INFORMATION THEORY & CODING NOTES

AKSHANSH CHAUDHARY

Information Theory and Coding Notes, First Edition

Presented by: Akshansh Chaudhary
Graduate of BITS Pilani, Dubai Campus
Batch of 2011

Course content by: Dr. Anand Kumar
Then Faculty, BITS Pilani, Dubai Campus

Layout design by: AC Creations © 2013

The course content was prepared during Spring, 2014.
More content available at: www.Akshansh.weebly.com

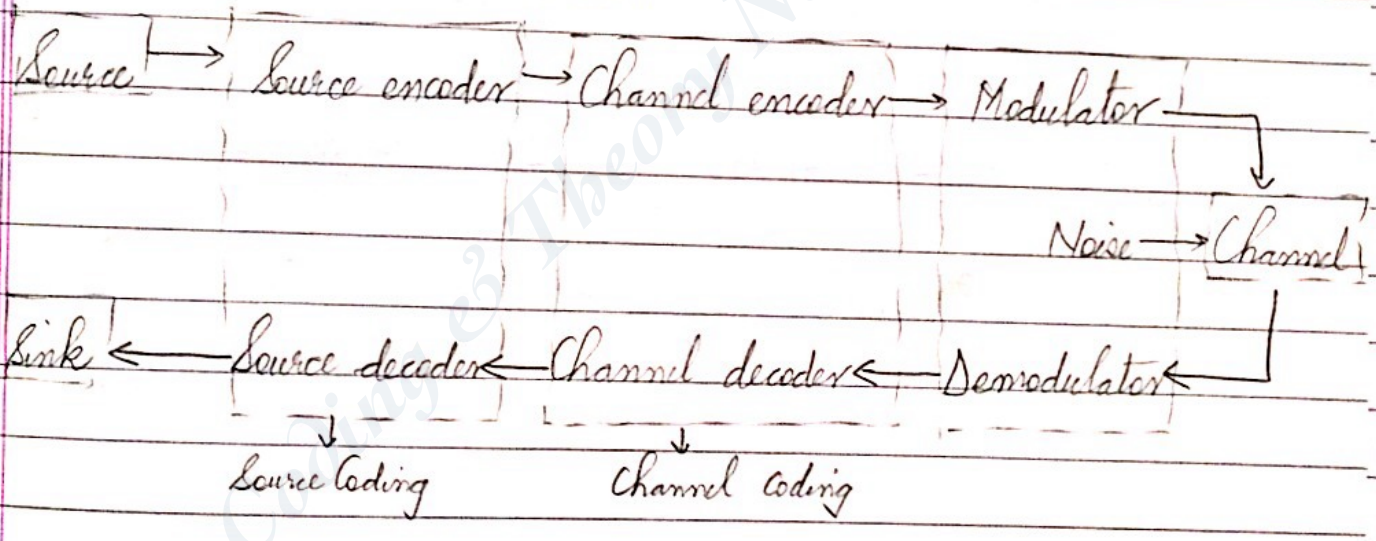
Puffin
Date 4/2/2014
Page

Info. Theory & Coding

✓ pre-requisite : probability
✓ topic deals with transmission of bits or using / manipulating bits to represent signals

Source Coding — Channel Coding

* COMM. SYS. BLOCK DIAGRAM

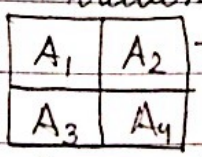


* Probability :

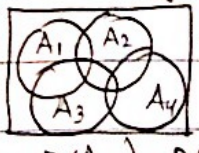
P1) Probability is ALWAYS positive.

* Venn-Diagram

↳ Notation : Universal set : Ω (written U before)



$P(A_1) + P(A_2) + P(A_3) + P(A_4) = 1$



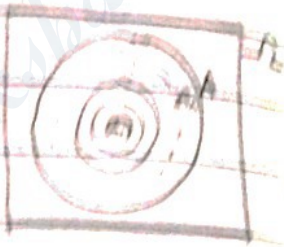
$P(A_1) + P(A_2) + P(A_3) + P(A_4) \neq 1$

P2) $P(\bar{A}) = 1 - P(A)$

P3) $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

P4) If $A \subset B$, $P(A) < P(B)$
If $A \subseteq B$, $P(A) \leq P(B)$

* P5) If $A_1 \subset A_2 \subset A_3 \ldots \subset A_n \subset A$
& $A \triangleq \bigcup A_n$
(A is defined as union of all A_n)



Then, $\lim_{n \to \infty} P(A_n) = P(\lim_{n \to \infty} A_n) = P(A)$

P6) If $A_1 \supset A_2 \supset A_3 - - - \supset A_n = A_1^c \subset A_2^c \quad \subset A_n^c \subset A^c$

$\Rightarrow P(A) = \lim_{n \to \infty} P(A_n)$

---

* BOREL CANTELLI LEMMA

---

* Random Variables :

Sample space $(\Omega, \mathcal{A})$
$X : \Omega \to R$
Cumulative distrib^n f^n, cdf, $F(x) = P(X \leq x)$

* Discrete Random Variable

Probability mass f^n, PMF, $P_X(x) = P(X = x)$

$$\text{cdf}, F(x) = \sum_{y \leq x} P_X(y)$$

* Continuous Random Variable

Probability distribution f^n, PDF,

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(y)\,dy$$

$$f(x) = \frac{d}{dx}(F(x))$$

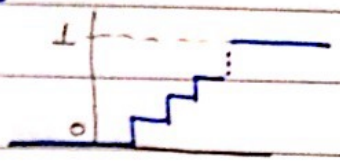
* Properties of Continuous Random Variables

P1) $F(x) \geq 0$

P2) $F(x)$ is right continuous

P3) $F(-\infty) = 0$

$F(\infty) = 1$



* Note: If 2 random variables are independent, then,

$$P(A \cap B) = P(A) \cdot P(B)$$

In general

$$P\left(\bigcap_{i=1}^{n} A_i\right) = \prod_{i=1}^{n} P(A_i)$$

Q. If $P(A \cap B) = P(A)\,P(B)$ (pairwise independent)
Are the random variables statistically independent?

Sol^n, Assume: $\Omega = \{1, 2, 3, 4\}$; $P(X = i) = 1/4$
↳ same probability ∀

$A = \{1, 2\}$
$B = \{1, 3\}$
$C = \{1, 4\}$

Now, $P(A \cap B) = P(\{1\}) = 1/4$
$P(B \cap C) = P(\{1\}) = 1/4$

Also, $P(A) = P(\{1, 2\}) = 1/2$
$P(B) = P(\{1, 3\}) = 1/2$
$P(C) = P(\{1, 4\}) = 1/2$

So, clearly,
$P(A \cap B) = P(A) \cdot P(B)$
& $P(B \cap C) = P(B) \cdot P(C)$

Now,
See if $P(A \cap B \cap C) = P(A) \cdot P(B) \cdot P(C)$

$P(A \cap B \cap C) = P(\{1\}) = 1/4$
But $P(A) \cdot P(B) \cdot P(C) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}$

So, A, B, C are not statistically independent
but are pairwise independent.

★ Conditional Probability:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad ; P(B) > 0$$

↳ If A, B are independent, $P(A|B) = P(A)$

In general

$$P\left(\bigcap_{i=1}^{n} A_i\right) = P(A_1) \cdot P(A_2 | A_1) \cdot P(A_3 | A_1, A_2) \cdots \cdots P(A_n | A_1, A_2, \ldots A_{n-1})$$

$$\left(\because P(A \cap B) = P(B) \cdot P(A|B) = P(A) \cdot P(B|A) \right)$$

* Expected value of X for discrete random variable,

$$E[X] = \sum_x x P(X = x)$$

$$E[h(X)] = \sum_x h(x) P(X = x)$$

$h(X)$ — fn of a random variable

$$E[X^n] = \sum_x x^n P(X = x)$$

E of n^{th} moment

$$Var[X] = E\left[(X - E[X])^2\right]$$

Variance

$$= E[X^2] - (E[X])^2$$

$$\sigma_X = \sqrt{Var[X]}$$

σ_X — standard deviation.

* I I D :

Independent & Identically distributed

* For X, Y : discrete random variables

$$P(X=x \mid Y=y) = \frac{P(X=x, Y=y)}{P(Y=y)} \; ; \; P(Y=y) > 0$$

cdf, $F(x|y) = P(X \leq x \mid Y=y)$

$$E[X \mid Y=y] = \sum_{x} x \, P(X=x \mid Y=y)$$

* Weak law of large numbers

for $X_1, X_2, \ldots$ sequence of I I D random variables,

Mean $= \mu$

Variance $= \sigma^2$

Let $S_n = \frac{1}{n} \sum_{k=1}^{n} X_k$

Then, $P(|S_n - \mu| \geq \delta) \to 0$ as $n \to \infty \; \forall \delta$.

* Markov Inequality

Suppose $X \geq 0$ for any $a > 0$

$$P(X \geq a) \leq \frac{E[X]}{a}$$

↳ Probability has an UPPER BOUND.

* $\log_2(\ ) = \frac{\log_{10}(\ )}{\log_{10} 2}$

* Chebyshev's Inequality

$$P(|X - E[X]| \geq \varepsilon) \leq \frac{Var[X]}{\varepsilon^2}$$

↳ Its random var. and expected value's difference

Entropy

defined as,

$$H(X) = \sum_{x \in X} P(x) \log_2 \left(\frac{1}{P(x)}\right)$$

↳ $P(x) = P(X = x)$

$$= -\sum_{x \in X} P(x) \log_2 P(x)$$

$$= E\left[\log \left(\frac{1}{P(x)}\right)\right]$$

* PROPERTIES.

P1) $H(X) \geq 0$

P2) $H_b(X) = (\log_a b) H_a(X)$ → entropy with base a

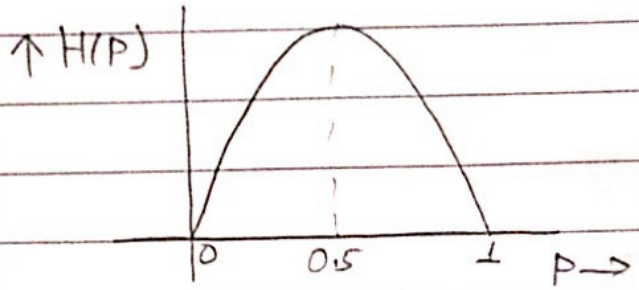
ex: If $X = \begin{cases} 1, & \text{prob} = p \\ 0, & \text{prob} = 1-p \end{cases}$

Q. find $H(X)$

$$H(X) = -[p \log p + (1-p)\log(1-p)]$$
$$\triangleq H(p)$$

If $p = 0$
$p \log p = 0 \times \infty = 0$, take

If $p = 1$
$(1-p)\log(1-p) = 0 \times \infty = 0$, take



ex(2): If X is defined as:-

$$X = \begin{cases} a, & \text{with prob} = 1/2 \\ b, & \text{prob} = 1/4 \\ c, & \text{prob} = 1/8 \\ d, & \text{prob} = 1/8 \end{cases}$$

Find entropy.

$$H(X) = -\left[\frac{1}{2}\log\left(\frac{1}{2}\right) + \frac{1}{4}\log\left(\frac{1}{4}\right) + \frac{1}{8}\log\left(\frac{1}{8}\right) + \frac{1}{8}\log\left(\frac{1}{8}\right)\right]$$

$$= -\left[\left(\frac{1}{2}\right)\left[0 - \log_2 2\right] + \left(\frac{1}{4}\right)\left[0 - \log_2 4\right] + \left(\frac{1}{8}\right)\left[0 - \log_2 2^3\right] + \left(\frac{1}{8}\right)\left(0 - \log_2 2^3\right)\right]$$

$$= -\left[\left(\frac{1}{2}\right)(-1) + \left(\frac{1}{4}\right)(-2) + \left(\frac{1}{8}\right)(-6)\right]$$

$$= 7/4 \text{ bits}$$

✱ JOINT ENTROPY

$$H(X,Y) = -\sum_x \sum_y P(x,y) \log P(x,y)$$

$$= -E[\log P(x,y)]$$

✱ CONDITIONAL ENTROPY

$$H(Y/X) = \sum_x P(x)\, H(Y|X=x)$$

$$= -\sum_x P(x) \sum_y P(y|x) \log P(y|x)$$

$$= -\sum_x \sum_y P(x,y) \log P(y|x)$$

$$= -E_{P(x,y)}[\log P(y|x)]$$

($\because P(x) \cdot P(y|x) = P(x,y)$)

✱ Theorem 2.2.1 : Chain Rule

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

Proof :-

by defin^n : $H(X,Y) = -\sum_x \sum_y P(x,y) \log P(x,y)$

$P(x,y) = P(x)\,P(y|x)$

$$= -\sum_x \sum_y P(x,y) \log [P(x) \cdot P(y|x)]$$

($\log ab = \log a + \log b$)

$$= -\sum_x \sum_y P(x,y) [\log P(x) + \log P(y|x)]$$

$$= -\sum_x \sum_y P(x,y) \log P(x)$$

$$-\sum_x \sum_y P(x,y) \log P(y|x)$$

$\rightarrow H(Y|X)$

$$= -\sum_x \log P(x) \sum_y P(x,y)$$

$\rightarrow P(x)$ (Marginal distrib^n)

$$+ H(Y|X)$$

$$= -\sum_x P(x) \log P(x) + H(Y|X)$$

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

Hence Proved.

Corollary : $H(X,Y) = H[X|Z] + H(Y|X,Z)$

Problem ①

X \ Y	1	2	3	4	P(X)
1	1/8	1/16	1/32	1/32	1/4
2	1/16	1/8	1/32	1/32	1/4
3	1/16	1/16	1/16	1/16	1/4
4	1/4	0	0	0	1/4
P(Y)	1/2	1/4	1/8	1/8	

$P(x,y)$ at $x=1=y$

$P(Y=1)$

$P(X=4, Y=1) = 1/32$

$\rightarrow P(X=1)$

$P(x,y)$

Given the table, find : $H(X)$, $H(Y)$, $H(X,Y)$, $H(X|Y)$, $H(Y|X)$

We know:

$$H(X) = -\sum_{x} P(x) \log P(x)$$

$$H(Y) = -\sum_{y} P(y) \log P(y)$$

From table,

$$H(X) = -[1/2 \log 1/2 + 1/4 \log 1/4 + 2 \times 1/8 \log 1/8]$$
$$= 7/4 \text{ bits}$$

& $H(Y) = -\left[\left(1/4 \log(1/4)\right) \times 4\right]$

$\Rightarrow H(Y) = -[-\log_2 4] = \log_2 2^2 = 2 \log_2 2 = 2$ bits

Now, $H(X|Y) = \sum_{i=1}^{4} P(Y=i) H(X|Y=i)$

$\hookrightarrow 1/4, Y = 1, 2, 3, 4$ (from table)

$\hookrightarrow H(X|Y=i) = -\sum_{y=1}^{4} P(x|y) \log P(x|y)$

$\hookrightarrow \frac{P(x,y)}{P(y)} = \frac{P(x,y)}{1/4}$

$= 4 P(x,y)$

for $H(X|Y=1) =$

$$-\left[4 \times \frac{1}{8} \log\left(4 \times \frac{1}{8}\right) + 4 \times \frac{1}{16} \log\left(4 \times \frac{1}{16}\right) + 4 \times \frac{1}{32} \log\left(4 \times \frac{1}{32}\right) + 4 \times \frac{1}{32} \log\left(4 \times \frac{1}{32}\right)\right]$$

$$= -\left[\frac{1}{2} \log(1/2) + (1/4) \log(1/4) + (1/8) \log 1/8) + (1/8) \log(1/8)\right]$$

$$= H(1/2, 1/4, 1/8, 1/8)$$

$\rightarrow 1/32 \div 1/4$ (table)

By, $H(X|Y=2) = H\left(\frac{1}{4}, \frac{1}{2}, \frac{1}{8}, \frac{1}{8}\right)$

After solving & putting values,

$$H(X|Y) = \frac{11}{8} \text{ bits}$$

Now, do $H(Y|X) = \sum_{i=1}^{4} P(X=i) H(Y|X=i)$

$$= \frac{13}{8} \text{ bits} \quad (\text{from computation})$$

So, we find $H(X|Y) \neq H(Y|X)$

Now,

$$H(X,Y) = \sum_{x=1}^{4} \sum_{y=1}^{4} P(x,y) \log P(x,y)$$

take = 0

$$= -\left[\underbrace{\frac{1}{8}\log\left(\frac{1}{8}\right)}_{x=1,\, y=1} + \underbrace{\frac{1}{16}\log\left(\frac{1}{16}\right)}_{x=2,\, y=1} + \ldots \underbrace{0 \log 0}_{x=4,\, y=4} \right]$$

$$= \frac{27}{8} \text{ bits}$$

Ans

* Entropy of random variable is a measurement of uncertainty of random variable.

  - like, for $P=0 \Rightarrow$ It is certain that it won't occur.
  - $P=1 \Rightarrow$ It is certain that it will definitely occur.
  - $H(P)=0$

* Relative entropy is the measure of distance b/w distribⁿˢ.

* KULLBACK LEIBLER (KL) distance

$$D(p\|q) = \sum P(x) \log \frac{P(x)}{Q(x)}$$

$$= E_p\left[\log \frac{P(x)}{q(x)}\right]$$

↳ If $p(x) = q(x)$, $D(p\|q) = 0$

≡ ite like relative entropy $\left(\log \frac{P(x)}{Q(x)}\right.$

$= \log P(x) - \log Q(x)$

eg: Its use in stock market
If actual distrib^n = $p(x)$
estimated distrib^n = $q(x)$
& we find $p(x)$ is close to $q(x)$ [by finding $D(p\|q)$ & seeing 'D' is small] we see our estimation was correct

* Note: $D(q\|p) \neq D(p\|q)$

ie $\left[\sum_x q(x) \log \frac{q(x)}{p(x)} \neq \sum_x p(x) \log \frac{p(x)}{q(x)}\right]$

eg Given: $P(0) = P(1) = 1/2$
$q(0) = 3/4$, $q(1) = 1/4$
Using above formula, we find that
$D(p\|q) = 0.2075$ bits & $D(q\|p) = 0.1887$ bits

✡ Mutual information
(b/w 2 random variables x & y)

$$I(X;Y) = \sum_x \sum_y P(x,y) \log \frac{P(x,y)}{P(x)\,P(y)}$$

$$D(P(x,y) \,||\, P(x)P(y))$$

Joint distribⁿ of x & y

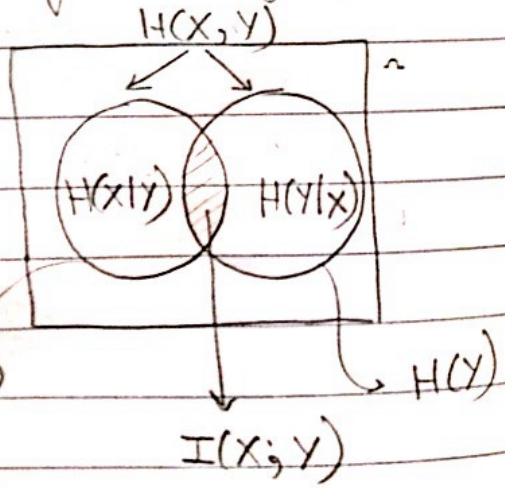
x, y are independent

✡ THEOREM 2.4.1

$$I(X;Y) = H(X) - H(X|Y)$$
$$= H(Y) - H(Y|X)$$
$$= H(X) + H(Y) - H(X,Y)$$

Joint entropy of X, Y

$$I(X;X) = H(X)$$
$$I(X;Y) = I(Y;X)$$



Problem ① from Problem ① (done before)
continued.

$H(X) - H(X|Y) = 7/4 - 11/8 = 3/8 = I(X;Y)$
$H(Y) - H(Y|X) = 2 - 13/8 = 3/8 = I(X;Y)$
$H(X) + H(Y) - H(X,Y) = 7/4 + 2 - 27/8 = I(X;Y)$

} Same ∴ Verified

Eg English language, considering 26 letters

P (in any text a particular letter occurs) $\frac{1}{26}$

Entropy, $H(P) = -[p \log p + (1-p)\log(1-p)]$

$= 4.7$ bits

So, on an average, 5 bits are req^d

Considering probability of occurence into account, (varying for diff^t letters), we get no. of bits = 4.19 bits

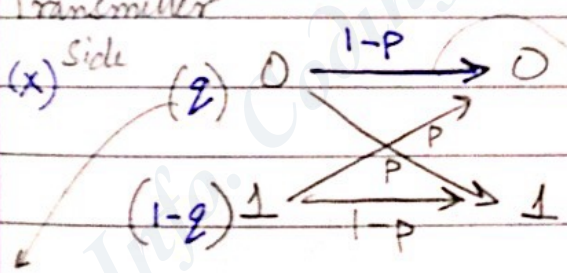
Further considering cases like T→h, Q→u pairs, no. of bits (entropy) decreases to 3.56 bits

Going on similar lines, on an average, we get

$$1 \leq H_L \leq 1.5 \text{ bits}$$

(So, data compression in some way)

* Binary Symmetric Channel (BSC)



$P(Y=0|X=0) = 1-p$

$= P(Y=1|X=1)$

$P(Y=0|X=1) = p = P(Y=1|X=0)$

→ i.e., Probability that 1 was transmitted & 9 get 0 at receiver (or vice versa). So, 9 am finding Probability of error.

$$H(X) = -\sum_{i=1}^{2} P_i \log P_i$$

(entropy of source)

$$= -[q \log q + (1-q)\log(1-q)]$$

$H(X|Y)$: Entropy of x given Y. It is of importance ∵ we are at receiver side & are seeing what was transmitted.

$$H(X|Y) = -\sum_{i=1}^{2}\sum_{j=1}^{2} P(x_i, y_j) \log\left[P(x_i|y_j)\right]$$

$P(x_i, y_j)$: Joint probability; $P(x_i|y_j)$: cond^{nal} probability

We know,

$$P(x,y) = P(x|y)\,P(y) = P(y|x)\,P(x)$$

So, find $P(y|x)\,P(x)$ & replace in above eq^{n}

here, $P(y_1|x_1) = P(0|0) = 1-p$.

Ily $P(y_1|x_2) = P(0|1) = p$

$P(y_2|x_1) = P(1|0) = p$

$P(y_2|x_2) = P(1|1) = 1-p$

Also, $P(x_1) = q$, $P(x_2) = 1-q$.

Now, finding $P(y)$:

$$P[y=0] = P[x=0]\,P[y=0|x=0] + P[x=1]\,P[y=0|x=1]$$

Ily, $$P[y=1] = P[x=0]\,P[y=1|x=0] + P[x=1]\,P[y=1|x=1]$$

So, we know $P(y|x)$, $P(x)$, $P(y)$

find $P(x,y)$

find $P(x|y)$

ie $P(x_1,y_1)$, $P(x_2,y_2)$, $P(x_1,y_2)$, $P(x_2,y_1)$

* Theorem 2.5.1 : (Chain rule for entropy)

Say, $X_1, X_2, \ldots, X_n \sim P(x_1, x_2, \ldots, x_n)$

$$H(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} H(X_i | X_{i-1}, \ldots, X_1)$$

(Proof : textbook)

$$= H(X_1) + H(X_2|X_1) + H(X_3|X_2, X_1) + \cdots$$

* Theorem 2.5.2 : (Chain rule for inform^n)

$$I(X_1, X_2, \ldots, X_n; Y) = \sum_{i=1}^{n} I(X_i; Y | X_{i-1}, \ldots, X_1)$$

$$= I(X_1; Y) + I(X_2; Y|X_1) + I(X_3; Y|X_2, X_1) + \cdots$$

* Theorem 2.5.3 : (Chain rule for relative entropy)

$$\underbrace{D(P(x,y)\|q(x,y))}_{2D} = \overbrace{D(P(x)\|q(x))}^{1D \text{ distance}} + \cancel{D(p(y|x))} + D(p(y|x)\|q(y|x))$$

Distance b/w cond^{nal} prob^s, 1D

* Theorem 2.6.2 (Jensen's Inequality)

If f is a convex function,
X is a random variable,

$$E[f(x)] \geq f(E[X])$$

If f is strictly convex,
$E[f(X)] = f(E[X]) \Rightarrow X = E[X]$ with Prob 1 $\Rightarrow$ X is a constant

* Theorem 2.6.3 (Info inequality)

$D(P\|Q) \geq 0$ with equality iff $p(x) = q(x)\ \forall\ x$

* Corollary :

$I(X;Y) \geq 0$

↳ with equality when X, Y are independent

* Corollary :

$D(P(y|x)\|Q(y|x)) \geq 0$

↳ with equality when $p(y|x) = q(y|x)$

* Corollary :

$I(X;Y|Z) \geq 0$

↳ with equality if X & Y are conditionally independent (ie, X & Y are independent, given Z)

* Theorem 2.6.4

$H(X) \leq \log_2 M$ (entropy of any random variable has upper bound)

↳ M: no. of elements in range of X.

eg :- for 1 bit transmission, message = [0 1]

So, $M = 2$.

So, $H(X) \leq 1$.

for 2 bit, message = $\begin{bmatrix} 0\,0 & 1\,0 \\ 0\,1 & 1\,1 \end{bmatrix}$

$\Rightarrow M = 4$.

$\Rightarrow H(X) \leq 2$.

equality iff X: uniform random variable $\forall M$

* Message is transmitted in SYMBOLS
↳ for binary, its called BITS

Puffin
Date
Day

* Theorem 2.6.5 : (Conditioning reduces entropy)
$H(X|Y) \leq H(X)$
i.e equality $\iff$ X, Y are independent.

eg:

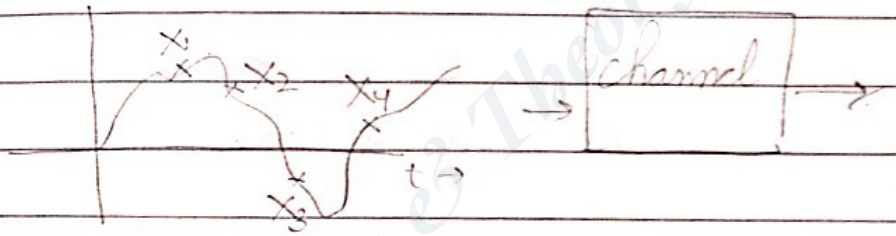
Y \ X	1	2
1	0	3/4
2	1/8	1/8

We find from table, $H(X) = 0.5$
$H(X|y=1) = 0$ bits
$H(X|y=2) = 1$ bit

$H(X|Y) = P[y=1] \cdot H[X|y=1] + P[y=2] H[X|y=2]$

* Theorem 2.6.6 (Independence bound on entropy)

$X_1, X_2, X_3 \ldots, X_n \sim P(X_1, X_2, \ldots, X_n)$



$$H(X_1, X_2, \ldots, X_n) \leq \sum_{i=1}^{n} H(X_i)$$

↳ equality, if X_i are ~~so~~ statistically independent.

* Theorem 2.7.4 (Preceeding theorems 2.7.1, 2.7.2, 2.7.3)
$(X, Y) \sim P(x, y) = \underbrace{P(x)}_{\equiv \text{source coding}} \underbrace{P(y|x)}_{\equiv \text{channel coding}}$

$I(X; Y) \equiv$:-

properties
(P1) Is a concave fn of $P(x)$ for fixed $P(y|x)$
(P2) A convex fn of $P(y|x)$ for fixed $P(x)$

✱ DATA PROCESSING INEQUALITY

Let ∃ 3 random variables X, Y & Z.

~~If~~ $X \to Y \to Z$ (≡ X, Y, Z form a Markov chain)

↳ They form a Markov chain

iff $P(x, y, z) = P(x) \cdot P(y|x) P(z|y)$

(i.e., given Y, X & Z are independent of each other)

① i.e., $X \to Y \to Z$ iff {x, z are conditionally independent given y.

$$\Downarrow$$

$$P(x, z|y) = P(x|y) P(z|y)$$

② $X \to Y \to Z \Rightarrow Z \to Y \to X$

③ If $Z = f(Y) \Rightarrow X \to Y \to Z$.

(Way to see: Consider a time scale:

X_{n-1} X_n X_{n+1} $n \to$

If $X_{n-1} \equiv X$, $X_n \equiv Y$, $X_{n+1} \equiv Z$

$\equiv$ Past $\equiv$ Present $\equiv$ Future.

So, Future depends only on present, not in past)

✱ Theorem 2.8.1:

If $X \to Y \to Z$ (X, Y, Z form a Markov chain)

then,

$$I(X;Y) \geq I(X;Z)$$

(, Mutual info.

* Corollary : If $Z = g(Y)$
then, $I(X;Y) \geq I(X;g(Y))$

* Corollary : If $X \to Y \to Z$
then $I(X;Y|Z) \leq I(X;Y)$

—x—

* Entropy : General info :

① Relative entropy can decrease

$$D(\mu_n \| \mu_n') \geq D(\mu_{n+1} \| \mu_{n+1}')$$

n → time instant

$\mu_n, \mu_n' \equiv p(x), q(x)$

② $D(\mu_n \| \mu) \geq D(\mu_{n+1} \| \mu)$

satisfies Markov chain property

(Stationary process)

$\Rightarrow \exists$ no variation with time

③ Entropy increases if stationary distribⁿ is uniform.

$$D(\mu_n \| \mu) = \log M - H(\mu_n)$$
$$= \log M - H(X_n)$$

eg : for 7 bits in a code,
$M = 3$.

If $n \uparrow$ (time ↑)
$\Rightarrow D(\mu_n \| \mu) \downarrow$
$\Rightarrow H(X_n) \uparrow$

cond^nl entropy

④ If $n \uparrow \Rightarrow H(X_n|X_1) \downarrow$ for stationary Markov process.

⑤ $H(T[X]) \geq H(X)$

↳ T ≡ statistic, say, shuffling cards (& finding entropy)

✱ Sufficient Statistics

If $\exists$ a no. of random variable,

consider $\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} X_i$

↳ How do we know exact requirement & if it's good.

• If $T[X]$ is a statistic,

$$\theta \rightarrow X \rightarrow T[X]$$

~~$\theta \rightarrow X \rightarrow T[X]$~~

$$I(\theta; X) \geq I(\theta; T[X])$$

✓ $\theta \rightarrow T[X] \rightarrow X$

$$\Downarrow$$

$$I(\theta; X) = I(\theta; T[X])$$

→ cond^nal entropy

④ If $n \uparrow \Rightarrow H(X_n|X_1) \downarrow$ for stationary Markov process.

⑤ $H(T[X]) \geq H(X)$

$T \equiv$ statistic, say, shuffling cards & finding entropy

✱ Sufficient Statistics

If ∃ a no. of random variable,

Consider $\hat{\mu} = \frac{i}{N} \sum_{i=1}^{N} X_i$ → Sufficient statistic.

