بسم الله الرحمن الرحیم

جزوه تئوری اطلاعات و کدینگ

دکتر حمید وحدتی

منبع: Digital Communication [Ian Glover , Peter Grant ۳rd Edition]

لطفا برای هر چه بهتر شدن فعالیتمان در زمینه مخابرات امن ، ما رو از نظراتتان مطلع سازید.

www.stofahar.blogsky.com

تهیه : مهسا باغبان اورندی

Informatic Theory

«تئوری اطلاعات و کدینگ»

عنوان: ...

p=1

۹۶ / ۲۶۹

جلسه اول –

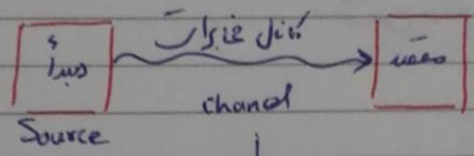
اجزای کلی یک سیستم مخابرات دیجیتال :

هدف : انتقال پیام (message) از یک مبدا به یک مقصد را انجام می‌دهیم.

با در نظر گرفتن معیارهای مثل تطبیق، حداکثر سرعت، صحت و ...

پیام : دنباله‌ای از سمبل‌های است مثل : دود، ...... مدولاسیون مخابرات

مبدا و مقصد از لحاظ فیزیکی جدا از هم هستند. یعنی ارسال به خود، معنا ندارد.



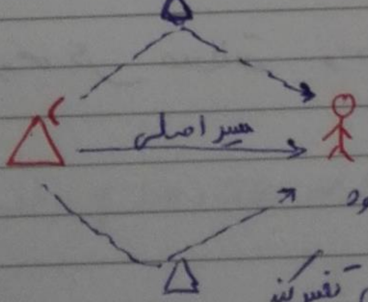
۱- فضای آزاد space
۲- کابل مسی
۳- فیبر نوری

۹۷ / ۲۶۸

پیام هنگام مخابره به مکان دیگر، دچار تغییراتی می‌شود.

مثل :
۱- تضعیف
۲- نویز (عمدی - غیرعمدی) : اگر سیستمی در دما صفر کلوین کار کند، حرکت الکترون‌ها مختل شده و نویز ایجاد می‌شود.
۳- اعوجاج

مثال برای اعوجاج :



در مسیرهای مختلف ممکن است ارسال شود

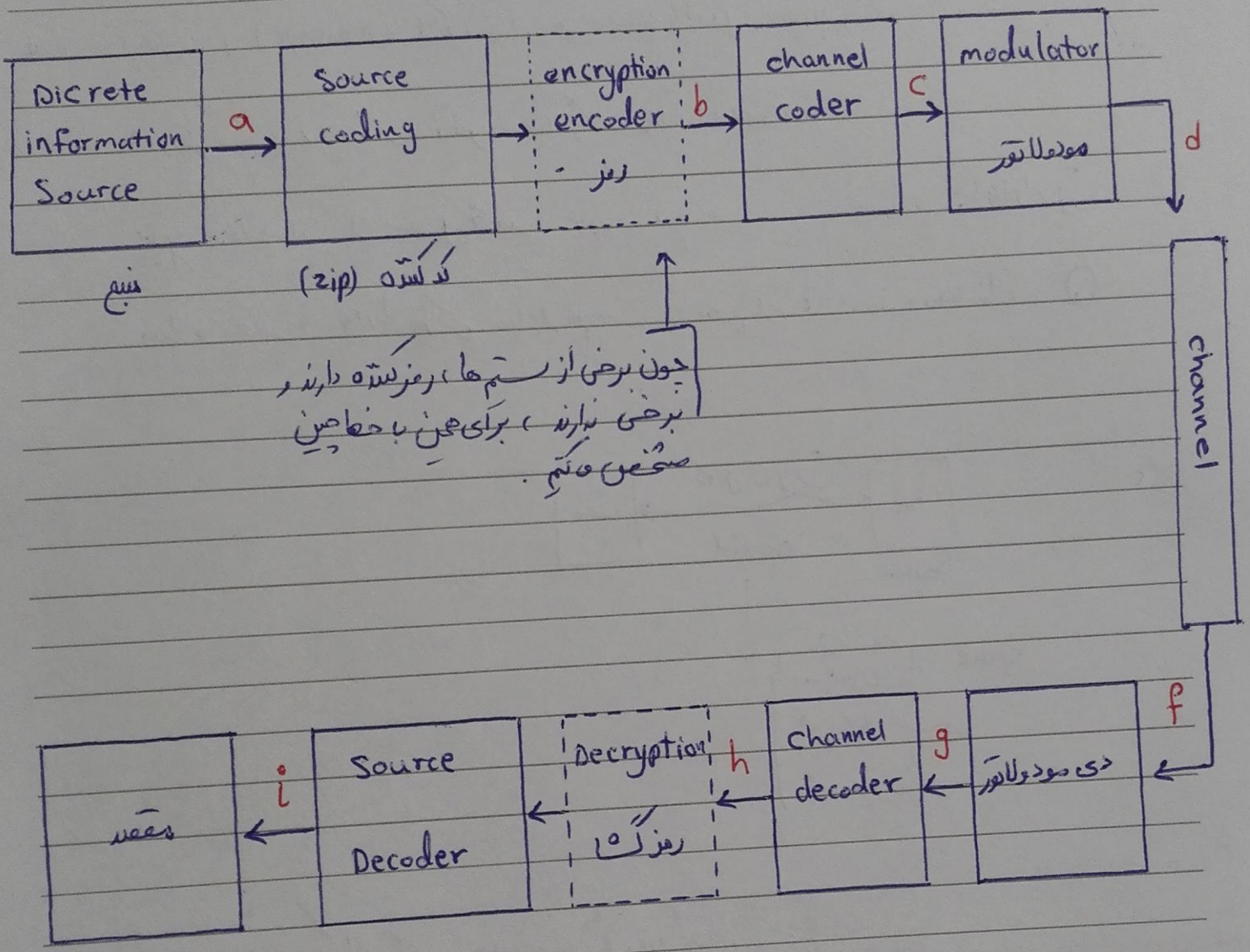
و موجب اعوجاج شده و شکل سیگنال تغییر کند.

p=2

۸۶۱ / ۱۶۷

هدف اصلی در این درس، بالا بردن صحت اطلاعات دریافتی در مقصد است.

شمای کلی سیستم مخابراتی:



توجه:

۱- عبارت انتقال دنباله‌ای از صفر و یک هم در مبدا، هم در مقصد

۲- بعد از مرحله Source Coding، کاراکتر از دنباله‌ی بیت‌ها تغییر کند و بیت‌ها کاراکتر نشوند.

۳- خروجی کد مودولاتور و دی‌مودولاتور، سیگنال‌های (تابع) موج‌دار باید باشند.

شروع توضیح اجزا :

P=3

۱- منبع اطلاعات "Information Source"

وظیفه‌اش : تولید سمبل‌ی که حاوی اطلاعات

- آنالوگ مثل: دوربین و mic
- دیجیتال مثل: PC ، TeleType

99/256

وقتی‌که یک منبع اطلاعاتی گسسته : با پارامترهای زیر مشخص می‌شود.

۱- الفبای منبع «سمبل‌ها یا حروف»

A (۸ بیتی) در اسکی → 65

حال فرض کنیم سمبل‌ها، صورت unicode باشند که در این صورت ۱۶ بیتی در نظر می‌گیریم.

۲- نرخ تبادل «نرخ سمبل»

یعنی با چه سرعتی باشد. [سرعت در عالم رفتن، چه تایپ، چه LAN، ...]

نرخ سمبل منبع baud می‌باشد که واحد آن $\frac{\text{Symbol}}{\text{sec}}$

مثال = تایپ ۱۰ کاراکتر در ثانیه $\frac{10\,sy}{se}$

که اگر، صورت اسکی باشد، اندازه‌اش می‌شود $80\,\frac{\text{bit}}{\text{sec}}$

۳- احتمال رخداد هر یک از الفبای منبع

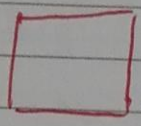
$x, z < e^{\uparrow}$

۴- وابستگی بین سمبل‌ها در یک دنباله

مثال = $q \times c$ → می‌دانیم که در ۹۹ درصد حالات، بعد از q، حرف u می‌آید پس می‌توان تشخیص داد.

خب پس چکار خواهیم کرد؟

می‌آییم و مدل اعدادی برای منبع در نظر می‌گیریم ← آنتروپی منبع را تعریف می‌کنیم. H



از روی این دو ویژگی می‌توانیم نرخ تولید اطلاعات (منبع) $\frac{bit}{sec}$ تولید کنیم.

هدف از رمزگذاری «coding» این است که اطلاعات را مخفی کنیم.

Channel Coding با بالای صفاوت است. اگر هم شنود شود، اطلاعات قابل فهم و مفید نخواهد بود.

