数理解析研究所講究録
第1907巻 2014年 80-84

BCH符号におけるOrsini-Sala復号アルゴリズムの改良

伏里 拓也

TAKUYA FUSHISATO

金沢大学大学院 自然科学研究科 数物科学専攻

GRADUATE SCHOOL OF NATURAL SCIENCE AND TECHNOLOGY, KANAZAWA UNIVERSITY *

小原 功任

KATSUYOSHI OHARA

金沢大学 理工研究域 数物科学系

FACULTY OF MATHEMATICS AND PHYSICS, KANAZAWA UNIVERSITY †

Abstract

2005年, E. Orsini と M. Sala は, グレブナー基底の理論を用いた BCH 符号の復号アルゴリズムを考案した ([4]). Orsini-Sala 復号アルゴリズムは計算量が大きいという特徴があり, 大きな符号長で符号を構成するとき, 復号が困難となる. そこで, 本論文では有限体の代数構造をうまく利用することによって, Orsini-Sala 復号アルゴリズムの計算量を改善する方法を提案する. また計算機実験により, 実際に計算時間が改善され, また Orsini-Sala の方法では復号が困難であった場合においても, 復号が可能であることを示す.

1 BCH符号

以下, q は素冪, $\gcd(n,q)=1$ とする.

定義 1. 空集合でない $S \subset \{0,1,\ldots,n-1\}$ を **defining set** と呼び, 各 $i \in S$ に対して円分剰余類を $C_i = \{i, iq, \ldots, iq^{\ell-1}\}$ と表す. ここで, ℓ は $i \equiv iq^{\ell} \pmod{n}$ を満たす最小の自然数である.

定義 2. α を $\mathbb{F}_q$ 上 1 の原始 n 乗根とする. $S_C = \bigcup_{i\in S} C_i$ の中に最大で $\delta-1$ 個の連続する数 $b, b+1, \ldots, b+\delta-2$ が存在するとき, $g_C = \prod_{i\in S_C}(x-\alpha^i)$ を生成多項式とする巡回符号 $C = \langle g_C \rangle$ を**設計距離 δ の BCH 符号** という. $S = \{i_1, \ldots, i_r\}$ のとき, C のパリティ検査行列 H は

$$H = \begin{pmatrix} 1 & \alpha^{i_1} & \alpha^{2i_1} & \cdots & \alpha^{(n-1)i_1} \\ 1 & \alpha^{i_2} & \alpha^{2i_2} & \cdots & \alpha^{(n-1)i_2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^{i_r} & \alpha^{2i_r} & \cdots & \alpha^{(n-1)i_r} \end{pmatrix}$$

で定められる. また, 受信ベクトル $\boldsymbol{r} \in (\mathbb{F}_q)^n$ に対して $\boldsymbol{s} = H\boldsymbol{r} \in (\mathbb{F}_{q^m})^r$ を**シンドローム**という. ただし, $\mathbb{F}_{q^m}$ は $\mathbb{F}_q$ 上 x^n-1 の分解体とする.

*tfushis@gmail.com
†ohara@air.s.kanazawa-u.ac.jp

2 Orsini-Sala 復号アルゴリズム

以下, 簡単のため $q=2$ とする.

Hamming 重み $\mu \leq t$ の誤りベクトル $\boldsymbol{e}=(e_0,\ldots,e_{n-1})$ に対して, $e_{k_1},\ldots,e_{k_\mu} \neq 0$ のとき $k_1,\ldots,k_\mu$ を **誤りの位置**, $\alpha^{k_1},\ldots,\alpha^{k_\mu}$ を **error locator** という. また, $L_e(z)=\prod_{l=1}^{\mu}(z-\alpha^{k_l})$ を **error locator polynomial** という.

