Supélec
9 Jan. 2009

Three Coding Problems

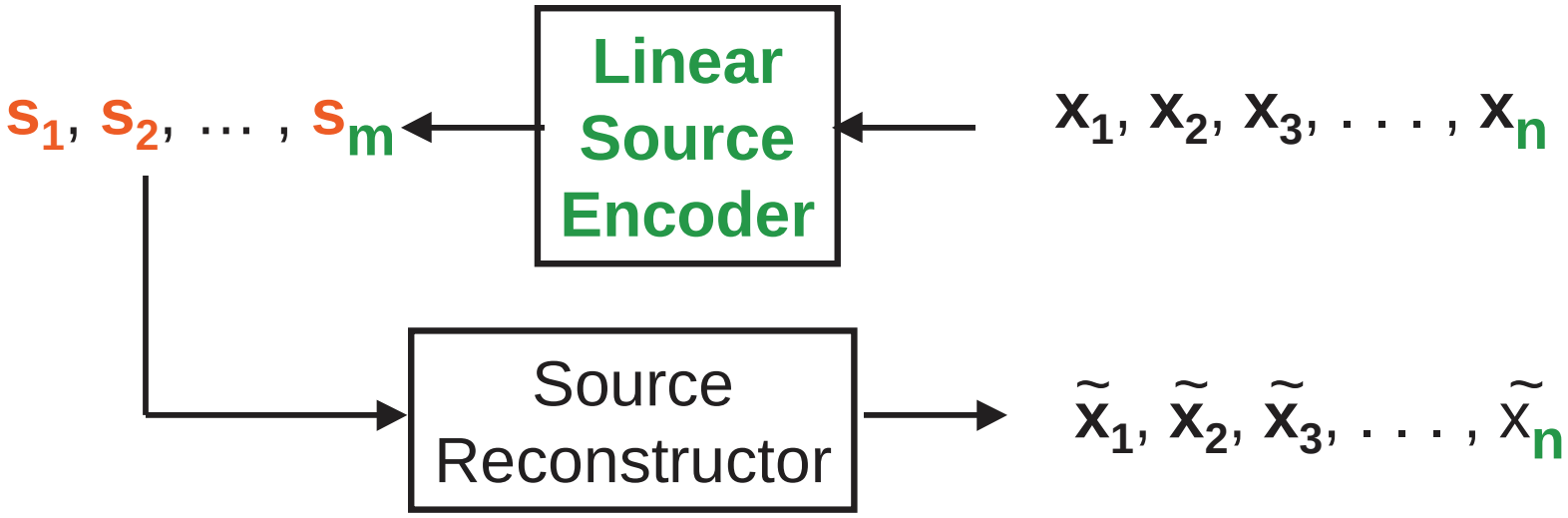
James L. Massey
(Prof.-*em*. ETH Zurich)
Trondhjemsgade 3, 2TH
DK-2100 Copenhagen, Denmark

jamesmassey@compuserve.com

Three Coding Problems

- Linear Source Coding
- Ambiguous Decoding and Erroneous Decoding
- Access Structures for Secret Sharing

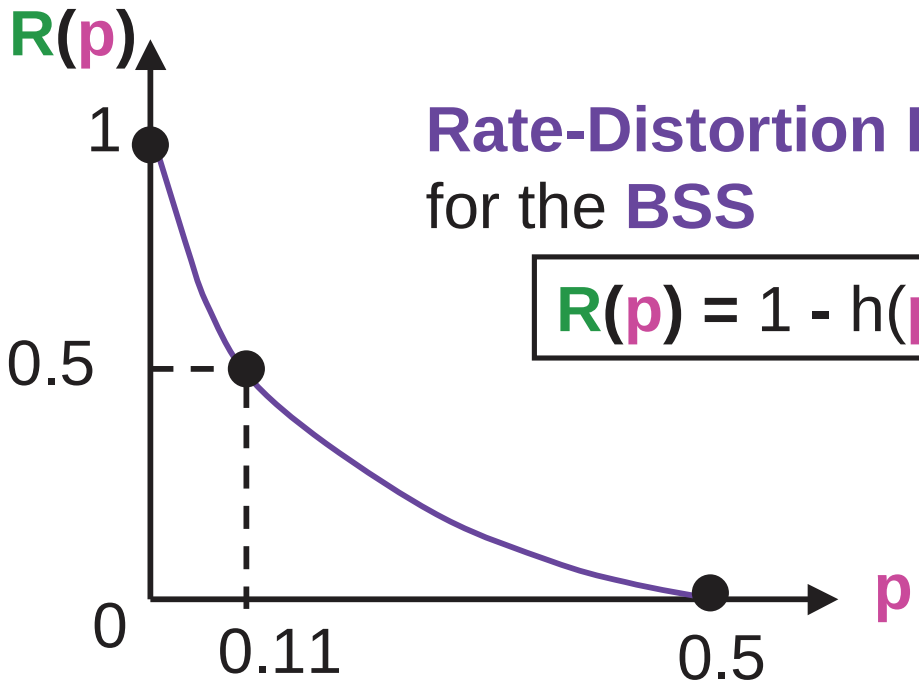
By a **binary linear source encoder** we mean a device that implements a **linear transformation** from the input sequence of length **n** to the output sequence of length **m**, where the digits of the sequences are from the finite field GF(2).