کدینگ (رمزگذاری) کانال:

هدف: انتقال داده با کارایی و قابلیت اطمینان بالا

"efficiency & reliability"

چکار می‌کنیم؟ به صورت سیستماتیک (با قاعده) بیت‌هایی به خروجی Source encoder اضافی می‌شود که اگرچه هیچ اطلاعاتی را حمل نمی‌کنند ولی امکان

- آشکارسازی خطا — Error detection
- تصحیح خطا — Error Correction

را فراهم می‌کنند.

«همان بحث بیت توازن»

توجه: قدرت آشکارسازی بیشتر از تصحیح است.

P=6

101 / 264

منظور از BER : "bit error rate"

نظریه شانون : که می‌گوید این BER را نمی‌توانیم تا حدود زیادی کاهش دهیم ولی نمی‌توانیم صفر کنیم. مثلاً 10^{-5} یا 10^{-7}

جلسه ۲

روش‌های کدینگ کانال :

① روش کدینگ بلوکی Block Code

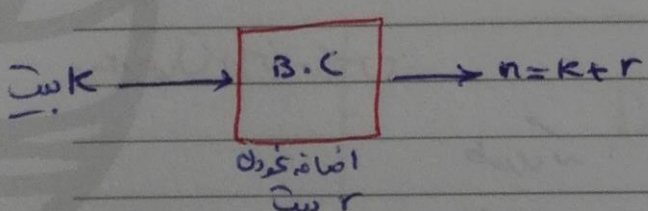
② کدینگ کانولوشنال یا چرخشی Convolational Coding

} در درس ما این دو صورت است.

۱- کدینگ بلوکی : یک آرایه k بیتی از ورودی می‌گیرد و به آن r بیت اضافه می‌کند و خروجی اش n بیت می‌شود. $n = k + r$

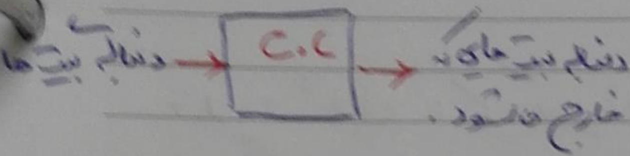
مثلاً parity بیت ، ۸ بیت دریافت $k=8$ یک بیت به آن اضافه می‌کرد $r=1$ و یک کد ۹ بیتی به ما می‌داد. $n=9$

به هر k بیت ورودی r بیت مربوط به اشکال‌زدایی و تصحیح خطا را اضافه می‌کند. و یک بلوک $n=k+r$ تایی تولید می‌کند.

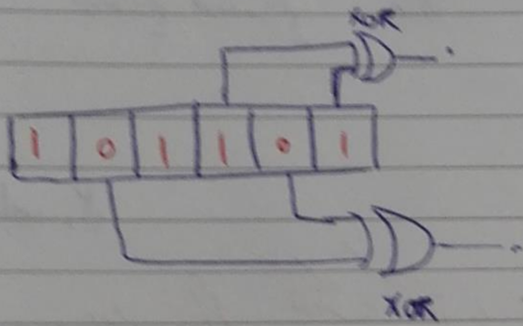


در نتیجه $n > k$

۲- کدینگ جریانی: دنباله یا stream های پیام به صورت پیوسته به کد کننده وارد و خروجی آن هم به صورت پیوسته تولید می‌شود.



مثال: مثلاً از یک shift register استفاده می‌کند.



....

بعداً به طور مفصل توضیح خواهد داد.

توجه: در بلوکی تأخیر داریم ولی در جریانی نداریم.

پارامترهای مهم در کدینگ کانال:

① روش کدینگ

② نرخ یا کارایی کد $\frac{n}{k} = \frac{\text{نرخ داده خروجی}}{\text{نرخ داده ورودی}}$

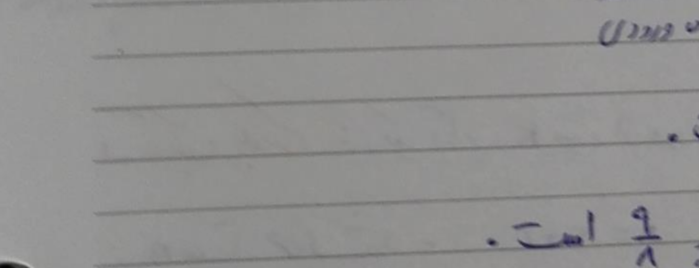
نرخ کارایی در مثال بالا برابر ۲ است.

و در بحث مربوط به parity برابر $\frac{9}{8}$ است.

③ قابلیت کنترل و تصحیح خطا

④ پیچیدگی

هدف‌های طراحی درس:



طوری که:
انتقال قابل اعتماد با خطای کم داشته باشیم.

p=6

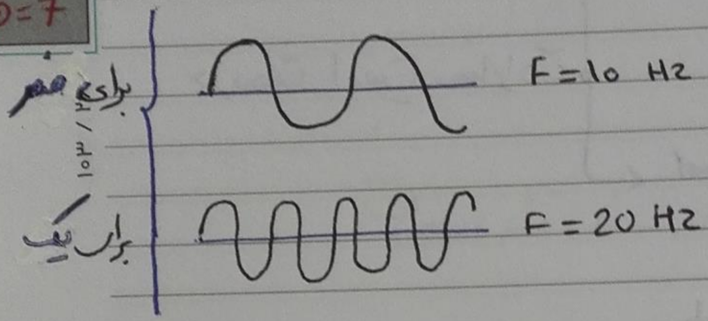
۱۴۰۲/۱۲/۲۳

وظیفه مدولاتور :

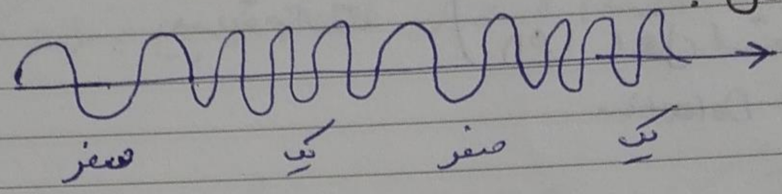
۱	۰	۱	۰

p=7

این روش به FSK موسوم است.
Frequency Shift Keying



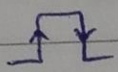
موقع ارسال :



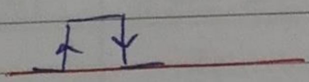
حالا دی مودلاتور : با گرفتن فرکانس و شکل موج، آن را به تعداد صفر و یک تبدیل می کند.

۱ـ فرکانس حامل (سنکرون سازی Carier)

۲ـ Bit Synchronization (مثال) همان بوق هایی که مودم در هر طرف برای سنکرون کردن باهم توافق می کنند. حال مثلاً روی $10^4 \frac{bit}{sec}$ توافق می کنند. مرحله ی بعدی توافق روی لبه های بالارونده و پایین رونده است.



که در رجیستری عبارات مطرح نشود.



در هر $\frac{1}{10^4}$ ثانیه

منبع این درس :

Digital Communication

فصل ۹ و ۱۰ { IAN A Glover
Peter M Grant

از منابع دیگر نظیر کاسترلو و گارسیو هم خواهد گفت.

رمزنگاری و کدینگ ✓ ، دسترسی X ، Authentication X

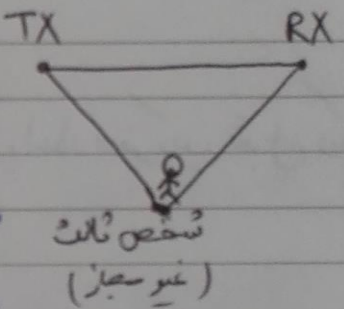
در بحث امنیت مخابرات

امنیت ترنسفر ✓

LPI اعمال آشکارپذیری پایین
Low Probability of intercept

باید خودمان دید داشته باشیم
مورد توجه است

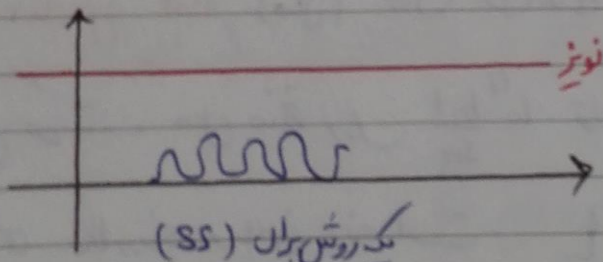
LPD اعمال شنود پایین
Low " " Detection

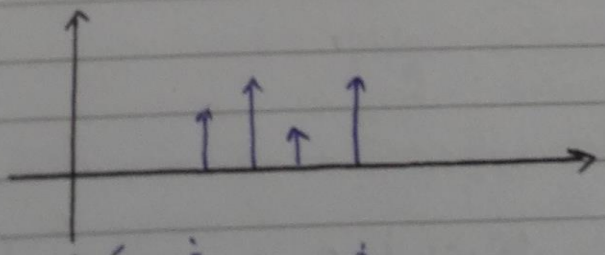


وظیفه ی مخابرات امن این است که کاربر ثالث را خنثی کند.

