается применение обобщенного мажоритарного алгоритма Дж. Мэсси к групповым кодам. Основными результатами этого раздела являются два алгоритма: алгоритм построения всех декодирующих деревьев для группового кода по одному дереву и алгоритм декодирования индуцированных кодов. Эти результаты позволяют декодировать широкий класс групповых кодов. Полученные результаты иллюстрируются в третьем разделе на группе аффинных преобразований $\mathrm{Aff}(\mathbb{F}_{2^n})$ и ее абелевой 2-подгруппе $\mathcal{H}$ порядка 2^n.

1. Мажоритарный декодер Дж. Мэсси для MLD–кодов и его обобщение на случай неравномерной защиты кодовых символов

1.1. Мажоритарный декодер Дж. Мэсси

В работе Дж. Мэсси [2] разработан мажоритарный декодер для линейных кодов на основе M–ортогональных множеств. Следуя [8], приведем необходимые сведения о конструкции соответствующего декодера. Для натурального n символом $\underline{n}$ будем обозначать множество $\{1;...;n\}$. Пусть V – векторное пространство над конечным полем $\mathbb{F}$. Зафиксируем в V базис B и символом (V, d_B) обозначим метрическое пространство V с метрикой Хэмминга d_B, построенной относительно базиса B. Для вектора $\mathbf{x}(\in V)$ множество базисных векторов, коэффициенты при которых в разложении $\mathbf{x} = \sum_{\mathbf{b}\in B} x_{\mathbf{b}}\mathbf{b}$ ненулевые, называется носителем вектора x относительно базиса B и обозначается $\mathrm{supp}_B(\mathbf{x})$. Вес $w_B(\mathbf{x})$ вектора $\mathbf{x}$ определяется как $|\mathrm{supp}_B(\mathbf{x})|$. (Здесь и далее символом $|A|$ обозначается мощность множества A.) Всякое линейное подпространство C метрического пространства (V, d_B) называется линейным кодом. Размерность и длину кода будем обозначать соответственно $k(C)$ и $n(C)$, а минимальное кодовое расстояние кода C обозначим $\mathrm{dist}_B(C)$. Двойственный код к коду C обозначим $C^{\perp}$. Множество векторов $\mathcal{M}_{\mathbf{v}} = \{\mathbf{v}^{(1)};...;\mathbf{v}^{(r)}\}(\subset V)$ называется M-ортогональным вектору $\mathbf{v}(\in V)$, если для любого $i \in \underline{r}$ найдется $\mathbf{w}^{(i)}(\in V\backslash\{(0,...,0)\})$ такой, что:

1)$\mathbf{v}^{(i)} = \mathbf{v} + \mathbf{w}^{(i)}$;

2)$\mathrm{supp}_B(\mathbf{v}) \cap \mathrm{supp}_B(\mathbf{w}^{(i)}) = \emptyset$;

3)$\mathrm{supp}_B(\mathbf{w}^{(j')}) \cap \mathrm{supp}_B(\mathbf{w}^{(j)}) = \emptyset$ для всех $j, j' \in \underline{r}$ таких, что $j \neq j'$.

Свойство M-ортогональности множества $\mathcal{M}_{\mathbf{v}}$ вектору $\mathbf{v}$, в частности, означает, что $|\mathrm{supp}_B(\mathbf{v}^{(i)})| > |\mathrm{supp}_B(\mathbf{v})|$ для всех $i = 1, ..., r$ и

$$\forall\ i \neq j: \quad \mathrm{supp}_B(\mathbf{v}^{(i)}) \cap \mathrm{supp}_B(\mathbf{v}^{(j)}) = \mathrm{supp}_B(\mathbf{v}). \tag{1}$$

Исходные параметры: $\mathbf{b}(\in B)$, $\mathcal{M}_{\mathbf{b}} = \{\mathbf{v}^{1,\mathbf{b}};...;\mathbf{v}^{r_{\mathbf{b}},\mathbf{b}}\}(\subseteq C^{\perp})$, $\mathbf{v} = \mathbf{c} + \mathbf{e}$ — принятый из канала вектор, $w_B(\mathbf{e}) \leq \lfloor r_{\mathbf{b}}/2 \rfloor$, $\mathbf{c} \in C$

Результат: значение координаты $c_{\mathbf{b}}$ кодового вектора

Вычислить последовательность $l[\mathcal{M}_{\mathbf{b}}, \mathbf{v}]$ вида

$$l[\mathcal{M}_{\mathbf{b}}, \mathbf{v}] := \left(l(\mathbf{v}^{1,\mathbf{b}}), ..., l(\mathbf{v}^{r_{\mathbf{b}},\mathbf{b}})\right); \tag{2}$$

// $l(\mathbf{v}^{i,\mathbf{b}}) = \left\langle \mathbf{v}, \mathbf{v}^{(i,\mathbf{b})} \right\rangle_B = \left\langle \mathbf{e}, \mathbf{v}^{(i,\mathbf{b})} \right\rangle_B$ – скалярное произведение

// векторов $\mathbf{v}$ и $\mathbf{v}^{(i,\mathbf{b})}$ в базисе B, $i = 1, ..., r_{\mathbf{b}}$;

$c_{\mathbf{b}} := v_{\mathbf{b}} - \mathrm{MajorVote}(r_{\mathbf{b}}, l[\mathcal{M}_{\mathbf{b}}, \mathbf{v}])$;

возвратить $c_{\mathbf{b}}$

Алгоритм 1: Decoder1

В [2] описана схема декодирования значения $\mathbf{b}$-координаты кодового слова с использованием имеющегося M-ортогонального множества $\mathcal{M}_{\mathbf{b}}$ $(\subseteq C^{\perp})$, где $\mathbf{b} \in B$. В соответствии с этой схемой в удобном для дальнейшего виде построим соответ-

ствующий алгоритм декодирования Decoder1, обращающийся к вспомогательному алгоритму MajorVote.

Лемма 1. *Пусть C – линейный код в V, B – базис в V, $\mathbf{b} \in B$, $\mathbf{v}$ – вектор на выходе канала связи,*

$$\mathbf{v} = \mathbf{c} + \mathbf{e}, \quad w_B(\mathbf{e}) = t \le \lfloor r_{\mathbf{b}}/2 \rfloor, \tag{3}$$

где $r_{\mathbf{b}} \in \mathbb{N}$, $\mathbf{c} \in C$. Если в коде $C^{\perp}$ найдется $r_{\mathbf{b}}$-элементное множество векторов $\mathcal{M}_{\mathbf{b}} = \{\mathbf{v}^{1,\mathbf{b}};\ldots;\mathbf{v}^{r_{\mathbf{b}},\mathbf{b}}\}$, которое M-ортогонально базисному вектору $\mathbf{b}$, то значение координаты $_{\mathbf{b}}$ в кодовом векторе может быть восстановлено с помощью алгоритма Decoder1. *Кроме того,*

$$\operatorname{dist}_B(C) \ge r_{\mathbf{b}} + 1. \tag{4}$$

Доказательство леммы 1 вытекает из [2], с. 24. Если для какого-то $\mathbf{b}$ в (4) выполняется строгое неравенство, то это означает, что декодер Decoder1 исправляет меньше ошибок, нежели теоретически возможно для кода C. Ниже строится модификация алгоритма Decoder1, которая может исправлять большее количество ошибок. Для описания обобщенного алгоритма Decoder2 нам понадобится понятие помеченного дерева, использованное в [8]. Символом $\mathrm{WB}_{\mathbf{b},r_{\mathbf{b}},L_{\mathbf{b}}} = \mathrm{WB}_{\mathbf{b},r_{\mathbf{b}},L_{\mathbf{b}}}[C]$ будем обозначать помеченное дерево с корнем $\mathbf{b}$, обладающее следующими свойствами:

- множество вершин этого дерева состоит из $L_{\mathbf{b}} + 1$ уровня; корень $\mathbf{b}(\in B)$ находится на уровне 0, а листья – на уровне $L_{\mathbf{b}}$;
- каждая вершина, не являющаяся листом, имеет не менее $r_{\mathbf{b}}(\in \mathbb{N})$ непосредственно следующих за ней вершин;
- листья дерева помечены элементами из $C^{\perp}$;
- метки вершин, непосредственно следующих из произвольной вершины $\mathbf{v}$, находящейся на уровне $i(0 \le i < L_{\mathbf{b}})$, образуют в совокупности множество, M-ортогональное $\mathbf{v}$.