定義 3 ([4]). t を符号 C の誤り訂正能力とする. 変数を $X=(x_1,\ldots,x_r), Z=(z_t,\ldots,z_1)$ とする多項式

$$f_j=\sum_{u=1}^{t} z_u^{i_j}-x_j,\quad \sigma_j=x_j^{q^m}-x_j,\quad \eta_i=z_i^{n+1}-z_i,\quad \chi_{\tilde{i},i}=z_{\tilde{i}}\cdot z_i\cdot\sum_{u=0}^{n-1} z_{\tilde{i}}^{u} z_i^{n-1-u}$$

$(1\leq j\leq r, 1\leq i\leq t, 1\leq \tilde{i}<i)$ で生成されるイデアルを $I_C\subset \mathbb{F}_q[X,Z]$ で表す.

定義 3 の f_j は $H\boldsymbol{e}-\boldsymbol{s}=\boldsymbol{0}$, σ_j はシンドロームの各成分が $\mathbb{F}_{q^m}$ の元, η_i は error locator が 0 または α^{k_i} (1 の原始 n 乗根), $\chi_{\tilde{i},i}$ は $\alpha^{k_i}\neq\alpha^{k_j}$ $(i\neq j)$ により定められた多項式である.

定義 4 ([4]). 変数を $X=(x_1,\ldots,x_r)$, $Z=(z_t,\ldots,z_1)$ とする多項式 $\mathscr{L}_C\in\mathbb{F}_q[X,Z]$ が次の性質を満たすとき, $\mathscr{L}_C$ を符号 C の **general error locator polynomial** という.

(1) $\mathscr{L}_C\in\mathbb{F}_q[X,z_t]\subset\mathbb{F}_q[X,Z]$

(2) $\mathscr{L}_C(X,z_t)={z_t}^t+a_{t-1}(X){z_t}^{t-1}+\cdots+a_0(X)\quad (a_j(X)\in\mathbb{F}_q[X],\ 0\leq j\leq t-1)$

(3) シンドローム $\boldsymbol{s}$ について, $\mathscr{L}_C(\boldsymbol{s},z_t)\in\mathbb{F}_{q^m}[z_t]$ の根が $\alpha^{k_1},\ldots,\alpha^{k_\mu},\underbrace{0,\ldots,0}_{t-\mu}$

定理 1 ([4]). G を $I_C\subset\mathbb{F}_q[X,Z]$ の辞書式順序 $X\prec Z$ に関する被約グレブナー基底とする. このとき, G に general error locator polynomial が含まれる.

G は被約グレブナー基底であるから, z_t に関する次数が t である G の元は一意に定まる. よって, グレブナー基底計算により general error locator polynomial が求められる. また, 定義 4 より誤りの数 μ と error locator polynomial L_e を求める次のアルゴリズムを得る.

Algorithm 1 Orsini-Sala 復号アルゴリズム

Input: $\boldsymbol{s}=(s_1,\ldots,s_r)$, $\mathscr{L}_C(X,z_t)=z_t^t+a_{t-1}(X)z_t^{t-1}+\cdots+a_0(X)$

$\mu\leftarrow t$

while $a_{t-\mu}(\boldsymbol{s})=0$ **do**

$\mu\leftarrow\mu-1$

end while

$L_e(z_t)\leftarrow\dfrac{\mathscr{L}_C(\boldsymbol{s},z_t)}{z_t^{t-\mu}}$

Output: μ, $L_e(z_t)$

3 多項式イデアルの再構成と復号アルゴリズムの改良

Orsini-Sala 復号アルゴリズムにおいて最も計算量が大きいのは, イデアル I_C の辞書式順序に関する被約グレブナー基底の計算である. その他の計算の計算量は比較的小さいので無視できる. 一般に, グレブナー

基底計算の最悪計算量は, イデアルを生成する多項式系の最大全次数の 2 重指数となることが知られている ([1]).

