Cryptanalysis

Lecture 10 : Error Correcting Codes

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

This material is provided without warranty of any kind including, without limitation, warranty of non-infringement or suitability for any purpose. This material is not guaranteed to be error free and is intended for instructional use only.
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 \( p > 0.99 \), 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 \( P_{\text{error}} = 1 - (p^8 + 8p^7(1-p)) \) instead of \( P_{\text{error}} = 1 - p^8 \). If \( p = 0.99 \), \( P_{\text{error}} \) drops from about 7% to 0.3%.

• 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:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_1$</td>
<td>$b_2$</td>
<td>$c_1 = b_1 + b_2$</td>
</tr>
<tr>
<td>$b_3$</td>
<td>$b_4$</td>
<td>$c_2 = b_3 + b_4$</td>
</tr>
<tr>
<td>$c_3 = b_1 + b_3$</td>
<td>$c_4 = b_2 + b_4$</td>
<td>$c_5 = b_1 + b_2 + b_3 + b_4$</td>
</tr>
</tbody>
</table>

- 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 $P_{\text{correct}} = p^3 + 3p^2q > p$ and the chance of error is $P_{\text{error}} = 3pq^2 + q^3 < q$. For $p = 0.99$, $P_{\text{error}} = 0.000298$ and $P_{\text{correct}} = 0.999702$. 
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 $\text{Dist}(\mathbf{v}, \mathbf{w})$, between two elements $\mathbf{v}, \mathbf{w} \in \text{GF}(2)^n$ 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 $\mathbf{v} \in \text{GF}(2)^n$, denoted, $||\mathbf{v}||$ is the number of 1’s.

• If $\mathbf{v}, \mathbf{w} \in \text{GF}(2)^n$, $\text{Dist}(\mathbf{v}, \mathbf{w}) = ||\mathbf{v} \oplus \mathbf{w}||$. 
Definition of a Code

- In the case of the “repeat three times” code, $C_{\text{repeat}x3}$, $M=1$ and $n=3$. There are two “codewords,” namely 111 and 000. $d(C_{\text{repeat}x3})=3$, 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.
<table>
<thead>
<tr>
<th>Code</th>
<th>n</th>
<th>M</th>
<th>d</th>
<th>R</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$P_{1,e}$</th>
<th>$P_{2,e}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repetition x 3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1/3</td>
<td>3/4</td>
<td>7/8</td>
<td>0.156</td>
<td>0.043</td>
</tr>
<tr>
<td>Repetition x 5</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>1/5</td>
<td>3/4</td>
<td>7/8</td>
<td>0.103</td>
<td>0.016</td>
</tr>
<tr>
<td>Repetition x 7</td>
<td>7</td>
<td>2</td>
<td>7</td>
<td>1/7</td>
<td>3/4</td>
<td>7/8</td>
<td>0.071</td>
<td>0.006</td>
</tr>
<tr>
<td>Repetition x 9</td>
<td>9</td>
<td>2</td>
<td>9</td>
<td>1/9</td>
<td>3/4</td>
<td>7/8</td>
<td>0.049</td>
<td>0.004</td>
</tr>
<tr>
<td>Hamming(7,4)</td>
<td>7</td>
<td>16</td>
<td>3</td>
<td>4/7</td>
<td>3/4</td>
<td>7/8</td>
<td>0.556</td>
<td>0.215</td>
</tr>
<tr>
<td>Golay(24,12,8)</td>
<td>24</td>
<td>4096</td>
<td>17</td>
<td>1/2</td>
<td>3/4</td>
<td>7/8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hadamard (64,32,16)</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>3/16</td>
<td>3/4</td>
<td>7/8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RM(4,2)</td>
<td>16</td>
<td>11</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCH[7,3,4]</td>
<td>7</td>
<td>8</td>
<td>4</td>
<td>3/7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• 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 \( C_{\text{BSC}}(p) = 1 + p \lg(p) + q \lg(q) \). So for BSC \( R = 1 - H(P) \).

