Научная статья
DOI: 10.18287/2541-7525-2021-27-1-44-61

УДК 512.531; 519.7

Дата: поступления статьи: 10.01.2021
после рецензирования: 17.02.2021
принятия статьи: 28.02.2021

С.М. Рацеев
Ульяновский государственный университет,
г. Ульяновск, Российская Федерация
E-mail: ratseevsm@mail.ru. ORCID: https://orcid.org/0000-0003-4995-9418
А.Д. Лавриненко
Ульяновский государственный университет,
г. Ульяновск, Российская Федерация
E-mail: anutalavrinenko@gmail.com. ORCID: https://orcid.org/0000-0003-3652-1097
Е.А. Степанова
Ульяновский государственный университет,
г. Ульяновск, Российская Федерация
E-mail: kate_stepanova@bk.ru. ORCID: https://orcid.org/0000-0003-0276-1615

ОБ АЛГОРИТМЕ БЕРЛЕКЭМПА — МЕССИ И ЕГО ПРИМЕНЕНИИ В АЛГОРИТМАХ ДЕКОДИРОВАНИЯ

АННОТАЦИЯ

В работе содержатся описание алгоритма Берлекэмпа — Месси и его эквивалентный вариант на основе обобщенного алгоритма Евклида. Также приводится оптимизированный алгоритм Берлекэмпа — Месси для случая поля характеристики два. Алгоритм Берлекэмпа — Месси имеет квадратичную сложность и применяется, например, для решения систем линейных уравнений, у которых матрица системы является матрицей Тёплица. В частности, такие системы уравнений появляются в алгоритмах синдромного декодирования кодов БЧХ, кодов Рида — Соломона, обобщенных кодов Рида — Соломона, кодов Гоппы. Приводятся алгоритмы декодирования перечисленных кодов на основе алгоритма Берлекэмпа — Месси.

Ключевые слова: алгоритм Берлекэмпа — Месси; обобщенный алгоритм Евклида; код Рида — Соломона; декодирование кода.

Цитирование. Рацеев С.М., Лавриненко А.Д., Степанова Е.А. Об алгоритме Берлекэмпа — Месси и его применении в алгоритмах декодирования // Вестник Самарского университета. Естественнонаучная серия. 2021. Т. 27, № 1. С. 44–61. DOI: http://doi.org/10.18287/2541-7525-2021-27-1-44-61.

Информация о конфликте интересов: авторы и рецензенты заявляют об отсутствии конфликта интересов.

Сергей Михайлович Рацеев — доктор физико-математических наук, доцент, профессор кафедры информационной безопасности и теории управления, Ульяновский государственный университет, 432017, Российская Федерация, г. Ульяновск, ул. Льва Толстого, 42.

Анна Дмитриевна Лавриненко — студент кафедры информационной безопасности и теории управления, Ульяновский государственный университет, 432017, Российская Федерация, г. Ульяновск, ул. Льва Толстого, 42.

Екатерина Алексеевна Степанова — студент кафедры информационной безопасности и теории управления, Ульяновский государственный университет, 432017, Российская Федерация, г. Ульяновск, ул. Льва Толстого, 42.

Введение

В технике один из наиболее распространенных методов генерации битовых последовательностей реализуется с помощью регистров сдвига с обратной линейной связью (Linear Feedback Shift Register – LFSR). Регистр сдвига с обратной связью (LFSR) — это регистр, который можно рассматривать как множество ячеек памяти, в каждой из которых записан один бит информации. На каждом шаге содержимое нескольких заранее определенных ячеек (отводов) пропускается через функцию обратной связи. Значение функции записывается в самую левую ячейку регистра, сдвигая все остальные его биты на одну позицию вправо. Выходом регистра на текущем шаге является «вытолкнутый» справа бит (рис. 1).

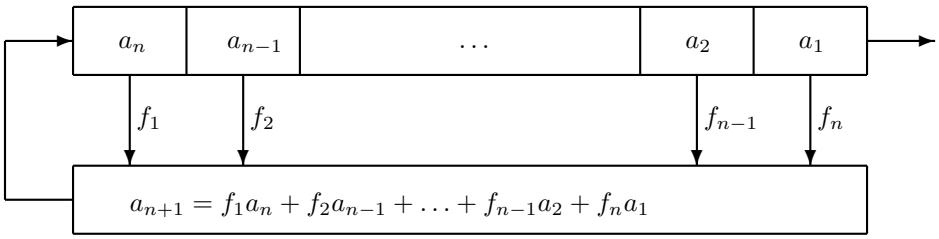

Рис. 1. Регистр сдвига с обратной связью
Fig. 1. Feedback shift register

Рассмотрим систему линейных уравнений над некоторым полем F относительно неизвестных $f_1, \ldots, f_n$:

$$\begin{pmatrix} a_n & a_{n-1} & \ldots & a_2 & a_1 \\ a_{n+1} & a_n & \ldots & a_3 & a_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{2n-1} & a_{2n-2} & \ldots & a_{n+1} & a_n \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ \ldots \\ f_n \end{pmatrix} = \begin{pmatrix} -a_{n+1} \\ -a_{n+2} \\ \ldots \\ -a_{2n} \end{pmatrix}. \tag{1}$$

У матрицы данной системы все диагонали, параллельные главной, имеют одинаковые элементы. Матрица такого вида называется диагонально-постоянной матрицей, или матрицей Тёплица. Решение данной системы уравнений методом Гаусса имеет сложность вычислений порядка n^3. Такие системы практичнее решать с помощью алгоритма Берлекэмпа — Месси, который дает сложность вычислений порядка n^2.

Алгебраические алгоритмы декодирования кодов БЧХ (в частности, кодов Рида — Соломона) можно разделить на два класса: синдромные и бессиндромные. Хорошо известные алгоритмы синдромного декодирования построены на основе алгоритма Питерсона — Горенстейна — Цирлера, алгоритма Берлекэмпа — Месси, алгоритма Сугиямы [1; 2]. К алгоритмам безсиндромного декодирования относится, например, алгоритм Гао [3]. Процесс синдромного декодирования можно разбить на четыре шага: 1) вычисление компонентов синдромного вектора на основе полученного вектора; 2) нахождение многочлена локаторов ошибок $\sigma(x)$, который содержит информацию о позициях ошибок; 3) нахождение корней многочлена $\sigma(x)$, по которым определяются позиции ошибок; 4) нахождение значений ошибок (для недвоичных кодов).

Второй этап декодирования является самым сложным. Многочлен локаторов ошибок $\sigma(x)$ можно найти несколькими способами. Один из первых способов нахождения $\sigma(x)$ — алгоритм Питерсона для двоичных кодов (на основе тождеств Ньютона) и Горенстейна — Цирлера для недвоичных кодов, который сводит данную задачу к решению системы из t линейных уравнений (например, с помощью метода Гаусса), где t — максимальное число ошибок, исправляемое кодом. Сложность такого декодирования пропорциональна t^3. Берлекэмп, используя особенности матрицы коэффициентов системы уравнений, уменьшил сложность нахождения многочлена $\sigma(x)$ до величины порядка t^2. Месси сумел интерпретировать алгоритм Берлекэмпа как алгоритм построения рекуррентного фильтра, тем самым упростив и понимание алгоритма, и его реализацию. В окончательном виде этот алгоритм получил название алгоритма Берлекэмпа — Месси.

Приведенные в данной работе алгоритмы декодирования позволяют исправлять не только ошибки, но и ошибки и стирания одновременно, поэтому они являются обобщением случая только ошибок. Поскольку авторы данной работы не нашли аналогов алгоритмов в других работах, то, вероятно, некоторые из них являются новыми. Все эти алгоритмы снабжены строгими математическими доказательствами.

1. Алгоритм Берлекэмпа — Месси

Лучшим подходом к алгоритму Берлекэмпа — Месси является интерпретация матричного уравнения (1) как описания рекурсивного фильтра. Предположим, что вектор $f = (f_1, \ldots, f_n)$ известен. Последовательность $a_1, a_2, \ldots$ в этом случае называется линейной рекуррентной последовательностью, члены которой вычисляются с помощью уравнения

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

Изначально на элементы последовательности $a_1, a_2, \ldots$ каких-либо ограничений не накладывается. Произвольный LFSR можно задать многочленом $f(x)$ обратных связей $f(x) = 1 + f_1 x + \ldots + f_n x^n$ и длиной L регистра. Заметим, что длина регистра может быть больше степени многочлена $f(x)$, так как самые правые ячейки могут не включаться в обратную связь.

Для построения регистра сдвига нужно определить две величины: длину L регистра и многочлен $f(x)$ обратных связей при условии $\deg f(x) \leqslant L$. Обозначим эту пару через $(L, f(x))$. Данный LFSR должен порождать заданную последовательность $a_1, a_2, \ldots, a_{2n}$ и иметь минимальную длину L.

Основная идея алгоритма Берлекэмпа — Месси состоит в следующем. Алгоритм построения LFSR минимальной длины является рекурсивным. Для каждого $r = 2, 3, \ldots, 2n$ он строит LFSR, порождающий последовательность $a_1, \ldots, a_r$. LFSR минимальной длины, порождающий последовательность $a_1, \ldots, a_r$, обозначим через $(L_r, f^{(r)}(x))$. Такой регистр не обязательно должен определяться однозначно, так как возможно существование нескольких LFSR минимальной длины. К началу r-го шага имеется совокупность LFSR:

$$(L_1, f^{(1)}(x)),\ (L_2, f^{(2)}(x)), \ldots, (L_{r-1}, f^{(r-1)}(x)).$$

Алгоритм Берлекэмпа — Месси вычисляет новый LFSR $(L_r, f^{(r)}(x))$ минимальной длины, генерирующий последовательность $a_1, \ldots, a_r$. Для этого используется самый последний из вычисленных LFSR, в котором, при необходимости, модифицируются длина и весовые множители в отводах. На r-м шаге алгоритма вычисляется элемент с номером r на выходе $(r-1)$-го регистра сдвига:

$$\widetilde{a}_r = -\sum_{i=1}^{L_{r-1}} f_i^{(r-1)} a_{r-i}.$$

Если $\Delta_r = 0$, то полагаем $(L_r, f^{(r)}(x)) = (L_{r-1}, f^{(r-1)}(x))$ и завершаем этим самым r-й шаг алгоритма. Если $\Delta_r \neq 0$, то изменим весовые множители в LFSR по правилу

$$f^{(r)}(x) = f^{(r-1)}(x) + \left(-\frac{\Delta_r}{\Delta_m}\right) x^{r-m} f^{(m-1)}(x),$$

где число m удовлетворяет условию $L_{r-1} = L_{r-2} = \ldots = L_m > L_{m-1}$.