The (optimum) source **reconstructor** will in general be a **nonlinear** device!

How well can one do with **general** source coding for the **Binary Symmetric Source (BSS)?**

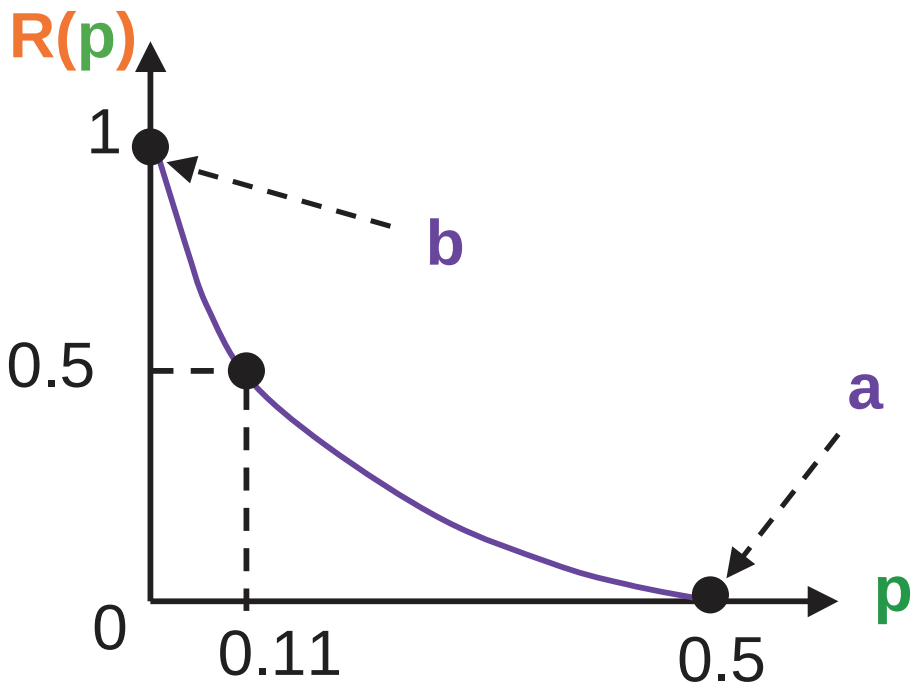
p = Hamming distortion (bit error probability)



Rate-Distortion Function
for the **BSS**

$$R(p) = 1 - h(p)$$

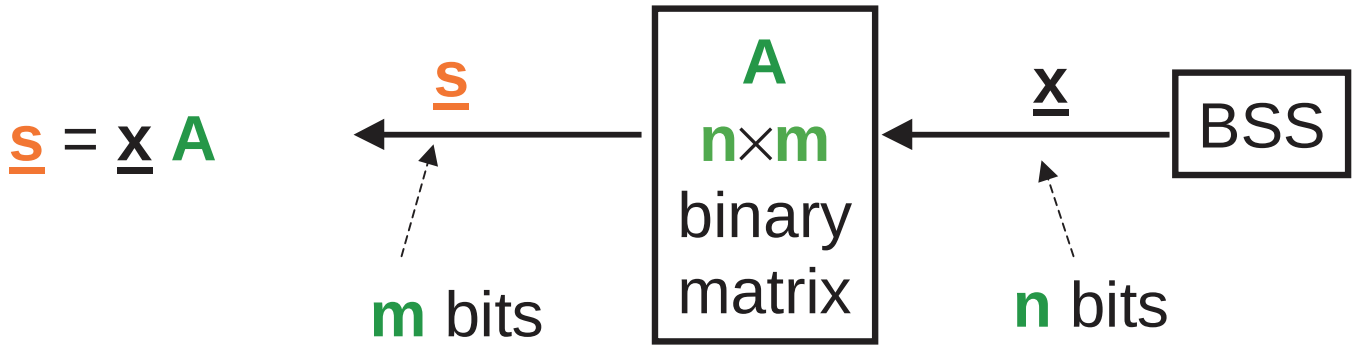
Can this be achieved with linear source coding, i.e., can we approach $R = m/n = 1 - h(p)$?