Замечание 1. *В силу определения дерева $\mathrm{WB}_{\mathbf{b},r_{\mathbf{b}},L_{\mathbf{b}}}$ и определения M-ортогональных систем, из существования для базисного вектора $\mathbf{b}$ дерева $\mathrm{WB}_{\mathbf{b},r_{\mathbf{b}},L_{\mathbf{b}}}$ следует существование дерева $\mathrm{WB}_{\mathbf{b},r_{\mathbf{b}}-1,L_{\mathbf{b}}}$ для $r_{\mathbf{b}} > 1$.*

Исходные параметры: $r(\in \mathbb{N})$, $\mathcal{A}$ – последовательность чисел из $\mathbb{F}$, $|\mathcal{A}| = r$
Результат: элемент $v(\in \mathbb{F})$, который в последовательности $\mathcal{A}$ встречается наибольшее число раз
для каждого $a \in \mathbb{F}$ **выполнять**
  вычислить величину $\mathrm{count}(a)$, равную числу появления элемента a в последовательности $\mathcal{A}$
конец цикла
если *найдется только один* $a'(\in \mathbb{F})$, *что* $\mathrm{count}(a') \ge \lceil r/2 \rceil$ **тогда**
  $v := a'$
иначе
  $v := 0$
конец условия
возвратить v

Алгоритм 2: MajorVote

С каждой меткой $\mathbf{p}$ в дереве $\mathrm{WB}_{\mathbf{b},r_\mathbf{b},L_\mathbf{b}}$ будет связано числовое значение $l(\mathbf{p})(\in \mathbb{F})$ метки. Алгоритм вычисления этих меток, по сути, и есть алгоритм вычисления значения вектора ошибок в координате $\mathbf{b}$. Поэтому в дальнейшем дерево $\mathrm{WB}_{\mathbf{b},r_\mathbf{b},L_\mathbf{b}}$ будем называть *декодирующим* деревом. Для удобства далее символом $\mathcal{M}_\mathbf{p}$ будем обозначать M-ортогональное множество, состоящее из векторов, которыми помечены вершины, непосредственно следующие из вершины с меткой $\mathbf{p}$, а символом V_i обозначим множество всех вершин на уровне i дерева $\mathrm{WB}_{\mathbf{b},r_\mathbf{b},L_\mathbf{b}}$. Пусть также $l[\mathcal{M}_\mathbf{p}] := (l(\mathbf{q}))_{\mathbf{q}\in\mathcal{M}_\mathbf{p}}$.

Исходные параметры: $\mathbf{b}(\in B)$, принятый из канала вектор $\mathbf{v}$ вида (3),
декодирующее дерево $\mathrm{WB}_{\mathbf{b},r_\mathbf{b},L_\mathbf{b}}$
Результат: значение координаты $c_\mathbf{b}$ кодового вектора
для каждого $\mathbf{p} \in V_{L_\mathbf{b}-1}$ **выполнять**
  $l(\mathbf{p}) := \mathrm{Decoder1}(\mathbf{p}, \mathcal{M}_\mathbf{p}, \mathbf{v})$
конец цикла
$\lambda := L_\mathbf{b} - 2$;
повторять
  для каждого $\mathbf{p} \in V_\lambda$ **выполнять**
    $l(\mathbf{p}) := \mathrm{MajorVote}(r_\mathbf{b}, l[\mathcal{M}_\mathbf{p}])$
  конец цикла
  $\lambda := \lambda - 1$;
до тех пор, пока $\lambda \geq 1$;
$c_\mathbf{b} := v_\mathbf{b} - l(\mathbf{b})$;
возвратить $c_\mathbf{b}$

Алгоритм 3: Decoder2

Лемма 2. *Пусть C – линейный код в V, B – базис в V, $\mathbf{b} \in B$, $\mathbf{v}$ – вектор на выходе канала связи вида (3). Если найдется декодирующее дерево $\mathrm{WB}_{\mathbf{b},r_\mathbf{b},L_\mathbf{b}}[C]$, то значение координаты $c_\mathbf{b}$ в кодовом векторе может быть восстановлено с помощью алгоритма* Decoder2.

Доказательство. В алгоритме Decoder2 с помощью алгоритмов Decoder1 и MajorVote вычисляются значения линейных комбинаций координат вектора ошибок. Корректность этих значений следует из леммы 1, так как по условию $w_B(\mathbf{e}) \leq \lfloor r_\mathbf{b}/2 \rfloor$. При $\lambda = 1$ множество V_λ состоит из одной вершины с меткой $\mathbf{b}$, поэтому значение $l(\mathbf{b})$ также будет вычислено корректно и $l(\mathbf{b}) = e_\mathbf{b}$. Отсюда $c_\mathbf{b} = v_\mathbf{b} - l(\mathbf{b})$. □

1.2. Обобщение мажоритарного декодера Дж. Мэсси

Рассмотрим код $C(\subseteq V)$, B – базис в V,

$$\mathcal{WB}(C) = \{\mathrm{WB}_{\mathbf{b},r_\mathbf{b},L_\mathbf{b}}\}_{\mathbf{b}\in B} \tag{5}$$

– набор всех декодирующих деревьев для кода C. Если для всех $\mathbf{b}(\in B)$ выполняется равенство $\mathrm{dist}_B(C) = r_\mathbf{b} + 1$, то такой код называется MLD-кодом, при этом если $L_\mathbf{b} = L$ для каждого $\mathbf{b}(\in B)$, то такой код называется L-MLD-кодом [8]. Примеры

L-MLD кодов приведены в разделах 2.2 и 2.3. Код C, такой что для каждого $\mathbf{b}(\in B)$ можно построить декодирующее дерево $\mathrm{WB}_{\mathbf{b},r_{\mathbf{b}},L_{\mathbf{b}}}[C]$ с $r_{\mathbf{b}} \geq 2$, будем называть *обобщенным* MLD-кодом. Очевидно, что MLD-коды и L-MLD-коды входят в более широкий класс обобщенных MLD-кодов. Далее будем полагать, что $r_{\mathbf{b}}$ принимает максимальное из возможных значений для данного $\mathbf{b}$. Величина $\mathrm{dmaj}_B(C) := \min_{\mathbf{b}\in B}\{r_{\mathbf{b}}\}$ называется MLD-расстоянием кода C [8]; из (4) следует, что $\mathrm{dist}_B(C) \geq \mathrm{dmaj}_B(C)+1$.

В силу леммы 2 алгоритм Decoder2 позволяет гарантированно исправить любые векторы ошибок веса не более $\lfloor \mathrm{dmaj}_B(C)/2 \rfloor$. И в этом случае декодирующее дерево для каждого базисного вектора $\mathbf{b}$ можно минимизировать так, чтобы каждый узел на уровне $i (0 \leq i \leq L_{\mathbf{b}} - 1)$ был соединен с $\mathrm{dmaj}_B(C)$ узлами на уровне $i+1$. Такая минимизация позволит, во-первых, сократить количество вычислений скалярных произведений в алгоритме Decoder1 (который также используется в алгоритме Decoder2) и, во-вторых, сократить длину последовательности (2), которая используется в обоих алгоритмах декодирования посредством обращения их к алгоритму MajorVote.

Построим алгоритм Decoder3, который в ряде случаев может исправить ошибки веса больше, чем $\lfloor \mathrm{dmaj}_B(C)/2 \rfloor$. Упорядочим множество (5):

$$\mathcal{WB}(C) := \{\mathrm{WB}_{\mathbf{b}_1,r_{\mathbf{b}_1},L_{\mathbf{b}_1}}; ...; \mathrm{WB}_{\mathbf{b}_n,r_{\mathbf{b}_n},L_{\mathbf{b}_n}}\}, \quad r_{\mathbf{b}_1} \geq ... \geq r_{\mathbf{b}_n} (\geq 2). \tag{6}$$

Исходные параметры: принятый вектор $\mathbf{v} = (v_1, ..., v_n) = \mathbf{c} + \mathbf{e}$, набор декодирующих деревьев $\mathcal{WB}(C)$ вида (6)
Результат: вектор $\mathbf{c}'$ – результат декодирования
$w := \lfloor r_{\mathbf{b}_1}/2 \rfloor$, $i := 1$, $exit := 0$;
до тех пор, пока $i \leq n$ *и* $exit = 0$ **выполнять**
  $a := \mathrm{Decoder2}(\mathbf{v}, \mathbf{b}_i, \mathrm{WB}_{\mathbf{b}_i,r_{\mathbf{b}_i},L_{\mathbf{b}_i}})$;
  если $a \neq v_i$ **тогда**
    $v_i := a$;
    $w := w - 1$;
  конец условия
  если $w = 0$ **тогда**
    $exit := 1$;
  иначе
    если $\lfloor r_{\mathbf{b}_{i+1}}/2 \rfloor < w$ **тогда**
      $w := \lfloor r_{\mathbf{b}_{i+1}}/2 \rfloor$;
    конец условия
  конец условия
  i:=i+1;
конец цикла
$\mathbf{c}' := \mathbf{v}$;
возвратить $\mathbf{c}'$

Алгоритм 4: Decoder3

Поясним алгоритм. При упорядочении (6) $\mathbf{b}_i$-координаты кодовых слов являются не менее значащими, чем $\mathbf{b}_j$-координаты для всех $1 \leq i < j \leq n$, так как

в этом случае декодирующее дерево $\mathrm{WB}_{\mathbf{b}_i, r_{\mathbf{b}_i}, L_{\mathbf{b}_i}}$ позволяет корректно восстановить значение $\mathbf{b}_i$-координаты при весе ошибки, не меньшем чем при корректном восстановлении $\mathbf{b}_j$-координаты. Переменная w хранит максимальный вес ошибки, при котором может быть правильно декодировано значение хотя бы одной координаты. Если декодированное значение координаты отличается от принятого из канала значения, то считается, что исправлена ошибка и значение переменной w уменьшается. Если исправлены все возможные ошибки ($w = 0$), то алгоритм завершает работу и возвращается результат декодирования. Если же $w > 0$, то далее проверяется, что следующее дерево в упорядоченном наборе (5) позволяет исправить w и менее ошибок. Если $\lfloor r_{\mathbf{b}_{i+1}}/2 \rfloor < w$, то это означает, что следующее дерево (вообще говоря, все последующие деревья) не позволяет исправить w и менее ошибок. В этом случае значение переменной w корректируется так, чтобы следующее дерево могло исправить w и менее ошибок. Такая коррекция гарантирует, что алгоритм всегда будет корректно работать, если ошибок произошло не более чем $\lfloor \mathrm{dmaj}_B(C)/2 \rfloor$.

