Cryptanalysis

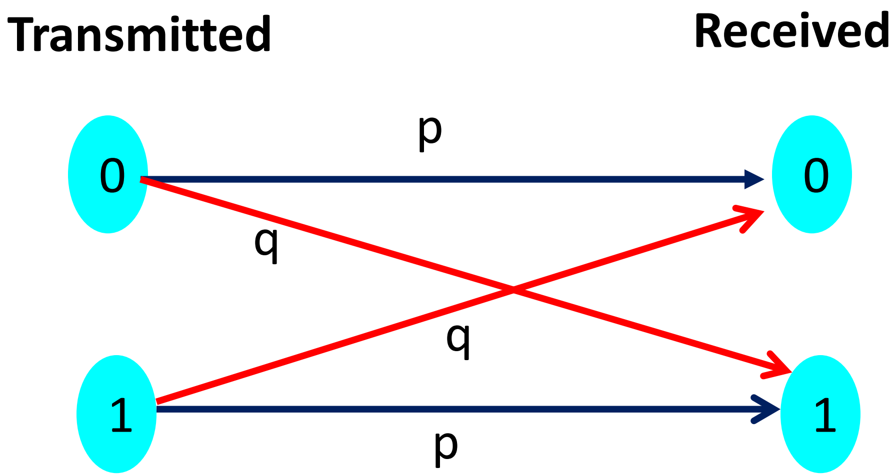
Lecture 10 : Error Correcting Codes

John Manferdelli
jmanfer@microsoft.com
JohnManferdelli@hotmail.com

jlm20090102

Binary symmetric channel (BSC)

- Each bit transmitted has an independent chance of being received correctly with probability p and incorrectly received with probability q=1-p.



- Can we transmit m bits more reliably over this channel if we have spare bandwidth?

Error Detection

- Suppose we want to transmit 7 bits with very high confidence over a binary symmetric channel. Even if p>.99, we occasionally will make a mistake.
- We can add an eight bit, a check sum, which makes any valid eight bit message have an even number of 1's.
- We can thus detect a single bit transmission error. Now the probability of a relying on a "bad" message is $P_{error}=1-(p^8+8p^7(1-p))$ instead of $P_{error}=1-p^8$. If p=.99, P_{error} drops from about 7% to .3%.
- This allows us to detect an error and hopefully have the transmitter resend the garbled packet.
- Suppose we want to avoid retransmission?

Error Correction

- We can turn these "parity checks" which enable error *detection* to error *correction* codes as follows. Suppose we want to transmit $b_1b_2b_3b_4$. Arrange the bits in a 2 x 2 rectangle:

b_1	b_2	$c_1=b_1+b_2$
b_3	b_4	$c_2=b_3+b_4$
$c_3=b_1+b_3$	$c_4=b_2+b_4$	$c_5=b_1+b_2+b_3+b_4$

- We transmit $b_1b_2b_3b_4c_1c_2c_3c_4c_5$.
- The receiver can detect any single error and locate its position.

- Another simple "encoding scheme" that corrects errors is the following. We can transmit each bit three times and interpret the transmission as the majority vote. Now the chance of correct reception is $P_{correct}=p^3+3p^2q>p$ and the chance of error is $P_{error}=3pq^2+q^3<q$. For $p=.99$, $P_{error}= 0.000298$ and $P_{correct}= .999702$.

Codewords and Hamming distance

- To correct errors in a message "block," we increase the number of bits transmitted per block. The systematic scheme to do this is called a code, C.
-
- If there are M valid messages per block (often $M=2^m$) and we transmit $n>\lg(M)$ bits per block, the M "valid" messages are spread throughout the space of 2^n elements.
- If there are no errors in transmission, we can verify the message is equal to a codeword with high probability.
- If there are errors in the message, we decode the message as the codeword that is "closest" (i.e.-differs by the fewest bits) from the received message.
- The number of differences between the two nearest codewords is called the distance of the code or d(C).

Hamming distance

- The best decoding strategy is to decode a message as the codeword that differs least from a codeword. So, for a coding scheme, C, if d(C)=2t+1 or less bits, we can correct t or less errors per block.

- If d(C)=s+1, we can detect s or fewer errors.

- The Hamming distance, denoted Dist(**v, w**), between two elements **v, w**$\in$GF(2)^n is the number of bits they differ by. The Hamming distance satisfies the usual conditions for a metric on a space.

- The Hamming weight of a vector **v**$\in$GF(2)^n , denoted, ||v|| is the number of 1's.

- If **v, w**$\in$GF(2)^n, Dist(**v, w**),= ||**v**$\oplus$**w**||.

Definition of a Code

- In the case of the “repeat three times” code, $C_{repeatx3}$, M=1 and n=3. There are two “codewords,” namely 111 and 000. $d(C_{repeatx3})=3$, so d=2t+1 with t=1.

- In general, a C(n,M,d) denotes a code in $GF(2)^n$ with M codewords with d(C)=d the minimum distance, n is dimension.