• We can approach point **a** by the linear source encoder with **n** >> 1, **m** = 1 and $s_1 \equiv 0$.

• We can achieve point **b** by the linear source encoder with **n** = **m** and the identity transformation.

Q: How well can we do in between?



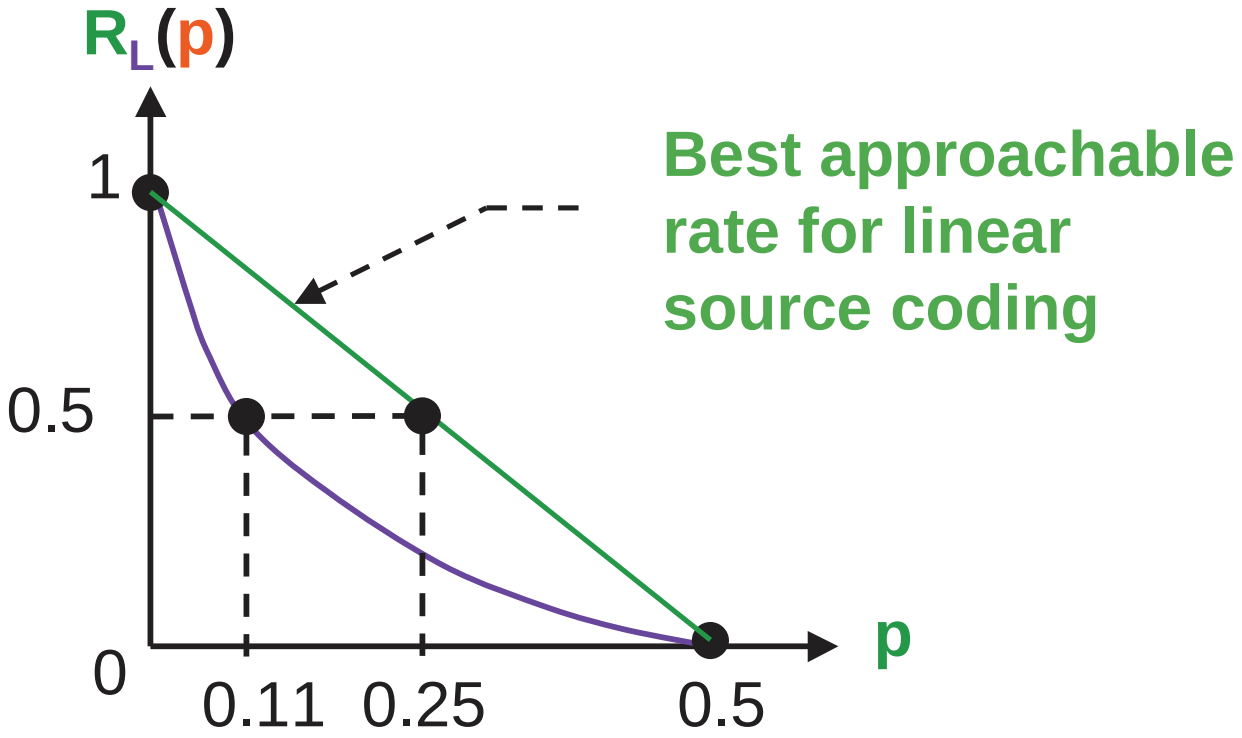
Linear Source Encoding of a BSS

Ancheta's Theorem: An **n**×**m** binary matrix **A** with 1 ≤ **m** ≤ **n**, used as a linear source encoder for a BSS together with an optimum reconstructor, gives Hamming distortion **p** satisfying

$$p \geq (1 - m/n)/2$$

with **equality** if and only if **A** is a rank **m** matrix with **n** - **m** all-zero rows.