• Channel Coding Theorem: \( R < C_{\text{max}} \), \( D > 0 \), \( D \in \mathbb{R} \) \( C(n,M,d) \) of length n with M codewords: \( M \geq 2^{Rn} \) and \( P^{(i)}_{\text{error}} \leq \delta \) for \( i = 1, 2, ..., M \).

• 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 $c_{w_i[j]}$ mean bit $j$ of codeword $i$. Transmit $c_{w_1[1]}$, $c_{w_2[1]}$, ..., $c_{w_k[1]}$, $c_{w_1[2]}$, ... 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, \( R_C = \frac{\log(M)}{n} \).
• 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: \( c = B \log(1+S/N) \), \( B \) 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, $C = 1 + \text{plg}(p) + \text{qlg}(q)$.
- Recall $c = \sup_X I(X;Y)$.
- The distribution $P(X=0) = P(X=1) = 1/2$ maximizes this.
- $c = 1/2 + 1/2 + \text{plg}(p) + \text{qlg}(q)$
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 \(c = mG\).

- For a linear code, \(d = \min_{u \neq 0, u \in C} \{wt(u)\}\).
  - 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 \(vH^T = 0\).

- 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 $r = c + e$, where $r$ is the received word, $c$ is the transmitted word and $e$ is the error added by the channel.

• Note codewords are linear combinations of rows of $G$ and $rH^T = cH^T + eH^T = eH^T$.

• Coset leader table

<table>
<thead>
<tr>
<th>Minimum weight</th>
<th>Coset leader</th>
<th>Error</th>
<th>Syndrone</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$c_1$</td>
<td>$c_2$</td>
<td>$c_3$</td>
</tr>
<tr>
<td>$c_1 + e_1$</td>
<td>$c_2 + e_1$</td>
<td>$c_3 + e_1$</td>
<td>...</td>
</tr>
<tr>
<td>$c_1 + e_2$</td>
<td>$c_2 + e_2$</td>
<td>$c_3 + e_2$</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$c_1 + e_{h-1}$</td>
<td>$c_2 + e_{h-1}$</td>
<td>$c_3 + e_{h-1}$</td>
<td>...</td>
</tr>
</tbody>
</table>

JLM 20081102
Syndrome and decoding Linear Codes

- $S(r) = rH^T$ is called the syndrome.
- 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 syndrome.

Now, here’s how to systematically decode a linear code:
1. Calculate $S(r)$.
2. Find coset leader, $e$, with syndrome $S(r)$.
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])

\[
G = [I_4|A] = \begin{bmatrix}
1 & 0 & 0 & 1 & 1 & 0 & 0
\end{bmatrix},
H = \begin{bmatrix}
1 & 0 & 0 & 1 & 1 & 0 & 0
\end{bmatrix},
H^T = \begin{bmatrix}
1 & 0 & 1
\end{bmatrix}
\]

• 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)
Syndrome decoding example (H[7,4])

• Coset table (Left)

<table>
<thead>
<tr>
<th>Syn</th>
<th>Coset Leader</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0000000 1000110 0100101 1100110 0010110 1010101 0110011 1110000</td>
</tr>
<tr>
<td>110</td>
<td>0000001 1000100 1100111 0010111 1010100 0100100 1010111 0110001</td>
</tr>
<tr>
<td>101</td>
<td>0000010 1000011 1100100 0010100 1010111 0110010 1110010 0010101</td>
</tr>
<tr>
<td>011</td>
<td>0000100 1000111 1100010 0010010 1010001 0110111 1110100 0010011</td>
</tr>
<tr>
<td>111</td>
<td>0001000 1001011 0101101 1101110 0011110 1011101 0111011 1111000</td>
</tr>
<tr>
<td>100</td>
<td>0010000 1010111 1110110 0000110 1000101 0100110 1110101 0010011</td>
</tr>
<tr>
<td>010</td>
<td>0100000 1100111 0000101 1000110 0110110 1110101 0010011 1010000</td>
</tr>
<tr>
<td>001</td>
<td>1000000 0000111 1100101 0100110 1010110 0010110 1110110 0110000</td>
</tr>
</tbody>
</table>