- As discussed, such codes can correctly decode transmissions containing t errors or less.

- The rate of the code is (naturally) R=lg(M)/n.

- Error correcting codes strive to find “high rate” codes that can efficiently encode and decode messages with acceptable error.

Example rates and errors

Code	n	M	d	R	p_1	p_2	$P_{1,e}$	$P_{2,e}$
Repetition x 3	3	2	3	1/3	3/4	7/8	0.156	0.043
Repetition x 5	5	2	5	1/5	3/4	7/8	0.103	0.016
Repetition x 7	7	2	7	1/7	3/4	7/8	0.071	0.006
Repetition x 9	9	2	9	1/9	3/4	7/8	0.049	0.004
Hamming(7,4)	7	16	3	4/7	3/4	7/8	0.556	0.215
Golay(24,12,8)	24	4096	17	1/2	3/4	7/8		
Hadamard (64,32,16)	64	32	16	3/16	3/4	7/8		
RM(4,2)	16	11	4					
BCH[7,3,4]	7	8	4	3/7				

Shannon

- Source Coding Theorem: The n random variables can be encoded by nH bits with negligible information loss.

- Channel Capacity: $C= \max_{P(x)} (H(I|O)-H(I))$. For a DMC, BSC with error rate p, this implies $C_{BSC}(p)= 1+p \lg(p) + q \lg(q)$. So for BSC $R=1-H(P)$.

- Channel Coding Theorem: $\forall R<C_{max}$, $\forall \varepsilon>0$, $\exists$ C(n,M,d) of length n with M codewords: $M \geq 2^{[Rn]}$ and $P^{(i)}_{error} \leq \varepsilon$ for $i=1,2,\ldots,M$.

- Translation: Good codes exist that permit transmission near the channel capacity with arbitrarily small error.

The Problem of Coding Theory

- Despite Shannon's fundamental results, this is not the end of the coding problem!
  - Shannon's proof involved random codes
  - Finding the closest codeword to a random point is the shortest vector problem, so "closest codeword" decoding is computationally difficult. Codes must be systematic to be useful.
  - The Encoding Problem: Given an m bit message, **m**, compute the codeword, **t** (for transmitted), in C(n,M,d).
  - The Decoding Problem: Given an n bit received word, **r=t+e**, where **e** was the error, compute the codeword in C(n,M,d) closest to **r**.
  - General codes are hard to decode

Bursts

- Bursty error correction: Errors tend to be "bursty" in real communications.

- Burst error correcting codes can be constructed by "spreading out codewords". Let $cw_i[j]$ mean bit j of codeword i. Transmit $cw_1[1]$, $cw_2[1]$,…, $cw_k[1]$, $cw_1[2]$,… where k is the size of a "long" error.

- Some specific codes (RS, for example) are good at bursty error correction.

Channel capacity for Binary Symmetric Channel

- Discrete memoryless channel: Errors independent and identically distributed according to channel error rate. (No memory).
- Rate for code, R_C= lg(M)/n.
- Channel capacity intuition: How many bits can be reliably transmitted over a BSC?
  - The channel capacity, c, of a channel is c= $\sup_X I(X;Y)$, where X is the transmission distribution and Y is the reception probability
  - Shannon-Hartley: c= Blg(1+S/N), B is the bandwidth, S is the signal power and N is the noise power.
  - Information rate, R=rH.

How much information can be transmitted over a BSC with low error?

- How many bits can be reliably transmitted over a BSC? Answer (roughly): The number of bits of bandwidth minus the noise introduced by errors.
- Shannon’s channel coding theorem tells us we can reliably transmit up to the channel capacity.
- However, good codes are hard to find and generally computationally expensive.

Calculating rates and channel capacity

- For single bit BSC, C=1+plg(p)+qlg(q).
- Recall c= $\sup_X I(X;Y)$.
- The distribution P(X=0)=P(X=1)=1/2 maxmizes this.
- c= 1/2+1/2+plg(p)+qlg(q)

Linear Codes

- A [n,k,d] linear code is an k-subspace of an n-space over F (usually GF(2)) with minimum distance d.
  - An [n,k,d] code is also a (n, 2^k,d) code
- Standard form for generator is G= $(I_k|A)$ with k message bits, n codeword bits. Codeword **c**=**m**G.
- For a linear code, d=$\min_{\mathbf{u}\neq\mathbf{0},\, \mathbf{u}\in C} \{wt(\mathbf{u})\}$.
  - Proof: Since C is linear, dist(**u, w**)= dist(**u-w,0**)=wt(**u-v**). Since the code is linear, **u-v**$\in$C. That does it.
- Parity check matrix is H: **v**$\in$C iff **v**H^T=0.
- If G is in standard form, H=$[-A^T|I_{n-k}]$. Note that GH=0.

- Example: Repetition code is the subspace in $GF(2)^3$ generated by (1,1,1).