یعنی Authentication داشته باشیم و همین امنیت ترنسفر داشته باشیم. (SS)

مثل بارز SS :





روش دوم : پرش فرکانسی hop

تا اینجا فقط با صورت موضوعی و تیتروار درس رو نگاه کردیم. از این پس بطور جدی‌تر شروع میکنیم:

P=9

۱۰۶ / ۲۵۹

تئوری اطلاعات:

پدرگان علوم تئوری اطلاعات "Shannon, Nyquist, Hartly"

تعریف (Information): مقدار اطلاعاتی که پیام با میزان کمیاب و قابل پیش‌بینی بودن آن مرتبط است.

یعنی هرچقدر یک پیام محتمل‌تر باشد، قابل پیش‌بینی‌تر باشد، حاوی اطلاعات کمتری است.

مثال: (que → حاوی اطلاعات کم)

اگر $p(\text{پیام}) = 1$ آنگاه $I_m = 0$

اگر $p(\text{پیام}) = 0$ آنگاه $I_m = \infty$

عبارت دیگر باید گفت:

$\rightarrow I = I_1 + I_2$

محتوای اطلاعاتی دو پیام مستقل و $\rightarrow p_1 \times p_2$ پی‌درپی برابر مجموع محتوای اطلاعاتی هرکدام است.

پیام	$p(\text{پیام})$	محتوا اطلاعاتی
m_1	p_1	I_1
m_2	p_2	I_2
(m_1, m_2)	$p_1 p_2$	I_1+I_2

یعنی اطلاعات تابعی از احتمال است. $I_m = f(p)$

حال باید گفت چه تابعی؟؟

تابعی که حاصلضرب احتمال‌ها را به حاصل جمع اطلاعات تبدیل کند. که نگاهتم چنین عملکردی دارد.

← پس

$$I_m \triangleq \log_2 \frac{1}{p(m)} = -\log p(m)$$

میزان مزعول:

$I_n = 0 \leftarrow p(1)=1$
$I_n = 1 \leftarrow p(1)=0$

واحد اندازه‌گیری این مزعول (اطلاعات) bit است.

(binary Information)

جلسه ۳ :

منبع: الفبا می‌تواند باینری باشد. $\begin{matrix}0\\1\end{matrix}$ که دو تا الفبا دارد و یک بیت نیاز دارد. و اگر هم احتمال یکسان باشند: احتمال رخداد صفر و یک هرکدام $\frac{1}{2}$ است.

تعداد الفبا	تعداد ارقام مورد نیاز برای نمایش	احتمال رخداد هر سمبل
۲	۱	$\frac{1}{2}$
۴	۲	$\frac{1}{4}$
⋮	⋮	⋮
۱۲۸	۸	$\frac{1}{128}$

مقدار بار آنتروپی:

آنتروپی منبع: (Entropy) (H)

واحد آنتروپی $\left(\frac{bit}{symbol}\right)$

$$H \triangleq \sum_{m=1}^{\text{تعداد سمبل‌ها}} p(m) \cdot \underbrace{\log_2 \frac{1}{p(m)}}_{I_m}$$

آنتروپی بیان می‌کند که به طور متوسط هر سمبل دارای چه مقدار (چند واحد اطلاعاتی) داده‌ای باشد.

ادامه ←

P=11

۱۰۸ / ۲۵۷

M

$I_1 \quad I_2 \quad I_3 \quad \dots \quad I_m$ وقتی اطلاعات محل ها

$P_1 I_1 + P_2 I_2 + \dots + P_m I_m =$ مقدار صورت ها (امید ریاضی)

آنتروپی منبع باینری :

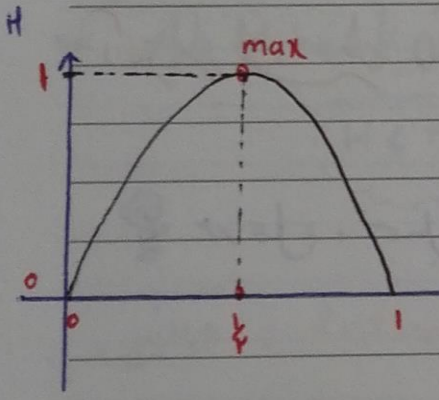
S

$0 \rightsquigarrow 1-p$

$1 \rightsquigarrow p$

$$H = \sum_{m=1}^{2} P(m) \cdot \text{Log}_2 \frac{1}{P(m)} = P(1) \cdot \text{Log} \frac{1}{P(1)} + P(2) \cdot \text{Log} \frac{1}{P(2)}$$

$$= \boxed{p \text{Log} \frac{1}{p} + (1-p) \text{Log} \frac{1}{(1-p)}} \quad \frac{\text{bit}}{\text{symbol}}$$ واحد



به عبارت دیگر، آنتروپی وقتی ماکزیمم می‌شود که سمبل‌ها هم احتمال باشند.

$p = 1/2$ احتمال سمبل ۱

$1-p = 1-1/2 = 1/2$ احتمال سمبل ۲

به عبارت بهتر، حداکثر آنتروپی وقتی رخ می‌دهد که بیشترین شک و دودلی را داشته باشیم.

بهترین حالت، در حالی است که ۵۰-۵۰ باشد.

تمرین (تکلیف درسی) : ثابت کنید که برای الفبای ۲ سمبلی، مقدار H برابر $p = 1/2$ ، ماکزیمم است. همچنین نمودار H را با کمک Matlab رسم کنید.

در اغلب منابع اطلاعاتی، سمبل‌ها هم احتمال نمی‌باشند. به عبارت دیگر آنتروپی آنها کمتر از مقدار ماکزیمم می‌باشد.

مثلاً اگر آنتروپی را برای کد اسکی حساب کنیم،

اگر فرض کنیم $p(m)=\frac{1}{128}$ که قبلاً گفتیم کدهای اسکی هم احتمال هستند.

$$H=\sum_{m=1}^{128} p(m)\cdot \log\frac{1}{p(m)} < 7\ bit$$

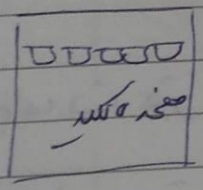
چون هر سمبل اسکی، اطلاعاتی کمتر از 7 بیت را حمل می‌کند.

$$H=7\ bit$$

آنتروپی شرطی و افزونگی (Redundency)

در عمل، سمبل‌های مختلف هم احتمال و مستقل از هم نمی‌باشند.

در کد اسکی



→ احتمال e خیلی بیشتر است.

۹۶ بیت مستقل هستند

مزیت کدینگ: برای نشان دادن وابستگی بین کدها باید از احتمال شرطی استفاده کنیم. که با تقریب خوبی از آن، آنتروپی شرطی می‌رسیم.

در این حالت باید احتمال مشترک و شرطی (conditional) دنباله سمبل‌ها را در نظر بگیریم.

$$H=\sum_i\sum_j p(i,j)\log_2\frac{1}{p(j|i)}$$

$p(i,j)$: احتمال اینکه منبع سمبل‌های i و j را تولید کند.

$p(j|i)$: احتمال اینکه منبع سمبل j را تولید کند، مشروط به اینکه قبلاً سمبل i را تولید کرده باشد.

فرض کنیم :

کلید: ←

t	S	R	I	F	E	H	T

P=13

۱۱۰ / ۲۵۵

حرف اول	حرف دوم	احتمال مشترک	
T	H	۰/۰۰۱	p(T, H)
T	Z	۰/۰۰۰۱	p(T, Z)
T	P	۰/۰۰۰۰۰۰۱	p(T, P)
⋮	⋮		

احتمال شرطی	حرف دوم	حرف اول
۰/۰۰۱	e	i
۰/۰۰۰۰۱	i	e
۰/۹۹	u	q
۰/۰۰۰۰۱	a	a

تا 26*26 سطر خواهد داشت

دیدیم ، حداکثر آنتروپی منبع درحالتی است که سمبل‌ها مستقل و هم احتمال باشند. (H_{max})

ولی در عمل ، منابع سمبل‌های تولید شده توسط منبع ، ویژگی‌های فوق را ندارند ولذا دارای یک آنتروپی‌ای هستند که از H_{max} کمتر است. ($H < H_{max}$)

به تفاوت بین H و H_{max} ، اصطلاحاً افزونگی یا " Redundancy " می‌گویند.

