Coding & Cryptography

March 11, 2020

0 Communication Channels

This course will be about modelling communication. In general, we have the following idea:

$$\begin{tikzcd} \text{Source} \arrow[r] & \text{Encoder} \arrow[r, rightsquigarrow, "\text{Channel}", "\text{Errors, Noise}"'] & \text{Decoder} \arrow[r] & \text{Receiver} \end{tikzcd}$$

For example, the channel might be an optical or electrical telegraph, modems, audio CDs, satellite relays. The encoding and decoding might be something like ASCII, so that each character in the email "Call at 2pm" would be encoded into 8 bits using $a = 1100101, \ldots$, giving an 84 bit message to be transmitted via the internet, and decoded by the receiver's email client. Our general aim here will be, given some source and channel (modelled probabilistically), to design an encoder and decoder to send messages economically and reliably.

Examples

- (Noiseless coding) Morse Code. In this code, more common letters are assigned shorter codes, so that we have $A = \cdot - *, E = \cdot *, Q = - - \cdot - *, Z = - - \cdot \cdot *$. This is adapted to the *source*, in the sense that we chose the codes based off the expected distribution of letters that we will have to transmit.

- (Noisy coding) ISBN. In the ISBN encoding, every book is given a 10 digit number $a_1 a_2 \ldots a_{10}$, with $\sum_{i=1}^{10} (11 - i) a_i \equiv 0 \mod 11$. This is adapted to the *channel*, in the sense that the likely errors to occur will be 1 incorrect digit, or accidentally transposing two digits, which this code is resistant to (will return an error rather than an erroneous result).

A ***communication channel*** accepts symbols from some alphabet $\mathscr{A} = \{a_1, a_2, \ldots, a_r\}$ (e.g. $\{0,1\}, \{a, b, \ldots, z\}$), and outputs symbols from an alphabet $\mathscr{B} = \{b_1, \ldots, b_s\}$. The channel is modelled by the probabilities:

$$\mathbb{P}(y_1, y_2, \ldots, y_n \text{ received} | x_1, x_2, \ldots, x_n \text{ sent}) = \textstyle\prod_{i=1}^n \mathbb{P}(y_i \text{ received} | x_i \text{ sent})$$

A ***discrete memoryless channel (DMC)*** is a channel with $p_{ij} = \mathbb{P}(b_j \text{ received} | a_i \text{ sent})$ the same for each channel usage and independent of any past or future channel usages.

The ***channel matrix*** is $P = (p_{ij})$, an $r \times s$ stochastic matrix.

Examples: The ***binary symmetric channel (BSC)*** with error probability $p \in [0,1]$ has $\mathscr{A} = \mathscr{B} = \{0,1\}$. The channel matrix is $\begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix}$. A symbol is transmitted correctly with probability $1-p$.

The ***binary erasure channel*** has input alphabet $\{0,1\}$, and output alphabet $\{0,1,*\}$, where we miss a bit is probability p, giving channel matrix $\begin{pmatrix} 1-p & 0 & p \\ 0 & 1-p & 0 \end{pmatrix}$. We can model n uses of a channel by the ***n^{th} extension*** with input alphabet $\mathscr{A}^n$, and output alphabet $\mathscr{B}^n$.

A ***code*** c of ***length*** n is a function $c : M \to \mathscr{A}^n$ where M is the set of all possible messages. Implicitly, we also have a decoding rule $\mathscr{B}^n \to M$.
The ***size*** of c is $m = |M|$.
The ***information rate*** is $\rho(c) = \frac{1}{n}\log_2(m)$.
The ***error rate*** is $\widehat{e}(c) = \max_{x\in M}\{\mathbb{P}(\text{error}|x \text{ sent})\}$.

A channel can transmit reliably at a rate R if there exists a sequence of codes $(c_n : n \geq 1)$ with c_n a code of length n, $\lim_{n\to\infty}(\rho(c_n)) = R, \lim_{n\to\infty}(\widehat{e}(c_n)) = 0$. The capacity of a channel is the supremum of all reliable transmission rates.

Theorem 0.1. *A BSC with error probability $p < \frac{1}{2}$ has a non-zero capacity (i.e. good codes exist).*

Proof. See **9.3** □

1 Noiseless Coding

1.1 Prefix-free Codes

For an alphabet $\mathscr{A}, |\mathscr{A}| < \infty$, let $\mathscr{A}^* = \bigcup_{n\geq 0}\mathscr{A}^n$, the set of all finite strings from $\mathscr{A}$. The ***concatenation*** of strings $x = x_1 \dots x_r$ and $y = y_1 \dots y_s$ is $xy = x_1 \dots x_r y_1 \dots y_s$.

Let $\mathscr{A}, \mathscr{B}$, be alphabets. A ***code*** is a function $c : \mathscr{A} \to \mathscr{B}^*$. The strings $c(a)$ for $a \in \mathscr{A}$ are called ***codewords*** (cws). If $x, y \in \mathscr{B}^*$ then x is a ***prefix*** of y if $y = xz$ for some $z \in \mathscr{B}^*$.

For example, we have the Greek fire code, found in the writings of Polybius around 280 BC. $\mathscr{A} = \{\alpha, \beta, \dots, \omega\}, \mathscr{B} = \{1,2,3,4,5\}$, with code $\alpha \mapsto 11, \beta \mapsto 12, \dots, \psi \mapsto 53, \omega \mapsto, 54$, where xy means "x torches held up, and another y torches nearby".

The English language is even a code: we can let $\mathscr{A}$ be words in a given dictionary, and $\mathscr{B} = \{a, b, \dots, z, \sqcup\}$, where the coding function is to spell the word and follow it with a space.

We send a message $x_1 \dots x_n \in \mathscr{A}^*$ as $c(x_1)\dots c(x_n) \in \mathscr{B}^*$. So c extends to a function $c^* : \mathscr{A}^* \to \mathscr{B}^*$.

c is ***decipherable/decidable*** if c^* is injective, so that each string in $\mathscr{B}^*$ could have come from at most one message. Note that it isn't sufficient to just have c injective, although clearly this is necessary:

$\mathscr{A} = \{1,2,3,4\}, \mathscr{B} = \{0,1\}, c : 1 \mapsto 0, 2 \mapsto 1, 3 \mapsto 00, 4 \mapsto 01$. Then $c^*(114) = 0001 = c^*(312)$.

If $|\mathscr{A}| = m, |\mathscr{B}| = a$, then we say c is an ***a-ary code of size m***. 2-ary = ***binary***, 3-ary=***ternary***.

We aim to construct decipherable codes with short word lengths. Assuming c is injective, the following are always decipherable:

- Block codes, where every codeword has the same length (e.g. Greek fire, ASCII)

- Comma codes, where we have an "end of word" character (e.g. English language)
- Prefix-free codes, where no codeword is a prefix of any other distinct words.

Note that both of the first two are special cases of prefix-free codes. Prefix-free codes are often called ***instantaneous*** or ***self-punctuating*** codes. Note that not all decipherable codes are prefix-free: $0 \mapsto 01, 1 \mapsto 011$ is decipherable but not prefix free.

Theorem 1.1 (Kraft's Inequality)**.** *Let $|\mathscr{A}| = m, |\mathscr{B}| = a$. A prefix-free code $c : \mathscr{A} \to \mathscr{B}^*$ with word lengths $\ell_1, \dots, \ell_m$ exists if and only if:*

$$\sum_{i=1}^{m} a^{-\ell_i} \leq 1 \tag{$*$}$$

Proof. Rewrite $(*)$ as $\sum_{\ell=1}^{s} n_\ell a^{-\ell} \leq 1(\star)$, where n_ℓ is the number of codewords of length ℓ and $s = \max_{1 \leq i \leq m} \ell_i$.

$\Longrightarrow$ If $c : \mathscr{A} \to \mathscr{B}^*$ is prefix-free, then $n_1 a^{s-1} + n_2 a^{s-2} + \dots n_s \leq a^s$, since the LHS is the number of strings of length s in $\mathscr{B}$ with some codeword of c as a prefix, and RHS is the number of strings of length s. Dividing by a^s gives $(\star)$.

$\Longleftarrow$ Given $n_1, \dots, n_s$ satisfying $(\star)$, we need to construct a prefix-free code c with n_ℓ codewords of length ℓ for all $\ell \leq s$. We use induction on s. The case $s = 1$ is clear: we have $(\star)$ gives $n_1 \leq a$, so we can choose a code.

By the induction hypothesis there is a prefix-free code $\widehat{c}$ with n_ℓ codewords of length ℓ for all $\ell \leq s - 1$. Then $(\star)$ gives:

$$n_1 a^{s-1} + n_2 a^{s-2} + \ldots + n_{s-1} a + n_s \leq a^s$$

where the first $s - 1$ terms on LHS sum to the number of strings of length s with some codeword of $\widehat{c}$ as a prefix, and the RHS is the number of strings of length s. Hence we can add at least n_s new codewords of length s to $\widehat{c}$ and maintain the prefix-free property, giving our code.

□

Theorem 1.2 (McMillan)**.** *Any decipherable code satisfies Kraft's inequality*

Proof (Karush). Let $c : \mathscr{A} \to \mathscr{B}^*$ be a decipherable code with codewords of lengths $\ell_1, \dots, \ell_m$. Let $s = \max_{1 \leq i \leq m} \ell_i$. Then for $R \in \mathbb{N}$:

$$\left(\sum_{i=1}^{m} a^{-\ell_i} \right)^R = \sum_{l=1}^{Rs} b_\ell a^{-\ell}$$

where $b_\ell = |\{x \in \mathscr{A}^R : c^*(x) \text{ has length } \ell\}| \leq |\mathscr{B}^\ell| = a^\ell$, using the fact that c^* is injective. Then:

$$\left(\sum_{i=1}^{m} a^{-\ell_i} \right)^R \leq \sum_{l=1}^{R} a^\ell a^{-\ell} = Rs$$
$$\sum_{i=1}^{m} a^{-\ell_i} \leq (Rs)^{\frac{1}{R}} \to 1 \text{ as } R \to \infty$$

□

Corollary 1.3. *A decipherable code with prescribed word lengths exists iff a prefix-free code with same word lengths exists.*

Proof.

$\Longrightarrow$ Use **1.2** to generate a prefix-free code by **1.1**

$\Longleftarrow$ Prefix-free codes are decipherable.

$\square$

2 Shannon's Noiseless Coding Theorem

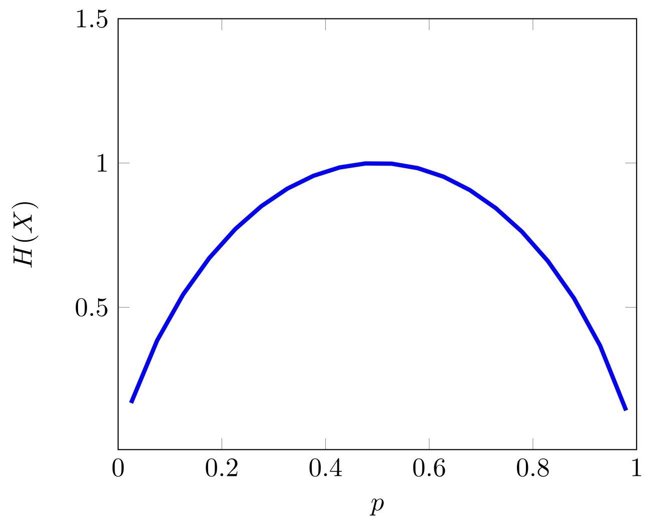
Entropy is a measure of 'randomness' or 'uncertainty'. Suppose we have a random variable X that takes values $x_1, \dots, x_n$ with probabilities $p_1, \dots, p_n$. Then the ***entropy*** (roughly speaking) is the expected number of fair coin tosses needed to simulate X.

Examples:

- $p_1 = p_2 = p_3 = p_4 = \frac{1}{4}$. We can identify $\{x_1, x_2, x_3, x_4\}$ with $\{HH, HT, TH, TT\}$, and so the entropy of this random variable is 2.
- $(p_1, p_2, p_3, p_4) = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$. Here, the entropy is $1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} = \frac{7}{4}$. We might say then, since the entropy is greater, that the first example is "more random" than the second.

More concretely, the ***Shannon (or information) entropy*** of X is $H(X) = -\sum_{i=1}^n p_i \log_2 p_i$. Note that $H(X) \geq 0$ with equality if and only if $\mathbb{P}(X = x_i) = 1$. This is measured in ***bits***. We take the convention that $0 \log 0 = 0$.

Example: Consider a biased coin, where $\mathbb{P}(X = H) = p, \mathbb{P}(X = T) = 1 - p$. Then $H(X) = -p \log p - (1-p) \log(1-p) = p \log(\frac{1-p}{p} - \log(1-p)$.



Proposition 2.1 (Gibb's Inequality). *Let $(p_1, \ldots, p_n)$ and $(q_1, \ldots, q_n)$ be probability distributions. Then:*

$$-\sum_{i=1}^{n} p_i \log p_i \leq -\sum_{i=1}^{n} p_i \log q_i$$

with equality if and only if $p_i = q_i$ for all i.

Proof. Since $\log x = \frac{\ln x}{\ln 2}$, we may replace log by ln in the proof. Put $I = \{1 \leq r \leq n : p_i \neq 0\}$. Now $\ln x \leq x - 1$ with equality if and only if $x = 1$. So we have $\ln \frac{q_i}{p_i} \leq \frac{q_i}{p_i} - 1$, and hence:

$$\begin{aligned}
\sum_{i \in I} p_i \ln \frac{q_i}{p_i} &\leq \sum_{i \in I} q_i - \sum_{i \in I} p_i \\
&= \sum_{i \in I} q_i - 1 \leq 0 \\
\therefore -\sum_{i \in I} p_i \ln p_i &\leq -\sum_{i \in I} p_i \ln q_i \\
\therefore -\sum_{i=1}^{n} p_i \log p_i &\leq -\sum_{i=1}^{n} p_i \log q_i
\end{aligned}$$

If equality holds, then $\sum_{i \in I} p_i = 1$ and $\frac{p_i}{q_i} = 1$ for all $i \in I$, so $p_i = q_i$ □

Corollary 2.2. *$H(p_1, \ldots, p_n) \leq \log n$ with equality if and only if $p_1 = \ldots = p_n = \frac{1}{n}$.*

Proof. Take $q_1 = \ldots = q_n = \frac{1}{n}$ in **2.1**. □

Let $\mathscr{A} = \{\mu_1, \ldots, \mu_m\}$, and $|\mathscr{B}| = a$, where $m, a \geq 2$. The random variable X takes values $\mu_1, \ldots, \mu_m$ with probabilities $p_1, \ldots, p_m$. We say a code $c : \mathscr{A} \to \mathscr{B}^*$ is ***optimal*** if it is a decipherable code with smallest possible expected word length, $\mathbb{E}S = \sum_i p_i \ell_i$.

Theorem 2.3 (Shannon's Noiseless Coding Theorem). *The expected word length $\mathbb{E}S$ of an optimal code satisfies:*

$$\frac{H(X)}{\log a} \leq \mathbb{E}S < \frac{H(X)}{\log a} + 1$$

Proof. For the lower bound, take $c : \mathscr{A} \to \mathscr{B}^*$ decipherable with word lengths $\ell_1, \ldots, \ell_m$. Then set $q_i = \frac{a^{-\ell_i}}{D}$ where $D = \sum_{i=1}^{m} a^{-\ell_i}$. Now we have that $\sum_{i=1}^{m} q_i = 1$. By Gibbs,

$$\begin{aligned}
H(X) &\leq -\sum_{i=1}^{m} p_i \log q_i \\
&= -\sum p_i \left(-\ell_i \log a - \log D\right) \\
&= \left(\sum_{i=1}^{m} p_i \ell_i\right) \log a + \log D
\end{aligned}$$

By McMillan, $D \leq 1$, so $\log D \leq 0$, and so $H(X) \leq (\sum_{i=1}^{m} p_i \ell_i) \log a = \mathbb{E}S \cdot \log a$, and we have equality if and only if $p_i = a^{-\ell_i}$ for some integers $\ell_1, \ldots \ell_m$.

For the upper bound, take $\ell_i = \lceil -\log_a p_i \rceil$. Then $-\log_a p_i \le \ell_i \implies p_i \ge a^{-\ell_i}$.