$\Longrightarrow$ $R_L(p) = 1 - 2p$ **Linear rate-distortion function.**



Teofilo C. Ancheta Jr., “Bounds and techniques for linear source coding” (Ph.D. Thesis abstr.) *IEEE Trans. Inform. Th.*, Vol. IT-24, March 1978 p. 276.

Proof of Ancheta's theorem:

If the **m** columns of **A** are not linearly independent, we can remove one column and calculate the missing bit of **s**. We can also permute the rows of **A**, which results in the same **s** for **x'** obtained by applying the same permutation to **x**. The statistics of **x** and **x'** are identical. Thus, with no loss of generality or optimality, we may consider

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_I} \\ \text{--} \\ \mathbf{A_{II}} \end{bmatrix} \begin{array}{l} \blacktriangleleft \text{----} \ \mathbf{m}\times\mathbf{m} \text{ nonsingular matrix} \\ \blacktriangleleft \text{----} \ (\mathbf{n-m})\times\mathbf{m} \text{ matrix} \end{array}$$

$$\mathbf{s} = \mathbf{x}\mathbf{A} = \mathbf{x_I}\mathbf{A_I} + \mathbf{x_{II}}\mathbf{A_{II}}$$

so that

$\mathbf{x_I} = (\mathbf{s} - \mathbf{x_{II}}\mathbf{A_{II}})\mathbf{A_I}^{-1}$ is the general solution.

We can arbitrarily choose the **n-m** tuple $\underline{\mathbf{x}}_{\mathbf{II}}$.

$\underline{\mathbf{x}}_{\mathbf{I}} = (\underline{\mathbf{s}} - \underline{\mathbf{x}}_{\mathbf{II}}\mathbf{A}_{\mathbf{II}})\mathbf{A}_{\mathbf{I}}^{-1}$, which then determines $\underline{\mathbf{x}}_{\mathbf{I}}$.

BSS $\Longrightarrow$ all $2^{\mathbf{n-m}}$ solutions $[\underline{\mathbf{x}}_{\mathbf{I}}, \underline{\mathbf{x}}_{\mathbf{II}}]$ are equiprobable.

Let $\underline{\tilde{\mathbf{x}}} = [\underline{\tilde{\mathbf{x}}}_{\mathbf{I}} : \underline{\tilde{\mathbf{x}}}_{\mathbf{II}}] = F(\underline{\mathbf{s}})$ be the reconstructed sequence. (This need not be a solution of the above equation.)

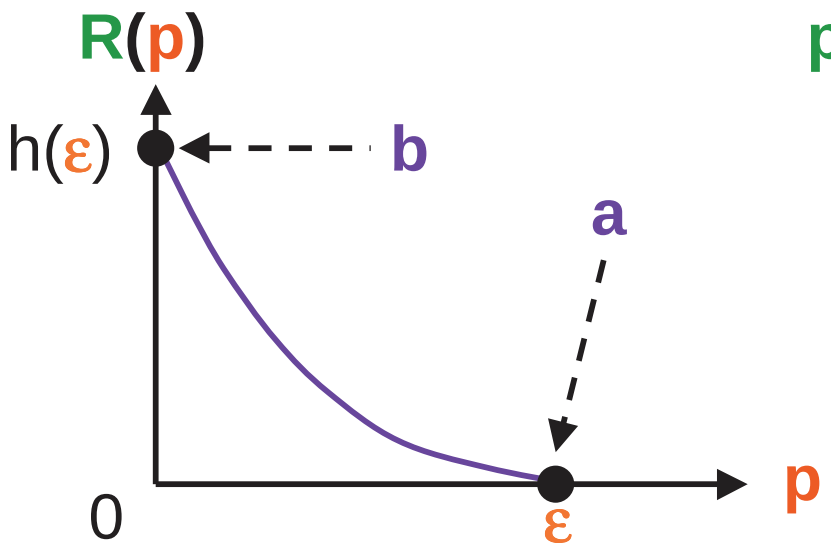
Because every choice of the **n-m** tuple $\underline{\mathbf{x}}_{\mathbf{II}}$ appears in one solution $[\underline{\mathbf{x}}_{\mathbf{I}}, \underline{\mathbf{x}}_{\mathbf{II}}]$, the reconstruction $\underline{\tilde{\mathbf{x}}}_{\mathbf{II}}$ will always have a 50% error rate in its **n-m** bits.