۱۱۱ / ۲۵۴

اطلاع اضافی $R = H_{max} - H$ $\frac{bit}{symbol}$

همه

۲۶ رجب ۱۴۳۱
9 Jul 2010

مثال : یک منبع چهار سمبل (A, B, C, D) را با نرخ $baud = 1024 \frac{symbol}{sec}$ و با احتمال $p(A) = ۰/۵$ ، $p(B) = ۰/۲$ ، $p(C) = ۰/۲$ و $p(d) = ۰/۱$.

مطلوبست :

$Baud = 1.24 \frac{symbol}{sec}$

$p(A) = ۰/۵$
$p(B) = ۰/۲$
$p(C) = ۰/۲$
$p(d) = ۰/۱$

الف) آنتروپی منبع (H) ؟

$$H = \sum_{m=1}^{۴} p(m) . \log_{۲} \frac{1}{p(m)} =$$

ادامه در صفحه بعد ←

ادامه

$$= p(1)\log\frac{1}{p(1)} + p(2)\log\frac{1}{p(2)} + p(3)\log\frac{1}{p(3)} + p(4)\log\frac{1}{p(4)} =$$

$$= \frac{1}{2}\log_2 2 + \frac{2}{10}\log_2 \frac{10}{2} + \frac{2}{10}\log_2 \frac{10}{2} + \frac{1}{10}\log 10 =$$

$$= \frac{1}{2} + \frac{1}{5}(0.2988) + \frac{1}{5}(0.2988) + \frac{1}{10}$$

$$= 1.72 \text{ bit/symbol}$$

یادآوری:

$$\log_B A = \frac{\log A \text{ هرپایه ای}}{\log B \text{ هرپایه ای}}$$

ب) افزونگی R ؟

$$H_{max} = 4\left[\frac{1}{4}\log_2 4 + \frac{1}{4}\log_2 4\right] = \frac{1}{4}+\frac{1}{4}+\frac{1}{4}+\frac{1}{4} = 2 \frac{bit}{Symbol}$$

$$R = H_{max} - H = 2 - 1.72 = 0.28 \frac{bit}{Sym}$$

ج) نرخ اطلاعات ؟

هر ثانیه ← 1024 Symbol

نرخ اطلاعات واقعی $= 1024 \frac{Sym}{sec} * 1.72 \frac{bit}{Sym} = 1760 \frac{bit}{sec}$

$R_i = R_s H$

نرخ نماد Symbol rate

نرخ اطلاعات information rate

$h_2(0.3) = 0.3$ $h_3(0.7) = 0.7$ $h_4(0.4) = 0.4$

$h_5(0.7) = 0.7$ $h_6(0.7) = 0.77$ $h_7(0.14) = 0.14$

$h_8(0.9) = 0.90$ $h_9(0.95) = 0.95$ $h_{10}(1) = 1$

حل مسئله ⊛

$$= \frac{1}{2} + \frac{2}{5}\left(h_2\left(\frac{1}{5}\right)\right) + \frac{1}{10}\log 10 = \frac{1}{2} + \frac{2}{5}\left(\log_2 5 - 1\right) + \frac{1}{10}\left(\frac{\log 10}{\log 2}\right)$$

$$= \frac{1}{2} + \frac{2}{5}\left(\frac{7}{3}\right) + \frac{1}{10}\left(\frac{1}{0.3}\right) = \frac{1}{2} + \frac{4}{5}\left(\frac{7}{3}\right) + \frac{1}{10}\times\frac{10}{3} =$$

$$= \frac{1}{2} + \frac{14}{15} + \frac{1}{3} = \frac{23}{30} \simeq 1.76 \frac{bit}{Sym}$$

9.13 Problems

9.1 (a) Consider a source having an $M = 3$ symbol alphabet where $P(x_1) = ½$, $P(x_2) = P(x_3) = ¼$ and symbols are statistically independent. Calculate the information conveyed by the receipt of the symbol x_1. Repeat for x_2 and x_3. [$I_{x_1} = 1$ bit, $I_{x_2} = I_{x_3} = 2$ bits]

(b) Consider a source whose statistically independent symbols consist of all possible binary sequences of length k. Assume all symbols are equiprobable. How much information is conveyed on receipt of any symbol? [k bits]

(c) Determine the information conveyed by the specific message $x_1x_3x_2x_1$ when it emanates from each of the following, statistically independent, symbol sources: (i) $M = 4$; $P(x_1) = ½$, $P(x_2) = ¼$, $P(x_3) = P(x_4) = 1/8$ [7 bits]; (ii) $M = 4$; $P(x_1) = P(x_2) = P(x_3) = P(x_4) = ¼$. [8 bits]

9.2 (a) Calculate the entropy of the source in Problem 9.1(a). [1½ bit/symbol]

(b) Calculate the entropy of the sources in Problem 9.1(c). [1¾ bit/symbol, 2 bit/symbol]

(c) What is the maximum entropy of an eight-symbol source and under what conditions is this situation achieved? What are the entropy and redundancy if $P(x_1) = ½$, $P(x_i) = 1/8$ for $i = 2, 3, 4$ and $P(x_i) = 1/32$ for $i = 5, 6, 7, 8$? [3 bit/symbol, 2.25 bit/symbol]

9.3 Find the entropy, redundancy and code efficiency of a three-symbol source A, B, C, if the following statistical dependence exists between symbols. There is a 20% chance of each symbol being succeeded by the next symbol in the *cyclical* sequence $A\,B\,C$ and a 30% chance of each symbol being succeeded by the previous symbol in this sequence. [1.485 bit/symbol; 0.1 bit/symbol; 93.7%]

9.4 Show that the number of redundant symbols per bit of information transmitted by an M-symbol source with code efficiency η_{code} is given by $(1 - \eta_{code})/(\eta_{code} \log_2 M)$ symbol/bit.

9.5 Estimate the maximum information content of a black and white television picture with 625 lines and an aspect ratio of 4/3. Assume that 10 brightness values can be distinguished and that the picture resolution is the same along a horizontal line as along a vertical line. What maximum data rate does a picture rate of 25 picture/s correspond to and what, approximately, must be the bandwidth of the (uncoded and unmodulated) video signal if it is transmitted using binary symbols? (If necessary you should consult Chapter 16 to obtain TV scanning format information.) [4 bit/symbol and 2.0832 Mbit/picture; 52.08 Mbit/s; 26.04 MHz]

9.6 Calculate the loss in information due to noise, per transmitted digit, if a random binary signal is transmitted through a channel, which adds zero mean Gaussian noise, with an average signal-to-noise ratio of: (a) 0 dB; (b) 5 dB; (c) 10 dB. [0.6311; 0.2307; 0.0094 bit/binit]

9.7 An information source contains 100 different, statistically independent, equiprobable symbols. Find the maximum code efficiency, if, for transmission, all the symbols are represented by binary codewords of equal length. [7 bit words and 94.9%]

9.8 (a) Apply Huffman's algorithm to deduce an optimal code for transmitting the source defined in Problem 9.1(c)(i) over a binary channel. Is your code unique?

(b) Define the efficiency of a code and determine the efficiency of the code devised in part (a).

(c) Construct another code for the source of part (a) and assign equal-length binary words irrespective of the occurrence probability of the symbols. Calculate the efficiency of this source. [(a) 0, 10, 110, 111, Yes; (b) 100%; (c) 87.5%]

9.9 Design a Lempel–Ziv coding scheme with a history buffer of 16 8 bit characters and a maximum match length of four characters. Use your coding scheme to encode the character sequence '1415927' where the next character to be transmitted is to the right. You should assume the history buffer contains '1414213617920408' with the last character to have been transmitted to the left. Calculate the compression factor obtained by your code on this character sequence. [0.536]

تمرینات فصل ۹ :

P=15

(a) = 9-1

$$I_{x_1} = \log_2 \frac{1}{p(x_1)} = \log_2 \left(\frac{1}{1/2}\right) = 1 \text{ bit}$$

$$I_{x_2} = I_{x_3} = \log_2 \frac{1}{p(x_2)} = \log_2 \left(\frac{1}{1/4}\right) = 2 \text{ bit}$$

(b)

$$I_x = \log_2 \frac{1}{p(x)} = \log_2 \frac{1}{(1/2^K)} = \log_2 2^K = K \text{ bit}$$

(c)

$$I_m = \log_2 \frac{1}{p(m)}$$

$$I_{x_1} = \log_2 \frac{1}{1/2} = 1 \text{ bit} \qquad I_{x_4} = \log_2 \frac{1}{1/8} = 3 \text{ bit}$$

$$I_{x_2} = \log_2 \frac{1}{1/4} = 2 \text{ bit}$$

x_1	x_3	x_2	x_1	→	1 bit	3 bit	2 bit	1 bit	= 7 bit

(a) (9-2)

$$H = \sum_{m=1}^{M} p(m) \log_2 \frac{1}{p(m)} = \sum_{m=1}^{3} p(m) \log_2 \frac{1}{p(m)} = p(1) \cdot \log_2 \frac{1}{p(1)} + p(2) \log_2 \frac{1}{p(2)} + p(3) \log_2 \frac{1}{p(3)} =$$

$$= \frac{1}{2} \log_2 2 + \frac{1}{4} \log_2 4 + \frac{1}{4} \log_2 4 = \frac{3}{2} = \boxed{1\frac{1}{2}} \text{ bit/sym}$$