Лемма 3. *Пусть C – линейный код в V, B – базис в V, $\mathbf{v} = \mathbf{c} + \mathbf{e}$, (6) – упорядоченный набор декодирующих деревьев. Если $w_B(\mathbf{e}) \le \lfloor \mathrm{dmaj}_B(C)/2 \rfloor$, то кодовый вектор $\mathbf{c}$ восстанавливается алгоритмом* Decoder3.

Доказательство. В этом случае кодовый вектор восстанавливается путем применения n раз алгоритма Decoder2, корректность результата работы которого при $w_B(\mathbf{e}) \le \lfloor \mathrm{dmaj}_B(C)/2 \rfloor$ следует из леммы 2. □

Заметим, что у каждого базисного вектора $\mathbf{b}(\in B)$ в общем случае число $r_{\mathbf{b}}$ для обобщенного MLD-кода свое. Это означает, что символы кодового слова имеют неравную защиту при использовании декодера Decoder2. Это свойство, например, потенциально позволяет с помощью декодера Decoder2 исправлять наиболее значащие символы кодового слова даже при большом уровне помех. Методам кодирования и декодирования кодов с неравной защитой символов посвящена, например, работа [15]. Применение алгоритма Decoder3 представляется целесообразным в случае, когда почти всегда ошибки имеют вес не более $\lfloor \mathrm{dmaj}_B(C)/2 \rfloor$ и в редких случаях появляются ошибки большего веса.

В разделах 2 и 3 рассматриваются коды, для которых применим алгоритм декодирования Decoder3.

2. Групповые MLD-коды

2.1. Групповые алгебры и групповые коды

Пусть $\mathcal{G}$ — конечная группа, $\mathbb{F}$ — поле Галуа. Рассмотрим групповую алгебру $\mathbb{F}\mathcal{G}$, элементами которой являются формальные суммы (функции):

$$\sum_{g \in \mathcal{G}} a_g g, a_g \in \mathbb{F}. \tag{7}$$

Сложение в $\mathbb{F}\mathcal{G}$ выполняется покомпонентно, а умножение $\sum_{g\in\mathcal{G}} a_g g$ и $\sum_{g\in\mathcal{G}} b_g g$ выполняется по правилу:

$$\left(\sum_{g\in\mathcal{G}} a_g g\right)\left(\sum_{g\in\mathcal{G}} b_g g\right) := \sum_g \left[\sum_w a_{gw^{-1}} b_w\right] g, \tag{8}$$

при этом внешняя сумма в правой части (8) является формальной, а внутренняя сумма – это сумма над полем $\mathbb{F}$.

Функцию, принимающую значение 1 ($\in \mathbb{F}$) на нейтральном элементе $\hat{1}$ группы $\mathcal{G}$, а на всех остальных элементах группы принимающая значение 0 ($\in \mathbb{F}$), будем называть единицей групповой алгебры $\mathbb{F}\mathcal{G}$ и обозначать $\mathbf{1}$; функцию, принимающую значение 0 на всех элементах группы $\mathcal{G}$, назовем нулем групповой алгебры и обозначим $\mathbf{0}$. В формальных суммах $\sum_{g\in\mathcal{G}} a_g g$, где не все коэффициенты a_g равны нулю, слагаемые с нулевыми коэффициентами обычно будут опускаться. Далее функцию Дирака в точке g, как принято, будем обозначать $\delta_g = 1g$. Отметим, что функция $1\hat{1}$ совпадает с $\delta_{\hat{1}} = \mathbf{1}$, а произведение функций $1x = \delta_x$ и $1y = \delta_y$ в $\mathbb{F}\mathcal{G}$ равно $\delta_x\delta_y = \delta_{xy}$. Элементы групповой алгебры можно записывать в виде: $\sum_{g\in\mathcal{G}} a_g\delta_g, a_g \in \mathbb{F}$.

В конечномерной групповой алгебре $\mathbb{F}\mathcal{G}$ зафиксируем базис $B = B^{\mathcal{G}} := \{\mathbf{g} = \delta_g\}_{g\in\mathcal{G}}$. Это позволяет рассматривать в $\mathbb{F}\mathcal{G}$ метрику Хэмминга d_B. Отметим, что в категории конечномерных линейных пространств $\mathbb{F}\mathcal{G}$ и $\mathbb{F}^{|\mathcal{G}|}$ изоморфны. Группа $\mathcal{G}$ действует слева на групповой алгебре $\mathbb{F}\mathcal{G}$ следующим естественным образом (см. [8], с. 32):

$$\mathcal{G}\times\mathbb{F}\mathcal{G} \ni (g, \phi = \sum_{h\in\mathcal{G}} \phi_h h) \mapsto \phi g^{-1} := \sum_{h\in\mathcal{G}} \phi_{hg^{-1}} h \in \mathbb{F}\mathcal{G}. \tag{9}$$

Пусть A – алгебра, $A_0 \subset A$. Через $< A_0 >$ обозначим наименьшую подалгебру алгебры A, содержащую A_0. Если $A =< A_0 >$, то будем говорить, что A_0 порождает A. Если $\mathcal{G}_0$ — подгруппа группы $\mathcal{G}$, то, как легко видеть, $\mathbb{F}\mathcal{G}_0$ — подалгебра алгебры $\mathbb{F}\mathcal{G}$. Если $\mathcal{I}$ — левый (правый, двусторонний) идеал алгебры $\mathbb{F}\mathcal{G}$, то через $\mathcal{I}^t$ обозначим подалгебру алгебры $\mathbb{F}\mathcal{G}$, порожденную элементами вида $\mathbf{x}_1 ... \mathbf{x}_t$, где $\mathbf{x}_i \in \mathcal{I}$; известно, что $\mathcal{I}^t$ – это левый (правый, двусторонний) идеал алгебры $\mathbb{F}\mathcal{G}$.

В соответствии с [8], с. 39, всякий отличный от $\{\mathbf{0}\}$ *левый* идеал C в групповой алгебре $\mathbb{F}\mathcal{G}$ называется *групповым кодом* ($\mathbb{F}\mathcal{G}$–кодом) длины $n(C) = |\mathcal{G}|$. Отметим, что как левый идеал код является также левым $\mathbb{F}\mathcal{G}$–модулем. Пусть $< \cdot, \cdot >$ — невырожденная симметрическая $\mathbb{F}$-билинейная форма на $\mathbb{F}\mathcal{G}$, которая однозначно определяется условием: для любых $g(\in \mathcal{G})$ и $h(\in \mathcal{G})$

$$< \delta_g, \delta_h >= \begin{cases} 1, & \text{если } g = h, \\ 0, & \text{если } g \neq h. \end{cases}$$

Если C — произвольный $\mathbb{F}\mathcal{G}$-код, то множество

$$C^{\perp} = \{\mathbf{x} \in \mathbb{F}\mathcal{G} \mid \forall \mathbf{z} \in C \ < \mathbf{x}, \mathbf{z} >= 0\}$$

является $\mathbb{F}\mathcal{G}$-кодом (см. [8], теорема 1.6.5c) и называется дуальным кодом к коду C.

Для групповых кодов с целью их применения в борьбе с помехами в системах передачи данных необходимо, разумеется, задать правило кодирования. Пусть C —

групповой код размерности $k = k(C)$, $C \subseteq \mathbb{F}\mathcal{G}$, $B(C) := \{\epsilon_1; ...; \epsilon_k\}$ – базис кода C, где для $i = 1, ..., k$ имеет место представление:

$$\epsilon_i = \sum_{g \in \mathcal{G}} \alpha_{i,g}\delta_g, \quad \alpha_{i,g} \in \mathbb{F}.$$

Тогда при кодировании произвольному информационному вектору $\mathbf{s} = (s_1, ..., s_k)$ ставится в соответствие кодовый вектор $\mathbf{c}(\in C)$ вида:

$$\mathbf{c} = \sum_{i=1}^{k} s_i\epsilon_i = \sum_{i=1}^{k} s_i \sum_{g \in \mathcal{G}} \alpha_{i,g}\delta_g = \sum_{g \in \mathcal{G}} (\sum_{i=1}^{k} s_i\alpha_{i,g})\delta_g.$$

Для определения порождающей матрицы $G(C)$ группового кода $C(\subseteq \mathbb{F}\mathcal{G})$ необходимо задать линейный порядок на множестве элементов группы $\mathcal{G}$ и, таким образом, на базисе B. Тогда строками порождающей матрицы $G(C)$ будут являться элементы базиса $B(C)$, представленные в виде упорядоченных наборов коэффициентов в формальных суммах вида (7), а столбцами этой матрицы будут векторы вида $(\alpha_{1,g}, ..., \alpha_{k,g})$, $g \in \mathcal{G}$. Другими словами, порождающая матрица $G(C)$ – это матричное представление базиса $B(C)$ через базис B групповой алгебры $\mathbb{F}\mathcal{G}$. Аналогично можно определить и проверочную матрицу $H(C)$, совпадающую с $G(C^{\perp})$.