G and H and decoding

- Let $\mathbf{r}=\mathbf{c}+\mathbf{e}$, where $\mathbf{r}$ is the received word, $\mathbf{c}$ is the transmitted word and $\mathbf{e}$ is the error added by the channel.
- Note codewords are linear combinations of rows of G and $\mathbf{r}H^T=\mathbf{c}H^T+\mathbf{e}H^T=\mathbf{e}H^T$.

- Coset leader table

Minimum weight Coset leader					**Error**	**Syndrone**
c_1	c_2	c_3	…	c_M	0	$\mathbf{0}=\mathbf{0}H^T$
c_1+e_1	c_2+e_1	c_3+e_1	…	c_M+e_1	e_1	e_1H^T
c_1+e_2	c_2+e_2	c_3+e_2	…	c_M+e_2	e_2	e_2H^T
…	….	…		….		
c_1+e_{h-1}	c_2+e_{h-1}	c_3+e_{h-1}	…	c_M+e_{h-1}	e_{h-1}	$e_{h-1}H^T$

Syndrome and decoding Linear Codes

- $S(\mathbf{r}) = \mathbf{r}H^T$ is called the syndrone.
- A vector having minimum Hamming weight in a coset is called a *coset leader*.
- Two vectors belong to the same coset iff they have the same syndrone.

- Now, here's how to systematically decode a linear code:
  1. Calculate $S(\mathbf{r})$.
  2. Find coset leader, **e**, with syndrone $S(\mathbf{r})$.
  3. Decode **r** as **r-e**.

- This is more efficient than searching for nearest codeword but is only efficient enough for special codes.

Syndrone decoding example (H[7,4])

$$G = [I_4|A] = \begin{matrix} 1&0&0&0&1&1&0 \\ 0&1&0&0&1&0&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&1&1&1 \end{matrix} = [-A^T|I_3],\quad H = \begin{matrix} 1&1&0&1&1&0&0 \\ 1&0&1&1&0&1&0 \\ 0&1&1&1&0&0&1 \end{matrix},\quad H^T = \begin{matrix} 1&1&0 \\ 1&0&1 \\ 0&1&1 \\ 1&1&1 \\ 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{matrix}$$

- Message: 1 1 0 0.
- Codeword transmitted: 1 1 0 0 0 1 1.
- Received: 1 1 0 0 0 0 1. (Error in 6th position)

Syndrone decoding example (H[7,4])

- Coset table (Left)

Syn	*Coset Leader*							
000	0000000	1000011	0100101	1100110	0010110	1010101	0110011	1110000
110	0000001	1000010	0100100	1100111	0010111	1010100	0110010	1110001
101	0000010	1000001	0100111	1100100	0010100	1010111	0110001	1110010
011	0000100	1000111	0100001	1100010	0010010	1010001	0110111	1110100
111	0001000	1001011	0101101	1101110	0011110	1011101	0111011	1111000
100	0010000	1010011	0110101	1110110	0000110	1000101	0100011	1100000
010	0100000	1100011	0000101	1000110	0110110	1110101	0010011	1010000
001	1000000	0000011	1100101	0100110	1010110	0010101	1110011	0110000

- (1 1 0 0 0 1) H^T= (0 1 0) which is the syndrone of the seventh row whose coset leader is **e**= (0 0 0 0 0 1 0).
- Decode message as (1 1 0 0 0 1) + (0 0 0 0 0 1 0) = (1 1 0 0 1 1).

Syndrone decoding example (H[7,4])

- Coset table (Right)

Syn

0001111	1001100	0101010	1101001	0011001	1011010	0111100	1111111
0001110	1001101	0101011	1101000	0011000	1011011	0111101	1111110
0001101	1001110	0101000	1101011	0011011	1011000	0111110	1111101
0001011	1001000	0101110	1101101	0011101	1011110	0111000	1111011
0000111	1000100	0100010	1100001	0010001	1010010	0110100	1110111
0011111	1011100	0111010	1111001	0001001	1001010	0101100	1101111
0101111	1101100	0001010	1001001	0111001	1111010	0011100	1011111
1001111	0001100	1101010	0101001	1011001	0011010	1111100	0111111

Bounds: How good can codes be?

- Let $A_q(n,d)$ denote the largest code with minimum distance d.
- **Sphere Packing (Hamming) Bound:** If d=2e+1, $A_q(n,d) \leq \sum_{k=0}^{e} {}_nC_k(q-1)^k \leq q^n$.
  - Proof: Let l be the number of codewords.
  - $l(1+(q-1)_nC_1+(q-1)^2{}_nC_2+\ldots+(q-1)^e{}_nC_e) \leq q^n$ because the e-spheres around the codewords are disjoint.
- **GSV Bound:** There is a linear [n, k, d] code satisfying the inequality: $A_q(n,d) \geq 2^n/(1+(q-1)\,{}_nC_1+(q-1)^2{}_nC_2+\ldots+(q-1)^{d-1}{}_nC_{d-1})$
  - Proof: The d-1 columns of the check matrix are linearly independent iff the code has distance d. So $q^{n-k} \geq (1+(q-1)\,{}_nC_1+(q-1)^2{}_nC_2+\ldots+(q-1)^{d-1}{}_nC_{d-1})$
- **Singleton Bound:** $M \leq q^{n-d+1}$, so $R \leq 1-(d-1)/n$.
  - Proof: Let C be a (n,M,d) code. Since every codeword differs by at least d-1 positions, $q^{n-(d-1)} \geq M$.

