



Общероссийский математический портал

С. М. Рацеев, А. Д. Лавриненко, Е. А. Степанова, О декодировании алгебраических кодов на основе алгоритма Берлекэмп–Месси, *Ученые записки УлГУ. Серия “Математика и информационные технологии”*, 2021, выпуск 2, 31–42

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 18.97.9.168

25 января 2025 г., 09:29:53





Ссылка на статью:

// Ученые записки УлГУ. Сер. Математика и информационные технологии. УлГУ. Электрон. журн. 2021, № 2, с. 31–42.

Поступила: 26.10.2021

Окончательный вариант: 30.10.2021

© УлГУ

УДК 519.725

О декодировании алгебраических кодов на основе алгоритма Берлекэмпа-Мессе

Рацеев С. М., Лавриненко А. Д., Степанова Е. А.*

[*ratseevsm@mail.ru](mailto:ratseevsm@mail.ru)

УлГУ, Ульяновск, Россия

В обзорной работе приводятся алгоритмы декодирования кодов БЧХ, кодов Рида-Соломона, обобщенных кодов Рида-Соломона, кодов Гошпы на основе алгоритма Берлекэмпа-Мессе для случая ошибок. Также приведены три вариации алгоритма Берлекэмпа-Мессе.

Ключевые слова: алгоритм Берлекэмпа-Мессе, код Рида-Соломона, код БЧХ, код Гошпы, декодирование кода.

Введение

В работе авторов [1] приводилось описание алгоритма Берлекэмпа-Мессе, его модификации для случая минимизации вычисления обратных элементов поля, эквивалентного варианта алгоритма Берлекэмпа-Мессе на основе обобщенного алгоритма Евклида, также приводились алгоритмы декодирования кодов БЧХ, кодов Рида-Соломона, обобщенных кодов Рида-Соломона, кодов Гошпы на основе алгоритма Берлекэмпа-Мессе для случая, когда в канале связи могут происходить ошибки и стирания. С одной стороны, понятно, что во всех представленных алгоритмах в случае только ошибок и отсутствия стираний в этих алгоритмах полагается $s = 0$. С другой стороны, в силу важности перечисленных кодов, совсем нелишне привести алгоритмы декодирования для случая только ошибок в явном виде. Этому и посвящена данная работа. Все необъясняемые ниже понятия можно найти в работах [2, 3].

Актуальность рассматриваемых алгоритмов состоит в том, что они применимы для декодирования кодов Гоппы, которые лежат в основе некоторых перспективных постквантовых криптосистем (см., напр. [3–5]).

Напомним, что алгоритм Берлекэмпа–Месси имеет следующий вид.

Алгоритм 1 (алгоритм Берлекэмпа–Месси).

Вход: последовательность a_1, \dots, a_n над некоторым полем.

Выход: LFSR $(L, f(x))$ минимальной длины L , для которого:

$$-a_j = \sum_{i=1}^L f_i a_{j-i}, \quad j = L + 1, L + 2, \dots, n. \quad (1)$$

1. Определить $r := 0$, $f(x) := 1$, $b(x) := 1$, $L := 0$.

2. Цикл $r := 1, \dots, n$:

2.1. Определить $\Delta := a_r + \sum_{i=1}^L f_i a_{r-i}$.

2.2. Если $\Delta = 0$, то $b(x) := x \cdot b(x)$.

2.3. Если $\Delta \neq 0$:

2.3.1 $buf(x) := f(x) - \Delta \cdot x \cdot b(x)$.

2.3.2. Если $2L < r$:

$$b(x) := \Delta^{-1} \cdot f(x),$$

$$f(x) := buf(x),$$

$$L := r - L.$$

2.3.3. Иначе (т.е. выполнено $2L \geq r$):

$$f(x) := buf(x),$$

$$b(x) := x \cdot b(x).$$

С помощью алгоритма Берлекэмпа–Месси можно решать системы уравнений следующего вида:

$$\begin{pmatrix} a_n & a_{n-1} & \dots & a_2 & a_1 \\ a_{n+1} & a_n & \dots & a_3 & a_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{2n-1} & a_{2n-2} & \dots & a_{n+1} & a_n \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ \dots \\ f_n \end{pmatrix} = \begin{pmatrix} -a_{n+1} \\ -a_{n+2} \\ \dots \\ -a_{2n} \end{pmatrix}. \quad (2)$$

Пример 1. Решим систему линейных алгебраических уравнений над полем $GF(5)$:

$$\begin{pmatrix} 3 & 1 & 2 \\ 3 & 3 & 1 \\ 1 & 3 & 3 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} = \begin{pmatrix} -3 \\ -1 \\ -4 \end{pmatrix}.$$

Алгоритм решения системы приведем в виде таблицы на основе алгоритма 1:

r	Δ	$f(x)$	$b(x)$	L
0		1	1	0
1	2	$1 + 3x$	3	1
2	2	$1 + 2x$	$3x$	1
3	0	$1 + 2x$	$3x^2$	1
4	4	$1 + 2x + 3x^3$	$4 + 3x$	3
5	0	$1 + 2x + 3x^3$	$4x + 3x^2$	3
6	0	$1 + 2x + 3x^3$	$4x^2 + 3x^3$	3

Таким образом, $f(x) = 1 + 2x + 3x^3$. Так как $L = 3$, то $f_1 = 2$, $f_2 = 0$, $f_3 = 3$.