Now $\sum_{i=1}^m a^{-\ell_i} \le \sum_{i=1}^m p_i = 1$. By Kraft, there is some prefix-free code c with word lengths $\ell_1, \dots, \ell_m$, and the expected word length of c is $\mathbb{E}S = \sum p_i \ell_i < \sum p_i(-\log_a p_i + 1) = \frac{H(X)}{\log a} + 1$. $\square$

Example: ***Shannon-Fano coding***
We mimic the above proof: given probabilities $p_1, \dots, p_n$, set $\ell_i = \lceil -\log_a p_i \rceil$. Construct the prefix-free code with word lengths $\ell_1, \dots, \ell_m$ by choosing in order of increasing length, ensuring that previous codewords are not prefixes. For example, if $a = 2, m = 5$ we have:

i	p_i	$\lceil -\log_2 p_i \rceil$	Codewords
1	0.4	2	00
2	0.2	3	010
3	0.2	3	011
4	0.1	4	1000
5	0.1	4	1001

$\mathbb{E}S = \sum p_i \ell_i = 2.8$, entropy $= 2.12$

3 Huffman Coding Algorithm

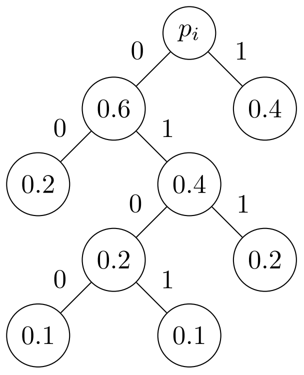
Huffman was a student of Fano, and was thinking about how to construct an optimal code. For simplicity, we will take $a = 2$. Suppose we get messages with orders $p_1 \ge p_2 \ge \dots \ge p_m$. Huffman gave a recursive definition of codes that we can prove are optimal. If $m = 2$, then take codewords 0 and 1.

If $m > 2$, we first have a Huffman code for messages $\mu_1, \dots, \mu_{m-2}, \nu$ with probabilities $p_1, \dots, p_{m-2}, p_{m-1} + p_m$, then append 0 and 1 to give codewords for μ_{m-1} and μ_m.

Note:

- Huffman codes are prefix-free.
- We have some choices to make if some of the p_j are equal, so Huffman codes are not unique.

Example: Reconsider the previous example:



i	p_i	Codewords
1	0.4	1
2	0.2	00
3	0.2	011
4	0.1	0100
5	0.1	0101

This code has expected length 2.2, which is less than Shannon-Fano gave.

Theorem 3.1 (Huffman, 1952)**.** *Huffman codes are optimal.*

Proof. We show this by induction on m. The case of $m = 2$ is trivial. For $m > 2$, let c_m be a Huffman code for source X_m which takes values $\mu_1, \dots, \mu_m$ with probabilities $p_1 \geq \dots \geq p_m$. Then c_{m-1} is constructed from a Huffman code c_{m-1} for values $\mu_1, \dots, \mu_{m-1}, \nu$ with probabilities $p_1, \dots, p_{m-2}, p_{m-1} + p_m$.

Observe that $\mathbb{E}S_m = \mathbb{E}S_{m-1} + p_{m-1} + p_m$ by construction of c_m from c_{m-1}.

Now let c'_m be an optimal code for X_m. Without loss of generality, we may take c'_m to be prefix-free and the last two codewords of c'_m have maximal length and differ only in the last digit (see **3.2** below). Say $c'_m(\mu_{m-1}) = y0, c'_m(\mu_m) = y1$ for some $y \in \{0,1\}^*$.

Let c'_{m-1} be the prefix free code for X_{m-1} given by $c'_{m-1}(\mu_i) = c'_m(\mu_i), c'_{m-1}(\nu) = y$.

Then the expected word length is $\mathbb{E}S'_m = \mathbb{E}S'_{m-1} + p_{m-1} + p_m \geq \mathbb{E}S_{m-1} + p_{m-1} + p_m = \mathbb{E}S_m$ by the inductive hypothesis, and so c_m is optimal. $\square$

Lemma 3.2. *Suppose messages $\mu_1, \dots, \mu_m$ are sent with probabilities $p_1, \dots, p_m$, with an optimal code c with word lengths $\ell_1, \dots, \ell_m$. Then:*

1. *If $p_i > p_j$ then $\ell_i \leq \ell_j$.*
2. *Among all codewords of maximal length, there are two that differ only in the last digit.*

Proof. Otherwise, modify c by swapping the i^{th} and j^{th} codewords, or deleting the last letter of each codeword of maximal length. The modified code is still prefix-free but has shorter expected word length, contradicting optimality of c. $\square$

4 Joint Entropy

If X, Y are random variables with value in $\mathscr{A}$ and $\mathscr{B}$. Then (X, Y) is also a random variable with entropy $H(X, Y)$, the ***joint entropy*** of X, Y.

$$H(X,Y) = -\sum_{x \in \mathscr{A}} \sum_{y \in \mathscr{B}} \mathbb{P}(X = x, Y = y) \log \mathbb{P}(X = x, Y = y)$$

We can of course generalise this to any finite number of random variables. We will use Gibb's (**2.1**) to prove:

Lemma 4.1. *Let X, Y be random variables taking values in $\mathscr{A}, \mathscr{B}$. Then:*

$$H(X,Y) \leq H(X) + H(Y)$$

with equality if and only if X and Y are independent.

Proof. Let $\mathscr{A} = \{x_1, \dots, x_m\}, \mathscr{B} = \{y_1, \dots, y_n\}$. Set $p_{ij} = \mathbb{P}(X = x_i, Y = y_i), p_i = \mathbb{P}(X = x_i), q_i = \mathbb{P}(Y = y_i)$. Then Gibb's inequality with $\{p_{ij}\}$ and $\{p_i q_j\}$ gives:

$$-\sum_{i,j} p_{ij} \log p_{ij} \le -\sum_{i,j} p_{ij} \log(p_i q_j) = -\sum_i \left(\sum_j p_{ij}\right) \log p_i - \sum_j \left(\sum_i p_{ij}\right) \log q_j$$
$$= -\sum_i p_i \log p_i - \sum_j q_j \log q_j$$

i.e. $H(X,Y) \le H(X) + H(Y)$, with equality if and only if $p_{ij} = p_i q_j$ for all i, j, i.e. when X, Y are independent. □

Example: Let X be a random variable that takes D values with probability $\frac{1}{D}$. Then $H(X) = \log_2(D)$. Suppose $X_1, \dots, X_N$ are i.i.d. with the same distribution as X. Then $H(X_1, \dots, X_N) = N \log_2 D$.

5 Error Correcting Codes

5.1 Noisy Channels and Hamming's Code

A ***binary [n,m]-code*** is a subset $C \subseteq \{0,1\}^n$ of ***size*** $m = |C|$, ***length*** n. The elements of C are called ***codewords***. We use an $[n,m]$-code to send one of m messages through a binary symmetric channel, making n uses of the channel. Clearly $1 \le m \le 2^n$, so $0 \le \frac{1}{n} \log m \le 1$. If $|C| = 1$ then $\rho(C) = 0$, and if $C = \{0,1\}^n$ then $\rho(C) = 1$.

For $x, y \in \{0,1\}^n$, the ***Hamming distance*** $d(x,y) = |\{i : 1 \le i \le n, x_i \ne y_i\}|$, i.e. the number of positions where x and y differ.

We have three possible decoding rules:

1. The ***ideal observer*** decoding rule decodes $x \in \{0,1\}^n$ as $c \in C$ maximising $\mathbb{P}(c \text{ sent}|x \text{ received})$.
2. Then ***maximum likelihood*** decoding rule decodes $x \in \{0,1\}^n$ as $c \in C$ maximising $\mathbb{P}(x \text{ received}|c \text{ sent})$.
3. The ***minimum distance*** decoding rule decodes $x \in \{0,1\}^n$ as $c \in C$ minimising $d(x,c)$.

Lemma 5.1.

1. *If all the messages are equally likely, then 1. and 2. agree.*
2. *If $p < \frac{1}{2}$, then 2. and 3. agree.*

Proof.

1. By Bayes' Rule:

$$\mathbb{P}(c \text{ sent}|x \text{ received}) = \frac{\mathbb{P}(c \text{ sent})}{\mathbb{P}(x \text{ received})} \mathbb{P}(x \text{ received}|c \text{ sent})$$

By Hypothesis, $\mathbb{P}(c \text{ sent})$ is independent of $c \in C$, and so for fixed x, maximising $\mathbb{P}(c \text{ sent}|x \text{ received})$ is the same as maximising $\mathbb{P}(x \text{ received}|c \text{ sent})$.

Let $r = d(x,c)$. Then $\mathbb{P}(x \text{ received}|c \text{ sent}) = p^r(1-p)^{n-r} = (1-p)^n\left(\frac{p}{1-p}\right)^r$. Since $p < \frac{1}{2}$, $\frac{p}{1-p} < 1$, and so maximising $\mathbb{P}(x \text{ received}|c \text{ sent})$ is the same as minimising $d(x,c)$. $\square$

For instance, suppose 000 is sent with probability $\frac{9}{10}$, and 111 with probability $\frac{1}{10}$, through a binary symmetric channel with error probability $\frac{1}{4}$. If we receive 110, the ideal receiver computes $\mathbb{P}(000 \text{ sent}|110 \text{ received}) = \frac{3}{4}; \mathbb{P}(110 \text{ sent}|110 \text{ received}) = \frac{1}{4}$, and so decodes it as 000. But the minimum distance (and so maximal likelihood) code is 111. Henceforth, we will decide to use minimal distance decoding.

Note that minimal distance decoding can be expensive in terms of time and storage if $|C$ is large, and we also need to specify a convention in the case of a tie (e.g. make a random choice, request the message again).

A code is ***d-error detecting*** if changing up to d digits in each codeword can never produce another codeword. It is ***e-error correcting*** if, knowing that $x \in \{0,1\}^n$ differs from some codeword in at most e places, we can deduce uniquely what the codeword is.

Examples

1. A ***repetition code*** of length n has codewords $00\dots0, 11\dots1$. This is an $[n,2]$-code. It is $(n-1)$ error detecting and $\lfloor\frac{n-1}{2}\rfloor$-error correcting. But the information rate is only $\frac{1}{n}$.
2. A ***simple parity check code*** or ***paper tape code***: identify $\{0,1\}$ with $\mathbb{F}_2$ (i.e. arithmetic modulo 2), and let $C = \{(x_1,\dots,x_n) \in \{0,1\}^n : \sum x_i = 0\}$. This is an $[n, 2^{n-1}]$-code. It is 1-error detecting, but cannot correct errors. Its information rate is $\frac{n-1}{n}$.
3. ***Hamming's Original Code*** is a 2-error detecting and 1-error correcting binary [7,16]-code:

$$C = \left\{ c \in \mathbb{F}_2^7 : \begin{array}{c} c_1 + c_3 + c_5 + c_7 = 0 \\ c_2 + c_3 + c_6 + c_7 = 0 \\ c_4 + c_5 + c_6 + c_7 = 0 \end{array} \right\}$$

The bits c_3, c_5, c_6, c_7 are arbitrary and c_1, c_2, c_4 are forced. The information rate is $\frac{4}{7}$.

Given $x \in \mathbb{F}_2^7$, we form the ***syndrome*** $z = (z_1, z_2, z_4) \in \mathbb{F}_2^7$, where $z_1 = x_1 + x_3 + x_5 + x_7$, $z_2 = x_2 + x_3 + x_6 + x_7$, $z_4 = x_4 + x_5 + x_6 + x_7$. If $x \in C$ then $z = (0,0,0)$. If $d(x,c) = 1$ for some $c \in C$ then x_i and c_i differ for $i = z_1 + 2z_2 + 4z_4$. This can be checked easily for $c = 0$ with a case by case check of the seven binary sequences of six 0s and one 1, e.g. $x = 0010000$ gives a syndrome $z = (1,1,0), i = 1 + 2 + 0 = 3$.

Lemma 5.2. *d is a metric on $\mathbb{F}_2^n$.*

Proof. Immediately, $d(x,y) \geq 0$, with equality if and only if $x = y$, and $d(x,y) = d(y,x)$. For the triangle inequality, note that if x and z differ at position i then either x, y differ at i or y, z differ at i. So every difference appearing in $d(x,z)$ appears in $d(x,y) + d(y,z)$, so $d(x,z) \leq d(x,y) + d(y,z)$. $\square$

Note that $d(x,y) = \sum_i d_1(x_i, y_i)$ where d_1 is the discrete metric on $\mathbb{F}_2$. We define the ***minimum distance*** of a code to be $\min_{c_1 \neq c_2} d(c_1, c_2)$.

Lemma 5.3. *Let C have minimal distance d. Then:*

1. *C is $(d-1)$-error detecting, but cannot detect all sets of d errors.*
2. *C is $\lfloor \frac{d-1}{2} \rfloor$-error correcting, but cannot correct all sets of $\lfloor \frac{d-1}{2} \rfloor + 1$ errors.*

Proof.

1. $d(c_1, c_2) \geq d$ for all distinct $c_1, c_2 \in C$. So C is $(d-1)$-error detecting. But $d(c_1, c_2) = d$ for some $c_1, c_2 \in C$. So C cannot detect all sets of errors.
2. Define the closed Hamming ball with center $x \in \mathbb{F}_2^n$, radius $r \geq 0$ as $B(x, r) = \{y \in \mathbb{F}_2^n : d(x, y) \leq r\}$. Now C is e-error correcting if and only if, for all $c_1 \neq c_2 \in C$, we have $B(c_1, e) \cap B(c_2, e) = \emptyset$.

   If $x \in B(c_1, e) \cap B(c_2, e)$, then $d(c_1, c_2) \leq d(c_1, x) + d(x, c_2) \leq 2e$. So if $d \geq 2e + 1$, then C is e-error correcting, with $e = \lfloor \frac{d-1}{2} \rfloor$. For the second part, take $c_1, c_2 \in C$ with $d(c_1, c_2) = d$. Then suppose $x \in \mathbb{F}_2^n$ differs from c_1 in e digits where c_1, c_2 differ too. Then $d(x, c_1) = e, d(x, c_2) = d - e$. If $d < 2e$ then $B(c_1, e) \cap B(c_2, d-e) \neq 0$, and so C cannot correct all sets of e-errors. Then take $e = \lceil \frac{d}{2} \rceil = \lfloor d - 1 \rfloor 2 + 1$.

□

As a point of a notation, an $[n, m]$-code with minimum distance d will be denoted as an $[n, m, d]$-code.

Examples:

1. Repetition of length n is an $[n, 2, n]$-code.
2. Simple parity check code of length n is an $[n, 2^{n-1}, 2]$-code.
3. Hamming's code is 1-error correcting, so $d \geq 3$. 0000000, 1110000 are both codewords, so it is a $[7, 16, 3]$-code, and hence 2-error correcting.

6 Covering Estimates

Denote $V(n, r) = |B(x, r)| = \sum_{i=0}^{r} \binom{n}{i}$, independent of $x \in \mathbb{F}_2^n$, as the ***volume*** of the ball (i.e. the number of points it contains).

Lemma 6.1 (Hamming's Bound)**.** *An e-error correcting code of length n has:*

$$|C| \leq \frac{2^n}{V(n, e)}$$

Proof. Suppose C is e-error correcting. Then $B(c_1, e) \cap B(c_2, e) = \emptyset$ for all $c_1 \neq c_2 \in C$. Then $\sum_{c \in C} |B(c, e)| \leq |\mathbb{F}_2^n| = 2^n$, i.e. $|C| V(n, e) \leq 2^n$. □

A code C of length n that can correct e errors is ***perfect*** if $|C| = \frac{2^n}{V(n,e)}$. Equivalently, a code is perfect if for all $x \in \mathbb{F}_2^n$ there is a unique $c \in C$ such that $d(x, c) \leq e$, or $\mathbb{F}_2^n = \bigcup_{c \in C} B(c, e)$, i.e. any $e + 1$ errors will make you decode incorrectly.

For example, Hamming's $[7, 16, 3]$-code is perfect, as $\frac{2^7}{7+1} = 2^4 = |C|$. Note that if $\frac{2^n}{V(n,e)} \notin \mathbb{Z}$ then there is no perfect e-error correcting code of length n, and even if $2^n / V(n, e)$ is an integer is may be the case that no perfect code exists.