MDS

- Singleton Bound: $M \leq q^{n-d+1}$, so $R \leq 1-(d-1)/n$.
- Code meeting Singleton bound is an MDS code.
- If L is an MDS code so it $L^{\perp}$.
- If L is an [n,k] code with generator G, L is MDS iff there are k linearly independent columns.
- Binary 3-repetition code is an MDS

Hamming

- A Hamming code is a [n,k,d] linear code with
  - $n = 2^m - 1$,
  - $k = 2^m - 1 - m$
  - $d=3$.

- To decode **r=c+e**:
  - Calculate $S(r) = \mathbf{r}H^T$.
  - Find j which is the column of H with the calculated syndrome.
  - Correct position j.

[7,4] Hamming code

- The [7,4] code has encoding matrix G, and parity check H where:

$$G = \begin{matrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{matrix} \qquad H = \begin{matrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{matrix}$$

- The code words are:

0000000, weight: 0	0001111, weight: 4
1000011, weight: 3	1001100, weight: 3
0100101, weight: 3	0101010, weight: 3
1100110, weight: 4	1101001, weight: 4
0010110, weight: 3	0011001, weight: 3
1010101, weight: 4	1011010, weight: 4
0110011, weight: 4	0111100, weight: 4
1110000, weight: 3	1111111, weight: 7

Decoding Hamming code

$$G = [I_4|A] = \begin{matrix} 1&0&0&0&1&1&0 \\ 0&1&0&0&1&0&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&1&1&1 \end{matrix} = [-A^T|I_3],\quad H = \begin{matrix} 1&1&0&1&1&0&0 \\ 1&0&1&1&0&1&0 \\ 0&1&1&1&0&0&1 \end{matrix},\quad H^T = \begin{matrix} 1&1&0 \\ 1&0&1 \\ 0&1&1 \\ 1&1&1 \\ 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{matrix}$$

- Message: 1100 → 1100011.
- Received as 1100001.
- 1100001 H^T= 010 which is sixth row of H^T. Error in sixth bit.
- 1100001+0000010= 110011

Dual Code

- If C is an [n,k] linear code, then $C^{\perp} = \{\mathbf{u} : \mathbf{u} \cdot \mathbf{c} = 0, \forall \mathbf{c} \in C\}$ is an [n, n-k] linear code called the dual code.
- The parity check matrix, H, of a code, C, is the generator of its dual code.
- A code is self-dual if $C = C^{\perp}$.
- Weight enumerator: Let A_i be the number of codewords in C of weight i, then $A(z) = \sum_i A_i z^i$ is the weight enumerator.

Example: dual code of (7,4) Hamming code

- G=

```
1101100
1011010
0111001
```

Codewords:

```
0000000    0111001
1101100    1010101
1011010    1100011
0110110    0001111
```

Hadamard Code

- Hadamard Matrix: $H H^T = nI_n$. If H is Hadamard of order m, J=

  H H

  H -H

  is Hadamard of order 2m.
- Hadamard code uses this property. Generator matrix for this code is G= $[H|-H]^T$. For message I, $0 \leq i < 2^i$ send the row corresponding to i.
  - Used on Mariner spacecraft (1969).
- To decode, a 2^i bit received word, **r**, compute $\mathbf{d_i = r \cdot R_i}$**,** where $\mathbf{R_i}$ is the 2^i bit row i.
  - If there are no errors, the correct row will have $d_i = 2^{i-1}$ and all other rows will have $d_i=0$.
  - If one error, $d_i = 2^i-2$ (all dot products but 1 will be ± 2), etc.

Hadamard Code example

- Let $h_{ij} = (-1)^{a0\ b0\ +\ \ldots\ +\ a4\ b4}$, where **a** and **b** index the rows and columns respectively. This gives a 32 times 32 entry matrix, H.
- H(64, 32, 16): $64=2^6$ bit codewords, 6 messages. First 32 rows:

```
00000000000000000000000000000000  00      00000000000000001111111111111111  16
01010101010101010101010101010101  01      01010101010101011010101010101010  17
00110011001100110011001100110011  02      00110011001100111100110011001100  18
01100110011001100110011001100110  03      01100110011001101001100110011001  19
00001111000011110000111100001111  04      00001111000011111111000011110000  20
01011010010110100101101001011010  05      01011010010110101010010110100101  21
00111100001111000011110000111100  06      00111100001111001100001111000011  22
01101001011010010110100101101001  07      01101001011010011001011010010110  23
00000000111111110000000011111111  08      00000000111111111111111100000000  24
01010101101010100101010110101010  09      01010101101010101010101001010101  25
00110011110011000011001111001100  10      00110011110011001100110000110011  26
01100110100110010110011010011001  11      01100110100110011001100101100110  27
00001111111100000000111111110000  12      00001111111100001111000000001111  28
01011010101001010101101010100101  13      01011010101001011010010101011010  29
00111100110000110011110011000011  14      00111100110000111100001100111100  30
01101001100101100110100110010110  15      01101001100101101001011001101001  31
```