Заметим, что в алгоритме 1 можно минимизировать число вычислений обратных элементов в поле F , т.е. вычислений вида Δ^{-1} . Отразим это в следующем алгоритме.

Алгоритм 2 (алгоритм Берлекэмпа-Месси).

Вход: последовательность a_1, \dots, a_n над некоторым полем.

Выход: LFSR $(L, f(x))$ минимальной длины L , для которого:

$$-a_j = \sum_{i=1}^L f_i a_{j-i}, \quad j = L + 1, L + 2, \dots, n.$$

1. Определить $r := 0$, $f(x) := 1$, $b(x) := 1$, $L := 0$, $\delta := 1$.

2. Цикл $r := 1, \dots, n$:

2.1. Определить $\Delta := \sum_{i=0}^L f_i a_{r-i}$.

2.2. Если $\Delta = 0$, то $b(x) := x \cdot b(x)$.

2.3. Если $\Delta \neq 0$:

2.3.1 $buf(x) := \delta \cdot f(x) - \Delta \cdot x \cdot b(x)$.

2.3.2. Если $2L < r$:

$$b(x) := f(x),$$

$$f(x) := buf(x),$$

$$L := r - L,$$

$$\delta := \Delta;$$

2.3.3. Иначе (т.е. выполнено $2L \geq r$):

$$f(x) := buf(x),$$

$$b(x) := x \cdot b(x),$$

3. $f(x) := f_0^{-1} \cdot f(x)$.

Пример 2. Решим систему из примера 1 с помощью алгоритма 2:

r	Δ	δ	$f(x)$	$b(x)$	L
0		1	1	1	0
1	2	2	$1 + 3x$	1	1
2	2	2	$2 + 4x$	x	1
3	0	2	$2 + 4x$	x^2	1
4	3	2	$4 + 3x + 2x^3$	$2 + 4x$	3
5	0	2	$4 + 3x + 2x^3$	$2x + 4x^2$	3
6	0	2	$4 + 3x + 2x^3$	$2x^2 + 4x^3$	3

Поэтому $f(x) = 4^{-1}(4 + 3x + 2x^3) = 1 + 2x + 3x^3$.

В работах [1, 6, 7] показано, что алгоритм Берлекэмпа-Месси при определенном ограничении можно заменить на эквивалентный алгоритм на основе обобщенного алгоритма Евклида (причем в работе [1] приведено иное доказательство данной эквивалентности).

Алгоритм 3 (нахождение решения системы (2) с помощью обобщенного алгоритма Евклида).

Вход: последовательность a_1, a_2, \dots, a_{2n} над некоторым полем F , для которой система (2) имеет решение.

Выход: многочлен $f(x)$ степени $\leq n$, для которого $f(0) = 1$ и

$$-a_j = \sum_{i=1}^n f_i a_{j-i}, \quad j = n+1, n+2, \dots, 2n.$$

1) Определить $r_{-1}(x) = x^{2n}$, $r_0(x) = \sum_{i=1}^{2n} a_i x^{i-1}$, $v_{-1}(x) = 0$, $v_0(x) = 1$.

2) Производится последовательность вычислений обобщенного алгоритма Евклида:

$$\begin{aligned} r_{i-2}(x) &= r_{i-1}(x)q_{i-1}(x) + r_i(x), \\ v_i(x) &= v_{i-2}(x) - v_{i-1}(x)q_{i-1}(x), \\ i &= 1, 2, \dots, \end{aligned}$$

до тех пор, пока для некоторого $r_j(x)$ не будет выполнено условие:

$$\deg r_{j-1}(x) \geq n, \quad \deg r_j(x) \leq n-1.$$

3) Определить $f(x) = \lambda v_j(x)$, где константа $\lambda \in F$ задается так, чтобы удовлетворялось условие $f(0) = 1$.

Теорема 1 ([1]). Пусть для последовательности a_1, a_2, \dots, a_{2n} над некоторым полем F система (2) имеет решение. Пусть $(L, f(x))$ — результат работы алгоритма 1, которому на вход подается данная последовательность, $\tilde{f}(x)$ — результат работы алгоритма 3. Тогда $f(x) = \tilde{f}(x)$.