Define $\boldsymbol{A(n,d)} = \max\{m : \exists\ [n,m,d]\text{-code}\}$. For instance, $A(n,1) = 2^n, A(n,n) = 2, A(n,2) = 2^{n-1}$.

Lemma 6.2. $A(n,d+1) \leq A(n,d)$

Proof. Let $m = A(n,d+1)$, and pick a code C with parameters $[n,m,d+1]$. Let $c_1, c_2 \in C$ with $d(c_1,c_2) = d+1$. Let c_1' differ from c_1 in exactly one of the places where c_1, c_2 differ. Then $d(c_1', c_2) = d$. If $c \in C \setminus \{c_1\}$, then $d(c,c_1) \leq d(c,c_1') + d(c_1', c_1) \implies d(c_1, c_1') \geq d.$

Replacing c_1 by c_1' gives an $[n,m,d]$-code i.e. $m \leq A(n,d)$. □

Corollary 6.3. *Equivalently,* $A(n,d) = \max\{m : \exists\ [n,m,d']-\textit{code for some } d' \geq d\}$.

Theorem 6.4.

$$\frac{2^n}{V(n,d-1)} \leq A(n,d) \leq \frac{2^n}{V(n, \lfloor \frac{d-1}{2} \rfloor)}$$

The lower bound is called the Gilbert-Shannon-Varshanov (GSV) bound, whilst the upper bound follows from Hamming's bound.

Proof of GSV. Let $m = A(n,d)$, and let C be a $[n,m,d]$-code. Then there cannot exist $x \in \mathbb{F}_2^n$ with $d(x,c) \geq d$ for all $c \in C$, otherwise we could replace C by $C \cup \{x\}$, contradicting maximality of $d(x,c)$. Hence $\mathbb{F}_2^n = \bigcup_{c \in C} B(c, d-1)$. Hence $2^n \leq mV(n,d-1)$. □

For example, take $n = 10, d = 3$. Then $V(n,2) = 56, V(n,1) = 11$, and so these bounds give $\frac{2^10}{56} \leq A(10,3) \leq \frac{2^10}{11}$, i.e. $19 \leq A(10,3) \leq 93$, but in fact we know computationally that it is between 72 and 79.

6.1 Asymptotics

We study $\frac{\log A(n, \lfloor n\delta \rfloor)}{n}$ as $n \to \infty$ to see how large the information rate can be for a given error rate.

Proposition 6.5. *Let* $0 < \delta < \frac{1}{2}$. *Then:*

1. $\log V(n, \lfloor n\delta \rfloor) \leq nH(\delta)$.
2. $\frac{1}{n} \log A(n, \lfloor n\delta \rfloor) \geq 1 - H(\delta)$.

Proof. Assuming *1.* we see by the GSV bound, $A(n, \lfloor n\delta \rfloor) \geq \frac{2}{V(n, \lfloor n\delta \rfloor - 1} \geq \frac{2^n}{V(n, \lfloor n\delta \rfloor)}$, so $\frac{\log A(n, \lfloor n\delta \rfloor)}{n} \geq 1 - \frac{\log V(n, \lfloor n\delta \rfloor)}{n} \geq 1 - H(\delta)$, and so *1.* $\implies$ *2.*

For *1.* observe $H(\delta)$ is increasing for $\delta \leq \frac{1}{2}$, so WLOG we can assume $n\delta \in \mathbb{Z}$. Then:

$$\begin{aligned}
1 &= (\delta + (1-\delta))^n \\
&= \sum_{i=0}^{n} \binom{n}{i} \delta^i (1-\delta)^{n-i} \\
&\geq \sum_{i=0}^{n\delta} \binom{n}{i} \delta^i (1-\delta)^{n-i} \\
&= (1-\delta)^n \sum_{i=0}^{n\delta} \binom{n}{i} \left(\frac{\delta}{1-\delta}\right)^i \\
&\geq (1-\delta)^n \sum \binom{n}{i} \left(\frac{\delta}{1-\delta}\right)^{n\delta} \\
&= \delta^{n\delta}(1-\delta)^{n(1-\delta)} V(n, n\delta) \\
0 &\geq n\delta \log \delta + n(1-\delta)\log(1-\delta) + \log V(n, n\delta) \\
0 &\geq -nH(\delta) + \log V(n, n\delta)
\end{aligned}$$

□

This constant $H(\delta)$ is the best possible, in the sense that:

Proposition 6.6.

$$\lim_{n\to\infty} \frac{\log V(n, \lfloor n\delta \rfloor)}{n} = H(\delta)$$

Proof. WLOG we may assume that $0 < \delta < \frac{1}{2}$. Let $0 \leq r \leq \frac{n}{2}$. Recall that $V(n,r) = \sum_{i=0}^{r} \binom{n}{i}$. Then:

$$\binom{n}{r} \leq V(n,r) \leq (r+1)\binom{n}{r} \tag{(*)}$$

From Stirling's approximation, $\log \binom{n}{r} = -r \log \frac{r}{n} - (n-r)\log \frac{n-r}{n} + \mathcal{O}(\log n) = nH(\frac{r}{n}) + \mathcal{O}(\log n)$.

Then from $(*)$, we have:

$$\begin{aligned}
H(\frac{r}{n}) + \mathcal{O}\left(\frac{\log n}{r}\right) \leq \frac{\log V(n,r)}{n} &\leq H\left(\frac{r}{n}\right) + \mathcal{O}\left(\frac{\log n}{n}\right) \\
\therefore \lim_{n\to\infty} \frac{\log V(n, \lfloor n\delta \rfloor)}{n} &= H(\delta)
\end{aligned}$$

□

7 New Codes from Old

Suppose C is an $[n, m, d]$-code. Then the ***parity check digit extension*** C^+ of C is the code of length $n+1$ given by:

$$C^+ = \{(c_1, \ldots, c_n, \sum_{i=1}^{n} c_i) : (c_1, \ldots, c_n) \in C\}$$

where the summation is done modulo 2. It is an $[n+1, m, d']$-code where $d' = d$ or $d+1$.

We can also delete the i^{th} digit from each codeword for $1 \le i \le n$, giving a ***truncated*** or ***punctured*** codeword (depending on if $i = n$ or $i < n$ respectively), called C^-, with parameters $[n-1, m, d']$ where $d - 1 \le d' \le d$.

Finally, given some $1 \le i \le n, \alpha \in \mathbb{F}_2$, we can create the ***shortened*** or ***punctured*** code C' of C is $\{(c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_n) : (c_1, \dots, c_{i-1}, \alpha, c_{i+1} \dots, c_n) \in C\}$. This has parameters $[n-1, m', d']$ with $d' \ge d$ and $m' \ge \frac{m}{2}$ for c a suitable choice of α.

8 AEP and Shannon's First Coding Theorem

A ***source*** is a sequence of random variables $X_1, X_2, \dots$ taking values in some alphabet $\mathscr{A}$. A source is ***Bernoulli*** or ***memoryless*** if $X_1, \dots$ are independently identically distributed (IID). A source $X_1, \dots$ is ***reliably encodable at rate r*** if there are subsets $A_n \subseteq \mathscr{A}^n$ such that:

1. $\lim_{n\to\infty} \frac{\log |A_n|}{n} = r$
2. $\lim_{n\to\infty} \mathbb{P}[(X_1, \dots, X_n) \in A_n] = 1$

The ***information rate*** H of a source is the infimum of all reliable encoding ratesso that $0 \le H \le \log |\mathscr{A}|$. Shannon's first coding theorem computes the information rate of certain sources, including Bernoulli sources.

8.1 Reminder from 1A Probability

A ***probability space*** is given by a triple $(\Omega, \mathscr{F}, \mathbb{P})$ where $\mathscr{F} \subset \mathcal{P}(\Omega)$ is a set of ***events*** and $\mathbb{P}$ is a ***probability measure***, and a ***random variable*** X is a function defined on Ω with some range. It has a ***probability mass function*** $p : x \mapsto \mathbb{P}(X = x)$

We say that a sequence of random variables $X_1, X_2, \dots$ ***converges in probability*** to $\lambda \in \mathbb{R}$ means that

$$\forall\, \epsilon > 0, \lim_{n\to\infty} \mathbb{P}(|X_n - \lambda| \ge \epsilon) = 0$$

We write $X_n \xrightarrow{\mathbb{P}} \lambda$ as $n \to \infty$.

Theorem 8.1 (Weak Law Of Large Numbers, WLLN)**.** *Let $X_1, X_2, \dots$ be* IID *discrete real-valued random variables with finite expected value μ. Then:*

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{\mathbb{P}} \mu \text{ as } n \to \infty$$

Proof. See Carne, theorem 10.3. □

Lemma 8.2. *The information rate of a Bernoulli source $X_1, X_2, \dots$ is at most the expected word length of an optimal code $c : \mathscr{A} \to \{0,1\}^*$ for X_i.*

Proof. Let $\ell_1, \ell_2, \dots$ be the lengths of codewords when we encode $X_1, X_2, \dots$ using c. Then given $\epsilon > 0$, let $A_n = \{x \in \mathscr{A}^n : c^*(x) \text{ has length } < n(\mathbb{E}[\ell_i] + \epsilon)\}$. Then:

$$\begin{aligned}
\mathbb{P}\left[(X_1, \dots, X_n) \in A_n\right] &= \mathbb{P}\left[\sum \ell_i < n(\mathbb{E}[\ell_i] + \epsilon)\right] \\
&\geq \mathbb{P}\left(\left|\frac{1}{n}\sum \ell_i - \mathbb{E}[\ell_i]\right| < \epsilon\right) \\
&\to 1 \text{ as } n \to \infty
\end{aligned}$$

Now c is decipherable so c^* is injective, and hence $|A_n| \leq 2^{n(\mathbb{E}[\ell_i]+\epsilon)}$. Making A_n larger if required, we may take $|A_n| = \lfloor 2^{n(\mathbb{E}[\ell_i]+\epsilon)} \rfloor$. Hence $\frac{\log |A_n|}{n} \to \mathbb{E}[\ell_i] + \epsilon$. So $X_1, X_2, \dots$ is reliably encodable at a rate $r = \mathbb{E}[\ell_i] + \epsilon$ for any $\epsilon > 0$, and hence the information rate is at most $\mathbb{E}[\ell_i]$. □

Corollary 8.3. *A Bernoulli source has information rate less than $H(X_1) + 1$.*

Proof. Use **8.2** and the Noiseless Coding theorem **2.3**. □

Now suppose we encode $X_1, X_2, \dots$ in blocks:

$$\underbrace{X_1, \dots, X_N}_{Y_1}, \underbrace{X_{N+1}, \dots, X_{2N}}_{Y_2}, \dots$$

such that $Y_1, Y_2, \dots$ take values in $\mathscr{A}^N$. We can check that if $X_1, X_2, \dots$ has information rate H, then $Y_1, Y_2, \dots$ has information rate NH.

Proposition 8.4. *The information rateH of a Bernoulli source $X_1, X_2, \dots$ is at most $H(X_1)$.*

Proof. Apply **8.3** to $Y_1, Y_2, \dots$ to get:

$$NH < H(Y_1) + 1 = H(X_1, \dots, X_N) + 1 = \sum_{i=1}^{N} H(X_i) + 1 = NH(X_i) + 1$$

i.e. $H < H(X_1) + \frac{1}{N}$ for all $N \geq 1$, and so $H \leq H(X_1)$. □

8.2 Typical Sequences

This content is nonexaminable, but is required to prove the examinable result that $H = H(X_1)$.

As a motivational example, toss a biased coin with head probability p, and let X_i be the outcome of the i^{th} flip. If we toss a large number, say N, times, we expect that we will get about pN heads and $(1-p)N$ tails. The probability of any particular sequence of pN heads and $(1-p)N$ tails is $p^{pN}(1-p)^{(1-p)N} = 2^{-NH(X)}$.

We say that a source $X_1, X_2, \dots$ satisfies the ***Asymptotic Equipartition Property (AEP)*** for some constant $H \geq 0$ if:

$$-\frac{1}{n}\log p(X_1, \dots, X_n) \to H \text{ as } n \to \infty$$

Lemma 8.5. *The AEP for a source $X_1, x_2, \dots$ is equivalent to the following:*
$\forall\, \epsilon > 0 \exists\, n_0(\epsilon)$ *s.t.* $\forall\, n \geq n_0(\epsilon) \exists\, T_n \subseteq \mathscr{A}^n$ *s.t.*

- $P[(X_1, \dots, X_n) \in T_n] > 1 - \epsilon$
- $\forall\ (x_1, \dots, x_n) \in T_n, 2^{-n(H+\epsilon)} \le p(x_1, \dots, x_n) \le 2^{-n(H-\epsilon)}$

The T_n are called **typical sets***, and the $(x_1, \dots, x_n) \in T_n$ are* **typical sequences***.*

Proof. If $(x_1, \dots, x_n) \in \mathscr{A}^n$ then we have the following equivalence:

$$2^{-n(H+\epsilon)} \le p(x_1, \dots, x_n) \le 2^{-n(H-\epsilon)} \iff |-\frac{1}{n}\log p(x_1, \dots, x_n) - H| \le \epsilon \qquad (\dagger)$$

Both AEP and the claimed equivalent results say that $P((X_1, \dots, X_N)$ satisfies $\dagger) \to 1$ as $n \to \infty$. □

Theorem 8.6 (Shannon's First Coding Theorem)**.** *If a source $X_1, X_2, \dots$ satisfies the AEP with constant H then it has information rate H.*

Proof. Let $\epsilon > 0$ and let $T_n \subseteq \mathscr{A}^n$ be typical sets. Then for all $(x_1, \dots, x_n) \in T_n$:

$$p(x_1, \dots, x_n) \ge 2^{-n(H+\epsilon)} \implies 1 \ge |T_n| 2^{-n(H+\epsilon)} \implies \frac{\log |T_n|}{n} \le (H+\epsilon)$$

Taking $A_n = T_n$ in the definition of reliable encoding, we see that the source is reliably encodeable at rate $H + \epsilon$. As $\epsilon > 0$, the information rate is $\le H$.

Conversely, if $H = 0$ we're done, otherwise pick $0 < \epsilon < \frac{H}{2}$. Suppose for a contradiction that the source is reliably encodeable at rate $H - 2\epsilon$, say, with sets $A_n \subseteq \mathscr{A}^n$. Let $T_n \subseteq \mathscr{A}^n$ be typical sets. Then for all $(x_1, \dots, x_n) \in T_n, p(x_1, \dots, x_n) \le 2^{-n(H-\epsilon)}$

Hence $\mathbb{P}(A_n \cap T_n) \le 2^{-n(H-\epsilon)}$, and so $\frac{\log \mathbb{P}(A_n \cap T_n)}{n} \le (H - \epsilon) + \frac{\log |A_n|}{n} \xrightarrow{n \to \infty} -\epsilon$. So $\mathbb{P}(A_n \cap T_n) \to 0$ as $n \to \infty$. But $\mathbb{P}(T_n) \le \mathbb{P}(T_n \cap A_n) + \mathbb{P}(\mathscr{A}^n \setminus A_n) \to 0 + 0 = 0$ as $n \to \infty$, contradicting typicality of T_n. Hence the source cannot be reliably encoded at rate $H - 2\epsilon$, and so the information rate must be $\ge H$, and hence $= H$. □

9 Capacity and Shannon's Second Coding Theorem

Given a random variable X with mass function p_X, we can construct a new random variable $p(X) = p_X \circ X$, taking values in $[0, 1]$. Then $H(X) = \mathbb{E}(-\log p(X))$. For example, if X, Y are independent, then $p(X, Y) = p(X)p(Y)$, and so $-\log p(X, Y) = -\log p(X) - \log p(Y) \implies H(X, Y) = H(X) + H(Y)$.