グレブナー基底計算の計算量を減らすために, I_C の生成系の中で次数が最も高い $\sigma_j = x_j^{q^m} - x_j$ に着目した. σ_j はすべてのシンドロームの各成分が自明に満たす代数方程式である. しかし, σ_j を定めるとき, 誤りが t 個以下である条件が考慮されていない. これを考慮すると, σ_j を割り切る多項式で, t 個以下の誤りに対応するシンドロームの各成分を根として持つ多項式 τ_j が存在する. そこで, イデアル $I_C = \langle f_j, \sigma_j, \eta_i, \chi_{\tilde{i},i} \rangle$ において, 多項式 σ_j のかわりに τ_j をとり, イデアル $J_C = \langle f_j, \tau_j, \eta_i, \chi_{\tilde{i},i} \rangle$ を用いても同様の復号アルゴリズムが構成できるのではないかと考えられる. また, Orsini-Sala の I_C では 全次数の上限が q^m であるが, J_C では $\deg(\tau_j) \leq q^m$ となり, 計算量の改善が期待できる.

いま, 誤り訂正能力 $t = 2$ の BCH 符号を考える. さらに, $d \mid q^m - 1$ ならば $x^{d+1} - x \mid x^{q^m} - x$ が成り立つので, σ_j を割り切る多項式 τ_j を 2 項式と仮定する. シンドロームの定義式より τ_j は 0, α^i, $\alpha^i + \alpha^j$ $(0 \leq i, j \leq n-1)$ を根に持つ多項式であれば十分であることがわかる. これに関して, 次の補題を得ることができた.

補題 1. $n \in \mathbb{N}$, α を $\mathbb{F}_q$ 上 1 の原始 n 乗根, $m = [\mathbb{F}_q(\alpha) : \mathbb{F}_q]$ とする. ある $0 < \ell \leq \lfloor \frac{m}{2} \rfloor$ に対して, $n \mid (q^\ell + 1)$ を満たすとき,

$$\tau(x) = x^{n(q^\ell - 1)+1} - x$$

とおく. このとき, 任意の i, j $(0 \leq i, j \leq n-1)$ に対して次が成り立つ.

(1) $\tau(0) = 0$

(2) $\tau(\alpha^i) = 0$

(3) $\tau(\alpha^i + \alpha^j) = 0$

また, この $\tau = \tau_1 = \cdots = \tau_r$ を用いたイデアルから general error locator polynomial を求められることを示した.

定理 2. $q = 2$ とする. 符号長 n, 誤り訂正能力 $t = 2$ の BCH 符号を考える. n がある $0 < \ell \leq \lfloor \frac{m}{2} \rfloor$ に対して, $n \mid (q^\ell + 1)$ を満たすとき,

$$\tau_j = x_j^{n(q^\ell - 1)+1} - x_j \quad (1 \leq j \leq r)$$

とおく. このとき, イデアル $J_C = \langle f_j, \tau_j, \eta_i, \chi_{\tilde{i},i} \rangle$ の辞書式順序に関する被約グレブナー基底は general error locator polynomial を含む.

以下, この定理 2 を用いた具体例を述べるが, 計算には Core i7-3930K 3.20GHz, メモリ 32GB, Windows7 x64 の計算機で数式処理システム Risa/Asir を使用した.

例 1. 符号長 $n = 27$ $(\mathbb{F}_{q^m} = \mathbb{F}_{2^{18}})$, defining set $S = \{1, 9\}$ をとると, 設計距離 $\delta = 6$ の BCH 符号が得られる (誤り訂正能力 $t = 2$). イデアル I_C において,

$$\sigma_j = x_j^{262144} - x_j \quad (1 \leq j \leq r)$$

である. 一方, 定理 2 より $t \leq 2$ に対応するすべてのシンドロームの各成分を根として持つ多項式

$$\tau_j = x_j^{13798} - x_j \quad (1 \leq j \leq r)$$

が存在する ($13798 = 27 \cdot (2^9 - 1) + 1$). イデアル I_C の辞書式順序に関する被約グレブナー基底の計算には 50329.5 秒を要したが, イデアル J_C のグレブナー基底は 0.858006 秒で計算することができた. また, I_C と J_C から得られる general error locator polynomial は一致しており, 2 個以下の誤りを含んだ受信ベクトルを復号することができる.