Пример 3. Решим систему из примера 1 с помощью алгоритма 3. В этом случае $n = 3$. Определяем:

$$r_{-1}(x) = x^6, \quad r_0(x) = 2 + x + 3x^2 + 3x^3 + x^4 + 4x^5, \quad v_{-1}(x) = 0, \quad v_0(x) = 1.$$

Производим последовательность вычислений обобщенного алгоритма Евклида:

$$\begin{aligned} r_{-1}(x) &= r_0(x)q_0(x) + r_1(x), \\ q_0(x) &= 4 + 4x, \\ r_1(x) &= 2 + 3x + 4x^2 + x^3 + 4x^4, \\ v_1(x) &= v_{-1}(x) - v_0(x)q_0(x) = 1 + x, \\ r_0(x) &= r_1(x)q_1(x) + r_2(x), \\ q_1(x) &= x, \\ r_2(x) &= 2 + 4x + 4x^3, \\ v_2(x) &= v_0(x) - v_1(x)q_1(x) = 1 + 4x + 4x^2, \\ r_1(x) &= r_2(x)q_2(x) + r_3(x), \\ q_2(x) &= 4 + x, \\ r_3(x) &= 4, \\ v_3(x) &= v_1(x) - v_2(x)q_2(x) = 2 + 4x + x^3. \end{aligned}$$

После третьего шага останавливаемся, так как $\deg r_2(x) = 3 \geq n$, $\deg r_3(x) = 0 \leq n - 1$. Поэтому $f(x) = \lambda v_3(x)$. При $\lambda = 3$ получаем $\lambda v_3(0) = 1$, поэтому $f(x) = 1 + 2x + 3x^3$.

1. Декодирование кодов БЧХ и кодов Рида-Соломона на основе алгоритма Берлекэмпа-Мессе

Пусть $\alpha = (\alpha_0, \alpha_1, \dots, \alpha_{n-1})$, где α_i — различные элементы конечного поля $F = GF(q)$, $y = (y_0, y_1, \dots, y_{n-1})$ — ненулевые (не обязательно различные) элементы из F . Тогда обобщенный код Рида-Соломона (ОРС), обозначаемый $GRS_k(\alpha, y)$, состоит из всех кодовых векторов вида:

$$u = (y_0b(\alpha_0), y_1b(\alpha_1), \dots, y_{n-1}b(\alpha_{n-1})), \quad (3)$$

где $b(x)$ — информационные многочлены над полем F степени не выше $k - 1$. Кодовое расстояние кода $GRS_k(\alpha, y)$ равно $d = n - k + 1$. Если $n = q - 1$, вектор y состоит из единиц и $\alpha_i = \alpha^i$, $i = 0, 1, \dots, n - 1$, где α — примитивный элемент поля F , то в этом случае получаем код Рида-Соломона.

Пусть A — $[n, k, d - n - k + 1]$ -код РС над полем $GF(q)$, d — кодовое расстояние, $u \in A$ — переданный вектор. Пусть v — полученный на приемной стороне вектор (после отправки u), в котором могут быть ошибки. Пусть t — максимальное число возможных ошибок в векторе v , $d \geq 2t + 1$, $t = \lfloor (d - 1)/2 \rfloor$, m — реальное число ошибок, $m \leq t$. Пусть $X_1 = \alpha^{i_1}, \dots, X_m = \alpha^{i_m}$ — неизвестные локаторы ошибок, $Y_1 = e_{i_1}, \dots, Y_m = e_{i_m}$ — значения ошибок в векторе v . Найдем компоненты синдромного вектора:

$$S_j = v(\alpha^{j+1}) = Y_1 X_1^{j+1} + \dots + Y_m X_m^{j+1}, \quad j = 0, 1, \dots, 2t - 1.$$

Искомый многочлен локаторов ошибок имеет вид:

$$\begin{aligned}\sigma(x) &= (1 - X_1x)(1 - X_2x)\dots(1 - X_mx) = \\ &= \sigma_0 + \sigma_1x + \sigma_2x^2 + \dots + \sigma_mx^m.\end{aligned}$$

Подставив в $\sigma(x)$ вместо x значение X_i^{-1} , получим 0:

$$\sigma(X_i^{-1}) = 1 + \sigma_1X_i^{-1} + \sigma_2X_i^{-2} + \dots + \sigma_mX_i^{-m} = 0.$$

Правую и левую части последнего равенства умножим на $Y_iX_i^{j+m}$:

$$Y_i(X_i^{j+m} + \sigma_1X_i^{j+m-1} + \sigma_2X_i^{j+m-2} + \dots + \sigma_mX_i^j) = 0.$$