Corollary 9.1. *A Bernoulli source $X_1, X_2, \dots$ has information rate $H(X_1) = H$.*

Proof.* $p(X_1, \dots, X_n) = p(X_1) \dots p(X_n)$, and hence we have:

$$-\frac{\log p(X_1, \dots X_n)}{n} = -\frac{1}{n}\sum_{i=1}^{n} \log p(X_i) \xrightarrow{\mathbb{P}} H(X_i)$$

by the weak law of large numbers, and using the fact that the X_i are i.i.d. Check as an exercise that the AE holds with constant $H(X_1)$ using the definition of convergence in probability. Hence by **8.6** we are done. □

Note that **8.4** gave us an information rate $\leq H(X_1)$, without the use of the AEP. The AEP can also be used for noiseless coding - we encode the typical sequences with a block code and the atypical sequences arbitrarily, since they rarely occur. Many sources of interest, not just Bernoulli sources, satisfy the AEP. Under suitable conditions, the sequence $\frac{1}{n}H(X_1, \dots, X_n)$ is decreasing and the AEP is satisfied with constant $H \lim_{n\to\infty} \frac{H(X_1,\dots,X_n}{n}$.

Consider a communication channel with input of alphabet $\mathscr{A}$, and output B. A code of length n is a subset $C \subseteq \mathscr{A}^n$. The ***error rate***

$$\widehat{e}(C) = \max \mathbb{P}(\text{error } | c \text{ sent})$$

The ***information rate*** is $\rho(C) = \frac{\log |C|}{n}$.

The channel can ***transmit reliably*** at rate R if there are codes $C_1, C_2, \dots$ with C_n of length n, and:

- $\lim_{n\to\infty} \rho(C_n) = R$.
- $\lim_{n\to\infty} \widehat{e}(C_n) = 0$.

The ***operational capacity*** is the supremum of all reliable transmission rates.

Assume a source has information rate r bits per symbol, and emits symbols at s symbols per second, whilst the channel has capacity R bits per transmission and can can transmit symbols at S transmissions per second. Usually $S = s = 1$. If $rs \leq RS$ then we can encode and transmit reliably, and if $rs > RS$ we cannot.

Proposition 9.2. *A binary symmetric channel with error probability $p < \frac{1}{4}$ has non-zero capacity.*

Proof. We use the *GSV* bound. Pick δ with $2p < \delta < \frac{1}{2}$. We claim reliable transmission rate of $R = 1 - H(\delta) > 0$.

Let C_n be a code of length n with minimum distance $\lfloor n\delta \rfloor$ of maximal size. Them $|C_n| = A(n, \lfloor n\delta \rfloor) \geq 2^{n(1-H(\delta)} = 2^{nR}$.

Replacing C_n by a subcode we can assume $|C_n| = \lfloor 2^{nR} \rfloor$ with minimum distance still $\geq \lfloor n\delta \rfloor$

Now, with minimum distance decoding, $\widehat{e}(C_n) \leq \mathbb{P}$(in n uses the BSC makes more than $\frac{n\delta - 1}{2}$ errors).

Pick $\epsilon > 0$ with $p + \epsilon < \frac{\delta}{2}$. For n sufficiently large we have that $\frac{n\delta-1}{2} = n(\frac{\delta}{2} - \frac{1}{2n}) > n(p + \epsilon)$.

Hence $\widehat{e}(C) \leq \mathbb{P}$(BSC makes more than $n(p+\epsilon)$ errors) $\to 0$, as we will see in the next lemma. $\square$

Lemma 9.3. *Let $\epsilon > 0$. A BSC with error probability p is used to transmit n digits. Then:*

$$\lim_{n\to\infty} \mathbb{P}(\textit{BSC makes at least } n(p+\epsilon) \textit{ errors}) = 0$$

Proof. If U_i is the Bernoulli random variable that digit i is mistransmitted. Then U_i are i.i.d. with probability p. So $\mathbb{E}[U_i] = p$. Then the probability we are interested in is $\mathbb{P}(\sum U_i \geq n(p+\epsilon)) \leq \mathbb{P}(|\frac{1}{n}\sum U_i - p| \geq \epsilon) \to 0$ by the WLLN. $\square$

9.1 Fano's Inequality

Let X, Y be random variables taking values in the alphabets $\mathscr{A}, \mathscr{B}$. Then:

- $H(X|Y = y) = -\sum_{x\in\mathscr{A}} \mathbb{P}(X = x|Y = y) \log \mathbb{P}(X = x|Y = y)$
- $H(X|Y) = \sum_{y\in\mathscr{B}} \mathbb{P}(Y = y)H(X|Y = y)$

Clearly $H(X|Y) \geq 0$.

Lemma 9.4.

$$H(X, Y) = H(X|Y) + H(Y)$$

Proof.

$$\begin{aligned}
H(X|Y) &= -\sum_{x\in\mathscr{A}}\sum_{y\in\mathscr{B}} \mathbb{P}(X = x|Y = y)\mathbb{P}(Y = y) \log \mathbb{P}(X - x|Y = y) \\
&= -\sum_{x\in\mathscr{A}} \mathbb{P}(X = x, Y = y) \log \frac{\mathbb{P}(X = x, Y = y}{\mathbb{P}(Y = y)} \\
&= -\sum_{(x,y)\in\mathscr{A}\times\mathscr{B}} \underbrace{\mathbb{P}(X = x, Y = y) \log \mathbb{P}(X = x, Y = y)}_{H(X,Y)} + \underbrace{\mathbb{P}(X = x, Y = y) \log \mathbb{P}(Y = y)}_{-H(Y)} \\
&= H(X, Y) - H(Y)
\end{aligned}$$

□

Corollary 9.5. *$H(X|Y) \leq H(X)$ with equality if and only if X, Y are independent.*

Proof. Combine **4.1** with **9.4**. □

We can replace X, Y by random variables $X_1, X_2, \ldots, X_r$ and $Y_1, Y_2, \ldots, Y_s$, and similarly define $H(X_1, \ldots, X_r|Y_1, \ldots, Y_s)$[1].

Lemma 9.6.

$$H(X|Y) \leq H(X|Y, Z) + H(Z)$$

Proof.

$$\begin{aligned}
H(X, Y, Z) &= H(Z|X, Y) + H(X, Y) = H(Z|X, Y) + H(X|Y) + H(Y) \\
H(X, Y, Z) &= H(X|Y, Z) + H(Y, Z) = H(X|Y, Z) + H(Z|Y) + H(Y) \\
\therefore H(X|Y) &= -H(Z|X, Y) + H(X|Y, Z) + H(Z|Y) \\
&\leq H(X|Y, Z) + H(Z)
\end{aligned}$$

□

Proposition 9.7 (Fano's Inequality)**.** *Let X, Y be random variables taking values in $\mathscr{A}$, where $|\mathscr{A}| = m$. Let $p = \mathbb{P}(X \neq Y)$. Then:*

$$H(X|Y) \leq H(p) + p \log(m - 1)$$

[1] $H(X, Y|Z)$ means "the entropy of (X and Y) given Z", NOT "the entropy of X and (Y given Z)"

Proof. Let $Z = \begin{cases} 0 & X = Y \\ 1 & X \neq Y \end{cases}$. Then $\mathbb{P}(Z = 0) = 1 - p, \mathbb{P}(Z = 1) = p$. So $H(Z) = H(p)$. By **9.6**,

$$H(X|Y) \leq H(p) + H(X|Y, Z) \tag{$*$}$$

Then we have two cases:

$Z = 0$: We must have $X = y$, so $H(X|Y = y, Z = 0) = 0$.

$Z = 1$: Just $m - 1$ remaining possibilities for X, so $H(X|Y = y, Z = 1) \leq \log(m - 1)$.

Hence we have:

$$\begin{aligned} H(X|Y, Z) &= \sum_{y,z} \mathbb{P}(Y = y, Z = z) H(X|Y = y, Z = z) \\ &\leq \sum_{y} \mathbb{P}(Y = y, Z = 1) \log(m - 1) \\ &= \mathbb{P}(Z = 1) \log(m - 1) = p \log(m - 1) \end{aligned}$$

Then by $(*)$, $H(X|Y) \leq H(p) + p \log(m - 1)$. □

We will apply this result later when X takes values in $\mathscr{A}$ and Y is the result of passing codewords through a channel and then decoding, where p will be the probability of error.

If X, Y are random variables, then we define the ***mutual information*** of X and Y to be:

$$I(X; Y) \coloneqq H(X) - H(X|Y)$$

By **4.1** and **9.4**, $I(X; Y) = H(X) + H(Y) - H(X, Y) \geq 0$, with equality if and only if X and Y are independent. We also see from this form that $I(X; Y) = I(Y; X)$.

Given a DMC with input alphabet $\mathscr{A}$ of size m, and output alphabet $\mathscr{B}$. Let X be a random variable taking values in $\mathscr{A}$ used as an input to the channel. Let Y be the random variable output, depending on both X and the channel matrix.

We define the ***information capacity*** to be the $\max_X I(X; Y)$, where the maximum is taken over all probability distributions $(p_1, \ldots, p_m)$ for X. Since the space of random variables for X is a closed and bounded subset of $\mathbb{R}^m$, by Heine-Borel it is compact and, since I is continuous, the maximum is attained. Note that the information capacity depends only on the channel matrix.

Theorem 9.8 (Shannon's Second Coding Theorem)**.** *For a DMC, the operational capacity is equal to the information capacity.*

We will prove $\leq$ in general, and $\geq$ for a binary symmetric channel only.

For example, with a BSC with error probability p, input X, output Y, then $\mathbb{P}(X = 0) = \alpha, \mathbb{P}(X = 1) = 1 - \alpha$, and so $\mathbb{P}(Y = 0) = \alpha(1 - p) + (1 - \alpha)p; \mathbb{P}(Y = 1) = (1 - \alpha)(1 - p) + \alpha p$.

Then $C = \max_\alpha I(X; Y) = \max_\alpha [H(\alpha(1 - p) + (1 - \alpha)p) - H(p)] = 1 - H(p$, where the max attained at $\alpha = \frac{1}{2}$. So the information capacity $C = 1 + p \log p + (1 - p) \log(1 - p)$. We can plot this on a graph:

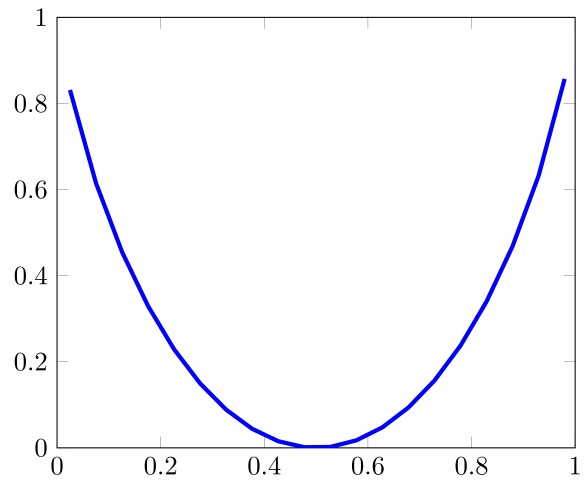


Figure 2: Capacity as a function of p for a binary symmetric channel

At $p = 0, 1$ the channel transmits perfectly, whilst at $p = \frac{1}{2}$ no information can be transmitted. We can choose to calculate $H(Y) - H(Y|X)$ or $H(X) - H(X|Y)$ to find the information - often one is easier than the other, for example with the binary erasure channel with erasure probability p.

$\mathbb{P}(X = 0) = \alpha, \mathbb{P}(X = 1) = 1 - \alpha, \mathbb{P}(Y = 0) = \alpha(1 - p), \mathbb{P}(Y = *) = p, \mathbb{P}(Y = 1) = (1 - \alpha)(1 - p)$

Then $H(X|Y = 0) = 0; H(X|Y = *) = H(\alpha); H(X|Y = 1) = 0$, and so $H(X|Y) = pH(\alpha)$. So $C = \max_\alpha(H(\alpha) - pH(\alpha)) = 1 - p$, where the maximum value is attained for $\alpha = \frac{1}{2}$.

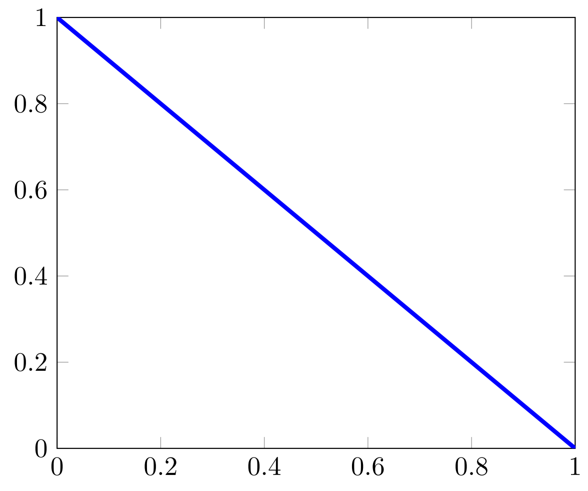


Figure 3: Capacity as a function of p for a binary erasure channel

We can now model using a channel n times, called the ***n^{th} extension***, i.e. replace input and output alphabets $\mathscr{A}, \mathscr{B}$ by $\mathscr{A}^n, \mathscr{B}^n$.

Lemma 9.9. *The n^{th} extension of a discrete memoryless channel with information capacity C has information capacity nC.*

Proof. The input $X_1, \dots, X_n$ determines the output $Y_1, \dots, Y_n$, and since the channel is memoryless, $H(Y_1, \dots, Y_n | X_1, \dots, X_n) = \sum_i H(Y_i | X_1, \dots, X_n) = \sum_i H(Y_i | X_i)$.

$$\begin{aligned} I(X_1, \dots, X_n; Y_1, \dots, Y_n) &= H(Y_1, \dots, Y_n) - H(Y_1, \dots, Y_n | X_1, \dots, X_n) \\ &= H(Y_1, \dots, Y_n) - \sum_{i=1}^n H(Y_i | X_i) \\ &\le \sum_{i=1}^n [H(Y_i) - H(Y_i | X_i)] \\ &= \sum_{i=1}^n I(X_i; Y_i) \le nC \end{aligned}$$

To finish, we must find $X_1, \dots X_n$ giving equality - take X_i to be i.i.d. with $I(X_i, Y_i) = C$, giving equality at the second $\le$. Then the Y_i are i.i.d. so we have equality at the first $\le$. So the maximum possible value is nC. □

Proposition 9.10. *For a DMC the operational capacity is at most the information capacity.*

Proof. Let C be the information capacity. Suppose that reliable transmission is possible at some $R > C$, i.e. there is a sequence $C_1, C_2, \dots$, with C_n of length n such that $\lim_{n \to \infty} \rho(C_n) = R$, and $\lim_{n \to \infty} \widehat{e}(C_n) = 0$.

We define the ***average error rate*** $e(C_n) = \frac{1}{|C_n|} \sum_{c \in C_n} \mathbb{P}(\text{error} | c \text{ sent})$, so that $e(C_n) \le \widehat{e}(C_n)$. Hence $e(C_n) \to 0$ as $n \to \infty$. Then let X be a random variable equidistributed in C_n. Transmit X and decode to obtain Y. So $e(C_n) = \mathbb{P}(X \ne Y)$. Then $H(X) = \log |C_n| = \log \lfloor 2^{nR} \rfloor \ge nR - 1$ for sufficiently large n. By Fano's inequality,

$$H(X|Y) \le 1 + e(C_n) \log(|C_n| - 1) \le 1 + e(C_n) n \rho(C_n)$$

By the previous proposition,

$$\begin{aligned} nC \ge I(X, Y) &= H(X) - H(X|Y) \\ &\ge \log |C_n| - (1 + e(C_n) n \rho(C_n)) \\ &= n\rho(C_n) - e(C_n) n \rho(C_n) - 1 \\ \therefore e(C_n) n \rho(C_n) &\ge n(\rho(C_n) - C) - 1 \\ e(C_n) &\ge \frac{\rho(C_n) - C}{\rho(C_n)} - \frac{1}{n\rho(C_n)} \to \frac{R - C}{R} \end{aligned}$$

But $R > C$, so $e(C_n) \not\to 0$ as $n \to \infty$. $\lightning$ □

Proposition 9.11. *Consider the Binary Symmetric Channel with error probability p. Let $R < 1 - H(p)$. Then there exists a sequence of codes $C_1, C_2, \dots$, with C_n of length n and size $\lfloor 2^{nR} \rfloor$, such that:*

$$\lim_{n \to \infty} \rho(C_n) = R$$
$$\lim_{n \to \infty} e(C_n) = 0$$

Note this is $e(C_n)$, not $\widehat{e}(C_n)$.

