Лекция №7

Декодирование кода RS

0. Код Рида-Соломона $1 \le k \le n$, $|\mathbb{F}| > n$, $S = \{d_1 .. d_n\} \subset \mathbb{F}$
поле

$$RS_{\mathbb{F},S}(n,k) = \{(p(d_1) .. p(d_n)) \in \mathbb{F}^n \mid p \in \mathbb{F}[x], \deg p \le k-1\}$$

Encode $(m \in \mathbb{F}^k)$: 1) $p(x) = m_0 + m_1 x .. + m_{k-1} x^{k-1}$

2) $p(d_1) . p(d_n)$.

$d(RS_{\mathbb{F},S}) = n-k+1$.

$RS_{\mathbb{F},S}(n,k)$ декодирует $\left\lfloor \frac{d-1}{2} \right\rfloor$ ошибок

$\tau := \left\lfloor \frac{d-1}{2} \right\rfloor = \left\lfloor \frac{n-k}{2} \right\rfloor$

I Алгоритм Welch-Berlekemp

$\exists y = (y_1 .. y_n)$ - полученное слово $(\notin RS)$, т.е.

$y_i \ne p(d_i)$ для максимум τ-индексов

из предыдущей лекции: Если мы знаем позиции ошибок, т.е. мн-во индексов $E = \{i \mid y_i \ne p(d_i)\}$, то мы можем восстановить $p(x)$ интерполяцией по точкам $y_j = p(d_j)$

$f(x) = p(x)$ (!)

определим многочлен $E(x) = \prod_{f(d_i) \ne y_i} (x - d_i)$, $\deg E(x) \le \tau$

↓ полином-локатор

для всех $1 \le i \le n$ и $E(x)$ выполняется: $E(d_i) \cdot y_i = E(d_i) \cdot f(d_i)$

(Если i-поз. ошибки, то оба выражения $=0$; иначе $y_i = f(d_i)$)

Положим $N(x) := E(x) \cdot f(x)$, $\deg N(x) \leq \tau + k - 1$

$$P(X, Y) := E(x) \cdot Y - N(X)$$

Заметим, что $P(d_i, y_i) = E(d_i) \cdot y_i - E(d_i) f(d_i) =$

$= E(d_i)(y_i - f(d_i)) = 0 \quad \forall i$

Алгоритм декодирования

Шаг 1 Найти ненулевой многочлен $Q(X, Y)$, т.ч.

- $Q(X, Y) = E_1(x) \cdot Y - N_1(x)$
- $\deg E_1(x) \leq \tau$, $\deg N_1(x) \leq \tau + k - 1$
- $Q(d_i, y_i) = 0 \quad \forall i$

Шаг 2 Вернуть $\dfrac{N_1(x)}{E_1(x)} = f(x)$

Корректность: 1) мн-н $Q(X, Y)$ существует, достаточно

найти $E_1(x) = E = \prod\limits_{f(d_i) \neq y_i} (x - d_i)$

$N_1(x) = E(x) \cdot f(x)$

2) Любое решение (E_1, N_1) удовлетворяет $\dfrac{N_1}{E_1} = f$

$\triangleleft$ Положим $R(x) = E_1 \cdot f - N_1$

- $\deg R(x) \leq \tau + k - 1$
- $R(x)$ имеет как min. $n - \tau$ корней, т.к. $\forall$ позиции

i без ошибки имеем $f(d_i) = y_i$ и

$R(d_i) = Q(d_i, y_i) = 0$

- при $n - \tau > \tau + k - 1$ $(*)$ $\Rightarrow R \equiv 0 \Leftrightarrow f = \dfrac{N_1}{E}$

(кол-во корней превосходит степень)

Н-ВО $(*)$ выполняется, т.к. по условию $\tau = \left\lfloor \frac{n-k}{2} \right\rfloor$

$$(*): \quad 2\tau < n-k+1$$
$$\tau < \frac{n-k+1}{2}$$

Сложность

$$\sum_{i=0}^{\tau} e_i x^i \qquad \sum_{i=0}^{\tau+k-1} n_i x^i$$

Для нахождения мн-ов E_1, N_1, используем $Q(d_i, y_i) = 0$

$\Rightarrow$ получаем систему из $\{(n_0 \ldots n_{\tau+k-1}), (e_0 \ldots e_\tau)\}$ неизвестных,

и из n уравнений $\qquad \tau+k+\tau = 2\tau+k \leq n-1 < n$.

