BCH 부호와 Peterson-Gorenstein-Zierler 디코딩에 관한 연구

인재휘, 안성현, 김동찬
국민대학교 정보보안암호수학과

{jhin0303, ashtree2901, dckim}@kookmin.ac.kr

A Study on BCH Code and Peterson-Gorenstein-Zierler Decoding

JaeHui In, SeongHyun An, Dong-Chan Kim
Kookmin Univ.

요 약

디지털 통신 시스템은 송신기, 채널, 수신기로 구성된다. 데이터는 송신기를 통해 신호로 변환되어 송신된다. 채널을 통과하여 수신된 신호는 수신기를 통해 정보로 변환된다. 수신자는 이 정보를 받게 된다. 이 때 채널을 통과하는 과정에서 오류가 발생할 수 있다. 수신자는 오류가 발생한 정보를 받으므로 오류를 정정해야 원본 데이터를 얻을 수 있다. 오류 정정 부호는 오류를 찾아내서 원래 값으로 복원할 수 있는 기술이다. 본 논문에서는 이동통신용 오류 정정 부호 중 하나인 BCH 부호와 BCH 부호 디코딩 알고리듬을 소개한다.

Ⅰ. 서 론

디지털 통신 시스템에서 데이터가 송신될 때 오류가 발생할 수 있다. 수신자는 송신자가 송신한 원본 데이터를 얻기 위해 오류를 정정해야 한다. 오류 정정 부호는 오류를 찾아내서 원본 데이터로 복원시키는 기술이다. 따라서 오류 정정 부호는 디지털 통신 시스템을 구성하는 필수적인 요소이다. 이동통신용 오류 정정 부호에는 BCH/RS 부호, LDPC 부호, Turbo 부호 등이 있다.

BCH 부호는 광통신 분야에서 1 세대, 2 세대 순방향 오류 정정기술로써 사용되며 SLC(Single Level Cell)플래시 메모리 기술에 적용되는 부호다[5,6]. 본 논문에서는 R. C. Bose, D. K. Ray-Chaudhuri 와 A. Hocquenghem 가 고안한 BCH 부호에 대해 소개한다[1,2]. II 장에서는 기호에 대해 설명하고 III 장에서는 BCH 부호를 살펴본다. IV 장에서는 BCH 부호의 최소거리에 대한 정리를 증명하고 V 장에서는 BCH 부호 디코딩 알고리듬, VI 장에서는 Peterson-Gorenstein-Zierler 디코딩 알고리듬에 대해 설명한다.

Ⅱ. 기호 및 정의

본 논문에서는 다음의 기호 및 정의를 사용한다.

q	소수 p의 거듭제곱
m	$q^m \equiv 1(mod\, n)$ 을 만족하는 가장 작은 정수
F_q	원소의 개수가 q개인 유한체
$F_q[x]$	F_q의 원소를 계수로 가지는 다항식환
F_q^n	F_q에서 정의된 n차원 벡터공간
해밍무게	벡터 $u \in \mathbb{F}_q^n$에 대해 0이 아닌 $u_i\,(0 \le i \le n-1)$의 개수
$LCM(A,B)$	A와 B의 최소 공배수
$\mathcal{H}$	선형부호 $\mathcal{C} = [n,k]_q$의 패리티검사행렬
신드롬	수신된 벡터 r에 대한 $r\mathcal{H}^T$의 값
d	최소 해밍거리
순환부호	부호 $\mathcal{C}$가 유한체 F_q에서 길이가 n인 선형부호일 때, $\mathcal{C}$의 모든 부호어 $c = (c_0, c_1, \dots, c_{n-1})$에 대하여 F_q에 있는 $(c_{n-1}, c_0, \dots, c_{n-2})$도 $\mathcal{C}$의 부호어인 부호 $\mathcal{C}$
$c(x)$	부호어 c를 계수로 하는 다항식 예시) $c = (c_0\, c_1\, \dots\, c_{n-1})$ $c(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$
$g(x)$	순환부호의 생성 다항식
$h(x)$	$c(x)h(x) \equiv 0\,(mod\, g(x))$을 만족시키는 패리티검사다항식
$e(x)$	오류벡터 e를 계수로 하는 다항식 예시) $e = (e_0 e_1\, \dots\, e_{n-1})$ $e(x) = e_0 + e_1 x + \cdots + e_{n-1}x^{n-1}$

Ⅲ. BCH 부호

BCH 부호는 다음을 사용한다.