Proof. We use the method of random codes.

Without loss of generality, assume $p < \frac{1}{2}$. Let $\epsilon > 0$ be such that $p+\epsilon < \frac{1}{2}$ and $R < 1-H(p+\epsilon)$. This is always possible since H is continuous. Let $m = \lfloor 2^{nR} \rfloor$ and pick $C \in \mathscr{C} = \{[n, m]-\text{binary codes}\}$ and random. Note $|\mathscr{C}| = \binom{2^n}{m}$. Let $\mathscr{X}$ be a random variable equidistributed throughout $\mathscr{C}$, say $\mathscr{X} = \{X_1, \dots, X_m\}$ where the X_i are random variables taking values in $\mathbb{F}_2^n$, such that:

$$\mathbb{P}(X_i = x_i | \mathscr{X} = C) = \begin{cases} \frac{1}{m} & x_i \in C \\ 0 & \text{otherwise} \end{cases}$$

Notice that $\mathbb{P}(X_2 = x_2 | X_1 = x_1) = \begin{cases} \frac{1}{2^n-1} & x_1 \neq x_2 \\ 0 & \text{otherwise} \end{cases}$

Send $X = X_1$ through the BSC, receive Y, and decode to obtain Z. Under the minimum distance decoding, $\mathbb{P}(X \neq Z) = \frac{1}{|\mathscr{C}|}\sum_{C \in \mathscr{C}} e(C)$.

It then suffices to show that $\mathbb{P}(X \neq Z) \to 0$ as $n \to \infty$. Let $r = \lfloor n(p+2) \rfloor$.

$$\begin{aligned} \mathbb{P}(X \neq Z) &\leq \mathbb{P}(B(Y,r) \cap \mathscr{X} \neq \{X\}) \\ &= \underbrace{\mathbb{P}(X \notin B(Y,r))}_{(i)} + \underbrace{\mathbb{P}(B(Y,r) \cap \mathscr{X} \supsetneq \{X\})}_{(ii)} \end{aligned}$$

(i): $\mathbb{P}(X \notin B(Y,r)) = \mathbb{P}(\text{BSC makes more than } r \text{ errors}) \to 0$ as $n \to \infty$ by **9.3**.

(ii):

$$\begin{aligned} \mathbb{P}(B(Y,r) \cap \mathscr{X} \supsetneq \{X\}) &\leq \sum_{i=2}^{m} \mathbb{P}(X_i \in B(Y,r) \text{ and } X_1 \in B(Y,r)) \\ &\leq \sum_{i=2}^{m} \mathbb{P}(X_i \in B(Y,r) | X_1 \in B(Y,r)) \\ &= (m-1)\frac{V(n,r)-1}{2^n-1} \\ &\leq m\frac{V(n,r)}{2^n} \\ &\leq 2^{nR}2^{nH(p+\epsilon)}2^{-n} = 2^{n(R-(1-H(p+\epsilon)))} \to 0 \end{aligned}$$

since $R < 1 - H(p+\epsilon)$

□

Proposition 9.12. *We can replace e by $\widehat{e}$ in the previous proposition.*

Proposition 9.13. *Pick R' such that $R < R' < 1 - H(p)$. Then the previous proposition constructs $C_1', C_2', \dots$, with C_n' of length n, size $\lfloor 2^{nR'} \rfloor$, and $e(C_n') \to 0$ as $n \to \infty$.*

Order then codewords of C_n' by $\mathbb{P}(\text{error}|c \text{ sent})$, and delete the worse half them to give C_n. Then we have $|C_n| = \lfloor \frac{|C_n'|-1}{2} \rfloor, \widehat{e}(C_n) \leq 2e(C_n')$. Then $\rho(C_n) \to R$ and $\widehat{e}(C_n) \to 0$ as $n \to \infty$.

Note that:

1. **9.12** says we can transmit reliably at any rate $R < 1 - H(p)$ so the capacity is at least $1 - H(p)$. But, by **9.10**, the capacity at most $1 - H(p)$, and hence the BSC with error probability has capacity $1 - H(p)$.
2. The proof shows that good codes exist, but it does not tell us how to find them.

10 Interlude: An Application to Gambling

Let $0 < p < 1$, $n > 0$, and $0 \leq w < 1$. A coin is tossed n times in succession. $P(H) = p$ and, if I pay k ahead of a particular throw, then I get back kn if the throw is head, but nothing if the throw is a tail.

What is my strategy?

- If $pn < 1$, then don't bet - your expected winnings is < 0.

- If $pn > 1$, then we want to bet, but how much? A larger bet means more winnings, but also more risk. What proportion w of our total wealth should we bet?

Note that w is always the same, only the size of the fortune changes. Let the fortune by X_j after the j^{th} throw. I bet wX_j, retaining $(1-w)X_j$. My fortune X_{j+1} after the $(j+1)^{\text{th}}$ throw is:

$$X_{j+1} = \begin{cases} X_j(Xn + 1 - w) & j^{\text{th}} \text{ throw is H} \\ X_j(1-w) & j^{\text{th}} \text{ throw is T} \end{cases}$$

Put $Y_{j+1} = \frac{X_{j+1}}{X_j} = \begin{cases} wn + (1-2) & H \\ 1 - w & T \end{cases}$.

Then we try to maximise the log of our fortune: let $\mathbb{E} \log Y_i = \bar{\mu}4, \mathrm{Var}(\log Y_i) = \bar{\sigma}^2$. If $a > 0$ then:

$$\mathbb{P}\left(\left| \frac{\log Y_1 + \ldots + \log Y_n}{n} - \bar{\mu} \right| \geq a \right) \leq \frac{\bar{\sigma}^2}{na^2}$$

by Chebyshev's inequality. But $\sum \log Y_i = \log X_i$, so we have:

$$\mathbb{P}\left(\left| \frac{\log X_n}{n} - \bar{\mu} \right| \geq a \right) \leq \frac{\bar{\sigma}^2}{na^2}$$

i.e., for all $\epsilon > 0$, $\delta > 0$, there is some N such that

$$\mathbb{P}\left(\left| n^{-1} \log(X_n) - \bar{\mu} \right| \geq \delta \right) \leq \epsilon \ \forall \, \mathbb{N}$$

Lemma 10.1. *Consider one single toss of a coin with $\mathbb{P}(H) = p < 1$. Suppose that a bet on heads has payout ratio of n. Suppose that we have a bankroll of 1 unit and we bet w on H, retaining $1 - w$ for $0 \leq 1 \leq 1$. If Y is the expected value of our fortune after the throw then*

$$\mathbb{E}(\log Y) = p \log(1 + (n-1)w) + (1-p)\log(1-w)$$

The value of $\mathbb{E}(\log Y)$ is maximised by taking $w = 0$ if $np \leq 1$ and setting $w = \frac{np-1}{n-1}$ if $np > 1$.

A better who maximises the log of his or her fortune is called a ***Kelly better***, and is following ***Kelly's rule***, since Kelly in 1956 showed how to do this using information theory.

Shannon's Second Coding theorem says that information can be transmitted over a channel at a rate close to the capacity with negligible error rate, provided that we're allowed arbitrarily long messages. Kelly's rule says that a gambler can, with high probability, increase her fortune at a certain optimum rate, provided that she can continue to bet for long enough.

11 Algebraic Coding Theory

11.1 Linear Codes

A code $C \in \mathbb{F}_2^n$ is ***linear*** if $0 \in C$ and $x, y \in C \implies x + y \in C$. I.e., C is an $\mathbb{F}_2$-vector subspace of $\mathbb{F}_2^n$. The ***rank*** of C is its dimension as an $\mathbb{F}_2$-vector subspace. A linear code of length n, rank k is denoted an ***(n,k)-code***, using round brackets instead of square brackets. If its minimum distance is d, it is an (n, k, d)-code.

Let $v_1, \dots, v_k$ be a basis for C. Then $C = \left\{\sum_{i=1}^k \lambda_i v_i : \lambda_i \in \mathbb{F}_2\right\}$, so $|C| = 2^k$, and (n, k)-codes are $[n, 2^k]$-codes. The information rate is then $\frac{k}{n}$.

The ***weight*** of $x \in \mathbb{F}_2^n$, denoted $w(x) = d(x, 0)$.

Lemma 11.1. *The minimum distance of a linear code is the minimum weight of a non-zero codeword.*

Proof. Suppose $d(x, y) = d$. Then $d(x + y, y + y) = d(x + y, 0) = w(x + y) = d$, and $x + y \in C$ as C is linear. □

For $x, y \in \mathbb{F}_2^n$, let $x \cdot y = \sum_{i=1}^n x_i y_i \in \mathbb{F}_2$. Note that there are $x \neq 0$ with $x \cdot x = 0$. We then define the ***parity check code*** defined by $P \subseteq \mathbb{F}_2^n$ to be $C = \{x \in \mathbb{F}_2^n : p \cdot x = 0 \ \forall p \in P\}$.

We've already seen some examples of this:

- $P = \{111\dots1\}$ gives the simple parity check code.
- $P = \{1010101, 0110011, 0001111\}$ gives Hamming's original code.

Lemma 11.2. *Every parity check code is linear.*

Proof. $0 \in C$ as $p \cdot 0 = 0 \ \forall p \in P$. If $x, y \in C$, then $p \cdot (x + y) = p \cdot x + p \cdot y = 0 + 0 = 0$, so $x + y \in P$, □

If $C \subset \mathbb{F}_2^n$ is a linear code, then the ***dual code*** of C, $C^\perp = \{x \in \mathbb{F}_2^n : x \cdot y = 0 \ \forall y \in C\}$.

This is a parity check code so is linear. Note that $C \cap C^\perp$ can be larger than $\{0\}$.

Lemma 11.3. $\operatorname{rank}(C) + \operatorname{rank}(C^\perp) = n$

Proof. Linear algebra using $C^\perp$ is the annihilator of C, or see later. □

Lemma 11.4. *If C is linear, then $(C^\perp)^\perp = C$, and in particular C is a parity check code.*

Proof. If $x \in C$ then $x \cdot y = 0$ for all $y \in C^\perp$. Hence $x \in (C^\perp)^\perp$, and so $C \subseteq (C^\perp)^\perp$.

Then $\operatorname{rank}(C) = n - \operatorname{rank}(C^\perp) = n - (n - \operatorname{rank}((C^\perp)^\perp)) = \operatorname{rank}(C^\perp)^\perp)$, and so $C = (C^\perp)^\perp$. □

Let C be a (n, k)-code. Then we have a couple of useful matrices:

1. A ***generator matrix*** G for C is a $k \times n$ matrix with rows a basis of C.
2. A ***parity check matrix*** H for C is a generator matrix for $C^\perp$.

The codewords in C can be viewed as linear combinations of row of G, or as linear dependence relations between the columns of H.

11.2 Syndrome Decoding

If C is an (n, k)linear code, then the syndrome of $x \in \mathbb{F}_2^n$ is Hx. If we receive $x = c + z$ where c is a codeword and z is an error pattern, then $Hx = Hc + Hz = Hz$.

If C is e-error correcting, we pre-compute Hz for all z with $w(z) \leq e$. On receiving $x \in \mathbb{F}_2^n$ we look for Hx in our list. Then if $Hx = Hz$, $H(x - z) = 0$, so $c = x - z \in C$ with $d(x, c) \leq e$.

Codes $C_1, C_2 \in \mathbb{F}_2^n$ are ***equivalent*** if, reordering each codeword of C_1 using the same permutation each time, gives the codewords of C_2. Usually we only consider codes up to equivalence.

Proposition 11.5. *Every (n, k)-linear code is equivalent to one with generator matrix of the form $G = (I_k|B)$ for some $k \times (n - k)$ matrix B.*

Proof. Using elementary row operations we can transform G into row echelon form

$$G_{ij} = \begin{cases} 0 & j < \ell(i) \\ 1 & j = \ell(i) \end{cases}$$

for some $\ell(1) < \ell(2) < \ldots < \ell(k)$. Permuting the columns replaces the code by an equivalent code, and so WLOG we can assume $\ell(i) = i$ for all i. Then we can cancel all the non-zero off diagonal entries on the left hand square of G, to get it in the form $(I_k|B)$. □

If we have a message $y \in \mathbb{F}_2^k$ viewed as a row matrix, we encode it as yG. If G is in the form of **11.5**, then $yG = (y|yB)$ where y is the message and yB are check digits. Any code whose codewords can be split up in this manner is called ***systematic***. This form of the generator matrix will let us prove **11.3** without working with dual spaces:

Proof of 11.3. Without any loss of generality, we may take C to have generator matrix $G = (I_k|B)$. G has k linearly independent columns, so the linear map $c : \mathbb{F}_2^n \to \mathbb{F}_2^k; x \mapsto Gx$ is surjective with kernel $C^\perp$. By the rank nullity, $\dim \mathbb{F}_2^n = \operatorname{rank} C + \operatorname{rank} C^\perp$. □

Lemma 11.6. *An (n, k)-linear code with generator matrix $G = (I_k|B)$ has parity check matrix $H = (B^t|I_{n-k})$.*

Proof. $GH^t = (I_k|B)\left(\frac{B}{I_{n-k}}\right) = B + B = 0$, so the rows of H generate a subcode of $C^\perp$. But the rank of $H = n - k$since H contains I_{n-k} as a submatrix and $n - k = \operatorname{rank} C^\perp$, so $C^\perp$ has generator matrix H, and H is a parity check matrix for C. □

12 Examples of Codes

12.1 Hamming Codes

For $d \geq 1$ let $n = 2^d - 1$. Let H be the $d \times n$ matrix whose columns are the non zero elements of $\mathbb{F}_2^d$. The ***Hamming (n,n-d)-code*** is the linear code with parity check matrix H. For example with $d = 3$:

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

Lemma 12.1. *The minimum distance of the $(n, n-d)$-Hamming code C is $d(C) = 3$. It is a perfect 1-error correcting code.*

Proof. The codewords of C are dependence relations between the columns of H. Any two columns of H are linearly independent, so there are no codewords of weight at most 2. Hence $d(C) \geq 3$. Noting that $11100\ldots0 \in C$, we have $d(C) = 3$. It is 1-error correcting by **5.3**. Moreover, $\frac{2^n}{V(n,1)} = \frac{2^n}{n+1} = 2^{n-d} = |C|$, so C is perfect. $\square$

12.2 Reed-Muller Codes

Take some finite set X of size n say. Then there is a correspondence between $\mathcal{P}(X)$ and $\mathbb{F}_2^n$ given by indicator functions. Moreover, we have that the following set-theoretic operations in $\mathcal{P}(X)$ correspond to vector operations on $\mathbb{F}_2^n$:

$$A \Delta B = (A \setminus B) \cup (B \setminus A) \longleftrightarrow x + y = (x_1 + y_1, \ldots, x_n + y_n)$$
$$A \cap B \longleftrightarrow x \wedge y = (x_1 y_1, \ldots, x_n y_n)$$

Take $X = F_2^d$, so $n = 2^d$. Let $v_0 = \mathbb{1}_X = (1, 1, \ldots, 1)$. Let $v_i = \mathbb{1}_{H_i}$ for $1 \leq i \leq d$, where $H_i = \{p \in X : p_i = 0\}$, the coordinate hyperplane.

Then the ***Reed-Muller code RM(d,r)*** of order r where $0 \leq r \leq d$ of length 2^d si the linear code spanned by v_0 and all wedge products of r or fewer of the v_i. By convention, the empty wedge is v_0. For example, with $d = 3, X = \mathbb{F}_2^3$ in binary number order:

X	000	001	010	011	100	101	110	111
v_0	1	1	1	1	1	1	1	1
v_1	1	1	1	1	0	0	0	0
v_2	1	1	0	0	1	1	0	0
v_3	1	0	1	0	1	0	1	0
$v_1 \wedge v_2$	1	1	0	0	0	0	0	0
$v_2 \wedge v_3$	1	0	0	0	1	0	0	0
$v_1 \wedge v_3$	1	0	1	0	0	0	0	0
$v_1 \wedge v_2 \wedge v_3$	1	0	0	0	0	0	0	0

Note that:

- RM(3,0): spanned by v_0, is a repetition code of length 8.
- RM(3,1): spanned by v_0, v_1, v_2, v_3, is a parity check extension of Hamming's original code.
- RM(3,2): an $(8, 7)$-code, which is actually a simple parity check code.