• \((1 1 0 0 0 1) H^T = (0 1 0)\) which is the syndrome 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 |
| Syn | 0001110 1001101 0101011 1101000 0011000 1011011 0111101 1111110 |
| Syn | 0001101 1001110 0101000 1101011 0011011 1011000 0111110 1111101 |
| Syn | 0001011 1001000 0101110 1101101 0011101 1011110 0111000 1111011 |
| Syn | 0000111 1000100 0100010 1100001 0010001 1001001 0100100 1101111 |
| Syn | 0011111 1011100 0111010 1111001 0001001 1001010 0101100 1110111 |
| Syn | 0101111 1101100 0001010 1001001 0111001 1111010 0011100 1011111 |
| Syn | 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 $d=2e+1$, $A_q(n,d) \leq \sum_{k=0}^{e} \binom{n}{k}(q-1)^k \leq q^n$.
- Proof: Let $l$ be the number of codewords.
- $l(1+(q-1)\binom{n}{1}+(q-1)^2\binom{n}{2}+\ldots+(q-1)^e\binom{n}{e}) \leq q^n$ because the $e$-spheres around the codewords are disjoint.

**GSV Bound:** There is a linear $[n, k, d]$ code satisfying the inequality: $A_q(n,d) \geq 2^n/(1+(q-1)\binom{n}{1}+(q-1)^2\binom{n}{2}+\ldots+(q-1)^{d-1}\binom{n}{d-1})$.
- Proof: The $d-1$ columns of the check matrix are linearly independent iff the code has distance $d$. So $q^{n-k} \geq (1+(q-1)\binom{n}{1}+(q-1)^2\binom{n}{2}+\ldots+(q-1)^{d-1}\binom{n}{d-1})$.

**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 $S(r) = rH^T$.
  – 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:
  \[
  G = \begin{bmatrix}
  1 & 0 & 0 & 0 & 1 & 1 & 0 \\
  0 & 1 & 0 & 0 & 1 & 0 & 1 \\
  0 & 0 & 1 & 0 & 0 & 1 & 1 \\
  0 & 0 & 0 & 1 & 1 & 1 & 1
  \end{bmatrix}
  \]
  \[
  H = \begin{bmatrix}
  1 & 1 & 0 & 1 & 1 & 0 & 0 \\
  1 & 0 & 1 & 1 & 0 & 1 & 0 \\
  0 & 1 & 1 & 1 & 0 & 0 & 1 \\
  0 & 0 & 1 & 1 & 1 & 1 & 1
  \end{bmatrix}
  \]

- The code words are:

  - 0000000, weight: 0
  - 1000111, weight: 3
  - 0100011, weight: 3
  - 1100110, weight: 4
  - 0010110, weight: 3
  - 1010101, weight: 4
  - 0110011, weight: 4
  - 1110000, weight: 3
  - 0001111, weight: 4
  - 1001100, weight: 3
  - 0101010, weight: 3
  - 1101001, weight: 4
  - 0011001, weight: 3
  - 1011010, weight: 4
  - 0111100, weight: 4
  - 1111111, weight: 7
Decoding Hamming code

\[
\begin{array}{cccc}
1000110 & 1101100 & 110 \\
0100101 & [-A^T|I_3], & H=1011010, & H^T=101 \\
0010011 & 0111001 & 011 \\
0001111 & & 111 \\
& & 100 \\
& & 010 \\
& & 001 \\
\end{array}
\]

- 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 $C^\perp = \{u: u \cdot c = 0, \forall c \in C\}$ 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