При фиксированном j просуммируем все последние тождества по $i = 1, 2, \dots, m$:

$$\begin{aligned}0 &= \sum_{i=1}^m Y_i(X_i^{j+m} + \sigma_1X_i^{j+m-1} + \sigma_2X_i^{j+m-2} + \dots + \sigma_mX_i^j) = \\ &= \sum_{i=1}^m Y_iX_i^{j+m} + \sigma_1 \sum_{i=1}^m Y_iX_i^{j+m-1} + \sigma_2 \sum_{i=1}^m Y_iX_i^{j+m-2} + \dots + \sigma_m \sum_{i=1}^m Y_iX_i^j.\end{aligned}$$

Получаем:

$$S_{j+m} + \sigma_1S_{j+m-1} + \sigma_2S_{j+m-2} + \dots + \sigma_mS_j = 0, \quad j = 0, 1, \dots, 2t - m - 1.$$

Таких уравнений ровно $2t - m$ и они составляют систему уравнений:

$$\begin{pmatrix} S_{m-1} & S_{m-2} & \dots & S_1 & S_0 \\ S_m & S_{m-1} & \dots & S_2 & S_1 \\ \dots & \dots & \dots & \dots & \dots \\ S_{2t-2} & S_{2t-3} & \dots & S_{2t-m} & S_{2t-m-1} \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \dots \\ \sigma_m \end{pmatrix} = \begin{pmatrix} -S_m \\ -S_m \\ \dots \\ -S_{2t-1} \end{pmatrix}. \quad (4)$$

Будем искать решение полученной системы с помощью алгоритма Берлекэмпа-Мессии.

Теорема 2. Пусть $d \geq 2t + 1$, $m \leq t$. Если на вход алгоритма 1 подать последовательность $S_0, S_1, \dots, S_{2t-1}$, то на выходе алгоритма будет верное значение многочлена локаторов ошибок $\sigma(x)$.

Доказательство следует из теоремы 5 работы [1].

Учитывая теорему 2, получаем следующий алгоритм декодирования кодов РС (кодов БЧХ).

Алгоритм 4 (декодирование кода РС на основе алгоритма Берлекэмпа-Мессии).

Вход: полученный вектор v .

Выход: исходный кодовый вектор u , если произошло не более $\lfloor (d-1)/2 \rfloor$ ошибок.

- 1) Определяется $t = \lceil (d-1)/2 \rceil$. Находятся компоненты $S_0, S_1, \dots, S_{2t-1}$ синдромного вектора: $S_i = v(\alpha^{i+1})$, $i = 0, 1, \dots, 2t-1$. Если синдромный вектор нулевой, то алгоритм завершается и возвращается $u = v$.
- 2) Для последовательности $S_0, S_1, \dots, S_{2t-1}$ с помощью алгоритма 1 находится многочлен локаторов ошибок $\sigma(x)$. Пусть $l = \deg \sigma(x)$.
- 3) Отыскиваются l корней многочлена $\sigma(x)$.
- 4) Находятся Y_1, \dots, Y_l , например, с помощью формул Форни:

$$Y_i = \frac{X_i^{-1} \omega(X_i^{-1})}{\prod_{\substack{1 \leq j \leq l, \\ j \neq i}} (1 - X_j X_i^{-1})}, \quad i = 1, 2, \dots, l,$$

где $\omega(x) \equiv \sigma(x)S(x) \pmod{x^{2t}}$. Наконец, у вектора v из символа с номером i_j , $X_j = \alpha^{i_j}$, вычитается значение Y_j , $j = 1, \dots, l$. Тем самым получается вектор u .

Пример 4. Рассмотрим расширение поля $GF(2) \subset GF(2^4)$. Пусть поле $GF(2^4)$ строится на основе примитивного многочлена $p(x) = x^4 + x + 1$, α — примитивный элемент поля $GF(2^4)$:

$\alpha^0 = 1$		$= 1000,$	$\alpha^1 = \alpha$		$= 0100,$
$\alpha^2 = \alpha^2$		$= 0010,$	$\alpha^3 = \alpha^3$		$= 0001,$
$\alpha^4 = 1 + \alpha$		$= 1100,$	$\alpha^5 = \alpha + \alpha^2$		$= 0110,$
$\alpha^6 = \alpha^2 + \alpha^3$		$= 0011,$	$\alpha^7 = 1 + \alpha + \alpha^3$		$= 1101,$
$\alpha^8 = 1 + \alpha^2$		$= 1010,$	$\alpha^9 = \alpha + \alpha^3$		$= 0101,$
$\alpha^{10} = 1 + \alpha + \alpha^2$		$= 1110,$	$\alpha^{11} = \alpha + \alpha^2 + \alpha^3$		$= 0111,$
$\alpha^{12} = 1 + \alpha + \alpha^2 + \alpha^3$		$= 1111,$	$\alpha^{13} = 1 + \alpha^2 + \alpha^3$		$= 1011,$
$\alpha^{14} = 1 + \alpha^3$		$= 1001,$	$\alpha^{15} = 1$		$= 1000.$

Рассмотрим код Рида-Соломона с параметрами $n = 15$, $k = 7$, $d = 9$.

