EC500
Design of Secure and Reliable Hardware

Lecture 12

Mark Karpovsky

1 Binary BCH Codes Correcting $\boldsymbol{l}$ errors

$n = 2^m - 1$, $k = 2^m - lm - 1$, $d = 2l + 1$ (Generalization of cyclic Hamming codes and double error correcting BCH).

Construct $GF(2^m)$ and let $\alpha \in GF(2^m)$ be primitive such that $p(\alpha) = 0$ and $\deg p(x) = m$. Take

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \cdots & \alpha^{3(n-1)} \\ 1 & \alpha^5 & \alpha^{10} & \alpha^{15} & \cdots & \alpha^{5(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{2l-1} & \alpha^{2(2l-1)} & \alpha^{3(2l-1)} & \cdots & \alpha^{(n-1)(2l-1)} \end{bmatrix}.$$

<u>Example</u>
$m = 8$, $n = 2^8 - 1 = 255$, $l = 5$, $d = 11$, $k = 255 - 5 \cdot 8 = 215$

For l-error correcting BCH codes, we have $R = \frac{k}{n} = \frac{2^m - l \cdot m - 1}{2^m - 1} = 1 - \frac{l \cdot m}{2^m - 1}$, so for a small l, the $R \to 1$.

Let C is l-error correcting BCH and $v \in C \to \begin{cases} v(\alpha) = 0 \\ v(\alpha^3) = 0 \\ v(\alpha^5) = 0 \\ \vdots \\ v(\alpha^{2l-1}) = 0 \end{cases}$. Thus C contains all polynomials with l different roots: α, α^3, α^5, $\cdots$, $\alpha^{2l-1} \to C$ is cyclic.

1.1 Decoding of BCH Codes

<u>Example</u>
$l = 3$ $(d = 7)$

$S = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} = He = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \cdots & \alpha^{3(n-1)} \\ 1 & \alpha^5 & \alpha^{10} & \alpha^{15} & \cdots & \alpha^{5(n-1)} \end{bmatrix} e$, if we let $e = (0 \cdots \underset{i}{010} \cdots \underset{j}{010} \cdots \underset{s}{010} \cdots 0)$ where errors are in bits i, j, and s, then $\begin{cases} S_1 = \alpha^i + \alpha^j + \alpha^s \\ S_2 = \alpha^{3i} + \alpha^{3j} + \alpha^{3s} \\ S_3 = \alpha^{5i} + \alpha^{5j} + \alpha^{5s} \end{cases}$. We can denote $x = \alpha^i$, $y = \alpha^j$, and $z = \alpha^s$ and rewrite as $\begin{cases} S_1 = x + y + z \\ S_2 = x^3 + y^3 + z^3 \\ S_3 = x^5 + y^5 + z^5 \end{cases}$. This is the system of three equations with three unknowns x, y, z and has a unique solution. Decoding is complex. $S_1, S_2, S_3 \xrightarrow[\text{difficult}]{} x, y, z \xrightarrow[\text{easy}]{} i, j, s$.

1.2 Extended BCH Codes

Let H_{BCH} be a check matrix for an $n = 2^m - 1,\ k = 2^m - l \cdot m - 1,\ d = 2l + 1,\ l$-error correcting code. We can add an overall parity to H_{BCH} to get

$$H = \underbrace{\begin{bmatrix} & & H_{BCH} & & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}}_{n+1} \Bigg\} l \cdot m + 1,$$

where H is the check matrix for an $n = 2^m$, $k = 2^m - l \cdot m - 1$, $d = 2l + 2$ extended BCH code. Furthermore, by extending or shortening, BCH codes with any distance and any length can be constructed.

<u>Example</u>

Construct a code with length $n = 24$ and distance $d = 6$.

1) Take $m = 5$ and construct a BCH code with $n = 2^5 - 1 = 31$ and $k = 2^5 - 1 - 2 \cdot 5 = 21$ ($l = 2$) to get a code with $d_{BCH} = 5$.
2) Extend the constructed BCH code to get $n = 32$, $k = 21$, $d = 6$.
3) Shorten this code by deleting $i = 32 - 24 = 8$ columns in the check matrix. Then finally we have $n = 24$, $k = 13$, $d = 6$.