いま, シンドロームの各成分を根に持つ多項式 τ を具体的に与えることができた. しかし, 条件に符号長 n に関する仮定や誤り訂正能力 $t = 2$ の仮定が必要である. そこで次は, n に関する条件を除くことを考えたい.

σ は $0, \alpha^i, \alpha^i + \alpha^j \in \mathbb{F}_{q^m}$ $(0 \le i, j \le n-1)$ を根に持つ多項式であれば十分であった. そこで, これらを根に持つ次数最小の多項式 τ' を σ のかわりに用いる方法を考えた.

$\theta \in \mathbb{F}_{q^m}$ を根に持つ次数最小の多項式は, θ の $\mathbb{F}_q$ 上最小多項式 $m_\theta(x) = (x-\theta)(x-\theta^q)\cdots(x-\theta^{q^{k-1}})$ である ([2]). ここで, k は $q^k \equiv 1 \pmod n$ を満たす最小の自然数である. 従って τ' としては, 各 $0, \alpha^i, \alpha^i + \alpha^j \in \mathbb{F}_{q^m}$ の $\mathbb{F}_q$ 上最小多項式の最小公倍元を用いればよいと考えられる. すなわち, τ' は $0, \alpha^i, \alpha^i + \alpha^j \in \mathbb{F}_{q^m}$ を含む代数多様体の定義多項式である. 次に, この τ' を用いた具体例を挙げる.

例 2. 符号長 $n = 27$, defining set $S = \{1, 9\}$ をとると, 設計距離 $\delta = 6$ の BCH 符号が得られる (誤り訂正能力 $t = 2$). 定義多項式 τ'_j は

$$\tau'_j = x_j^{352} + x_j^{325} + x_j^{298} + x_j^{271} + x_j^{109} + x_j^{82} + x_j^{55} + x_j \quad (1 \le j \le r)$$

と計算され, これを用いたイデアル $J'_C = \langle f_j, \tau'_j, \eta_i, \chi_{\tilde{i},i} \rangle$ のグレブナー基底は 0.0156001 秒で計算することができる.

例 1 と比較すると, 例 2 の定義多項式 τ' は, Orsini-Sala の σ や定理 2 の τ より次数が低く, グレブナー基底の計算量も改善された. しかし, 定義多項式の計算量は無視できず, 例 2 ($n = 27$, $t = 2$) の場合には計算に 2.27761 秒を要する. 従ってこの場合には, 定義多項式を用いると合計で 2.2932101 秒要するので, 定理 2 の τ を用いる方が高速に general error locator polynomial を計算できることとなる.

定義多項式を用いる方法では, $\mu \le t$ 個の誤りに対応するシンドロームの各成分が $\alpha^{b_1} + \cdots + \alpha^{b_\mu}$ $(0 \le b_1 < \cdots < b_\mu \le n-1)$ と表されることに注意すると, t 個以下のすべての誤りに対応するシンドロームの各成分を根にもつ定義多項式を計算することができるので, n に関する条件だけでなく t に関する条件を除くこともできる.

さらに, 定義多項式 τ' の次数は定理 2 の τ の次数以下であるので, τ を用いたイデアル J_C のグレブナー基底は計算できないが, τ' を用いたイデアル J'_C のグレブナー基底は計算できる場合もある.