- $G = \begin{bmatrix} 1101100 \\ 1011010 \\ 1011010 \\ 0111001 \end{bmatrix}$

Codewords:

$\begin{bmatrix} 0000000 \\ 1101100 \\ 1011010 \\ 0110110 \end{bmatrix}$

$\begin{bmatrix} 0111001 \\ 1010101 \\ 1100011 \\ 0001111 \end{bmatrix}$
Hadamard Code

• Hadamard Matrix: $H H^T = n I_n$. If $H$ is Hadamard of order $m$, $J =$
  
  $H  \quad H$
  $H  \quad -H$

  is Hadamard of order $2m$.

• Hadamard code uses this property. Generator matrix for this code is $G = [H\, -H]^T$. For message $l$, $0 \leq i < 2^i$ send the row corresponding to $i$.
  

• To decode, a $2^i$ bit received word, $r$, compute $d_i = r \cdot R_i$, where $R_i$ 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 $h_{ij} = (-1)^{a_0 b_0 + \ldots + a_4 b_4}$, 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 00000000000000000011111111111111 16
01010101010101010101010101010101 01 01010101010101011010101010101010 17
00110011001100110011001100110011 02 00110011001100110011001100110011 18
01100110011001100110011001100110 03 01100110011001100110011001100110 19
00001111000111100011110001111000 04 00001111000111100011110001111000 20
01011010011001100110011001100110 05 01011010011001100110011001100110 21
00111100011110001111000111100001 06 00111100011110001111000111100001 22
01101001011001100101100110010110 07 01101001011001100101100110010110 23
00000000111111111111111111111111 08 00000000111111111111111111111111 24
01010101101010101010101010101010 09 01010101101010101010101010101010 25
00110011110001111000111100011110 10 00110011110001111000111100011110 26
01100110011001100110011001100110 11 01100110011001100110011001100110 27
00001111111111111111111111111111 12 00001111111111111111111111111111 28
01011010101010101010101010101010 13 01011010101010101010101010101010 29
00111100110001111111111111111111 14 00111100110001111111111111111111 30
01101001100101100101100110011001 15 01101001100101100101100110011001 31
```
Hadamard Code example

- Last 32 rows:

<table>
<thead>
<tr>
<th>32</th>
<th>11111111111111111111111111111111</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>11111111111111111100000000000000</td>
</tr>
<tr>
<td>34</td>
<td>11111111111111111011010101010101</td>
</tr>
<tr>
<td>35</td>
<td>11111111111111110101010101010101</td>
</tr>
<tr>
<td>36</td>
<td>11111111111111110101010101010101</td>
</tr>
<tr>
<td>37</td>
<td>11111111111111110101010101010101</td>
</tr>
<tr>
<td>38</td>
<td>11111111111111110101010101010101</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>39</th>
<th>11111111111111110101010101010101</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>11111111111111110101010101010101</td>
</tr>
<tr>
<td>41</td>
<td>11111111111111110101010101010101</td>
</tr>
<tr>
<td>42</td>
<td>11111111111111110101010101010101</td>
</tr>
<tr>
<td>43</td>
<td>11111111111111110101010101010101</td>
</tr>
<tr>
<td>44</td>
<td>11111111111111110101010101010101</td>
</tr>
<tr>
<td>45</td>
<td>11111111111111110101010101010101</td>
</tr>
<tr>
<td>46</td>
<td>11111111111111110101010101010101</td>
</tr>
<tr>
<td>47</td>
<td>11111111111111110101010101010101</td>
</tr>
</tbody>
</table>