Hadamard Code example

- Last 32 rows:

11111111111111111111111111111111	32	11111111111111110000000000000000	48
10101010101010101010101010101010	33	10101010101010100101010101010101	49
11001100110011001100110011001100	34	11001100110011000011001100110011	50
10011001100110011001100110011001	35	10011001100110010110011001100110	51
11110000111100001111000011110000	36	11110000111100000000111100001111	52
10100101101001011010010110100101	37	10100101101001010101101001011010	53
11000011110000111100001111000011	38	11000011110000110011110000111100	54
10010110100101101001011010010110	39	10010110100101100110100101101001	55
11111111000000001111111100000000	40	11111111000000000000000011111111	56
10101010010101011010101001010101	41	10101010010101010101010110101010	57
11001100001100111100110000110011	42	11001100001100110011001111001100	58
10011001011001101001100101100110	43	10011001011001100110011010011001	59
11110000000011111111000000001111	44	11110000000011110000111111110000	60
10100101010110101010010101011010	45	10100101010110100101101010100101	61
11000011001111001100001100111100	46	11000011001111000011110011000011	62
10010110011010011001011001101001	47	10010110011010010110100110010110	63

Hadamard Code example

- Suppose received word is:
    - `1100110011001100001100110011000 1`

- Dot product with rows of matrix is:

- 00: 002, 01: 002, 02: -02, 03: -02, 04: -02, 05: -02, 06: 002, 07: 002.
- 08: -02, 09: -02, 10: 002, 11: 002, 12: 002, 13: 002, 14: -02, 15: -02.
- 16: -02, 17: -02, 18: -30, 19: 002, 20: 002, 21: 002, 22: -02, 23: -02.
- 24: 002, 25: 002, 26: -02, 27: -02, 28: -02, 29: -02, 30: 002, 31: 002.
- 32: -02, 33: -02, 34: 002, 35: 002, 36: 002, 37: 002, 38: -02, 39: -02.
- 40: 002, 41: 002, 42: -02, 43: -02, 44: -02, 45: -02, 46: 002, 47: 002.
- 48: 002, 49: 002, 50: 030, 51: -02, 52: -02, 53: -02, 54: 002, 55: 002.
- 56: -02, 57: -02, 58: 002, 59: 002, 60: 002, 61: 002, 62: -02, 63: -02.

- So we decode as 50 and estimate 1 error.

The amazing Golay code

- Golay Code $\mathcal{G}_{24}$ is a [24, 12, 8] linear code.
- $G= [I_{12} | C_0 | N] = [I_{12} | B]$
  - $C_0 = (1,1,1,1,1,1,1,1,1,1,1,0)^T$.
  - N is formed by circulating (1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0) 11 times and appending an row of 11 1's.
- The first row of N corresponds to the quadratic residues (mod 11).
- Note that $wt(\mathbf{r_1}+\mathbf{r_2}) = wt(\mathbf{r_1})+ wt(\mathbf{r_2})- 2[\mathbf{r_1}\cdot\mathbf{r_2}]$, all codewords have weight divisible by 4 and d(C)=8.
- $\mathcal{G}_{24} = \mathcal{G}_{24}^{\perp}$. To decode Golay, write $G=[I_{12} | B]$ and $B^T= (\mathbf{b}_1, \mathbf{b}_2 , ..., \mathbf{b}_{12})$ with $\mathbf{b}_i$ a column vector.

G for $\mathcal{G}$(24,12, 8)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	1	0
2	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	1	1	0	0	0	1
3	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	1	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	1	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	1	1	1	0
6	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1	1	1
7	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	1	0	1	1	0	1	1
8	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	1	1	0	1
9	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	0
10	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	1	0	0	0	1	0	1	1
11	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	1	1	0	0	0	1	0	1
12	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1

Properties of the Golay code

- The Golay code $\mathcal{G}(24,12, 8)$ is self dual. Thus, $GG^T=I+BB^T=0$
- Other properties:
  - Non-zero positions form a (24, 8, 5) Steiner system.
  - Weights are multiples of 4.
  - Minimum weight CW is 8 (hence d=8).
  - Codewords have weights 0, 8, 12, 16, 24.
  - Weight enumerator is $1+(759)x^8+(2576)x^{12}+(759)x^{16}+x^{24}$.
- Voyager 1, 2 used this code.
- Get $\mathcal{G}(23,12, 7)$ is obtained by deleting last column. It is a remarkable error correcting code. 7= 2 x 3 + 1, so it corrects 3 errors. It does this "perfectly."

The Golay code $\mathcal{G}(23,12, 7)$ is perfect!