Напомним, что пересечение всех максимальных левых идеалов $\mathbb{F}$–алгебры $\mathcal{A}$ называется радикалом $\mathrm{Rad}\mathcal{A}$ (см. [8], с.69); $\mathrm{Rad}\mathcal{A}$ — пример группового кода. Так как степень левого идеала — левый идеал, то и $(\mathrm{Rad}\mathcal{A})^t$ – групповой код.

Для произвольного подмножества X алгебры $\mathcal{A}$ в [8], с. 69, определен левый идеал

$$\mathcal{A} \cdot X := \sum_{x \in X} \mathcal{A}x(\subseteq \mathcal{A}), \tag{10}$$

порожденный множеством X. Следовательно, $\mathcal{A} \cdot X$ — тоже групповой код.

В заключение рассмотрим важное для дальнейшего семейство групповых кодов, открытое С.Д. Берманом [5]. Пусть $\mathbb{F} = \mathbb{F}_2$, $\Delta_{2,n}$ – 2-группа порядка 2^n, имеющая представление в виде внутреннего произведения n подгрупп, каждая из которых имеет порядок 2:

$$\Delta_{2,n} =< g_1 > \times ... \times < g_n >, \tag{11}$$

где для $i = 1, ..., n$, элемент g_i – образующий элемент порядка 2, $g_i^2 = \hat{1}$. Отметим, что $\Delta_{2,n}$ – это аддитивная группа поля $\mathbb{F}_{2^n}$. Множество $\{h_1; ...; h_n\}$ называется минимальной системой порождающих элементов группы $\Delta_{2,n}$, если любой элемент $g(\in \Delta_{2,n})$ может быть представлен в виде $g = h_1^{a_1}...h_n^{a_n}$, $(a_1, ..., a_n \in \{0; 1\})$, и никакое подмножество множества $\{h_1; ...; h_n\}$ этим свойством не обладает [8]. В групповой алгебре $\mathbb{F}_2\Delta_{2,n}$ Берманом построено семейство кодов, изоморфное семейству классических кодов Рида–Маллера $\mathcal{RM}_2(r, n)$. Приведем соответствующие конструкции. В коммутативной групповой алгебре $\mathbb{F}_2\Delta_{2,n}$, которая как $\mathbb{F}_2$–линейное пространство изоморфна пространству $\mathbb{F}_2^{2^n}$, кроме стандартного упорядоченного базиса

$$B = \{\mathbf{1}; \delta_{g_1}; ...; \delta_{g_n}; \delta_{g_1}\delta_{g_2}; ...; \delta_{g_1}\delta_{g_n}; ...; \delta_{g_{n-1}}\delta_{g_n}; ...; \delta_{g_1}...\delta_{g_n}\}$$

рассмотрим базис

$$B_0 = \left\{ (\delta_{g_1} - \mathbf{1})^{a_1}...(\delta_{g_n} - \mathbf{1})^{a_n} | a_1, ..., a_n \in \{0; 1\}, \sum_{s=1}^{n} a_s \geq 0 \right\}$$

(см. [8], с. 49) и зафиксируем *лексикографическое* упорядочение по степеням $(\delta_{g_i}-\mathbf{1})$ в этом базисе:

$$B_0 = \{\mathbf{1}; (\delta_{g_1}-\mathbf{1}); ...; (\delta_{g_n}-\mathbf{1}); (\delta_{g_1}-\mathbf{1})(\delta_{g_2}-\mathbf{1}); ...; (\delta_{g_1}-\mathbf{1})(\delta_{g_n}-\mathbf{1}); ...; (\delta_{g_1}-\mathbf{1})...(\delta_{g_n}-\mathbf{1})\}.$$

В $\mathbb{F}_2\Delta_{2,n}$ рассмотрим $\mathbb{F}_2$–подпространство RM_t, порожденное множеством:

$$B_t := \left\{ (\delta_{g_1} - \mathbf{1})^{a_1}...(\delta_{g_n} - \mathbf{1})^{a_n} | a_1, ..., a_n \in \{0; 1\}, \sum_{s=1}^{n} a_s \geq t \right\}, \tag{12}$$

где $t \in \mathbb{N}$. Из определения RM_t вытекает, что

$$\mathrm{RM}_n = \{\mathbf{0}; (\delta_{g_1} - \mathbf{1})...(\delta_{g_n} - \mathbf{1})\}. \tag{13}$$

Из вложения $B_0 \supset B_1 \supset B_2 \supset ... \supset B_n \supset \{\mathbf{0}\}$ следует вложение $\mathbb{F}_2\Delta_{2,n} = \mathrm{RM}_0 \supset \mathrm{RM}_1 \supset \mathrm{RM}_2 \supset ... \supset \mathrm{RM}_n \supset \{\mathbf{0}\}$.

В [8] показано, что для любого t пространство $\mathrm{RM}_t = (\mathrm{RM}_1)^t$ и является идеалом в $\mathbb{F}\Delta_{2,n}$, причем $\mathrm{RM}_1 = \mathrm{Rad}\mathbb{F}_2\Delta_{2,n}$. Таким образом, для $t = 1, ..., n$ пространство RM_t является групповым кодом размерности $k(\mathrm{RM}_t) = \sum_{i=t}^{n} C_n^t$ и длины $n(\mathrm{RM}_t) = 2^n$. Так как код RM_t является подпространством $\mathbb{F}_2$-пространства $\mathbb{F}_2^{2^n}$, то для RM_t можно предъявить порождающую матрицу $G(\mathrm{RM}_t)$, в которой $k(\mathrm{RM}_t)$ строк и $n(\mathrm{RM}_t)$ столбцов. Для явного выписывания матрицы $G(\mathrm{RM}_t)$ достаточно воспользоваться упорядоченным базисом B пространства $\mathbb{F}_2^{2^n}$ и записать матричное представление базиса B_t в базисе B. В работе [5] доказано, что для любого t код RM_t естественно изоморфен классическому коду Рида–Маллера $\mathcal{RM}_2(n-t, n)$. В частности, отсюда вытекает, что согласно [17], с. 203, $\mathrm{RM}_t^{\perp} = \mathrm{RM}_{n-t+1}$. Далее эти коды будем называть кодами Рида–Маллера–Бермана.

2.2. Декодирование групповых MLD–кодов

В классе MLD–кодов групповые MLD–коды играют особую роль. Именно, группа $\mathcal{G}$ действует транзитивно по правилу (9) на базисных элементах из $B^{\mathcal{G}}$, поэтому по декодирующему дереву для какого-либо одного базисного элемента $B^{\mathcal{G}}$ можно найти декодирующие деревья для всех остальных элементов из базиса. Построим соответствующий алгоритм CloneTree.

Исходные параметры: $\mathcal{G}$, $\mathrm{WB}_{\mathbf{b},r_{\mathbf{b}},L_{\mathbf{b}}}[C]$, $\mathbf{b}'$
Результат: $\mathrm{WB}_{\mathbf{b}',r'_{\mathbf{b}},L_{\mathbf{b}'}}[C]$
$\mathrm{WB}_{\mathbf{b}',r'_{\mathbf{b}},L_{\mathbf{b}'}}[C] := \mathrm{WB}_{\mathbf{b},r_{\mathbf{b}},L_{\mathbf{b}}}[C]$;
Найти $g(\in \mathcal{G})$ такой, что $(g, \mathbf{b}) = \mathbf{b}'$;
для каждой *метки* $\mathbf{p}$ *дерева* $\mathrm{WB}_{\mathbf{b}',r'_{\mathbf{b}},L_{\mathbf{b}'}}[C]$ **выполнять**
  // Действие элементом g на $\mathbf{p}$ по правилу (9);
  $\mathbf{p} := (g, \mathbf{p})$;
конец цикла
возвратить $\mathrm{WB}_{\mathbf{b}',r'_{\mathbf{b}},L_{\mathbf{b}'}}[C]$

Алгоритм 5: CloneTree

Так как $C^{\perp}$ – идеал, то действие группы $\mathcal{G}$ по правилу (9) на элементах кода $C^{\perp}$ не выводит за границы этого кода. Поэтому листья нового декодирующего дерева на выходе алгоритма CloneTree будут принадлежать коду $C^{\perp}$, при этом свойство M-ортогональности в этом дереве не нарушается.

Все декодирующие деревья кода C можно построить по алгоритму MakeAllTrees.

Исходные параметры: $\mathcal{G}$, $\mathrm{WB}_{\mathbf{1}_{\mathcal{G}},r_{\mathbf{1}_{\mathcal{G}}},L_{\mathbf{1}_{\mathcal{G}}}}[C]$
Результат: $\mathcal{WB}(C)$
$\mathcal{WB}(C) := \emptyset$;
для каждого $\mathbf{b} \in B^{\mathcal{G}}$ **выполнять**
  $\mathcal{WB}(C) := \mathcal{WB}(C) \cup \mathrm{CloneTree}(\mathcal{G}, \mathrm{WB}_{\mathbf{1}_{\mathcal{G}},r_{\mathbf{1}_{\mathcal{G}}},L_{\mathbf{1}_{\mathcal{G}}}}[C], \mathbf{b})$
конец цикла
возвратить $\mathcal{WB}(C)$

Алгоритм 6: MakeAllTrees

В силу алгоритма MakeAllTrees, все декодирующие деревья группового MLD-кода C имеют одинаковую структуру, а отличаются только метками узлов. В этом случае представления вида (5) и (6) совпадают. Таким образом, для применения алгоритма декодирования Decoder3 для группового MLD-кода C достаточно иметь одно декодирующее дерево $\mathrm{WB}_{\mathbf{1}_{\mathcal{G}},r_{\mathbf{1}_{\mathcal{G}}},L_{\mathbf{1}_{\mathcal{G}}}}[C]$, так как все остальные деревья строятся алгоритмом MakeAllTrees.