JLM 20081102
Hadamard Code example

• Suppose received word is:
  
  • 11001100110011000011001100110001

• Dot product with rows of matrix is:
  
  • 00: 002, 01: 002, 02: -02, 03: -02, 04: -02, 05: -02, 06: 002, 07: 002.

• So we decode as 50 and estimate 1 error.
The amazing Golay code

• Golay Code $G_{24}$ is a [24, 12, 8] linear code.
• $G = [I_{12} \mid C_0 \mid N] = [I_{12} \mid B]$
  – $C_0 = (1,1,1,1,1,1,1,1,1,1,1,0)^T$.
  – $N$ 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 $\text{wt}(r_1+r_2) = \text{wt}(r_1)+\text{wt}(r_2)-2[r_1 \cdot r_2]$, all codewords have weight divisible by 4 and $d(C)=8$.
• $G_{24} = G_{24}^\perp$. To decode Golay, write $G=[I_{12} \mid B]$ and $B^T=(b_1, b_2, \ldots, b_{12})$ with $b_i$ a column vector.
### G for $G(24,12,8)$

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

---

JLM 20081102
Properties of the Golay code

• The Golay code $G(24,12, 8)$ 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 $G(23,12, 7)$ 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 $G(23, 12, 7)$ is perfect!

- There are $2^{12}$ code words or sphere centers.
- There are $\binom{23}{1} = 23$ points in $\mathbb{Z}_{23}$ which differ by one bit from a codeword.
- There are $\binom{23}{2} = 253$ points in $\mathbb{Z}_{23}$ which differ by two bits from a codeword.
- There are $\binom{23}{3} = 1771$ points in $\mathbb{Z}_{23}$ which differ by two bits from a codeword.
- $2^{12}(1+23+253+1771) = 2^{12}(2048) = 2^{12} \times 2^{11} = 2^{23}$.
- 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 $G(24,12,8)$

• Suppose $r=c+e$ is received. $G=[l_{12} | B]=[c_1, c_2, ..., c_{24}]$ and $B^T=[b_1, b_2, ..., b_{12}]$.

• To decode:
  1. Compute $s=rG^T$, $sB$, $s+c_i^T$, $1 \leq i \leq 24$ and $sB+b_j^T$, $1 \leq j \leq 12$.
  2. If $wt(s)\leq3$, non-zero entries of $s$ correspond to non-zero entries of $e$.
  3. If $wt(sB)\leq3$, there is a non-zero entry in the $k$-th position of $sB$ if the $k+12$-th position of $e$ is non-zero.
  4. If $wt(s+c_i^T)\leq2$, for some $j$, $13 \leq j \leq 24$ then $e_j=1$ and non-zero entries of $s+e_j^T$ are in the same positions as non-zero entries of $e$.
  5. If $wt(sB+b_j^T)\leq2$, for some $j$, $1 \leq j \leq 12$ then $e_j=1$ and non-zero entries of $sB+b_j^T$ at position $k$ correspond to non-zero entries of $e_{k+12}$.
Decoding $G(24,12,8)$ example

- $G$ is 12 x 24. $G = [l_{12} | B] = (c_1, c_2, \ldots, c_{24})$.
- $B^T = (b_1, b_2, \ldots, b_{12})$.
- $m = (1,1,0,0,0,0,0,0,0,0,0,1,0)$.
- $m^G = (1,1,0,0,0,0,1,0,1,0,1,1,0)$.
- $r = (1,1,0,1,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0)$.
- $s = (011110110010)$.
- $sB = (101011001000)$.
- Neither has wt $\leq 3$, so we compute $s + c_j^T$, $sB + b_j^T$.
- $s + b_4^T = (0,0,0,0,0,0,0,1,0,1,0,0)$
- $c = r + (0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0)$
- $c = (1,1,0,0,0,0,0,0,0,1,0,1,0,0,0,0,1,0,1,0,1,1,0)$
- $m = (1,1,0,0,0,0,0,0,0,0,0,1,0)$. 
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