- There are 2^{12} code words or sphere centers.
- There are ${}_{23}C_1$=23 points in Z_{23} which differ by one bit from a codeword.
- There are ${}_{23}C_2$=253 points in Z_{23} which differ by two bits from a codeword.
- There are ${}_{23}C_3$=1771 points in Z_{23} which differ by two bits from a codeword.
- 2^{12} (1+23+253+1771)= 2^{12}(2048)=2^{12} x 2^{11}= 2^{23}.
- 23 bit strings which differ by a codeword by 0,1,2 or 3 bits partition the entire space.

- The three sporadic simple Conway's groups are related to the lattice formed by codewords and provided at least one Ph.D. thesis.

Decoding $\mathcal{G}(24,12, 8)$

- Suppose **r=c+e** is received. $G= [I_{12} \mid B]=[c_1, c_2, \ldots, c_{24}]$ and $B^T= [b_1, b_2, \ldots, b_{12}]$.

- To decode:
  1. Compute $\mathbf{s}= \mathbf{r}G^T$, $\mathbf{s}B$, $\mathbf{s}+\mathbf{c_i}^T$, 1 l$\leq i \leq 24$ and $\mathbf{s}B+\mathbf{b_j}^T$, $1 \leq j \leq 12$.
  2. If wt(**s**)$\leq$3, non-zero entries of **s** correspond to non-zero entries of **e**.
  3. If wt(**s**B)$\leq$3, there is a non-zero entry in the k-th position of **s**B if the k+12-th position of e is non-zero.
  4. If wt($\mathbf{s}+\mathbf{c_i}^T$) ≤ 2, for some j, $13 \leq j \leq 24$ then $\mathbf{e_j}$=1 and non-zero entries of $\mathbf{s}+\mathbf{e_j}^T$ are in the same positions as non-zero entries of e.
  5. If wt($\mathbf{s}B+\mathbf{b_j}^T$) ≤ 2, for some j, $1 \leq j$ l≤ 12 then $\mathbf{e_j}$=1 and non-zero entries of $\mathbf{s}B+\mathbf{b_j}^T$ at position k correspond to non-zero entries of $\mathbf{e_{k+12}}$.

Decoding $\mathcal{G}(24,12, 8)$ example

- G is 12 x 24. $G=[I_{12}|B]= (c_1, c_2, \ldots, c_{24})$.
- $B^T=(b_1, b2, \ldots, b_{12})$.
- **m**=(1,1,0,0,0,0,0,0,0,0,1,0).
- **m**G=(1,1,0,0,0,0,1,0,1,0,1,1,0).
- **r**=(1,1,0,1,0,0,0,0,0,0,1,0,1,0,0,0,0,1,0,0,0,0,1,0).
- **s**=(011110110010).
- **s**B=(101011001000).
- Neither has wt≤3, so we compute $\mathbf{s+c_j^T}$, $\mathbf{sB+b_j^T}$.
- $\mathbf{s+b_4^T}$=(0,0,0,0,0,0,0,1,0,1,0,0)
- **c**= **r**+(0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0)
- **c**= (1,1,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,1,0,1,0,1,1,0)
- **m**=(1,1,0,0,0,0,0,0,0,0,1,0).

Cyclic codes

- A cyclic code, C, has the property that if $(c_1, c_2, \ldots, c_n) \in C$ then $(c_n, c_1, \ldots, c_{n-1}) \in C$.
- Remember polynomial multiplication in F[x] is linear over F.
- Denoting $U_n(x)= x^n - 1$ we have

- **<u>Theorem:</u>** C is a cyclic code of length n iff its generator $g(x)= a_0 + a_1 x + \ldots + a_{n-1}x^{n-1} \mid U_n(x)$ where codewords c(x) have the form m(x) g(x). Further, if $U_n(x)= h(x)g(x)$, c(x) in C iff $h(x)c(x) = 0 \pmod{U_n(x)}$.

Cyclic codes

- Let C be a cyclic code of length n over F, and let $\mathbf{a}=(a_0, a_1, \ldots, a_{n-1}) \in C$ be associated with the polynomial $p_{\mathbf{a}}(x)=a_0+a_1x+ \ldots +a_{n-1}x^{n-1}$. Let g(x) the polynomial of smallest degree over such associated polynomials the g(x) is the generating polynomial of C and
  1. g(x) is uniquely determined.
  2. $g(x)|x^n-1$
  3. C: f(x)g(x) where $\deg(f(x)) \leq n-1-\deg(g)$
  4. If $h(x)g(x)=x^n-1$, m(x)C iff $h(x)m(x)=0 \pmod{x^n-1}$.

- The associated matrices G and H are on the next slide.