Решаем систему методом Гаусса $O(n^3)$

Деление мн-ов: $O(n \lg^{O(1)} n)$ операций в $\mathbb{F}$

$\Big\} \; O(n^3)$

Пример

$$GF(5) = \langle 2 \rangle$$

$$S = \{1, 2, 4, 3\}$$

$$n=4, \quad k=2 \Rightarrow d = n-k+1 = 3 \Rightarrow \tau = 1$$

$$m = (4, 3) \rightarrow f(x) = 4+3x$$

$$c = Enc(m) = (f(1), f(2), f(4), f(3)) = (2, 0, 1, 3) \overset{+e}{\rightsquigarrow} \underset{Y}{\underset{\shortparallel}{(2, 1, 1, 3)}}$$

$\deg E_1(X) = 1$, $\quad E_1(X) = e_0 + \underset{1}{\underset{\shortparallel}{e_1}} X = e_0 + X$

$\deg N_1(X) = 2 \qquad N_1(X) = n_0 + n_1 X + n_2 X^2$

$$Q(d_i, y_i) = 0 \qquad Q(X, Y) = E_1(X)\,Y - N_1(X)$$

$$Q(d_1, y_1) = Q(1, 2) = E_1(1)\cdot 2 - N_1(1) = (e_0+1)\cdot 2 - n_0 - n_1 - n_2 = 0$$
$$Q(d_2, y_2) = Q(2, 1) = E_1(2)\cdot 1 - N_1(2) = (e_0+2) - n_0 - 2n_1 - 4n_2 = 0$$
$$Q(d_3, y_3) = Q(4, 1) = e_0 + 4 - n_0 - 4n_1 - n_2 = 0$$
$$Q(d_4, y_4) = Q(3, 3) = 3(e_0+3) - n_0 - 3n_1 - 4n_2 = 0$$

$$\Longleftrightarrow$$

$$\begin{cases} 2e_0 + 2 - n_0 - n_1 - n_2 = 0 \\ e_0 + 2 - n_0 - 2n_1 - 4n_2 = 0 \\ e_0 + 4 - n_0 - 4n_1 - n_2 = 0 \\ 3e_0 + 4 - n_0 - 3n_1 - 4n_2 = 0 \end{cases} \Leftrightarrow \begin{cases} e_0 = 3 \\ n_0 = 2 \\ n_1 = 3 \\ n_2 = 3 \end{cases} \Leftrightarrow$$

$$E = x + 3 = (x - 2)$$

↑ второй эл-т в мн-ве $S \Rightarrow$ ошибка во втором символе

$$N(x) = 2 + 3x + 3x^2$$

$$\begin{array}{r|l} 3x^2 + 3x + 2 & x + 3 \\ \hline -\; 3x^2 + 4x & 3x + 4 \\ \cline{1-1} 4x + 2 & \\ -\; 4x + 2 & \\ \cline{1-1} 0 & \end{array}$$

II Алгоритм Петерсона (Peterson Alg.)

Альтернативное опр-ие кода RS: для $1 \le k \le n$, $|\mathbb{F}| = q + 1$, $S = \{1, .., \alpha^{n-1}\}$

$$RS = \{ (c_0 \dots c_{n-1}) \in \mathbb{F}^n : c(\alpha) = c(\alpha^2) = \dots = c(\alpha^{n-k}) = 0 \}$$

$y = c + e$ - слово

для $\ell \in \{1 \dots n-k\}$ вычислим $S_\ell = \sum_{j=0}^{n-1} y_j \alpha^{\ell \cdot j} =$

$$= \underbrace{\sum c_j \alpha^{\ell j}}_{\substack{= c(\alpha^\ell) \\ = 0}} + \sum e_j \alpha^{\ell \cdot j} = \sum e_j \alpha^{\ell \cdot j}$$

Определим $S(x) := \sum_{\ell=1}^{n-k} S_\ell \cdot x^{\ell - 1}$ и

$E(x) = \prod_{j \in T} (1 - \alpha^j \cdot x)$, $T = \{ i \mid e_i \ne 0 \}$ - индексы ошибок

$$\deg E(x) = |T| \le \tau, \quad E(\alpha^{-j}) = 0 \quad \forall j \in T$$

Лемма

$$S(x) = \sum_{j \in T} e_j \alpha^j \left(\frac{1 - (\alpha^j x)^{n-k}}{1 - \alpha^j x} \right)$$