Пусть на приемном конце получен вектор:

$$v = (\alpha^{10}, \alpha^3, \alpha^2, \alpha^{12}, \alpha^3, \alpha^{12}, \alpha^{14}, \alpha^4, \alpha^{11}, \alpha^4, \alpha^5, \alpha^3, \alpha^5, \alpha^3, \alpha),$$

в котором не более четырех ошибок. Применим алгоритм декодирования 4.

1. Полагаем $t = \lceil (d-1)/2 \rceil = 4$. Вычислим компоненты синдрома для вектора v :

$$S_0 = v(\alpha) = 0, \quad S_1 = v(\alpha^2) = \alpha^7, \quad S_2 = v(\alpha^3) = \alpha^{13},$$

$$S_3 = v(\alpha^4) = \alpha^8, \quad S_4 = v(\alpha^5) = \alpha^{10}, \quad S_5 = v(\alpha^6) = \alpha^5,$$

$$S_6 = v(\alpha^7) = 0, \quad S_7 = v(\alpha^8) = \alpha^6.$$

2. На вход алгоритма 1 подаем последовательность $S_0 = 0, S_1 = \alpha^7, S_2 = \alpha^{13}, S_3 = \alpha^8, S_4 = \alpha^{10}, S_5 = \alpha^5, S_6 = 0, S_7 = \alpha^6$. На выходе получаем многочлен $\sigma(x) = 1 + \alpha^4 x + \alpha^6 x^2 + x^3 + x^4$.

3. Корнями многочлена локаторов ошибок $\sigma(x)$ являются $x_1 = \alpha^{13}$, $x_2 = \alpha^{10}$, $x_3 = \alpha^4$, $x_4 = \alpha^3$, поэтому $X_1 = \alpha^2$, $X_2 = \alpha^5$, $X_3 = \alpha^{11}$, $X_4 = \alpha^{12}$.

4. После того, как все локаторы ошибок известны, можно воспользоваться формулой Форни для кодов РС при

$$\omega(x) \equiv \sigma(x)S(x) \equiv \alpha^7x + \alpha^4x^2 + \alpha^6x^3 \pmod{x^8}.$$

Находим значения ошибок: $Y_1 = \alpha^7$, $Y_2 = \alpha^8$, $Y_3 = \alpha^9$, $Y_4 = \alpha^3$. Таким образом:

$$e = (0, 0, \alpha^7, 0, 0, \alpha^8, 0, 0, 0, 0, 0, \alpha^9, \alpha^3, 0, 0),$$

$$u = v - e = (\alpha^{10}, \alpha^3, \alpha^{12}, \alpha^{12}, \alpha^3, \alpha^9, \alpha^{14}, \alpha^4, \alpha^{11}, \alpha^4, \alpha^5, \alpha, \alpha^{11}, \alpha^3, \alpha).$$

2. Декодирование обобщенных кодов Рида-Соломона на основе алгоритма Берлекэмпа-Месси

Напомним, что, в отличие от кодов РС, в обобщенных кодах РС одна из компонент вектора α может быть нулевой, что нужно учитывать в алгоритмах декодирования.

Пусть, как и ранее, $t = [(d-1)/2]$, m — истинное значение числа ошибок в принятом векторе, $m \leq t$, e_{i_1}, \dots, e_{i_m} — значения ошибок на соответствующих позициях i_1, \dots, i_m в принятом векторе. Вычисляя синдромный вектор, получаем:

$$\begin{aligned} S &= vH^T = eH^T = (\dots, e_{i_1}, \dots, e_{i_m}, \dots) \times \\ &\times \left(\left(\begin{array}{cccc} 1 & 1 & \dots & 1 \\ \alpha_0 & \alpha_1 & \dots & \alpha_{n-1} \\ \dots & \dots & \dots & \dots \\ \alpha_0^{r-1} & \alpha_1^{r-1} & \dots & \alpha_{n-1}^{r-1} \end{array} \right) \left(\begin{array}{cccc} w_0 & 0 & \dots & 0 \\ 0 & w_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & w_{n-1} \end{array} \right) \right)^T = \\ &= \left(\begin{array}{c} e_{i_1}w_{i_1} + \dots + e_{i_m}w_{i_m} \\ e_{i_1}w_{i_1}\alpha_{i_1} + \dots + e_{i_m}w_{i_m}\alpha_{i_m} \\ \dots \\ e_{i_1}w_{i_1}\alpha_{i_1}^{r-1} + \dots + e_{i_m}w_{i_m}\alpha_{i_m}^{r-1} \end{array} \right)^T. \end{aligned}$$