**Theorem:** 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 $a=(a_0, a_1, \ldots, a_{n-1}) \in C$ be associated with the polynomial $p_a(x)=a_0+a_1x+\ldots+a_{n-1}x^{n-1}$. 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. $C: f(x)g(x) \text{ where } \deg(f(x)) \leq n-1-\deg(g)$
  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 $C$.

\[
G = \begin{array}{cccccccc}
g_0 & g_1 & g_2 & \cdots & \cdots & \cdots & g_k & 0 & 0 & 0 & 0 \\
0 & g_0 & g_1 & g_2 & \cdots & \cdots & \cdots & g_k & 0 & 0 & 0 \\
0 & 0 & g_0 & g_1 & g_2 & \cdots & \cdots & \cdots & g_k & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & 0 & g_0 & g_1 & g_2 & \cdots & \cdots & \cdots & g_k \\
\end{array}
\]

\[
H = \begin{array}{cccccccc}
h_l & h_{l-1} & h_{l-2} & \cdots & \cdots & \cdots & h_0 & 0 & 0 & 0 & 0 \\
0 & h_l & h_{l-1} & h_{l-2} & \cdots & \cdots & \cdots & h_0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & h_l & h_{l-1} & h_{l-2} & \cdots & \cdots & \cdots & h_0 \\
\end{array}
\]
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 \( \mathbb{F}_q \), \( q=p^m \). Assume \( p \) does not divide \( n \) and \( g(x) \) is the generator. Let \( \bar{x} \) be a primitive root of \( x^n-1 \) and suppose that for some \( l, \bar{x} \), we have \( g(\bar{x}^l)= g(\bar{x}^{l+1})= \ldots = g(\bar{x}^{l+d-1})=0 \), then \( d \geq \bar{x}+2 \).
- 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 \( \bar{x} \), a primitive root of \( 1 \).
  3. \( x^n-1= (x- \bar{x}^1)(x- \bar{x}^2)\ldots(x- \bar{x}^{n-1}) \) and \( f_i(x)= \bar{x}^t(x-\bar{x}^j) \).
  4. \( q_j(x)= f_i(x) \), where \( f_i(\bar{x})=0 \). \( q_j(x) \) are not necessarily distinct.
  5. BCH code at designed distance \( d \) has generator \( g(x)=\text{LCM}[q_{k+1}(x), \ldots, q_{k+d-1}(x)] \).
- 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 $\alpha$, a root of $(x^3+x+1)$ as a primitive element.
• Note that $\alpha^2$ and $\alpha^4$ are also primitive roots of $(x^3+x+1)$, so $x^3+x+1=(x-\alpha^2)(x-\alpha^4)$ and $x^3+x^2+1=(x-\alpha^3)(x-\alpha^6)(x-\alpha^6)$
• $q_0(x)=x-1$, $q_1(x)= q_2(x)= q_4(x)= x^3+x^2+1$.
• $k=-1$, $d=3$, $g(x)=[x-1, x^3+x^2+1]= x^4+x^3+1$.
• This yields a $[7,3,4]$ linear code.
Decoding BCH Codes

• For \( r = c + e \):
  1. Compute \( (s_1, s_2) = rH^T \),
  2. If \( s_1 = 0 \), no error,
  3. If \( s_1 \neq 0 \) put \( s_2/s_1 = \square^{-1} \), error is in position \( j \) (of \( p \neq 2 \), \( e_i = s_1/\square^{-1} \)),
  4. \( c = r - e \).
Example Decoding a BCH Code

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

- Reed-Solomon code is BCH code over $F_q$ with $n = q-1$. Let $\alpha$ 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:
  – $F = \text{GF}(2^2) = \{0, 1, 0, 1\}$
  – $n = q - 1 = 3$, $\bar{w} = \bar{0}$.
  – Choose $d = 2$, $g(x) = (x - \bar{0})$.
  – $G = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

• Code consists of all 16 linear combinations of the rows of $G$.

• For CD’s:
  – $F = \text{GF}(2^8)$, $n = 2^8 - 1 = 255$, $d = 33$.
  – 222 information bytes. 33 check bytes.
  – Codewords have $8 \times 255 = 2040$ 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 $\leq m$.

- $RM(r,m) \subseteq RM(m-r-1,m)$.