Теорема 1 [4]. Пусть $\left\{(L_r, f^{(r)}(x))\right\}_{r=1}^{n}$ — последовательность LFSR минимальной длины, таких, что $(L_r, f^{(r)}(x))$ генерирует последовательность $a_1, \ldots, a_r$. Если для некоторого r, $2 \leqslant r \leqslant n$, выполнено $f^{(r-1)}(x) \neq f^{(r)}(x)$, то

$$L_r = \max\{L_{r-1}, r - L_{r-1}\}.$$

Замечание 1. Пусть для некоторого r выполнено неравенство $f^{(r-1)}(x) \neq f^{(r)}(x)$. Тогда возможны два случая.

1. $L_r > L_{r-1}$. Тогда из теоремы 1 следует, что $L_r = r - L_{r-1} > L_{r-1}$. Поэтому $2L_{r-1} < r$.
2. $L_r = L_{r-1}$. Тогда $L_{r-1} \geqslant r - L_{r-1}$. Поэтому $2L_{r-1} \geqslant r$.

Учтем данное замечание при описании алгоритма Берлекэмпа — Месси. В следующем алгоритме многочлен $f(x)$ отвечает за многочлен LFSR $(L_r, f^{(r)}(x))$, а многочлен $b(x)$ — за многочлен LFSR $(L_{m-1}, f^{(m-1)}(x))$.

Алгоритм 1. (алгоритм Берлекэмпа — Месси).
Вход: последовательность $a_1, \ldots, a_n$ над некоторым полем F.
Выход: LFSR $(L, f(x))$ минимальной длины L, для которого

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

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

2. Цикл $r := 1, \ldots, n$:
   2.1. Определить $\Delta := a_r + \sum\limits_{i=1}^{L} f_i a_{r-i}$;
   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 $2L \geqslant r$):
         $f(x) := buf(x)$,
         $b(x) := x \cdot b(x)$.

Заметим, что в алгоритме 1 можно минимизировать число вычислений обратных элементов в поле F, т. е. вычислений вида Δ^{-1}. Для этого стоит лишь заметить, что если для некоторого унитарного многочлена $f(x)$ выполнено равенство $a_r + \sum\limits_{i=1}^{L} f_i a_{r-i} = 0$, то для любого $\delta \in F$ выполнено равенство $\sum\limits_{i=0}^{L} \widetilde{f}_i a_{r-i} = 0$, где $\widetilde{f}(x) = \delta \cdot f(x)$. Поэтому алгоритм 1 можно записать в следующем виде.

Алгоритм 1' (алгоритм Берлекэмпа — Месси).
Вход: последовательность $a_1, \ldots, a_n$ над некоторым полем.
Выход: LFSR $(L, f(x))$ минимальной длины L, для которого

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

1. Определить $r := 0$, $f(x) := 1$, $b(x) := 1$, $L := 0$, $\delta := 1$.
2. Цикл $r := 1, \ldots, n$:
   2.1. Определить $\Delta := \sum\limits_{i=0}^{L} f_i a_{r-i}$;
   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 $2L \geqslant r$):
         $f(x) := buf(x)$,
         $b(x) := x \cdot b(x)$;
3. $f(x) := f_0^{-1} \cdot f(x)$.

2. Взаимосвязь алгоритма Берлекэмпа — Месси и обобщенного алгоритма Евклида

Приведенный алгоритм Берлекэмпа — Месси при определенном ограничении можно заменить на эквивалентный алгоритм на основе обобщенного алгоритма Евклида. Этот факт отражен в нескольких работах, например в [5; 6]. В данной работе приведем другое доказательство данной эквивалентности.

Алгоритм 2 (нахождение решения системы (1) с помощью обобщенного алгоритма Евклида).
Вход: последовательность $a_1, a_2, \ldots, a_{2n}$ над некоторым полем F, для которой система (1) имеет решение.
Выход: многочлен $f(x)$ степени $\leqslant n$, для которого $f(0) = 1$ и

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

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

2. Производится последовательность вычислений обобщенного алгоритма Евклида:

$$\begin{array}{l} r_{i-2}(x) = r_{i-1}(x)q_{i-1}(x) + r_i(x), \\ v_i(x) = v_{i-2}(x) - v_{i-1}(x)q_{i-1}(x), \\ i = 1, 2, \ldots, \end{array}$$

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

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

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

Теорема 2. Если для последовательности $a_1, a_2, \ldots, a_{2n}$ над некоторым полем F система (1) имеет решение, то алгоритм 2 всегда возвращает решение системы (1). Более того, если $f_1, \ldots, f_n$ — некоторое решение системы (1), $f(x) = 1 + f_1 x + \ldots + f_n x^n$ — соответствующий многочлен, $\widetilde{f}(x)$ — результат работы алгоритма 2, то $\widetilde{f}(x)$ делит $f(x)$, т. е. из всех решений системы (1) многочлен $\widetilde{f}(x)$ имеет минимальную степень.

Доказательство. Пусть $f_1, \ldots, f_n$ — некоторое решение системы (1). Определим $f_0 = 1$ и обозначим:

$$\begin{array}{l} g_0 = a_1 f_0, \\ g_1 = a_2 f_0 + a_1 f_1, \\ \ldots \\ g_{n-1} = a_n f_0 + a_{n-1} f_1 + \ldots + a_1 f_{n-1}. \end{array} \tag{3}$$

Тогда на основе системы (1) получаем такую систему:

$$\begin{pmatrix} a_1 & 0 & \ldots & 0 & 0 \\ a_2 & a_1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_n & a_{n-1} & \ldots & a_1 & 0 \\ a_{n+1} & a_n & \ldots & a_2 & a_1 \\ a_{n+2} & a_{n+1} & \ldots & a_3 & a_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{2n} & a_{2n-1} & \ldots & a_{n+1} & a_n \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \ldots \\ f_n \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ \ldots \\ g_{n-1} \\ 0 \\ 0 \\ \ldots \\ 0 \end{pmatrix}, \tag{4}$$

в которой система их последних n уравнений полностью совпадает с системой (1). Пусть

$$a(x) = \sum_{i=1}^{2n} a_i x^{i-1}, \quad f(x) = \sum_{i=0}^{n} f_i x^i, \quad g(x) = \sum_{i=0}^{n-1} g_i x^i.$$

Система (4) эквивалентна сравнению

$$a(x)f(x) \equiv g(x) \pmod{x^{2n}} \tag{5}$$

с условиями

$$f(0) = 1, \quad \deg f(x) \leqslant n, \quad \deg g(x) \leqslant n-1. \tag{6}$$

Действительно, пусть выполнено сравнение (5) с условиями (6). Так как выполнено неравенство $\deg g(x) \leqslant n-1$, то коэффициенты многочлена $a(x)f(x)$ при степенях $x^n, x^{n+1}, \ldots, x^{2n-1}$ равны нулю. Это значит, что должны выполняться последние n уравнений системы (4). При этом первые n уравнений системы (4) напрямую следуют из сравнения (5). Обратно, из системы (4) очевидным образом следует сравнение (5) с условиями (6).

Применим к многочленам $r_{-1}(x) = x^{2n},\ r_0(x) = a(x),\ u_{-1}(x) = 1,\ u_0(x) = 0,\ v_{-1}(x) = 0,\ v_0(x) = 1$ обобщенный алгоритм Евклида:

$$\begin{array}{l} r_{i-2}(x) = r_{i-1}(x)q_{i-1}(x) + r_i(x), \\ u_i(x) = u_{i-2}(x) - u_{i-1}(x)q_{i-1}(x), \\ v_i(x) = v_{i-2}(x) - v_{i-1}(x)q_{i-1}(x), \\ i = 1, 2, \ldots \end{array}$$

Тогда на каждом шаге алгоритма будем получать равенства

$$u_i(x)x^{2n} + v_i(x)a(x) = r_i(x),$$

из которых следуют сравнения $v_i(x)a(x) \equiv r_i(x) \pmod{x^{2n}}$. Так как

$$2n = \deg r_{-1}(x) > \deg r_0(x) > \deg r_1(x) > \dots,$$

то для некоторого j будут выполнены неравенства

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

Поэтому индекс j определен однозначно. При этом

$$\deg v_j(x) = \deg x^{2n} - \deg r_{j-1}(x) \leqslant 2n - n = n.$$

Следовательно, мы получили сравнение $v_j(x)a(x) \equiv r_j(x) \pmod{x^{2n}}$, где $\deg r_j(x) \leqslant n-1$, $\deg v_j(x) \leqslant n$.

Покажем, что $v_j(0) \neq 0$. Для многочленов $v_j(x)$ и $r_j(x)$, а также для многочленов $f(x)$, $g(x)$ и некоторого многочлена $t(x)$ выполнены равенства

$$u_j(x)x^{2n} + v_j(x)a(x) = r_j(x), \tag{7}$$

$$t(x)x^{2n} + f(x)a(x) = g(x). \tag{8}$$

Домножив обе части первого равенства на $f(x)$, а второго — на $v_j(x)$, получим

$$\begin{aligned} f(x)u_j(x)x^{2n} + f(x)v_j(x)a(x) &= f(x)r_j(x), \\ v_j(x)t(x)x^{2n} + v_j(x)f(x)a(x) &= v_j(x)g(x). \end{aligned} \tag{9}$$

Из данных равенств следует сравнение

$$f(x)r_j(x) \equiv v_j(x)g(x) \pmod{x^{2n}}.$$

Учитывая степени многочленов в данном сравнении, получаем равенство

$$f(x)r_j(x) = v_j(x)g(x).$$

Поэтому из (9) с учетом последнего равенства следует такое равенство:

$$f(x)u_j(x) = v_j(x)t(x).$$

Из свойства взаимной простоты многочленов $u_j(x)$ и $v_j(x)$ следует, что $v_j(x) \,|\, f(x)$, поэтому для некоторого многочлена $\mu(x)$ выполнено $f(x) = \mu(x)v_j(x)$. Подставим это равенство в (8):

$$t(x)x^{2n} + \mu(x)v_j(x)a(x) = g(x).$$

Теперь домножим равенство (7) на $\mu(x)$:

$$\mu(x)u_j(x)x^{2n} + \mu(x)v_j(x)a(x) = \mu(x)r_j(x).$$

Учитывая степени многочленов $g(x)$, $\mu(x)$ и $r_j(x)$, из последних двух равенств следует равенство $g(x) = \mu(x)r_j(x)$.