Пусть $X_1 = \alpha_{i_1}$, $X_2 = \alpha_{i_2}, \dots, X_m = \alpha_{i_m}$ — локаторы ошибок, $Y_1 = e_{i_1}, Y_2 = e_{i_2}, \dots, Y_m = e_{i_m}$ — значения ошибок. Обозначим $Z_j = Y_j w_{i_j}$, $j = 1, \dots, m$. Тогда:

$$S_j = Z_1 X_1^j + \dots + Z_m X_m^j, \quad j = 0, 1, \dots, 2t - 1.$$

Как и ранее, подставив в $\sigma(x)$ вместо x значение X_i^{-1} , получим 0:

$$\sigma(X_i^{-1}) = 1 + \sigma_1 X_i^{-1} + \sigma_2 X_i^{-2} + \dots + \sigma_m X_i^{-m} = 0.$$

Правую и левую части последнего равенства умножим на $Z_i X_i^{j+m}$:

$$Z_i (X_i^{j+m} + \sigma_1 X_i^{j+m-1} + \sigma_2 X_i^{j+m-2} + \dots + \sigma_m X_i^j) = 0.$$

При фиксированном j просуммируем все последние тождества по $i = 1, 2, \dots, m$:

$$\begin{aligned} 0 &= \sum_{i=1}^m Z_i (X_i^{j+m} + \sigma_1 X_i^{j+m-1} + \sigma_2 X_i^{j+m-2} + \dots + \sigma_m X_i^j) = \\ &= \sum_{i=1}^m Z_i X_i^{j+m} + \sigma_1 \sum_{i=1}^m Z_i X_i^{j+m-1} + \sigma_2 \sum_{i=1}^m Z_i X_i^{j+m-2} + \dots + \sigma_m \sum_{i=1}^m Z_i X_i^j. \end{aligned}$$

Получаем:

$$S_{j+m} + \sigma_1 S_{j+m-1} + \sigma_2 S_{j+m-2} + \dots + \sigma_m S_j = 0, \quad j = 0, 1, \dots, 2t - m - 1.$$

Таких уравнений ровно $2t - m$ и они составляют систему уравнений:

$$\begin{pmatrix} S_{m-1} & S_m & \dots & S_1 & S_0 \\ S_m & S_{m-1} & \dots & S_2 & S_1 \\ \dots & \dots & \dots & \dots & \dots \\ S_{2t-2} & S_{2t-3} & \dots & S_{2t-m} & S_{2t-m-1} \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \dots \\ \sigma_m \end{pmatrix} = \begin{pmatrix} -S_m \\ -S_{m+1} \\ \dots \\ -S_{2t-1} \end{pmatrix}. \quad (5)$$

Теорема 3. Пусть $d \geq 2t + 1$, $m \leq t$. Если на вход алгоритма 1 подать последовательность $S_0, S_1, \dots, S_{2t-1}$, то на выходе алгоритма будет верное значение многочлена локаторов ошибок $\sigma(x)$.

Доказательство следует из теоремы 6 работы [1].

Алгоритм 5 (декодирование обобщенного кода РС на основе алгоритма Берлекэмп-Месси).

Вход: принятый вектор v , в котором не более t ошибок.

Выход: исходный кодовый вектор u , если $d \geq 2t + 1$.

- 1) Определяется $t = \lceil (d-1)/2 \rceil$. Находятся первые $2t$ компоненты $S_0, S_1, \dots, S_{2t-1}$ синдромного вектора vH^T . Если они все равны нулю, то полагается, что ошибок нет и процедура окончена.
- 2) С помощью алгоритма 1 на основе последовательности $S_0, S_1, \dots, S_{2t-1}$ находится многочлен локаторов ошибок $\sigma(x)$. Пусть $l = \deg \sigma(x)$.
- 3) Отыскиваются l корней многочлена $\sigma(x)$.
- 4) Находятся Z_1, \dots, Z_l , например, с помощью алгоритма Форни для ОРС кодов:

$$Z_i = \frac{\omega(X_i^{-1})}{\prod_{j \neq i} (1 - X_j X_i^{-1})}, \quad i = 1, \dots, l, \quad (6)$$

где $\omega(x) \equiv \sigma(x)S(x) \pmod{x^{2t}}$. После этого находятся значения ошибок $Y_j = Z_j/w_{i_j}$, $j = 1, \dots, l$. У вектора v из i_j -го символа, $X_j = \alpha_{i_j}$, вычитается значение Y_j , $j = 1, \dots, l$. При этом получается вектор \tilde{u} .