Отметим, что коды Рида–Маллера–Бермана, рассмотренные в разделе 2.1., являются групповыми MLD-кодами.

2.3. Индуцированные групповые MLD-коды

Пусть $\mathcal{G}$ — конечная группа, $\mathcal{H}$ — ее нормальная подгруппа, $\lambda = [\mathcal{G} : \mathcal{H}]$ — индекс подгруппы $\mathcal{H}$ в группе $\mathcal{G}$. Группа $\mathcal{G}$ разбивается на множество непересекающихся правых классов смежности $\{\mathcal{H}y\}_{y\in Y}$, где $Y = \{y_1 = \hat{1}; y_2; ...; y_\lambda\}(\subset \mathcal{G})$ — линейно упорядоченная правая трансверсаль $\mathcal{G}$ по $\mathcal{H}$. Групповая алгебра $\mathbb{F}\mathcal{G}$ является свободным левым $\mathbb{F}\mathcal{H}$-модулем с базисом Y, т.е.

$$\mathbb{F}\mathcal{G} = \bigoplus_{i=1}^{\lambda} (\mathbb{F}\mathcal{H})\delta_{y_i}. \tag{14}$$

Если $B^{\mathcal{H}} = \{\delta_h\}_{h\in\mathcal{H}}$ — базис в линейном векторном пространстве $\mathbb{F}\mathcal{H}$, то базис $B^{\mathcal{G}} = \{\delta_g\}_{g\in\mathcal{G}}$ в $\mathbb{F}\mathcal{G}$ представим в виде: $B^{\mathcal{G}} = \cup_{i=1}^{\lambda} B^{\mathcal{H}}\delta_{y_i}$ (см. (14)).

Рассмотрим $\mathbb{F}\mathcal{H}$–код N размерности $k(N) = |B(N)|$ и длины $n(N) = |\mathcal{H}|$ с $\mathbb{F}$-базисом $B(N) = \{\eta_1; ...; \eta_{k(N)}\}$, $\eta_i = \sum_{h\in\mathcal{H}} \beta_{i,h}\delta_h$; пусть $\mathcal{WC}(N)$ – множество декодирующих деревьев кода N, например, построенное с помощью алгоритма MakeAllTrees. В силу [8], с. 84, код $\mathbb{F}\mathcal{G}\cdot N$ как левый $\mathbb{F}\mathcal{G}$–модуль изоморфен тензорному произведению $\mathbb{F}\mathcal{G} \otimes_{\mathbb{F}\mathcal{H}} N(\subseteq \mathbb{F}\mathcal{G})$, определенному в [16], с. 80, при этом размерность $k(\mathbb{F}\mathcal{G} \otimes_{\mathbb{F}\mathcal{H}} N) = |Y|k(N)$, а длина $n(\mathbb{F}\mathcal{G} \otimes_{\mathbb{F}\mathcal{H}} N) = |\mathcal{G}|$ и совпадает с $|Y|n(N)$. Этот изоморфизм является также $\mathbb{F}$–изометрией относительно метрики Хемминга. Говорят, что код $\mathbb{F}\mathcal{G} \otimes_{\mathbb{F}\mathcal{H}} N$ индуцирован кодом N. Из определений вытекает, что базис индуцированного кода состоит из векторов (функций) вида:

$$\kappa_{i,j} = \eta_i\delta_{y_j}, \quad i = 1, ..., k(N), \quad j = 1, ..., \lambda.$$

Введем линейное упорядочение на базисных векторах индуцированного кода: $\kappa_l := \kappa_{i,j}$, где $l = i + (j-1)k(N)$. Рассмотрим две такие вспомогательные функции $i : \{1;...;k(N)\lambda\} \to \{1;...;k(N)\}$ и $j : \{1;...;k(N)\lambda\} \to \{1;...;\lambda\}$, что $i(l) + (j(l) - 1)k(N) = l$. Эти две функции позволяют переходить от одинарной нумерации базисных векторов к двойной. Тогда информационному вектору $\mathbf{s} = (s_1, ..., s_{k(N)\lambda})(\in \mathbb{F}^{k(N)\lambda})$ соответствует кодовое слово вида:

$$\mathbf{c} = \sum_{l=1}^{k(N)\lambda} s_l\kappa_l = \sum_{l=1}^{k(N)\lambda} s_l\kappa_{i(l),j(l)} = \sum_{l=1}^{k(N)\lambda} s_l\eta_{i(l)}\delta_{y_{j(l)}}$$
$$= \sum_{l=1}^{k(N)\lambda} s_l \sum_{h\in\mathcal{H}} \beta_{i(l),h}\delta_h\delta_{y_{j(l)}} = \sum_{h\in\mathcal{H}} \sum_{l=1}^{k(N)\lambda} s_l\beta_{i(l),h}\delta_{hy_{j(l)}}.$$

Таким образом, при фиксированном $h(\in \mathcal{H})$ вычисляются λ коэффициентов кодового вектора $\mathbf{c} = (c_1, ..., c_{n(N)\lambda})$ при базисных элементах $\delta_{hy_1}, ..., \delta_{hy_\lambda}$:

$$c_{\delta_{hy_v}} = \left\langle (s_{1+(v-1)k(N)}, ..., s_{vk(N)}), (\beta_{1,h}, ..., \beta_{k(N),h}) \right\rangle \quad v = 1, ..., \lambda,$$

где $\langle \mathbf{a}, \mathbf{b} \rangle$ — скалярное произведение векторов $\mathbf{a}$ и $\mathbf{b}$. Пусть $\chi = |\mathcal{H}|$. Занумеруем элементы подгруппы $\mathcal{H}$: $\mathcal{H} = \{h_1; ...; h_\chi\}$. Тогда можно упорядочить базис алгебры $\mathbb{F}\mathcal{G}$ так, что кодовый вектор $\mathbf{c}$ можно представить в виде:

$$\mathbf{c} = (\mathbf{c}^1||...||\mathbf{c}^\lambda), \mathbf{c}^i = (c_{\delta_{h_1 y_i}}, ..., c_{\delta_{h_\chi y_i}}) \in N, i = 1, ..., \lambda, \tag{15}$$

где $(\mathbf{a}||\mathbf{b})$ — приписывание вектора $\mathbf{b}$ справа к вектору $\mathbf{a}$. Итак, если $G(N)$ – порождающая $(k(N) \times n(N))$-матрица кода N, то кодирование для индуцированного кода представимо в виде умножения вектора длины $k(N)\lambda$ на матрицу вида:

$$G(\mathbb{F}\mathcal{G} \otimes_{\mathbb{F}\mathcal{H}} N) = \left(\begin{array}{c|c|c|c} G(N) & O & ... & O \\ \hline O & G(N) & ... & O \\ \hline ... & ... & ... & ... \\ \hline O & ... & ... & G(N) \end{array}\right), \tag{16}$$

где O – нулевая $(k(N) \times n(N))$-матрица. Из представления (16) порождающей матрицы индуцированного кода вытекают доказанные в [8] равенства: $\mathrm{dmaj}_{B^\mathcal{G}}(\mathbb{F}\mathcal{G}\cdot N) = \mathrm{dmaj}_{B^\mathcal{H}}(N)$ и $\mathrm{dist}_{B^\mathcal{G}}(\mathbb{F}\mathcal{G}\cdot N) = \mathrm{dist}_{B^\mathcal{H}}(N)$. Вид (16) порождающей матрицы позволяет построить алгоритм мажоритарного декодирования Decoder4.