(b)

$$H_C = \sum_{m=1}^{4} p(m) \log_2 \frac{1}{p(m)} = \frac{1}{2} \log_2 2 + \frac{1}{4} \log_2 4 + \frac{1}{8} \log_2 8 + \frac{1}{8} \log_2 8 = \frac{1}{2} + \frac{1}{2} + \frac{3}{8} + \frac{3}{8} = \boxed{1\frac{3}{4}} \text{ bit/sym}$$

(c) حداکثر آنتروپی زمانی است که هر ۸ سمبل، احتمال مساوی و مستقل داشته باشند.

$$p(m) = 1/8$$

$$H_{max} = 8\left(\frac{1}{8} \log_2 8\right) = 3 \text{ bit/sym}$$

(ب)

$$H = \sum_{m=1}^{8} p(x_m) \log_2 \frac{1}{p(x_m)} = \frac{1}{2} \log_2 2 + 3\left(\frac{1}{8} \log_2 8\right) + 3\left(\frac{1}{32} \log_2 32\right) = \frac{1}{2} + \frac{9}{8} + \frac{15}{8} = \boxed{2,25} \text{ bit/sym}$$

$$R = H_{max} - H = 3 - 2,25 = \boxed{0,75 \text{ bit/sym}}$$

(۳-۹) مطلوبست محاسبه آنتروپی و افزونگی رکاری منبعی که دارای ۳ سمبل A, B, C است. ۲۰٪ احتمال اینکه سمبلی با سمبل بعد سرهم واقع شده باشد. ۳۰٪ حدس اول اینکه هر سمبل بعد از سمبل قبل خود بیاید.

ص ۱۱۴ / ۲۵۱

$p = 16$

سمبل بعدی

$p(B|A) = ۰٫۲$ $\quad p(A|B) = ۰٫۳$

$p(C|B) = ۰٫۲$ $\quad p(B|C) = ۰٫۴$

$p(A|C) = ۰٫۲$ $\quad p(C|A) = ۰٫۴$

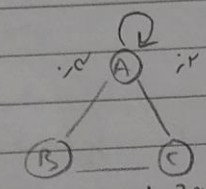
$$H = \sum_i p(i) \sum_j p(j|i) \log \frac{1}{p(j|i)} =$$

$$= p(A)\left(p(A|A) \log \frac{1}{p(A|A)} + p(B|A) \log \frac{1}{p(B|A)} + p(C|A) \cdot \log \frac{1}{p(C|A)} \right) +$$

$$p(B)\left(p(A|B) \log \frac{1}{p(A|B)} + p(B|B) \log \frac{1}{p(B|B)} + p(C|B) \log \frac{1}{p(C|B)} \right) +$$

$$p(C)\left(p(A|C) \cdot \log \frac{1}{p(A|C)} + p(B|C) \log \frac{1}{p(B|C)} + p(C|C) \cdot \log \frac{1}{p(C|C)} \right) =$$

توجه: $p(A|A) \neq 1$ (پیتاگوراس‌اه)



$1 - (۰٫۴ + ۰٫۲) = ۰٫۴$ احتمال $(A|A)$

یعنی اولی A باشه، شرط اینکه قبلی هم A بود.

باید $p(A)$ ، $p(B)$ ، $p(C)$ را حساب کنیم.

برای محاسبه آنها از معادلات احتمال کل استفاده می‌کنیم:

$$p(C) = p(C|A)p(A) + p(C|B)p(B) + p(C|C)p(C) \quad (۲)$$

$$\Rightarrow p(C|A)p(A) + p(C|B)p(B) = \frac{1}{2} p(C)$$

برای $p(B)$ ، $p(A)$ هم همینطور می‌شود.

$$p(B) \rightarrow p(B|A)p(A) + p(B|C)p(C) = \frac{1}{2} p(B)$$

$$p(A) \rightarrow p(A|B)p(B) + p(A|C)p(C) = \frac{1}{2} p(A)$$

ادامه در صفحه بعد

p=17

فرض معادلات احتمال کل برای A ، B ، C یکسان است ، هم احتمال اند .

$p(A) = p(B) = p(C) = 1/3$

بعد در فرمولی که نوشته بودیم ، جایگذاری میکنیم تا جواب ۳-۹ بدست بیاید .

$H \cong 1.48$

Reduction $\Rightarrow H_{max} = \log_2 3 = 1.58 \ \frac{bit}{sym}$

↓

$R = H_{max} - H = 1.58 - 1.48 = 0.1$

$\eta = \frac{H}{H_{max}} = \frac{1.48}{1.58} = $

Symbol : A member of source alphabet. May or may not be binary .eg. 2 symbol binary , 4 symbol PSK , 128 symbol ASCII

Baud : Rate of symbol transmission , i.e. 100 baud = 100 Sym/s.

کدینگ منبع (فشرده سازی):

سمبل هایی که منبع تولید کرده است. → Source encoder → پیام کد شده

باید مقدار اطلاعات هر سمبل کد شده > مقدار اطلاعات سمبل های تولید شده توسط منبع کد نشده

۱۰۰ → ۵۰
[۵۰]

۷۵ → ۲۰
[۱۵۰۰]

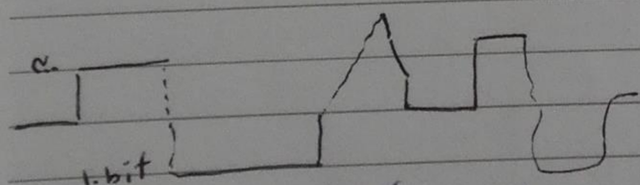
مزیت های Source Coding:

* Source Coding باعث کاهش حجم اطلاعات برای ذخیره سازی، انتقال یا پردازش می شود.

توجه: Source coding، انتروپی منبع را تغییر نمی دهد و در حقیقت باعث تغییر (معمولا افزایش) انتروپی سمبل های کد شده می شود.

* باعث کاهش تغییرات (Flactuation) در نرخ اطلاعات می شود.

مثلاً فرض کنید در یک کانال اطلاعاتی (مثل تلفن، وایرلس، ...) به ما بگویند به طور متوسط ها ۱۰۰ $\frac{Kb}{sec}$ تبادل داده داشته باشید. در این حالت می گویند داده ای که دارید، دچار تغییرات زیاد می شود.



باعث افزایش زیاد Surge سمبل ها.

باعث overload شدن خطا انتقالی می شود. به خصوص وقتی که دنباله پیام حاوی سمبل های با اطلاعات کم باشد.

کارایی efficiency در (Source Coding) :

سمبل‌های منبع → [Source Coding] → سمبل‌های کد شده

برای منبع مستقل $H = \sum_{m=0}^{M} p(m) \log_2 \frac{1}{p(m)}$ (آنتروپی منبع)

قبلاً دیدیم که اگر سمبل‌ها مستقل و هم‌احتمال باشند، آنوقت :

$$H = H_{max} = \log_2 M$$

احتمال هر سمبل $\frac{1}{M}$

توجه : اگر سمبل‌های منبع به صورت باینری کد شده باشند، کمیت دیگری برای تعریف کارایی کد وجود دارد.

فرمول کارایی که قبلاً داشتیم

$$\eta_{code} = \frac{H}{H_{max}} \times 100$$

← کارایی بر حسب درصد (I)
، بدون واحد

اگر یک مجموعه از سمبل‌ها با یک کد (Code word) به طول l_m رقم باینری کد شده باشند، طول متوسط کد به صورت :

$$\bar{l} = \sum_{m=1}^{M} p(m)\, l_m \quad \frac{bit}{sym} \leftarrow \left(\frac{binary\ digit}{Sym}\right)$$

از (I) و (II) → $H_{max} = L \ \frac{bit}{codeword}$ ؛ واحد $\frac{bit}{Sym}$

فرمول کارایی بر حسب طول

$$\eta_{code} = \frac{H}{\bar{l}} \times 100$$

(II) بدون واحد

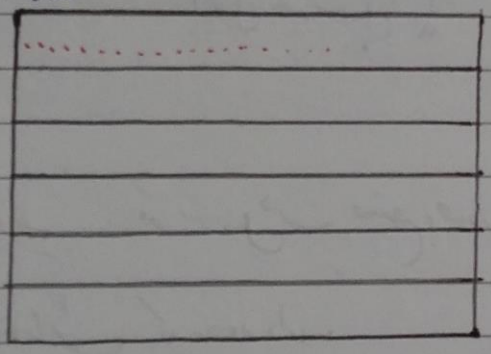
مثلاً فرض کنیم :

	احتمال رخداد		
Symbol #1	$P_1 = 0.5$	001	3 bit
Symbol #2	$P_2 = 0.3$	10	2 bit
Symbol #3	$P_3 = 0.2$	100001	6 bit

$\Rightarrow$ $(0.5 \times 3 + 0.3 \times 2 + 0.2 \times 6) = \boxed{3.4 \text{ bit}}$ / در این حالت، مثل فرمول حدول قبل، طول متوسط کد بدست می‌آید.