- α : F_{q^m}에서 위수가 n인 원소
- b : 임의의 양의 정수
- δ : 설계된 거리, $2 \le \delta \le n$

정의 1. 길이가 n인 (일반적)BCH 부호 $\mathcal{C}$는 다음 행렬 $\mathcal{H}$가 패리티검사행렬인 순환부호이다.

$$\mathcal{H} = \begin{bmatrix} 1 & \alpha^b & \alpha^{2b} & \cdots & \alpha^{(n-1)b} \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \cdots & \alpha^{(n-1)(b+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{b+\delta-2} & \alpha^{2(b+\delta-2)} & \cdots & \alpha^{(n-1)(b+\delta-2)} \end{bmatrix}.$$

α^i을 근으로 가지는 최소다항식을 $\phi_i(x)(\in F_q[x])$라 하자. BCH 부호의 생성다항식은 다음과 같다.

$$g(x) = LCM\big(\phi_b(x), \phi_{b+1}(x), \dots, \phi_{b+\delta-2}(x)\big).$$

이 때 $b = 1$이면 Narrow-sense BCH 부호, $n = q^m - 1$이면 Primitive BCH 부호, 두 조건 모두 만족한다면 Primitive Narrow-sense BCH 부호라 한다. $n = q - 1$이면 Reed-Solomon 부호다.

Ⅳ. BCH 부호의 최소 거리

정리 1. BCH 부호 $\mathcal{C}$의 패리티검사행렬이 $\delta - 1$개의 행을 가지면 최소거리는 δ보다 크거나 같다.

증명. BCH 부호 $\mathcal{C}$의 패리티검사행렬의 임의의 $\delta - 1$개의 열이 일차 독립일 때, $\mathcal{C}$의 최소거리는 δ보다 크다. BCH 부호 $\mathcal{C}$의 패리티검사행렬에서 $l_1, l_2, \dots, l_{\delta-1}$번째 열을 선택하여 생성한 행렬의 행렬식을 계산하면 다음과 같다.

$$det\begin{bmatrix} \alpha^{l_1 b} & \alpha^{l_2 b} & \cdots & \alpha^{l_{\delta-1} b} \\ \alpha^{l_1(b+1)} & \alpha^{l_2(b+1)} & \cdots & \alpha^{l_{\delta-1}(b+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{l_1(b+\delta-2)} & \alpha^{l_2(b+\delta-2)} & \cdots & \alpha^{l_{\delta-1}(b+\delta-2)} \end{bmatrix} =$$

$$(\alpha^{l_1 b}\,\alpha^{l_2 b}\cdots\alpha^{l_{\delta-1} b}) det\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{l_1} & \alpha^{l_2} & \cdots & \alpha^{l_{\delta-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{l_1(\delta-2)} & \alpha^{l_2(\delta-2)} & \cdots & \alpha^{l_{\delta-1}(\delta-2)} \end{bmatrix}.$$

위의 식에서 서로 다른 $\alpha^{l_\tau}(1 \le \tau \le \delta-1)$는 0이 아니고 두번째 행렬이 $Vandermonde$ 행렬이기에 0이 될 수 없다. 그러므로 BCH 부호 $\mathcal{C}$의 패리티검사행렬은 임의로 뽑은 $\delta-1$개의 열은 일차독립이다. ■

V. BCH 부호 디코딩 알고리듬

BCH 부호 디코딩 알고리듬은 다음을 사용한다.

- k_i : 오류가 생긴 위치
- L_i : 오류벡터의 위치 정보 α^{k_i}
- Z_i : 오류벡터의 크기 정보 e_{k_i}

BCH 부호 디코딩 알고리듬은 다음 단계로 구성된다.