G, H for cyclic codes

- Let g(x) be the generating polynomial of the cyclic code C.

$$G=\begin{array}{cccccccccccccccc}
g_0 & g_1 & g_2 & \dots & \dots & \dots & \dots & g_k & 0 & 0 & 0 & 0 \\
0 & g_0 & g_1 & g_2 & \dots & \dots & \dots & \dots & g_k & 0 & 0 & 0 \\
0 & 0 & g_0 & g_1 & g_2 & \dots & \dots & \dots & \dots & g_k & 0 & 0 \\
 & & \dots & & \dots & & \dots & & \dots & & & \\
0 & \dots & 0 & 0 & g_0 & g_1 & g_2 & \dots & \dots & \dots & \dots & g_k
\end{array}$$

$$H=\begin{array}{cccccccccccccccc}
h_l & h_{l-1} & h_{l-2} & \dots & \dots & \dots & \dots & h_0 & 0 & 0 & 0 & 0 \\
0 & h_l & h_{l-1} & h_{l-2} & \dots & \dots & \dots & \dots & h_0 & 0 & 0 & 0 \\
 & \dots & \dots & & & & \dots & & & \dots & & \\
0 & 0 & 0 & 0 & h_l & h_{l-1} & h_{l-2} & \dots & \dots & \dots & \dots & h_0
\end{array}$$

Cyclic code example

- $g(x)= 1+x^2+x^3$, $h(x)= 1+x^2+x^3+x^4$, $g(x)h(x)= x^n-1$, $n=7$.
- Message 1010 corresponds to $m(x)= 1+x^2$.
- $g(x)m(x)=c(x)= 1+x^3+x^4+x^5$, which corresponds to the codeword 1001110.
- G, H are
- Codewords are
  - 1011000 0101100 0010110 0001011 1110100 0111010 0011101 1001110
  - 0100111 1100010 0110001 1101001 1010011 1000101 1101001 1111111

BCH Codes

- Cyclic codes; so generator, $g(x)$ satisfies $g(x) \mid x^n - 1$.
- Theorem: Let C be a cyclic [n, k, d] code over F_q, $q=p^m$. Assume p does not divide n and $g(x)$ is the generator. Let $\square$ be a primitive root of x^n-1 and suppose that for some l, $\square$, we have $g(\square^l)= g(\square^{l+1})= \ldots = g(\square^{l+\square})=0$, then $d \geq \square+2$.
- Constructing a BCH code:
  1. Factor $x^n-1= f_1(x)\, f_2(x) \ldots f_r(x)$, each $f_i(x)$, irreducible.
  2. Pick $\square$, a primitive root of 1.
  3. $x^n-1= (x-\square)(x-\square^2)\ldots(x-\square^{n-1})$ and $f_i(x)= \square_t (x-\square^{j(t)})$.
  4. $q_j(x)= f_i(x)$, where $f_i(\square)=0$. $q_j(x)$ are not necessarily distinct.
  5. BCH code at designed distance d has generator $g(x)=\mathrm{LCM}[q_{k+1}(x),\ldots, q_{k+d-1}(x)]$.
- Theorem: A BCH code of designed distance d has minimum weight $\geq$ d. Proof uses theorem above.

Example BCH code

- $F=F_2$, $n=7$.
- $x^7-1=(x-1)(x^3+x^2+1)(x^3+x+1)$
- We pick □, a root of (x^3+x+1) as a primitive element.
- Note that $□^2$ and $□^4$ are also primitive roots of (x^3+x+1), so $x^3+x+1=(x-□)(x-□^2)(x-□^4)$ and $x^3+x^2+1=(x-□^3)(x-□^6)(x-□^6)$
- $q_0(x)=x-1$, $q_1(x)= q_2(x)= q_4(x)= x^3+x^2+1$.
- $k= -1$, $d=3$, $g(x)=[x-1,\ x^3+x^2+1]= x^4+x^3+1$.
- This yields a [7,3,4] linear code.

Decoding BCH Codes

- For **r=c+e**:
  1. Compute $(s_1, s_2) = \mathbf{r}H^T$,
  2. If $s_1 = 0$, no error,
  3. If $s_1 \neq 0$ put $s_2/s_1 = \square\square^{j-1}$, error is in position j (of $p \neq 2$, $e_i = s_1/\square\square^{(j-1)(k+1)}$,
  4. **c=r-e**.

Example Decoding a BCH Code

- x^7-1, α, a root of $x^3+x+1=0$. This is the 7-repetition code.
- $rH^T = (1,1,1,1,0,1,1,1)\ H^T = (\alpha+\alpha^2, \alpha)$
- $H = \begin{matrix} 1, & \alpha, & \alpha^2, & \alpha^3, & \alpha^4, & \alpha^5, & \alpha^6 \\ 1, & \alpha^2, & \alpha^4, & \alpha^6, & \alpha^8, & \alpha^{10}, & \alpha^{12} \end{matrix}$
- $s_1 = \alpha+\alpha^2 = 1+\alpha+\alpha^2+\alpha^3+\alpha^4+\alpha^5+\alpha^6$
- $s_2 = \alpha = 1+\alpha^2+\alpha^4+\alpha^6+\alpha^8+\alpha^{10}+\alpha^{12}$
- $s_1/s_2 = \alpha^4$, $j-1=4$, $j=5$, $\mathbf{e}=(0,0,0,0,1,0,0)$.
- $s_1 = e_j \alpha^{(j+1)(k+1)}$
- $s_2 = e_j \alpha^{(j+1)(k+2)}$