Таким образом, $f(x) = \mu(x)v_j(x)$, $g(x) = \mu(x)r_j(x)$. Так как $f(0) = 1$, то выполнено $v_j(0) \neq 0$.

Определим $\widetilde{f}(x) = \lambda v_j(x)$, где константа $\lambda \in F$ задается так, чтобы удовлетворялось условие $\widetilde{f}(0) = 1$. Тогда из $\lambda v_j(x)a(x) \equiv \lambda r_j(x) \pmod{x^{2n}}$, $\deg \lambda v_j(x) \leqslant n$ получаем, что $\widetilde{f}_1, \dots, \widetilde{f}_n$ — решение системы (1). При этом $\widetilde{f}(x) \,|\, f(x)$. □

Предложение 1. Пусть $(L, f(x))$ — результат работы алгоритма 1, которому на вход подавалась последовательность $a_1, \dots, a_n$. Пусть

$$\begin{aligned} g_0 &= a_1 f_0, \\ g_1 &= a_2 f_0 + a_1 f_1, \\ &\dots \\ g_{L-1} &= a_L f_0 + a_{L-1} f_1 + \dots + a_1 f_{L-1}, \end{aligned} \tag{10}$$

где $f_0 = 1$. Тогда $L = \max\{\deg f(x), \deg g(x) + 1\}$, где $g(x) = g_0 + g_1 x + \dots + g_{L-1}x^{L-1}$.

Доказательство. Очевидно, что $L \geqslant \max\{\deg f(x), \deg g(x) + 1\}$. Предположим, что $L > \max\{\deg f(x), \deg g(x) + 1\}$. В этом случае $f_L = 0 = g_{L-1}$. Из равенств (2) и $f_L = 0$ следуют равенства

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

Из равенства $g_{L-1} = 0$, учитывая (10), получаем такое равенство:

$$a_L + a_{L-1}f_1 + \dots + a_1 f_{L-1} = 0.$$

Поэтому будут выполнены такие равенства:

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

Это значит, что алгоритм 1 возвратил LFSR $(L, f(x))$ не минимальной длины, что противоречит тому, что алгоритм Берлекэмпа — Месси возвращает LFSR минимальной длины. □

Следующая теорема показывает, что в случае совместности системы (1) алгоритмы 1 и 2 эквивалентны.

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

Доказательство. Пусть $(L, f(x))$ — результат работы алгоритма 1, $g(x)$ — многочлен, вычисленный с помощью формул (10), $\widetilde{f}(x) = \lambda v_j(x)$, $\widetilde{g}(x) = \lambda r_j(x)$ — результат работы алгоритма 2. По предложению 1 выполнено равенство $L = \max\{\deg f(x), \deg g(x) + 1\}$. По теореме 2 для некоторого многочлена $\mu(x) \in F[x]$ выполнены равенства $f(x) = \mu(x)\widetilde{f}(x)$, $g(x) = \mu(x)\widetilde{g}(x)$. Обозначим $\widetilde{L} = \max\{\deg \widetilde{f}(x), \deg \widetilde{g}(x) + 1\}$. Тогда

$$-a_j = \sum_{i=1}^{\widetilde{L}} \widetilde{f}_i a_{j-i}, \quad j = \widetilde{L}+1, \widetilde{L}+2, \ldots, 2n.$$

Предположим, что $\deg \mu(x) \geqslant 1$. Тогда из неравенств $\deg f(x) > \deg \widetilde{f}(x)$, $\deg g(x) > \deg \widetilde{g}(x)$ следует, что $\widetilde{L} < L$, т. е. LFSR $(L, f(x))$ имеет не минимальную длину. Противоречие со свойством минимальности длины LFSR алгоритма Берлекэмпа — Месси.

Таким образом, $\mu(x)$ является многочленом нулевой степени. Так как $f(0) = 1 = \widetilde{f}(0)$, то $\mu(x) = 1$. □

Следствие 1. Алгоритм 2 возвращает LFSR $(L, f(x))$ минимальной длины $L = \max\{\deg v_j(x), \deg r_j(x) + 1\}$.

3. Алгоритм Берлекэмпа — Месси в случае поля характеристики два

Хорошо известно следующее утверждение (см., напр., [1]).

Теорема 4. Пусть F — поле характеристики два, $a_1, a_2, \ldots, a_{2n} \in F$, $a_{2i} = a_i^2$ для любого $i = 1, \ldots, n-1$. Пусть $(L, f(x))$ — LFSR, построенный для последовательности $a_1, a_2, \ldots, a_{2n-1}$, причем $L \leqslant n-1$. Тогда следующие условия эквивалентны:

1) $a_{2n} = \sum\limits_{i=1}^{L} f_i a_{2n-i}$, т. е. элемент a_{2n} можно получить с помощью LFSR $(L, f(x))$;
2) $a_{2n} = a_n^2$.

Следствие 2. Пусть F — поле характеристики два, $a_1, a_2, \ldots, a_{2n} \in F$, для которой система (1) имеет решение, причем $a_{2i} = a_i^2$ для любого $i = 1, \ldots, n$. Если к данной последовательности применить алгоритм Берлекэмпа — Месси, то $\Delta_{2i} = 0$ для любого $i = 1, \ldots, n$.

Следствие 2 показывает, что для указанной последовательности $a_1, a_2, \ldots, a_{2n}$ в алгоритме Берлекэмпа — Месси достаточно рассматривать только итерации с нечетными номерами, что ускоряет алгоритм 1. Этот момент отражен в следующем алгоритме.

Алгоритм 3 (алгоритм Берлекэмпа — Месси для случая $\operatorname{char} F = 2$).

Вход: последовательность $a_1, a_2, \ldots, a_{2n} \in F$, $\operatorname{char} F = 2$, для которой система (1) имеет решение, причем $a_{2i} = a_i^2$ для любого $i = 1, \ldots, n$.

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

$$a_j = \sum_{i=1}^{L} f_i a_{j-i}, \quad j = L+1, L+2, \ldots, 2n. \tag{11}$$

1. Определить $r := 0,\ f(x) := 1,\ b(x) := 1,\ L := 0.$
2. Если $a_1 = 0$, то $b(x) := x.$
3. Иначе (т. е. если $a_1 \neq 0$) :
   $f(x) := 1 + a_1 \cdot x, \quad b(x) := a_1^{-1}, \quad L := 1.$
4. Цикл $i := 2, \ldots, n$:
   4.1. Определить $r := 2i - 1;$
   4.2. Определить $\Delta := a_r + \sum\limits_{i=1}^{L} f_i a_{r-i};$
   4.3. Если $\Delta = 0$, то $b(x) := x^2 \cdot b(x);$
   4.4. Если $\Delta \neq 0$:
      4.4.1. Если $2L < r$:
         $buf(x) := f(x) + \Delta \cdot x^2 \cdot b(x),$
         $b(x) := \Delta^{-1} \cdot f(x),$
         $f(x) := buf(x),$
         $L := r - L;$
      4.4.2. Иначе (т. е. выполнено $2L \geqslant r$):
         $f(x) := f(x) + \Delta \cdot x^2 \cdot b(x),$
         $b(x) := x^2 \cdot b(x).$

4. Декодирование кодов БЧХ и кодов Рида — Соломона с использованием алгоритма Берлекэмпа — Месси

Кодом Боуза — Чоудхури — Хоквингема (БЧХ) над полем $GF(q)$ называется такой циклический код длины n, порождающий многочлен $g(x)$ наименьшей степени которого имеет своими корнями последовательность идущих подряд степеней некоторого произвольного элемента $\alpha \in GF(q^m)$: $: \alpha^b,\ \alpha^{b+1}, \ldots, \alpha^{b+\delta-2}$, где b — некоторое неотрицательное целое число, $\delta \geqslant 2$, причем

$$n = \begin{cases} \operatorname{ord}(\alpha), & \delta > 2, \\ \operatorname{ord}(\alpha^b), & \delta = 2, \end{cases}$$

где ord — порядок элемента. Из данного определения следует, что длина кода n делит число $q^m - 1$ (порядок мультипликативной группы поля $GF(q^m)$).

Кодом Рида — Соломона (РС) называется код БЧХ над полем $GF(q)$, $q > 2$, который имеет длину $q-1$. Из данного определения следует, что при $\delta > 2$ выполнены такие условия: $\alpha \in GF(q)$ и α является примитивным элементом поля $GF(q)$. Также из этого определения следует, что порождающий многочлен кода РС имеет вид

$$g(x) = (x - \alpha^b)(x - \alpha^{b+1}) \ldots (x - \alpha^{b+\delta-2}),$$

т. е. коэффициенты данного многочлена принадлежат полю $GF(q)$. Хорошо известно, что коды РС являются МДР-кодами, т. е. кодами с максимально достижимым расстоянием ($d = n - k + 1$).

Рассмотрим случай, когда в канале связи действуют ошибки и стирания. Заметим, что в полученных ниже алгоритмах достаточно положить $s = 0$ (число стираний), чтобы рассматривать случай только ошибок.

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 и стирания. Пусть t — максимальное число возможных ошибок при фиксированном числе стираний s в векторе v, $d \geqslant 2t + s + 1$, $t = [(d - s - 1)/2]$, m — реальное число ошибок, $m \leqslant t$. Так как позиции стертых символов известны, то при $s > 0$ заменим эти символы в векторе v на нули и будем обращаться с полученным вектором $\widetilde{v}$ как с вектором, содержащим только ошибки. Пусть ошибки произошли на позициях $i_1, \ldots, i_m$, а стирания — на позициях $i_{m+1}, \ldots, i_{m+s}$. При этом известны только позиции $i_{m+1}, \ldots, i_{m+s}$. После того как на данные позиции поместили нули, с какими-то позициями могли угадать (если в кодовом векторе там действительно стояли нули). Поэтому $\widetilde{v} = u + e$, где e — вектор ошибок веса не более $m + s$.