↳ how do we know exact requirement & if it's good

- If $T[X]$ is a statistic,

$$\theta \rightarrow X \rightarrow T[X]$$

~~$\theta \rightarrow X \rightarrow T[X]$~~

$$I(\theta; X) \geq I(\theta; T[X])$$

✓ $\theta \rightarrow T[X] \rightarrow X$

$$\Downarrow$$

$$I(\theta; X) = I(\theta; T[X])$$

✱ Fano's Inequality

If we want to estimate a random variable X with probability distribution $p(x)$, we observe random variable Y

$X \rightarrow$ [Channel] $\rightarrow Y$

(T_x) (R_x)

We want estimate of Y.

So, we calculate $g(Y) = \hat{X}$

Error $\equiv f(\hat{X} - X)$

And, Probability of error $= P(\epsilon) = P(\hat{X} - X)$

Now, we want a bound on probability

$P(E) = P(\hat{X} - X) \leq \alpha \Rightarrow (?)$

We find:

$X \rightarrow Y \rightarrow \hat{X}$, form a Markov chain

$\& P_e = P(E) = P(\hat{X} \neq X)$ → (1)

we need to choose P_e in such a way that both are satisfied;

Fano's inequality:

$$H(P_e) + P_e \log(M-1) \geq H(X|Y) \rightarrow (2)$$

↓ uncertainty of what was transmitted

→ Weaker form:

$$P_e \geq \frac{H(X|Y) - 1}{\log M}$$

for $M = 2$

$\Rightarrow P_e \geq H(X|Y) - 1$

$\Rightarrow H(P_e) \geq H(X|Y)$

for $M = 2$

$\Rightarrow H(P_e) + (\log 3) P_e \geq H(X|Y)$

non linear eqn in P_e

from next calculations, $P_e \geq 0.307$ ← for $m > 1$

eg from previous example

$H(X|Y) = \frac{11}{8}$

Idea: eqns (1) & (2) should be satisfied

Using weaker form:-

$$\frac{(11/3) + 1}{\log 3} = \frac{22}{5}$$

Thomas

Q. 21 $X_1 \to X_2 \to \cdots \to$ form a Markov Chain

P.T. $\Rightarrow P(x_1, x_2, \ldots, x_n) = P(x_1) \cdot P(x_2|x_1) \cdots$

$I(X_1; X_2; \ldots; X_n)$

$$\text{Say } \sum^{2}(X, Y) = \left[\sum_x \sum_y P(x,y) \log \frac{P(x,y)}{P(x)\,P(y)}\right]$$

$$= \sum_{x_1} \sum_{x_2} \cdots \sum_{x_n} P(x_1, x_2, \ldots, x_n) \times \log \frac{P(x_1, x_2, \ldots, x_n)}{P(x_1)\,P(x_2, x_3, \ldots x_n)}$$

$$= \frac{P(x_1)\,P(x_2|x_1) \cdots P(x_n|x_{n-1})}{P(x_1)\,P(x_2)\,P(x_3|x_2) \cdots P(x_n|x_{n-1})}$$

$$\equiv \frac{P(x_2|x_1)}{P(x_2)} = \frac{P(x_1, x_2)}{P(x_1) \cdot P(x_2)}$$

$$= \sum_{x_1} \sum_{x_2} P(x_1, x_2) \log \frac{P(x_1, x_2)}{P(x_1, x_2)}$$

$$= I(X_1; X_2)$$

(i.e., consider only for 2 samples)

* Theorem 4.4.1 :

If $X_1, X_2, \ldots, X_n$ form a Stationary Markov Chain, & $Y_i = \phi(X_i)$

$$H(Y_n|Y_{n-1}, \ldots, Y_1, X_1) \leq H(Y) \leq H(Y_n|Y_{n-1}, \ldots, Y_1)$$

and

$$\lim_{n\to\infty} H(Y_n|Y_{n-1},\ldots,Y_1,X_1) = H(Y) = \lim_{n\to\infty} H(Y_n|Y_{n-1},\ldots,Y_1)$$

→ $H(Y)$: entropy rate of a STOCHASTIC process generates random variables. [5]

→ here, we have 2 stochastic processes :

- Y : generating $Y_1, Y_2, \ldots Y_n$.
- X : generating $X_1, X_2, \ldots, X_n$

→ related by ϕ.

end of Ch-4

* Code Word (CW)

Ch-5
SOURCE CODING

Defin^n: If $C : X \rightarrow D^*$ (C: σ for from $X \rightarrow D^*$)
$l(x)$ = length of $C(x)$ (length of code / no. of bits)

eg:- $X = \{Red, Blue\}$
$D = \{0, 1\}$ ← an alphabet
$D^* = \{00, 11\}$

We can have a D-ary alphabet $D = (0, 1, \ldots D-1)$
Here, its binary $(D=2)$
$D = (0, 1)$

I want to encode X. So, code for that,
$C(Red) = 00$ & $C(Blue) = 11$
So, $l(Red) = 2$ → 2 bits
$l(Blue) = 2$

* Expected length of source code, $C(X)$

$$L(C) = \sum_x P(x) \cdot l(x)$$

eg:
$P[X=1] = 1/2$; Codeword, $C(1) = 0$
$P[X=2] = 1/4$ (CW) $C(2) = 10$
$P[X=3] = 1/8$ $C(3) = 110$
$P[X=4] = 1/8$ $C(4) = 111$

↳ A D-ary alphabet, with $D=4$

$H(X) = H(1/2, 1/4, 1/8, 1/8) = 1.75$ bits (by prev. formula)

& $L(C) = \sum_x P(x) l(x)$

$= \sum_{x=1}^{4} P(x) l(x)$

$$\Rightarrow L(C) = \left(\frac{1}{2}\right)(1) + \left(\frac{1}{4}\right)(2) + \left(\frac{1}{8}\right)(3) + \left(\frac{1}{8}\right)(3)$$

$$\left[P(1)\,l(1) + P(2)l(2) + P(3)l(3) + P(4)l(4)\right]$$

$\Rightarrow L(C) = 1.75$ bits

(= same as $H(X)$; not always true)

Now, suppose I am decoding a sequence. I have the code :-

1 → 0
2 → 10
3 → 110
4 → 111

} All these CWs together form a CODE BOOK or CODE

Code :- 0 | 110 | 111 | 10 | 0 | 110.

1 3 4 2 1 3.

So, it is 134213 ← decrypted CW.

* Defin^n: Code is non singular, if

$$x_i \neq x_j \Rightarrow C(x_i) \neq C(x_j)$$

i.e,

1 → 0	1 → 0
2 → 10	2 → 10
3 → 110	3 → 10
4 → 111	4 → 111
Non singular Code Book	Singular Code / Code Book

* Extension, C^* of a code is defined as

$$C(x_1, x_2, \ldots, x_n) = C(x_1) \cdot C(x_2) \cdot C(x_3) \ldots C(x_n)$$

ex :- $C(x_1) = 00$

$C(x_2) = 11$

So, $C(x_1 x_2) = C(x_1)\, C(x_2)$

$= 0011$

* Defin^n : Uniquely decodable code, if extension is non singular.

* eg : $X = 1 \longleftrightarrow 0$

$X = 2 \longleftrightarrow 01$

$X = 3 \longleftrightarrow 110$

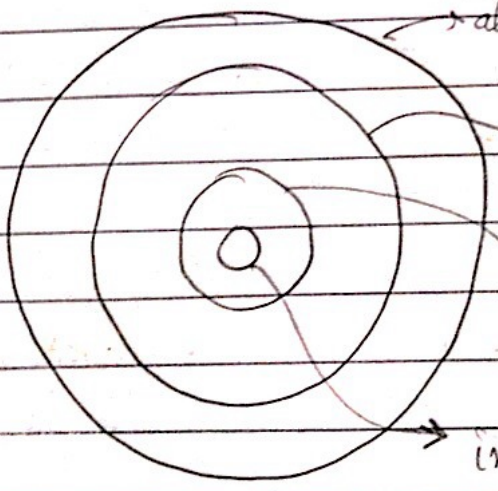
$X = 4 \longleftrightarrow 111$

for the code $\underbrace{0}_{2}\ \underbrace{11}_{4}\ \underbrace{111}_{4}\ \underbrace{1}_{1}\ \underbrace{0}_{2}\ \underbrace{11}_{3}\ \underbrace{0}_{2}$

↳ We not 1.
we have to visit after $\underbrace{0 \to 1}_{2}$

* Defin^n : A code is called a ~~pre~~ prefix code or an instantaneous code if no CW is a prefix of any other CW

* Graphically,



eg: Consider a set: Non singular but not uniquely decodable :-

{ 0, 010, 01, 10 }

→ Combining 1, 2
= 0010 ←

→ Combining 1, 3
= 001 → prefix of 0010

→ Combining 1, 4
= 010

→ Combining 2, 1
= 0100

→ Combining 2, 3
= 01001

→ Combining 1, 2, 3 & 4
= 00100110

So, not instantaneous code

But, all codes are Non-singular (none repeats)

So, although above are non instantaneous codes, they are uniquely decodable codes.

eg ②. Code like
0
10
110
111

No codeword is prefix of any other CW. So, Its instantaneous code.

This code is also uniquely decodable

* Theorem 5.2.1 : (Kraft inequality)
For any instantaneous code (prefix code) over an alphabet Σ of size D, the CW lengths $l_1, l_2, \ldots, l_n$ must satisfy the inequality :

$$\sum_i D^{-l_i} \leq 1$$

Conversely, given a set of codeword lengths that satisfy this inequality, $\exists$ an instantaneous code with these word lengths

eg :- With code :- 0 $\rightarrow l_1 = 1$
10 $l_2 = 2$
110 $l_3 = 3$
111 $l_4 = 4$

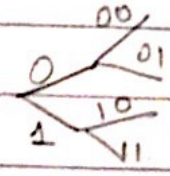
Condition to Code : with $D = 2$ (binary code)
we can write :-

$$2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} \leq 1$$

↳ will be true

Condition to Code : Conversely, if $l_1 = l_2 = l_3 = l_4 = 2$
$\Rightarrow$ Code :- 00
01
10
11

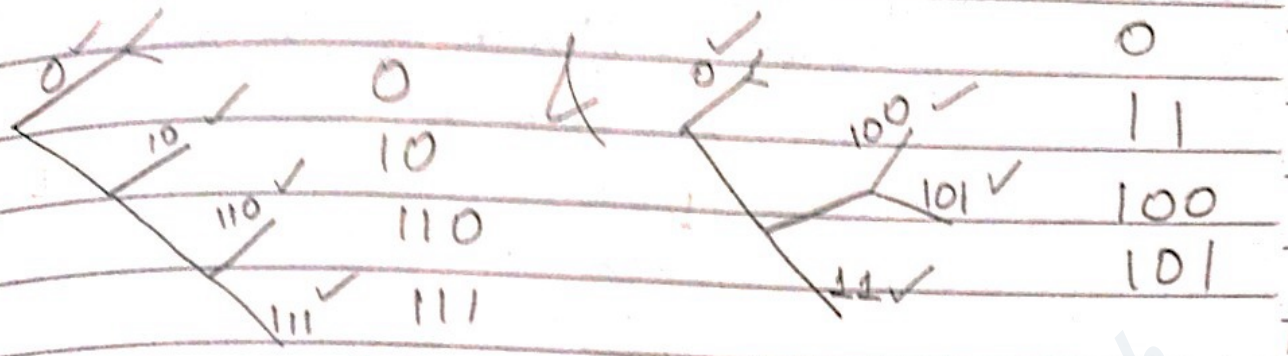
Code construction done by tree diagram



* For any countably infinite set of codewords that form a prefix code, CW lengths satisfy EXTENDED KRAFT INEQUALITY :- $\sum_{i=1}^{\infty} D^{-l_i} \leq 1$

↳ Converse is true.

Make Instantaneous code :



* Construction of Instt. code is (not) unique, but, while making the codes, they should satisfy the length requirement ($l_1=1, l_2=2, l_3=3, l_4=3$) (or any other requirement depending on question)

* Theorem 5.3.1 :
Expected length 'L' of any instt. D-ary code for a random variable X is $\geq$ entropy $H_D(X)$

i.e,

$$L \geq H_D(X)$$

↳ equality iff $D^{-l_i} = P_i$

eg: consider code : 0, 10, 110, 111

$P_1 = 1/2 = 2^{-1}, l_1 = 1$

$P_2 = 1/4 = 2^{-2}, l_2 = 2$

$P_3 = 1/8 = 2^{-3}, l_3 = 3$

$P_4 = 1/8 = 2^{-3}, l_4 = 3$ → D^{-l_i}

$L = \sum P_i l_i = 1.75$ bits

& $H_2(X) = H\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right) = 1.75$ bits, same

* We are concerned with length of code (these), not their values.

i.e 0, 10, 110, 111 ≡ 11, 100, 101. (lengths are being satisfied)

Defin^{n}: A probability distrib^{n} is called D-adic w.r.t D if each of the probabilities equals D^{-n} for some n.

eg: let $P(X=1) = 1/3$

$P(X=2) = 1/3$

$P(X=3) = 1/3$.

& $H_2(X) = H(1/3, 1/3, 1/3)$

&

let codes = 0, 10, 11

Now,

$$L = \sum p_i l_i = \frac{1}{3}\times 1 + \frac{1}{3}\times 2 + \frac{1}{3}\times 2$$

$$= \frac{5}{3} = 1.67$$

Now

$$H(1/3) = \frac{1}{3}\log(1/3) = \frac{-1}{3}\frac{\log_{10} 3}{\log_{10} 2}$$

$$= -\frac{1}{3}\left(\frac{0.48}{0.3}\right)^{0.16}$$

$$\simeq -\frac{0.16}{.3}$$

$$\simeq -0.5$$

So,

$$H_2(X) = -[H(1/3)\times 3]$$

$$= -(-0.5\times 3)$$

$$= 1.5.$$

So, $1.5 \leq L^* (= 1.67) \leq 1.5+1$

Theorem 5.4.3

By Theorem $l_1^*, l_2^*, \ldots, l_m^*$ be the optimal CW lengths for a source distrib^{n} p & a D-ary alphabet, & let L^* be associated expected length of optimal code ($L^* = \sum p_i l_i^*$), Then,

$$H_D(X) \leq L^* < H_D(X) + 1$$

L_n^*

* Theorem 5.4.2 : The min. expected codeword length per symbol satisfies :

$$\frac{H(X_1, X_2, \ldots, X_n)}{n} \leq L_n^* < \left[\frac{H(X_1, X_2, \ldots X_n)}{n} + Y_n\right]$$

↳ If $X_1, X_2, \ldots, X_n$ is stationary stochastic process

$$L_n^* \longrightarrow H(\mathcal{X})$$

$\rightarrow H(X_1, X_2, \ldots X_n)$

↳ $H(\mathcal{X})$: entropy rate of the process.

* Defin^n : The entropy rate of stochastic process $\{X_i\}$ is defined by :

$$H(\mathcal{X}) = \lim_{n\to\infty} \frac{1}{n}\left(H(X_1, X_2, \ldots, X_n)\right) \longrightarrow (1)$$

↳ when limit exists. ↳ time index

$$\text{A } H'(\mathcal{X}) = \lim_{n\to\infty} H(X_n | X_{n-1}, X_{n-2}, \ldots, X_1) \longrightarrow (2)$$

* Theorem 4.2.1 : For stationary stochastic process, limits in eq^n (1) & eq^n (2) exists & are equal.

i.e, $H(\mathcal{X}) = H'(\mathcal{X})$

* Theorem 5.4.3 : The expected length under $p(x)$ of the code assignment $l(x) = \left\lceil \log \frac{1}{q(x)} \right\rceil$ satisfies :

$\Rightarrow$ take upper value (ceil value)

$$H(p) + D(p\|q) \leq E_p\, l(X) < H(p) + D(p\|q) + 1$$

$$\left(\equiv H_D(X) \leq L^* < H_D(X) + 1.\right)$$

Note:

$\lceil 0.4 \rceil = 1 = \text{ceil}(0.4)$

$\lfloor 0.4 \rfloor = 0 = \text{floor}(0.4)$

$0.4 \simeq 0. = \text{round}(0.4)$

★ Theorem 5.5.1 : (McMillan)
CW lengths of any uniquely decodable code must satisfy Kraft inequality:

$$\sum D^{-l_i} \leq 1$$

(converse is also true)

★ (5.2.1) Kraft Inequality : (done before)
For any instt code over alphabet of size D, CW lengths $l_1, l_2, \ldots, l_m$ must satisfy

$$\sum_i D^{-l_i} \leq 1.$$

→ Basically, ∵ instt. code ⊂ uniquely decodable
⇒ anything that satisfies instt. code, has to satisfy uniquely decodable.

eg Huffman codes example (done in Communic^n Systems)

Given :

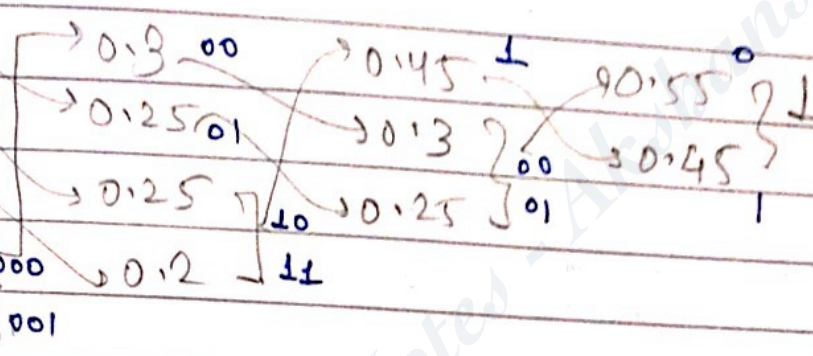
X	P_i	X	P_i
1	0.25	3	0.2
2	0.25	4	0.15
		5	0.15

(S1) Order the probabilities highest to lowest (here, it's already like that)

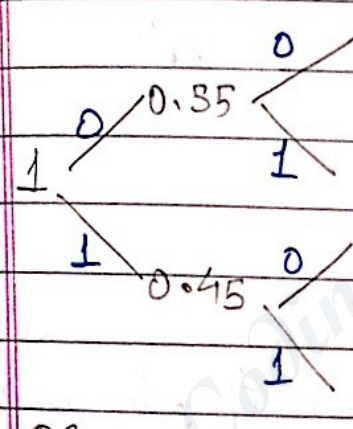
S2) Take bottom-most 2 and add them. Now, rearrange the order

S3) Again take last 2 & do the same.

X	Pi
1	0.25
2	0.25
3	0.2
4	0.15
5	0.15



(S4) Method 1) Now, redraw the above:-



0.15 = 000 X=4

0.15 = 001 X=5

0.25 = 01 X=1

0.25 = 10 X=2

0.2 = 11 X=3

OR

Method 2) Idea: ① Assign 0 to 0.55 & 1 to 0.45 (0 to bigger, 1 to smaller - Convention)

② 0.55 is made of 0.3 & 0.25

So, assign ⓪ 0 to 0.3

⓪ 1 to 0.25

from 0.55

& 0.45 remains 1.

③ 0.45 is made of 0.2 & 0.25. So, assign 10 to 0.25 & 11 to 0.2

→ 00

Next, (0.3) is made of 0.15 & 0.15. So we have 000 to 0.15 X=4 } Random
& 001 to 0.15 X=5 }

Hence, total code becomes

X	P_i	Code
1	0.25	01
2	0.25	10
3	0.2	11
4	0.15	000
5	0.15	001

S5) Now,

$$L = 0.25 \times 2 + 0.25 \times 2 + 0.2 \times 2 + 0.15 \times 3 + 0.15 \times 3$$

$$\Rightarrow L = 2.3 \text{ bits}$$

Ans

~

✱ Lemma 5.8.1

For any distribⁿ, ∃ optimal instt. code that satisfies properties :-

P1) If $P_j > P_k$, then $l_j \leq l_k$.

P2) 2 longest CWs have same length

P3) 2 longest CWs ~~correspond~~/differ by last bit

✱ Theorem 5.8.1 :
Huffman coding is optimal, i.e, if C^* is Huffman code and C' is any other code,
$$L(C^*) \leq L(C')$$ #

eg Just like we used Hoffman code in previous eg.
Use Shannon's code

X	Pi	CW length = $\lceil \log 1/P_i \rceil$	Code
1	0.25	2	00
2	0.25	2	01
3	0.2	$\lceil 2.32 \rceil = 3$	100
4	0.15	$\lceil 2.32 \rceil = 3$	101
5	0.15	$\lceil 2.32 \rceil = 3$	111

Idea: find CW length. Then, use binary bits to write the code.
ie if X(1) = 00 } → or choose any 2
X(2) = 01 } out of 00, 01, 10, 11

Why for any other variety { Then, for code of 3 bits, take such values s.t it is instantaneous code
So, we cannot choose (00)0, (00)1, (01)0, (01)1
So, choose any 3 out of 100, 101, 110, 111

So here,
$$L_{avg} = 2 \times 0.25 + 2 \times 0.25 + 0.2 \times 3 + 0.15 \times 3 + 0.15 \times 3$$
$$= 2.5 \text{ bits}.$$

So, clearly Shannon requires more BW than Hoffmann.

HW eg If we have

X	P(X)
1	0.9999
2	0.0001

Determine code using (a) Huffman code
(b) Shannon's code

☆ Shannon Fano Elias (SFE) Code

eg ↳ Dyadic ⇒ probabilities will be powers of 2.

	X	P(x)	(S1) F(x) → Cumulative distribution	(S2) $\bar{F}(x)$ → Previous value of P(x) + (current val / 2)
(2^{-2})	1	0.25	→ 0.25	$0 + \frac{0.25}{2} = 0.125$
(2^{-1})	2	0.5	→ 0.5 + 0.25 = 0.75	$0.25 + \frac{0.5}{2} = 0.5$
(2^{-3})	3	0.125	→ 0.125 + 0.75 = 0.875	$0.5 + \frac{0.125}{2} = 0.8125$
(2^{-3})	4	0.125	→ 0.125 + 0.875 = 1	$0.125 + \frac{0.125}{2} = 0.9375$

$\lceil \log(1/P_i) \rceil + 1$

(S4) $\bar{F}(x)$ in Binary	(S3) l(x)	(S5) Code Word
0.001	3	001 (take after decimal)
0.1 ≡ 0.10	2	10
0.1101	4	1101
0.1111	4	1111

0.125 to binary = 0.125 × 2 = 0.25 | 0
0.25 × 2 = 0.5 | 0
0.5 × 2 = 1 | 1

⇒ $(0.125)_{10} = 0.001$

Also,
$L_{av} = 0.25 \times 3 + 0.5 \times 2 + 0.125 \times 4 + 0.125 \times 4$
$= 2.75$

Why do ↑

* $|1\rangle = [\ ]$

* For non dyadic case (not in powers of 2), the conversion to binary will be non terminating. So, we stop, seeing $l(x)$.

Arithmetic Coding

→ a way to encode a set of symbols

→ fully closed, → half open

$$\Phi_0(s) = |b_0, l_0\rangle = |0, 1\rangle \equiv [0 \quad 1)$$

$$\Phi_k(s) = |b_k, l_k\rangle = |b_{k-1} + c(s_k)\, l_{k-1},\ P(s_k)\, l_{k-1}\rangle$$

$\hookrightarrow k = 1, 2, \ldots, N$

eg: Consider a source (numerical set, Ω) which as 4 symbols.

So, $M = 4$

X		
0	$P(0) = 0.2$	
1	$P(1) = 0.5$	
2	$P(2) = 0.2$	
3	$P(3) = 0.1$	

$$P = [\underset{P(0)}{0.2} \quad \underset{P(1)}{0.5} \quad \underset{P(2)}{0.2} \quad \underset{P(3)}{0.1}]$$

$$C = [\underset{C(0)}{0} \quad \underset{C(1)}{0.2} \quad \underset{C(2)}{0.7} \quad \underset{C(3)}{0.9} \quad \underset{C(4)}{1}]$$

Suppose I want to transmit symbols

$S = \{2, 1, 0, 0, 1, 3\}$

$\quad\ s_1\ s_2\ s_3\ s_4\ s_5\ s_6$

& $N = 6$

Now, finding $\Phi_k(s)$

$$\Phi_1(s) = |b_1, l_1\rangle = |b_0 + C(s_1) l_0, P(s_1) \times l_0\rangle$$

$$= |0 + C(2) \times 1, P(2) \times 1\rangle$$

$\because |b_0, l_0\rangle = |0, 1\rangle$

$$= |0 + (0.7)(1), (0.2)(1)\rangle$$

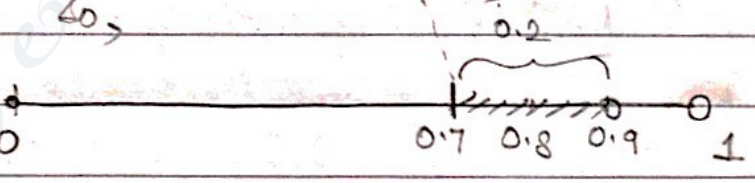
Now, we have only 4 symbols.

$S_i \rightarrow 0, 1, 2, 3$

↳ from set S, ∃ only 4 values 0, 1, 2 & 3

$$l_0, |b_1, l_1\rangle = |0.7, 0.2\rangle$$

starting point → length



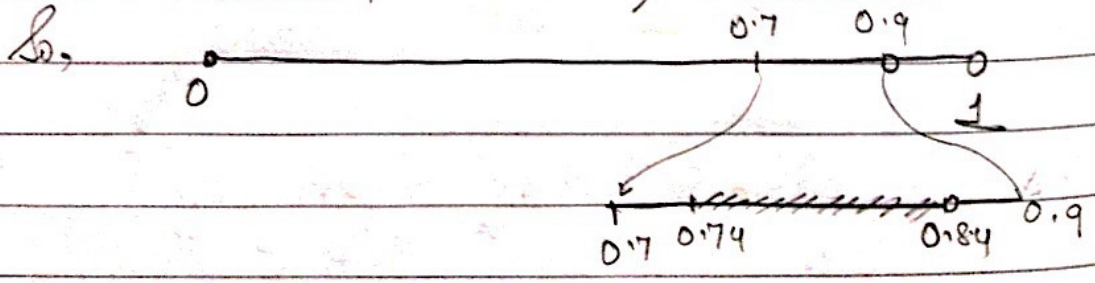
Now,

$$\Phi_2(s) = |b_2, l_2\rangle = |b_1 + C(s_2) \times l_1, P(s_2) \times l_1\rangle$$

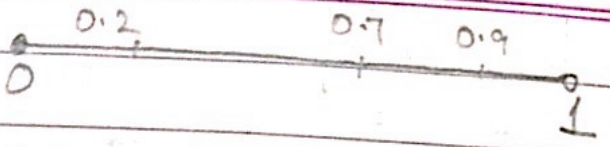
$$= |0.7 + C(1) \times 0.2, P(1) \times 0.2\rangle$$

$$= |0.7 + (0.2)(0.2), 0.5 \times 0.2\rangle$$

$$= |0.74, 0.1\rangle$$



Puffin
Date
Page

Going similarly, 

$\overline{\Phi}_6(s) = |b_6, l_6\rangle = |0.7426, 0.7428\rangle$

So, choose any value in this range
Convert to binary & transmit
Suppose we choose:-
0.7427. It's non terminating in binary. So, we choose a value which has a terminating binary represent^n. That will be

$\hat{v} = (0.74267578125)$

$\hat{v} = (0.1011111000 1)_2$

So, basically, this binary no. represents the sequence $s = \{2, 1, 0, 0, 1, 3\}$. This binary represent^n is unique. & have uniquely coded it

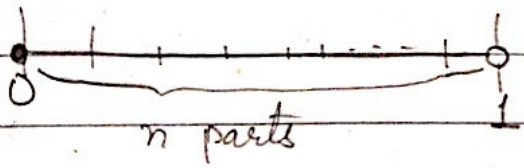
Note: had the symbol sequence was

$s = \{0, 1, 0, 0, 1, 3\}$ → changed,

$|b_1, l_1\rangle = |0, 0.2\rangle$

So, the range (0.7 to 0.9) changes. Hence, code changes in the end

* No. of symbols = no. of parts of the line $|0, 1\rangle$ (segment)



* Note: had P been $P = [0.25, 0.25, 0.25, 0.25]$, we would have got equal 4 segments.

Also,

$$B_{min} = \lceil -\log_2 l_N \rceil = \lceil -\log_2 0.0002 \rceil = 13 \text{ bits}$$

An approximate lower bound on the min. no. of bits that we should have to give the final code = 13.

✡ Arithmetic Decoding

$(0.7426757812)_{10}$

We got the incoming value, $\hat{V} = 0.1011111000_2$

& reconstruct each of the value that was transmitted to me.

$$\hat{S}(\hat{V}) = \{ S_1(\hat{V}), S_2(\hat{V}), \ldots \}$$

Say, estimated value that I got at receiving end

$$\tilde{V}_1 = \hat{V}$$

&

$$\hat{S}_k(\hat{V}) = \{ s : C(s) \leq \tilde{V}_k < C(s+1) \}$$

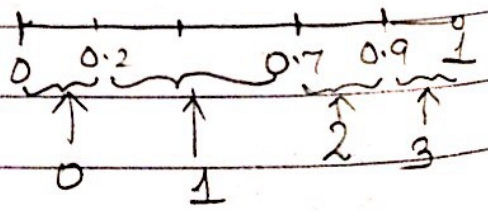
↳ $k = 1, 2, 3, \ldots N$

From coding part, we know

$P(0) = 0.2$
$P(1) = 0.5$
$P(2) = 0.2$
$P(3) = 0.1$

} known both to transmitter & receiver.

Now, if I get $\tilde{V}_1$ in say 0.7 range or something I say, the value is 2.



Now, finding $\tilde{V}_2$, $\tilde{V}_3$:-

$$\tilde{V}_{k+1} = \frac{\tilde{V}_k - c(\hat{S}_k(\hat{V}))}{P(\hat{S}_k(\hat{V}))}, \quad k = 1, 2, 3, \ldots N-1$$

So, $\tilde{V}_2 = \frac{\tilde{V}_1 - c(\hat{S}_1(\hat{V}))}{P(\hat{S}_1(\hat{V}))}$ $\quad c = [0 \;\; 0.2 \;\; 0.7 \;\; 0.9]$

$= \frac{\tilde{V}_1 - c(2)}{P(2)}$ ($c(2) \to 0.7$, $P(2) \to 0.2$)

$\Rightarrow \tilde{V}_2 = \frac{\tilde{V}_1 - 0.7}{0.2}$ ✓

Again, I got $\tilde{V}_1$ around 0.74 - - -

So, $\tilde{V}_2 = \frac{0.74 - 0.7}{0.2}$ So, around,

$\tilde{V}_2 = 0.21337 > 0.2$

So, $\tilde{V}_2$ fits in 1 (0.2 to 0.7 range)

So, till now, I have :-

$\tilde{V}_1 \to (2) \to \hat{S}_1(\hat{V})$ & $\tilde{V}_2 \to (1) \to \hat{S}_2(\hat{V})$

Ny, we can do for $\tilde{V}_3, \tilde{V}_4, \ldots$

We can get the complete sequence like this

The only problem is, I don't know where to stop.