Reed Solomon

- Reed-Solomon code is BCH code over F_q with n= q-1. Let α be a primitive root of 1 and choose d: $1 \leq d < n$ with $g(x) = (x-\alpha)(x-\alpha^2) \ldots (x-\alpha^{d-1})$.
  - Since $g(\alpha) = g(\alpha^2) = \ldots = g(\alpha^{d-1}) = 0$, BCH bound shows $d(C) \geq d$.
  - Codewords are g(x)f(x), $\deg(f(x)) \leq n-d$. There are q^{n-d+1} such polynomials so q^{n-d+1} codewords.
  - Since this meets the Singleton bound, the Reed Solomon code is also an MDS code.
  - The Reed Solomon Code is an [n,n-d+1,d] linear code for these parameters

Reed Solomon example

- Example:
  - $F=GF(2^2)=\{0,1,□,□^2\}$
  - n=q-1=3, □= □.
  - Choose d=2, g(x)= (x-□).
  - G= □ □ 1 0

    0 □ □ 1
- Code consists of all 16 linear combinations of the rows of G.
- For CD's:
  - $F=GF(2^8)$, $n= 2^8-1=255$, d=33.
  - 222 information bytes.33 check bytes.
  - Codewords have 8 x 255 = 2040 bits.

Polynomials and RM codes

- R(r,m) has parameters [$n=2^m$, $k= 1 + {}_mC_1 + ... + {}_mC_r$ $d=2^{m-r}$], it consists of boolean functions whose polynomials are of degree $\leq$ m.
- RM(r,m)□□= RM(m-r-1,m).
- RM(0,m)= {0 , 1}, RM(r+1, m+1)= RM(r+1, m) * R(r, m).
- RM(n,0) is a repetition code with rate 1/n.
- Min distance in R(r,m)= 2^{m-r}.
- $G(r+1, m+1)= \begin{matrix} G(r+1,m) & G(r+1,m) \\ 0 & G(r,m) \end{matrix}$

RM(4,0) and RM(4,1)

- $n=2^4=16$.
- Constants
  - 0000 0000 0000 0000, 1111 1111 1111 1111.
- Linear
  - 1010 1010 1010 1010, 0101 0101 0101 0101,
  - 0000 1111 0000 1111, 0000 0000 1111 1111

RM(r,4) code example

1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
x_4	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
x_3	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
x_2	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
x_1	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
x_3x_4	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
x_2x_4	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
x_1x_4	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
x_2x_3	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
x_1x_3	0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1
x_1x_2	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$x_2x_3x_4$	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
$x_1x_3x_4$	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
$x_1x_2x_4$	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
$x_1x_2x_3$	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1
$x_1x_2x_3x_4$	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

McEliece Cryptosystem

- Bob chooses G for a large [n, k, d] linear code, we particularly want large d (for example, a [1024, 512, 101] Goppa code which can correct 50 errors in a 1024 bit block). Pick a k x k invertible matrix, S, over GF(2) and P, an n x n permutation matrix, and set G_1=SGP. G_1 is Bob's public key; Bob keeps P, G and S secret.
- To encrypt a message, **x**, Alice picks an error vector, **e**, and sends $\mathbf{y}=\mathbf{x}G_1+\mathbf{e}$ (mod 2).
- To decrypt, Bob, computes $\mathbf{y_1}=\mathbf{y}P^{-1}$ and $\mathbf{e_1}=\mathbf{e}P^{-1}$, then $\mathbf{y_1}=\mathbf{x}SG+\mathbf{e_1}$. Now Bob corrects $\mathbf{y_1}$ using the error correcting code to get $\mathbf{x_1}$. Finally, Bob computes $\mathbf{x}=\mathbf{x_1}S^{-1}$.
- Error correction is similar to the "shortest vector problem" and is believed to be "hard." In the example cited, a [1024, 512, 101] Goppa code, finding 50 errors (without knowing the shortcut) requires trying ${}_{1024}C_{50}>10^{85}$ possibilities.
- A drawback is that the public key, G_1, is largest.

McEliece Cryptosystem example - 1

- Using the [7, 4] Hamming code, G=

$$\begin{matrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{matrix}$$

- **m**=1011.

- S= $\begin{matrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{matrix}$ P= $\begin{matrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{matrix}$

McEliece Cryptosystem example - 2

- $G_1 =$

$$\begin{matrix} 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{matrix}$$

- $\mathbf{e} = (0\ 1\ 0\ 0\ 0\ 0\ 0)$
- $\mathbf{y_1} = \mathbf{y}P^{-1} = (0\ 0\ 1\ 0\ 0\ 0\ 1)$
- $\mathbf{x_1} = (0\ 0\ 1\ 0\ 0\ 1\ 1)$
- $\mathbf{x_0} = (0\ 0\ 1\ 0)$
- $\mathbf{x} = \mathbf{x_0}S^{-1} = (1\ 0\ 1\ 1)$

End