The error rate in the **m** bits of $\underline{\mathbf{x}}_{\mathbf{I}}$ will be 0 if and only if the solution for $\underline{\mathbf{x}}_{\mathbf{I}}$ is unique, i.e., if and only if $\mathbf{A}_{\mathbf{II}} = \mathbf{0}$. Thus the Hamming distortion satisfies

$$\mathbf{p} \geq \tfrac{1}{2}(\mathbf{n}-\mathbf{m})/\mathbf{n} + 0\times\mathbf{m}/\mathbf{n} = (1 - \mathbf{m}/\mathbf{n})/2$$

with equality if and only if $\mathbf{A}_{\mathbf{II}} = \mathbf{0}$.

Rate-distortion function for a **binary memoryless source (BMS)** with probability ε of outputting a 1, where $0 \le \varepsilon \le 1/2$.



p = Hamming distortion

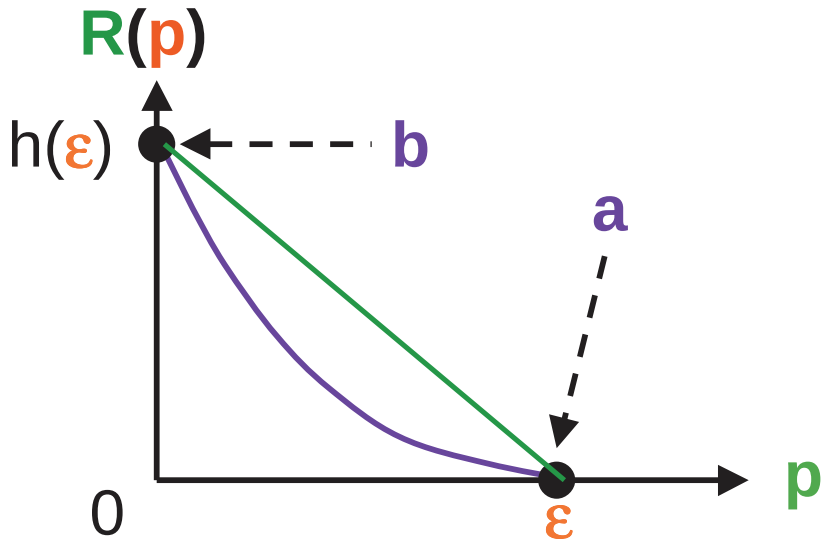
$$R(p) = h(\varepsilon) - h(p)$$

- We can approach point **a** by the linear source encoder with **n** >> 1, **m** = 1 and $s_1 \equiv 0$.
- We can approach point **b** by taking A^T to be the parity-check matrix for a linear code that operates close to capacity on a BSC with crossover prob. ε .

Q: How well can we do in between?

Nobody knows!

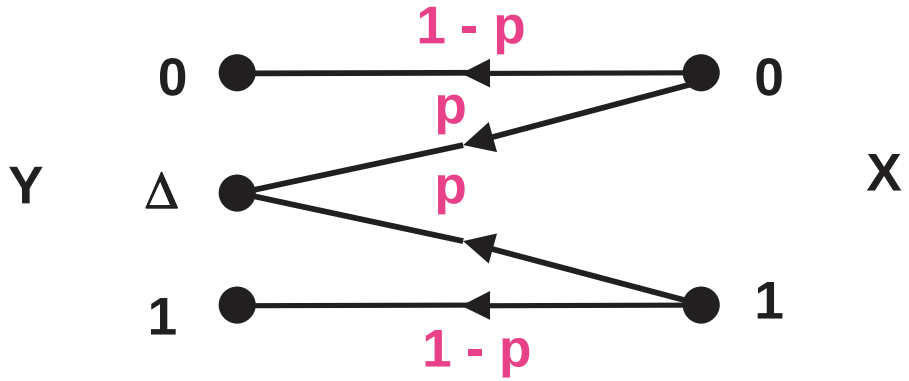
So here is a little problem for you.



Either give a proof that the green line above, namely $R_L(p) = h(\varepsilon)(1 - p/\varepsilon)$, is the linear rate-distortion function for the **BMS** with probability ε of outputting a 1, **or** give a counterexample to this assertion.

Ambiguous Decoding and **Erroneous** Decoding

For block coding without feedback on a DMC, we will say a decoding decision is **unambiguous** if the **decoder knows for certain that this decoding decision is correct**.



The Binary Erasure Channel (BEC)

The probability of erroneous decoding can be made arbitrarily small at any rate less than capacity **C** = 1 - **p**.