(단계 1) 신드롬 S_i를 계산한다.

(단계 2) 오류벡터의 해밍무게 γ을 계산한다.

(단계 3) 오류벡터의 위치정보 L_i을 계산한다.

(단계 4) 오류벡터의 크기정보 Z_i을 계산한다.

(단계 5) $y(x) - e(x)$를 통해 $c(x)$를 복원한 후 부호어인지 확인한다.

복원한 c가 부호어가 아니라면 복원할 수 없다.

VI. Peterson-Gorenstein-Zierler 디코딩 알고리듬

Peterson-Gorenstein_Zierler 디코딩 알고리듬은 BCH 부호 디코딩 알고리듬의 (단계 2)을 해결하는 알고리듬이다. 정정할 수 있는 오류벡터의 해밍무게가 최대 t 일 때 오류벡터의 해밍무게 γ는 다음 방식으로 구한다.

$e(x)$를 해밍무게가 $\gamma(\le t)$인 오류벡터라 하자. $c(x)$는 BCH 부호의 성질에 의해 $c(\alpha^i) = 0$이다. 따라서 $y(\alpha^i) = c(\alpha^i) + e(\alpha^i) = e(\alpha^i)$이다. $y(x)$의 i번째 신드롬 값은 다음과 같다.

$$S_i = y(\alpha^i) = e_{k_1}(\alpha^{k_1})^i + e_{k_2}(\alpha^{k_2})^i + \cdots + e_{k_\gamma}(\alpha^{k_\gamma})^i.$$

이를 L_i와 Z_i을 이용하여 단순화시킬 수 있다.

$$S_i = Z_1 L_1^i + Z_2 L_2^i + \cdots + Z_\gamma L_\gamma^i = \sum_{j=1}^{\gamma} Z_j L_j^i, (1 \le i \le 2t).$$

오류벡터 위치 다항식 $\sigma(x)$은 다음과 같이 정의하자.

$$\sigma(x) = (1 - xL_1)(1 - xL_2)\ldots(1 - xL_\gamma) = 1 + \sum_{i=1}^{\gamma} \sigma_i x^i.$$

이 때 다음이 성립된다.

$$\sigma(L_j^{-1}) = 1 + \sigma_1 L_j^{-1} + \sigma_2 L_j^{-2} + \cdots + \sigma_\gamma L_j^{-\gamma} = 0, (1 \le j \le \gamma).$$

양 변에 $Z_j L_j^{i+\gamma}$ 을 곱하면

$$Z_j L_j^{i+\gamma} + \sigma_1 Z_j L_j^{i+\gamma-1} + \sigma_2 Z_j L_j^{i+\gamma-2} + \cdots + \sigma_\gamma Z_j L_j^i = 0$$

이다. $1 \le j \le \gamma$에 대한 위의 식을 모두 더하면

$$\sum_{j=1}^{\gamma} Z_j L_j^{i+\gamma} + \sigma_1 \sum_{j=1}^{\gamma} Z_j L_j^{i+\gamma-1} + \cdots + \sigma_\gamma \sum_{j=1}^{\gamma} Z_j L_j^i = 0.$$

이다. $S_i = \sum_{j=1}^{\gamma} Z_j L_j^i, (1 \le i \le 2t)$ 이므로 다음의 식을 구할 수 있다.

$$S_{i+\gamma} + \sigma_1 S_{i+\gamma-1} + \cdots + \sigma_\gamma S_i = 0.$$

이를 아래의 행렬식으로 표현할 수 있다.

$$\begin{bmatrix} S_1 & S_2 & \cdots & S_{\gamma-1} & S_\gamma \\ S_2 & S_3 & \cdots & S_\gamma & S_{\gamma+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ S_\gamma & S_{\gamma+1} & \cdots & S_{2\gamma-2} & S_{2\gamma-1} \end{bmatrix} \begin{bmatrix} \sigma_\gamma \\ \sigma_{\gamma-1} \\ \vdots \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} -S_{\gamma+1} \\ -S_{\gamma+2} \\ \vdots \\ -S_{2\gamma} \end{bmatrix}.$$