★ how is above better ?

In above 11 bits are req^d to transfer the sequence.

If Hoffman code was used, say we had $0 : 00, 1 = 01, 2 = 10, 3 = 11$

Then, the sequence 2, 1, 0, 0, 1, 3 will take $\underbrace{10, 01, 00, 00, 01, 11}_{2+2+2+2+2+2} = 12$ bits

→ UNIVERSAL Code

Puffin Date Page

★ LEMPEL ZIV WELCH Algorithm

- mainly used for compression of text.

- # = 00000 = 0 (5 bits)
  A = 00001 (= 1)
  B = 00010 (= 2)
  C = 00011 (= 3)
  ⋮
  Z = 11010 (= 26)

We represent alphabets in binary no. where 0 stands for a full stop.

eg: Say I have to transmit "ing"
It needs 5+5+5 bits = 15 bits.
Then, how to reduce no. of bits?
That is done using Lempel Ziv.

Note: We dont take spaces b/w words

eg If I want to transmit MYNAMEISKHAN
Idea: I am using 5 bits. But, alphabets only till 26. So, from 27 to 31 (5 bits: 0 to 31) I can put an "extended dictionary".
eg: I put MY in the no. 27
YN in 28 So, "." represent"
NA = 29
⋮
KH = 35 → When I exceed 31, I use 6 bits now. to go from 31 to 127
HA = 36
AN = 37

eg: If my sequence is MYNMY

Current Sequence	Next Char.	o/p	Extended dictionary
NULL	M	—	—
M	Y	13=01101	MY=27
Y	N	25=11001	YN=28
N	M	14=01110	NM=29
M	Y.	27	

↳ Now, I see if MY is in my dictionary. It is there. So, I stop here.

Send stop code = 000000

So, If I generally sent:

MYNMY = 5+5+5+5+5 bits = 25 bits

Now,

from o/p column, I have 13, 25, 24, 27

So, its 5+5+5+5 = 20 bits

↳ no. reduced.

* Decoding of LZW

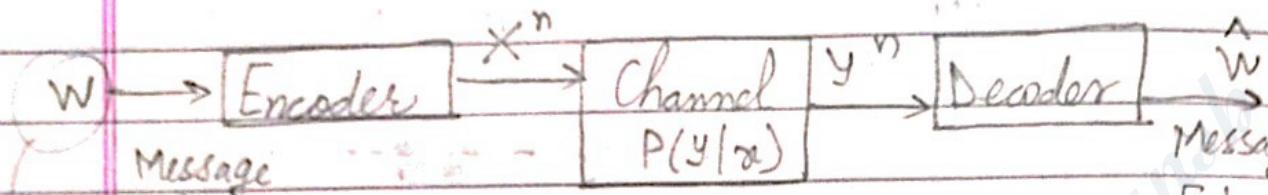
i/p	o/p sequence	New Dictionary	Entry Conjecture
01101	M	—	M?
11001	Y	MY=27	Y?
01110	N	YN=28	N?
00001	A	'	A?
01101	;	'	M?
00101	E	ME=31	32: E? (6 bit codes)
001001			

Idea :- I keep taking in 5 bit collections & detecting. I also keep making dictionary. As soon as it its 31 → 6 bits

PART

✱ CHANNEL ENCODING

(B)



Source encoded message (assumed)

Assume:
Discrete memoryless channel (DMC)

✱ Channel capacity, $C = \max_{P(x)} I(X;Y)$

§ • Noiseless Binary Channel

i.e, get the same as transmitted : $0 \longrightarrow 0$
$1 \longrightarrow 1$

↳ i.e
$P(y=0|x=0)=1$
$P(y=1|x=1)=1$

For this channel,

$C = \max_{P(x)} I(X;Y)$

$\Rightarrow C = \max \{ I(X;Y) \}$
∀ combinations of $P(x)$ or
Prob.$(X=0)$, Prob.$(X=1)$

We know

$$I(X;Y) = \sum_y \sum_x P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

$$= \sum_y \sum_x P(y|x)\,P(x) \log \frac{P(y|x)P(x)}{P(y)P(x)}$$

$$= \sum_y \sum_x P(y|x)P(x) \log \frac{P(y|x)}{P(y)}$$

Now, $P(y|x)$ = 1 (Noiseless Binary Channel)

$$\Rightarrow I(X;Y) = \sum_{y=0}^{1} \sum_{x=0}^{1} P(x) \log \frac{1}{P(y)}$$

Assume: $P(X=0) = p_0$ & $P(X=1) = 1-p_0$

Now,

$P(Y=0) = P(y=0|X=0) \cdot P(X=0)$

(∵ $y=0$ can be got only when $X=0$)

So, $P(Y=0) = p$ & $P(Y=1) = 1-p$

Now, $I(X;Y) =$

$$-\Big[\underbrace{p_0 \log p_0}_{X=0,Y=0} + \underbrace{p_0 \log(1-p_0)}_{X=0,Y=1} + \underbrace{(1-p_0)\log p_0}_{X=1,Y=0} + \underbrace{(1-p_0)\log(1-p_0)}_{X=1,Y=1}\Big]$$

These combinations can't happen ($P(Y=0|X=1)=0$ & $P(Y=1|X=0)=0$)

$$I(X;Y) = -[\log p_0]$$

Now, for $C = \max_{P(x)} I(X;Y)$, I am concerned with max. value of $I(X;Y)$

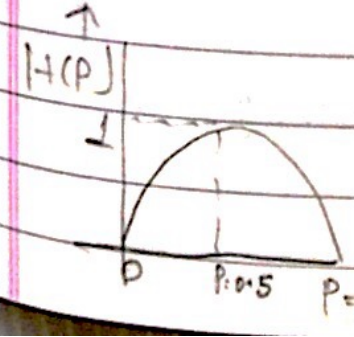
So, finding $I(X;Y)\Big]_{max}$: $\frac{d}{dP_0} = 0$ & find P_0

we have $C\left(P_0 = \frac{1}{2}\right) = -\left[\log \frac{1}{2}\right]$

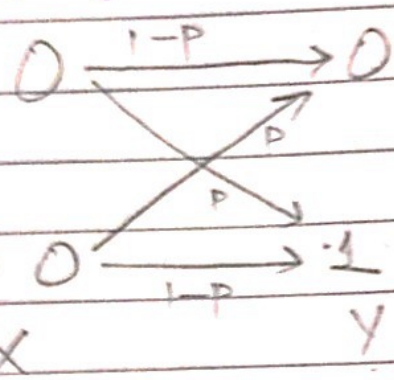
$$= -\log(2^{-1})$$

$$= \log_2 2$$

$\Rightarrow C(P_0 = 1/2) = 2^{\circ}$. = max. capacity



§. Binary Symmetric Channel



$\Rightarrow P(y=0|x=0) = 1-p$
$P(y=1|x=1) = p$
$P(y=0|x=1) = p$
$P(y=1|x=0) = 1-p$

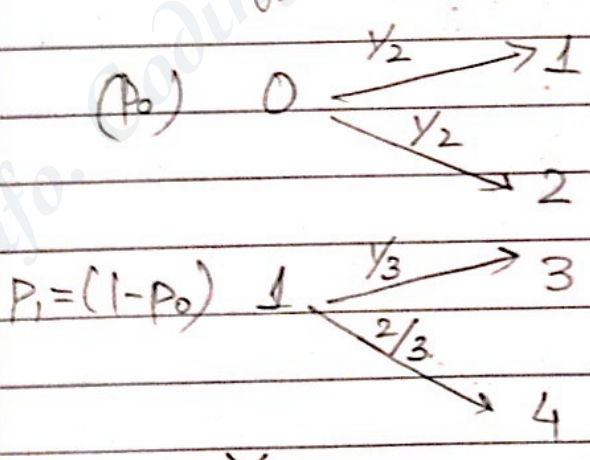
We saw,

$$I(X,Y) = H(Y) - H(P)$$
$$\leq 1 - H(P)$$
$$\Rightarrow \boxed{C = 1 - H(P) \text{ bits}}$$

C → channel capacity

§. Noisy Channel with Non-Overlapping o/p's.

i.e, for each possibility 0 & 1, they have diff^t o/p's.



$\Rightarrow P(y=1|x=0) = 1/2$
$P(y=2|x=0) = 1/2$
$\star P(y=3|x=0) = 0$
$\star P(y=4|x=0) = 0$

Ily for $x=1$.

We have

$$I(X,Y) = \sum_{x=0}^{1} \sum_{y=1}^{4} P(y|x)\,P(x) \log \frac{P(y|x)}{P(y)}$$

Puffin
Date
Page

Now, finding $P(Y=1)$, $P(Y=2)$, $P(Y=3)$, $P(Y=4)$:

$P(Y=1|X=0)P(X=0)$
$+ P(Y=1|X=1)P(X=1)$
$= (1/2)(P_0) + 0(0)$
$= P_0/2$

$P(Y=2|X=0)P(X=0)$
$+ P(Y=2|X=1)P(X=1)$
$= \frac{1}{2}(P_0) + 0(0)$
$= P_0/2$

$P(Y=3|X=0)P(X=0)$
$+ P(Y=3|X=1)P(X=1)$
$= 0(0) + (\frac{1}{3})(1-P_0)$
$= \frac{1-P_0}{3}$

$P(Y=4|X=0)P(X=0)$
$+$
$P(Y=4|X=1)P(X=1)$
$= 0(0) + (1-P_0)\times\frac{2}{3}$
$= \frac{(1-P_0)2}{3}$

So, $P(Y=1) = P_0/2$
$P(Y=2) = P_0/2$
$P(Y=3) = (1-P_0)/3$
$P(Y=4) = (1-P_0)\times 2/3$

So,

$$I(X;Y) = \left(\frac{1}{2}\right)(P_0)\log\frac{(1/2)}{(P_0/2)} + \left(\frac{1}{2}\right)(P_0)\log\frac{(1/2)}{(P_0/2)} + \left(\frac{1}{3}\right)(P_1)\log\left(\frac{1/3}{(1-P_0)/3}\right) + \frac{2}{3}P_1\log\left(\frac{2/3}{(P_1\times 2/3)}\right)$$

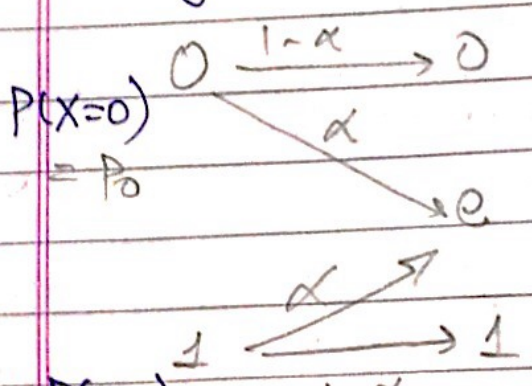
($P(Y=1|X=0)$, $P(X=0)$, $P(Y=1|X=0)$, $P(Y=1)$)

$$\Rightarrow I(X;Y) = P_0\log\frac{1}{P_0} + (1-P_0)\log\frac{1}{(1-P_0)}$$

$$\Rightarrow I(X;Y) = H(P)$$

★ SELF : NOISY TYPEWRITER

☆ Binary Erasare Channel.



$$C = \max_{P(x)} I(X;Y)$$

$$\Rightarrow C = \max_{P(x)} \left(H(Y) - H(Y|X) \right)$$

$$\sum_x \sum_y p(x,y) \log P(y|x)$$

$$= \sum_x \sum_y p(y|x) p(x) \log p(y|x)$$

$$= H(\alpha), \text{ say}$$

Now,

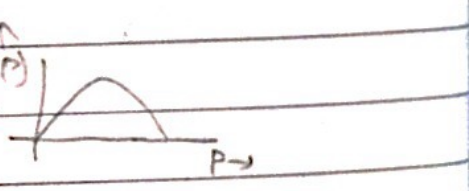
$$H(Y) = P(y=0) \log P(y=0) + P(y=e) \log P(y=e) + P(y=1) \log P(y=1)$$

$(x = 0, 1 \quad y = 0, 1, e)$

$P(y=0|x=0) = 1-\alpha$
$P(y=e|x=0) = \alpha$
$P(y=1|x=0) = 0$
$P(y=0|x=1) = 0$
$P(y=e|x=1) = \alpha$
$P(y=1|x=1) = 1-\alpha$

$$P(y=0) = P(y=0|x=0) P(x=0) + P(y=0|x=1) P(x=1)$$
$$= (1-\alpha) P(x=0) + 0 \, P(x=1)$$

Why, $P(Y=1), P(Y=e)$



After solving;

$$H(Y) = H(\alpha) + (1-\alpha) H(P)$$

$$\text{So, } C = \max_{P(x)} \left((1-\alpha) H(P) + H(\alpha) - H(\alpha) \right)$$

$$\Rightarrow C = \max_{P(x)} (1-\alpha) H(P) \rightarrow 1, \text{ max value}$$

$$\Rightarrow C = 1 - \alpha = \text{channel capacity.}$$

↳ transn probability for channel.

☆ Symmetric channels :

eg

$$P(Y|X) = \begin{array}{|ccc|l} y=0 & y=1 & y=2 & \\ \hline 0.3 & 0.2 & 0.5 & x=0 \\ 0.5 & 0.3 & 0.2 & x=1 \\ 0.2 & 0.5 & 0.3 & x=2 \end{array}$$

$\Rightarrow P(Y=1 \mid X=1) = 0.3$

$P(Y=2 \mid X=0) = 0.5$

→ Seeing the matrix, there is a symmetry. Only 3 values (0.2, 0.3, 0.5). Just that arranged like :

a	b	c
c	a	b
b	c	a

* $Y = X + Z \pmod c$

Now,

$I(X;Y) = H(Y) - H(Y|X)$

$= H(Y) - H(A)$

$H(Y|X) \rightarrow \sum_x \sum_y P(x,y) \log P(y|x)$

$= P(y|x)\,P(x)$

$H(Y)$ can be bounded by $\log|y|$

$H(0.3, 0.2, 0.5)$

→ i.e taking any row (1st row)

$= -[0.3 \log(0.3) + 0.2 \log(0.2) + 0.5 \log(0.5)]$

$\leq \log|y| - H(A)$

→ Possible value of y

$\Rightarrow I(X;Y) \leq \log 3 - H(0.3, 0.2, 0.5)$

→ ∵ y can take values 0, 1, 2

So, channel capacity,

$C = \log_2 3 - H(0.3, 0.2, 0.5)$

✡ Theorem 8.2.1

Weakly symmetric channel :-

eg :- $P(y|x) = \begin{pmatrix} 1/3 & 1/6 & 1/2 \\ 1/3 & 1/2 & 1/6 \end{pmatrix}$ } rows are permut^{n} of each other (not columns)

$\underbrace{\frac{2}{3} \quad \frac{2}{3} \quad \frac{2}{3}}$ → Sum of all columns is same.

for this,

$$C = \log|y| - H(\text{row of trans}^{n} \text{ matrix})$$

↳ Achieved by uniform distrib^{n} of $p(x)$

✡ Properties

P1) $C \geq 0$

P2) $C \leq \log|x|$ } ⇒ C (Channel capacity) is finite.

P3) $C \leq \log|y|$

P4) $I(X;Y)$ = continuous f^{n} of $p(x)$

P5) $I(X;Y)$ = concave f^{n} of $p(x)$ (Theorem # 2.6.2 ch-2)

↳ ⇒ local maxima is also global maxima

Puffin
Date 6/3/14
Page

✱ Theorem 3.1.1 (AEP: Asymptotic Equipartition Property)

(independent & identically distributed)

If $X_1, X_2, \ldots, X_n$ are $i.i.d \sim p(x)$

then,

$$-\frac{1}{n} \log p(X_1, X_2, \ldots X_n) \longrightarrow H(X) \text{ in prob.}$$

(converging to)

✱ Defin^n: Typical Set: $A_\epsilon^{(n)}$ w.r.t $p(x)$ is the set of sequences $x_1, x_2 \ldots \ldots \in X^n$ with the following property:

$$2^{-n(H(X)+\epsilon)} \leq p(x_1, x_2, \ldots x_n) \leq 2^{-n(H(X)-\epsilon)}$$

↳ ϵ : parameter

↳ n : sequence length

⇒ joint probability mass fn has a Bound

Idea: get n as large as possible, and, find bound on probability mass fn

✱ Theorem 3.1.2:

From above:-

① $H(X) - \epsilon \leq -\frac{1}{n} \log\{P(x_1, x_2 \ldots x_n) \leq H(X) + \epsilon$

② $P\{A_\epsilon^{(n)}\} \geq 1 - \epsilon$ for sufficiently large n

③ $|A_\epsilon^{(n)}| \leq 2^{n(H(X)+\epsilon)}$

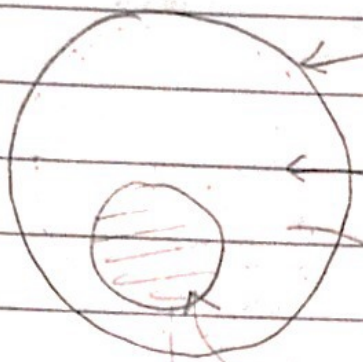
↳ $|A_\epsilon^{(n)}|$ = no. of elements in set $A_\epsilon^{(n)}$

↳ no. of elements have an upper bound

④ $|A_\epsilon^{(n)}| \geq (1-\epsilon)\, 2^{n(H(X)+\epsilon)}$; for sufficient large n.

Fig.

Seeing no. of bits in each CW



★ Theorem 3.2.1 :

If X^n = i.i.d $\sim p(x)$.

Let $\epsilon > 0$. Then, $\exists$ a code which maps sequence of length n to binary strings $\Rightarrow$ mapping is ONE-ONE

Also,

~~$\left(E\left[\frac{1}{n}\left[X^n\right]\right]\right)$~~

$$E\left[\frac{1}{n}\left|(X^n)\right|\right] \leq H(X)+\epsilon$$

↳ for large n, $X^n \rightarrow nH(X)$

(discussed on next page)

Date 9/3/14
Page

✱ Theorem 3.2.1

$$E\left[\frac{1}{n} l(X^n)\right] \leq H(X) + \epsilon$$

↳ for sufficiently large n, (n → no. of elements in sequence of symbols)

$X^n \longrightarrow n H(X)$ bits on avg.

Consider a sequence of symbols

$S = \{1, 3, 2, 0\}$

Here, ∃ 4 symbols ⇒ $n = 4$

✱ Theorem 5.4.2

$$\frac{H(X_1, \ldots, X_n)}{n} \leq L_n^* < \frac{H(X_1, \ldots X_n)}{n} + \frac{1}{n}$$

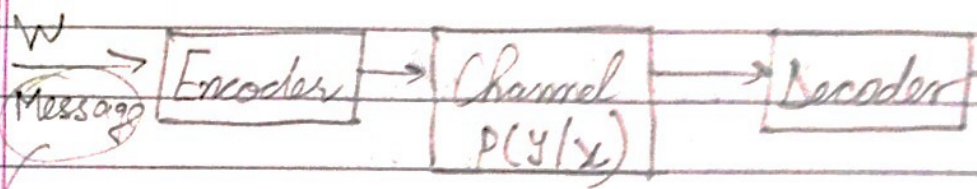
↳ If $n \to \infty$; $X_1, X_2, \ldots, X_n$: independent

Just to give idea

$L_n^* \longrightarrow H(X)$ $\quad (H(X) \leq L_n^* < H(X))$

(L_n^* → avg. length)

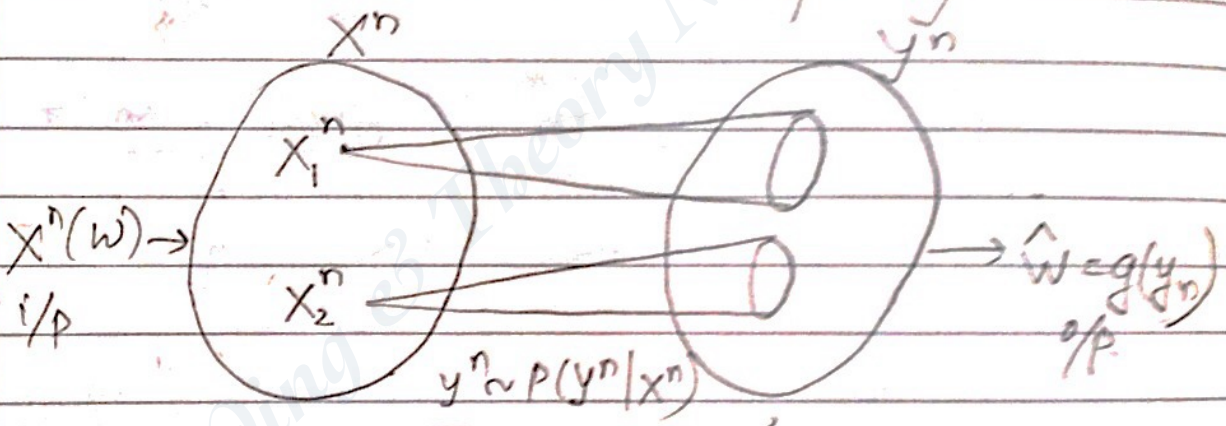
Chapter - 8.



= {0, 1, 2, 0.0} , say.

Ŵ → estimate of message.

→ Assuming source encoding has already happened. So, using channel separately



the way encoding is happening

★ $P(Y_k | X^k, Y^{k-1}) = P(Y_k | X_{xk})$

↳ $k = 1, 2, \ldots, n$.

↳ ⇒ Y_k is independent of Y^{k-1}

↳ Its a discrete memoryless channel.

↳ present o/p doesn't depend on past o/p.

☆ Channel without Feedback

$$P(y^n | x^n) = \prod_{i=1}^{n} P(y_i | x_i)$$

→ X: Transmitted, Y: Received

→ $y^n = y_1, y_2, \ldots, y_n$

→ $x^n = x_1, x_2, \ldots, x_n$.

☆☆ (M, n) code for channel $(X, P(y|x), Y)$

n → sequence length

$(X, P(y|x), Y)$ → Suppose we define a channel like that:

X: i/p

P(y|x): cond^n

Y: o/p.

(M, n) code is a collection of code words.

① An index set $\{1, 2, \ldots, M\}$ → for M = 4, say, we have 4 code words 00, 01, 10, 11 (say)

② An encoding fn:

$X^n : \{1, 2, \ldots, M\} \rightarrow X^n$.

So, it is like:

$1 \longrightarrow X^n(1)$

$2 \longrightarrow X^n(2)$

$\vdots$

$M \longrightarrow X^n(M)$

③ A decoding fn.

$g : y^n \rightarrow \{1, 2, \ldots, M\}$

$\Rightarrow g(y^n(1)) \rightarrow 1$

$g(y^n(2)) \rightarrow 2$ . . .

Puffin Date: Page:

④ Probability of error:

$$\lambda_i = P\left[\, g(y^n) \neq i \;\middle|\; X^n = X^n(i) \right]$$

λ_i → Probability of error

↳ Suppose I transmit "i".
So, after encoding, its $X^n(i)$
after decoding its $g(y^n(i))$
If $g(y^n(i)) \neq i$, then, there is error.

* Maximal probability of error for (M, n) code

$$\lambda^{(n)} = \max_{i \in \{1,2,\ldots,M\}} \lambda_i$$

* Average probability of error (P_e)

$$P_e^{(n)} = \frac{1}{M} \sum_{i=1}^{M} \lambda_i$$

* Rate (R) of (M, n) code:

$$R = \frac{\log M}{n} \text{ bits per transmission.}$$

↳ Rate, R is achievable if ∃ a sequence of

$(\lceil 2^{nR} \rceil, n)$ codes s.t $\lambda^{(n)} \to 0$ as $n \to \infty$.

Puffin
Date
Page

* Capacity (C)

$$C = \sup_{(\max)} R$$

↓
Supremum

* Typical Sequences:

X^n, Y^n: Joint typical sequences (Theorem 8.6.1)

* Theorem 8.7.1 : Channel Coding Theorem.

All rates, $R < C$ (Capacity) are achievable

⇓

$\exists$ sequence $(2^{nR}, n)$ of codes for $\lambda^{(n)} \to 0$ as $n \to \infty$

• Converse to theorem 8.7.1

Any sequence of $(2^{nR}, n)$ codes with $\lambda^{(n)} \to 0$ must have $R \leq C$.

* Lemma 8.9.1: (Applies to discrete memoryless channel (DMC))

$$H(X^n|Y^n) < 1 + P_e^{(n)} nR$$

→ avg. prob. of error.

* Lemma 8.9.2 (for DMC)

$$I(X^n; Y^n) \leq nC \quad \forall \; P(X_n)$$

→ capacity of channel

i.e., we can't increase channel capacity by using channel again and again.

Puffin
Date
Page

* Consider

$X_k \longrightarrow$ [Channel] $\longrightarrow Y_k$ (statistically independent o/p)

$X_k \longrightarrow (+) \longrightarrow Y_k$; $k = 1, 2, \ldots, K$

N_k (Noise)

$\Rightarrow Y_k = X_k + N_k$,

$N_k \longrightarrow$ zero mean.

$\sigma^2 = N_0 W$

σ^2 : Variance; N_0 : Noise spectral density; W : Band limited channels.

Assumption: Transmitter is power limited,

i.e., $E[X_k^2] = P$; $k = 1, 2, \ldots, K$

* $C = \max_{f_{X_k}(x)} \left[I(X; Y) \,\middle|\, E[X^2] = P \right]$

C : Channel capacity

Now, $I(X_k; Y_k) = H(Y_k) - H(Y_k|X_k)$

$= H(Y_k) - H(N_k)$

↳ Max $I(X_k; Y_k)$ is possible if $H(Y_k)$ is max. This is possible if Y_k is Gaussian.

$\Rightarrow$ Channel is Gaussian.

$$H(Y_k) = \frac{1}{2} \log_2 [2\pi e (P + N_0 W)]$$

$$H(N_k) = \frac{1}{2} \log_2 [2\pi e (N_0 W)]$$

$$\text{i.e., } I(X_k; Y_k)\Big|_{\max} = \frac{1}{2} \log \left(\frac{P + N_0 W}{N_0 W} \right) = C,$$

$$\Rightarrow \overbrace{I(X_k; Y_k)}^{C} = \frac{1}{2} \log \left(1 + \frac{P}{N_0 W} \right) \text{ bits per use.}$$

$$\Rightarrow C = W \log_2\left[1 + \frac{P}{N_0 W}\right] \text{ bits per second}$$

* W → BW ($\simeq 4$ kHz usually)
* (Comes in Shannon's theorem)

We know, $R = \frac{\log_2 M}{n} = \frac{k}{n}$

$$\Rightarrow R = \frac{1}{2}\log_2\left(1 + \frac{P}{\sigma^2}\right) \text{ bits per use}$$

R → Coding rate

- P: avg. power per bit.
- σ^2: noise variance assuming Gaussian channel.

* MIMO SYSTEMS

↳ Multiple I/p Multiple O/p systems

$$y[k] = \sqrt{\frac{E}{M_T}}\, H\, S[k] + n[k]$$

- $y[k]$ → received signal
- M_T → Dimension of vector size.
- $S[k]$ → Transmitted signal
- $n[k]$ → Channel noise

For such a sys, $C = WM \log_2\left(1 + \frac{E_s}{N_0}\right)$ bps

↳ $M \uparrow \Rightarrow$ MIMO

$W \uparrow\uparrow$, $M \downarrow \Rightarrow$ SISO.

eg CDMA sys

eg Suppose I want to transmit "1". There can be errors in reception, So,

Transmitter → Receiver

1 → 0

Now, if we transmit 5 1's

11111 ⇝ R = 2

$R = \log_2 \frac{M}{n} = \frac{1}{5}$

look at 5 bits. If majority is 1, we say 1 is transmitted.

$\left\{\begin{matrix} 3 \cdot 1^5 \\ 4 \times 1^5 \\ 5 \times 1^5 \end{matrix}\right\}$ → "1" was transmitted

eg Suppose I have even parity & I want to transmit Tx = 1101 (odd no. of 1's). So, take another message & combine it with 1101

So,

1101|1101

→ Takes 5 bits keeping LSB = 5 bits

* Error detection

↳ Even & Odd parity sys tells about the ~~which~~ & error (doesn't tell which bit)

* Error correction

* Consider a code consisting of 2 code words.

$C = [0100, 1111]$

* Hamming weight (w)
No. of 1's in any code word.

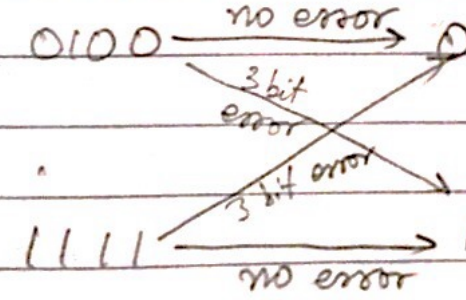
* Hamming distance (d)
How many bit difference is there in any CW PAIR.

$w(0100) = 1$ $\quad d(0100, 1111) = 3$
$w(1111) = 4.$

* In typical channel communi^{ns}, probability of

1 bit error > 2 bit error > 3 bit error - - - -

* What is error?
I was sending some CW (say 0100) and I get a diff^{t} CW (say 1111)
So, in that case, ∃ 3 bit error.



eg: Consider a code book,
$C = [00000, 10100, 11110, 11001]$

Puffin
Date
Page

Block length : (n)

- No. of bits in a CW.
- Block code is called of a fixed size, if all the CWs have same block length.

For

$C = \{00000, 10100, 11110, 11001\}$

Block length (n) = 5

↳ Fixed size CWs

Idea : Increase hamming distance ⇒ reduce chances of bit errors

eg. If I want to transmit 00, 01, 10, 11

$d = 1$

Suppose I choose 5 bits as :

	Uncoded	Coded
Total CWs (M) = 4	00	00000
	01	10100
	10	11110
	11	11001
	$d = 1$	$d = 2$ → Increased, so, better.

So, $K = \log_2 M$

$= \log_2 4$

$= 2$

$n = 5$.