Пусть $X_1 = \alpha^{i_1}, \ldots, X_m = \alpha^{i_m}$ — неизвестные локаторы ошибок, $X_{m+1} = \alpha^{i_{m+1}}, \ldots, X_{m+s} = \alpha^{i_{m+s}}$ — известные локаторы стираний, $Y_1 = e_{i_1}, \ldots, Y_{m+s} = e_{i_{m+s}}$ — значения ошибок в векторе $\widetilde{v}$. Найдем компоненты синдромного вектора:

$$\begin{aligned} &S_0 = \widetilde{v}(\alpha) = Y_1 X_1 + \ldots + Y_m X_m + Y_{m+1} X_{m+1} + \ldots + Y_{m+s} X_{m+s}, \\ &S_1 = \widetilde{v}(\alpha^2) = Y_1 X_1^2 + \ldots + Y_m X_m^2 + Y_{m+1} X_{m+1}^2 + \ldots + Y_{m+s} X_{m+s}^2, \\ &\ldots \\ &S_{2t+s-1} = \widetilde{v}(\alpha^{2t+s}) = Y_1 X_1^{2t+s} + \ldots + Y_m X_m^{2t+s} + Y_{m+1} X_{m+1}^{2t+s} + \ldots + Y_{m+s} X_{m+s}^{2t+s}. \end{aligned}$$

Запишем синдромный многочлен в виде

$$S(x)=\sum_{i=0}^{2t+s-1}S_ix^i=\sum_{i=0}^{2t+s-1}\left(\sum_{j=1}^{m+s}Y_jX_j^{i+1}\right)x^i=\sum_{j=1}^{m+s}Y_jX_j\left(\sum_{i=0}^{2t+s-1}(X_jx)^i\right)=$$

$$=\sum_{j=1}^{m+s}Y_jX_j\frac{1-(X_jx)^{2t+s}}{1-X_jx}=\sum_{j=1}^{m+s}\frac{Y_jX_j}{1-X_jx}-x^{2t+s}\sum_{j=1}^{m+s}\frac{Y_jX_j^{2t+s+1}}{1-X_jx}.$$

Полагая

$$\widetilde{\sigma}(x)=\prod_{i=1}^{m+s}(1-X_ix)=\sum_{i=0}^{m+s}\widetilde{\sigma}_ix^i,\quad \widetilde{\sigma}_0=1,$$

$$\widetilde{\omega}(x)=\sum_{i=1}^{m+s}Y_iX_i\prod_{\substack{1\leqslant j\leqslant m+s,\\ j\neq i}}(1-X_jx),\quad \widetilde{\Phi}(x)=\sum_{i=1}^{m+s}Y_iX_i^{2t+s+1}\prod_{\substack{1\leqslant j\leqslant m+s,\\ j\neq i}}(1-X_jx),$$

после приведения всех дробей к общему знаменателю получим

$$S(x)=\frac{\widetilde{\omega}(x)}{\widetilde{\sigma}(x)}-x^{2t+s}\frac{\widetilde{\Phi}(x)}{\widetilde{\sigma}(x)}.$$

Тогда

$$S(x)\widetilde{\sigma}(x)=\widetilde{\omega}(x)-x^{2t+s}\widetilde{\Phi}(x).$$

Данное выражение называют ключевым уравнением, которому можно придать иной вид:

$$\widetilde{\sigma}(x)S(x)\equiv\widetilde{\omega}(x)\pmod{x^{2t+s}}. \tag{12}$$

Заметим, что $\widetilde{\sigma}(x)=\sigma(x)\nu(x)$, где $\sigma(x)$ — это многочлен неизвестных локаторов ошибок, $\nu(x)$ — многочлен известных локаторов стираний:

$$\widetilde{\sigma}(x)=\prod_{i=1}^{m}(1-X_ix)\prod_{i=1}^{s}(1-X_{m+i}x)=\sigma(x)\nu(x).$$

Введем в рассмотрение многочлен $\widetilde{S}(x)=S(x)\nu(x)$ — модифицированный синдромный многочлен. Тогда ключевое уравнение (12) примет вид

$$\sigma(x)\widetilde{S}(x)\equiv\widetilde{\omega}(x)\pmod{x^{2t+s}}, \tag{13}$$

где

$$\deg\sigma(x)\leqslant m,\quad \deg\widetilde{\omega}(x)\leqslant m+s-1,\quad \sigma(0)=1. \tag{14}$$

Пусть

$$\widetilde{S}(x)=\widetilde{S}_0+\widetilde{S}_1x+\ldots+\widetilde{S}_{2t+2s-1}x^{2t+2s-1}=$$

$$=S(x)\nu(x)=(S_0+S_1x+\ldots+S_{2t+s-1}x^{2t+s-1})(\nu_0+\nu_1x+\ldots+\nu_sx^s),$$

где $\nu_0=1$, $\nu_i=(-1)^i\sigma_i(X_{m+1},\ldots,X_{m+s})$ — элементарный симметрический многочлен от $X_{m+1},\ldots,X_{m+s}$, $i=1,\ldots,s$.

Так как в сравнении (13) $\deg\widetilde{\omega}(x)\leqslant m+s-1$, $\deg\widetilde{S}(x)\leqslant 2t+2s-1$, $\deg\sigma(x)\leqslant m$, то необходимым условием выполнения данного сравнения является тот факт, что коэффициенты многочлена $\sigma(x)\widetilde{S}(x)$ при степенях $j=m+s,m+s+1,\ldots,2t+s-1$ равны нулю. Поэтому получаем такую систему линейных уравнений:

$$\begin{cases}\sigma_0\widetilde{S}_{s+m}+\sigma_1\widetilde{S}_{s+m-1}+\ldots+\sigma_m\widetilde{S}_s=0,\\ \sigma_0\widetilde{S}_{s+m+1}+\sigma_1\widetilde{S}_{s+m}+\ldots+\sigma_m\widetilde{S}_{s+1}=0,\\ \ldots\\ \sigma_0\widetilde{S}_{s+2t-1}+\sigma_1\widetilde{S}_{s+2t-2}+\ldots+\sigma_m\widetilde{S}_{s+2t-m-1}=0.\end{cases}$$

Так как $\sigma_0=1$, то данная система в матричной форме примет такой вид:

$$\begin{pmatrix}\widetilde{S}_{s+m-1} & \widetilde{S}_{s+m-2} & \ldots & \widetilde{S}_s\\ \widetilde{S}_{s+m} & \widetilde{S}_{s+m-1} & \ldots & \widetilde{S}_{s+1}\\ \ldots & \ldots & \ldots & \ldots\\ \widetilde{S}_{s+2t-2} & \widetilde{S}_{s+2t-3} & \ldots & \widetilde{S}_{s+2t-m-1}\end{pmatrix}\begin{pmatrix}\sigma_1\\ \sigma_2\\ \ldots\\ \sigma_m\end{pmatrix}=\begin{pmatrix}-\widetilde{S}_{s+m}\\ -\widetilde{S}_{s+m+1}\\ \ldots\\ -\widetilde{S}_{s+2t-1}\end{pmatrix}. \tag{15}$$

Будем искать решение данной системы с помощью алгоритма Берлекэмпа — Месси.

Теорема 5. Пусть $d \geqslant 2t+s+1$, $m \leqslant t$. Если на вход алгоритма 1 подать последовательность $\widetilde{S}_s$, $\widetilde{S}_{s+1}, \ldots$, $\widetilde{S}_{s+2t-1}$, то на выходе алгоритма будет верное значение многочлена локаторов ошибок $\sigma(x)$.

Доказательство. Пусть $\widetilde{\sigma}(x)$ — многочлен, полученный после применения алгоритма 1. Так как коэффициенты многочлена локаторов ошибок $\sigma(x)$ являются решением системы (15), то в теореме 1 будет выполнено неравенство $L \leqslant m$. Удалив в системе (15) $2t-2m$ последних уравнений, получим новую систему с квадратной матрицей системы порядка m. Из теоремы 5 работы [7] следует, что данная матрица невырождена, поэтому полученная новая система имеет единственное решение. Это значит, что $\widetilde{\sigma}(x) = \sigma(x)$ и $L = m$. □

Учитывая теорему 5, получаем следующий алгоритм декодирования кодов РС (кодов БЧХ).

Алгоритм 4 (декодирование кода РС на основе алгоритма Берлекэмпа — Месси на случай ошибок и стираний).

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

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

1. Определяется $t = [(d-s-1)/2]$. В векторе v все стирания заменяют нулями, получая тем самым вектор $\widetilde{v}$. Находятся компоненты $S_0, S_1, \ldots, S_{2t+s-1}$ синдромного вектора: $S_i = \widetilde{v}(\alpha^{i+1})$, $i = 0, 1, \ldots, 2t+s-1$. Если они все равны нулю, то возвращается вектор $\widetilde{v}$ и процедура окончена.

   Вычисляются значения локаторов стираний $X_{t+1} = \alpha^{i_{t+1}}, \ldots, X_{t+s} = \alpha^{i_{t+s}}$ на основе известных позиций стираний $i_{t+1}, \ldots, i_{t+s}$. Вычисляются коэффициенты модифицированного синдромного многочлена $\widetilde{S}(x)$ (если $s = 0$, то $\widetilde{S}(x) = S(x)$).

2. На вход алгоритма 1 подается последовательность $\widetilde{S}_s$, $\widetilde{S}_{s+1}, \ldots$, $\widetilde{S}_{s+2t-1}$. На выходе данного алгоритма получается многочлен $\sigma(x)$. Пусть $l = \deg \sigma(x)$.

3. Отыскиваются l корней многочлена $\sigma(x)$ последовательной подстановкой в него ненулевых элементов поля F. При этом локаторы ошибок — это величины, обратные корням многочлена $\sigma(x)$.

4. Определяется множество $M = \{1, \ldots, l\} \cup \{t+1, \ldots, t+s\}$. По формулам Форни

$$Y_j = \frac{X_j^{-1}\widetilde{\omega}(X_j^{-1})}{\prod\limits_{i \in M \setminus \{j\}} (1 - X_i X_j^{-1})}, \quad j \in M, \tag{16}$$

где $\widetilde{\omega}(x) \equiv \sigma(x)\widetilde{S}(x) \pmod{x^{2t+s}}$, находятся значения ошибок Y_j, $j \in M$. У вектора $\widetilde{v}$ из i_j-го символа, $X_j = \alpha^{i_j}$, вычитается значение Y_j, $j \in M$. При этом получается кодовый вектор u.