(درصد ↑ bit ↑)

نقاط سیاه و سفید



به تعداد کاراکترهایی که به صورت پیکان پشت سرهم سفید یا سیاه هستند، run گفته می‌شود.

A	A	A	A	A	A	A	B	C	C	C

A7 اسلوش ارسال می‌کند.

در اینجا، حال ارسال ۷ کاراکتر، ۸ بیتی، یعنی ۵۶ بیت، یک کاراکتر ۸ بیتی و یک عدد حداکثر ۸ بیتی ارسال می‌کند.

مثال ۸ اسکنر یک نسخه سیاه و سفید را به صورت خط به خط پویش (Scan) کرده و آن را به داده‌های باینری برای انتقال تبدیل می‌کند. (یک اسکنر خاص می‌تواند طول run های تا ۶ پیکسل را کد کند. احتمال رخداد هریک از run ها به شکل زیر است :

یعنی run های به طول ...

p=21

۱۲۰ / ۲۳۵

تعداد run / تعداد پیکسل‌های یکسان پشت سرهم	۱	۲	۳	۴	۵	۶
احتمال رخداد	۰٫۲	۰٫۴	۰٫۱۵	۰٫۱	۰٫۰۶	۰٫۰۹

مطلوبست :

الف) محاسبه‌ی طول متوسط run ها برحسب pixel ؟

$$l = \sum_{m=1}^{6} p(m)\, l_m = 1\times0.2 + 2\times0.4 + 3\times0.15 + 4\times0.1 + 5\times0.06 + 6\times0.09 =$$

$$= 0.2 + 0.8 + 0.45 + 0.4 + 0.3 + 0.54 = 2.69 \text{ pixel}$$

یعنی به طور متوسط متن دارای اجزای ۲٫۶۹ پیکسلی می‌باشد. (Average Length)

* هر سمبل دارای طول ۲٫۶۹ پیکسل است.

ج) نرخ موثر تولید [اطلاعات] در حالتی که اسکنر با سرعت ۱۰۰۰ پیکسل بر ثانیه، متن را scan (روبش) می‌کند، چقدر است؟

$$H = \sum_{m=1}^{6} p(m) \log_2 \frac{1}{p(m)} = 0.2\times2.32 + 0.4\times1.32 + 0.15\times2.74 + 0.1\times3.32 +$$

$$0.06\times4 + 0.09\times3.47 = 2.29 \ \frac{\text{bit}}{\text{sym}}$$

$$1^{\text{sec}} \rightarrow 1000^{\text{pixel}} \rightarrow \frac{1000}{2.69} \text{ سمبل} \rightarrow \frac{2.29 \text{ bit}}{\text{Symbol}} \times \frac{1000}{2.69} \text{ symbol} =$$

$$\Rightarrow = \frac{2290}{2.69} \text{ bit} = 852 \ \frac{\text{bit}}{\text{sec}}$$

پس این اسکنر را می‌توانم با Dialup $1200 \ \frac{\text{bit}}{\text{sec}}$ وصل کنم.

(مثلاً)

فلسفه کلی فشرده سازی :

اگر ما بتوانیم بیت ها یا سمبل‌هایی که دارای تکرار (احتمال حضور) بالایی هستند را با تکه کد های کوتاه و سمبل‌هایی که دارای تعداد تکرار کم هستند را با طول کدهای بلندتر ارسال کنیم، می‌توانیم عمل فشرده سازی منبع را انجام دهیم.

«از کدینگ با طول کد متغیر استفاده کنیم.»

طول کد کم ↓ — در symbol پرتکرار

طول کد بلند ↑ — در symbol کم تکرار

مشکلی که پیش می‌آید این است که در طول کدهای متغیر، عمل دی‌کدینگ سخت‌تر است.
وقتی می‌خواهیم تکه کد با طول متغیر را دی‌کد کنیم، باید ملاحظات زیر را مد نظر قرار دهید:

۱- عمل دی‌کدینگ باید منحصر بفرد Unique باشد.

حال می‌بینیم که چگونه می‌توان به این ویژگی منحصر بفرد رسید؟

مثال: فرض کنید منبع دارای یک الفبای ۴ سمبلی است که با رقم‌های باینری کد شده اند.
فرض می‌کنیم که کدی به صورت 0011 در گیرنده دریافت کردیم، حال می‌خواهیم تشخیص دهیم که کدام سمبل بوده است؟

A → 0
B → 01
C → 11
D → 00

0011 → AB یا DC یا AAC

توجه: مشکل از اینجاست که یکی از کدها prefix شده است، یعنی کد 0 هم پیشوند B هست و هم پیشوند D.

p=23

۱۳۲ / ۲۴۳

۲- دی‌کدینگ آنی یا لحظه‌ای "Instantaneous decoding"

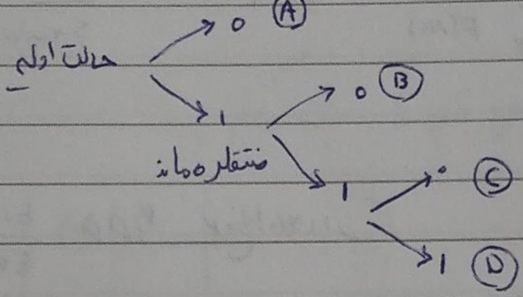
(مثل تصمیم گیری بچه‌ی کلاس)

مثال: فرض کنید یک الفبا ۴ سمبلی داریم که به این صورت کد شده اند:

A → ۰
B → ۱۰
C → ۱۱۰
D → ۱۱۱

تا حال اینهایی که داریم، unique هستند. یعنی هیچکدام پیشوند دیگری نیست.

(درخت تصمیم گیری Decision Tree)



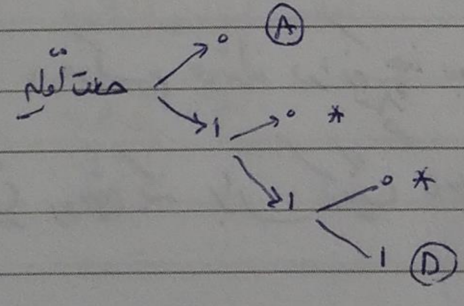
به این اصطلاحاً Comma Code هم می‌گویند.

به این نوع کدها که صفر حکم بیت انتهای کد را نشان میدهد. البته به جز در حالت ۱۱۱.

مثال: در این مثال، کدهای مثال قبل را Reverse می‌کنیم:

A → ۰
B → ۰۱
C → ۰۱۱
D → ۱۱۱

این کد را نمی‌توان دی‌کدینگ لحظه‌ای کرد.



در ضمن مشکل پیشوندی هم دارد یعنی unique نیست.

کدینگ با طول متغیر : درس را با مثال توضیح می‌دهد :

M	A	B	C	D	E	F	G	H
p(M)	۰٫۱	۰٫۱۸	۰٫۴	۰٫۰۵	۰٫۰۶	۰٫۱	۰٫۰۷	۰٫۰۴

$$H = \sum_{M=1}^{8} p(M) \log_2 \frac{1}{p(M)} = 2.55 \ \frac{bit}{symbol}$$

اگر نرخ سمبل $= 1 \ \frac{symbol}{sec}$ نرخ اطلاعات $2.55 \ \frac{bit}{sec}$

$$H_{max} = \log_2 M = \log_2 8 = 3$$

$$\eta_{source} = \frac{H}{H_{max}} \times 100 = \frac{2.55}{3} \times 100 = 85\%$$

یعنی ۸۵ درصد محتوای یک قطعه حاوی اطلاعات است.

هدف : می‌خواهیم این اطلاعات را با طول متغیر فشرده‌سازی (کد) کنیم.

برای کد کردن، هر کسی می‌تواند در نظر بگیرد چند بیت کد کنیم، تصمیم متفاوتی بگیرد.

بهترین و بهینه‌ترین روشی که برای کدینگ با طول متغیر انجام می‌شود، روش کدگذاری هافمن است.