So, Rate, $R = \frac{K}{n} = \frac{2}{5} = 0.4$

Let say a sequence of - 10, 01, 01, 00 ... is being sent

This is the CW: 11110 10100 10100 00000...
which will be sent

So, for this, we write is (n, k) code.

here, its (5, 2) code

We also specify a codebook in the way (n, k, d')
↳ d': distance b/w CW, (Hamming)
So, here, (5, 2, 2) code.

min distance

Note: our choice of CW, can vary.
We can choose, say
00000, 00111, 11011, 11100.
↳ $d_{min} = 3$

w': min. weight for a ~~CW~~ Code Book
So, w (C) = 2

* Linear Code: → A MODULO 2 ADDITION, i.e, add binary bits, but dont take carry

Properties: P1. Sum of two CW, is a CW.
P2. All zero CW is also in code book.
P3. Min. distance (d') = Min. weight (w')
↳ Why? ∵ Implementation of d' is difficult. So, If d' = w', So, to get d', just find w'.

eg: If C = (0000 0, 10100, 11110, 11001)
Check if it's a linear code.

(P1) 00000 + 10100 = 10100 ✓
10100 + 11110 = 01010 ✗ → not in code book

So, not a linear code

* Concept of Field and Rings.

Consider a field (F) with 2 opers, multiplicⁿ & addition
Then: P1) $a, b \in F \Rightarrow a+b \in F$ & $a \cdot b \in F$

P2) $\forall\ a, b, c \in F$
(a) Commutative property
$a+b = b+a$
$a \cdot b = b \cdot a$
(b) Associative property
$a+(b+c) = (a+b)+c$
$a \cdot (b \cdot c) = (a \cdot b) \cdot c$
(c) Distributive property
$a \cdot (b+c) = (a \cdot b) + (a \cdot c)$

P3) $\exists\ 0, 1 \in F$
s.t (a) $a+0 = a$
(b) $a \cdot 1 = a$

P4) $a + (-a) = 0$

P5) $a \cdot (a^{-1}) = 1 \quad (a \neq 0)$

> If all properties P1 to P5 are satisfied :
called GALOUIS field (GF)
(considering finite no. of elements only)
> If P1 to P4 are satisfied :
called GALOUIS Ring (GR)

eg: Consider a set $S = \{1100, 0100, 0011\}$
It is a $GF(2^4)$
→ size of vectors = 4
→ 2 : binary system.

Now, Generating code book from set.

$C = \langle S \rangle$
↓
span of S

What to do ?
(i) All zero CW has to be added to code book
(ii) All words in S, put in C
(iii) All linear combin^{ns} of 2 or more CWs have to be added (modulo-2) with each other & incorporated in code book.

So, (iii) :

1100 + 0100 = 1000
1100 + 0011 = 1111
0100 + 0011 = 0111
1100 + 0100 + 0011 = 1011

So, code book becomes

$C = \{0000, 1100, 0100, 0011, 1000, 1111, 0111, 1011\}$

→ $d^* = w^* = 1$

* Consider a code book of a nature : $(46, 2^4)$ (n, k)

So, No. of CWs $= 2^k = 2^{24}$

Idea : (M1) To get cw : Store in Lookup table

Uncoded bits $\longleftrightarrow$ CWs

Lookup table → mapping stored here.

Size of lookup table $= n \times 2^k = 46 \times 2^{24}$

$= 771,751,936$ bits

So, here, ∃ problems with storage space & search delay

(M2) Generate CWs instantaneously

$C = i \times G$ → given G : Generator matrix

Storage $= 46 \times 24 = 1104$ bits

Problems : Computation done instantaneously

eg : $G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}_{2\times3}$ → (3×2) code; columns, rows

$C_1 = [0\,0] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = [0\ 0\ 0]$

$C_2 = [0\,1] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = [0\ 1\ 0]$

$C_3 = [1\,0] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = [1\ 0\ 1]$

$C_4 = [1\,1] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = [1\ 1\ 1]$

So, CWs are C_1, C_2, C_3 & C_4

$\Rightarrow C = \{000, 010, 101, 111\}$

$\hookrightarrow d' = 1$, (1) → excluding (000)

✡ Systematic form for Generator Matrix (G)

eg: $G = \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right] = [I \mid P]$

→ Partition
→ Identity

↳ can be divided into 2 parts: Identity matrix & Partition matrix

eg (2) :-

$$G = \begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 2 & 2 & 1 \end{bmatrix} \rightarrow \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2 \end{array}\right]$$

$I \quad | \quad P$

↳ can be done using row & col operⁿ

$$G = [I \mid P]$$

We can get H matrix from G matrix,

Parity check, $H = [-P^T \mid I]$ matrix

s.t,

$$GH^T = 0.$$ (just as done in Communicⁿ systems)

eg: $(\underset{n}{7}, \underset{k}{4})$ linear block code

$$G = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 1\ 0\ 1 \\ 0 & 1 & 0 & 0 & 1\ 1\ 1 \\ 0 & 0 & 1 & 0 & 0\ 1\ 0 \\ 0 & 0 & 0 & 1 & 0\ 1\ 0 \end{array}\right]$$

$I_{4\times 4}$ | P

Now:

$$P^T = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Self: we can find

$$-P^T = P^T$$

& $-1 = -1$ in MODULO 2 binary

$$\rightarrow H = \left[\begin{array}{cccc|ccc} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

$-P^T$ | $I_{3\times 3}$

: Systematic form.

eg (4): Consider:

$$H = \left[\begin{array}{cccc|ccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right]$$

not identity matrix

→ So, not in Systematic form

eg (5) Consider 4 bit sequence of codes

0000, 0001, 0010, - - - - , 1111

So, generating CWs:

$C_1 = iG$

$$= [0000] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

$\Rightarrow C_1 = [0000000]$

So, we have $C_1 = [0000]\ [G] = [0000 \mid 000]$

$C_2 = [0001]\ [G] = [0001 \mid 010]$

$C_3 = [0010]\ [G] = [0010 \mid 010]$

Some sort of parity bits attached to my actual codes.

Now, consider a non systematic G, then we won't get parity bits.

Now, consider $H(\underline{r}) = HC_2 = \begin{bmatrix} 1100 & 100 \\ 0111 & 010 \\ 1100 & 001 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$

r → received vector

do Modulo-2 Add^n (1+1 = 0, no carry taken)

$\Rightarrow HC_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \overline{0}$

If we don't get zero vector $\Rightarrow \exists$ error in transmission.

Similarly, we can do for $HC_1, HC_3 \ldots$

Suppose $HC = [H] \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ → Its not zero vector

So, $\exists$ error

Now $HC = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ is there in H.

$$H = \left[\begin{array}{ccc|ccc} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

→ HC matches first column. So, I had error in 1st bit of C.

(I took $C = \textcircled{1}0\,0\,1\,0\,1\,0$

It should be, $C = \textcircled{0}0\,0\,1\,0\,1\,0$)

Imp.

★ Note: We might have multiple column matching if H has identical columns (some).

So, for unique error detection, H should have all non-identical columns.

Observation

① ★ Consider a set of CW's as:

$C = \{000, 001, 010, 011, 100, 101, 110, 111\}$

↳ $d' = 1$

& for say 000, 1 bit error can give

100 010 001

So, ∵ these 3 code words are already present in code book, I CAN't Detect error.

② ★ Consider now CW's,

$C = \{000, 011, 110, 101\}$

↳ $d' = 2$, now.

Now, if for 000, ∃ 1 bit error

100 010 001

These 3 are not there in my set now. So, I get to know that error has occured. (Error detection ✓)

Next, if ∃ 2 bit errors taken into account for 000 (say)

→ 110, 101, 011.

These are present in my Code Book. So, I cant detect 2 bit error.

③ let $C = \{000, 111\}$

↳ $d' = 3$

So, ✓now single bit and double bit errors can be detected

✓ 1-bit errors can be corrected

- So, in general, if min. distance = d', we can detect upto $(d'-1)$ bit errors. we can correct upto $(d'-2)$ bit errors.

- let $G = \left[\begin{array}{c|c} 1000 & 101 \\ 0100 & 111 \\ 0010 & 010 \\ 0001 & 010 \end{array}\right]$ & $H = \left[\begin{array}{c|c} 1100 & 100 \\ 0111 & 010 \\ 1100 & 001 \end{array}\right]$

So, codes for each CW are (PTO)

CW. ⟶ Code for CW.

[0000] ⟶ [0000000]
[0001] ⟶ [0001010]
[0010] ⟶ [0010 0101]
[0011] ⟶ [0011 0007].
↳ $d' = 2$, so far.

⋮

So, obviously $d' \leq 2$ (if all the codes are seen)

So, from previous note,
we can detect $d'-1 = 1$ bit error
& we can correct $d'-2 = 0$
↳ So, we can NOT correct errors

* For correction capability, $d' = d_{min} = 3$.

eg Consider a systematic matrix G

$$G = \overset{\overset{I}{\downarrow}}{\left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]}$$

k rows
n columns.

We can find parity check matrix

$$H = \overset{\overset{I}{\downarrow}}{\left[\begin{array}{cccc|ccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right]}$$

(n-k) rows
(n columns)

So, $C = iG$

Hence, Codes are

[0000] → [0 0 0 0 0 0 0]

[0001] → [0 0 0 1 0 1 1]

[0010] → [0 0 1 0 1 1 0]

doing entirely, we get $d' = 3$

So, we can detect 2 bit errors & correct 1 bit errors.

With no error

If $\underline{x}$ is a CW

Then, $H\underline{x} = \underline{0}$ (parity check)

Now, we take 1 bit error

Transmission 0 0 0 0 0 0 0

Reception 0 0 0 1 0 0 0

1 bit error occurred

Now, do

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

matching, we find 4th bit was at error

* Note, here, I can't correct 2 bit errors. But, if I detect the 2 bit errors, I can request for retransmission.

Theorem 3.5 Singleton Bound

(Ranjan Bose)

$d^* (= d') \leq n-k+1$

d^* → d_{min}

↳ for (n, k) code.

↳ for (7,4) code,

$d^* \leq 7-4+1$

$\Rightarrow d' \leq 4$

Note: If any CW, we take max. distance (d_{max}) instead of d', we say it as Max. distance code.

For that, $d^* = n-k+1$

☆ COSET VECTORS

If $G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$, say

$C = iG$

So, we find codes for CW's = 00, 01, 10 & 11.

[00] → [000]

[01] → [010]

[10] → [101]

[11] → [111]

} 4

out of 8 possibilities, we have only 4 here

Now, Coset Vectors are the vectors which give me complete 8 combin^ns when combined with these 4.

So, coset of $C = \{000, 001\}$

Puffin
Date
Day

Codes Coset

So, 000 + (000) = 000
000 + 001 = 001
010 + 000 = 010
010 + 001 = 011
101 + 001 = 100
101 + 000 = 101
111 + 001 = 110
111 + 000 = 111

all 8 ✓

* Its modulo-2 addition

$$P_{err} = 1 - \sum_{i=0}^{n} \alpha_i P^i (1-P)^{n-i}$$

P_{err} → Probability of error

- n = size of each CW (length)
- Assume equally probable CW.
- 1 bit error (symbol error) = P.
- Assume Binary Symmetric Channel
- α_i = no. of coset vectors with weight "i"
  - α_0 = no. of coset vectors with weight 0.
  - α_1 = no. of coset vectors with (weight) 1

Hamming weight

for $C = \{000, 001\}$

$\alpha_0 = 1$, $\alpha_1 = 1$

So, we have $P = 0.9$, $P_{err} = 0.981$

eg 2

$$P_{err} = 1 - \sum_{i=0}^{t} \alpha_i P^i (1-P)^{n-i}$$

Given: $n = 3$

$\alpha_0 = \alpha_1 = 1$

$\alpha_2 = 0$

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

$C = \{000, 010, 101, 111\}$

↳ $d' = 1$

Coset $= \{000, 001\}$

$P = 0.1$, $P_{err} = 0.01$

↳ probability of 1 bit error.

Code $C = \{0000, 1011, 0101, 1110\}$.

Coset leaders $= \{\underbrace{0000}_{\alpha_0 = 1}, \underbrace{1000, 0100, 0010}_{\alpha_1 = 3}\}$

$\alpha_2 = \alpha_3 = \alpha_4 = 0$

$n = 4$

if $P = 0.1$

$P_{err} = 0.0523$

'm' errors in a block of 'n' bits.

$$P_{m,n} = \binom{n}{m} P^m (1-P)^{n-m}.$$

$$P_M \leq \sum_{m=t+1}^{n} P_{m,n}$$: Prob. of error of decoding CW if $> t$ errors occur.

✱ Perfect codes

like for a binary $q = 2$

q-ary (n, k) code with M CWs

Min. distance $= 2t + 1$

↳ no. of errors

Perfect code : The one that achieves Hamming bound.

$$M\left\{\binom{n}{0} + \binom{n}{1}(q-1) + \cdots + \binom{n}{t}(q-1)^t\right\} = q^n.$$

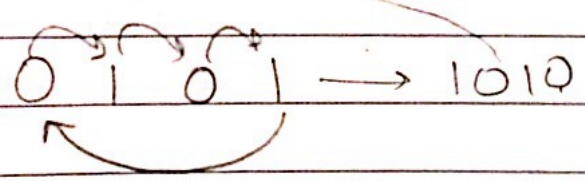
eg	n	q	M	t	$2t+1$
(1)	23	2	2^{12}	3	7
(2)	90	2	2^{78}	2	5
(3)	11	3	3^6	2	5

✱ Cyclic Codes (CC)

Defin^n : 1) C is a linear code.
2) Any cyclic shift of CW gives CW.

eg : C = { 0000, 0101, 1010, 1111 }

Cyclic shift :



eg (2) C = { 0000, 0110, 1001, 1111 }

Consider 0 1 1 0 → 0 0 1 1

0011 ∉ C

So, it is a CW

Hence, C is not a cyclic code.

Let

* Defin^{n} $f(x) = f_0 + f_1(x) + \cdots f_m x^m$.

s.t $f_i \in GF(q)$

↳ GF : Gamma field

$q = 2 \Rightarrow 0, 1$

$3 \Rightarrow 0, 1, 2$

⋮

✓ If $f_m \neq 0$, degree of $f(x) = m$.

✓ If $f_m = 1$, its called a Monic polynomial.

eg $f(x) = 2 + x + x^2 + 2x^4$

$g(x) = 1 + 2x^2 + 2x^4 + x^5$

Here, its GF(3) → $\Rightarrow 0, 1, 2$

↳ Value of coeff

$f(x) + g(x) = (1+2) + x + (1+2)x^2 + (2+2)x^4 + x^5$

If GF(3) $\Rightarrow$ MODULO 3 ADDITION

$= 0 + x + 0 + 1 \times x^4 + x^5$

$= x + x^4 + x^5$.

GF(2) $\Rightarrow 0, 1 \Rightarrow$ MODULO 2 ADDITION

$f(x) + g(x) = 1 + x + x^2 + x^5$.

$f(x) * g(x)$ → in modulo 3

$= (2 + x + x^2 + 2x^4)(1 + 2x^2 + 2x^4 + x^5)$

$= 2x + 2(2x^2) + 2(2x^4) + 2(1 \times x^5)$

$+ \cdots$

GF(3) $= 2 + x^2 + x^4 + 2x^5$.

Puffin
Date
Page

Division algorithm for polynomials :-

$a(x) = x^3 + x + 1$
$b(x) = x^2 + x + 1$
} over GF(2)

Then, $a(x) = q(x) \cdot b(x) + r(x)$

(∵ Dividend = Divisor × Quotient + Rem.)

So,

$$\begin{array}{r|l} x^3 + x + 1 & x + 1 \\ x^3 + x + x^2 & \\ - \quad - \quad (-) & \\ \hline 0 + 0 + 1 + x^2 & \\ 1 + x^2 + x & \\ - \quad - \quad (-) & \\ \hline x & \end{array}$$

→ take positive instead

→ Take +ve instead

→ Continue till power of $r(x)$ < power of $b(x)$

Note :
$(-1) = (+1)$
↳ in GF(2)
⇒ Addition is same as subtraction in GF(2)

Hence, we have $q(x) = x + 1$
$r(x) = x$.

* Degree of remainder :
$\deg(r(x)) < \deg(b(x))$

* We know $\frac{4}{10} \neq \frac{10}{4}$ (Division in decimal value system)
Ily, $\frac{a(x)}{b(x)} \neq \frac{b(x)}{a(x)}$ in GF(2)

Moreover, in above, we can't divide $b(x)$ by $a(x)$ as $\deg(a(x)) > \deg(b(x))$.

Puffin
Date 20/3/14
Page

* Defin^n: Concept of Congruent Modulo:

we say $g(x) \equiv h(x) \bmod f(x)$ is true,

if $g(x) - h(x)$ is divisible by $f(x)$

(In simple terms $4 \equiv 1 \pmod 3$
$\Rightarrow \frac{4-1}{3} =$ integer $(= 1)$)

eg :- If $g(x) = x^4 + x^2 + 1$
$h(x) = x^5 + x^2 + 1$
$f(x) = x^4 + 1$
defined over GF(2)
↳ ⇒ see modulo 2.

So, $g(x) - h(x) = x^9 - x^5 = x^9 + x^5$
$= x^5(x^4 + 1)$
$= x^5 f(x)$

∴ Subtraction = addition for modulo 2.

So, $\frac{g(x) - h(x)}{f(x)}$ = integer.

So, $f(x) \mid (g(x) - h(x))$

[$a \mid b \Rightarrow a$ divides b]

* Concept of Ring: $\left(\frac{F[x]}{f(x)}\right)$ ← field

$\frac{F[x]}{f(x)}$ are polynomials in $F[x]$
↳ $\deg\left(\frac{F[x]}{f(x)}\right) < \deg f(x)$

Puffin
Date
Page

eg: Consider $(x+1)^2$ in $\frac{F[x]}{f(x)}$

↳ Given :- $f(x) = x^2 + x + 1$

Use GF(2)

Now, $(x+1)^2$ in $F[x] = x^2 + x + x + 1$

$= x^2 + (1+1)x + 1$

$= x^2 + 1$

→ 0 (in modulo 2)

So, the ring $\frac{F[x]}{f(x)}$ ↳

$$x^2+x+1 \;\overline{)\; x^2+1 \;(\; 1}$$
$$x^2 + 1 + x$$
$$x$$

$\Rightarrow (x+1)^2$ in $F[x]$ $= x$ in $\frac{F[x]}{f(x)}$

Now, see using GF(3)

→ $\neq 0$ in GF(3)

So, $(x+1)^2 = x^2 + 2x + 1$

So, ring can be found as

$$x^2+x+1 \;\overline{)\; x^2+2x+1 \;(\; 1}$$
$$x^2 + x + 1$$
$$x$$

Hence, again

$(x+1)^2$ in $F[x] = x$ in $\frac{F[x]}{f(x)}$

Puffin
Date
Page

eg (2) :- Consider $\frac{F[x]}{x^2+x+1}$ defined over GF(2)

Note : highest degree of Polynomial in ring i.e., the remainder will be less than deg (f(x)) i.e < 2

So, deg (remainder) = $1^+ \left(x^1\right)$

Ily, if deg. (f(x)) = n, then, deg (remainder) $\leq n-1$

★ Note : $\frac{F[x]}{f(x)}$ has q^n elements.

q → value of GF [GF(q)].

n → degree of f(x)

So, here, no. of elements

$= q^n = 2^2 =$ 4 elements

0, 1, x, x+1

Adding the elements :

+	0	1	x	x+1
0	0	1	x	x+1
1	1	0	x+1	x
x	x	x+1	0	1
x+1	x+1	x	1	0

Ily, we can do for multiplicⁿ

·	0	1	x	x+1
0	0	0	0	0
1	0	1	x	x+1
x	0	x	(x+1)	(1)
x+1	0	x+1	(1)	(x)

$= x^2 + x$

$x^2+x+1 \overline{)x^2+x}$ (1)

x^2+x+1

(1)

* Irreducible and Monic Polynomials are Prime Polynomials.

deg ≥ 1 always → Poly. in which coeff. of highest power = 1

Puffin Date Page

* Defin^n: $f(x)$ in $F[x]$ is reducible if

$$f(x) = a(x) \cdot b(x)$$

↳ $a(x)$, $b(x)$: polynomials

↳ $\deg(a(x)), \deg(b(x)) < \deg(f(x))$

else, its irreducible*.

* Theorem:

① $f(x)$ has linear factor $x-a$ if $f(a) = 0$

② $f(x)$ in $F[a]$ of deg. 2 or 3 over $GF(q)$ is irreducible iff $f(a) \neq 0 \ \forall \ a \in GF(q)$

③ $x^n - 1 = (x-1)(x^{n-1} + x^{n-2} + \dots + 1)$

eg Let $f(x) = x^3 - 1$ considered over $GF(2)$

We know

$x^3 - 1 = (x+1)(x^2 + x + 1)$; We have $GF(2)$

$\Rightarrow a = \begin{cases} 0 \longrightarrow f(a) = 0+0+1 = 1 \neq 0 \\ 1 \longrightarrow f(a) = 1+1+1 = 1 \neq 0 \end{cases}$

$\Rightarrow x^2 + x + 1$ is irreducible & is a Monic polynomial.

$\Rightarrow$ Hence, for x^2+x+1 in $GF(2)$, it's a Prime polynomial

* Notation : $\frac{F[x]}{f(x)} \triangleq R_n$

* Theorem 4.2
A ring $\frac{F[x]}{f(x)}$ is a field iff $f(x)$ is a Prime Polynomial in $F[x]$

ex:- $p(x) = x^3 + x + 1$ over GF(2)
we find :- $p(0) \neq 0$ & $p(1) \neq 0$. } & monic
⇓
Its a prime polynomial

for $n = 3$:- $\frac{F[x]}{f(x)}$ field in GF(8)

The 8 elements are

0	x^2
1	x^2+1
x	x^2+x
$x+1$	x^2+x+1

} Choosing any poly. in $F[x]$, when divided by $f(x)$ (Gmier) → we get one of the polys from this set.

* Properties :
• Consider $f(x) = x^n - 1$

P1 $x^n \bmod x^n - 1 = 1$
$x^{n+1} \bmod x^n - 1 = x$

P2) $C_0\ C_1 \dots C_{n-1}$
$C(x) = C_0 + C_1 x + \dots + C_{n-1} x^{n-1}$
↳ C_i are 0's or 1's.
Eg, $01101 \equiv 0 + x + x^2 + 0 \cdot x^3 + x^4 + x^5$.

P3 $x \cdot C(x) = C_0 x + C_1 x^2 + \dots + C_{n-2} x^{n-1} + C_{n-1} x^n$

$\therefore x^n = 1 \Rightarrow C_{n-1}$ is a constant $= 1$

$\Rightarrow x\,C(x) = C_{n-1} + C_0 x + C_1 x^2 + \dots + C_{n-2} x^{n-1}$

$\hookrightarrow = C_{n-1} C_0 C_1 C_2 \dots C_{n-2}$.

It's kind of forming a cycle

↳ a CYCLIC CODE

---

* Theorem 4.3

Code C in R_n is cyclic iff C satisfies:

① $a(x), b(x) \in C \Rightarrow a(x) + b(x) \in C$.

② $a(x) \in C$, $r(x) \in R_n \Rightarrow a(x) r(x) \in C$.

eg Consider $f(x) = x^2 + 1$ in R_3 over GF(2)

So,

a polynomial in $R_3 = \dfrac{F[x]}{x^3 - 1}$

$r(x)$ can be written as :-

$r(x) = r_0 + r_1 x + r_2 x^2$.

↳ $r_0, r_1, r_2 = 0, 1$ only ($\because$ GF(2))

↳ 8 possibilities.

$\Rightarrow r(x) = 0 \quad 1 + x$

$1 \quad 1 + x^2$

$x \quad x + x^2$

$x^2 \quad 1 + x + x^2$

Creating codewords :-

$f(x) \cdot r(x) \pmod{(x^3-1)}$

$\Rightarrow (x^2+1)\,0 = 0$

$(x^2+1)\,1 = x^2+1$

$(x^2+1)\,x \bmod (x^3-1) = 1+x$

$x^3-1 \;\overline{)\; x^2+x\;(}$

$-x^3 \quad -1$

$x+1$

Why do it

So, the reduced codes :

$0 \longrightarrow 0$

$1 \longrightarrow 1+x^2$

$x \longrightarrow 1+x$

$x^2 \longrightarrow x+x^2$

$1+x \longrightarrow x+x^2$

$1+x^2 \longrightarrow 1+x$

$x+x^2 \longrightarrow 1+x^2$

$1+x+x^2 \longrightarrow 0.$

Leaving the poly,

$[0, 1+x, 1+x^2, x+x^2]$

Includes all polynomials.

So, CW are

$= \begin{bmatrix} 000 \to 0+0x+0x^2 \\ 110 \to 1+1x+0x^2 \\ 101 \\ 011 \end{bmatrix}$

So, Code book is

$C = [000, 110, 101, 011]$

↳ Code book is cyclic

eg Find all binary codes of block length 3.

GF(2)

*

Idea: ① Factorize $x^3-1 = (x+1)(x^2+x+1)$

Generator Poly.	Code Poly. $\{R_3\}$ → includes all poly. in R_3	Code
will be there by default ← (1)		$\{000, 001, \ldots, 111\}$
		$d_{min} = 1$
$x+1$	$\{0, x+1, x^2+x, x^2+1\}$	$\{000, 011, 110, 101\}$
	* Take $(x+1)$ & multiply all poly in R_3. Then do modulo (x^3-1)	$d_{min} = 2$
x^2+x+1	$\{0, x^2+x+1\}$	$\{000, 111\}$
$x^3+1=0$ ← zero polynomial. → modulo 2 ($GF(2)$) ⇒ $x^3-1 = x^3+1$	$\{0\}$	$\{000\}$
		$d_{min} = 3$

★ If generator polynomial = $g(x)$ & $i_{(x)}$ are encoded bits

So, $C_{(x)} = i_{(x)}\, g_{(x)}$

★ Theorem 4.5 :

$$g(x) = g_0 + g_1 x + \cdots + g_r x^r$$

↳ deg = r.

Then, we can have a G matrix, defined as:

$$G = \begin{bmatrix} g_0 & g_1 & \cdots & g_k & 0 & \cdots & 0 \\ 0 & g_0 & \cdots & g_k & 0 & \cdots & 0 \\ & & & & & & \end{bmatrix}$$

$K (= n - k)$ → no. of rows

n → no. of columns

Idea :- We have k elements of g. ($g_1, g_2 \ldots g_k$). So, fill that in 1st row. For the remaining elements fill zero.

Now, shift right by one element for next row.

Proceed this way.

eg If $g(x) = 1 + x$

$k = 1, n = 3$.

⇒

$$G = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad K = 2$$

$n = 3$

→ shifted by one element to right.

$i_{(x)}$ can have diff possibilities :

$i = 0\ 0 \longrightarrow 0 + 0x = 0.$

$0\ 1 \longrightarrow 0 + 1x = x$

$1\ 0 \longrightarrow 1 + 0x = 1$

$1\ 1 \longrightarrow 1 + 1x = 1 + x$

Note: We make CW for a binary bit as

$$C_0 C_1 C_2 C_3 \ldots C_n = C_0 + C_1 x + C_2 x^2 + \ldots C_n x^n$$

Now, $g(x) = 1 + x$

$\Rightarrow (x^3 - 1 = g(x)(x^2 + x + 1)$

$\Rightarrow x^3 - 1 = g(x) \underbrace{h(x)}$

parity check polynomial

→ assume if not given

(2): Consider $n = 7$, GF(2)

So, poly. to start with $= x^7 - 1$

$$x^7 - 1 = \underbrace{(x-1)(x^3 + x + 1)(x^3 + x^2 + 1)}$$

not exact expansion.

This comes after using

$-1 = +1 \rightarrow$ (modulo 2)

We can use any of these 3 poly. as generator

let $g(x) = x^3 + x + 1$ deg 3

then, $h(x) = (x-1)(x^3 + x^2 + 1) = x^4 + x^2 + x + 1$

↳ deg 4

$g(x) = \underbrace{x^3 + x + 1}$

writing in standard form to avoid confusion

$\Rightarrow g(x) = \underbrace{1 + x + 0 \cdot x^2 + x^3}$

1101

Ex.

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

$k = n - k = 4$

$n = 7$

& for H matrix, follow reverse process.

$$H = \begin{bmatrix} h_k & h_{k-1} & \cdots & h_0 & 0 & 0 & \cdots & 0 \\ 0 & h_k & \cdots & \cdots & h_0 & 0 & \cdots & 0 \\ \vdots & & & & & & & \\ & & & & & & & \end{bmatrix}$$

no. of rows $= n - k = k$

n columns

Here, $n = 7$, $k = 4 \Rightarrow n - k = 3$.

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

$n - k = 3$

$n = 7$

$h(x) = x^4 + 0 \cdot x^3 + x^2 + x + 1$

10111

(11) are 10000, ... 1111 }
to, at receiver side.

$H\underline{v} = \underline{0}$ if $\underline{v} = (11)$

Alter Option 2 : Choose $g(x) = x^3 + x^2 + 1$

then, $h(x) = (x-1)(x^3 + x + 1)$

$= x^4 + x^2 - x^3 - 1$

$\Rightarrow h(x) = x^4 + x^3 + x^2 + 1$

(modulo 2 : + = −)

Note :

We dont choose $g(x) = x - 1$ ∵

difficulty in error correction

4 ∵ for $d^* = 3$, we can

↳ detect → 2 bit error

↳ correct → 1 bit error

If $g(x)$ is small, we may not be able to detect error.

Note :- $t(x) = \begin{cases} c(x) & ; \text{no error} \\ c(x) + e(x) & ; \text{with error} \end{cases}$

↳ $c(x) = i(x) g(x)$

Date 26/3/14
Page

§ CRC (Cyclic Redundancy Check)

Terms :
- ✓ Transmitted data, denoted by T
- ✓ Message, denoted by D.
- ✓ length of message, k bits.
- ✓ Pattern, P = n-k+1
- ✓ Frame check sequence (FCS) = n-k.
- ✓ Total no. of bits transmitted = n

eg :- D = 1010001101 (k = 10)

P = 110101 (n-k+1 = 6)

↳ FCS = n-k = 5 bits

n = 15 = n-k

✡ $T = 2^{FCS} D + R$

T : Transmitted data

Now, do $P \overline{)2^5 D}$ (Q)

...

R

✡ Note :

$2^5 D$: It is shifting my message by 5 bits in binary

So, why all this :-

$2^5 D \rightarrow$ 1010001101 00000

10 bits. FCS.

15 bits transmitted data.