Пример 1. 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 $GF(2^4)$:

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

Рассмотрим код Рида — Соломона с параметрами $n = 15$, $k = 7$, $d = 9$. В этом случае код может исправить четыре и менее ошибок, либо три и менее ошибок и два и менее стираний, либо две и менее ошибок и четыре и менее стираний, либо одну ошибку и шесть и менее стираний, либо восемь и менее стираний.

Рассмотрим случай возможности исправления до двух ошибок и до четырех стираний. Пусть на приемном конце получен вектор

$$v = (\ \alpha^7,\ \alpha^{10},\ \alpha,\ 1,\ \alpha^{12},\ \alpha^{12},\ \alpha^5,\ *,\ \alpha^2,\ *,\ *,\ \alpha^6,\ *,\ \alpha^3,\ 1\ ),$$

в котором не более двух ошибок и четыре стирания. Применим алгоритм декодирования 4.

1. Полагаем $s = 4$, $t = [(d-s-1)/2] = 2$. В данном случае нам известно, что

$$X_3 = \alpha^7, X_4 = \alpha^9, X_5 = \alpha^{10}, X_6 = \alpha^{12}.$$

Поэтому

$$\nu(x) = (1-\alpha^7 x)(1-\alpha^9 x)(1-\alpha^{10}x)(1-\alpha^{12}x) = \\ = 1+\alpha^{14}x+x^2+\alpha^3x^3+\alpha^8x^4.$$

Заменим в векторе v $*$ на 0:

$$\widetilde{v} = (\ \alpha^7,\ \alpha^{10},\ \alpha,\ 1,\ \alpha^{12},\ \alpha^{12},\ \alpha^5,\ 0,\ \alpha^2,\ 0,\ 0,\ \alpha^6,\ 0,\ \alpha^3,\ 1\ ).$$

Вычислим компоненты синдрома для вектора $\widetilde{v}$:

$$S_0 = \widetilde{v}(\alpha) = \alpha^8,\ \ S_1 = \widetilde{v}(\alpha^2) = \alpha^5,\ \ S_2 = \widetilde{v}(\alpha^3) = \alpha^4,$$
$$S_3 = \widetilde{v}(\alpha^4) = \alpha^7,\ \ S_4 = \widetilde{v}(\alpha^5) = \alpha^4,\ \ S_5 = \widetilde{v}(\alpha^6) = \alpha^{11},$$
$$S_6 = \widetilde{v}(\alpha^7) = \alpha^7,\ \ S_7 = \widetilde{v}(\alpha^8) = \alpha^9.$$

Поэтому синдромный многочлен имеет такой вид:

$$S(x) = \alpha^8+\alpha^5x+\alpha^4x^2+\alpha^7x^3+\alpha^4x^4+\alpha^{11}x^5+\alpha^7x^6+\alpha^9x^7.$$

Тогда

$$\widetilde{S}(x) = S(x)\nu(x) = \widetilde{S}_0+\widetilde{S}_1x+\ldots+\widetilde{S}_{11}x^{11} = \\ = \alpha^8+\alpha^{13}x+\alpha^8x^2+\alpha^7x^3+\alpha^7x^4+\alpha^7x^5+\alpha^{10}x^6+\alpha x^7+\alpha^3x^8+\alpha^{11}x^9+\alpha^{11}x^{10}+\alpha^2x^{11}.$$

2. На вход алгоритма 1 подаем последовательность $\widetilde{S}_4 = \alpha^7$, $\widetilde{S}_5 = \alpha^7$, $\widetilde{S}_6 = \alpha^{10}$, $\widetilde{S}_7 = \alpha$. После вычислений

r	Δ	$\sigma(x)$	$b(x)$	L
0		1	1	0
1	α^7	$1+\alpha^7x$	α^8	1
2	α	$1+x$	α^8x	1
3	α^6	$1+x+\alpha^{14}x^2$	$\alpha^9+\alpha^9x$	2
4	α^{14}	$1+\alpha^2x+\alpha^6x^2$	$\alpha^9x+\alpha^9x^2$	2

получаем $\sigma(x) = 1+\alpha^2x+\alpha^6x^2$.

3. Корнями многочлена локаторов ошибок $\sigma(x)$ являются $x_1 = \alpha^{14}$, $x_2 = \alpha^{10}$, поэтому $X_1 = \alpha$, $X_2 = \alpha^5$.

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

$$Y_i = \frac{X_i^{-1}\widetilde{\omega}(X_i^{-1})}{\prod\limits_{\substack{1\leqslant j\leqslant t+s,\\ j\neq i}}(1-X_jX_i^{-1})},\quad i = 1,2,\ldots,t+s,$$

где

$$\widetilde{\omega}(x) \equiv \sigma(x)\widetilde{S}(x) \equiv \\ \equiv \alpha^8+\alpha^9x+\alpha^{13}x^2+\alpha^{12}x^3+\alpha^3x^4+\alpha^6x^5\ \pmod{x^8}.$$

Находим значения ошибок: $Y_1 = \alpha^6, Y_2 = \alpha, Y_3 = 0,\ Y_4 = \alpha^6, Y_5 = \alpha^{10},\ Y_6 = \alpha^{11}$. Таким образом:

$$e = (\ 0,\ \alpha^6,\ 0,\ 0,\ 0,\ \alpha,\ 0,\ 0,\ 0,\ \alpha^6,\ \alpha^{10},\ 0,\ \alpha^{11},\ 0,\ 0\ ),$$
$$u = v-e = (\ \alpha^7,\ \alpha^7,\ \alpha,\ 1,\ \alpha^{12},\ \alpha^{13},\ \alpha^5,\ 0,\ \alpha^2,\ \alpha^6,\ \alpha^{10},\ \alpha^6,\ \alpha^{11},\ \alpha^3,\ 1\ ).$$

5. Декодирование двоичных кодов БЧХ с использованием алгоритма Берлекэмпа — Месси

В данном параграфе рассмотрим случай только ошибок для возможности применения алгоритма 3. На основе алгоритмов 3 и 4 получаем следующий алгоритм декодирования.

Алгоритм 5 (декодирование двоичного кода БЧХ на основе алгоритма Берлекэмпа — Месси).
Вход: полученный вектор v.
Выход: исходный кодовый вектор u, если произошло не более $[(d-1)/2]$ ошибок.

1. Определяется $t = [(d-1)/2]$. Находятся компоненты $S_0, S_1, \ldots, S_{2t-1}$ синдромного вектора: $S_i = v(\alpha^{i+1})$, $i = 0,1,\ldots,2t-1$. Если синдромный вектор нулевой, то алгоритм завершается и возвращается $u = v$.

2. Для последовательности $S_0, S_1, \ldots, S_{2t-1}$ с помощью алгоритма 3 находится многочлен локаторов ошибок $\sigma(x)$. Пусть $l = \deg\sigma(x)$.

3. Отыскиваются l корней многочлена $\sigma(x)$. При этом локаторы ошибок — это величины, обратные корням многочлена $\sigma(x)$.

   У вектора v из символа с номером i_j, $X_j = \alpha^{i_j}$, вычитается 1, $j = 1, \ldots, l$. Тем самым получается вектор u.

Пример 2. Пусть поле $GF(2^4)$ порождается примитивным многочленом $p(x) = x^4 + x + 1$ (см. пример 1), код БЧХ длины $n = 15$ порождается многочленом:

$$g(x) = 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10}.$$

В данном случае $k = 15 - 10 = 5$. Подряд идущици корнями многочлена $g(x)$ будут:

$$\alpha,\ \alpha^2,\ \alpha^3,\ \alpha^4,\ \alpha^5,\ \alpha^6.$$

Поэтому конструктивное расстояние данного кода равно $d = 7$, т. е. код гарантированно может исправлять до трех ошибок.

Пусть на приемном конце получен вектор

$$v = \left(\ 1,\ \ 1,\ \ 0,\ \ 0,\ \ 1,\ \ 0,\ \ 0,\ \ 1,\ \ 1,\ \ 0,\ \ 1,\ \ 0,\ \ 1,\ \ 1,\ \ 0\ \right),$$

в котором не более трех ошибок. Данный вектор соответствует многочлену

$$v(x) = 1 + x + x^4 + x^7 + x^8 + x^{10} + x^{12} + x^{13}.$$

Вычисляем

$$\begin{aligned}
&S_0 = v(\alpha) = 1 + \alpha + \alpha^4 + \alpha^7 + \alpha^8 + \alpha^{10} + \alpha^{12} + \alpha^{13} = \alpha^7,\\
&S_2 = v(\alpha^3) = 1 + \alpha^3 + \alpha^{12} + \alpha^6 + \alpha^9 + 1 + \alpha^6 + \alpha^9 = \alpha^{10},\\
&S_4 = v(\alpha^5) = 1 + \alpha^5 + \alpha^5 + \alpha^5 + \alpha^{10} + \alpha^5 + 1 + \alpha^5 = 1,\\
&S_1 = S_0^2 = \alpha^{14}, \quad S_3 = S_1^2 = \alpha^{13}, \quad S_5 = S_2^2 = \alpha^5.
\end{aligned}$$

Применим к этой последовательности алгоритм 3

r	Δ	$\sigma(x)$	$b(x)$	L
0		1	1	0
1	α^7	$1 + \alpha^7 x$	α^8	1
3	α^7	$1 + \alpha^7 x + x^2$	$\alpha^8 + x$	2
5	0	$1 + \alpha^7 x + x^2$	$\alpha^8 x^2 + x^3$	2

Поэтому многочлен локаторов ошибок имеет вид

$$\sigma(x) = 1 + \alpha^7 x + x^2.$$

Находим корни многочлена $\sigma(x)$: $x_1 = \alpha^{14}$, $x_2 = \alpha$. Поэтому

$$X_1 = x_1^{-1} = \alpha,\ \ X_2 = x_2^{-1} = \alpha^{14},$$

$$e = \left(\ 0,\ \ 1,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 1\ \right),$$

$$u = v - e = \left(\ 1,\ \ 0,\ \ 0,\ \ 0,\ \ 1,\ \ 0,\ \ 0,\ \ 1,\ \ 1,\ \ 0,\ \ 1,\ \ 0,\ \ 1,\ \ 1,\ \ 1\ \right).$$

6. Декодирование обобщенных кодов Рида — Соломона с использованием алгоритма Берлекэмпа — Месси

Рассматриваемая в данном параграфе теория частично отражена в работе [8], являющейся продолжением работы [7]. Эти две работы посвящены описанию различных алгоритмов декодирования обобщенных кодов Рида — Соломона на случай ошибок и стираний. Так как данная работа посвящена алгоритмам декодирования на основе алгоритма Берлекэмпа — Месси, то для удобства читателя приведем алгоритм декодирования и здесь.