- RM(3,3): the trivial code for $\mathbb{F}_2^8$.

Theorem 12.2.

1. *The vectors* $v_{i_0} \wedge \ldots \wedge v_{i_j}$ *form a basis for* $\mathbb{F}_2^n$.
2. *The rank of* $RM(d,r) = \sum_{s=0}^{r} \binom{d}{s}$.

Proof.

1. We've listed $\sum_{s=0}^{d} \binom{d}{s} = (1+1)^d = 2^d$ vectors, so it's enough to check spanning, i.e. that $RM(d,d) = \mathbb{F}_2^n$. Let $p \in X$. Put $y_i = \begin{cases} v_i & p_i = 0 \\ v_0 + v_i & p_i = 1 \end{cases}$. Then $\mathbb{1}_{\{p\}} = y_1 \wedge \ldots \wedge y_d$, and we can expand this using distributivity to get $\mathbb{1}_{\{p\}} \in RM(d,d)$. But $\{\mathbb{1}_{\{p\}} : p \in X\}$ spans $\mathbb{F}_2^n$, and so the given vectors form a basis.
2. $RM(d,r)$ is spanned by $v_{i_1} \wedge \ldots \wedge v_{i_s}$ for $1 \leq i < \ldots < i_s \leq d$ with $0 \leq s \leq r$. These vectors are linearly independent by part *1.*, and so form a basis. Hence rank $RM(d,r) = \sum_{s=0}^{r} \binom{d}{s}$.

□

Let C_1, C_2 be linear codes of length n with $C_2 \subseteq C_1$. The ***bar product*** $C_1|C_2 = \{(x|x+y) : x \in C_1, y \in C_2\}$. It is a linear code of length $2n$.

Lemma 12.3.

1. $\operatorname{rank}(C_1|C_2) = \operatorname{rank} C_1 + \operatorname{rank} C_2$
2. $d(C_1|C_2) = \min\{2d(C_1), d(C_2)\}$

Proof.

1. C_1 has a basis $x_1, \ldots, x_k$, C_2 has a basis $y_1, \ldots, y_\ell$. Then $C_1|C_2$ has a basis given by $\{(x_i|x_i : 1 \leq i \leq k\} \cup \{(0|y_i) : 1 \leq i \leq \ell\}$.

   So $\operatorname{rank}(C_1|C_2) = k + \ell = \operatorname{rank} C_1 + \operatorname{rank} C_2$.
2. Take $x \in C_1, y \in C_2$ not both zero so that $(x|x+y) \in C_1|C_2$ is non-zero. Then if $y \neq 0, w(x|x+y) \geq w(y) \geq d(C_2)$. Meanwhile if $y = 0$, then $w(x|x+y) = 2w(x) \geq 2d(C_1)$.

   Hence $d(C_1|C_2) \geq \min\{2d(C_1), d(C_2)\}$. For equality, take x of weight $d(C_1)$, y of weight $d(C_2)$. Then $x|0$ or $0|y$ attains the bound.

□

Theorem 12.4.

1. $RM(d,r) = RM(d-1, r-1)|RM(d-1, r-1)$
2. $RM(d,r)$ *has minimum distance* 2^{d-r}

Proof.

1. $RM(d-1,r-1) \subseteq RM(d-1,r)$, so the bar product is defined. Order the elements of $X = \mathbb{F}_2^d$ such that $v_d = (0\dots0|1\dots1)$ where each part has length 2^{d-1}, and let $v_i = (v_i'|v_i')$

   Take $z \in RM(d,r)$, then z is a sum of wedge products of $v_1,\dots,v_d$. So $z = x + (y \wedge v_d)$ for x, y sums of wedge products of $v_1,\dots,v_{d-1}$. Then $x = (x'|x')$ for some $x' \in RM(d-1,r)$, and $y = (y'|y')$ for some $y' \in RM(d-r,r-1)$.

   Then $z = (x'|x') + (y'|y') \wedge (0\dots0|1\dots1) = (x'|x'+y') \in RM(d-1,r)|RM(d-1,r-1)$. The converse is similar, or argue using cardinalities.

2. If $r = 0$ then $RM(d,0)$ is a repetition code of length $n = 2^d$, which has minimum distance 2^{d-0}. If $r = d$, then $RM(d,d)$ is the trivial code which has minimum distance $1 = 2^{d-d}$. We prove the case $0 < r < d$ by induction on d. The minimum distance of $RM(d-1,r) = 2^{d-1-r}$ and of $RM(d-1,r-1)$ is 2^{d-r}. So by the previous lemma, the minimum distance of $RM(d,r) = \min\{2 \cdot 2^{d-r-1}, 2^{d-r}\} = 2^{d-r}$.

□

13 Cyclic Codes

A linear code $C \subseteq \mathbb{F}_2^n$ is ***cyclic*** if $(a_0, a_1, \dots, a_{n-1}) \in C \implies (a_{n-1}, a_0, a_1, \dots, a_{n-2}) \in C$, i.e. rotations of all of our codewords are also codewords. We can identify $\mathbb{F}_2^n$ with $\mathbb{F}_2[x]/(x^n-1)$, via $(a_0, a_1, \dots, a_{n-1}) \leftrightarrow a_0 + a_1x + \dots + a_{n-1}x^{n-1}$.

Lemma 13.1. *$C \subseteq \mathbb{F}_2[x]/(x^n-1)$ is cyclic if and only if $C \trianglelefteq \mathbb{F}_2[x]/(x^n-1)$.*

Proof. If $g(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1} \mod x^n - 1$ then $xg(x) = a_{n-1} + a_0x + \dots x_{n-2}x^{n-2}$, and so if $C \subseteq \mathbb{F}_2[x]/(x^n-1)$ is closed under multiplication by x then C has this cyclic property.

But we also know C is linear, and so this gives us closure under addition of codewords, so by repeated application of additions and multiplications, we have $f(x)g(x) \in C$ for any polynomial $f(x)$, and in particular C is an ideal of $\mathbb{F}_2[x]/(x^n-1)$. □

Theorem 13.2. *Let $C \subset \mathbb{F}_2[x]/(x^n-1)$ be a cyclic code. Then there is a unique $g(x) \in \mathbb{F}_2[x]$ such that:*

1. *$C = \{f(x)g(x) \mod x^n - 1 : f(x) \in \mathbb{F}_2[x]\}$*
2. *$g(x)|x^n - 1$*

In particular, $p(x) \in \mathbb{F}_2[x]$ represents a codeword if and only $g(x)|p(x)$. We say $g(x)$ is **the generator polynomial** *of C.*

Proof. Let $g(x) \in \mathbb{F}_2[x]$ be of least degree representing a non-zero codeword. Then $\deg g < n$. Since C is cyclic, we have $\supseteq$ in *1*.

Let $p(x) \in \mathbb{F}_2[x]$ represent a codeword. Then $p(x) = q(x)g(x) - r(x)$ where $\deg r < \deg g$. So $r(x) = p(x) - q(x)g(x) \in C$ as C is an idea and so by minimality of $\deg g$, $r(x) = 0$. So $g(x)|p(x)$.

For uniqueness, suppose $g_1(x), g_2(x)$ both satisfy the properties. Then $g_1(x)|g_2(x)$ and $g_2(x)|g_1(x)$, so $g_1(x) = g_2(x)$. □

We can actually think of C as a vector space over $\mathbb{F}_2$:

Lemma 13.3. *Let C be a cyclic code of length n generated by $g(x) = a_0 + a_1 x + \ldots + a_k x^k$, $a_k \neq 0$. Then C has $g(x), xg(x), \ldots, x^{n-k-1}g(x)$, and in particular* $\operatorname{rank} C = n - k$.

Proof. Suppose $f(x)g(x) = 0 \mod x^n - 1$ for some $f(x) \in \mathbb{F}_2[x]$ with $\deg f < n - k$. Then $\deg fg < n$, so $f(x)g(x) = 0 \implies f(x) = 0$, and so these terms are linearly independent.

For spanning, let $p(x)$ represent a codeword. Since $g(x)$ is the generator polynomial, $g(x)|p(x)$, i.e. $p(x) = f(x)g(x)$ for some $f(x)$. But $\deg f = \deg p - \deg g < n - k$, so $p(x)$ is in the span of $g(x), xg(x), \ldots, x^{n-k-1}g(x)$. □

Corollary 13.4. *The $(n-k) \times n$ generator matrix is:*

$$G = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_k & 0 & \cdots & 0 \\ 0 & a_0 & a_1 & \cdots & a_{k-1} & a_k & \cdots & 0 \\ \vdots & & \ddots & \ddots & & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_0 & a_1 & \cdots & a_k \end{pmatrix}$$

Lemma 13.5. *The* **parity check polynomial** *$h(x) \in \mathbb{F}_2[x]$ is defined by $x^n - 1 = g(x)h(x)$. If $h(x) = b_0 + b_1 x + \ldots + b_{n-k}x^{n-k}$, then the parity check matrix is the $k \times n$ matrix:*

$$H = \begin{pmatrix} b_{n-k} & b_{n-k-1} & \cdots & b_1 & b_0 & 0 & \cdots & 0 \\ 0 & b_{n-k} & \cdots & b_2 & b_1 & b_0 & \cdots & 0 \\ \vdots & & \ddots & & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & b_{n-k} & \cdots & \cdots & b_1 & b_0 \end{pmatrix}$$

Proof. Indeed the dot product of the i^{th} row of G and the j^{th} column of H is the coefficient of $x^{n-k-i+j}$ in $g(x)h(x)$. The coefficients of $g(x)h(x) = x^n - 1$ are 0, and hence the rows of G and H are orthogonal. Also, $\operatorname{rank} H = k = rkC^{\perp}$, so H is a parity check matrix. □

Note that the check polynomial is the reverse of the generator polynomial for the dual code.

Lemma 13.6. *If n is odd then $x^n - 1 = f_1(x) \ldots f_t(x)$ with $f_i(x)$ distinct irreducibles in $\mathbb{F}_2[x]$. In particular, there are 2^t cyclic codes of length n.*

Proof.* If $x^n - 1$ has a repeated factor then there is a field extension $K/\mathbb{F}_2$ such that $x^n - 1 = (x - \lambda)^2 g(x)$ for some $\lambda \in K, g(x) = \mathbb{F}_2[x]$. Taking formal derivatives, $nx^{n-1} = 2(x-\lambda)g(x) + (x-\lambda)^2 g'(x)$.

So $n\lambda^{n-1} = 0 \implies \lambda = 0$, and so $0 = \lambda^n = 1$ ↯. □

14 BCH Codes

Let n be an odd integer, and $r \geq 1$ such that $2^r \equiv 1 \mod n$, which exists since $(2, n) = 1$. Letting $K = \mathbb{F}_{2^r}$, let $\mu_n(K) = \{x \in K : x^n = 1\} \leq K^*$. Since $n|(2^r - 1) = |K^*|$, $\mu_n(K)$ is a cyclic group of order n. So $\mu_n(K) = \{1, \alpha, \ldots, \alpha^{n-1}\}$ for some $\alpha \in K$ called a primitive n^{th} root of unity.

The cyclic code of length n with ***defining set*** $A \subseteq \mu_n(K)$ is:

$$C = \{f(x) \mod x^n - 1 : f(x) \in \mathbb{F}_2[x], f(a) = 0 \forall a \in A\}$$

The ***generator polynomial*** is the nonzero polynomial $g(x)$ of least degree with $g(x) = 0$ for all $a \in A$. Equivalently, $g(x)$ is the lowest common multiple of the minimal polynomials of the elements $a \in A$.

The cyclic code with defining set $\{\alpha, \alpha^2, \ldots, \alpha^{\delta-1}\}$ is called a ***BCH Code*** with ***design distance*** δ.

Lemma 14.1 (Vandermonde Determinant)**.**

$$\left|\begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & & \vdots \\ {x_1}^n - 1 & x_2^{n-1} & \cdots x_n^{n-1} & \end{pmatrix}\right| = \prod_{1 \le j < i \le n} (x_i - x_j)$$

Proof. Exercise. □

Theorem 14.2. *A BCH code C with design distance δ has $d(C) \ge \delta$.*

Proof. Consider:

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^{\delta-1} & \alpha^{2(\delta-1)} & \cdots & \alpha^{(\delta-1)(n-1)} \end{pmatrix}$$

By the Vandermonde determinant the columns of H are linearly independent. But any codeword of C is a dependence relation between columns of H. Hence every non-zero codeword has weight $\ge \delta$. Hence $d(C) \ge \delta$. □

Note that H is not a parity check matrix as defined, because the entries are in K and not in $\mathbb{F}_2$.

14.1 Decoding BCH Codes

Let C be a cyclic code with defining set $\{\alpha, \alpha^2, \ldots, \alpha^{\delta-1}\}$, where $\alpha \in K$ is a primitive n^{th} root of unity. Then we should be able correct $\lfloor \frac{\delta-1}{2} \rfloor = t$ errors. We send $c \in C$ and receive $r = c + e$ where e is an error pattern with $\le t$ non-zero entries. Now we have a correspondence $\mathbb{F}_2^n \leftrightarrow \mathbb{F}_2[x]/(x^n - 1)$, so $r, c, e \leftrightarrow r(x), c(x), e(x)$.

Then we define the ***error locator polynomial*** to be $\sigma(x) = \prod_{i \in \mathcal{E}}(1 - \alpha^i X) \in K[x]$, where $\mathcal{E} = \{0 \le i \le n-1 : e_i = 1\}$.

Theorem 14.3. *Assume* $\deg \sigma = |\mathcal{E}| \le t$*. Then $\sigma(x)$ is the unique polynomial in $K[x]$ of least degree such that:*

- $\sigma(0) = 1$
- $\sigma(x) \sum_{j=1}^{2t} r(\alpha^i) x^j = \omega(x)$

for some $\omega(x) \in K[x]$ with $\deg \omega \le t$.

Proof. Define $\omega(x) = -x\sigma'(x)$. Then:

$$\omega(x) = \sum_{i\in\mathcal{E}} \alpha^i x \prod_{i\neq j\in\mathcal{E}} (1-\alpha^j x)$$

Working in $K[[x]]$, the ring of formal power series, we get:

$$\begin{aligned}\frac{\omega(x)}{\sigma(x)} &= \sum_{i\in\mathcal{E}} \frac{\alpha^i x}{1-\alpha^i x}\\ &= \sum_{i\in\mathcal{E}} \sum_{j=1}^{\infty} (\alpha^i x)^j\\ &= \sum_{j=1}^{\infty} x^j \sum_{i\in\mathcal{E}} (\alpha^j)^i\\ &= \sum_{j=1}^{\infty} e(\alpha^j)x^j\end{aligned}$$

So $\omega(x) = \sigma(x)\sum_{j\geq 1} e(\alpha^j)x^j$. By the definition of C, $c(\alpha^i) = 0$, and so $r(\alpha^i) = e(\alpha^i)$. Hence:

$$\sigma(x)\sum_{j=1}^{2t} r(\alpha^j)x^j = \omega(x) \mod x^{2t+1}$$

This verifies *1.* and *2.* with $\omega(x) = -x\sigma'(x)$, so $\deg\omega = \deg\sigma = |\mathcal{E}| \leq t$.

For uniqueness, suppose we have $\widetilde{\sigma}, \widetilde{\omega}$ also satisfying these properties with $\deg\widetilde{\sigma} \leq \deg\sigma$. Then if $i \in \mathcal{E}$, $\omega(\alpha^{-i}) = \prod_{i\neq j\in\mathcal{E}}(1-\alpha^{j-i}) \neq 0$ and $\sigma(x), \omega(x)$ are coprime.