So, in general, my transmitted polynomial, $T(x)$ is

$$T(x) = x^{n-k} D(x) + R(x)$$

If received poly, $V(x) = T(x) + E(x)$ → (1)
Divide (1) by P. If $R = 0$, error free.

eg Consider a standard of CRC: CRC-CCITT
Then,

$$P(x) = x^{16} + x^{12} + x^5 + 1$$

↳ $\Rightarrow n - k = 16$.

& $D(x) = x^9 + x^7 + x^3 + x^2 + 1$

So, $x^{n-k} D(x) = x^{16}(x^9 + x^7 + x^3 + x^2 + 1)$

$= x^{25} + x^{23} + x^{19} + x^{18} + x^{16}$

Now, do $x^{16} D(x) \div P(x)$

$$\begin{array}{r} x^9 + x^7 + x^5 \\ \hline x^{16} + x^{12} + x^5 + 1 \;\big)\; x^{25} + x^{23} + x^{19} + x^{18} + x^{16} \\ x^{25} + \qquad\qquad + x^{21} + x^{14} \\ \hline x^{23} + x^{21} + x^{19} + x^{18} + x^{16} + x^{14} + x^9 \\ x^{23} + \qquad + x^{19} \qquad\qquad + x^{12} \\ \hline x^{21} + x^{18} + x^{16} + x^{14} + x^{12} + x^9 + x^7 \\ x^{21} \qquad\qquad\qquad\qquad + x^5 \\ \hline x^{18} + x^{16} + x^{14} + x^{12} + x^9 + x^7 + x^5 \end{array}$$

GF(2)
$\Rightarrow -1 = +1$,

Proceeding similarly, we get

$$R(x) = x^{13} + x^{12} + x^{10} + x^9 + x^6 + x^5 + x^2 + x$$

So,

$$T(x) = x^{n-k} D(x) + R(x)$$

$$\Rightarrow T(x) = (x^{25} + x^{23} + x^{19} + x^{18} + x^{16}) + (x^{13} + x^{12} + x^{10} + x^9 + x^6 + x^5 + x^2 + x)$$

$$\Rightarrow T(x) = 1010001101001101100110$$

x^{25} x^{24} x^{23} ... x^0

★ Note: GF(8) was derived from GF(2) using poly. $P(x) = x^3 + x + 1$.

Element of GF(8):

$0, 1, x, x+1, x^2, x^2+x, x^2+1, x^2+x+1$

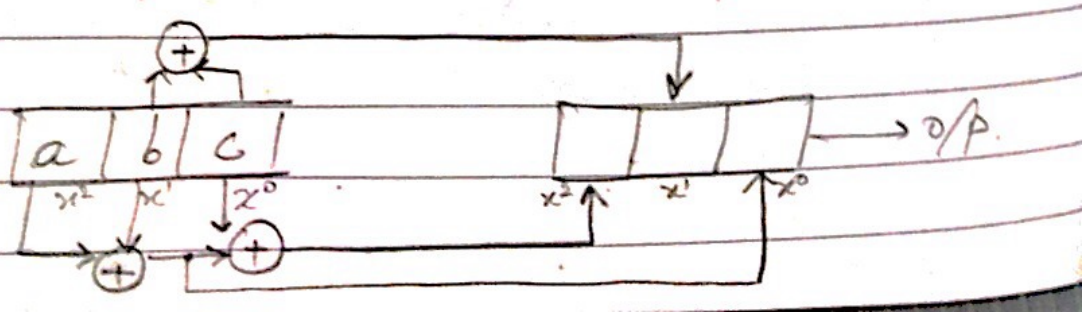
Now,

Take an arbitrary poly.: $ax^2 + bx + c$.

Multiplying with $(x^2 + x)$ (any element of GF(8))

$$\Rightarrow (ax^2 + bx + c)(x^2 + x) = (a+b+c)x^2 + (b+c)x + (a+b)$$

i/p → o/p



* Note : $x^n P(x) \equiv 2^n P$.

multiplying a poly.

⇒ shifting it left (in binary code)

↳ eg: If $x^n = x^3 \rightarrow P(x) = x^2 + x + 1$

$\Downarrow$

$P = 111$.

$x^3 P(x) = x^5 + x^4 + x^3$

$\Rightarrow 2^3 (111) \equiv 111000$. (shifted by 3 bits)

(Above is implemented using shift registers → adders (↓ modulo 2) and multipliers)

* PART OF CONVOLUTION ENCODER (PTO)

• Polynomial multiplication :

Consider two polynomials :- $a(x) = a_2 x^2 + a_1 x + a_0$

& $g(x) = g_2 x^2 + g_1 x + g_0$

So, $a(x) \cdot g(x) = a_2 g_2 x^4 + (a_2 g_1 + a_1 g_2) x^3 + (a_2 g_0 + a_0 g_2 + a_1 g_1) x^2 + (a_1 g_0 + a_0 g_1) x + a_0 g_0$.

Note : Code is given by $C(x) = i(x)\, g(x)$ ⟶ (1)

* Coefficients of multiplic^n can be seen by sliding one poly. over other.

(1) $g_2\ g_1\ g_0$

$a_0\ a_1\ a_2 \rightarrow$ → coeff. of x^4.

(2) $g_2\ g_1\ g_0$

$a_0\ a_1\ a_2 \rightarrow$

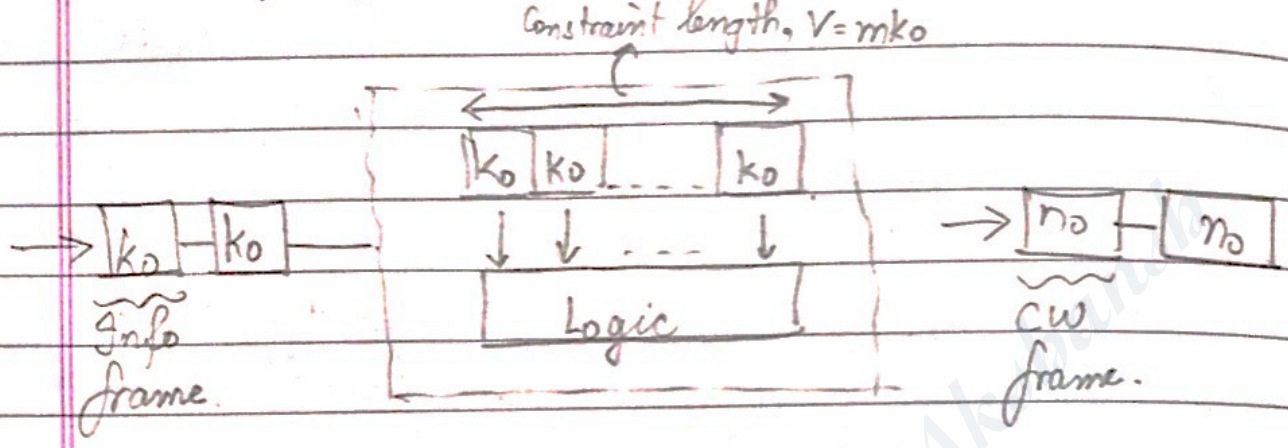
(3) $g_2\ g_1\ g_0$ over $a_0\ a_1\ a_2$ → coeff. of x^2.

(4) $g_2\ g_1\ g_0$ over $a_0\ a_1\ a_2$ → coeff. of x

(5) ---- → coeff. of x^0

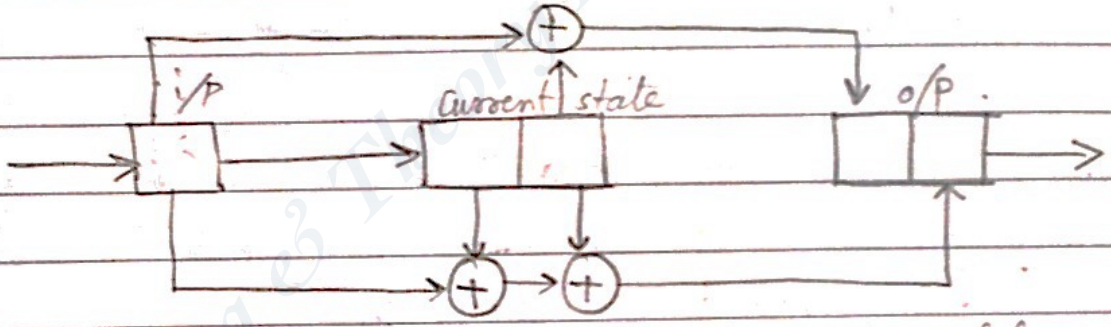
Shifting oper^n : Done in Shift registers. (implement^n shown later)

CONVOLUTION ENCODER
(n_0, k_0)



* $R = \frac{k_0}{n_0}$

eg



→ m=2 (taking 2 bits in shift register)

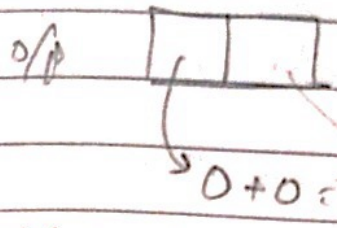
$k_0 = 1$, $V = (m)k_0 = 2$, $n_0 = 2$

$R = \frac{k_0}{n_0} = \frac{1}{2}$

Reference Table

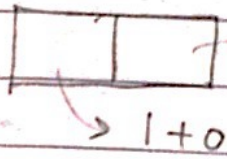
Incoming bit	Current state (CS) of encoder	Next state (NS) of encoder	Outgoing bits
0	00	00	00
1	00	10	11
0	01	00	11
1	01	10	00
0	10	01	01
1	10	11	10
0	11	01	10
1	11	11	01

If i/p bit = 0, current state = 00, what's o/p



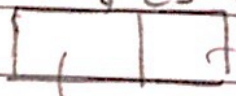
⇒ o/p = 00

If i/p = 1, CS = 00



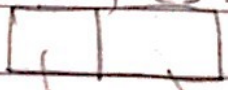
⇒ o/p = 11

If i/p = 1, CS = 01



⇒ o/p = 00

If i/p = 0, CS = 01

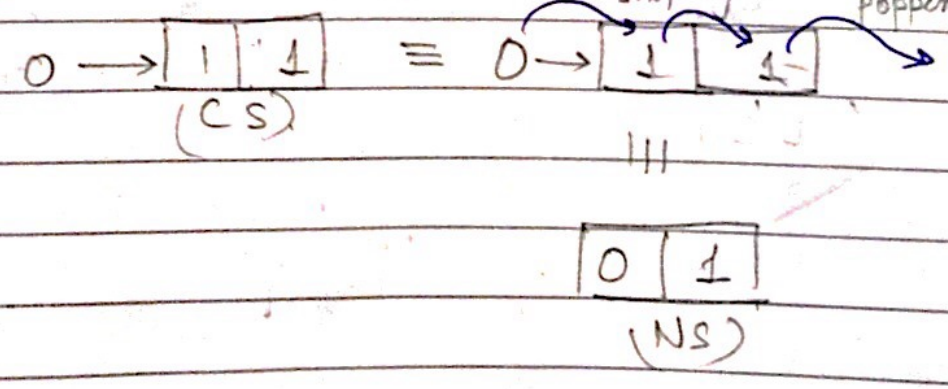


⇒ o/p = 11

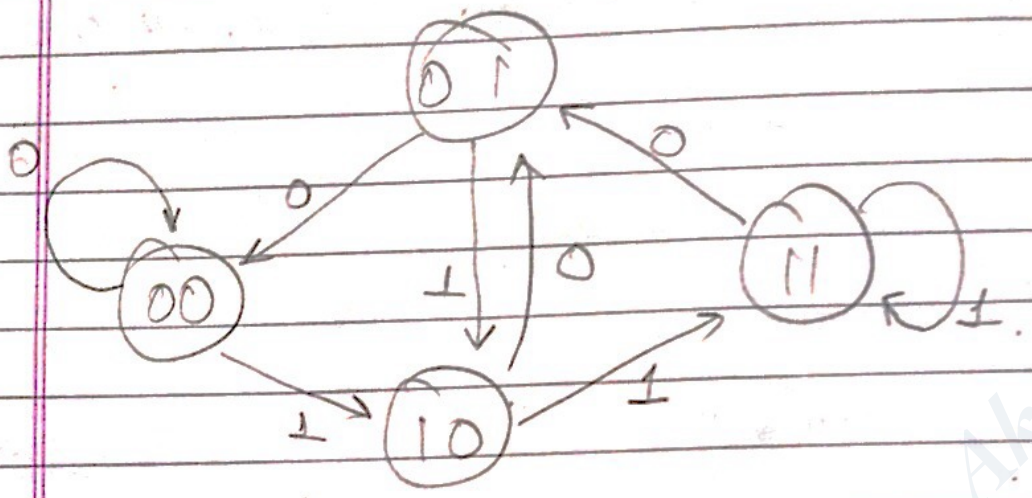
This creates the table (on prev. page)

So, just like in control systems, we can make a state diagram.

Note: Next state in table is got by shifting and popping.

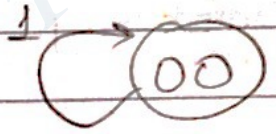


State diagram :-



How? : 11 —1→ 01

∴ I go from CS = 11 to NS = 01 when incoming bit = 1.



⇒ I go from CS = 00 to NS = 00 when i/p = 1

eg Consider a sample of i/ps → 1 1 0 1 0 1

Incoming bits	CS of encoder	NS of encoder	Outgoing bits
1	0 0 (assumed at start)	1 0 (NS becomes CS)	1 1 (come from state diagram or table)
1	1 0	1 1	1 0
0	1 1	0 1	1 0
1	0 1	1 0	0 0
0	1 0	0 1	0 1
1	0 1	1 0	0 0

↓ See i/p & cs & pick ans. from table here.

Puffin
Date
Page

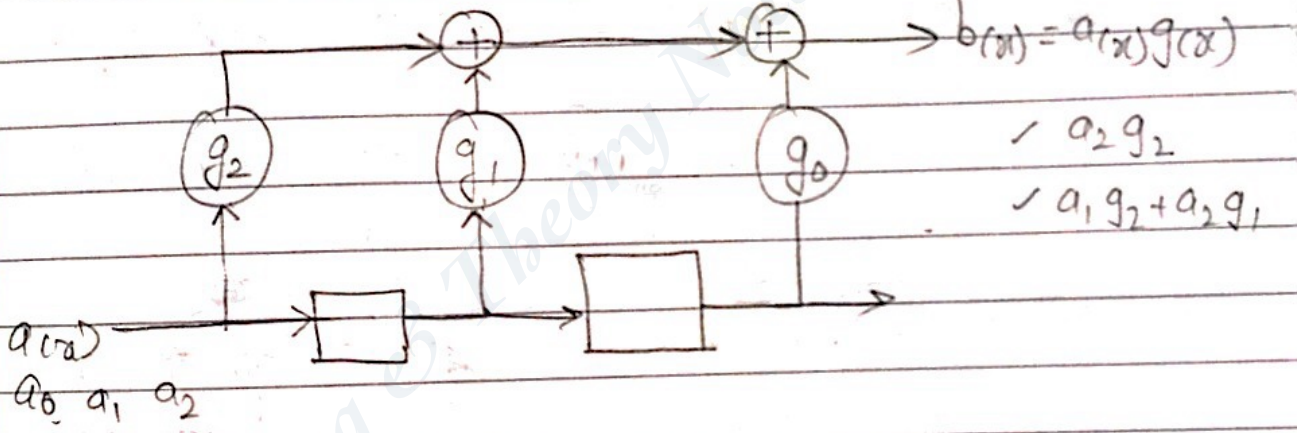
Eg: i/p frame → Convolution encoded i/p frame

110101 → 111010000102

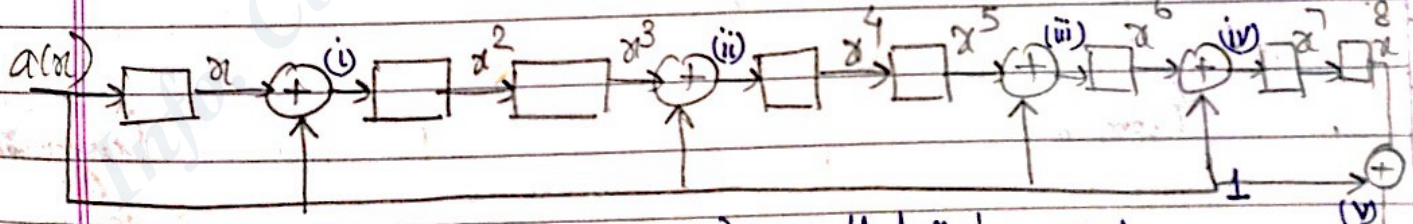
* Polynomial Multiplic^n :- $a_2x^2+a_1x+a_0$ $g_2x^2+g_1x+g_0$

As seen earlier, let the poly. be $a(x)$ & $g(x)$.

Using shift registers : FIR filter multiplication :-



Eg: Now, consider $g(x) = x^8+x^6+x^5+x^3+x+1$



Feed forward (FIR) → Multiplic^n by $g(x)$

↳ what is happening ? → We are doing $a(x).g(x)$ →

At (i) : $(1+x)\,a(x)$

(ii) : $(1+x+x^3)\,a(x)$

(iii) : $(1+x+x^3+x^5)\,a(x)$

(iv) : $(1+x+x^3+x^5+x^6)\,a(x)$

(v) : $(1+x+x^3+x^5+x^6+x^8)\,a(x) = \overbrace{g(x).a(x)}^{b(x)}$.

Now, at Receiver Side

$$\frac{C(x)}{g(x)} = f(x) + 0. \text{ (Remainder)}$$

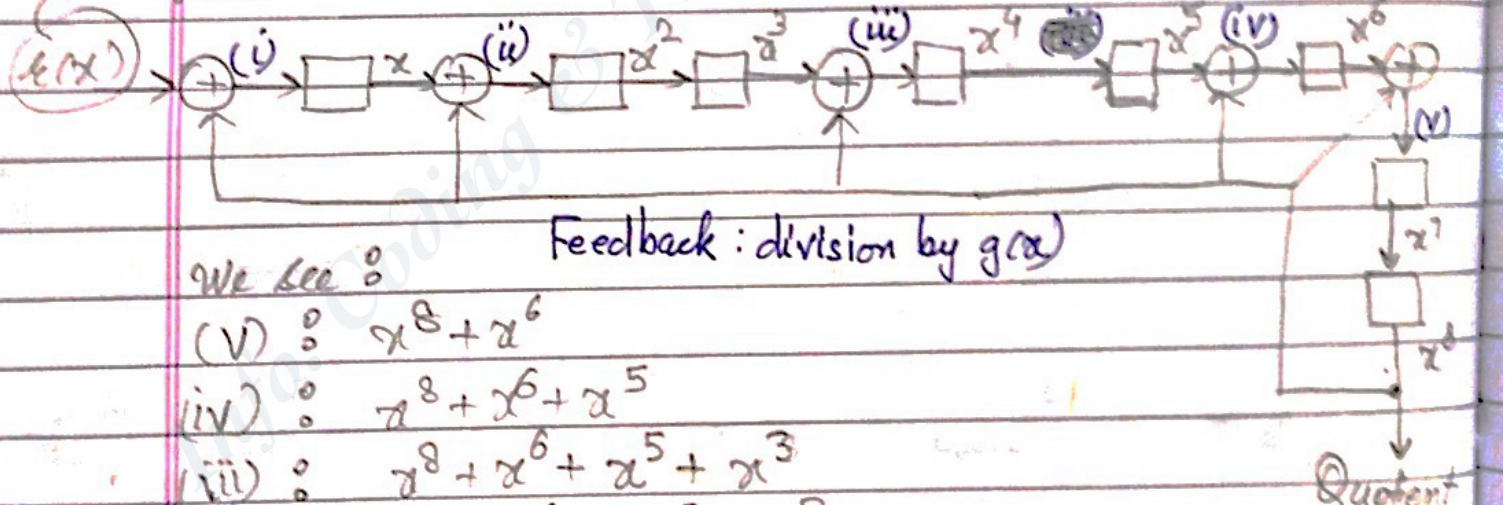
$$\left(\because C(x) = f(x)\, g(x)\right)$$

$f(x) = C(x) + e(x)$

$f(x)$ → received vector

$\frac{f(x)}{g(x)} \neq 0$ remainder

Q Using same $g(x)$ as before, make a block diagram to do $\frac{C(x)}{g(x)}$



We see :

(v) : $x^8 + x^6$

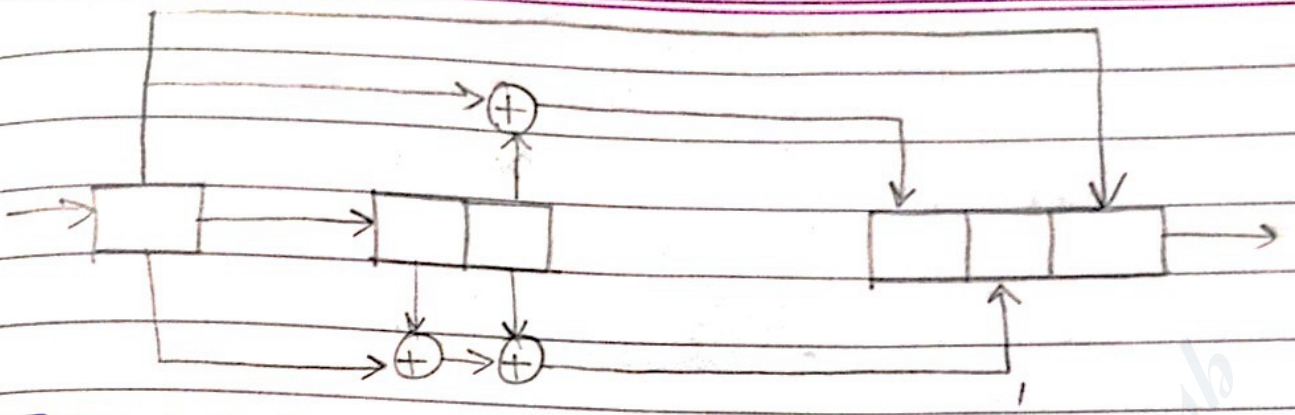
(iv) : $x^8 + x^6 + x^5$

(iii) : $x^8 + x^6 + x^5 + x^3$

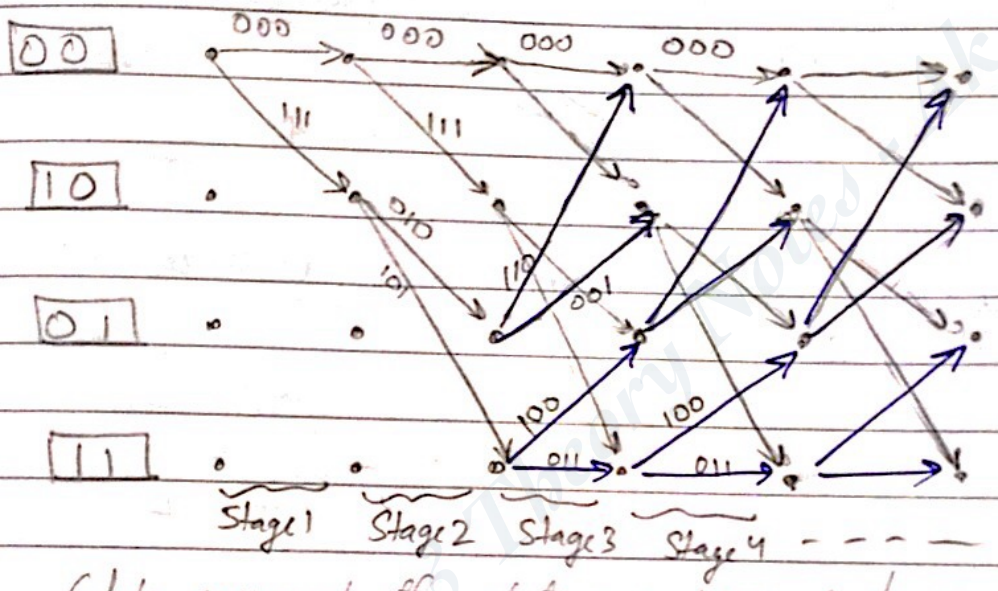
(ii) : $x^8 + x^6 + x^5 + x^3 + x$

(i) : $x^8 + x^6 + x^5 + x^3 + x + 1$

★ The boxes are shift registers & after shift oprns, it contains Remainder (R(x))

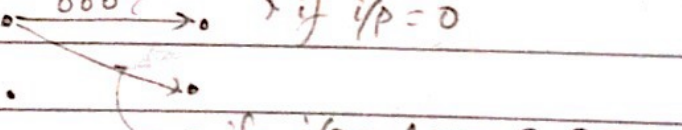


TRELLIS DIAGRAM



(dots represent the state → present to next state)

See:



= 1 0

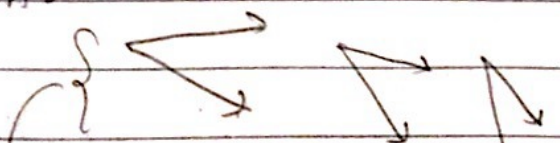
o/p =

1+0=1

1+0+0

1

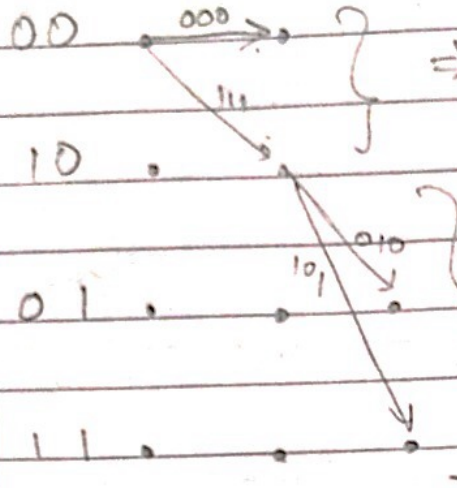
Convention:



line above : i/p = 0

line below : i/p = 1

* o/p is written above the line way →)



⇒ If I give i/p = 0 with CS = 00, o/p = 000 & NS = 00.

& if I give i/p = 1 with CS = 00, o/p = 111 & NS = 10.

⇒ If I give i/p = 0 with CS = 10, o/p = 010 & NS = 01

& if I gave i/p = 1 with CS = 10, o/p = 101 & NS = 11

☆ Example: Consider: unencoded data = 0000

So, in binary:

Transmitted (T_x) = 000 000 000 000

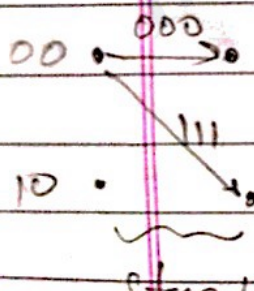
Received (r) = 010 000 100 001

Note: Bits are received in groups of three.

Stage 0: On receiving 010, we have the diagram & tree with us. We know CS = 00.

Now, if i/p = 1, o/p = 111

i/p = 0, o/p = 000



Comparing with 010, we see that for

111 → distance = 2

000 → distance = 1

So, 000 more probable ⇒ its 000.

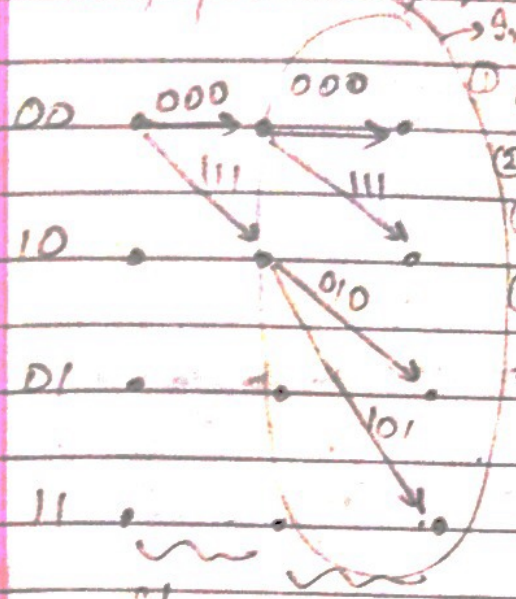
(may/maynot be true → will be cleared later)

* Note: Whenever you find distance (i.e hamming distance), find cumulative distance (Previous stage(s) + Current stage) (for shortest)

Puffin
Date
Page

Stage ② Next set of 3 bits is 000

Now, for this, we see the second stage.