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

$$u = (y_0 b(\alpha_0), y_1 b(\alpha_1), \ldots, y_{n-1} b(\alpha_{n-1})), \tag{17}$$

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 $\alpha_i = \alpha^i$, $i = 0, 1, \ldots, n-1$, где α — примитивный элемент поля F, то в этом случае получаем код Рида — Соломона.

Заметим, 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 v — полученный на приемной стороне вектор, в котором могут быть ошибки и стирания. Пусть t — максимальное число возможных ошибок при фиксированном числе стираний s в векторе v, $d \geqslant 2t+s+1$, $t=[(d-s-1)/2]$, m — реальное число ошибок, $m \leqslant t$. Как и ранее, заменим стертые символы в векторе v, например, на нули и будем обращаться с полученным вектором $\widetilde{v}$ как с вектором, содержащим только ошибки. Пусть ошибки произошли на позициях $i_1,\ldots,i_m$, а стирания — на позициях $i_{m+1},\ldots,i_{m+s}$. При этом известны только позиции $i_{m+1},\ldots,i_{m+s}$. Получаем $\widetilde{v}=u+e$, где e — вектор ошибок веса не более $m+s$.

Вычисляя синдромный вектор, получаем

$$S=\widetilde{v}H^T=eH^T=\left(\ \ldots,\quad e_{i_1},\quad \ldots,\quad e_{i_{m+s}},\quad \ldots\ \right)\times$$

$$\times\left(\left(\begin{array}{cccc}1 & 1 & \ldots & 1\\ \alpha_0 & \alpha_1 & \ldots & \alpha_{n-1}\\ \ldots & \ldots & \ldots & \ldots\\ \alpha_0^{n-k-1} & \alpha_1^{n-k-1} & \ldots & \alpha_{n-1}^{n-k-1}\end{array}\right)\left(\begin{array}{cccc}w_0 & 0 & \ldots & 0\\ 0 & w_1 & \ldots & 0\\ \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & \ldots & w_{n-1}\end{array}\right)\right)^T,$$

$$S_j=Z_1X_1^j+\ldots+Z_mX_m^j+Z_{m+1}X_{m+1}^j+\ldots+Z_{m+s}Z_{m+s}^j,$$
$$j=0,1,\ldots,2t+s-1,$$

где $X_1=\alpha_{i_1},\ldots$, $X_m=\alpha_{i_m}$ — неизвестные локаторы ошибок, $X_{m+1}=\alpha_{i_{m+1}},\ldots$, $X_{m+s}=\alpha_{i_{m+s}}$ — известные локаторы стираний, $Y_1=e_{i_1},\ldots,Y_{m+s}=e_{i_{m+s}}$ — значения ошибок в векторе $\widetilde{v}$, $Z_j=Y_jw_{i_j}$, $j=1,\ldots,m+s$. Пусть $\sigma(x)$ — это многочлен неизвестных локаторов ошибок, $\nu(x)$ — многочлен известных локаторов стираний. Полагая

$$S(x)=\sum_{i=0}^{2t+s-1}S_ix^i,\quad \widetilde{S}(x)=S(x)\nu(x),\quad \widetilde{\sigma}(x)=\prod_{i=1}^{m}(1-X_ix)\prod_{i=1}^{s}(1-X_{m+i}x)=\sigma(x)\nu(x),$$

$$\widetilde{\omega}(x)=\sum_{i=1}^{m+s}Z_i\prod_{\substack{1\leqslant j\leqslant m+s,\\ j\neq i}}(1-X_jx),\quad \widetilde{\Phi}(x)=\sum_{i=1}^{m+s}Z_iX_i^{2t+s}\prod_{\substack{1\leqslant j\leqslant m+s,\\ j\neq i}}(1-X_jx),$$

получим ключевое уравнение

$$\sigma(x)\widetilde{S}(x)\equiv\widetilde{\omega}(x)\pmod{x^{2t+s}},\tag{18}$$

где

$$\deg\sigma(x)\leqslant m,\quad \deg\widetilde{\omega}(x)\leqslant m+s-1,\quad \sigma(0)=1.\tag{19}$$

Пусть

$$\widetilde{S}(x)=\widetilde{S}_0+\widetilde{S}_1x+\ldots+\widetilde{S}_{2t+2s-1}x^{2t+2s-1}=$$
$$=S(x)\nu(x)=(S_0+S_1x+\ldots+S_{2t+s-1}x^{2t+s-1})(\nu_0+\nu_1x+\ldots+\nu_sx^s),$$

где $\nu_0=1$, $\nu_i=(-1)^i\sigma_i(X_{m+1},\ldots,X_{m+s})$ — элементарный симметрический многочлен от $X_{m+1},\ldots,X_{m+s}$, $i=1,\ldots,s$.

Так как в сравнении (18) $\deg\widetilde{\omega}(x)\leqslant m+s-1$, $\deg\widetilde{S}(x)\leqslant 2t+2s-1$, $\deg\sigma(x)\leqslant m$, то необходимым условием выполнения данного сравнения является тот факт, что коэффициенты многочлена $\sigma(x)\widetilde{S}(x)$ при степенях $j=m+s,m+s+1,\ldots,2t+s-1$ равны нулю. Поэтому получаем такую систему линейных уравнений:

$$\begin{cases}\sigma_0\widetilde{S}_{s+m}+\sigma_1\widetilde{S}_{s+m-1}+\ldots+\sigma_m\widetilde{S}_s=0,\\ \sigma_0\widetilde{S}_{s+m+1}+\sigma_1\widetilde{S}_{s+m}+\ldots+\sigma_m\widetilde{S}_{s+1}=0,\\ \ldots\\ \sigma_0\widetilde{S}_{s+2t-1}+\sigma_1\widetilde{S}_{s+2t-2}+\ldots+\sigma_m\widetilde{S}_{s+2t-m-1}=0.\end{cases}$$

Так как $\sigma_0=1$, то данная система в матричной форме примет такой вид:

$$\left(\begin{array}{cccc}\widetilde{S}_{s+m-1} & \widetilde{S}_{s+m-2} & \ldots & \widetilde{S}_s\\ \widetilde{S}_{s+m} & \widetilde{S}_{s+m-1} & \ldots & \widetilde{S}_{s+1}\\ \ldots & \ldots & \ldots & \ldots\\ \widetilde{S}_{s+2t-2} & \widetilde{S}_{s+2t-3} & \ldots & \widetilde{S}_{s+2t-m-1}\end{array}\right)\left(\begin{array}{c}\sigma_1\\ \sigma_2\\ \ldots\\ \sigma_m\end{array}\right)=\left(\begin{array}{c}-\widetilde{S}_{s+m}\\ -\widetilde{S}_{s+m+1}\\ \ldots\\ -\widetilde{S}_{s+2t-1}\end{array}\right).\tag{20}$$

Теорема 6. Пусть $d\geqslant 2t+s+1$, $m\leqslant t$. Если на вход алгоритма 1 подать последовательность $\widetilde{S}_s$, $\widetilde{S}_{s+1},\ldots$, $\widetilde{S}_{s+2t-1}$, то на выходе алгоритма будет верное значение многочлена локаторов ошибок $\sigma(x)$.

Доказательство аналогично доказательству теоремы 5. □

Алгоритм 6 (декодирование обобщенного кода РС на основе алгоритма Берлекэмпа — Месси на случай ошибок и стираний).

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

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

1. Определяется $t = [(d - s - 1)/2]$. В векторе v все стирания заменяют нулями, получая тем самым вектор $\widetilde{v}$. Находятся компоненты $S_0, S_1, \ldots, S_{2t+s-1}$ синдромного вектора $\widetilde{v}H^T$. Если они все равны нулю, то возвращается вектор $\widetilde{v}$ и процедура окончена.

   Вычисляются значения локаторов стираний $X_{t+1} = \alpha_{i_{t+1}}, \ldots, X_{t+s} = \alpha_{i_{t+s}}$ на основе известных позиций стираний $i_{t+1}, \ldots, i_{t+s}$. Вычисляются коэффициенты модифицированного синдромного многочлена $\widetilde{S}(x)$ (если $s = 0$, то $\widetilde{S}(x) = S(x)$).

2. На вход алгоритма 1 подается последовательность $\widetilde{S}_s$, $\widetilde{S}_{s+1}, \ldots$, $\widetilde{S}_{s+2t-1}$. На выходе данного алгоритма получается многочлен $\sigma(x)$. Пусть $l = \deg \sigma(x)$.

3. Если $l > 0$, то отыскиваются l корней многочлена $\sigma(x)$ последовательной подстановкой в него ненулевых элементов поля F. При этом локаторы ошибок — это величины, обратные корням многочлена $\sigma(x)$.

4. При вычислении значений ошибок выполняется один из следующих пунктов.

   4.1. Если среди локаторов стираний $X_{t+1}, \ldots, X_{t+s}$ имеется нулевое значение (в противном случае переходим в пункт 4.2), скажем, $X_p = 0$, то пусть:

   $$M = \{1, \ldots, l\} \cup \{t + 1, \ldots, t + s\} \backslash \{p\}.$$

   Находятся Z_j, $j \in M$, например, с помощью формул Форни для ОРС кодов:

   $$Z_j = \frac{\widetilde{\omega}(X_j^{-1})}{\prod\limits_{i \in M \backslash \{j\}} (1 - X_i X_j^{-1})}, \quad j \in M. \tag{21}$$

   После этого находятся значения ошибок $Y_j = Z_j / w_{i_j}$, $j \in M$. У вектора $\widetilde{v}$ из i_j-го символа, $X_j = \alpha_{i_j}$, вычитается значение Y_j, $j \in M$. При этом получается вектор $\widetilde{u}$. Пусть для некоторого i выполнено $\alpha_i = 0$. Вычисляется значение Z_p, равное скалярному произведению вектора $\widetilde{u}$ на первую строку матрицы H. Вычисляется значение ошибки $Y_p = Z_p / w_i$. Осталось в векторе $\widetilde{u}$ из i-го символа вычесть Y_p.