For what range of rates can the probability of ambiguous decoding be made arbitrarily small?

For the **BEC**, the decoder knows that all unerased received digits are correct but that erased digits could have been either 0 or 1.

For the **BEC**, if there is **only one codeword** that **agrees** with the received word in **all unerased positions**, then the decoding is unambiguous.

Let P_A be the probability of **ambiguous decoding**.

Let P_{ML} be the error probability for **maximum-likelihood decoding** for the same code .

P_{ML} can be made arbitrarily small on the **BEC** at any rate less than capacity $C = 1 - p$.

A simple but useful theorem:
For block coding on a **BEC** with $0 < p < 1$,

$$P_A \leq 2\, P_{ML}.$$

Suppose **y** is the received block and **x** is a codeword. Then $P(\mathbf{y}|\mathbf{x}) = 0$ unless **x** agrees with **y** in all unerased positions, in which case $P(\mathbf{y}|\mathbf{x}) = (1-p)^{N-e}\, p^e$ where e is the number of erasures.

$\Rightarrow$ every "all-agreeing" codeword is a valid choice for the maximum-likelihood decoding decision.

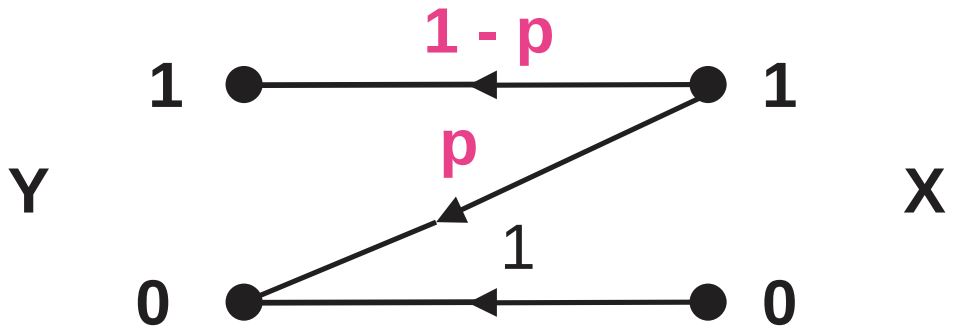
The correct codeword must agree with **y** in all unerased positions. Thus the conditional probability of error for the ML decoder is 0 when the decoding is unambiguous and is at least ½ when decoding is ambiguous. It follows that $P_{ML} \geq \frac{1}{2} P_A$.

Thus,

P_A can be made arbitrarily small on the **BEC** at any rate less than capacity $C = 1 - p$.

What about other channels?

For a code with at least two codewords to have $P_A < 1$, the DMC must have an output letter that is a **disprover for some input letter**, i.e, that cannot be reached with nonzero probability from this input letter.



The **Z-channel** ($0 < p < 1$)

The output letter 1 is a **disprover** for the input letter 0.

Suppose **y** is the received block and **x** is a codeword. Then P(**y**|**x**) = 0 unless **x** agrees with **y** in all positions where **y** contains a 1, in which case $P(\mathbf{y}|\mathbf{x}) = (1-p)^{v}p^{w-v}$ where **v** is the number of 1's in **y** and **w** is the number of 1's in the codeword.

$\Rightarrow$ every "matching" codeword with the minimum weight w is a valid choice for the maximum-likelihood decoding decision.

$\Rightarrow$ if the code is a constant-weight code, then every "matching" codeword is a valid choice for the maximum-likelihood decoding decision.

Thus the conditional probability of error for the ML decoder is 0 when the decoding is unambiguous and is at least ½ when decoding is ambiguous. It follows that

$$P_{ML} \geq \tfrac{1}{2} P_A.$$

For a constant-weight block code on the Z-channel,
$P_A \leq 2\, P_{ML}$.

Q: Can P_A be made arbitrarily small on the **Z-channel** at any rate less than capacity **C**?

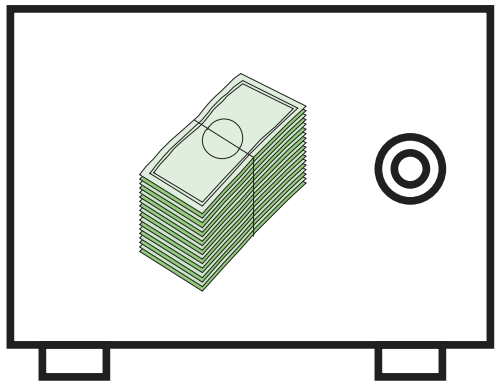
I don't know the answer.