Stage 1 Stage 2

→ For 2nd stage, consider all the 4 scenarios: (Stage 1 → Stage 2)

① 000 → 000 ⇒ d = 1 + 0 (d for stage 1 + d of stage 2)
= 1

② 000 → 111 ⇒ d = 1 + 3 → distance b/w 000 & 111 what got stage 2

③ 111 → 010 ⇒ d = 2 + 1 → Stage 2 distance
→ Stage 1 distance

④ 111 → 101 ⇒ d = 2 + 2

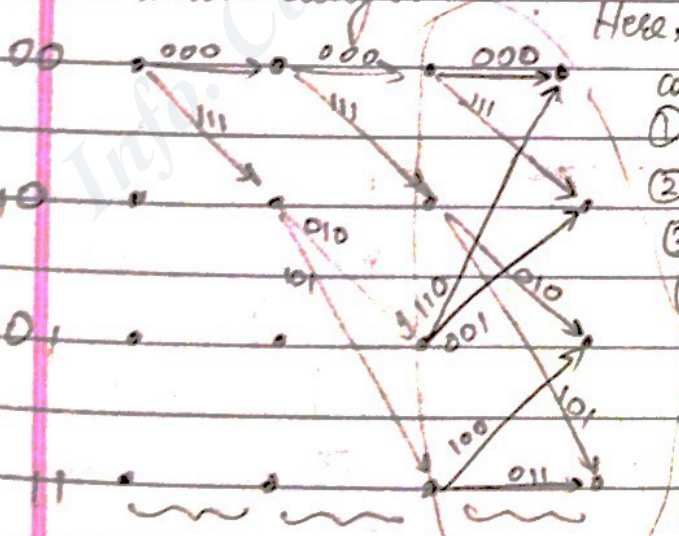
So, in Path ① → d = 1
Path ② → d = 4.
Path ③ → d = 3
Path ④ → d = 3

The shortest path is Path ① (Note: we find d by seeing the bit changes from the bit sequence we got (000, here) and the various bit combinations of stage 2)

So as of now, we have 000 000
Stage 1 Stage 2

Stage ③ Next set is 100

For the 3rd stage, the combinations are got from Trellis diagram



Stage 1 Stage 2 Stage 3

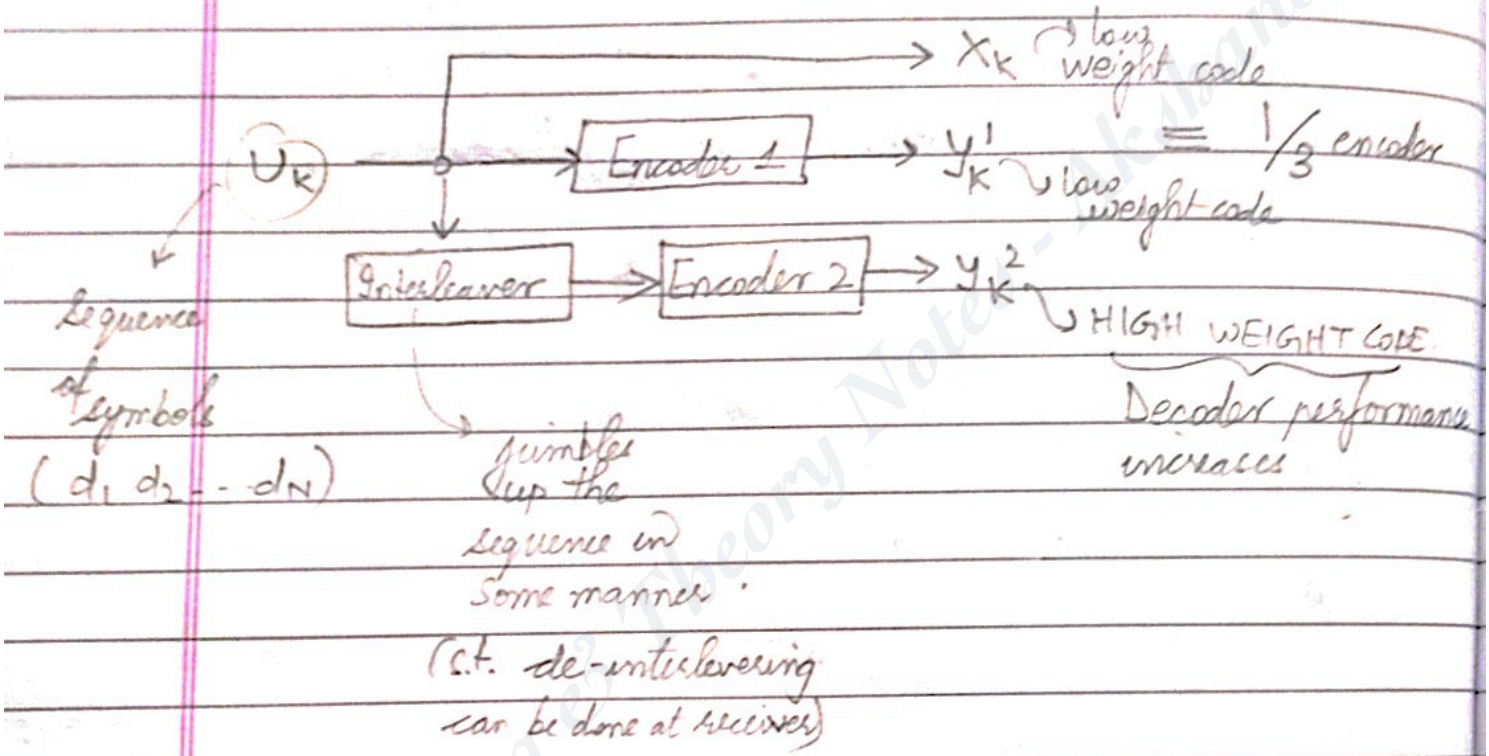
Here, for stage 3, the possible paths (or combinations) are:

① 000 → 000 → 000 : d = 1 + 0 + 1 = 2 (1 → 3rd stage)
② 000 → 000 → 111 : d = 1 + 0 + 2 = 3
③ 000 → 111 → 010 : d = 1 + 3 + 2 = 6
④ 000 → 111 → 101 : d = 1 + 3 + 1 = 5
⑤ 111 → 010 → 110 : d = 2 + 1 + 1 = 4
⑥ 111 → 010 → 001 : d = 2 + 1 + 2 = 5
⑦ 111 → 101 → 100 : d = 2 + 2 + 0 = 4
⑧ 111 → 101 → 011 : d = 2 + 2 + 3 = 7

From all these combinations, path ① is shortest. So, 000 000 000
Stage 1 Stage 2 Stage 3.

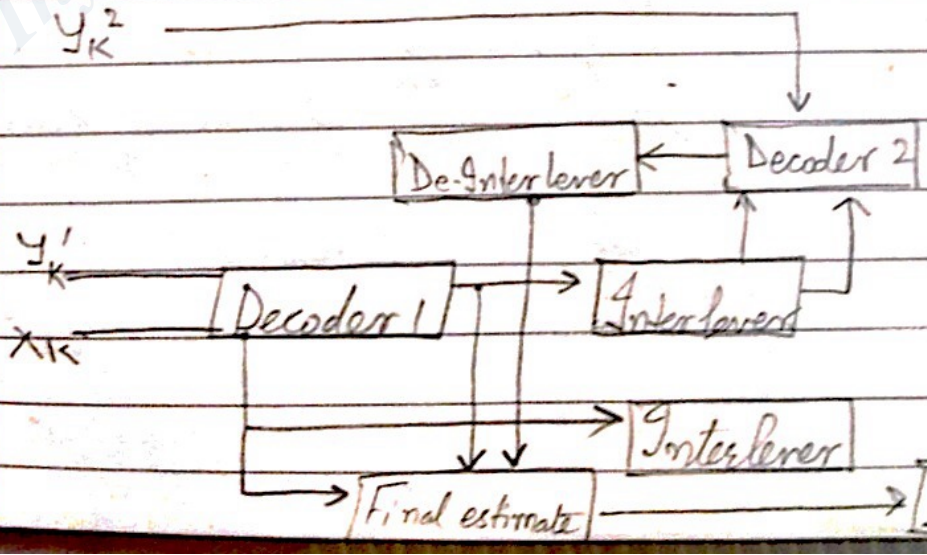
... Similarly, we can find the path, and hence get back our message.

✡ TURBO CODING.



✡ Weight or Hamming weight of a code : No. of 1's in a CW

✡ TURBO DECODING.



Puffin
Date
Page

Interleaving happens in case of burst errors.

✱ Rate Distortion Theory :-

Talks about determining the minimal no. of bits per symbol, as measured by rate R, that should be communicated over a channel, so that source (i/p signal) can be approximately reconstructed at receiver (o/p signal) without exceeding a given distortion D.

✱ Source coding → Quantizⁿ → Distortion ⇒ vary channel capacity.

✱ Channel Coding.

eg Consider $X \rightarrow \hat{X}$

Assuming 1 bit

$\mathcal{X}$ = 00, 01, 10, 11

$X^n \rightarrow$ [Encoder] $\rightarrow$ [Decoder] $\rightarrow X^n$

$f^n(X^n) \in (1, 2, \ldots 2^{nR})$

↳ R rate of channel

Puffin
Date
Page

* Defⁿˢ:

1) Distortion Measure: It is bounded, & defined as:

$$d_{max} \triangleq \max\left[d(x, \hat{x})\right] < \infty$$

$\hookrightarrow \forall\ x \in X, \hat{x} \in \hat{X}$.

* From hamming distance,

$$d(x, \hat{x}) = \begin{cases} 0, & x = \hat{x} \\ 1, & x \neq \hat{x} \end{cases}$$

* Squared error distance:

$$d(x, \hat{x}) = (x - \hat{x})^2$$

2 * Distance between sequences

$$d(x^n, \hat{x}^n) = \frac{1}{n} \sum_{i=1}^{n} d(i, i)$$

3 * Rate Distortion Code: (consider sequence: $2^{nR}, n$)

We have

encoder: $\equiv n$

$$f_n : X^n \to \{1, 2, \ldots, 2^{nR}\}$$

$$g_n : \{1, 2, \ldots, 2^{nR}\} \to \hat{X}^n.$$

4. Distortion :-

$$D = E\left[d\left(X^n, g_n\left(f_n(X^n)\right)\right) \right]$$

Distortion

↳ if $X^n \leq D$.

↳ $\lim_{n \to \infty}$

$D < 0$

We notice, (R, D) form a pair

↳ called Achievable Rate Distortion region.

R(D), D(R)

R(D): Rate Distortion fn. ($f_n(D)$) Gives (R, D) region

D(R): Varying D with fixed R. We get (R, D) region.

$$\star \quad R^{(I)}_D = P(\hat{x}|x) : \sum_{x, \hat{x}} P(x)\, P(\hat{x}|x)\, d(x, \hat{x}) \leq D$$

↳ we are minimizing $I(X; \hat{X})$ → distortion is bounded

Theorem:- 13.2.1

$\star$ $R(D) = R^{(I)}(D)$

R(D): source coding; $R^{(I)}(D)$: channel coding

Theorem 13.3.1:

If we have a binary source (0 or 1) → Bernoulli's distribⁿ; probability = p

& consider hamming distance

$$R(D) = \begin{cases} H(P) - H(D) & ; 0 \leq D \leq \min(P, 1-P) \\ 0 & ; D > \min(P, 1-P) \end{cases}$$

Theorem 8 - If ∃ Gaussian Source $N(0, \sigma^2)$

Consider Square error distortion

$$R(D) = \begin{cases} \frac{1}{2} \log\left(\frac{\sigma^2}{D}\right) & ; 0 \leq D \leq \sigma^2 \\ 0 & ; D > \sigma^2. \end{cases}$$

~

* Source Coding ↓ Distortion (Quantization)

Channel Coding ↔ Capacity of Channel.

(dirⁿˢ show what affects what)

* Transmitter side — Receiver side.

Random Variable: x → $p(x)$ → $p(y|x)$ → y

* Rate of distortion,

$$R(D) = \min_{q \in B} \min_{p \in A} D(p \| q)$$

→ Kullback Leibner distance

→ Let $A \equiv p(x) \leftrightarrow p(x, y)$

$q(x) \equiv p(x)\hat{r}(x)$

→ $$R(D) \equiv \min_{q \in B} \min_{p \in A} D\left(p(x) \| p(x)\hat{r}(x)\right)$$

* Kullback Leibner distance is related to mutual info. which is related to channel capacity :

$$C = \max_{q(x|y)} \max_{r(x)} \sum_{x} \sum_{y} r(x) P(y|x) \log \frac{q(x|y)}{r(x)}$$

comes from the distortion at receiver side

→ $$q(x|y) = \frac{r(x)\, P(y|x)}{\sum_{x} r(x)\, p(y|x)}$$

Sum of all possible symbols

→ $$r(x) = \frac{\prod_{y} \left(q(x|y)\right)^{P(y|x)}}{\sum_{x} \prod_{y} \left(q(x|y)\right)^{P(y|x)}}$$

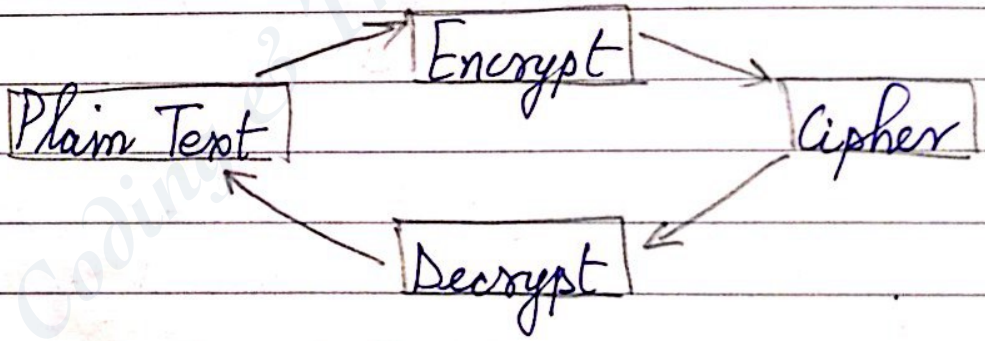
Date 8/4/14
Page

CRYPTOGRAPHY

↳ Mainly from Reference Books R3, R4, R5

Terminologies :

- Cryptosystem : A sys that encrypts at one end & decrypts at another
- Cryptanalysis ⟷ Cryptanalyst
  analysing sys. & seeing if it can be hacked / not
  ↳ who analyses
- Attacker
  ↳ who attacks or hacks into systems
- Cracking, Hacking.



- Data Encryption Standard (DES)
- Riviest Shamir Adelman (RSA)
- Diffie Hellman

} Public key Data encryption Algorithms

- IDEA
- Pretty Good Privacy (PGP)
- Elliptic Curve Cryptography
- Quantum Cryptography
- Biometric encryption

} other algorithms

- Message Authentication
- Hashing Functions.
- Digital Signatures.

---

eg① Say, we are given a key, x = 1011 and data in form: Voice, video, text → 0110 0010 1001 1111

Break the data in blocks of 4 & XOR data with key

```
 0110    0010    1001    1111
⊕1011   ⊕1011   ⊕1011   ⊕1011
 1101    1001    0010    0100
```

So, transmitted data : 1101 1001 0010 0100.

At receiver side, XOR the blocks with key again

```
 1101    1001    0010    0100
⊕1011   ⊕1011   ⊕1011   ⊕1011
 0110    0010    1001    1111
```

So, received data : 0110 0010 1001 1111

hence, we get our data back.

(Basically, we are doing Data ⊕ key ⊕ key
= Data ⊕ 0
= Data)

With the key known, data can be decrypted.

Examples of ① & ② : SYMMETRIC Cryptography. or Single key Cryptography or Secret Key Cryptography

eg② Encryption algorithm for text messages.

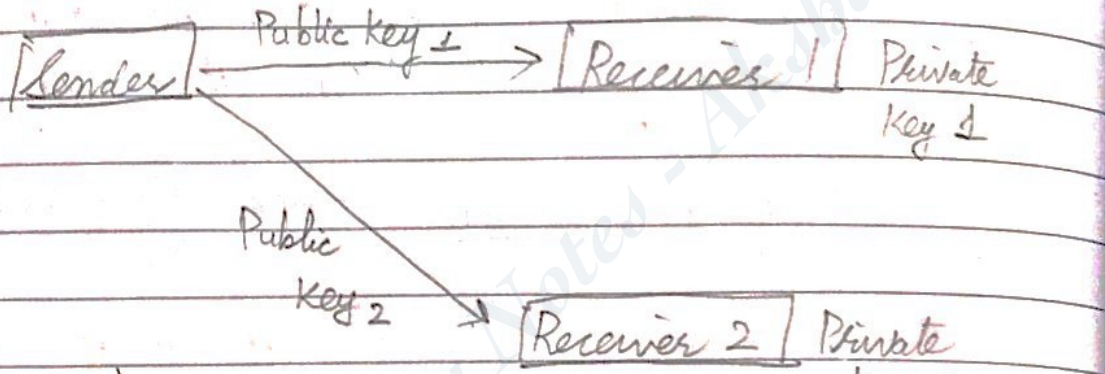
like, shifting every character

Character + 5, say.

A → F
B → G
... Z → E.

So, MYNAMEIS → RDSFRJNX

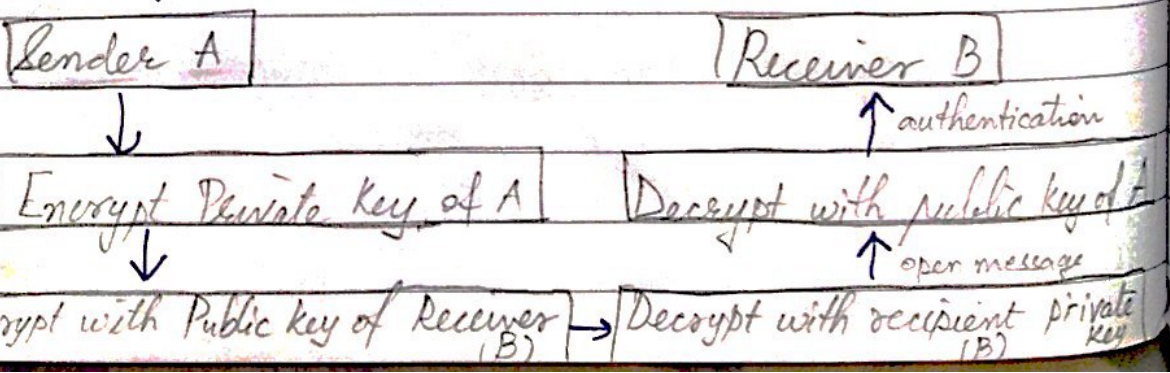
eg Public Key Cryptography
↳ by Diffie Hellman
↳ Is asymmetric encryption
↳ ∃ public + private key
↳ extra key with sender & receiver.



↳ ∃ public keys 1 & 2 for each receiver to encrypt
↳ Corresponding private keys 1 & 2 are used by receivers to decrypt

* Idea behind encrypting or using cryptographic techniques is delaying hacking. It can never be prevented.

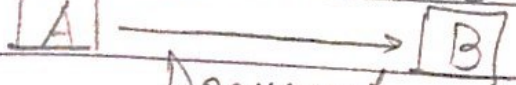
* Digital Signatures (example of Public Key Cryptography)



Puffin
Date
Page

* To Detect TAMPERING

eg Suppose A sends a doc. to B & want it to be secure



So, we add a Hash (Message Digest) to the document

Doc. + Hash ⟶ Fixed value.
(Run one-way fn on doc)
We get → Data H

* Concepts in Cipher

✓ Diffusion
Take uncoded symbol & replace it with some value.

✓ Confusion
With some pattern or sense, rearrange my sequence that was sent : ABCDE ⟶ CEDBA

* Operations :-

① Substitution

② Transposition / Permutation

③ XOR

✱ Block Ciphers and Stream Ciphers.

operates on blocks of data

operates on 1 bit of plain text at a time

✱ Considering the case of any type of info. in form of encryption, the lifetime is decided (i.e, how long should it take to break the code, or, how hard is decryption) and the key lengths are also decided similarly.

✱ DES (Data Encryption Standard)

Algorithm: Same for encryption & decryption

✱ It has keys → sort of header that's added at each stage for better encryption

We have bits (say 64 bits → 1 to 64) So, now permute it

eg:

1	2	3	Initial Permutn	5	9	7
4	5	6		2	8	1
7	9	8		3	6	4

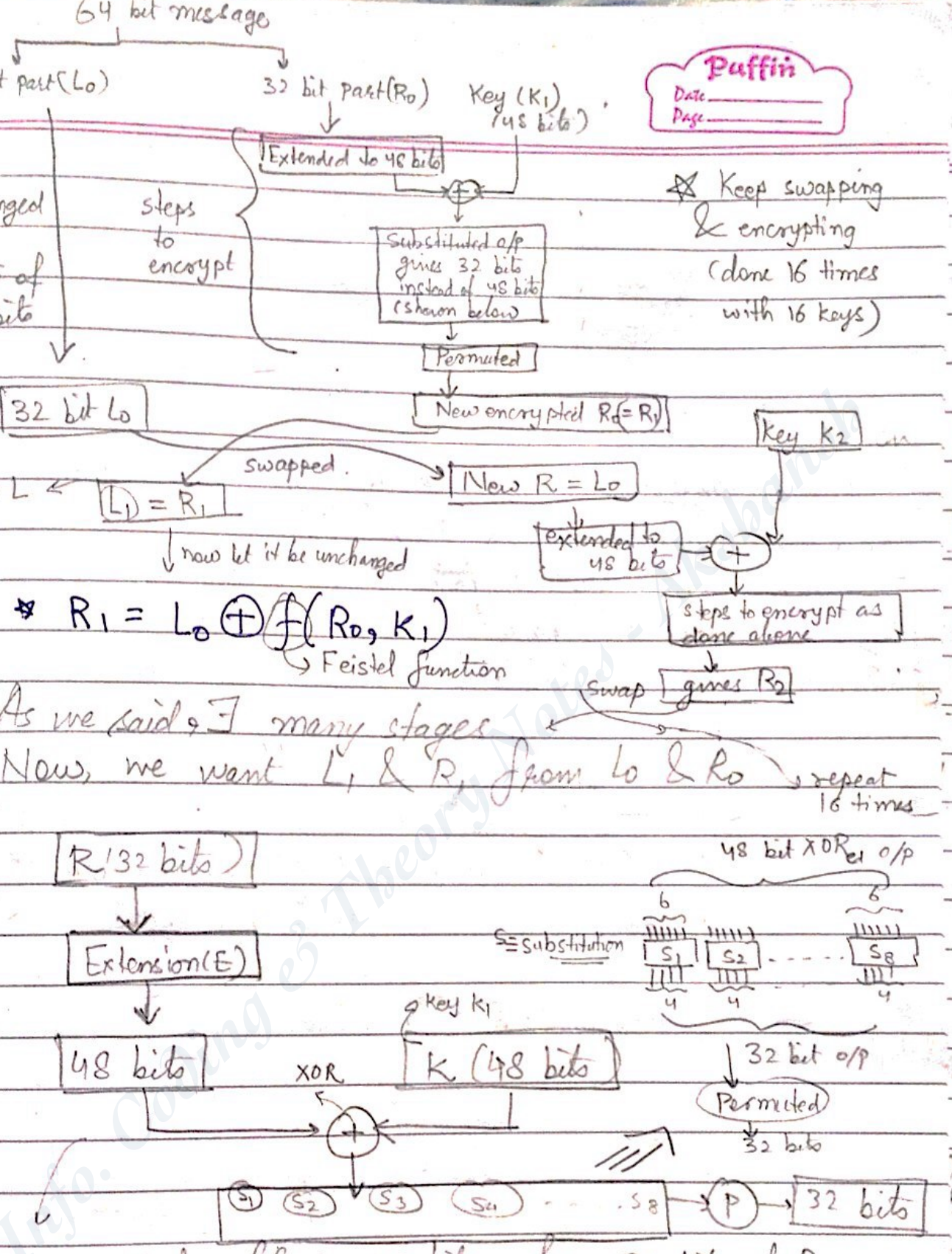
The permutn & inverse permutn is done in such a manner s.t ∃ some pattern, &, we can get original bits back (in sequence)

Once I do that, break 64 bits into 2 chunks

- One 32 bit part = L_0
- Other 32 bit part = R_0

A 64 bit message has to be encrypted. So, it divided into two 32 bit parts → L_0, R_0.

Now, R_0 is encrypted & combined with L_0. Then, they & swapped & L_0 is encrypted...



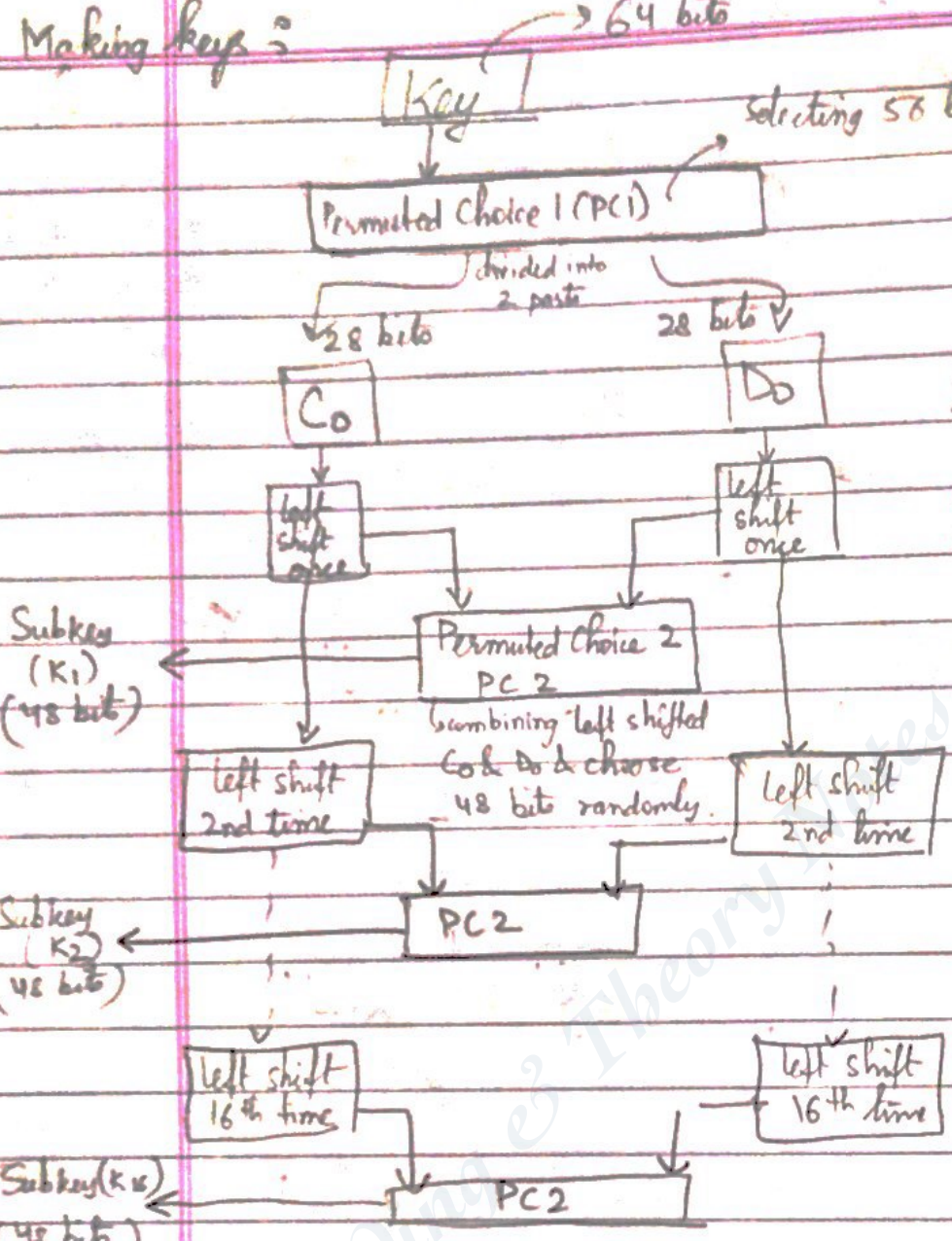
We want to obtain 48 bits, from 32 bits of R_0. (So, using these 32 bits, duplicate a few values & make a 48 bit matrix)

Now, say we have 16 layers (comparatively more secure). So, dividing each key (i.e. keep doing the above process for keys $K_1, K_2, \ldots, K_{16}$)

PTO

Puffin
Date
Page

Making keys :



* Getting C_0 & D_0 :
We have 56 bits in key. Divide in 2 parts & so, one part C_0 & other part is D_0 (each 28 bits)

* From prev. page, we saw swapping & encrypting oper^n is done 16 times. This requires 16 keys $K_1, k_2, \ldots k_{16}$ which are generated as shown in fig.

eg: for code = 0111101

Club 1st & last bit.

01 & (1110) > Clubbing others.

= 1 = 14

= row entry = column entry.

* 3 permut^n matrix.

Note :- $a \equiv b \pmod{c} \Rightarrow \frac{a-b}{c} = \text{integer}$

✡ RSA ALGORITHM

Theory : easy to multiply two large prime numbers but extremely hard (time consuming) to obtain prime factors from product.

eg : Given : $2^{113} - 1$,

Getting : $2^{113} - 1 = 3391 \times 23279 \times 65993 \times 1868569 \times 1066818132868207$.

Observation :

- Hamming code (7,4) - cyclic code
- Generator poly : $x^7 - 1$

Algorithm :

Steps

- $N = A \times B$
  - A, B : very large prime nos., randomly chosen, of equal length.
- Define : $T = (A-1) \times (B-1)$.
- Choose E (public key)
  - ↳ Randomly chosen
  - ↳ has no common factors with T.
- D is defined. Its private key.
  $D = E^{-1} \pmod{T}$ — should know public key
- Encryption is done : $C = M^E \pmod{N}$
- Decryption at receiver : $M = C^D \pmod{N}$ — should know private key

✡ • Neither transmitter, nor receiver has to know A & B.

eg :- A = 37, B = 23

N = 37 × 23 = 851 (known to both sender & receiver)

So, by algorithm (Built in)

$T = (A-1)(B-1)$

$= (36)(22) = 792$

Now, assume E = public key = 5 (has no common factor with T.)

Now, $D \equiv 5^{-1} \pmod{792}$ (⇒ congruent to)

$\Rightarrow 5D \equiv 1 \pmod{792}$

i.e, $\frac{5D-1}{792}$ = integer.

Assume, M = "G" = 7 (7th char. in alphabet)

- Encryption :

$C \equiv 7^5 \pmod{851}$

$\Rightarrow \frac{C - 7^5}{851}$ = integer = 19

So, C = 638

- Decryption $M \equiv 638^{317} \pmod{851}$ (317 → value of D)

So, M = 7

Now, finding D

$\frac{5D-1}{792}$ = integer

↳ taking D = 317, we get an integer = 2

I am encrypting "7" to 638.

Then, decrypting 638 back to 7. ✓

Parameter	Sender	Receiver	Hacker
A	K	U	U
B	K	U	U
N	K	K	K
$T = (A-1)(B-1)$	U	U	U
E	K (Public key)	K	K
$D = E^{-1} \bmod T$	U (not needed)	K (Private key)	U (E is known, but not T)
M	K	U	U
C	K	K	K

→ U : Unknown
→ K : Known

DIFFIE-HELLMAN KEY ALGORITHM

- protocol allows users to exchange a secret key in a secure medium.

Setup

Suppose 2 people wish to communicate (Alice & Bob).
They don't want Eve (an eavesdropper) to know their message.
Alice & Bob decide on public numbers q & p
p : prime no., q : primitive root mod p.

$\Rightarrow$ (q, p) are relatively prime
& $q^n \pmod p \equiv 1 \pmod p$
(discussed later)

Date
Page

Exchange

1. Alice chooses a random no. a & computes $U \equiv g^a \pmod p$ & sends u to Bob
2. Bob chooses a random no. b & computes $V \equiv g^b \pmod p$ & sends V to Alice
3. Bob computes the key $k \equiv U^b \equiv (g^a)^b \pmod p$
4. Alice computes the key $k \equiv V^a \equiv (g^b)^a \pmod p$

So, now, both Alice and Bob have the same key, namely, $k = g^{ab} \pmod p$

> Notes: From Number Theory:
> ① $a \equiv b \pmod c$ means a is congruent to b, mod c
> i.e $\frac{a-b}{c}$ is an integer
>
> ② Also, if $a \equiv b \pmod c$
> $\rightarrow a^x \equiv b^x \pmod c$

eg Say, Alice & Bob choose/agree to use $p = 47$ & $g = 5$. ($g.c.d(p, g) = 1$)

1. Alice chooses a no. b/w 0 & 46.
   Say $a = 18$
2. Bob chooses a no. b/w 0 & 46
   Say $b = 22$
3. Alice computes $U \equiv 5^{18} \pmod{47}$
   i.e $\frac{U - 5^{18}}{47}$ should be integer.
   So, choose u appropriately

If $U = 2$

$\Rightarrow \frac{2 - 5^{18}}{47} \in \mathbb{Z}$

So, $U = 2$.