이 행렬식을 계산하면 σ_k을 구할 수 있다. 오류의 해밍무게 γ는 다음 행렬의 가역 여부를 판별하여 구할 수 있다.

$$M_\omega = \begin{bmatrix} S_1 & S_2 & \cdots & S_\omega \\ S_2 & S_3 & \cdots & S_{\omega+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_\omega & S_{\omega+1} & \cdots & S_{2\omega-1} \end{bmatrix}, (\omega \in \mathbb{Z}^+).$$

M_ω 행렬은 다음의 A_ω 행렬, B_ω 행렬, A_ω^T 행렬로 인수분해가 가능하다.

$$A_\omega = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ L_1 & L_2 & \cdots & L_\omega \\ \vdots & \vdots & \ddots & \vdots \\ L_1^{\omega-1} & L_2^{\omega-1} & \cdots & L_\omega^{\omega-1} \end{bmatrix}, B_\omega = \begin{bmatrix} Z_1 L_1 & 0 & \cdots & 0 \\ 0 & Z_2 L_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Z_\omega L_\omega \end{bmatrix}.$$

$$M_\omega = A_\omega B_\omega A_\omega^T.$$

$\omega > \gamma$ 일 때 B_ω의 대각성분에 0 이 포함되어 $\det(B_\omega) = 0$ 이므로 M_ω 은 비가역이다. $\omega \le \gamma$ 일 때 M_ω 행렬은 가역행렬이 된다. 따라서 처음에 ω를 t로 잡은 뒤 ω의 값을 줄여가면서 M_ω 행렬을 가역으로 만드는 최대 정수를 오류벡터의 해밍무게 γ로 결정한다. 해밍무게 γ을 구하는 알고리듬의 유사부호는 다음과 같다.

입력 : $S_i(1 \le i \le 2t),\ t$
출력 : ω

1: $\omega = t + 1$
2: ***repeat***
3: $\quad \omega \leftarrow \omega - 1$
4: ***until*** $\det(M_\omega) \ne 0$
5: ***return*** ω

벡터의 해밍무게 γ 을 알면 $\sigma(x)$의 근을 구할 수 있다. $\sigma(x)$의 근은 $L_j^{-1}(1 \le j \le \gamma)$이므로 이를 통해 (단계 3)을 해결할 수 있다. (단계 4)는 Forney 알고리듬을 통해 구할 수 있다. 만약 복원하려는 부호어 c가 이진 BCH 부호의 부호어일 경우 Z_i는 1이기에 생략할 수 있다.

VII. 결론

본 논문에서는 오류 정정 부호 중 하나인 BCH 부호의 정의와 성질, 디코딩 알고리듬에 대해 알아보았다. BCH 부호는 파라미터에 따라 Primitive BCH 부호, Narrow-sense BCH 부호, Reed-Solomon 부호로 구분된다. BCH 부호 디코딩 알고리듬은 Peterson-Gorenstein-Zierler 알고리듬 외에 이진 BCH 부호에서 사용하는 Berlekamp-Massey 알고리듬, 확장 유클리드 알고리듬을 사용한 디코딩 등이 있다. 해당 알고리듬에 대해서는 추후 연구계획이다.

참 고 문 헌

[1] R. C. Bose, and D. K. Ray-Chaudhuri, "On A Class of Error Correcting Binary Group Codes," Information and Control, 3 (1): 68-79, March 1960.

[2] A. Hocquenghem, "Codes correcteurs d'erreurs," Chiffres (in French), Paris, 2: 147-156, September 1959.

[3] W. C. Huffman and V. Pless, "Fundamentals of Error-Correcting Codes," Cambridge, MA: University Press, 2003.

[4] I.S. Reed and X. Chen, "Error-Control Coding for Data Networks," Kluwer Academic, 1999.

[5] 이기준 외 3 인, "NAND Flash 메모리 저장 장치에서의 Error Control Code 응용", 2015.1

[6] 이한호, 최창석, "광통신용 FEC 설계", 기술동향컬럼, 2009.12.