   4.2. Если условие 4.1 не выполнено, то пусть:

   $$M = \{1, \ldots, l\} \cup \{t + 1, \ldots, t + s\}.$$

   По формуле (21) находятся значения Z_j, затем значения ошибок $Y_j = Z_j / w_{i_j}$, $j \in M$. У вектора $\widetilde{v}$ из i_j-го символа, $X_j = \alpha_{i_j}$, вычитается значение Y_j, $j \in M$. При этом получается вектор $\widetilde{u}$.

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

7. Декодирование кодов Гоппы с использованием алгоритма Берлекэмпа — Месси

Определение кода Гоппы опирается на два объекта: многочлен $G(x)$ с коэффициентами из поля $GF(q^m)$, который называется многочленом Гоппы; подмножество $L = \{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$ 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 $u = (u_0, u_1, \ldots, u_{n-1})$ 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. Множество L называется множеством нумераторов позиций кодового слова. Имеют место следующие оценки параметров для кодов Гоппы (см., напр., [9]).

Теорема 7. Параметры $[n,k,d]$-кода $\Gamma(L,G)$ над полем $GF(q)$, где $L \subseteq GF(q^m)$, связаны соотношением

$$n = |L|, \quad k \geqslant n - mr, \;\; r = \deg G(x), \quad d \geqslant r+1.$$

Нам понадобится следующее утверждение (см., напр., [1]).

Теорема 8. Код $\Gamma(L,G)$ представляет собой ограничение кода $GRS_{n-r}(L,y)$ на подполе $F = GF(q)$: $\Gamma(L,G) = GRS_{n-r}(L,y) \cap F^n$, где $r = \deg G(x)$, $y = (y_0, y_1, \ldots, y_{n-1})$,

$$y_i = G(\alpha_i) \prod_{j \neq i} \frac{1}{\alpha_i - \alpha_j}, \quad i = 0,1,\ldots,n-1. \tag{22}$$

Следствие 3. Проверочная матрица кода $GRS_{n-r}(L,y)$, который задает код $\Gamma(L,G)$, имеет вид

$$\begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha_0 & \alpha_1 & \ldots & \alpha_{n-1} \\ \ldots & \ldots & \ldots & \ldots \\ \alpha_0^{r-1} & \alpha_1^{r-1} & \ldots & \alpha_{n-1}^{r-1} \end{pmatrix} \begin{pmatrix} G(\alpha_0)^{-1} & 0 & \ldots & 0 \\ 0 & G(\alpha_1)^{-1} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & G(\alpha_{n-1})^{-1} \end{pmatrix},$$

т. е. совпадает с проверочной матрицей кода $\Gamma(L,G)$.

Таким образом, 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 $\overline{G}(x)$ — полный квадрат некоторого многочлена над $GF(2^m)$ наименьшей степени, делящийся на $G(x)$. В случае сепарабельного кода $\overline{G}(x) = G^2(x)$. Для минимального расстояния сепарабельного кода $\Gamma(L,G)$ верна оценка $d \geqslant 2r+1$ и выполнено равенство $\Gamma(L,G) = \Gamma(L,\overline{G})$ (см., напр., [9]). Эти факты позволяют строить сепарабельный код $\Gamma(L,G) = \Gamma(L,\overline{G})$, 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: $\Gamma(L,G) = GRS_{n-r}(L,y) \cap F^n$, $F = GF(q)$, $r = \deg G(x)$, $\widetilde{k} = n - r$ — размерность кода $GRS_{n-r}(L,y)$ длины n, $\overline{H}$ — проверочная матрица кода $GRS_{n-r}(L,y)$. Пусть $d, \widetilde{d}$ — кодовые расстояния кодов $\Gamma(L,G)$ и $GRS_{n-r}(L,y)$ соответственно. Так как $d \geqslant r+1$, $\widetilde{d} = n - \widetilde{k} + 1 = r + 1$, то если в кодовом векторе $u \in \Gamma(L,G)$ произошло t ошибок и s стираний, причем $r \geqslant 2t + s$, то для его декодирования можно применять алгоритмы декодирования для ОРС-кодов.

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

Алгоритм 7. (декодирование кода Гоппы на основе алгоритма Берлекэмпа — Месси на случай ошибок и стираний)

Вход: принятый вектор v.

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

1. Определяется $t = [(r-s)/2]$ ($t = [(2r-s)/2]$ в случае двоичного сепарабельного кода Гоппы). В векторе v все стирания заменяют нулями, получая тем самым вектор $\widetilde{v}$. Находятся компоненты $S_0, S_1, \ldots, S_{2t+s-1}$ синдромного вектора $\widetilde{v}\overline{H}^T$. Если они все равны нулю, то возвращается вектор $\widetilde{v}$ и процедура окончена.

   Вычисляются значения локаторов стираний $X_{t+1} = \alpha_{i_{t+1}}, \ldots, X_{t+s} = \alpha_{i_{t+s}}$ на основе известных позиций стираний $i_{t+1}, \ldots, i_{t+s}$. Вычисляются коэффициенты модифицированного синдромного многочлена $\widetilde{S}(x)$ (если $s = 0$, то $\widetilde{S}(x) = S(x)$).

2. На вход алгоритма 1 подается последовательность $\widetilde{S}_s$, $\widetilde{S}_{s+1}, \ldots$, $\widetilde{S}_{s+2t-1}$. На выходе данного алгоритма получается многочлен $\sigma(x)$. Пусть $l = \deg \sigma(x)$.

3. Отыскиваются l корней многочлена $\sigma(x)$ последовательной подстановкой в него ненулевых элементов поля $GF(q^m)$. При этом локаторы ошибок — это величины, обратные корням многочлена $\sigma(x)$.

4. При вычислении значений ошибок выполняется один из следующих пунктов.

   4.1. Если среди локаторов стираний $X_{t+1}, \dots, X_{t+s}$ имеется нулевое значение (в противном случае переходим в пункт 4.2), скажем, $X_p = 0$, то пусть

   $$M = \{1, \dots, l\} \cup \{t+1, \dots, t+s\} \backslash \{p\}$$

   — множество индексов локаторов ошибок и стираний без учета индекса p. Находятся Z_j, $j \in M$, например, с помощью алгоритма Форни (21) для обобщенных кодов РС. После этого находятся значения ошибок $Y_j = Z_j G(X_j)$, $j \in M$. У вектора $\widetilde{v}$ из i_j-го символа, $X_j = \alpha_{i_j}$, вычитается значение Y_j, $j \in M$. При этом получается вектор $\widetilde{u}$. Пусть для некоторого i выполнено $\alpha_i = 0$ (в противном случае все локаторы стираний были бы ненулевые). Вычисляется значение Z_p, равное скалярному произведению вектора $\widetilde{u}$ на первую строку матрицы $\overline{H}$. Вычисляется значение ошибки $Y_p = Z_p G(\alpha_i)$. Осталось в векторе $\widetilde{u}$ из i-го символа вычесть Y_p.

   4.2. Если условие 4.1 не выполнено, то пусть $M = \{1, \dots, l\} \cup \{t+1, \dots, t+s\}$. По формуле (21) находятся значения Z_j, затем — значения ошибок $Y_j = Z_j G(X_j)$, $j \in M$. У вектора $\widetilde{v}$ из i_j-го символа, $X_j = \alpha_{i_j}$, вычитается значение Y_j, $j \in M$. При этом получается вектор $\widetilde{u}$.

   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 $\widetilde{u}$ на первую строку матрицы $\overline{H}$, а затем вычисляется значение ошибки $Y_0 = Z_0 G(\alpha_i)$. Осталось в векторе $\widetilde{u}$ из i-го символа вычесть Y_0.

Пример 3. Рассмотрим расширение поля $GF(2) \subset GF(2^4)$, где поле $GF(2^4)$ строится на основе примитивного многочлена $p(x) = x^4 + x + 1$ и рассматривалось в примере 1. Пусть $L = GF(2^4) = \{0, 1, \alpha, \alpha^2, \dots, \alpha^{14}\}$, $G(x) = x^2 + x + \alpha^3$. Так как след элемента α^3 в поле $GF(2^4)$ не равен нулю, то многочлен $G(x)$ в этом поле не имеет корней. Поэтому из неприводимости $G(x)$ над $GF(2^4)$ следует, что код $\Gamma(L, G)$ является сепарабельным, поэтому он может исправлять до двух ошибок, либо одну ошибку и до двух стираний, либо до четырех стираний. Проверочная матрица H кода $\Gamma(L, G)$ примет такой вид:

$$H = \begin{pmatrix} G(0)^{-1} & G(1)^{-1} & G(\alpha)^{-1} & \dots & G(\alpha^{14})^{-1} \\ 0G(0)^{-1} & 1G(1)^{-1} & \alpha G(\alpha)^{-1} & \dots & \alpha^{14} G(\alpha^{14})^{-1} \end{pmatrix} =$$

$$= \begin{pmatrix} \alpha^{12} & \alpha^{12} & \alpha^4 & \alpha^3 & \alpha^9 & \alpha^4 & \alpha & \alpha^8 & \alpha^6 & \alpha^3 & \alpha^6 & \alpha & \alpha^2 & \alpha^2 & \alpha^8 & \alpha^9 \\ 0 & \alpha^{12} & \alpha^5 & \alpha^5 & \alpha^{12} & \alpha^8 & \alpha^6 & \alpha^{14} & \alpha^{13} & \alpha^{11} & 1 & \alpha^{11} & \alpha^{13} & \alpha^{14} & \alpha^6 & \alpha^8 \end{pmatrix} =$$

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

Так как все строки матрицы H линейно независимы, то $n - k = 8$, $k = 8$. Выписав построчно фундаментальную систему решений системы однородных линейных уравнений $HX = O$, находим порождающую матрицу кода $\Gamma(L, G)$:

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

В данном случае $\Gamma(L, G) \subseteq GRS_{12}(L, y)$, при этом проверочная матрица $\overline{H}$ кода $GRS_{12}(L, y)$, учитывая следствие 3, имеет вид

$$\overline{H} = \begin{pmatrix} \alpha^9 & \alpha^9 & \alpha^8 & \alpha^6 & \alpha^3 & \alpha^8 & \alpha^2 & \alpha & \alpha^{12} & \alpha^6 & \alpha^{12} & \alpha^2 & \alpha^4 & \alpha^4 & \alpha & \alpha^3 \\ 0 & \alpha^9 & \alpha^9 & \alpha^8 & \alpha^6 & \alpha^{12} & \alpha^7 & \alpha^7 & \alpha^4 & \alpha^{14} & \alpha^6 & \alpha^{12} & 1 & \alpha & \alpha^{14} & \alpha^2 \\ 0 & \alpha^9 & \alpha^{10} & \alpha^{10} & \alpha^9 & \alpha & \alpha^{12} & \alpha^{13} & \alpha^{11} & \alpha^7 & 1 & \alpha^7 & \alpha^{11} & \alpha^{13} & \alpha^{12} & \alpha \\ 0 & \alpha^9 & \alpha^{11} & \alpha^{12} & \alpha^{12} & \alpha^5 & \alpha^2 & \alpha^4 & \alpha^3 & 1 & \alpha^9 & \alpha^2 & \alpha^7 & \alpha^{10} & \alpha^{10} & 1 \end{pmatrix}.$$

Пусть на приемном конце получен вектор:

$$v = (1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1),$$

в котором до двух ошибок (как видно, стирающих символов в принятом векторе нет).

Для декодирования принятого вектора применим алгоритм 7.

1. Определяем $s = 0$, $t = \deg G(x) = 2$. Находим синдромный вектор $S = v\overline{H}^T = (\ \alpha^9,\ \alpha^9,\ \alpha^{10},\ \alpha^6\ )$.

2. Применяя к данной последовательности алгоритм Берлекэмпа — Месси 1, получаем $\sigma(x) = 1 + \alpha^2 x + \alpha^9 x^2$.

3. Корнями многочлена $\sigma(x)$ являются $x_1 = \alpha^{12}$, $x_2 = \alpha^9$, поэтому $X_1 = x_1^{-1} = \alpha^3 = \alpha_4$, $X_2 = x_2^{-1} = \alpha^6 = \alpha_7$. Следовательно, ошибки произошли на 4-й и 7-й позициях.

4. Так как код двоичный, то исходный кодовый вектор равен:

$$u = (1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1),$$

при этом длина LFSR совпадает с $\deg \sigma(x)$, поэтому на позиции с номером 0 ошибок нет.

Литература

[1] Блейхут Р. Теория и практика кодов, контролирующих ошибки. Пер. с англ. Москва: Мир, 1986. 576 с. URL: http://publ.lib.ru/ARCHIVES/B/BLEYHUT_Richard_E/_Bleyhut_R.E..html

[2] Huffman W. Cary. Fundamentals of Error-Correcting Codes. Cambridge: Cambridge University Press, 2003. 646 p. DOI: https://doi.org/10.1017/CBO9780511807077.

[3] Gao S. A new algorithm for decoding Reed–Solomon codes // In: Bhargava V.K., Poor H.V., Tarokh V., Yoon S. (eds.) Communications, Information and Network Security. The Springer International Series in Engineering and Computer Science (Communications and Information Theory), vol 712, pp. 55–68. Springer, Boston, MA. DOI: https://doi.org/10.1007/978-1-4757-3789-9_5.

[4] Massey J.L. Shift-register synthesis and BCH decoding // IEEE Transactions on Information Theory. 1969. Vol. IT. 15, № 1. P. 122–127. DOI: https://doi.org/10.1109/TIT.1969.1054260.

[5] Sugiyama Y. et al. A method for solving key equation for decoding Goppa codes // Information and Control. 1975. Vol. 27. Issue 1. P. 87–99. DOI: https://doi.org/10.1016/S0019-9958(75)90090-X

[6] Dornstetter J.L. On the equivalence Between Berlekamp's and Euclid's Algorithm // IEEE Trans. Inform. Theory. 1987. Vol. IT-33, № 3. P. 428–431.

[7] Рацеев С.М., Череватенко О.И. Об алгоритмах декодирования обобщенных кодов Рида — Соломона на случай ошибок и стираний // Вестник Самарского университета. Естественнонаучная серия. 2020. Т. 26, № 3. С. 17–29. DOI: https://doi.org/10.18287/2541-7525-2020-26-3-17-29.

[8] Рацеев С.М., Череватенко О.И. Об алгоритмах декодирования обобщенных кодов Рида — Соломона на случай ошибок и стираний. II // Вестник Самарского университета. Естественнонаучная серия. 2021. Т. 27, № 2 (в печати).

[9] Мак-Вильямс Ф.Дж., Слоэн Н.Дж.А. Теория кодов, исправляющих ошибки: пер. с англ. Москва: Связь, 1979. 744 с. URL: https://ru.djvu.online/file/jZGJCUDSUc4AZ.

Scientific article

DOI: 10.18287/2541-7525-2021-27-1-44-61

Submited: 10.01.2021
Revised: 17.02.2021
Accepted: 28.02.2021

S.M. Ratseev
Ulyanovsk State University, Ulyanovsk, Russian Federation
E-mail: ratseevsm@mail.ru. ORCID: https://orcid.org/0000-0003-4995-9418
A.D. Lavrinenko
Ulyanovsk State University, Ulyanovsk, Russian Federation
E-mail: anutalavrinenko@gmail.com. ORCID: https://orcid.org/0000-0003-3652-1097
E.A. Stepanova
Ulyanovsk State University, Ulyanovsk, Russian Federation
E-mail: kate_stepanova@bk.ru. ORCID: https://orcid.org/0000-0003-0276-1615

ON THE BERLEKAMP — MASSEY ALGORITHM AND ITS APPLICATION FOR DECODING ALGORITHMS

ABSTRACT

The paper is devoted to the Berlekamp — Masssey algorithm and its equivalent version based on the extended Euclidean algorithm. An optimized Berlekamp — Massey algorithm is also given for the case of a field of characteristic 2. The Berlekamp — Massey algorithm has a quadratic complexity and is used, for example, to solve systems of linear equations in which the matrix of the system is the Toeplitz matrix. In particular, such systems of equations appear in algorithms for the syndrome decoding of BCH codes, Reed — Solomon codes, generalized Reed — Solomon codes, and Goppa codes. Algorithms for decoding the listed codes based on the Berlekamp — Massey algorithm are given.

Key words: Berlekamp — Massey algorithm; extended Euclidean algorithm; Reed — Solomon codes; code decoding.

Citation. Ratseev S.M., Lavrinenko A.D., Stepanova E.A. On the Berlekamp — Massey algorithm and its application for decoding algorithms. *Vestnik Samarskogo universiteta. Estestvennonauchnaia seriia = Vestnik of Samara University. Natural Science Series*, 2021, vol. 27, no. 1, pp. 44–61. DOI: http://doi.org/10.18287/2541-7525-2021-27-1-44-61. (In Russ.)

Information about the conflict of interests: authors and reviewers declare no conflict of interests.

Sergey Mihaylovich Ratseev — Doctor of Physical and Mathematical Sciences, associate professor, Department of Information Security and Control Theory, Ulyanovsk State University, 42, Leo Tolstoy Street, Ulyanovsk, 432017, Russian Federation.

Anna Dmitrievna Lavrinenko — student of the Department of Information Security and Control Theory, Ulyanovsk State University, 42, Leo Tolstoy Street, Ulyanovsk, 432017, Russian Federation.

Ekaterina Alekseevna Stepanova — student of the Department of Information Security and Control Theory, Ulyanovsk State University, 42, Leo Tolstoy Street, Ulyanovsk, 432017, Russian Federation.

References

[1] Blahut Richard E. Theory and practice of error control codes. Translation from English. Moscow: Mir, 1986, 576 p. Available at: https://scask.ru/h_book_tpc.php. (In Russ.)

[2] Huffman W. Cary. Fundamentals of Error-Correcting Codes. Cambridge: Cambridge University Press, 2003, 646 p. DOI: https://doi.org/10.1017/CBO9780511807077.

[3] Gao S. A new algorithm for decoding Reed—Solomon codes. In: Bhargava V.K., Poor H.V., Tarokh V., Yoon S. (eds.) Communications, Information and Network Security. The Springer International Series in Engineering and Computer Science (Communications and Information Theory), vol. 712, pp. 55–68. Springer, Boston, MA. DOI: https://doi.org/10.1007/978-1-4757-3789-9_5.

[4] Massey J.L. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, 1969, vol. IT. 15, issue 1, pp. 122–127. DOI: https://doi.org/10.1109/TIT.1969.1054260.

[5] Sugiyama Y. et al. A method for solving key equation for decoding Goppa codes. *Information and Control*, 1975, vol. 27, issue 1, pp. 87–99. DOI: https://doi.org/10.1016/S0019-9958(75)90090-X.

[6] Dornstetter J.L. On the equivalence Between Berlekamp's and Euclid's Algorithm. *IEEE Transactions on Information Theory*, 1987, vol. IT-33, issue 3, pp. 428–431. DOI: https://doi.org/10.1109/TIT.1987.1057299.

[7] Ratseev S.M., Cherevatenko O.I. On decoding algorithms for generalized Reed-Solomon codes with errors and erasures. *Vestnik Samarskogo universiteta. Estestvennonauchnaia seriia = Vestnik of Samara University. Natural Science Series*, 2020, vol. 26, no. 3, pp. 17–29. DOI: https://doi.org/10.18287/2541-7525-2020-26-3-17-29. (In Russ.)

[8] Ratseev S.M., Cherevatenko O.I. On decoding algorithms for generalized Reed-Solomon codes with errors and erasures. II. *Vestnik Samarskogo universiteta. Estestvennonauchnaia seriia = Vestnik of Samara University. Natural Science Series*. 2021, vol. 27, no. 2. (Print, in Russ.)

[9] Mac Williams F.J., Sloane N.J.A. The theory of error correcting codes. Moscow: Svyaz', 1979, 744 p. Available at: https://ru.djvu.online/file/jZGJCUDSUc4AZ. (In Russ.)