4. Bob computes $V \equiv 5^{22} \pmod{47}$

This gives $V = 28$.

5. Alice wants to find k.

He received $V = 28$ from Bob.

So, using his private no., he raises it to a value $a = 18$ s.t

So, $(V)^a \pmod{47}$

i.e $(28)^{18} \pmod{47} = k$

6. Illy, Bob finds k by raising u to power b s.t we find

$(U)^b \pmod{47}$

$\Rightarrow (2)^{22} \pmod{47} = k$

So, both Alice & Bob got k.

☆ What constitutes a primitive root?

eg: Say $p = 7$, $g = 3$ (1): p & g should be relatively prime

(2) Now, we start finding

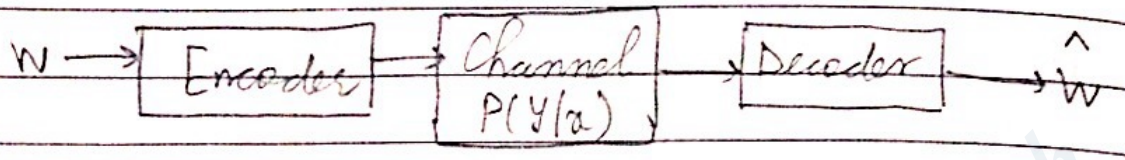
$3^1 \pmod{7} = 3$
$3^2 \pmod{7} = 2$
$3^3 \pmod{7} = 6$
$3^4 \pmod{7} = 4$
$3^5 \pmod{7} = 5$
$3^6 \pmod{7} = 1$

do till $p-1$ $(= 6)$
We should get all values from 1 to $(p-1)$ in these 6 possibilities.

If we get remainder = 1, then g is primitive root. So, $g = 3$.

Revision + Extra Concepts.

✱ Channel capacities for different types of channels :-



- For Discrete memoryless channel (DMC)

$$C = \max_{P(x)} I(X;Y)$$

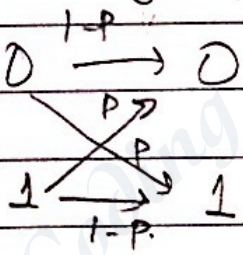
- For Noiseless Binary Channel

Give → Get $\quad C = H(P)$

0 → 0

1 → 1

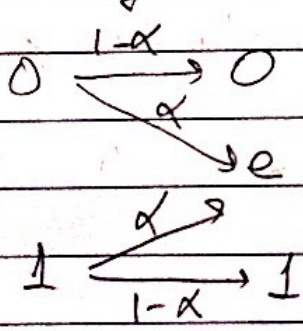
- For Binary Symmetric Channel (BSC)



$C = 1 - H(P)$ bits

✓ P : Probability of error

- For Binary Erasure Channel



$C = 1 - \alpha$

- For symmetric channel

eg $P(y/x) = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.5 & 0.3 & 0.2 \\ 0.2 & 0.5 & 0.3 \end{bmatrix}$ [3 values, 0.3, 0.2 & 0.5 symmetrically put]

- Weakly Symmetric Channel

$$P(y|x) = \begin{bmatrix} 1/3 & 1/6 & 1/2 \\ 1/3 & 1/2 & 1/6 \end{bmatrix}$$

∵ Not much symmetry & If we add columns, each column gives same probability.

- Symmetric channel is weakly symmetric
  Vice versa not true

So, here, $C = \log|y| - H\left(\frac{1}{3}, \frac{1}{2}, \frac{1}{6}\right)$

↳ max → P(x) uniform

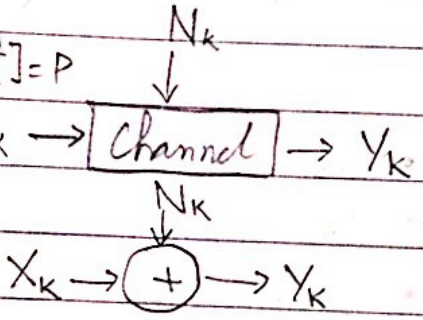
* Properties

P1. $C \geq 0$

P2. $C \leq \log|X|$

P3. $C \leq \log|y|$

$E[X_k^2] = P$



$$C = \max_{f_{X_k}(x)} \left[I(X;Y) \,\middle|\, E[X_k^2] = P \right]$$

↳ under Gaussian channel :-

$$C = W \log_2\left[1 + \frac{P_0}{N_0 W}\right] \text{ bps}$$

↳ W : BW of channel.

* We have seen,

$$\text{Rate, } R = \frac{\log_2 M}{n}$$

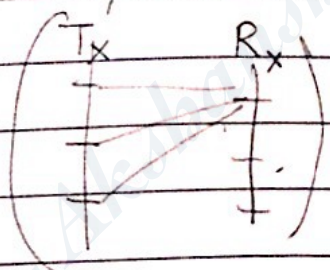
$$= \underbrace{\frac{1}{2}\log_2\left(1 + \frac{P}{\sigma^2}\right)}_{\text{bits per channel use}} = \frac{K}{n}$$

In MIMO

↳ ⇒ Transmitting & receiving antenna is not single. I assay of antennas. (In case one channel cant deliver, other will deliver)

$$C = W(M)\log_{\beta 2}\left(1 + \frac{E_s}{N_0}\right) \text{ bps}$$

M → no. of channels per array



- **Word** : Sequence of symbols
  like :- 00 01 10 11
- **Code or Codebook** : No./set of codewords.
- **Hamming weight** : no. of 1's in a CW.
- **Hamming distance** : Distance b/w any 2 CWs taken. eg: 0111 & 1101
  difference = 2.
  Transmitting through P = 16,

eg If C = [00000, 10100, 11110, 11001]

n = block length = 5 (n, k)

Uncoded bits	CWs
00	00000
01	10100
10	11110
11	11001

M = 4, K = 2, n = 5

Code rate, $R = \frac{k}{n} = 0.4$

↳ $k = \log_2 M$

In the given CWs, CW = 2

★ Linear Code:
(i) Sum of 2 CWs is a CW
(ii) All zero CWs is a CW
(iii) $d_{min} (= d') = W_{min} (= W')$
(min distance = min weight)

★ Field:
(i) If $a+b \in Z$
$a, b \in Z$
(ii) Commutative laws
(iii) Associative laws
(iv) Distributive laws
(v) $a+0 = 0$ (existence of identity)
(vi) $a \cdot 1 = a$ (Multiplicative identity)
(vii) Additive inverse
(viii) Multiplicative inverse
(ix) If a field has finite no. of elements its called Galva field : GF(q)

★ • Creating a linear code.

eg ★ Given: S = {1100, 0100, 0011} as vectors in GF(4)

As per the properties of linear code, Sum of 2 CWs is a CW.

$$\begin{array}{r} 1100 \\ +0100 \\ \hline 1000 \end{array} \rightarrow \notin S.$$

Also, all zero CW (0000) $\notin$ S. So, its not satisfying

Now, creating linear code

$C = \langle S \rangle$

C → linear code; $\langle S \rangle$ → span of S (i.e., set of all elements of S)

Now, start satisfying all properties of linear code.

(1) Sum of 2 cw should be a cw

1100	1100	0100
+0100	+0011	0011
1000	1111	0111

add these 3 elements to S.

(2) Add 0000 to set S.

So, set S = {1100, 0100, 0011, 1000, 1111, 0111, 0000}

Now, its linear.

(Also, check $d' = w'$ → true here)

★ Mathematically generating code :-

As done before :

$C = i G$

$G = [I \mid P]$ → Partition matrix

$H = [-P^T \mid I]$

→ Transponse of parity check matrix.

If $H\underline{c} = 0 \Rightarrow$ no error

$\neq 0 \Rightarrow$ error (bit with error can also be identified)

Date
Page

* Systematic form: $G = [I \mid P]$

eg $G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$

Clearly, it's not in systematic form

Suppose for one (u_2, u_3, u_4) do:

$C = iG = [0001]G$

$\Rightarrow C = [0001\ 101]$

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

Now $H \times C = H \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ → $\Rightarrow$ no error.

Now, change C to $[0000\ 101]$

Now

$H \times C = H \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \neq 0 \Rightarrow$ error

→ It's the 4th column of H matrix.

So, 4th bit is an error.

* No. of errors detected $(d^* - 1)$
* No. of errors corrected $(d^* - 2)$

↳ $d^* = d_{min}$

↳ $d^* \leq n - k + 1$; $n = 7, k = 4$ (from G matrix)

So, d^* can be found

d_{max} can be found

- **Coset vectors**
  - ↳ used to determine P(error)

eg: Say, $G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

& $C = \{000, 010, 101, 111\}$

& cosets are $\{000, 001\}$

↳ ⇒ If any of coset vector is added with C → those sum gives entire set

↳ add 000 to C :

$\{000, 010, 101, 111\}$

↳ add 001 to C :

$\{001, 011, 100, 110\}$

$\left\{\begin{matrix} 000 & 100 \\ 001 & 101 \\ 010 & 110 \\ 011 & 111 \end{matrix}\right\}$ ← Union of these gives all combinⁿ of 3 bits (i.e, entire set)

eg (2) Given: $C = \{0000, 1011, 0101, 1110\}$

& coset vectors/leaders = $\{0000, 1000, 0100, 0010\}$

Now,

$$P(\text{error}) = P_{err} = 1 - \sum_{i=0}^{n} \alpha_i \, P^i (1-P)^{n-i}$$

↳ n = no. of bits = 4, here.

↳ P : Prob. of 1 bit in error.

↳ P = 0.1, (say)

↳ α_i = related to coset leaders :

↳ α_0 ⇒ no. of 1's in coset vectors is zero. ⇒ $\alpha_0 = 1$

↳ $\alpha_1 = 3, \alpha_2 = 0, \alpha_3 = 0 \& \alpha_4 = 0$

After solving for the values, we get

$P_{err} = 0.0523$

- Cyclic Codes

Requirements: ① C should be a linear code.
② Any cyclic shift of a CW is also a CW.

eg:

If $C_1 = \{0000, 0101, 1010, 1111\}$

Requirement ①: It satisfied (do like before)
② doing cyclic shift:

$0101 = \underbrace{1010}_{\in C}$

After checking ∀ CWs, we find, its cyclic

eg ②:- If $C_2 = \{0000, 0110, 1001, 1111\}$

① : Satisfied. Can be checked

②: $0110 = 0011 \notin C$

So, its not cyclic code

* Cyclic polynomials:-

$f(x) = f_0 + f_1 x + \dots + f_m x^m$

↳ $f_i \in GF(q)$

(i.e. if $q=2$, $f_i = 0, 1$
3, $f_i = 0, 1, 2$)

eg) $f(x) = 2 + x + x^2 + 2x^4$
$g(x) = 1 + 2x^2 + 2x^4 + x^5$ } GF(3)

⇒ $1+1=2$
$2+1=0$
$2+2=1$

$f(x) + g(x)$

$$\begin{array}{r} 2 + x + x^2 + 2x^4 \\ 1 + 0.x + 2x^2 + 2.x^4 + x^5 \\ + \\ \hline 0 + 1.x + 0.x^2 + x^4 + x^5 \end{array}$$

$= x + x^4 + x^5$

Now,

If a polynomial can be written as:

$a(x) = q(x)\, b(x) + r(x)$

↳ $\deg(r(x)) < \deg(b(x))$

* Congruent modulo:

$g(x) \equiv h(x) \pmod{f(x)}$

$\Rightarrow g(x) - h(x)$ is divisible by $f(x)$

* Concept of Ring

eg If $f(x) = x^7 - 1$

$\Rightarrow x^7 - 1 = (x-1)(x^3+x+1)(x^3+x^2+1)$

let $g(x) = 1 + x + x^3$ (in ascending order)

& $h(x) = (x-1)(x^3+x^2+1)$

Generator matrix, G =

Const.	x	x^2	x^3	x^4	x^5	x^6	
1	1	0	1	0	0	0	
0	1	1	0	1	0	0	
0	0	1	1	0	1	0	
0	0	0	1	1	0	1	

$n = 7$

$n - k = 4$ ↓ $\deg(g(x))$

Now, $C = i G$

i: $n-k$; G: $(n-k) \times n$

$\Rightarrow h(x) = x^4 + x^2 + x + 1$ (in descending order)

$H =$

1	0	1	1	1	0	0
0	1	0	1	1	1	0
0	0	1	0	1	1	1

$n = 7$

$n - k = 3$ ↓ $\deg(h(x))$

* CRC

$P(x) = x^5 + x^4 + x^2 + 1 \quad \leftrightarrow 110101$

$D(x) = x^9 + x^7 + x^3 + x^2 + 1 \quad \leftrightarrow 1010001101$

Now, $T(x) = x^{n-k} D(x) + R(x)$

→ ⇒ shifting data

Now, $= x^5 D(x) + R(x)$

$P(x)$ { $x^5 D(x)$ {

$= P(x)$ } $x^{14} + x^{12} + x^8 + x^7 + x^5$ {

$x^9 + x^8 + x^6 + x^4 + x^2 + x$

So do: $x^5 + x^4 + x^2 + 1$) $x^{14} + x^{12} + x^8 + x^7 + x^5$ (

$x^{14} + \quad + x^{13} + x^{11} + x^9$

$x^{13} + x^{12} + x^{11} + x^9 + x^8 + x^7 + x^5$

$x^{13} + x^{12} \quad + x^8 \quad + x^{10}$

$x^{11} + x^{10} + x^9 + x^7 + x^5$

$x^{11} + x^{10} + \quad + x^8 + x^6$

Note:
Sum = difference
(∵ its modulo 2)

$x^9 + x^8 + x^7 + x^6 + x^5$

$x^9 + x^8 + \quad + x^6 \quad + x^4$

$x^7 + x^5 + x^4$

$x^7 \quad + x^4 + x^3 + x^2$

$x^6 + x^5 + x^2$

$x^6 + x^5 \quad + x^3 + x$

$R(x) \leftarrow x^3 + x^2 + x$

Now

$T(x) = x^5 D(x) + R(x)$

$= (x^{14} + x^{12} + x^8 + x^7 + x^5) + (x^3 + x^2 + x)$

$= x^{14} + x^{12} + x^8 + x^7 + x^5 + x^3 + x^2 + x$

$1.x^{14} + 0.x^{13} + 1.x^{12} + 0.x^{11} + 0.x^{10} + 0.x^9 + 0.x^8 + 1.x^7$
$+ 0.x^6 + 1.x^5 + 0.x^4 + 1.x^3 + 1.x^2 + 1.x + 0$

$\Rightarrow 1010000101011.10 = T(x)$

Ans

☆ Polynomial Multiplicⁿ :

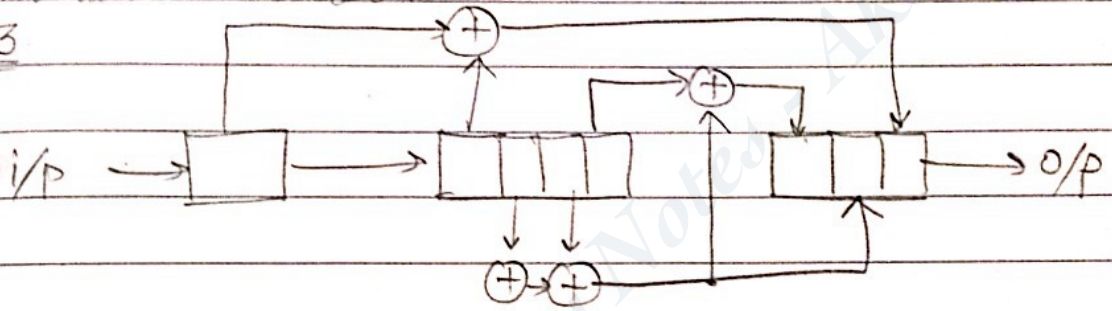
$a(x) \rightarrow [\ g(x)\ ] \rightarrow b(x)$

FIR Filter

$$b_n = \sum_{k=0}^{M} a_k g_{n-k}$$

☆ Convolution Encoder.

Q. 6.3



i/p — current/next state — o/p

i/p	CS	NS	o/p
0	0000	0000	000
1	0000	1000	001
0	0001	0000	100
1	0001	1000	101
0	0010	0001	110
1	0010	1001	111
0	0011	0001	010
1	0011	1001	011
0	0100		
1	0100		
	⋮	⋮	⋮
0	1111		
1	1111		

I am making a table showing that for any time if CS & i/p are given, what will be o/p & NS.

Puffin
Date
Page

Understanding RSA ALGORITHM

We will talk about conversations between Alice and Bob. In it, Alice sends an encrypted message through public channel. Bob receives this message & adds his private message (to be understood only by Alice) When this combined message is received by Alice, she removes her encryption (decrypts the message) to get Bob's private message to her.

* Theory used :

① Prime factoriz^n of very large numbers takes ages to solve. So, taken two large prime numbers & multiplied gives a no. (N) which cannot be easily broken down by any outsider (Eve) unless the factors are known.

② By Euler's theorem,

$$m^{\phi(N)} \equiv 1 \pmod N \; ; m : \text{any no.}$$

Also, if N = prime

$$\phi(N) = N-1$$

Moreover, consider 2 prime nos. P_1 & P_2

s.t $P_1 \times P_2 = N$

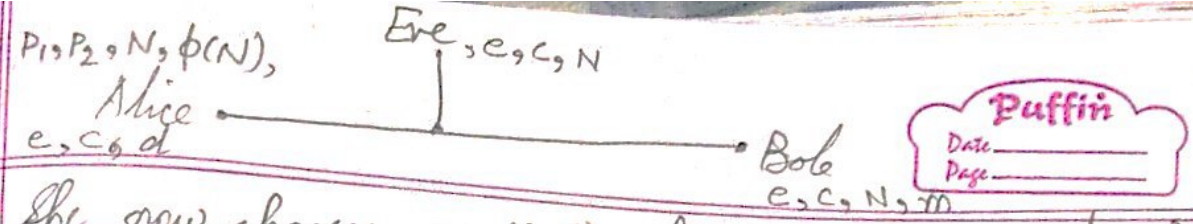
Then, $\phi(N) = \phi(P_1 \times P_2) = \phi(P_1)\,\phi(P_2)$

$$\Rightarrow \phi(N) = (P_1-1)(P_2-1)$$

* Sequence of steps :

Alice chooses 2 large prime nos. say P_1 & P_2 & multiplies them to get $N = P_1 \times P_2 \rightarrow ①$

So, $\phi(N) = (P_1-1)(P_2-1)$.



She now chooses no. "e" relatively prime to $\phi(N)$
i.e, $g.c.d(e, \phi(N)) = 1$
Now, this no. e & the no. N are publically displayed both to Eve & Bob
On receiving e & N, Bob chooses any information he wants to send to Alice, say m.
Then, he does

$$m^e \equiv ? \pmod N \equiv c \pmod N$$

This residue (remainder) = c is sent publically through channel reaching Alice (& Eve)
Now, if we see, Bob has sent $c (\equiv m^e)$
If Alice can find some exponent to remove her encryption (i.e e) :
something like: $(m^e)^{1/e}$, she can get m
This decrypting exponent is called d.
From theory:

$$m^{\phi(N)} \equiv 1 \pmod N$$
$$\Rightarrow m^{k\phi(N)} \equiv 1^k \pmod N \equiv 1 \pmod N$$
$$\Rightarrow m[m^{k\phi(N)}] \equiv 1 \times m \pmod N$$
$$\Rightarrow m^{k\phi(N)+1} \equiv m \pmod N$$
$$\Rightarrow m^{e\left[\frac{k\phi(N)+1}{e}\right]} \equiv m \pmod N$$

↳ this term $\frac{k\phi(N)+1}{e} = d$.

$$\text{So, } m^{ed} \pmod N \equiv m \pmod N$$

So, basically, Alice knows e & d. When she receives c from Bob, she finds

$$c^d \equiv (?) \pmod N$$

↳ This is message sent by Bob ✓

eg: Suppose Alice chooses:

$p_1 = 53$, $p_2 = 59$.

$\Rightarrow N = p_1 p_2 = 3127$

& $\phi(N) = (p_1 - 1)(p_2 - 1) = 52(58) = 3016$.

Now, she finds e (s.t $g.cd(e, \phi(N)) = 1$)

let $e = 3$

So, $d = \frac{k\,\phi(N) + 1}{e} = \frac{k\,(3016) + 1}{3}$.

let $k = 2$, for d to be integer $\Rightarrow d = 2011$

So, $e = 3$, $d = 2011$

Now, these values of e & N are shared publically.

Suppose Bob chooses to send $m = 89$.

So,

$$89^e \pmod N = 89^3 \pmod{3127} \equiv 1394 = c.$$

Alice gets this c (= 1394) & does

$$1394^d \pmod N = 1394^{2011} \pmod{3127}$$

$1394^{2011} \pmod{3127}$ gives remainder (residue) = 89.

So, Alice deciphered the message sent by Bob!

Diffie-Hellman

Eve: P = 17, g = 3, Alice's message = 6, Bob's message = 12

Alice

prime modulus = 17

generator = 3

a = 15

$3^{15} \pmod{17} \equiv 6$

Bob's message = 12

Alice's private no.

(Bob's msg) $\pmod{P} \equiv 10$

$3^{15} \pmod{17} \equiv ?$ → easy to find (by Alice)

$3^{?} \pmod{17} \equiv 6$ → hard to find (by Eve)

Bob

P = 17

g = 3

b = 13

Alice's message = 6

$3^{13} \pmod{17} \equiv 12$

Bob's private no.

(Alice's Msg) $\pmod{P} \equiv 10$

RSA

Eve: e, N, c

Alice

encryption exponent (e)

$(\underset{m}{\text{message}})^{e} \pmod{N} \equiv c$

decryption exponent (d)

$c^{d} \pmod{N} \equiv m$

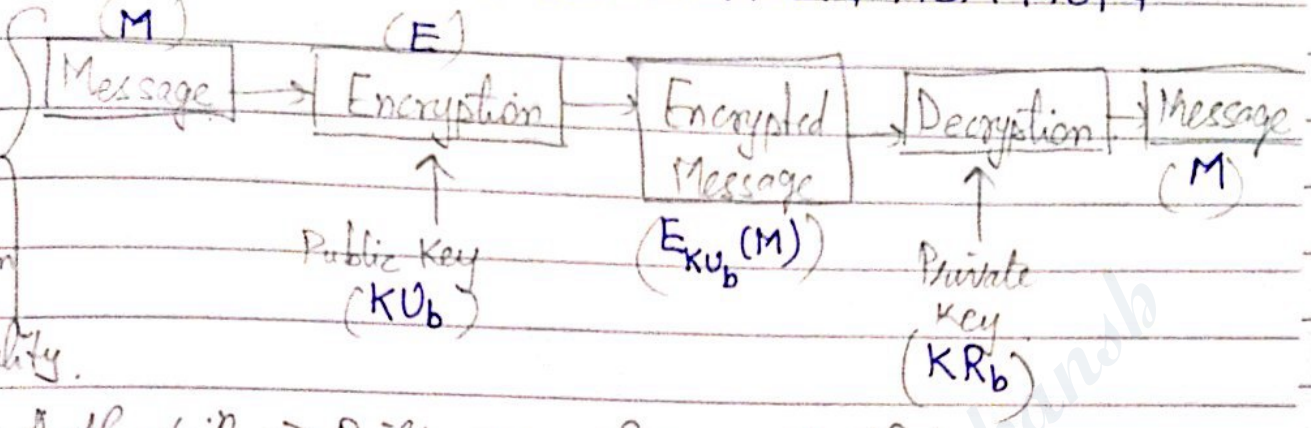
$\Rightarrow ed = k\phi(N) + 1 \Rightarrow d = \frac{k\phi(N)+1}{e}$

$m^{ed} \pmod{N} \equiv m \rightarrow m^{k\phi(N)+1} \pmod{N} \equiv m \pmod{N}$

Bob

e, N, c

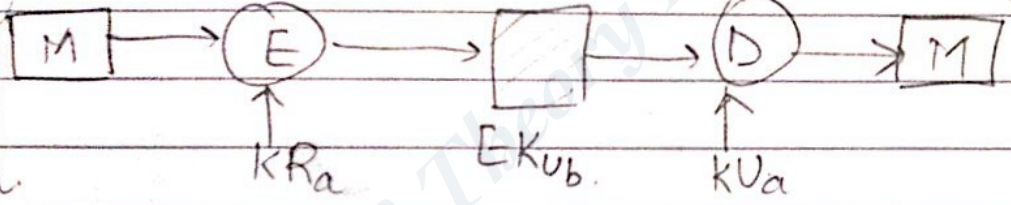
CONFIDENTIALITY & AUTHENTICATION



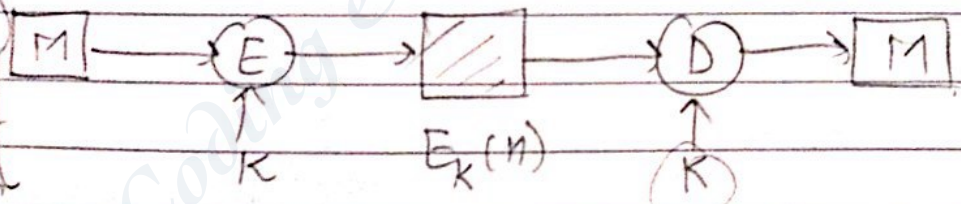
* Public key encryption → Confidentiality.

- Authentic^{n} ⇒ Right person has sent that message or not.
- Confidentiality ⇒ Right person receiving the message

* Public key encryption :- If sender sends through private key & decryption is done through public key.
  ↳ Authentic
  ↳ Signature



Symmetric encryption :-
↳ Confidentiality
↳ Authentic



→ only sender & receiver know this key.

* General Idea :-
  Confidentiality ≡ Receiver
  Authentic^{n} / Signature ≡ Sender

* Symmetric key encryption ≡ DES
* Public key encryption + private key decryption ≡ RSA.

Public key encryption: (Confidentiality, authentic & signature)

↳ 3 private key both for sender & receiver

* Hash function (h)

$h = h(M)$

↳ M : message

* If a message is hashed, we cant get message, M (i.e, one way property)

i.e, $M \longrightarrow H(M)$

then $M \not\longleftarrow H(M)$ (Computationally infeasible)

* Every message has separate mapping to hash fn

i.e

$M_1 \longrightarrow H(M_1)$ → doesn't happen

$M_2 \longrightarrow H(M_2)$ (Many one ✗)

$M_1 \longrightarrow H(M_1)$ }

$M_2 \longrightarrow H(M_2)$ } ✓

* Applic^ns:

- TLS (Transport layer Security)
- Web browsing.

* Internet Engineering Task Force : IETF

Puffin
Date
Page

* TLS/SSL

- ✓ Secure Sockets Layer (SSL): predecessor of Transport Layer Security (TLS)
- ✓ Are part of Layer 5 : Session Layer of OSI.
- ✓ For communicⁿ security over Internet.
- ✓ Support X.509 Certificate.

* HTTPS ⇒ HTTP sitting on top of TLS/SSL.

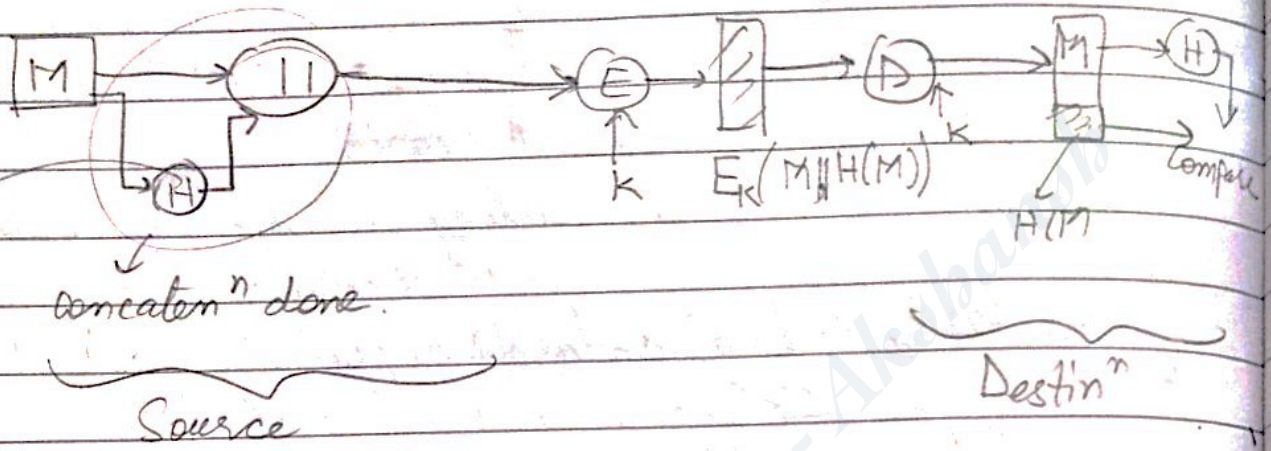
* Protocol & Standards supporting Security for X.509 Certificate.

→ IPSec.
- ✓ Layer 3 : OSI → Network layer
- ✓ Protocol for encrypting EACH IP packet (other side decrypts also)
- ✓ (Reduces BW)
- ✓ Techniques for implementing hash fⁿˢ:-
  ↳ HMAC, MD5 (↳ new), SHA-1 use IPSec

→ Secure Shell (SSH)
- ✓ Layer 7 : OSI → Applicⁿ
- ✓ Used to log in to a remote machine
- ✓ Supports tunneling.
- ✓ Used on Unix & Windows.
- ✓ A more secure version of rlogin/telnet

→ PGP: Pretty Good Privacy.
- ✓ For info & data security apart from communicⁿ security (like emails, files, disk partitions — encryption & decryption).
- ✓ Uses serial combinⁿˢ of hashing

* HASH Functions
  ↳ part of it has concatenation of message at i/p. Then, encryption ... are done



Different Techniques used:
* Why, for symmetric & other types of encryption.
* Other things that can be done - encrypting & decrypting value of hash fn
  Why, use of public key & private key for encrypting & decrypting respectively the hash value.
* We can also do encryption & decription of message as well as hash value.