◁ на практике ▷

Следовательно,

$$E(x) \cdot S(x) = \sum_{j \in T} e_j \alpha^j \left(\frac{1 - (\alpha^j x)^{n-k}}{1 - \alpha^j x} \right) \cdot \prod_{i \in T} (1 - \alpha^i x)$$

$$= \sum_{j \in T} e_j \alpha^j \left(1 - (\alpha^j x)^{n-k} \right) \cdot \prod_{\substack{i \in T \\ i \ne j}} (1 - \alpha^i x) =$$

$$= \underbrace{\sum_{j \in T} e_j \alpha^j \prod_{\substack{i \in T \\ i \ne j}} (1 - \alpha^i x)}_{\Gamma(x)} - \sum_{j \in T} e_j \alpha^j \cdot (\alpha^j \boxed{x)^{n-k}} \cdot \prod_{\substack{i \in T \\ i \ne j}} (1 - \alpha^i x)$$

$$\Rightarrow \boxed{E(x) \cdot S(x) \equiv \Gamma(x) \mod x^{n-k}} \quad (1)$$

— Key Equation

Алгоритм Петерсона использует ур-ие (1) для нахождения коэфф-тов $E(x)$:

- в (1) мы знаем $S(x)$
- $\deg \Gamma(x) \le \tau - 1 \Rightarrow$ Коэфф-ты левой части $E(x) \cdot S(x)$ при x^j $\tau \le j \le n-k-1$ $\equiv 0$.

$\Rightarrow$ $n-k-\tau$ уравнений для $\underline{\tau}$ неизвестных коэфф-ов $E(x)$

Зная $E(x)$, найдём его корни $\Rightarrow$ найдём позиции ошибок

Т.к. система, полученная из (1), может иметь несколько решений, соответствующий $E_1(x)$ будет кратен $E(x)$.

Алгоритм

Шаг 1 Вычислить $S(x)$ // $O(n^6)$

Шаг 2 Составить из (1) сист-му ур-ий для e_i^1 $\Rightarrow$ // $O(n^3)$

$E_1(x) = 1 + \sum e_i^1 x^i$

Шаг 3 Найти корни $E_1(x)$, //$O(n^3)$

Положим корни:

$\alpha^{-i_1}, \dots, \alpha^{-i_e}$, $e \le \tau$

Шаг 4 Удалить позиции $i_1 \dots i_e$ из полученного слова, восст-ть сообщение по нестёртым позициям с помощью интерполяции.

Корректность Мн-н E_1, найденный на шаге 2, кратен $E(x)$ ($\{$корни $E(x)\} \subseteq \{$корни $E_1(x)\}$)

$\triangleleft$ $E(x) = \prod_{j \in T} (1 - \alpha^j x)$

$$E^{-1}(x) \bmod x^{n-k} = \prod_{j \in T} \left(1 + \alpha^j \cdot x + \dots + (\alpha^j x)^{n-k-1}\right)$$

В самом деле, $E^{-1}(x) \cdot E(x) = \prod_{j \in T} \sum_{i=0}^{n-k-1} (\alpha^j x)^i \cdot \prod_{j \in T} (1 - \alpha^j x)$

$$= \prod_{j \in T} \left(\frac{1 - (\alpha^j x)^{n-k}}{1 - \alpha^j x}\right) \cdot \prod_{j \in T} (1 - \alpha^j x) = \prod_{j \in T} \left(1 - (\alpha^j x)^{n-k}\right) \equiv 1 \bmod x^{n-k}$$

$\Leftarrow$ из (1): $\quad S(x) \equiv \Gamma(x)\cdot E^{-1}(x) \mod x^{n-k} \quad (2)$

$\exists\ \Gamma_1(x)$ – мн-н степени $\leq \tau - 1$, т.ч.

$$E_1(x)\cdot S(x) \equiv \Gamma_1(x) \mod x^{n-k} \qquad (3)$$

из (2) и (3) $\Rightarrow \quad E_1(x)\cdot \Gamma(x)\cdot E^{-1}(x) \equiv \Gamma_1(x) \mod x^{n-k} \Leftrightarrow$

$$\underbrace{E_1(x)\cdot \Gamma(x)}_{\deg \leq \tau+\tau-1 = 2\tau-1} \equiv \underbrace{E(x)\,\Gamma_1(x)}_{\tau+\tau-1 = 2\tau-1 \leq n-k-1} \mod x^{n-k}$$

$\Leftarrow$ mod редукция не имеет смысла, сравнение есть р-во

$$E_1(x)\cdot\Gamma(x) = E(x)\cdot\Gamma_1(x) \Rightarrow E(x) \mid E_1(x)\cdot\Gamma(x)$$

$\gcd(E(x), \Gamma(x)) = 1$ (т.к. $\{\alpha^{-j}, j\in T\}$ – все корни $E(x)$

$$\Gamma(\alpha^{-j}) = e_j \prod_{\substack{i\in T\\ i\neq j}} (\alpha^{j}-\alpha^{i}) \neq 0\}$$

$\Rightarrow E(x) \mid E_1(x) \quad \blacktriangleright$