Теорема 1. *Пусть $\mathcal{H}$ – нормальный делитель группы $\mathcal{G}$, $N(\subseteq \mathbb{F}\mathcal{H})$ – групповой* MLD*-код, $\mathbb{F}\mathcal{G} \otimes_{\mathbb{F}\mathcal{H}} N(\subseteq \mathbb{F}\mathcal{G})$ – индуцированный код. Тогда алгоритм* Decoder4 *гарантированно исправляет любые ошибки веса не более $\lfloor \mathrm{dmaj}_{B^\mathcal{H}}(N)/2 \rfloor$.*

Доказательство. Так как, по условию, вес ошибки не более $\lfloor \mathrm{dmaj}_{B^\mathcal{H}}(N)/2 \rfloor$, то в представлении (17) каждый вектор $\mathbf{v}^i$ имеет вид: $\mathbf{v}^i = \mathbf{c}^i + \mathbf{e}^i$, где $w_{B^\mathcal{H}}(\mathbf{e}^i) \leq \lfloor \mathrm{dmaj}_{B^\mathcal{H}}(N)/2 \rfloor$, $\mathbf{c}^i \in N$ (см. (15)). Тогда, по лемме 3, алгоритм Decoder3 гарантированно восстановит вектор $\mathbf{c}^i$ для всех $i = 1, ..., \lambda$. □

Исходные параметры: $\mathcal{H}$ – нормальный делитель группы $\mathcal{G}$, $\mathcal{WC}(N)$ – множество декодирующих деревьев группового MLD-кода $N(\subseteq \mathbb{F}\mathcal{H})$, $H(N)$ – проверочная матрица кода N, $\mathbf{v} = (v_1, ..., v_{k(N)\lambda})$ – принятый вектор, W – максимальный вес ошибки в векторе $\mathbf{v}$

Результат: $\mathbf{c}'(\in \mathbb{F}\mathcal{G} \otimes_{\mathbb{F}\mathcal{H}} N)$ – результат декодирования

Представить вектор $\mathbf{v}$ в виде конкатенации λ векторов:

$$\mathbf{v} = (\mathbf{v}^1||...||\mathbf{v}^\lambda), \quad \mathbf{v}^i \in \mathbb{F}^{n(N)}, \quad i = 1, ..., \lambda; \tag{17}$$

$j := 1$, $\omega := 0$, exit $:= 0$;
до тех пор, пока $j \leq \lambda$ *и* exit $= 0$ **выполнять**
  если $\omega \geq W$ **тогда**
    exit $:= 1$;
    $i := j$ **до тех пор, пока** $i \leq \lambda$ **выполнять**
      $\mathbf{c}'^i := \mathbf{v}^i$;
      $i := i + 1$;
    конец цикла
  иначе
    если $\mathbf{v}^i H(N)^\top = \mathbf{0}(\in \mathbb{F}^{n(N)})$ **тогда**
      // Блок не содержит ошибок;
      $\mathbf{c}'^i := \mathbf{v}^i$;
    иначе
      $\mathbf{c}'^i = \text{Decoder3}(\mathbf{v}^i, \mathcal{WB}(N))$;
      $\omega := \omega + w_{B^\mathcal{H}}(\mathbf{c}'^i - \mathbf{v}^i)$;
    конец условия
  конец условия
  $j := j + 1$;
конец цикла
возвратить $\mathbf{c}' := (\mathbf{c}'^1||...||\mathbf{c}'^\lambda)$

Алгоритм 7: Decoder4

Отметим, что у индуцированного кода $\mathbb{F}\mathcal{G} \cdot N$ отношение размерность/длина равно значению аналогичного отношения для кода N, при этом отношение количества исправляемых ошибок кодом $\mathbb{F}\mathcal{G} \cdot N$ к длине кодового слова уменьшается в λ раз по отношению к аналогичной характеристике кода N. Это, в частности, означает, что если в канале передачи данных ошибки группирующиеся, то декодер Decoder4 с большой вероятностью ошибется при весе ошибки, большем чем $\lfloor \text{dmaj}_{B^\mathcal{H}}(N)/2 \rfloor$. Если же заранее известно, что вес группирующейся ошибки не более $\lfloor \text{dmaj}_{B^\mathcal{H}}(N)/2 \rfloor$, то для ее исправления в алгоритме Decoder4 (с входным параметром $W = \lfloor \text{dmaj}_{B^\mathcal{H}}(N)/2 \rfloor$) потребуется применить алгоритм Decoder3 не более двух раз, что может заметно увеличить скорость декодирования блоков длины $n(N)\lambda$. С другой стороны, если ошибки равномерно распределены по всему кодовому вектору длины $n(N)\lambda$, то алгоритм Decoder4 может исправить помехи веса до $\lambda \lfloor \text{dmaj}_{B^\mathcal{H}}(N)/2 \rfloor$. Поэтому алгоритм Decoder4 может быть применим для борьбы

с группирующимися ошибками веса до $\lambda\lfloor \mathrm{dmaj}_{B^{\mathcal{H}}}(N)/2\rfloor$, если кодовые слова перед поступлением в канал подвергаются перемежению (скремблированию). Таким образом, на основе кода N и декодера Decoder4 можно строить наборы кодов для подстраивания под помеховую обстановку в канале передачи данных. Далее приводится пример кодов, индуцированных кодами Рида–Маллера–Бермана RM_t.

3. Групповые MLD-коды типа Рида–Маллера–Бермана

3.1. Групповые коды на группе $\mathrm{Aff}(\mathbb{F}_{2^n})$, индуцированные кодами Рида–Маллера–Бермана

Рассмотрим группу аффинных преобразований $\mathcal{G} = \mathrm{Aff}(\mathbb{F}_{2^n})$ одномерного линейного пространства $\mathbb{F}_{2^n}$, элементами которой являются преобразования вида:

$$\varphi_{\alpha,\beta} : \mathbb{F}_{2^n} \ni x \to \alpha x + \beta \in \mathbb{F}_{2^n}, (\alpha, \beta \in \mathbb{F}_{2^n}, \alpha \neq 0, n > 1).$$

Эта группа неабелева и соответствие

$$\varphi_{\alpha,\beta} \mapsto \begin{pmatrix} \alpha & \beta \\ 0 & 1 \end{pmatrix}$$

определяет естественный изоморфизм $\mathcal{G}$ на группу невырожденных матриц второго порядка вида:

$$\begin{pmatrix} \alpha & \beta \\ 0 & 1 \end{pmatrix}.$$

Далее группу $\mathcal{G}$ и группу матриц будем отождествлять. Единичный элемент этой группы, как и раньше, будем обозначать

$$\hat{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Отметим, что

$$\begin{pmatrix} \alpha & \beta \\ 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} \alpha^{-1} & \alpha^{-1} - \alpha^{-1}\beta \\ 0 & 1 \end{pmatrix}.$$

Несложно проверить, что для любых $g_1, g_2 \in \mathcal{G}$ найдется элемент $g_3 (\in \mathcal{G})$ такой, что при его действии на функцию δ_{g_1} получается функция δ_{g_2} (см. (9)). Таким образом, группа $\mathcal{G}$ действует транзитивно на базисе $B^{\mathcal{G}}$ групповой алгебры $\mathbb{F}\mathcal{G}$ по правилу (9). Рассмотрим подгруппу

$$\mathcal{H} = \{\varphi_{1,\beta} | \beta \in \mathbb{F}_{2^n}\} \tag{18}$$

группы $\mathcal{G}$. Легко видеть, что $\mathcal{H}$ – нормальная подгруппа группы $\mathcal{G}$, изоморфная группе $\Delta_{2,n}$. Пусть

$$\varsigma : \mathbb{F}_2\Delta_{2,n} \to \mathbb{F}_2\mathcal{H} \tag{19}$$

— изоморфизм групповых алгебр, индуцированный изоморфизмом групп $\varsigma' : \mathcal{H} \to \Delta_{2,n}$, определенным равенством $\varsigma'(\varphi_{1,\beta}) = \beta$. Так как $\mathcal{H}$ — нормальная подгруппа

группы $\mathrm{Aff}(\mathbb{F}_{2^n})$ и $|\mathrm{Aff}(\mathbb{F}_{2^n})/\mathcal{H}| = 2^n - 1$ не делится на $\mathrm{char}(\mathbb{F}_2) = 2$, то, согласно [8], с. 74,

$$(\mathrm{Rad}\mathbb{F}_2\mathrm{Aff}(\mathbb{F}_{2^n}))^t = \mathbb{F}_2\mathrm{Aff}(\mathbb{F}_{2^n}) \cdot (\mathrm{Rad}\mathbb{F}_2\mathcal{H})^t.$$

Таким образом, в силу изоморфизма (19) получаем:

$$\mathbb{F}_2\mathrm{Aff}(\mathbb{F}_{2^n}) \otimes_{\mathbb{F}_2\mathcal{H}} \varsigma(\mathrm{RM}_t) \cong (\mathrm{Rad}\mathbb{F}_2\mathrm{Aff}(\mathbb{F}_{2^n}))^t = \mathbb{F}_2\mathrm{Aff}(\mathbb{F}_{2^n}) \cdot \varsigma(\mathrm{RM}_t).$$

Отметим, что размерность индуцированного кода $\mathbb{F}_2\mathrm{Aff}(\mathbb{F}_{2^n}) \otimes_{\mathbb{F}_2\mathcal{H}} \varsigma(\mathrm{RM}_t)$ равна $(2^n - 1)\sum_{i=t}^{n} C_n^i$, длина равна $2^n(2^n - 1)$, а порождающая матрица имеет вид (16), где $N = \mathrm{RM}_t$.