Type of Questions on Hash fnc
1. Design a sys. that only provides confidentiality (C)
2. " " " authentic^n (A)
3. " " " signature (S)
4. " " " C & A
5. " " " A & S
6. " " " C & S
7. " " " C & 2 levels of A
8. " " " C & 2 levels of A & S

① Use public key encryption & private key decryption

(Block diagram as done before)

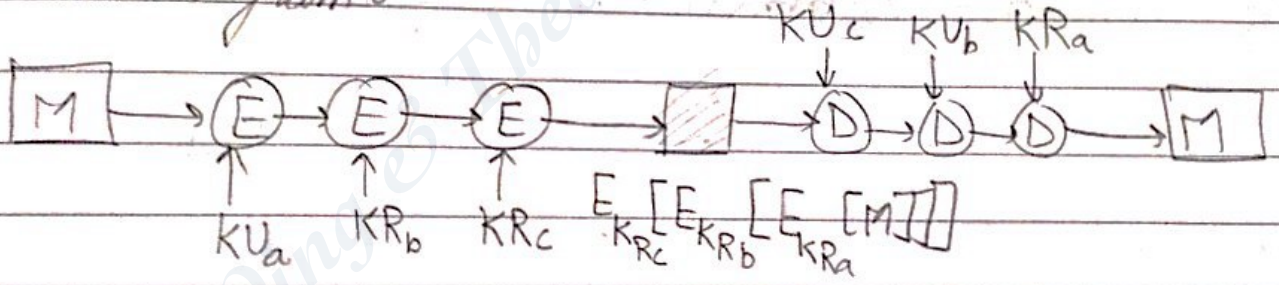
Note: Signature cannot be got WITHOUT authentic^n. So, ③ went happen.

⑧ If we get S, we automatically get A
So, using C & 2 levels of S will implement
C & 2 levels of S & A

Idea:-

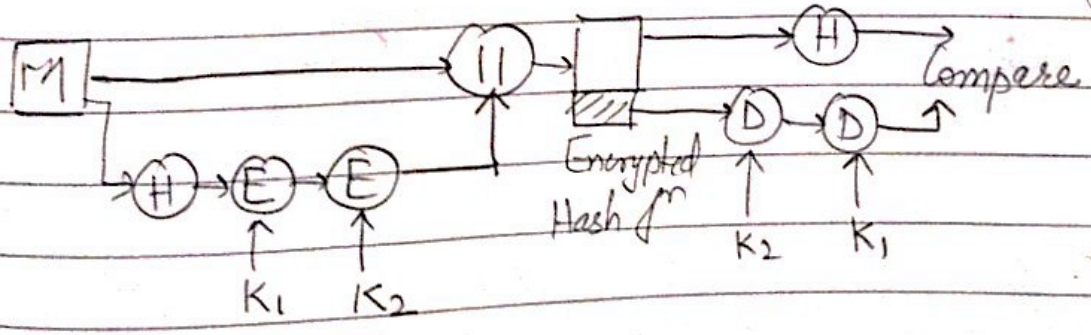
Use 2 private key encryptions & 1 public key
2 Signatures 1 Confidentiality

Block diagram:



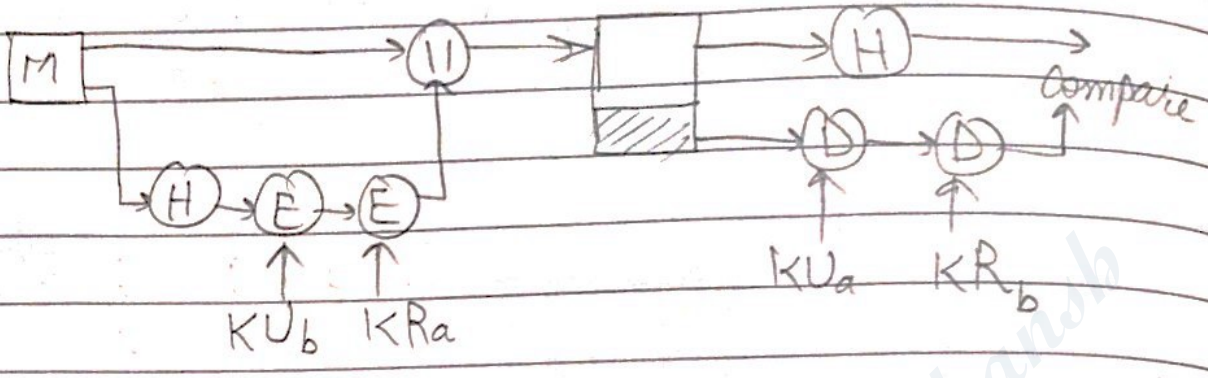
Encrypted last should be decrypted first. So, write the encrypted message accordingly.

⑨ Provides C & A using Hash fn.



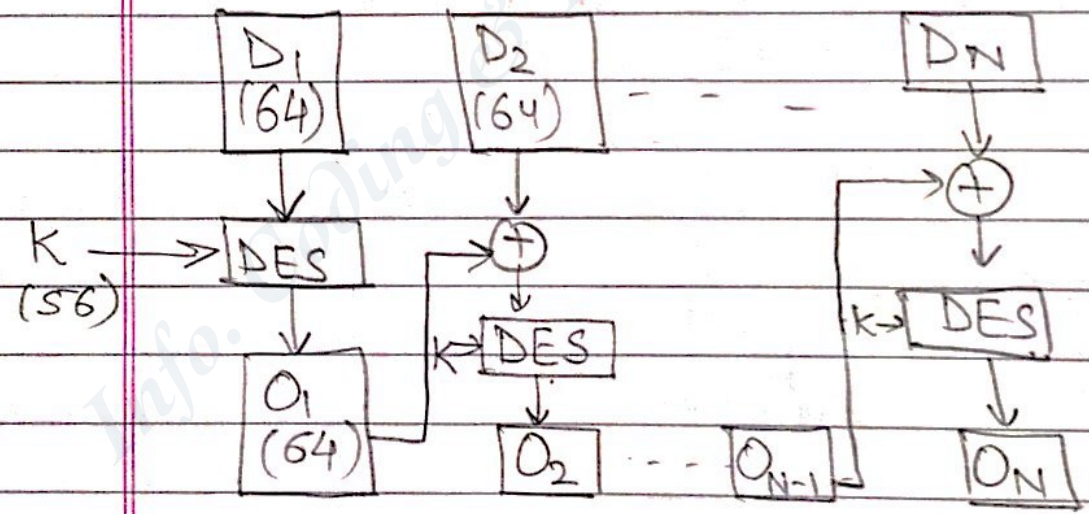
Idea: Use symmetric key

M2 : Use public & private keys.



- Sender & Receiver
  KU_b & KR_b : gives confidentiality
  KRa & KUa : gives authentication & signature
  (we didn't get only C & A)

✡ DATA AUTHENTICATION ALGORITHM (based on DES)



$O_1 = E(K, D_1)$

$O_2 = E(K, [D_2 \oplus O_1])$

$O_3 = E(K, [D_3 \oplus O_2])$

⋮

$O_N = E(K, [D_N \oplus O_{N-1}])$

* AES ≡ Advanced Version of DES

Puffin
Date
Page

* It is different from DES in the way, the key for every encryption is the same (K)

Suppose the decryption process:

$O_1 \rightarrow$ Decrypt $\rightarrow D_1$

$O_2 \rightarrow$ Decrypt $\rightarrow (D_2 \oplus O_1) \oplus O_1 \rightarrow D_2$

$(D_2 \oplus O_1) = 0$

Note: we get D_2 from O_2 ∵ we XOR O_1 (present in O_2) with O_1 (1st o/p originally got)

Now, if ∃ no change in the data i.e

not tampered : $(D_2 \oplus O_1) \oplus O_1 = D_2$

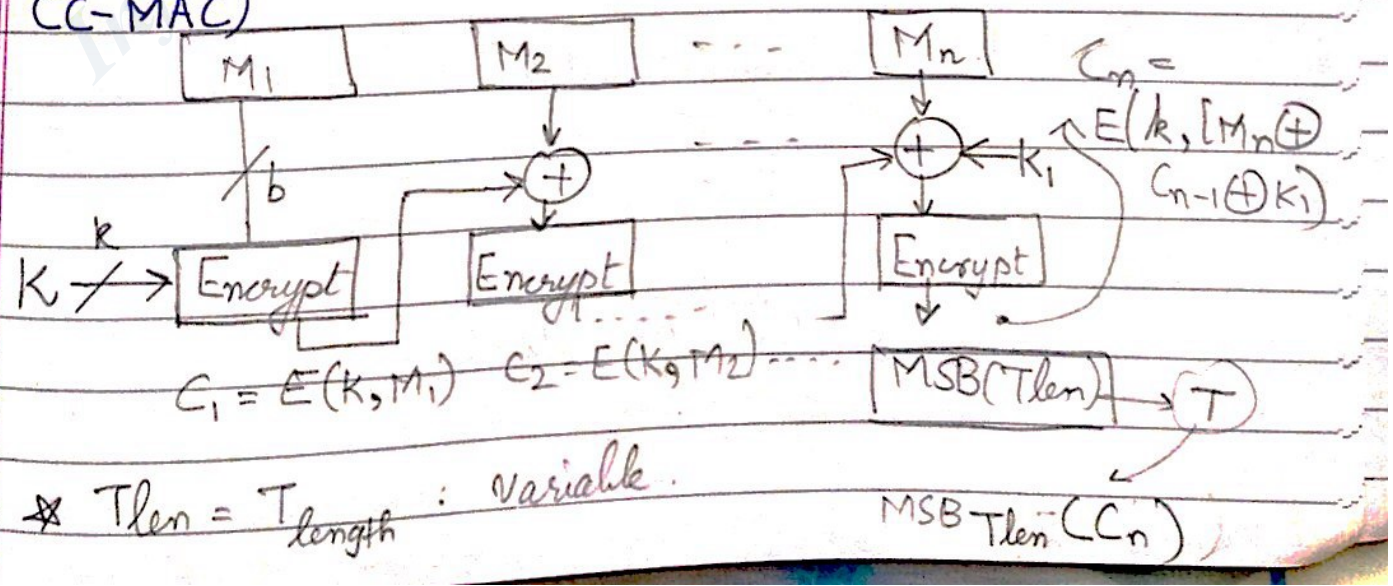
tampered : $(D_2 \oplus O_1) \oplus O_1 \neq D_2$

$= D_{2,t}$ tampered $(D_{2,t})$

* From prev. figure (K) only D_1 & D_2 are used (small sys).

$O_{1,t} \rightarrow D_{1,t} \rightarrow \underbrace{E(K, D_{1,t})}_{O_{1,t}} - E(K, [D_{2,t} \oplus O_{1,t}])$

* CIPHER BASED MESSAGE AUTHENTICATION CODE CC-MAC)



* Tlen = T_{length} : variable.

Encrypt

AES → b = 128, key size : 128, 192 or 256 bits

Triple DES → b = 64, key size : 112 or 168 bits

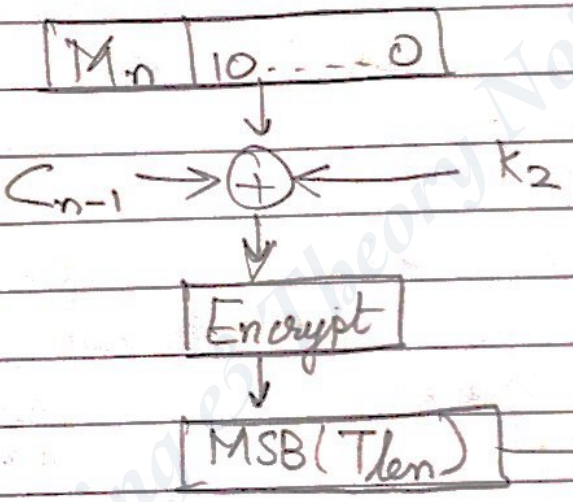
(b → block size)

* $L = E(K, 0^n)$

$K_1 = L \cdot x$

$K_2 = L \cdot x^2$ → Multiplic" in $GF(2^n)$

* If we need 128 bits, & we have 100 bits, then pad with 0.



→ DAA Algorithm :

* See how exactly data is tampered

$D_1 D_2 \ldots D_N | O_N$

↓ 64 bit

or $16 \leq M \leq 64$

Consider a block of data : 0101 1101 1001 0001

Now, XOR these values as :

```
  0101
  1101
  1001
⊕ 0001
  0000
```

→ This is my hash function, C_i

So, I send:

0101 1100 1001 0001 0000

Suppose ∃ some tampering

0101 1100 1001 0001 → 0001 1100 1001 0001

Taking XOR, we get

```
  0001
  1101
  1001
⊕ 0001
  0100
```

→ So, we get to know tampering happened.

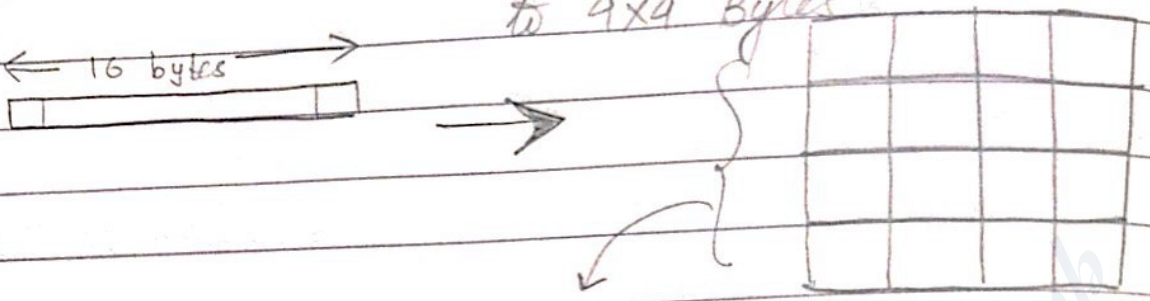
Advanced Encryption Standard

AES ALGORITHM

What happens with my data ?

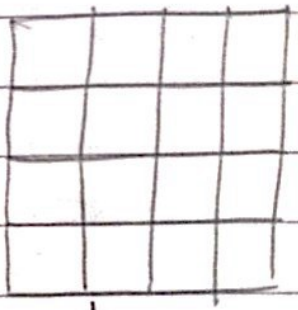
Key difference b/w AES & DES → a matrix. So, matrix operations are possible to be done on it

These 16 bytes are converted to 4×4 Bytes



Initial Transformation

No. of rounds	Key length (bytes)	Changed
10	16	4×4 matrix
12	24	
14	32	



no. of rounds depends on the key length (in DES, ∃ 16 rounds)

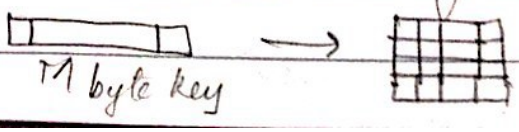
Transformation Round 1

Transformation will include:

- Substitution box (S-Box)
  - Does BYTE by BYTE Substitution
- Shift Rows → Permutation
- Mix Columns → GF(2^8)
- Add Round Key → XOR

This is also in matrix form.

✱ Key prepared in each round is also in matrix form



✡ PRETTY GOOD PRIVACY (PGP)

✡ NOTATIONS, as seen before:

K_s = Session key used in symmetric encryption scheme.

PR_a = Private key of User A.
(Public key encryption scheme)

PV_a = Public key of User A.
(Public key encryption scheme)

EP = Public key encryption.

DP = Public key decryption.

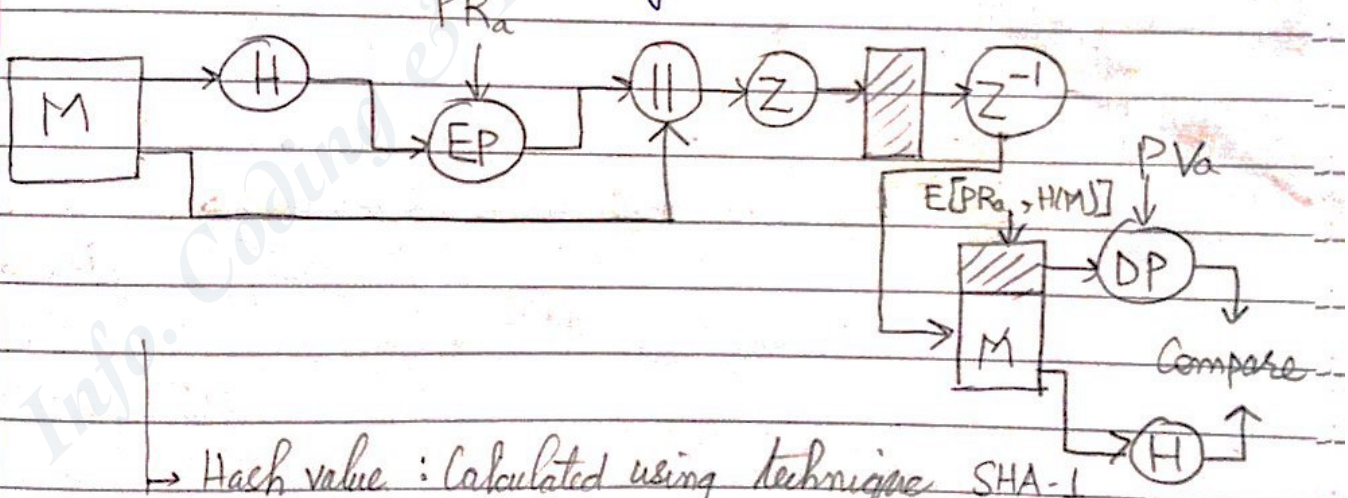
EC = Symmetric Encryption

DC = Symmetric Decryption.

H = Hash function.

|| = Concatenation.

Z = Compression using ZIP.

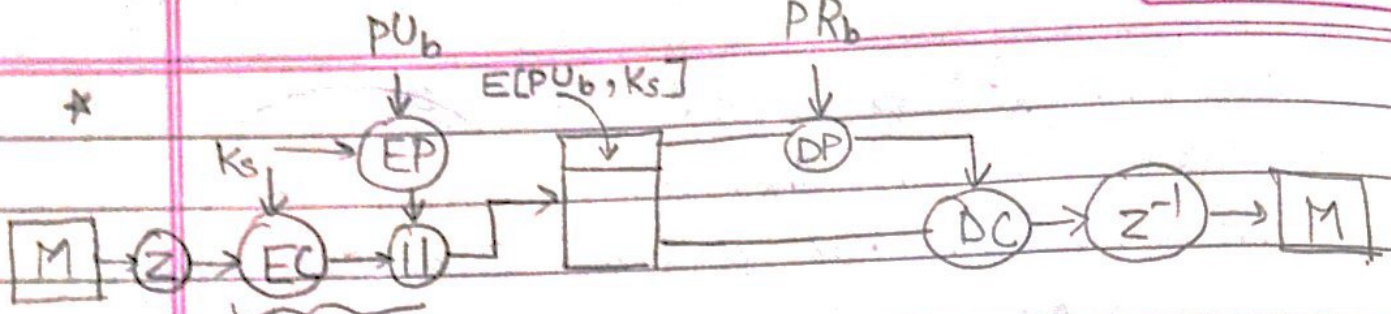


→ Hash value : Calculated using technique SHA-1
: 160 byte

Or Calculated using RSA, DSS Algorithms

✓ Only authentic^n is implemented in this scheme.
(No: we are encrypting only hash value, not complete message)

✓ Hash fn gives Data Authentic^n.



In above diagram
we have: Symmetric encryption & decryption
→ Symmetric key is encrypting with public key & is concatenated with compressed encrypted message

Key
Message

Now, K_s is decrypted using private key & message is then decrypted using it → finally getting message
↳ Similar is applied in Diffie Hellman algorithm.

Here, we have Authentic^n & Confidentiality.
✓ We don't have hash fn.
→ K_s: 128 bit key (PGP Standard)
→ Technique for EC: CAST-128 (or IDEA or 3DES)
↳ Triple DES
→ Technique for EP: RSA.

IEEE 802.11

↳ a committee for LAN standards
↳ setup in 1990's

Terminologies :

- ✓ Access point (AP)
- ✓ Basic service set (BSS)
- ✓ Coordination function.
- ✓ Distrib^n Sys (DS)
- ✓ Extended service unit (ESU)
- ✓ Protocol data Unit.
- ✓ Service data Unit
- ✓ Station

* Wireless Ethernet Compatibility Alliance (WECA) renamed to Wireless Fidelity Alliance (Wi-Fi)

* Protocol Architecture.

Logical Link Control
Medium Access Control
Physical

- Wired Equivalent Privacy (WEP)
-
- Robust Security Network (RSN)

- IP Address : Network layer (OSI)
- MAC Address : Data Link layer (OSI)
- Ports (Sockets) : Transport layer (OSI)

✡ RSN

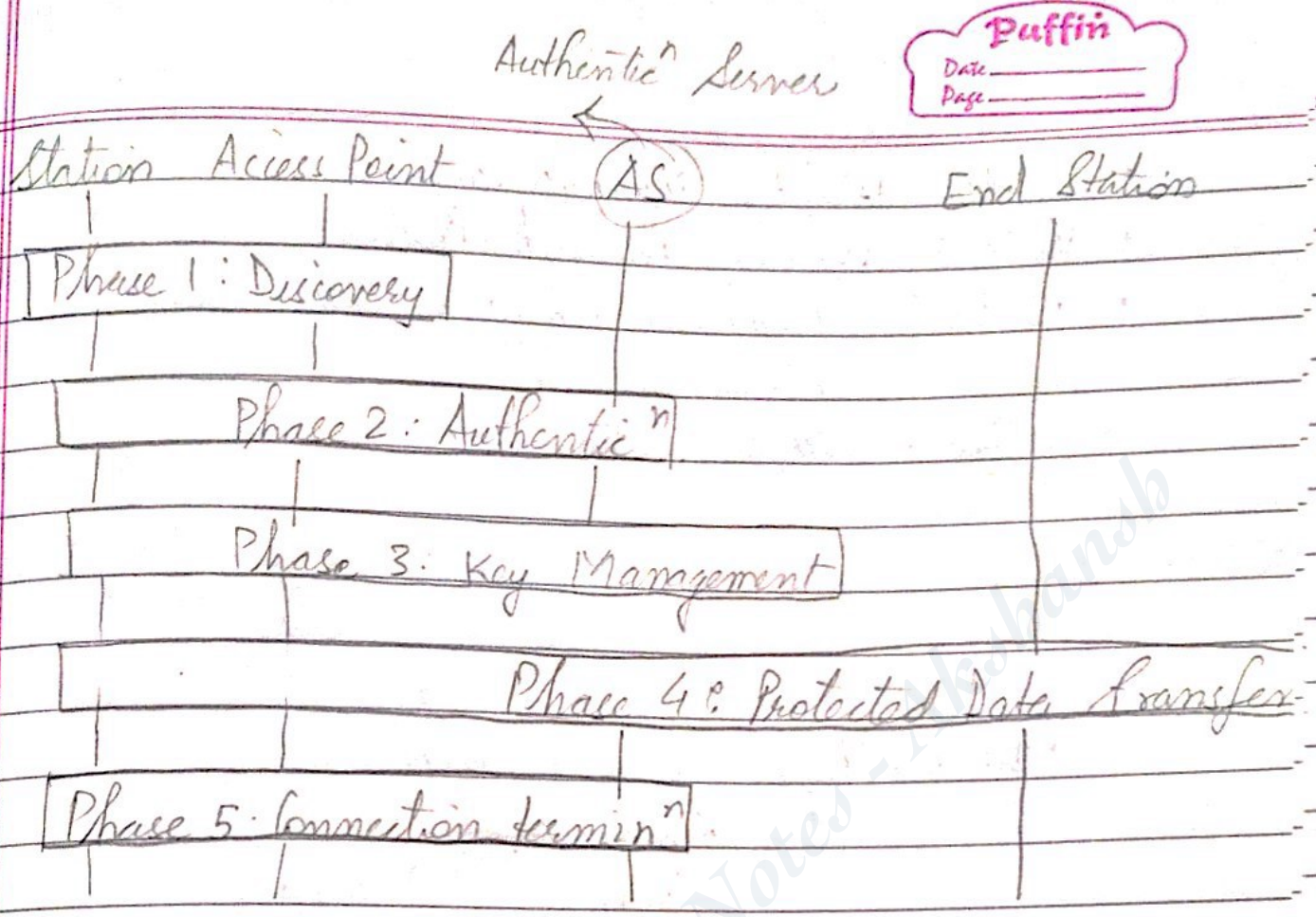
Services	Access Control	Authentic^n & Key Gener^n	confidentiality, Data Origin Authentic^n & Integrity and Replay Protection
Protocols	IEEE 802.1 Port based Access Control	Extensible Authentic^n Protocol (EAP)	TKIP, CCMP

TKIP → Temporary Key Integrity Protocol

CCMP → Counter Mode with Cipher Block Chaining MAC Protocol → Message Authentic^n Control

✡ 802.11(i) RSN Cryptographic Algorithms

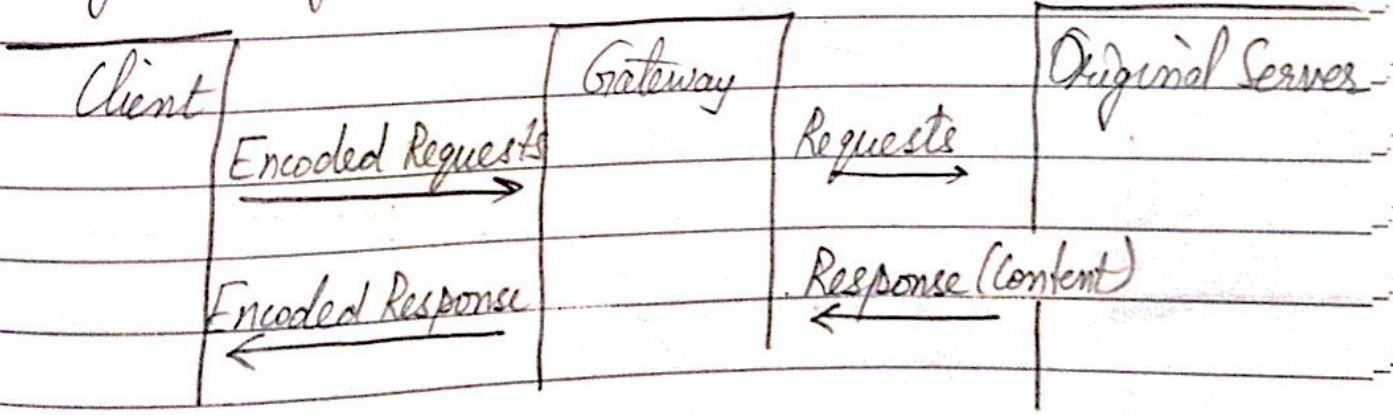
Services	Confidentiality	Integrity & Data Origin Authentic^n	Key Gener^n
Algorithms	TKIP, CCM, NIST → AES-CTR	HMAC-SHA-1, ----	HMAC-SHA, RFC-1750 ...



* 802.11i Protected Data Transfer Phase
  ↳ have 2 schemes for protecting data:
  ① Temporal Key Integrity Protocol (TKIP)
  ② Counter Mode CBC-MAC Protocol (CCMP)

• MIC: message integrity code.

* Wireless Application Protocol (WAP)
  Programming model:



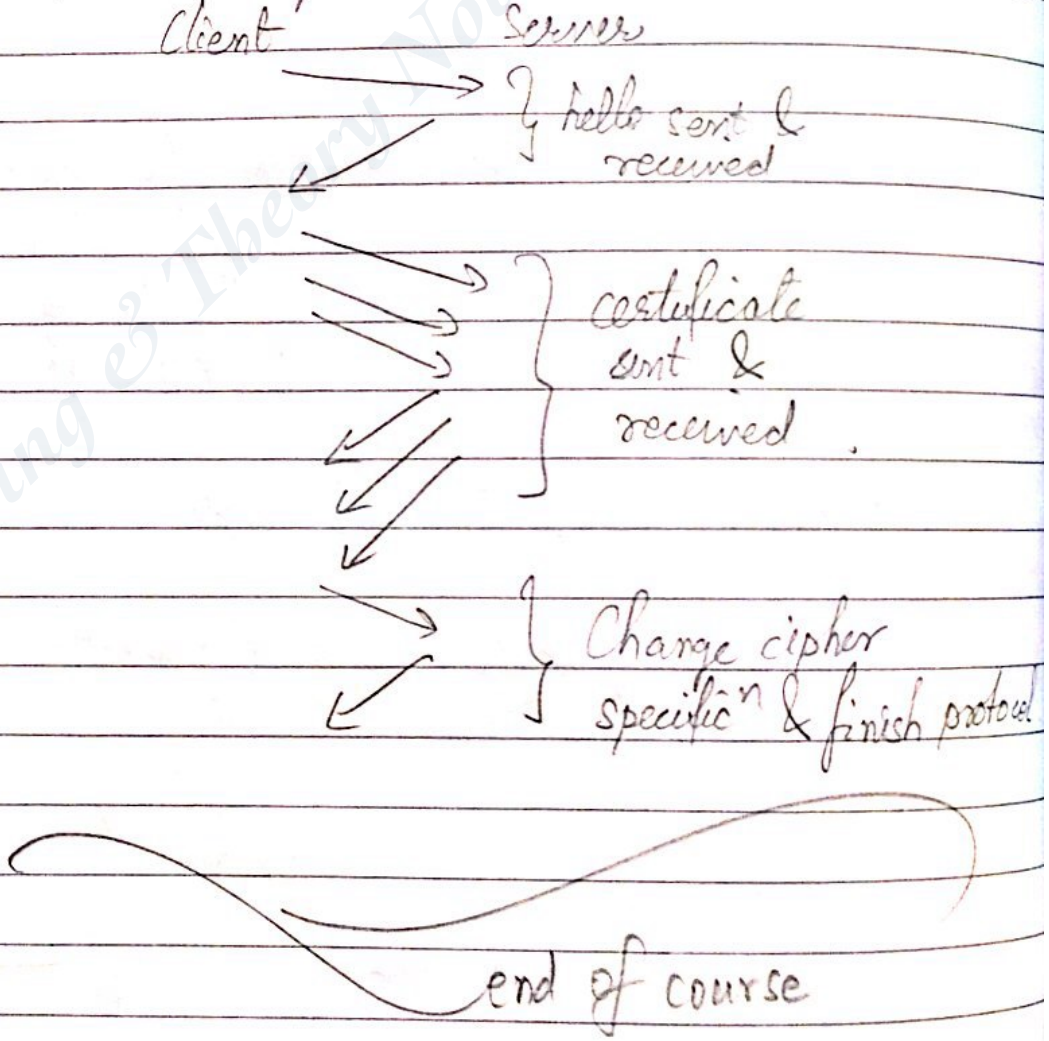
See the context & use that

* MAC : Message Authentic^{n} Control.
* MAC : Medium Access Control.

- WML : Wireless Markup Language.
- WTP : Wireless Transaction Protocol
- WTLS : Wireless Transport Layer Security
- WSP : Wireless Session Protocol.
- WDP : Wireless Datagram Protocol.

- WTLS provides denial-of-service protection, data integrity, privacy, authentication.

- WTLS higher layer protocols
  - Handshake protocol



end of course