Q: Are there other DMC's besides the BEC such that P_A can be made arbitrarily small at any rate less than capacity **C**?

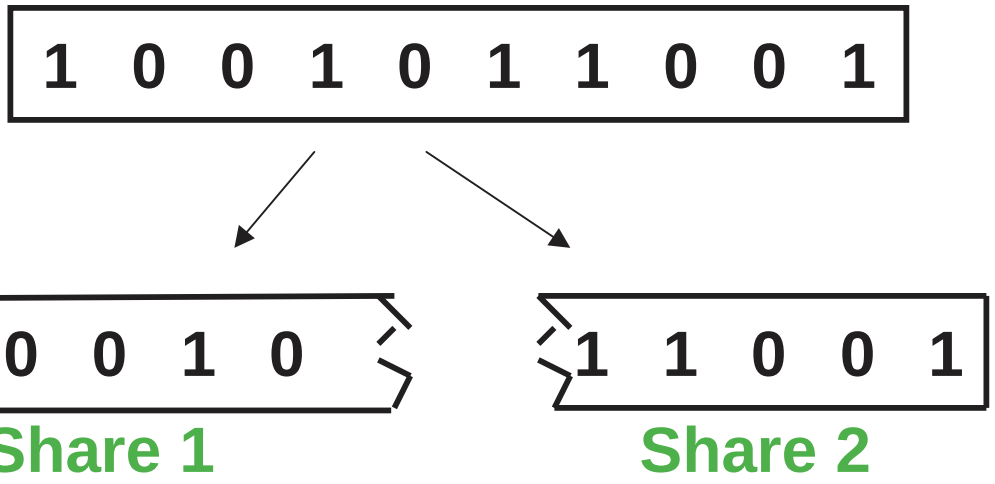
I don't know the answer.

Secret Sharing

The “classical” way that two crooks (or two bank vice presidents), who do not trust one another, can share a secret.

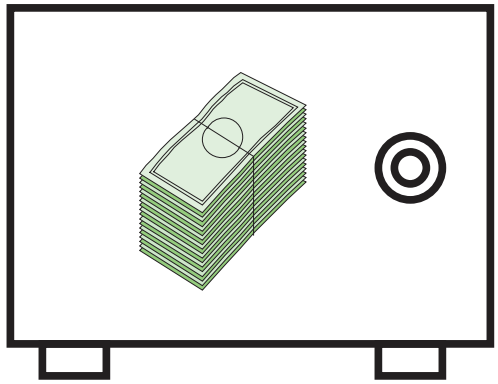


The **secret**:



The **secret** **“leaks out”** —one share is not worthless!

No-Leak Secret Sharing



The secret	1 0 0 1 0 1 1 0 0 1
Share 1: BSS output	0 0 1 1 0 1 0 1 1 1
Share 2: secret ⊕ BSS output	1 0 1 0 0 0 1 1 1 0

No leakage!

(Share 2 is the Vernam encryption of the secret.)

The idea of secret sharing is due independently to Shamir and to Blakley.

A. Shamir, "How to share a secret", *Communications of the ACM*, vol. 22, pp. 612-613, November 1979.

G. R. Blakley, “Safeguarding cryptographic keys”, Proc. AFIPS Natl. Conf., pp. 313-317, 1979.

Shamir showed how a secret can be “divided” into N shares so that **any T shares uniquely determine the secret**, but **any T - 1 or fewer shares give no information about the secret**.

This kind of secret-sharing scheme is called a threshold scheme and T is called the threshold.

In his formulation of “perfect” secret sharing for a threshold access structure, Shamir actually reinvented the Reed-Solomon codes !

The connection to Reed-Solomon codes was made explicitly in R. J. McEliece and D. V. Sarwate, “On sharing secrets and Reed-Solomon codes”, *Comm. ACM*, Vol. 24, pp. 583-584, September 1981.

We now show that linear codes can be used to give a convenient description of perfect secret sharing for a **general access structure**.

Non-threshold access structures for secret sharing:

Example: Consider the 2^m-ary (5, 3) code with **parity-check matrix**

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$\Rightarrow$ $v_1 + v_2 + v_3 = 0$ and $v_1 + v_3 + v_4 + v_5 = 0$.