- $RM(0,m) = \{0, 1\}$, $RM(r+1, m+1) = RM(r+1, m) \ast R(r, m)$.

- $RM(n,0)$ is a repetition code with rate $1/n$.

- Min distance in $R(r,m) = 2^{m-r}$.

- $G(r+1,m)$

  - $G(r+1, m+1) = 0$ $G(r,m)$
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

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_4$</td>
<td>0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1</td>
<td></td>
</tr>
<tr>
<td>$x_3$</td>
<td>0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1</td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td>0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td>$x_1$</td>
<td>0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1</td>
<td></td>
</tr>
<tr>
<td>$x_3x_4$</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1</td>
<td></td>
</tr>
<tr>
<td>$x_2x_4$</td>
<td>0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td>$x_1x_4$</td>
<td>0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1</td>
<td></td>
</tr>
<tr>
<td>$x_2x_3$</td>
<td>0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td>$x_1x_3$</td>
<td>0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 1</td>
<td></td>
</tr>
<tr>
<td>$x_1x_2$</td>
<td>0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1</td>
<td></td>
</tr>
<tr>
<td>$x_2x_3x_4$</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td>$x_1x_3x_4$</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td>$x_1x_2x_4$</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1</td>
<td></td>
</tr>
<tr>
<td>$x_1x_2x_3$</td>
<td>0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1</td>
<td></td>
</tr>
<tr>
<td>$x_1x_2x_3x_4$</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1</td>
<td></td>
</tr>
</tbody>
</table>
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 \(k \times k\) invertible matrix, \(S\), over GF(2) and \(P\), an \(n \times n\) 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 \(y = xG_1 + e \pmod{2}\).

- To decrypt, Bob, computes \(y_1 = yP^{-1}\) and \(e_1 = eP^{-1}\), then \(y_1 = xSG + e_1\). Now Bob corrects \(y_1\) using the error correcting code to get \(x_1\). Finally, Bob computes \(x = x_1S^{-1}\).

- 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 \(\binom{1024}{50} > 10^{85}\) possibilities.

- A drawback is that the public key, \(G_1\), is largest.
McEliece Cryptosystem example - 1

• Using the [7, 4] Hamming code, \( G = \)
  
  \[
  \begin{array}{cccccc}
  1 & 0 & 0 & 0 & 1 & 1 \\
  0 & 1 & 0 & 0 & 1 & 0 \\
  0 & 0 & 1 & 0 & 0 & 1 \\
  0 & 0 & 0 & 1 & 1 & 1 \\
  \end{array}
  \]

• \( m = 1011. \)

• \( S = 1 \ 0 \ 0 \ 1 \quad P = 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \)
  
  \[
  \begin{array}{cccccc}
  1 & 1 & 0 & 1 & 1 & 1 \\
  0 & 1 & 0 & 1 & 1 & 1 \\
  1 & 1 & 1 & 0 & 1 & 1 \\
  \end{array}
  \]

JLM 20081102
McEliece Cryptosystem example - 2

- $G_1 =
  \begin{bmatrix}
  0 & 0 & 1 & 1 & 0 & 1 & 0 \\
  1 & 0 & 1 & 0 & 0 & 1 & 1 \\
  1 & 1 & 0 & 0 & 0 & 1 & 0 \\
  1 & 0 & 1 & 0 & 1 & 0 & 0 
  \end{bmatrix}$

- $e = (0 1 0 0 0 0 0)$
- $y_1 = yP^{-1} = (0 0 1 0 0 0 1)$
- $x_1 = (0 0 1 0 0 1 1)$
- $x_0 = (0 0 1 0)$
- $x = x_0S^{-1} = (1 0 1 1)$
End