3.2. Декодирование кодов RM_t и $\mathbb{F}_2\mathrm{Aff}(\mathbb{F}_{2^n}) \otimes_{\mathbb{F}_2\mathcal{H}} \varsigma(\mathrm{RM}_t)$

Приведем сначала алгоритм построения декодирующих деревьев для кодов Рида–Маллера. Отметим, что код RM_n имеет вид (13), $\mathrm{RM}_1 = \mathrm{RM}_n^{\perp}$ и

$$\mathcal{M} = \left\{\delta_g - \mathbf{1} : g \in \Delta_{2,n}, g \neq \hat{1}\right\} \subseteq \mathrm{RM}_1.$$

Множество $\mathcal{M}$ является M-ортогональным относительно базисного элемента $\mathbf{1}(\in B^{\Delta_{2,n}})$, поэтому для $\mathbf{1}$ можно построить декодирующее дерево $\mathrm{WB}_{\mathbf{1},2^n-1,1}$, состоящее из двух уровней. Несложно видеть, что группа $\Delta_{2,n}$ по правилу (9) действует транзитивно на каноническом базисе $B^{\Delta_{2,n}}$ групповой алгебры $\mathbb{F}_2\Delta_{2,n}$. Код RM_t является идеалом в $\mathbb{F}_2\Delta_{2,n}$, поэтому он также является и $\mathbb{F}_2\Delta_{2,n}$-модулем, то есть действие любого элемента $\alpha(\in \mathbb{F}_2\Delta_{2,n})$ на функцию из RM_t не выводит за границы RM_t. Тогда для любого базисного элемента $\mathbf{b} \in B^{\Delta_{2,n}}$ декодирующее дерево кода RM_n можно построить по дереву $\mathrm{WB}_{\mathbf{1},2^n-1,1}$, используя алгоритм CloneTree:

$$\mathrm{WB}_{\mathbf{b},r_{\mathbf{b}},L_{\mathbf{b}}}[\mathrm{RM}_n] := \mathrm{CloneTree}(\Delta_{2,n}, \mathrm{WB}_{\mathbf{1},r_{\mathbf{1}},L_{\mathbf{1}}}[\mathrm{RM}_n], \mathbf{b}).$$

Таким образом, $\mathrm{dmaj}_{B^{\Delta_{2,n}}}(\mathrm{RM}_n) = \mathrm{dist}_{B^{\Delta_{2,n}}}(\mathrm{RM}_n) - 1$.

В [8] без обоснования показано, как для $1 \leq t < n$ и $\mathbf{1} \in B^{\Delta_{2,n}}$ декодирующее дерево RM_t может быть построено по декодирующему дереву для кода RM_{t+1}. Ниже построим соответствующий алгоритм и приведем обоснование. Нам понадобятся два простых естественных алгоритма. Первый алгоритм, MakeGenSystem, по элементу $\mathbf{p}(\in \mathbb{F}_2\Delta_{2,n})$ вида

$$\mathbf{p} = (\delta_{h_1} - \mathbf{1})...(\delta_{h_s} - \mathbf{1}), \tag{20}$$

где $S_{\mathbf{p}} = \{h_1; ...; h_s\}$ — подмножество некоторой минимальной системы порождающих элементов, достраивает $S_{\mathbf{p}}$ до, вообще говоря, другой минимальной системы порождающих $\mathcal{M}_{\mathbf{p}} = \{h_1; ...; h_s; f_1; ...; f_{n-s}\}$ и возвращает $\mathcal{M}_{\mathbf{p}} \backslash \mathcal{S}_{\mathbf{p}}$. Отметим, что для $\mathbf{p} \neq \mathbf{p}'$ в общем случае $\mathcal{S}_{\mathbf{p}} \neq \mathcal{S}'_{\mathbf{p}}$ и $\mathcal{M}_{\mathbf{p}} \neq \mathcal{M}'_{\mathbf{p}}$. Второй алгоритм, MakeGroup, по произвольному подмножеству S группы $\Delta_{2,n}$ строит подгруппу $< S >$ в виде списка.

Теперь приведем алгоритм MakeRMWBTree для построения декодирующего дерева кода RM_t с корнем $\mathbf{1}$ по декодирующему дереву кода RM_{t+1} с тем же корнем.

Теорема 2. *Алгоритм* MakeRMWBTree *по декодирующему дереву* $\mathrm{WB}_{\mathbf{1},2^{t+1}-1,L_{\mathbf{1}}}[\mathrm{RM}_{t+1}]$ *строит декодирующее дерево* $\mathrm{WB}_{\mathbf{1},2^t-1,L_{\mathbf{1}}+1}[\mathrm{RM}_t]$.

Исходные параметры: $t(\in\{1;...;n-1\})$, дерево $\mathrm{WB}_{\mathbf{1},2^{t+1}-1,L_{\mathbf{1}}}[\mathrm{RM}_{t+1}]$, каждая метка $\mathbf{p}$ которого на уровне $L_{\mathbf{1}}$ имеет вид (20), где $s=n-(t+1)+1$
Результат: дерево $\mathrm{WB}_{\mathbf{1},2^t-1,L_{\mathbf{1}}+1}[\mathrm{RM}_t]$
$\mathrm{WB}_{\mathbf{1},2^t-1,L_{\mathbf{1}}+1}[\mathrm{RM}_t] := \mathrm{WB}_{\mathbf{1},2^{t+1}-1,L_{\mathbf{1}}}[\mathrm{RM}_{t+1}]$;
для каждой *метки* $\mathbf{p}$ *на уровне* $L_{\mathbf{1}}$ *в дереве* $\mathrm{WB}_{\mathbf{1},2^t-1,L_{\mathbf{1}}+1}[\mathrm{RM}_t]$ **выполнять**
на уровень $L_{\mathbf{1}}+1$ добавить 2^t-1 вершину и соединить их с вершиной, имеющей метку $\mathbf{p}$;
добавленным вершинам однозначно присвоить метки

$$\mathbf{q}_h := \mathbf{p}(\delta_h-\mathbf{1}),\ h\in \mathrm{MakeGroup}(\mathrm{MakeGenSystem}(\mathbf{p})),\ h\neq\hat{1}; \tag{21}$$

конец цикла
возвратить $\mathrm{WB}_{\mathbf{1},2^t-1,L_{\mathbf{1}}+1}[\mathrm{RM}_t]$

Алгоритм 8: MakeRMWBTree

Доказательство. Для доказательства теоремы достаточно показать, во-первых, что в дереве $\mathrm{WB}_{\mathbf{1},2^t-1,L_{\mathbf{1}}+1}[\mathrm{RM}_t]$ на один уровень больше, чем в $\mathrm{WB}_{\mathbf{1},2^{t+1}-1,L_{\mathbf{1}}}[\mathrm{RM}_{t+1}]$, во-вторых, что листья дерева $\mathrm{WB}_{\mathbf{1},2^t-1,L_{\mathbf{1}}+1}[\mathrm{RM}_t]$ принадлежат коду $\mathrm{RM}_t^{\perp}=\mathrm{RM}_{n-t+1}$ и, в-третьих, что в алгоритме для каждой вершины на уровне $L_{\mathbf{1}}$ строится M-ортогональное множество.

Первый факт очевиден. Вновь добавленные узлы на уровне $L_{\mathbf{1}}+1$ имеют метки, представленные в виде произведения $n-(t+1)+1+1=n-t+1$ различных функций вида $(\delta_h-\mathbf{1})$, $h\in\Delta_{2,n}$. В [8], с. 73, доказано, что код RM_t содержит все элементы вида $(\delta_{h_1}-\mathbf{1})...(\delta_{h_s}-\mathbf{1})$, где $s\geq t$ и $h_1,...,h_s\in\Delta_{2,n}$. Таким образом, метки добавленных узлов принадлежат RM_{n-t+1}. Покажем, что строятся M-ортогональные множества для вершин на уровне $L_{\mathbf{1}}$. Пусть $\mathbf{p}$ имеет вид (20), $h\in\mathcal{M}_{\mathbf{p}}\backslash\mathcal{S}_{\mathbf{p}}$. Так как поле $\mathbb{F}_2$, то $-1=1$. Следовательно,

$$\mathrm{supp}_{B^{\Delta_{2,n}}}(\mathbf{p}(\delta_h-\mathbf{1}))=\mathrm{supp}_{B^{\Delta_{2,n}}}\mathbf{p}\delta_h+\mathbf{p}.$$

Учитывая, что $\mathcal{M}_{\mathbf{p}}$ – система порождающих группы $\Delta_{2,n}$, то

$$\mathrm{supp}_{B^{\Delta_{2,n}}}(\mathbf{p}\delta_h)\cap\mathrm{supp}_{B^{\Delta_{2,n}}}(\mathbf{p})=\emptyset. \tag{22}$$

Отметим, что из свойств системы порождающих следует, что равенство (22) выполняется и в случае, когда $h\in<\mathcal{M}_{\mathbf{p}}\backslash\mathcal{S}_{\mathbf{p}}>$. Поэтому для различных $h,g\in<\mathcal{M}_{\mathbf{p}}\backslash\mathcal{S}_{\mathbf{p}}>$ получим:

$$\mathrm{supp}_{B^{\Delta_{2,n}}}(\mathbf{p}(\delta_h-\mathbf{1}))\cap\mathrm{supp}_{B^{\Delta_{2,n}}}(\mathbf{p}(\delta_g-\mathbf{1}))=\mathrm{supp}_{B^{\Delta_{2,n}}}(\mathbf{p}).$$

Из (1) следует, что для каждой вершины на уровне $L_{\mathbf{1}}$ строится M-ортогональное множество. □

Так как RM_t — групповой MLD-код, то множество всех декодирующих деревьев $\mathcal{WB}(\mathrm{RM}_t)$ этого кода может быть построено по дереву $\mathrm{WB}_{\mathbf{1},2^t-1,L_{\mathbf{1}}+1}[\mathrm{RM}_t]$ с помощью алгоритма MakeAllTrees:

$$\mathcal{WB}(\mathrm{RM}_t)=\mathrm{MakeAllTrees}(\Delta_{2,n},\mathrm{WB}_{\mathbf{1},2^t-1,L_{\mathbf{1}}+1}[\mathrm{RM}_t]).$$