Если $\alpha_i = 0$ для некоторого i и $\deg \sigma(x) < L$, то вычисляется значение Z_0 , равное скалярному произведению вектора \tilde{u} на первую строку матрицы H . Вычисляется значение ошибки $Y_0 = Z_0/w_i$. Осталось в векторе \tilde{u} из i -го символа вычесть Y_0 .

3. Декодирование кодов Гоппы на основе алгоритма Берлекэмпа-Мессе

Определение кода Гоппы опирается на два объекта: многочлен $G(x)$ с коэффициентами из поля $GF(q^m)$, который называется многочленом Гоппы; подмножество $L = \{\alpha_0, \alpha_1, \dots, \alpha_{n-1}\}$ элементов поля $GF(q^m)$ таких, что $G(\alpha_i) \neq 0$ для всех $\alpha_i \in L$. Код Гоппы $\Gamma(L, G)$ состоит из всех векторов $u = (u_0, u_1, \dots, u_{n-1})$ с компонентами из $GF(q)$, для которых:

$$R_u = \sum_{i=0}^{n-1} \frac{u_i}{x - \alpha_i} \equiv 0 \pmod{G(x)}.$$

Если $G(x)$ неприводим, то код $\Gamma(L, G)$ называется неприводимым кодом Гоппы. Хорошо известно, что код $\Gamma(L, G)$ можно задать с помощью обобщенного кода Рида-Соломона.

Пусть код $\Gamma(L, G)$ является двоичным. Если $G(x)$ не имеет кратных корней, то код $\Gamma(L, G)$ называется сепарабельным кодом Гоппы. Пусть $\bar{G}(x)$ — полный квадрат некоторого многочлена над $GF(2^m)$ наименьшей степени, делящийся на $G(x)$. В случае сепарабельного кода $\bar{G}(x) = G^2(x)$. Для минимального расстояния сепарабельного кода $\Gamma(L, G)$ верна оценка $d \geq 2r + 1$ и выполнено равенство $\Gamma(L, G) = \Gamma(L, \bar{G})$. Эти факты позволяют строить сепарабельный код $\Gamma(L, G) = \Gamma(L, \bar{G})$, а некоторые алгоритмы декодирования кодов Гоппы применять относительно кода $GRS_{n-2r}(\alpha, y)$, $r = \deg G(x)$.

Пусть $[n, k, d]$ -код $\Gamma(L, G)$ задается на основе ОРС кода: $\Gamma(L, G) = GRS_{n-r}(L, y) \cap F^n$, $F = GF(q)$, $r = \deg G(x)$, $\tilde{k} = n - r$ — размерность кода $GRS_{n-r}(L, y)$ длины n , \bar{H} — проверочная матрица кода $GRS_{n-r}(L, y)$. Пусть d, \tilde{d} — кодовые расстояния соответственно кодов $\Gamma(L, G)$ и $GRS_{n-r}(L, y)$. Так как $d \geq r + 1$, $\tilde{d} = n - \tilde{k} + 1 = r + 1$, то если в кодовом векторе $u \in \Gamma(L, G)$ произошло t ошибок, причем $r \geq 2t$, то для его декодирования можно применять алгоритмы декодирования для ОРС кодов.

Если же код $\Gamma(L, G)$ двоичный и сепарабельный, то $\Gamma(L, G) = GRS_{n-2r}(L, y) \cap F^n$, $F = GF(2)$, $\tilde{k} = n - 2r$ — размерность кода $GRS_{n-2r}(L, y)$, \bar{H} — проверочная матрица кода $GRS_{n-2r}(L, y)$. Также $d \geq 2r + 1$, $\Gamma(L, G^2) \subseteq GRS_{n-2r}(L, y)$, $\tilde{d} = 2r + 1$, поэтому в этом случае алгоритмы декодирования для ОРС кодов можно применять для декодирования вектора u , в котором t ошибок, причем $r \geq t$.

В работе [8] приводятся различные алгоритмы декодирования кодов Гоппы, причем, желая сократить объем статьи, алгоритмы декодирования на основе алгоритма Берлекэмпа-Мессе и на основе алгоритма Питерсона-Горенштейна-Цирлера были совмещены в один, что

затруднило его восприятие. Поэтому приведем алгоритм декодирования на основе алгоритма Берлекэмпа-Месси в явном виде.

Алгоритм 6 (декодирование кодов Гоппы на основе алгоритма Берлекэмпа-Месси).

Вход: принятый вектор v , в котором не более t ошибок.

Выход: исходный кодовый вектор u , в котором произошло не более t ошибок, если $r \geq 2t$, $r = \deg G(x)$, $u \in \Gamma(L, G) \subseteq GRS_{n-r}(L, y)$ (для двоичного сепарабельного кода $r \geq t$, $u \in \Gamma(L, G) \subseteq GRS_{n-2r}(L, y)$).