- If v_1 is the **secret**, if v_2, v_3, v_4 and v_5 are the **shares**, and if v_2 and v_4 are chosen uniformly at random (note that {1, 2, 4} is an **information set**), then
- shares v_2 and v_3 **determine the secret**
- shares v_3, v_4 and v_5 **determine the secret**
- but **no other set of shares not containing one of these sets gives any information about the secret**.

What is really going on here?

If (and only if) we can solve for the **secret** (v_1) as a linear combination of some **shares**, i.e.,

$$v_1 = c_2 v_2 + c_3 v_3 + \ldots + c_n v_n$$

then $[1\ -c_2\ -c_3 \ldots -c_n]$ is a **parity check on the codewords of the code**, i.e., a codeword in the **dual code**.

Because the original code is a linear code, an n-tuple is a codeword if and only if it satisfies all the parity checks of the code. Thus, **linear combinations completely determine the manner in which the secret can be determined from shares.**

A codeword $[v_1, v_2, \ldots, v_n]$ in a q-ary (n, k) code is **minimal** if it is non-zero, if its **leftmost non-zero component** is a 1, and if its **non-zero coordinates cover all the non-zero coordinates of no other codeword whose leftmost non-zero component is a 1**.

In the previous example,

[1 1 1 0 0], [1 0 1 1 1] and [0 1 0 1 1]

are all the minimal codewords in the dual code.

The minimal codewords in the dual code with first component equal to 1 correspond to the minimal sets of shares that determine the secret.

A few facts about minimal codewords in a linear code.

All **minimum-weight codewords** with leftmost non-zero component 1 are minimal codewords.

Every non-zero non-minimal codeword is a **linear combination** of those minimal codewords that are covered by this non-zero codeword

Every **non-minimal non-zero codeword covers a minimal codeword** whose leftmost non-zero component occurs in the same position as the leftmost 1 component of this non-minimal codeword.

Example: Consider the (3. 2) linear code over GF(2^{10}) with the parity-check matrix

$$H = [1 \;\; 1 \;\; 1].$$

The only non-zero codeword in the dual code is [1, 1, 1], which is also the only minimal codeword.

[$\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$] is a codeword if and only if

$$\mathbf{v}_2 + \mathbf{v}_3 + \mathbf{v}_1 = \mathbf{0}$$

Suppose a random choice of $\mathbf{v}_1$ and $\mathbf{v}_2$ gives

$\mathbf{v}_1$ = [1 0 0 1 0 1 1 0 0 1]

$\mathbf{v}_2$ = [0 0 1 1 0 1 0 1 1 1].

Then $\mathbf{v}_3$ is determined as

$\mathbf{v}_3$ = [**1 0 1 0 0 0 1 1 1 0**].

This is the scheme that was used by the two crooks.

Summarizing for **a secret-sharing system determined by a linear q-ary (n, k) code** in the manner that
(i) the secret is chosen as the first digit of a codeword;
(ii) the n - 1 shares are the other codeword digits;
(ii) the digits in k - 1 positions, selected so that together with the first position they form an information set, are chosen uniformly at random over GF(q) and the codeword then computed.

Proposition: The minimal sets of shares giving access to the secret correspond to the minimal codewords in the **dual code** whose first component is a 1 in the manner that each minimal access set is the set of shares corresponding to those other locations where the corresponding minimal codeword is non-zero.

J. L. Massey, "Minimal Codewords and Secret Sharing," pp. 276-279 in Proc. 6th Joint Swedish-Russian Int. Workshop on Info. Theory, 1993.

A linear **(n, k)** code is **maximum-distance separable** (**MDS**) is its minimum distance satisfies **d** = **n - k +1**.

Some properties of a **linear (n, k) MDS code**.

- Every set of k code positions is an **information set**.
- For every choice of n - k +1 coordinates, there is a codeword that is **non-zero in these and only these coordinates**. The subset of these codewords whose leftmost non-zero component is a 1 are **all and only the minimial codewords** of the linear (n, k) MDS code.
- The **dual code** is an **(n, n - k) MDS code**.

This is why the **dual** of a linear **(n, k) MDS** code corresponds to a **threshold secrecy-sharing scheme** with T = **k** - 1.

N.B. Reed-Solomon codes are linear MDS codes.

Q: How can you tell whether a desired access structure can be realized with a linear code?

I don’t know.

Q: Suppose you are given an access structure that can be realized with a linear code. How do you find a linear code that realizes this access structure?

I don’t know.