∴ Codes obtained by extending and shortening BCH codes are good for small $\frac{d}{n}$. (e.g. $d = 5 \rightarrow l = 2$)

2 Double Error Correcting Cyclic Codes (BCH Codes)

$n = 2^m - 1$, $k = 2^m - 2m - 1$, $d = 5$.

<u>Binary Case:</u> $(q = 2)$

Consider the field $GF(2^m)$, construct a code of length $n = 2^m - 1$ with the check matrix $H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \cdots & \alpha^{3(n-1)} \end{bmatrix}$, where α is a primitive element in $GF(2^m)$.

<u>Example</u>

$m = 4$, $p(x) = x^4 + x + 1$

Binary				Exponential
0	0	0	0	0
0	0	0	1	1
0	0	1	0	α
0	0	1	1	α^4
0	1	0	0	α^2
0	1	0	1	α^8
0	1	1	0	α^5
0	1	1	1	α^{10}
1	0	0	0	α^3
1	0	0	1	α^{14}
1	0	1	0	α^9
1	0	1	1	α^7
1	1	0	0	α^6
1	1	0	1	α^{13}
1	1	1	0	α^{11}
1	1	1	1	α^{12}
α^3	α^2	α	1	

$\alpha^4 = \alpha + 1$

$\alpha^5 = \alpha^2 + \alpha$

$\alpha^6 = \alpha^3 + \alpha^2$

$\alpha^7 = \alpha^4 + \alpha^3 = \alpha^3 + \alpha + 1$

$\alpha^8 = \alpha^2 + 1$

$\alpha^9 = \alpha^3 + \alpha$

$\alpha^{10} = \alpha^4 + \alpha^2 = \alpha^2 + \alpha + 1$

$\alpha^{11} = \alpha^3 + \alpha^2 + \alpha$

$\alpha^{12} = \alpha^4 + \alpha^3 + \alpha^2 = \alpha^3 + \alpha^2 + \alpha + 1$

$\alpha^{13} = \alpha^4 + \alpha^3 + \alpha^2 + \alpha = \alpha^3 + \alpha^2 + 1$

$\alpha^{14} = \alpha^4 + \alpha^3 + \alpha = \alpha^3 + 1$

$\alpha^{15} = \alpha^4 + \alpha = 1$

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

$$= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & \vdots & 0 & 1 & 1 & 0 & 1 & \vdots & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & \vdots & 1 & 1 & 0 & 1 & 0 & \vdots & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & \vdots & 1 & 0 & 1 & 0 & 1 & \vdots & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & \vdots & 0 & 0 & 1 & 1 & 0 & \vdots & 1 & 0 & 1 & 1 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 1 & 1 & 1 & 1 & \vdots & 0 & 1 & 1 & 1 & 1 & \vdots & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & \vdots & 0 & 0 & 1 & 0 & 1 & \vdots & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & \vdots & 0 & 0 & 0 & 1 & 1 & \vdots & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & \vdots & 1 & 0 & 0 & 0 & 1 & \vdots & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

C is a $(15, 2^7, 5)$ BCH code. $v \in C \rightarrow Hv = 0 = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \cdots & \alpha^{12} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ \vdots \\ v_{15} \end{bmatrix}$

$\rightarrow \begin{cases} v_1 + \alpha v_2 + \alpha^2 v_3 + \alpha^3 v_4 + \cdots + \alpha^{14} v_{15} = 0 \\ v_1 + \alpha^3 v_2 + \alpha^6 v_3 + \alpha^9 v_4 + \cdots + \alpha^{12} v_{15} = 0 \end{cases}$, thus $v(x) = v_1 + x v_2 + x^2 v_3 + x^3 v_4 + \cdots + x^{14} v_{15}$ and $\begin{cases} v(\alpha) = 0 \\ v(\alpha^3) = 0 \end{cases}$ and finally we get $v \in C \leftrightarrow v(\alpha) = 0$ and $v(\alpha^3) = 0$. BCH code consists of all polynomials with roots α and α^3.

Let $v \in C$, consider ω where $\omega(x) = v(x)Q(x)$ for any $Q(x)$. Then $\omega(\alpha) = v(\alpha)Q(\alpha) = 0$ and $\omega(\alpha^3) = v(\alpha^3)Q(\alpha^3) = 0 \rightarrow \omega \in C$. If $v \in C$, any ω such that $\omega(x) = v(x)Q(x)$ belongs to C. Thus C is cyclic.