روش هافمن : (Huffman Coding)

شامل دو مرحله است :

① کاهش (Reduction)

② تقسیم (Splitting)

A=25

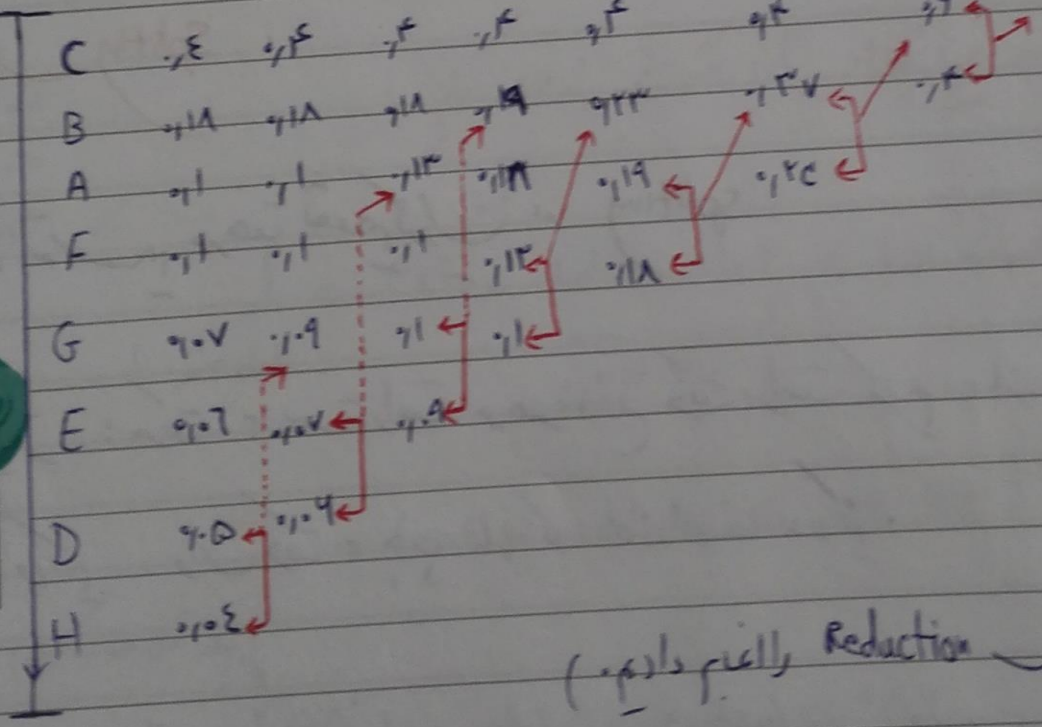
① مرحله Reduction :

لیست سمبل‌ها به ترتیب نزولی (از حیث احتمال آنها) مرتب شده و دو تا از کم احتمال ترین سمبل‌ها با هم ترکیب می‌شوند.

* احتمال‌های مربوط به هم جمع می‌شوند و یک سمبل جدید ساخته می‌شود و دوباره به ترتیب نزولی مرتب می‌شوند و این کار ادامه پیدا می‌کند تا فقط دو سمبل باقی بماند.

(احتمال‌ها را دنبال کنید)

A	B	C	D	E	F	G	H
۰٫۱	۰٫۱۸	۰٫۴	۰٫۰۵	۰٫۰۶	۰٫۱	۰٫۰۷	۰٫۰۴



(مرحله Reduction را تمام دادیم.)

یعنی :

سمبل‌های با احتمال رخداد بالا (فراوانی بالا) ← طول کدگذاری کوتاه

سمبل‌های با احتمال رخداد پایین (فراوانی کم) ← طول کدگذاری بلند

Symbol #1 $p_1 = 2$

Symbol #2 $p_2 = 0$

⋮

Symbol #n

حداقل کردن برای طول کدگذاری (خلاصه)

امکان حداقل طول کدگذاری بهینه

→ Reduction

→ Splitting

۲- مرحله Spliting :

قواعد ترسیم درخت تصمیم‌گیری در spliting

مقادیر کوچکتر : حرکت به سمت راست و هر موقع حرکت به راست کردیم، از بیت '۱' استفاده می‌کنیم. برای حرکت به چپ از کد '۰' استفاده می‌کنیم.

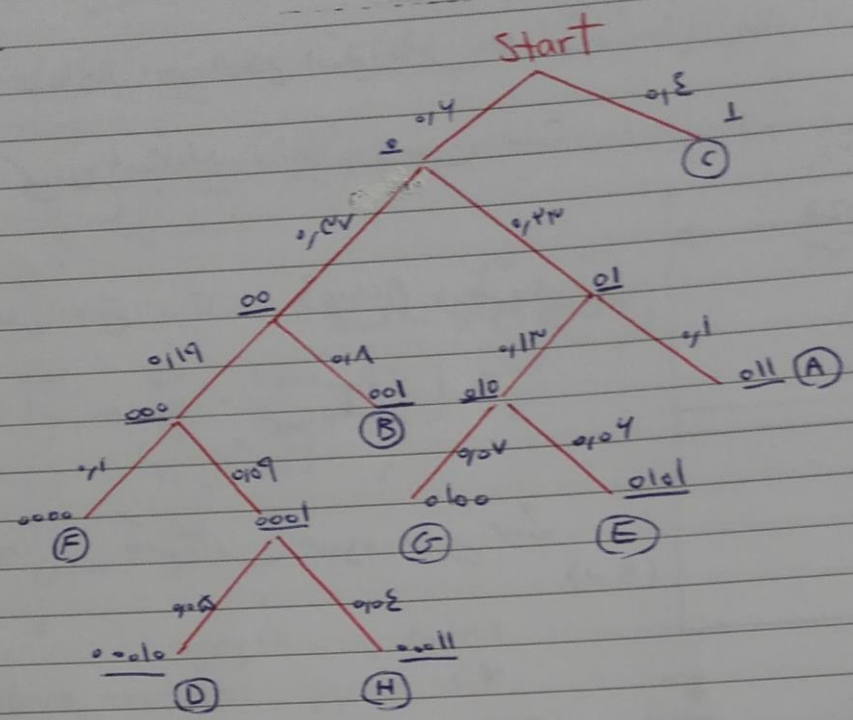
همه‌ی مقادیر را از روی نمودار مرحله reduction می‌نویسیم :

ادامه در صفحه بعد ←

P=27

۱۲۷ / ۲۳۸

A	۰٫۱
B	۰٫۱۸
C	۰٫۴
D	۰٫۰۵
E	۰٫۰۶
F	۰٫۱
G	۰٫۰۷
H	۰٫۰۴



کدهای حاصل:

A	۰۱۱
B	۰۰۱
C	۱
D	۰۰۰۱۰
E	۰۱۰۱
F	۰۰۰۰
G	۰۱۰۰
H	۰۰۰۱۱

حال می خواهیم کارایی این الگوریتم را حساب کنیم:

$$l = \underbrace{۴\times۱}_{\text{۴ بیت}} + \underbrace{(۰٫۱+۰٫۱۸)\times۳}_{\text{دو تا سه بیتی}} + (۰٫۱+۰٫۰۷+۰٫۰۶)\times۴ +$$

$$+ (۰٫۰۴+۰٫۰۵)\times۵ = ۲٫۴۱ \text{ bit}$$

(جلسه قبل حساب کردیم)

$$\eta = \frac{H}{l}\times۱۰۰ = \frac{۲٫۳۵}{۲٫۴۱}\times۱۰۰ = ۹۷٫۱۷\,\% \simeq ۹۸\,\%$$

که در الگوریتم قبلی ۸۵ درصد بود و این الگوریتم کارایی بالایی دارد.

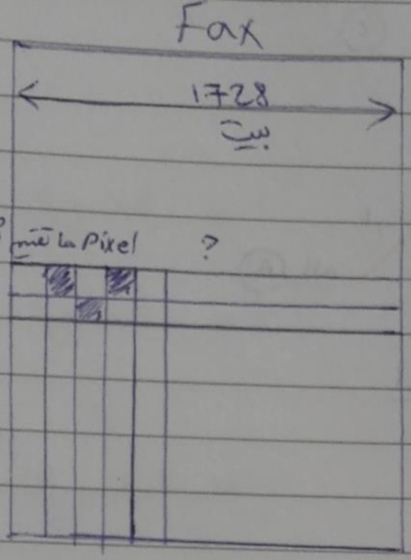
* یکی از کاربردهای کدینگ هافمن، فکس است.

الگوریتم هافمن هم unique است و هم دی‌کدینگ آن می‌شود. (هر دو قرار)

ITU : سازمان بین‌المللی استانداردهای مخابرات

P=28

group3 یک نوع استاندارد در این سازمان است.