Рассмотрим групповой код $\mathbb{F}_2\mathrm{Aff}(\mathbb{F}_{2^n})\otimes_{\mathbb{F}_2\mathcal{H}}\varsigma(\mathrm{RM}_t)$. Предъявим конструкцию правой трансверсали группы $\mathrm{Aff}(\mathbb{F}_{2^n})$ по подгруппе $\mathcal{H}$ вида (18). Пусть $\{\mathcal{H}g\}$ – множество классов смежности фактор–группы $\mathrm{Aff}(\mathbb{F}_{2^n})/\mathcal{H}$. Рассмотрим подгруппу

$$Y = \{m(\alpha) = \begin{pmatrix} \alpha & 0 \\ 0 & 1 \end{pmatrix} : \alpha \neq 0\} \tag{23}$$

группы $\mathrm{Aff}(\mathbb{F}_{2^n})$. Так как группа Y естественно изоморфна фактор–группе $\mathrm{Aff}(\mathbb{F}_{2^n})/\mathcal{H}$, то

$$\bigcup_{y\in Y}\mathcal{H}y = \mathrm{Aff}(\mathbb{F}_{2^n}).$$

Отметим, что группа $\mathrm{Aff}(\mathbb{F}_{2^n})$ не изоморфна прямой сумме $\mathcal{H}\oplus Y$. Построенная подгруппа Y является правой трансверсалью группы $\mathrm{Aff}(\mathbb{F}_{2^n})$ по подгруппе $\mathcal{H}$. В силу (14):

$$\mathbb{F}_2\mathrm{Aff}(\mathbb{F}_{2^n}) = \bigoplus_{y\in Y}(\mathbb{F}_2\mathcal{H})y. \tag{24}$$

Таким образом, для декодирования кода $\mathbb{F}_2\mathrm{Aff}(\mathbb{F}_{2^n})\otimes_{\mathbb{F}_2\mathcal{H}}\varsigma(\mathrm{RM}_t)$ может быть применим алгоритм Decoder4.

Список литературы / References

[1] Федоренко С. В., *Методы быстрого декодирования линейных кодов*, ГУАП, СПб., 2008, 199 с.; [Fedorenko S. V., *Metody bystrogo dekodirovaniya lineynykh kodov*, GUAP, SPb., 2008, 199 pp., (in Russian).]

[2] Massey J. L., *Threshold Decoding*, MIT Press, Cambridge, 1963.

[3] Clark G. C. Jr., Cain J. B., *Error-Correction Coding for Digital Communications*, Springer, 1981., 432 pp.

[4] Сидельников В. М., "Открытое шифрование на основе двоичных кодов Рида–Маллера", *Дискретная математика*, **6**:2 (1994), 3–20; [English transl.: Sidelnikov V. M., "Open coding based on Reed–Muller binary codes", *Diskr. Mat.*, **6:2** (1994), 3–20]

[5] Берман С. Д., "К теории групповых кодов", *Кибернетика*, **3**:17 (1967), 31–39; [English transl.: Berman S. D., "On the theory of group codes", *Kibernetika*, **3**:1 (1967), 31–39]

[6] Грушко И. И., "О мажоритарном декодировании обобщенных кодов Рида–Маллера", *Пробл. передачи информ.*, **26**:3 (1990), 12–20; [English transl.: Grushko I. I., "Majority-logic decoding of generalized Reed-Muller codes", *Probl. Inf. Transm.*, **26**:3 (1990), 189–196]

[7] Логачев О. А., Ященко В. В., "Коды типа Рида–Маллера на конечной абелевой группе", *Пробл. передачи информ.*, **34**:2 (1998), 32–46; [English transl.: Logachev O. A., Yashchenko V. V., "Codes of the Reed-Muller type on a finite abelian group", *Probl. Inf. Transm.*, **34**:2 (1998), 121–133]

[8] Циммерман К.-Х., *Методы теории модулярных представлений в алгебраической теории кодирования*, МЦНМО, М., 2011, 246 с.; (Tsimmerman K.-Kh., *Metody teorii modulyarnykh predstavleniy v algebraicheskoy teorii kodirovaniya*, MTsNMO, M., 2011, 246 pp., [in Russian].)

[9] Деундяк В. М., Косолапов Ю. В., "О стойкости кодового зашумления к статистическому анализу наблюдаемых данных многократного повторения", *Моделирование и анализ информационных систем*, **19**:4 (2012), 110—127; (Deundyak V. M., Kosolapov Yu. V., "On the Firmness Code Noising to the Statistical Analysis of the Observable Data of Repeated Repetition", *Modeling and Analysis of Information Systems*, **19**:4 (2012), 110—127, [in Russian].)

[10] Косолапов Ю. В., "Коды для обобщенной модели канала с подслушиванием", *Проблемы передачи информации*, **51**:1 (2015), 23—28; [English transl.: Kosolapov Yu. V., "Codes for a Generalized Wire-Tap Channel Model", *Problems of Information Transmission*, **51**:1 (2015), 20—24]

[11] Деундяк В. М., Мкртичян В. В., "Исследование границ применения схемы защиты информации, основанной на РС-кодах", *Дискретный анализ и исследование операций*, **18**:3 (2011), 21—38; (Deundyak V. M., Mkrtichyan V. V., "Issledovanie granits primeneniya skhemy zashchity informatsii, osnovannoy na RS-kodakh", *Diskretnyy analiz i issledovanie operatsiy*, **18**:3 (2011), 21—38, [in Russian].)

[12] Сидельников В. М., *Теория кодирования*, ФИЗМАТЛИТ, 2011, 323 с.; (Sidel'nikov V. M., *Teoriya kodirovaniya*, FIZMATLIT, 2011, 323 pp., [in Russian].)

[13] Деундяк В. М., Дружинина М. А., Косолапов Ю. В., "Модификация криптоаналитического алгоритма Сидельникова–Шестакова для обобщенных кодов Рида–Соломона и ее программная реализация", *Известия высших учебных заведений. Северо-Кавказский регион*, Технические науки, 2006, 15–20; (Deundyak V. M., Druzhinina M. A., Kosolapov Yu. V., "Modifikatsiya kriptoanaliticheskogo algoritma Sidel'nikova–Shestakova dlya obobshchennykh kodov Rida–Solomona i ee programmnaya realizatsiya", *Izvestiya vysshikh uchebnykh zavedeniy. Severo-Kavkazskiy region*, Tekhnicheskie nauki, 2006, 15–20, [in Russian].)

[14] Minder L., Shokrollahi A., "Cryptanalysis of the Sidelnikov cryptosystem", *Eurocrypt 2007 of Lecture Notes in Computer Science*, **4515** (2007), 347—360.

[15] Зиновьев В. А., Зяблов В. В., "Коды с неравной защитой информационных символов", *Проблемы передачи информ.*, **15**:3 (1979), 50–60; [English transl.: Zinovev, V. A., Zyablov V. V., "Codes with unequal protection of information symbols", *Probl. Inf. Transm.*, **15**:15 (1979), 197–205]

[16] Curtis C. W., Reiner I., *Representation Theory of Finite Groups and Associative Algebras*, Intersclence Publishers, New York, 1962.

[17] Логачев О. А., Сальников А. Ю., Ященко В. В., *Булевы функции в теории кодирования и криптологии*, МЦМНО, М., 2004, 470 с.; [English transl.: Logachev O. A., Salnikov A. A., Yashchenko V. V., *Boolean Functions in Coding Theory and Cryptography*, AMS. Translations of Mathematical Monographs, 2012, 334 pp.

DOI: 10.18255/1818-1015-2015-4-464-482

Algorithms for Majority Decoding of Group Codes

Deundyak V. M., Kosolapov Y. V.

Received April 29, 2015

We consider a problem of constructive description and justification of the algorithms necessary for a practical implementation of the majority decoder for group codes specified as left ideals of group algebras. In addition to the algorithms needed to implement a classical decoder of J. Massey, it is built a generalization of the classical decoder for codes with unequal protection of characters, which in some cases could be better than the classic one. For use as a classical decoder of J. Massey and its generalization to group codes it was developed an algorithm for constructing decoding trees that lie at the core of these algorithms for majority decoding. Because group codes are defined as left ideals of group algebras, the decoding algorithm for constructing decoding trees allows to build all decoding trees from one tree. On the basis of the generalized decoding algorithm it was developed an algorithm for decoding group codes induced on the subgroup. Application of the developed decoders was illustrated by an example of Reed-Muller-Berman codes and group codes induced by them on a non-Abelian group of affine transformations. In particular, for Reed–Muller–Berman code description and justification of the algorithm for constructing one decoding tree are provided. This three is used for constructing all decoding trees and then it is a built decoder for Reed-Muller-Berman codes and codes induced by them.

Keywords: majority decoder, group algebra, group codes, Reed–Muller–Berman Codes

For citation: Deundyak V. M., Kosolapov Y. V., "Algorithms for Majority Decoding of Group Codes", *Modeling and Analysis of Information Systems*, **22**:4 (2015), 464–482.

On the authors: Deundyak V. M., orcid.org/0000-0001-8258-2419, PhD,
FGNU NII "Specvuzavtomatika",
51 Gazetniy lane, Rostov-on-Don, 344002, Russia, e-mail: vlade@math.rsu.ru,

Kosolapov Y. V., orcid.org/0000-0002-1491-524X, PhD,
South Federal University,
105/42 Bolshaya Sadovaya Str., Rostov-on-Don, 344006, Russia, e-mail: itaim@mail.ru