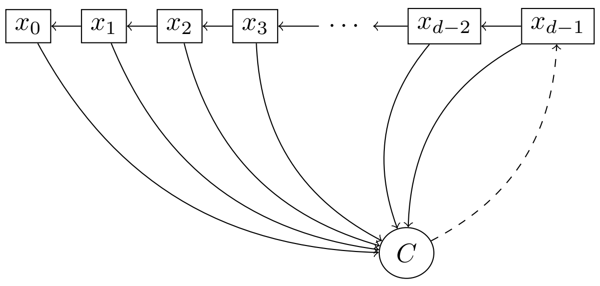
Then $\sigma(x)\widetilde{\omega}(x) \equiv \widetilde{\sigma}(x)\omega(x) \mod x^{2t-1}$, so $\sigma(x)\widetilde{\omega}(x) = \widetilde{\sigma}(x)\omega(x)$. But $\sigma(x), \omega(x)$ are coprime, so $\sigma(x)|\widetilde{\sigma}(x)$. By assumption, $\deg\widetilde{\sigma}(x) \leq \deg\sigma$, so $\widetilde{\sigma} = \lambda\sigma$ for some $\lambda \in K$. But $\lambda\sigma(0) = \lambda = 1$, so $\sigma = \widetilde{\sigma}$. □

So to decode a BCH code we have the following procedure:

- We receive $r(x)$.
- Compute $\sum_{j=1}^{2t} r(\alpha^j)x^j$.
- Set $\sigma(x) = 1 + \sigma_1 x + \ldots + \sigma_t x^t$ and compare coefficients of x^i for $t+1 \leq i\ 2T$ to obtain linear equations for $\sigma_1, \ldots, \sigma_t$.
- Solve these via Gaussian elimination taking solutions of least degree.
- Compute $\mathcal{E} = \{i : \sigma(\alpha^{-i}) = 0\}$, and check that $|\mathcal{E}| = \deg\sigma$.
- Set $e(x) = \sum_{i\in\mathcal{E}} x^i$, $c(x) = r(x) + e(X)$, and check $c(x)$ is a codeword.

15 Shift Registers

A ***general feedback shift register*** is a function $f : \mathbb{F}_2^d \to \mathbb{F}_2^d$ of the form $f(x_0, \ldots, x_{d-1}) = (x_1, \ldots, x_{d-1}, C(x_0, \ldots, x_{d-1}))$ for some function $C : \mathbb{F}_2^d \to \mathbb{F}_2^d$. We say the register has ***length*** d.



A register is ***linear*** (LFSR) if C is a linear map, say $(x_0, \ldots, x_{d-1}) \mapsto \sum_{i=0}^{d-1} a_i x_i$. The ***initial fill*** $(y_0, \ldots, y_{d-1})$ produces an output sequence (or stream) $(y_n)_{n\geq 0}$, where $y_{n+d} = C(y_n, y_{n+1}), \ldots, y_{n+d-1})$.

So in the linear case we have a sequence determined by a recurrence relation with ***auxiliary polynomial*** $P(x) = x^d + a_{d-1}x^{d-1} + \ldots + a_1 x + a_0$.

The ***feedback polynomial*** is $\widetilde{P}(x) = a_0 x^d + a_1 x^{d-1} + \ldots + a_{d-1}x + 1 = x^{\deg P} P(x^{-1})$.

Lemma 15.1. *The sequence $(y_n)_{n\geq 0}$ in $\mathbb{F}_2$ is the output from an LFSR with auxiliary polynomial $P(x)$ if and only if $G(x)\widetilde{P}(x) = A(x)$ where $G(x)$ is the polynomial $\sum_{i\geq 0} y_i x^i$, and $A(x)$ is some polynomial of degree $< \deg P$.*

Proof. Let $P(x), \widetilde{P}(x)$ be as above. The condition holds if and only if the coefficient of x^n in $G(x)\widetilde{P}(x)$ is 0 for $n \geq d$, which happens if and only if $\sum_{i=0}^{d} a_i y_{n-d+i} = 0$ for $n \geq d$.

This happens iff $y_{n+d} = \sum_{i=0}^{d-1} a_i y_{n-i}$ for $n \geq 0$, i.e. if $(y_n)_{n\geq 0}$ comes from a LFSR with feedback polynomial $\widetilde{P}(x)$. □

Compare this proof with the decoding procedure for BCH codes

15.1 Berlekamp-Massey Algorithm

This is an algorithm to find the shortest LFSR for a given binary output stream. Let (x_n) be the output of an LSFR. We want to find d and $a_0, a_1, \ldots, a_{d-1} \in \mathbb{F}_2$ such that $x_{n+d} = \sum_{i=0}^{d-1} a_i x_{n+i}$ for all $n \geq 0$.

Translating this into a linear algebra problem, we have:

$$\underbrace{\begin{pmatrix} x_0 & x_1 & \cdots x_d & \\ x_1 & x_2 & \cdots x_{d+1} & \\ \vdots & \vdots & & \vdots \\ x_d & x_{d+1} & \cdots & x_{2d} \end{pmatrix}}_{A_d} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{pmatrix} = 0 \qquad (*)$$

Then if we know the register has length $\geq r$, start with $d = r$. For each i, compute $\det A_i$.

- If $\det A_i \neq 0$ then $d > i$, so replace i by $i + 1$ and repeat.

- If $\det A_i = 0$, we can solve $(*)$ using Gaussian elimination, and test the solution over as many terms of the sequence as we like. If it fails, then $d > i$, and replace i by $i+1$ and continue.

We'll return to this in §17.

16 Cryptography

16.1 Introduction and Examples

We want to modify a message such that it becomes unintelligible to an eavesdropper. Certain secret information will be shared by A, B, called the ***key***, which lies in a space $\mathcal{K}$. The unencrypted message will be called ***plaintext***, lying in $\mathcal{M}$. The encrypted message will be called ***ciphertext***, lying in $\mathcal{C}$. A ***cryptosystem*** consists of sets $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, together with functions:

$$e : \mathcal{M} \times \mathcal{K} \to \mathcal{C} \qquad \text{(encrypt)}$$
$$d : \mathcal{C} \times \mathcal{K} \to \mathcal{M} \qquad \text{(decrypt)}$$

with the property that $d(e(m,k),k) = m$ for all $m \in \mathcal{M}, k \in \mathcal{K}$.

Example: s: $\mathcal{M} = \mathcal{C} = \Sigma = \{A, B, \dots, Z\}$

- Simple substitution: $\mathcal{K} = \{\text{permutations of } \Sigma\}$. Then each letter is encrypted as the image under the chosen permutation.

- Vigenère cipher: $\mathcal{K} = \Sigma^d$ for some $d \in \mathbb{N}$. We identify Σ and $\mathbb{Z}/26\mathbb{Z}$. and write the key repeatedly below the message and add mod 26:

$$\begin{aligned} &\texttt{ATTACKATDAWN} \\ +\ &\underline{\texttt{LEMONLEMONLE}} \\ =\ &\texttt{LXFOPVEFRNHR} \end{aligned}$$

If $d = 1$ this is called a ***Caesar cipher***.

16.2 Breaking the Cryptosystem

Some attacker E might know:

- The functions e and d (i.e. the mechanics of the cryptography)
- The probability distributions on $\mathcal{M}, \mathcal{K}$.

She does not know the key however. Her goal is to recover the plaintext from the ciphertext. There are three possible levels of attack:

Level 1: Ciphertext only - she just knows some piece of ciphertext.

Level 2: Known plaintext - she possesses a considerable length of plaintext and matching ciphertext.

Level 3: Chosen plaintext - she may acquire the ciphertext for any plaintext she wants to.

For example, the substitution cipher and Vigenère cipher fail at level 2 for sufficiently random messages. They can in fact fail at level 1, for instance if we know the source is a passage of English text. For modern work, level 3 is desirable.

We model the keys and messages as independent random variables K, M taking values in $\mathcal{K}, \mathcal{M}$. Put $C = e(M, K)$. We say a cryptosystem has ***perfect secrecy*** if M, C are independent, or equivalently if $I(M; C) = 0$.

Lemma 16.1. *If a code has perfect secrecy then* $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof. Pick $m_0 \in \mathcal{M}, k_0 \in \mathcal{K}$, both with strictly positive probability. Then $c_0 = e(m_0, k_0)$ also has strictly positive probability. Then for all $m \in \mathcal{M}$:

$$\mathbb{P}(C = c_0) = \mathbb{P}(C = c_0 | M = m)$$

by perfect secrecy. Hence for any $m \in \mathcal{M}$ there is a $k \in \mathcal{K}$ with $e(m, k) = c_0$. If two messages m_1, m_2 encrypt to c_0 under the same key k, then $e(m_1, k) = e(m_2, k) \implies m_1 = m_2$. So $\mathcal{M} \to \mathcal{K}$ is injective. □

From this we can conclude that perfect secrecy is an unrealistic goal in practice, as it hugely limits the number of messages we can send.

We define the:

- ***Message equivocation*** to be $H(M|C)$
- ***Key equivocation*** to be $H(K|C)$.

Proposition 16.2. $H(M|C) \leq H(K|C)$

Proof. Since $M = d(C, K), H(M|C, K) = 0$. So $H(C, K) = H(M, C, K)$. Hence we have:

$$\begin{aligned} H(K|C) &= H(M, C, K) - H(C) \\ &= H(M, C, K) - H(M, C) + H(M, C) - H(C) \\ &= \underbrace{H(K|M, C)}_{\geq 0} + H(M|C) \\ &\geq H(M|C) \end{aligned}$$

□

Let $\mathcal{M} = \mathcal{C} = \mathscr{A}$, our alphabet. We send n message $M^{(n)} = (M_1, \dots, M_n)$ encrypted as $C^{(n)} = (C_1, \dots, C_n)$, using the same key. We define the ***unicity distance*** to be the least n such that $H(K|C^{(n)}) = 0$, i.e. the smallest number of encrypted messages required to uniquely determine the key (note that uniquely determining the key does not mean that it is easy to find). Now,

$$\begin{aligned} H(K|C^{(n)}) &= H(K, C^{(n)}) - H(C^{(n)}) \\ &= H(K, M^{(n)}) - H(C^{(n)}) \\ &= H(K) + H(M^{(n)}) - H(C^{(n)}) \end{aligned}$$

We assume all keys are equally likely so that $H(K) = \log |\mathcal{K}|$; that $H(M^{(n)}) \sim nH$ for some constant H, for sufficiently large n[2]; and that all sequences of ciphertext are equally likely, so that $H(C^{(n)}) = n \log |\mathscr{A}|$. Good cryptosystems should satisfy all of these.

Then $H(K|C^{(n)}) = \log |\mathcal{K}| + nH - n \log |\mathscr{A}| \geq 0$, and so rearranging we have $n \leq U = \frac{\log |\mathcal{K}|}{R \log |\mathscr{A}|}$, where $R = 1 - \frac{H}{\log |\mathscr{A}}$, the ***redundancy***.

Now $0 \leq H \leq \log |\mathscr{A}|$. To make the unicity distance large we can make $|\mathcal{K}|$ large or use a message source with little redundancy. Many cryptosystems are believed to be secure beyond predicted unicity distance.

[2]This is true for many sources including Bernoulli sources. See §8 for details.

Example: Take a simple substitution cipher with 26! keys. $\log 26 = 4.7$, and the entropy of English is about 2 bits per symbol. So $U = \frac{\log 26!}{4.7-2} \approx 32$ symbols.

17 Stream Ciphers

A simple way to encipher is to use a ***stream cipher***: we have binary plaintext and key streams given by $p_0, p_1, p_2, \ldots$; $k_0, k_2, \ldots$ respectively. Then we generate the ciphertext stream $c_0, c_2, \ldots$ where $c_i = p_i + k_i$. It is a ***private key system*** - the security depends on a secret k shared between A, B.

One example of this is the ***one-time pad***: the key stream is a random sequence known only to A, B, so that K_i are i.i.d. $Ber(\frac{1}{2})$. then $C_i \sim Ber(\frac{1}{2})$ also, and so in the absence of the key stream deciphering is impossible, and the unicity distance is infinite. This also has perfect secrecy: $\mathbb{P}(M = m, C = c) = \mathbb{P}(M = m, K = c - m) = \mathbb{P}(M = m)\mathbb{P}(K = c - m) = \mathbb{P}(M = m)\frac{1}{2^n}$.

This seems like the perfect code, but we have 2 problems:

1. How do we construct random key sequences?
2. How do we share the key sequence?

(1.) is harder than might appear at first sight, but isn't a problem in practice. (2.) is the same problem we started with! We now have to communicate a key sequence instead of a message sequence. Because of this, in most applications a one-time pad is not practical. Instead we generate $k_0, k_1, \ldots$ using a feedback shift-register of some length d, with an initial fill given by the $k_0, \ldots, k_{d-1}$. Then:

Lemma 17.1. *Let $x_0, x_1, \ldots$ be the stream produced by the shift register of length d. Then there exists $N, M \leq 2^d$ with $x_{N+r} = x_r$ for $r \geq M$.*

Proof. Note that there are only 2^d different states for the shift register to be in, and so there must be times $0 \leq M \leq M + N \leq 2^d$ with the shift register in the same state at M as at $M + N$, and then inductively $x_{M+r} = x_{M+N+r}$ for $r \geq 0$. □

Note that this tells us the maximum period of a feedback shift register of length d is 2^d. Playing a similar game with a LFSR, the maximal period is $2^d - 1$. One can show using Galois theory that a LFSR attains its maximal period for a non-trivial initial fill when the roots of the feedback polynomial are primitive elements of $\mathbb{F}_2$. However, ciphers using LFSRs fail at level 2 due the Berlekamp-Massey algorithm. If we know a piece of plaintext and corresponding ciphertext, we can compute key sequences from their difference, and B-M tells us how to find the feedback polynomial and hence the key. Some advantages of this system are that it's cheap, fast, easy to use. Messages can be encrypted and decrypted on the fly, and it is error tolerant.

17.1 Notes about Linear Recurrence Relations

Over $\mathbb{C}$ the general solution is a linear combination of solutions $\alpha^n, n\alpha^n, n^2\alpha^n, \ldots, n^{\ell-1}\alpha^n$ for a root of the auxiliary polynomial $P(x)$ with multiplicity ℓ. Now, $n^2 \equiv n \mod 2$, so over $\mathbb{F}_2$ we need a couple of modifications:

- Work in a splitting field K for $P(x)$.
- Replace $n^i\alpha^n$ by $\binom{n}{i}\alpha^n$ (See Körner's book, §16).

We can generate new key streams from old ones:

Lemma 17.2. *Let $(x_n), (y_n)$ be outputs of LFSRs of lengths M, N respectively. Then:*

1. *$(x_n + y_n)$ is the output from an LFSR of length $M + N$.*
2. *$(x_n y_n)$ is the output from an LFSR of length MN*

Proof.* Assume the auxiliary polynomials $P(x), Q(x)$ each have distinct roots $\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_N$ lying in some extension field K of $\mathbb{F}_2$. Then $x_n = \sum_{i=1}^{M} \lambda_i \alpha_i^n, y_n = \sum_{j=1}^{N} \mu_j \beta_j^n$ for some $\lambda_i, \mu_j \in K$. Then $x_n + y_n$ is a solution to a difference equation with auxiliary polynomial $P(x)Q(x)$, and $x_n y_n$ to one with auxiliary polynomial $\prod\prod(x - \alpha_i \beta_j)$, lying in $\mathbb{F}_2[x]$, via symmetric function theory. $\square$

Example: Suppose we have 3 streams $(x_n), (y_n), (z_n)$ produced by LFSRs. Let:

$$k_n = \begin{cases} x_n & z_n = 0 \\ y_n & z_n = 1 \end{cases}$$

Apply **17.2** by writing $k_n = x_n + z_n(x_n + y_n)$ to see that k_n is again the output from an LFSR.

Note that adding the outputs of two LFSRs is no more economical than producing the same thing with a single LFSR. Multiplying two output streams looks promising until we realise that $x_n y_n = 0$ 75% of the time.

Non-linear registers look appealing but are hard to analyse.

18 Asymmetric Ciphers

The key is split into two parts:

- A private key for decryption.
- A public key for encryption.

Knowing the encryption and decryption algorithms it should be hard to find the private key or to decrypt messages, implying security at level 3. Also, there is no requirement to exchange keys. We base the system on mathematical problems that are believed to be "hard":

1. ***Factoring*** - Let $N = pq$ for p, q large primes. Given N, find p and q.
2. ***Discrete logarithms*** - Let p be a large prime and g a primitive root mod p. Given x, find a such that $a = g^a \mod p$.

We say an algorithm runs in ***polynomial time*** if the number of operations is less than some constant times the input size raised to some integer power.