- 1) Определяется $t = \lceil r/2 \rceil$ ($t = r$ в случае двоичного сепарабельного кода Гоппы). Находятся первые $2t$ компонент $S_0, S_1, \dots, S_{2t-1}$ синдромного вектора $v\overline{H}^T$. Если они все равны нулю, то полагается, что ошибок нет и процедура окончена.
- 2) С помощью алгоритма 1 на основе последовательности $S_0, S_1, \dots, S_{2t-1}$ находится многочлен локаторов ошибок $\sigma(x)$. Пусть $l = \deg \sigma(x)$.
- 3) Отыскиваются l корней многочлена $\sigma(x)$ последовательной подстановкой в него ненулевых элементов поля $GF(q^m)$.
- 4) Находятся Z_1, \dots, Z_l , например, с помощью алгоритма Форни (6). После этого находят значения ошибок $Y_j = Z_j G(X_j)$, $j = 1, \dots, l$. У вектора v из i_j -го символа, $X_j = \alpha_{i_j}$, вычитается значение Y_j , $j = 1, \dots, l$. При этом получаем вектор \tilde{u} .

Если $\alpha_i = 0$ для некоторого i и $\deg \sigma(x)$ строго меньше длины LFSR (полученного на выходе алгоритма 1), то вычисляется значение Z_0 , равное скалярному произведению вектора \tilde{u} на первую строку матрицы H . Вычисляется значение ошибки $Y_0 = Z_0 G(\alpha_i)$. Осталось в векторе \tilde{u} из i -го символа вычесть Y_0 .

Замечание 1. Если в алгоритмах 4, 5, 6 на втором шаге вместо алгоритма 1 применять алгоритм 3, то получим алгоритмы декодирования на основе алгоритма Сугиямы. При этом очень важно учитывать теорему 1.

Заключение

Во многих алгоритмах синдромного декодирования алгебраических кодов возникают системы линейных уравнений, у которых матрицы имеют такой вид, что все диагонали, параллельные главной, имеют одинаковые элементы. Матрицы такого вида называются матрицами Тёплица. Решения данных систем уравнений более практично искать с помощью алгоритма Берлекэмпа-Месси, который имеет сложность вычислений порядка n^2 .

В работе приведены три вариации алгоритма Берлекэмпа-Месси, а также алгоритмы декодирования некоторых алгебраических кодов на их основе. При этом одним из важнейших рассмотренных алгоритмов является декодирование кодов Гоппы на основе алгоритма Берлекэмпа-Месси, так как именно на основе кодов Гоппы строятся некоторые перспективные постквантовые криптосистемы.

Список литературы

1. Рацеев С.М., Лавриненко А.Д., Степанова Е.А. Об алгоритме Берлекэмп-Месси и его применении в алгоритмах декодирования // *Вестник Самарского университета. Естественнонаучная серия*. 2021, т. 27, № 1, с. 44–61.
2. Блейхут Р. *Теория и практика кодов, контролирующих ошибки*. Пер. с англ. М. : Мир, 1986. 576 с.
3. Рацеев С.М. *Элементы высшей алгебры и теории кодирования : учебное пособие для вузов*. СПб. : Лань, 2022. 656 с.
4. Bernstein D., Chou T., Lange T., Maurich I., Misoczki R., Niederhagen R., Persichetti E., Peters C., Schwabe P., Sendrier N., Szefer J., Wang W. Classic McEliece: conservative code-based cryptography. Project documentation: [Электронный ресурс]. Режим доступа: <https://classic.mceliece.org/nist/mceliece-20190331.pdf>, свободный. Яз. англ. (дата обращения: 24.12.2021).
5. Рацеев С.М. *Математические методы защиты информации : учебное пособие для вузов*. СПб. : Лань, 2022. 544 с.
6. Sugiyama Y. et al. A method for solving key equation for decoding Goppa codes // *Infor. Contr.* 1975, v. 27, p. 87–99.
7. Dornstetter J.L. On the equivalence Between Berlekamp's and Euclid's Algorithm // *IEEE Trans. Inform. Theory*. 1987, v. IT-33, № 3, p. 428–431.
8. Рацеев С.М. Об алгоритмах декодирования кодов Гоппы // *Челяб. физ.-матем. журн.* 2020, т. 5, № 3, с. 327–341.

On decoding algebraic codes based on the Berlekamp-Massey algorithm *Ratseev, S. M.* , Lavrinenko, A. D., Stepanova, E. A.*

*ratseevsm@mail.ru

Ulyanovsk State University, Ulyanovsk, Russia

The paper is devoted to the algorithms for the decoding of BCH codes, Reed-Solomon codes, generalized Reed-Solomon codes, and Goppa codes based on the Berlekamp-Massey algorithm.

Keywords: *Berlekamp-Massey algorithm, Reed-Solomon codes, Goppa codes, code decoding.*