例 3. 符号長 $n = 81$, defining set $S = \{1, 9\}$ をとると, 設計距離 $\delta = 6$ の BCH 符号が得られる (誤り訂正能力 $t = 2$). イデアル I_C において σ_j の次数は 18014398509481984, J_C において τ_j の次数は 10871635888 であるが, 次数が大きすぎるため, これらのグレブナー基底は計算できない. そこで, 定義多項式

$$\begin{aligned}\tau'_j = {} & x_j^{3241} + x_j^{3160} + x_j^{2998} + x_j^{2836} + x_j^{2755} + x_j^{2431} + x_j^{2269} + x_j^{2188} + x_j^{2107} + x_j^{2026} \\ & + x_j^{1945} + x_j^{1702} + x_j^{1540} + x_j^{1297} + x_j^{1135} + x_j^{973} + x_j^{730} + x_j^{568} + x_j^{487} + x_j^{406} + x_j^{325} \\ & + x_j^{244} + x_j^{163} + x_j \quad (1 \le j \le r)\end{aligned}$$

を用いると, イデアル $J'_C = \langle f_j, \tau'_j, \eta_i, \chi_{\tilde{i},i} \rangle$ のグレブナー基底は 0.48 秒で計算することができ, general error locator polynomial を

$$\begin{aligned}\mathscr{L}_C = {} & z_2^2 + x_1 z_2 + x_1^{74} x_2 + x_1^{3080} + x_1^{2999} + x_1^{2594} + x_1^{2270} + x_1^{2027} + x_1^{1784} + x_1^{1703} \\ & + x_1^{1298} + x_1^{1217} + x_1^{1055} + x_1^{893} + x_1^{812} + x_1^{650} + x_1^{488} + x_1^{407}\end{aligned}$$

と求めることができる. なお, 定義多項式の計算には 770.317 秒を要した.

本研究では $n < 100, t = 2$ の定義多項式, $n < 50, t = 3$ の定義多項式, さらに $n < 100, t = 2$ のときの general error locator polynomial を計算することができた ([6]).

4 まとめ

本論文では, Orsini-Sala 復号アルゴリズムの欠点であったグレブナー基底計算量の問題を改善する方法を提案した. 最後に, その結果をまとめる.

- Orsini-Sala の方法と比べてグレブナー基底の計算時間は, 例 1 の場合には約 60000 倍, 例 2 の場合には約 320000 倍速く計算することができ, 大きな改善が見られた.
- 例 3 のように Orsini-Sala の方法ではイデアルの生成系の次数が高すぎて general error locator polynomial が計算できない場合でも, 定義多項式を用いることにより計算できることがある.
- 定義多項式はほとんどの場合, σ と比べて次数を小さくすることに成功したが, C が原始 BCH 符号である $(n = q^m - 1)$ 場合や, m が小さい場合には $\tau' = \sigma$ となることがある ([6]).
- 定義多項式の計算には時間を要するが, 定義多項式を効率よく計算するアルゴリズムには着手していないので, 改善が見込まれる.
- 定義多項式の計算結果より, このすべての次数は n に関する関数になっていることがわかる. これは共通因子として x をくくり出すことによって確かめられる. これより, 定義多項式は数学的に求めることができるのではないかと考えられる.

[1] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, Third Edition, Cambridge University Press, 2013.

[2] R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*, Revised Edition, Cambridge University Press, 1994.

[3] T. Mora, E. Orsini and M. Sala, General error locator polynomials for binary cyclic codes with $t \leq 2$ and $n < 63$, IEEE Transaction on information theory **53** (2007), 1095-1107.

[4] E. Orsini and M. Sala, Correcting errors and erasures via the syndorome variety, Journal of Pure and Applied Algebra **200** (2005), 191-226.

[5] 野呂正行, 横山和弘, グレブナー基底の計算 基礎編 - 計算代数入門, 東京大学出版会, 2003.

[6] 伏里拓也, 多項式イデアルのグレブナー基底を用いた BCH 符号の復号アルゴリズム, 修士論文, 金沢大学, 2014.

[7] 三宅伸也, グレブナー基底による巡回符号の復号アルゴリズムについて, 修士論文, 神戸大学, 2012.