2.1 Decoding DEC BCH Codes with $d = 5$

Let $\tilde{v} = v + e$ and $v \in C$. $S = H(v + e) = Hv + He = He$. $H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \cdots & \alpha^{3(n-1)} \end{bmatrix}$.

1. For single errors
   $e = (00 \cdots 0\overset{i}{1}0 \cdots 00)$ - bit i is distorted. Then $S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} \alpha^i \\ \alpha^{3i} \end{bmatrix} \rightarrow \begin{matrix} S_1 = \alpha^i \\ S_2 = \alpha^{3i} \end{matrix}$. Thus a single error occur $(l = 1)$ iff ${S_1}^3 = S_2$ and the error is in bit i for $S_1 = \alpha^i$. For the previous example of $(15, 2^7, 5)$ BCH code, if $S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ \cdots \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha^7 \\ \cdots \\ \alpha^6 \end{bmatrix} \rightarrow \begin{matrix} S_1 = \alpha^7 \\ S_2 = \alpha^6 \end{matrix}$, since $(\alpha^7)^3 = \alpha^{21} = \alpha^6$ $(\alpha^{15} = \alpha^0)$, we have that bit 7 is distorted.

2. For double errors $(l = 2)$
   $e = (00 \cdots 0\overset{i}{1}0 \cdots 0\overset{j}{1}0 \cdots 00)$, $S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} \alpha^i + \alpha^j \\ \alpha^{3i} + \alpha^{3j} \end{bmatrix}$. If $(S_1)^3 \neq S_2$, then $l \neq 1$. We can denote α^i as y and α^j as z to get the following system of two equations with two unknowns y and $z \rightarrow$ $\begin{cases} y + z = S_1 \\ y^3 + z^3 = S_2 \end{cases}$. If there are two errors, this system is solvable for y and z. Thus, if we know S_1 and S_2, we can compute y and z and then the locations of errors i and j. However, the decoding procedure is complex, which is the main disadvantage of BCH codes.

3 Binary BCH Codes (revisited)

Let $n = 2^m - 1$, consider $GF(2^m)$ and α primitive in $GF(2^m)$. $p(x)$ is the primitive generating polynomial for $GF(2^m)$ and for $\alpha \in GF(2^m)$, $p(\alpha) = 0$, and $\deg p(x) = m$. Check matrix for a $\left(2^m - 1, 2^{2^m - l\cdot m - 1}, 2l + 1\right)$ cyclic BCH code C has the form $H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{3(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{(2l-1)} & \alpha^{2(2l-1)} & \cdots & \alpha^{(2l-1)(n-1)} \end{bmatrix}$ where $r = l \cdot m$. C consists of all polynomials $v(x) = v_0 + v_1 x + \cdots + v_{n-1} x^{n-1}$ $(v_i \in \{0,1\})$ such that $v(\alpha) = v(\alpha^3) = v(\alpha^5) = \cdots = v(\alpha^{2l-1}) = 0$. Take note that if a_1, a_2, $\cdots$, $a_s \in GF(2^m)$, then

$$(a_1 + a_2 + \cdots + a_s)^2 = {a_1}^2 + {a_2}^2 + \cdots + {a_s}^2 \quad (*)$$

Thus $v(\alpha^2) = v_0 + v_1\alpha^2 + v_2\alpha^4 + \cdots + v_{n-1}x^{2(n-1)} = (v_0 + v_1\alpha + v_2\alpha^2 + \cdots + v_{n-1}\alpha^{n-1})^2 = 0$ since $v_i = {v_i}^2$. Similarly, if $v \in C$ where C is a $\left(2^m - 1, 2^{2^m - l\cdot m - 1}, 2l + 1\right)$ BCH code, then

$$v(\alpha^2) = v(\alpha^4) = \cdots = v(\alpha^{2l}) = 0 \quad (1)$$

Thus $v \in C \leftrightarrow v\left(\alpha^i\right) = 0$ $(i = 1, \cdots, 2l)$ and $d = 2l + 1$. Conditions (1) should <u>not</u> be verified for computing syndrome wince they follow automatically for binary BCH from $(*)$.