Some polynomial time algorithms include:

- Arithmetic of integers
- Computation of GCD
- Modular exponentiation using repeated squaring

- Primality testing

Polynomial time algorithms are not known for factoring or discrete logarithms. Some elementary methods for computing them take much longer than polynomial time: dividing my successive primes up to $\sqrt{N}$ take $O(\sqrt{n})$. The baby-step giant-step problem for the discrete logarithm problem, whereby we set $m = \lfloor\sqrt{p}\rfloor$, write $a = qm + r$. Then $x = g^a \equiv g^{am+r}$, so $g^{qm} \equiv g^{-r}x$. Then list $g^{qm} \mod p$ for $q = 0, 1, \dots, m-1$, and $g^{-rx} \mod p$ for $r = 0, 1, \dots, m-1$. Then sort these two lists and look for match. This algorithm takes $O(\sqrt{p}\log p)$.

The best known method for solving the factoring problem is called the number field sieve. It has running time $\mathcal{O}(\exp[c(\log n)^{\frac{1}{3}}(\log\log N)^{\frac{2}{3}}])$, where c is a known constant. This is closer to polynomial time than $m\log N$ than to exponential time thanks to the exponents $\frac{1}{3}, \frac{2}{3}$.

Some revision of modular arithmetic: Given $\varphi(n) = |\{1 \le a \le n : (a,n) = 1\}|$, the Euler totient function, we have $\varphi(n) = |(\mathbb{Z}/n\mathbb{Z})^*|$, the size of the group of units in $\mathbb{Z}/n\mathbb{Z}$. Then, for any a coprime to n, we have $a^{\varphi(n)} \equiv 1 \mod n$, as $\operatorname{ord}_{\mathbb{Z}/n\mathbb{Z}}(a)|\varphi(n)$ by Lagrange's theorem. In the special case where $n = p$, a prime, $\varphi(p) = p-1$ and $a^{p-1} \equiv 1 \implies a^p \equiv a$, and we have Fermat's little theorem. We also have the Chinese remainder theorem, which tells us that if we have $x \equiv a \mod p, x \equiv b \mod q$ for $(p,q) = 1$, then there is a unique solution for x modulo pq.

Lemma 18.1. *Let $p = 4k-1$ be prime. Then for $d \in \mathbb{Z}$, if $x^2 \equiv d \mod p$ has a solution then a solution is $x \equiv d^k \mod p$. In particular, this congruence relation is solvable very quickly.*

Proof. Let x_0 be a solution. WLOG $x_0 \not\equiv 0 \mod p$. Then $d^{2k-1} \equiv x_0^{2(2k-1} \equiv x^{p-1} \equiv 1 mod p$, and so $(d^k)^2 \equiv d \mod p$. □

19 Examples of Public Key Cryptography

19.1 Rabin Cipher, 1979

We have a private key consisting of 2 large distinct primes congruent to 3 mod 4, and a public key $N = pq$. Then let $\mathcal{M} = \mathcal{C} = \{0, 1, \dots, N-1\}$.

We encrypt $m \in \mathcal{M}$ as $c \equiv m^2 \mod N$, so that the cipher text $c \in \mathcal{C}$ - we avoid $m < \sqrt{N}$, and ensure $(m, N) = 1$.

Then on receiving c, we can use **18.1** to solve for x_1, x_2 such that $x_1^2 \equiv c \mod p$ and $x_2^2 \equiv c \mod q$, and then find the original message using the Chinese remainder theorem.

Lemma 19.1.

1. *Let p be an odd prime and $(d,p) = 1$. Then $x^2 \equiv d \mod p$ has zero or two solutions.*
2. *Let $N = pq$ for p, q distinct odd primes, and $(d, N) = 1$. Then $x^2 \equiv d \mod N$ has zero or four solutions.*

Proof.

- $x^2 \equiv y^2 \mod p \iff p|(x^2 - y^2) \iff (p|x-y) \vee (p|x+y) \iff x \equiv \pm y \mod p$, so if there is a solution, there are two since p is odd.

- If x_0 is a solution then by CRT there are solutions x with $x \equiv \pm x_0 \mod p, x \equiv \pm x_0 \mod q$, for any of the choices of $\pm$, and these are the only solutions by part *1.*, and they are all distinct.

□

To decrypt Rabin, we find all four solutions to $x^2 \equiv c \mod N$. Messages should include enough redundancy that only one of these possibilities makes sense.

Theorem 19.2. *Breaking the Rabin cryptosystem is essentially as difficult as factoring N.*

Proof. We've already seen factoring N allows us to decrypt messages. For the converse, suppose we have an algorithm that can compute square roots mod N. Pick $x \mod N$ at random. Use the algorithm to find y such that $x^2 \equiv y^2 \mod N$. With probability $\frac{1}{2}$, $x \not\equiv y \mod N$. So then $\gcd(N, x-y)$ is a non-trivial factor of N. If this fails, pick another x. After r trials, $\mathbb{P}(\text{failure}) < \frac{1}{2^r} \to 0$. □

Recall that $N = pq$ for p, q distinct primes. We claim that if we know a multiple m of $\varphi(N) = (p-1)(q-1)$, then factoring N is easy. We write $\text{ord}_p(x)$ to mean the order of x in $(\mathbb{Z}/p\mathbb{Z})^*$. Write $m = 2^a b$ with $a \geq 1, b$ odd. Then let $\chi = \{x \in (\mathbb{Z}/N\mathbb{Z})^* : \text{ord}_p(x^b) \neq \text{ord}_q(x^b)\}$.

Theorem 19.3.

1. *If $x \in X$ then there exists $0 \leq t < a$ such that $\gcd(x^{2^t b} - 1, N)$ is a non-trivial factor of N.*
2. $|\chi| \geq \frac{1}{2}|(\mathbb{Z}/n\mathbb{Z})^*| = \frac{1}{2}\varphi(N)$

Proof. By Euler-Fermat, $x^{\varphi(N)} \equiv 1 \mod N \implies x^m \equiv 1 \mod N$. But $m = 2^a b$, so setting $y = x^b \mod N$ we get $y^{2^a} \equiv 1 \mod N$, and so $\text{ord}_p(y)$ and $\text{ord}_q(y)$ are powers of 2. We're given that $\text{ord}_p(y) \neq \text{ord}_q(y)$, so WLOG we assume $\text{ord}_p(y) \leq \text{ord}_q(y)$. Say $\text{ord}_q(x) = 2^t$ for $0 \leq t < a$. Then:

- $y^{2^t} \equiv 1 \mod p$.
- $y^{2^t} \equiv 1 \mod q$.

So $\gcd(y^{2^t} - 1, N) = p$ is a non-trivial factor.

The second part will come later. □

19.2 RSA (Rivest, Shamir, Adleman), 1977[3]

We have $N = pq$ for p, q distinct primes. Recall that $\varphi(N) = (p-1)(q-1)$. Pick e such that $(e, \varphi(N)) = 1$. We solve for d such that $de \equiv 1 \mod \varphi(N)$. Then we have:

- Public key: (N, e)
- Private key: d

We then encrypt $m \in \mathcal{M} = \{0, 1, \dots, N-1\}$, as $c = m^e \mod N$. Then decrypt $c \in \mathcal{C}$ as $x = c^d \mod N$.

Corollary 19.4. *Finding the RA private key from the public key is essentially as difficult as factoring N.*

Proof. We've seen that factoring N allows us to find d. Conversely, if we know $d, c, de \equiv 1 \mod \varphi(N)$. Then $\varphi(N) | (de-1)$. We write $m = de - 1 = 2^a b$, and then use **19.3** to factor N □

[3]This was actually first come up with by Clifford Cocks in 1973, but it was classified as he was working for the British government

Proof of second part of 19.3. We want $|\{x \in (\mathbb{Z}/N\mathbb{Z})^\times : \text{ord}_p(x)^b) \neq \text{ord}_q(x^b)\}| \geq \frac{1}{2}|(\mathbb{Z}/N\mathbb{Z})^\times|$. Using the CRT we have correspondence:

$$(\mathbb{Z}/N\mathbb{Z})^\times \to (\mathbb{Z}/p\mathbb{Z})^\times \times (\mathbb{Z}/\mathbb{Z})^\times ; x \mapsto (x \mod p, x \mod q)$$

So it suffices to show that, if we partition $(\mathbb{Z}/p\mathbb{Z})^\times$ into subsets according to the value of $\text{ord}_p(x^b)$, then each subsets has size $\leq \frac{1}{2}|(\mathbb{Z}/p\mathbb{Z})^\times| = \frac{p-1}{2}$. We do this by showing that some such subset has size $\frac{1}{2}|(\mathbb{Z}/p\mathbb{Z})^\times|$. Now $(\mathbb{Z}/p\mathbb{Z})^\times = \langle g \rangle$, and by Fermat, $g^{p-1} \equiv 1 \mod p$, so $\text{ord}_p(g^b)|2^a$. So writing $x = g^\delta$ then $x^b = (g^b)^\delta$, so $\text{ord}_p(x^b) = \frac{2^t}{(2^t, \delta)}$, so $\text{ord}_p(g^b)$ for δodd is $\text{ord}_p(g^b)$, and is less for δ even. So $\{g^\delta \mod p : \delta \text{ odd}\}$ is the required set. The remaining $\frac{1}{2}(p-1)$ values all have strictly smaller size. □

19.3 Diffie-Hellman Key Exchange

Let p be a large prime and g a primitive root mod p. This data is fixed and known to everyone. A, B wish to agree on a secret key $k \in \mathbb{Z}_p$ to use. A chooses random $\alpha \in \mathbb{Z}_{p-1}$, and sends g^α mod p to B. Similarly, B chooses random $\beta \in \mathbb{Z}_{p-1}$ and sends $g^\beta \mod p$ to A. Then they both compute $k \equiv (g^\beta)^\alpha \equiv (g^\alpha)^\beta \mod p$, and use it as their common secret key. If E can compute discrete logarithms efficiently, she can find $\alpha = \log_g(g^\alpha)$ from the published values of p, g, g^α. Then E can compute the key as $(g^\beta)^\alpha$. Diffie and Hellman conjectured that finding the key from these public values is as hard as solving the discrete logarithm problem.

20 Secret Sharing

If CMS is attacked by MIO, the faculty retreats to MR2. Entry involves inputting an integer $S > 0$, the ***secret***, known only to the "Leader". Each of the n members of the Faculty knows a certain pair of numbers (their ***shadow***), and we require that, in Imre's absence, any $k \leq n$ members of the Faculty can reconstruct S from their shadows but no $k-1$ can. How can we do this? Any solution is known as a ***(k,n)-threshold scheme***. Here's one dues to Shamir.

- Choose S such that $0 < S < N$ at random.

- Imre chooses a prime $p > N$, then integers $a_1, \ldots, a_{k-1}$ at random, and distinct integers $x_1, \ldots, x_n$ at random with $0 \leq a_j \leq p-1$, $1 \leq x_j \leq p-1$, and then sets $a_0 = S$. He computes:

$$P(r) = a_0 + a_1 x_r + a_2 x_r^2 + \ldots + a_{k-1} x_r^{k-1} \mod p$$

for $1 \leq r \leq n$.

- He then gives the ***shadow pair*** $(x_r, P(r))$ to the r^{th} Faculty member and tells everyone the value of p.

Then k members can solve their systems of equations as the Vandermonde determinant is nonzero, and we find $a_0 = S$, but $k-1$ members don't have enough information.

21 Signatures

Consider the message m sent by A to B. They may be concerned abut:

- Secrecy: A, B are sure no third party has read m.
- Integrity: A, B are sure no third party has altered m.

- Authenticity: B is sure A sent m.
- Non-Repudiation: B can prove to a third party that A sent m.

Example: A uses a private key (N, d) to encrypt m using RSA. Anyone can decrypt m using the public key, but they cannot forge messages sent by A. B prepares a random message μ and sends it to A. If B then receives a message which, after decryption ends in μ, then he can be sure it came from ÆExample: (Homomorphism attack) The bank sends messages of the form (M_1, M_2) where M_1 is the client's name and M_2 is the amount to be transferred to M_1's account. Messages are encoded using RSA as $(z_1, z_2) = (M_1^e \mod N, M_2^e \mod N)$. If I transfer 100 to my account and observe the encrypted message (z_1, z_2), I could then send (z_1, z_2^3), and become a millionaire without breaking RSA, or I could keep sending (z_1, z_2). This can be defeated time-stamping the messages.

A message is signed as (m, s), where s is a function of m and the private key. The ***signature*** function $s : \mathcal{M} \times \mathcal{K} \to \mathscr{S}$ is designed so that no one without knowledge of the private key can sign messages, but anyone can check the signature is valid.

We can do signatures with RSA - if A has private key (N, d); public key e, then A signs m as $(m, m^d \mod N)$, the signature is verified by checking $s^e \equiv m \mod N$. There are some problems - the homomorphism attack still works, and anyone can produce random valid signed messages of the form $(s^e \mod N, s)$. Hopefully these messages won't be meaningful, but they might be.

To solve these problems, we can use a better signature scheme, or sign messages using a hash, i.e. sign $h(m)$ instead of just m, for example the md5sum.

Example: The ElGamal signature scheme:

A chooses a very large prime p, g a primitive root mod p, and some random integer u. Then the public key is $p, g, y = g^u$, and the private key is u. To send $m \in \mathbb{N}, A$ randomly chooses k coprime to $p-1$ and computes r, s satisfying $r = g^k \mod p, m = ur + ks \mod p-1$. A signs m with signature (r, s). Then $g^m = g^{ur+ks} \mod p = (g^u)^r (g^k)^s \mod p = y^r r^s \mod p$. B accepts the signature if $g^m = y^r r^s \mod p$. How do we forge such a signature? The obvious attempts involve solving the discrete logarithm problem.

Lemma 21.1. *Given a, b, m, the congruence $ax \equiv b \mod m$ has either 0 or $\gcd(a, m)$ solutions.*

Proof. Let $d = \gcd(a, m)$. If $d \nmid b$, then there are no solutions. Otherwise, rewrite this equation as $a/dx \equiv b/d \mod m/d$. Then $(a/d, m/d) = 1$, so this has a unique solution by Euclid, and so we have d solutions mod m. $\square$

It's important that A chooses a new k to sign each message. Otherwise, suppose messages m_1, m_2 have signatures (r, s_1) and (r, s_2), with $m_1 = ur + ks_1, m_2 = ur + ks_2$, giving $m_1 - m_2 = k(s_1 - s_2)$. Then by the previous lemma we can solve these by running through all possibilities for k, and then solve for u, allowing us to sign messages.

Several existential forgeries are known, but this can be stopped by replacing m by some collision resistant hash function $h(m)$, so that messages can be verified by checking $g^{h(m)} = y^r r^s$.

22 Bit Commitment

Suppose A, B are playing a board game. They decide who goes first by simulating a coin toss each. If they agree, A goes first and if they're different B goes first. To prevent cheating we could have A put her choice into a sealed envelope before B makes his choice. The aim of this protocol is to achieve a system whereby A can message B such that B cannot read the message until A allows him to, but A cannot change the contents of the message once it has been sent. Applications of this include things like voting, where we want results revealed correctly to all parties only after everyone has voted.

We can do this in (at least) two different methods:

1. Using a public key cipher: We have an encryption function e_A published by A and a decryption function d_A kept secret by A. Then A makes a choice $m \in \mathbb{F}_2$ and commits $e_A(m)$ to B. Then when the message is to be revealed, she sends B d_A and e_A, so that B can be sure d_A and e_A are inverse functions.

2. Using a noisy channel: We have two channels from $A \to B$, one which is a BSC with error probability $0 < p < \frac{1}{2}$ corrupting bits independently of A or B, and another which is clear. B chooses a binary linear code C with parameters n, d (length, minimum distance). A chooses a random surjective linear map $\theta : C \to \mathbb{F}_2$. To send a bit $m \in \mathbb{F}_2$, A chooses $c \in C$ such that $\theta(c) = m$ and sends c to B over the noisy channel. B receives $r = c + e$ and $d(r, c) = w(e)$ and the expected value of $w(e) \approx np$. Suppose the variance of the BSC is chosen small. Later A, sends c via the clear channel, and B can check $d(r, c) \approx np$. B can't read the message early because we choose the minimal distance of C to be much less than np. A can't change her choice, as she doesn't know the r received by B.