یعنی طبق استاندارد group3 در هر میلی‌متر 3.85 line/mm

۳٫۸۵ خط باید اسکن شود. line / line

این استاندارد برای فکس‌های سیاه و سفید می‌باشد. (B W)

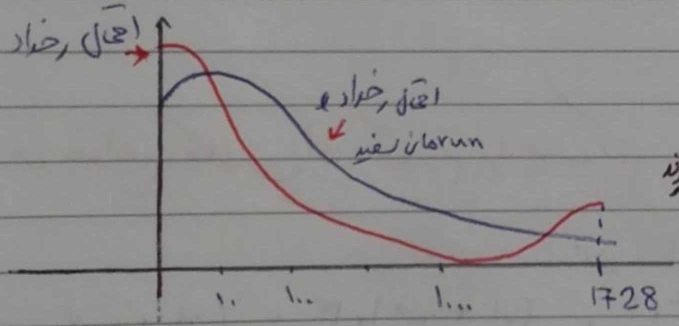
۲Mbit per A4 page
۴٫۸ kbit/s telephone modem } transmision take 7 min/page

روش محاسبه = $\frac{۲ \text{ Mbit}}{\text{page}} \times \frac{1s}{۴٫۸ \text{ kbit}} = \frac{۲۰۰۰ K}{۴٫۸ K} = ۴٫۱۶ \simeq ۷ \frac{min}{page}$

B	2	2
W	5	9

تعداد رنگ‌ها

در استاندارد ITU چندین سند سیاه و سفید را اسکن کردند و دیدند که به نمودار زیر می‌رسند:



جدول برای سفید

6W	1110
9W	10100
⋮	⋮

جدول برای سیاه

2B	11
5B	1010

مثال برای کدگذاری هافمن :

منبعی داریم که ۶ تا سمبل تولید می‌کند به نام‌های S_1 , S_2 , ... , S_6 که فراوانی‌های آنها را به صورت مقابل داریم:

	فراوانی
S_1	۵
S_2	۷
S_3	۱۰
S_4	۱۵
S_5	۲۰
S_6	۴۵

مطلوبست ترسیم جدول مربوط به مرحله‌ی Reduction در الگوریتم هافمن و سپس انجام مرحله‌ی Spliting و تعیین Codeword ؟

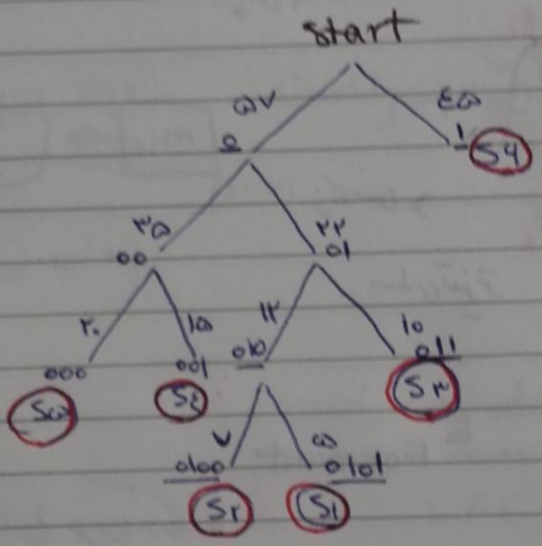
* فرق این مثال با قبلی این است که به جای احتمال، فراوانی داده شده است. می‌توانیم هم براساس احتمال و هم براساس فراوانی رسم کنیم که ساده‌تر است.
(برحسب فراوانی)

① Reduction

لزومی ندارد جمع آنها ۱۰۰ شود

S_6	۴۵	۴۵	۴۵	۴۵	۵۷	۱۰۲
S_5	۲۰	۲۰	۲۲	۳۵	۴۵	×
S_4	۱۵	۱۵	۲۰	۲۲	×	
S_3	۱۰	۱۲	۱۵	×		
S_2	۷	۱۰	×			
S_1	۵	×				

② Spliting



کدهایی که بدست آمد:

S_6	1
S_5	000
S_4	001
S_3	011
S_2	0100
S_1	0101

با روش شانون فانو هم می‌توان حل کرد.

کدینگ منبع (فشرده‌سازی) سیگنال‌های صوتی:

(*)

برای صوت استانداردهای مختلفی وجود دارد مثل:
- کیفیت خط تلفن → نرخ آن ۸ کیلوهرتز است.
- GSM
- CD, DVD play

(*) ادامه ← یعنی $8^K \times 10$ بیت، در هر ثانیه ۸۰،۰۰۰ بیت باید ارسال کنیم.

P=30

۱۳۰ / ۲۳۵

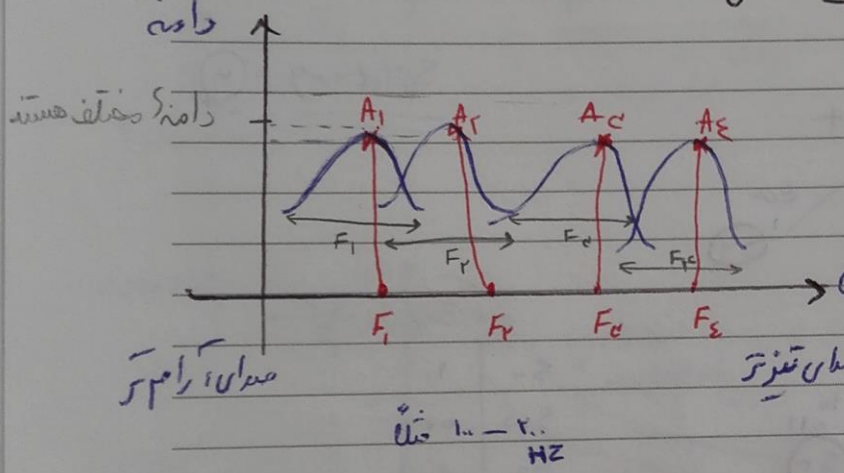
voice + coder } سیستم یا الگوریتمی که داده گر صوتی را کد می کند ، نام دارد vocoder

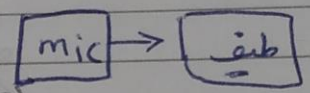
vocoder براساس ساختار تولید صدای انسان کار می کند.

Formant
Pitch
حروف صدادار (aieou)
→ فیلتر مکانیکی مدل تولید صدای انسان
→ شامل حنجره ، لب ، دهان + نویز در یک کانال
بازسازی صدای انسان در مخرج

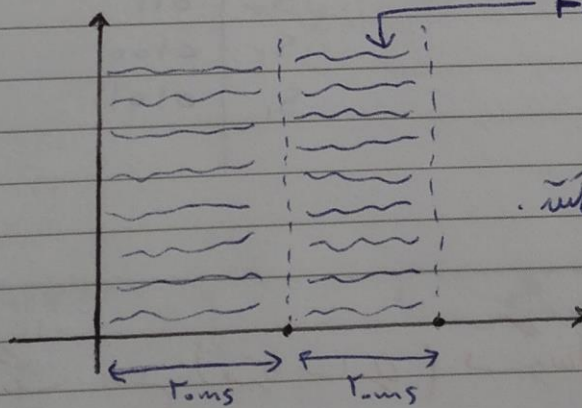
صدای انسان دارای Formant ی مختلف است.

Formant ها چیه؟





20ms

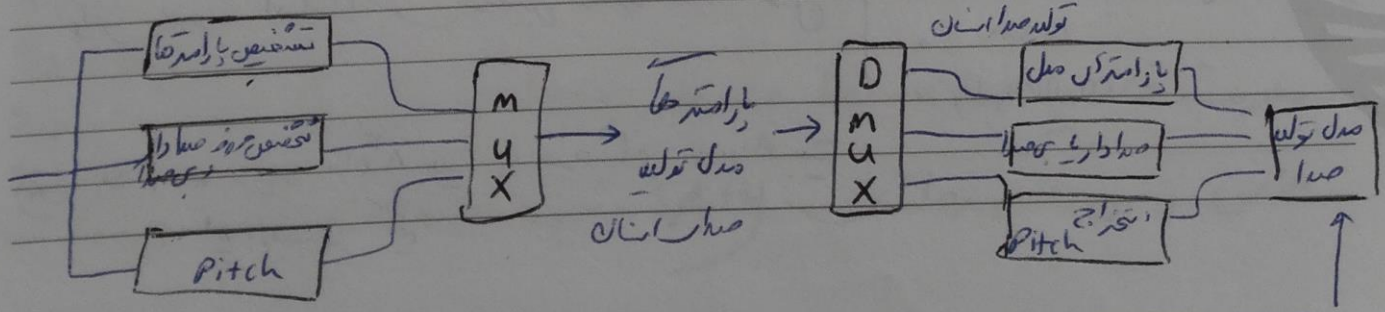


می آیند F ها را حس می کنند:

علاوه بر بالا ، می آیند Pitch را تعیین می کند.

که همان زیر یا بم بودن صداست ، یعنی

می آورند.



ورودی این مدل ، که این نویز توسط حنجره تولید می شود.

روش‌های مختلفی برای فشرده سازی صوت وجود دارد :

- LPC
- MELP
- CELP

جلسه ۵

فشرده سازی و کدینگ منبع صوت :

* افزونگی (Redundancy)

فشرده سازی صوتی مطرح شوند ← در کدام حوزه‌ها برای کاربردهای تلفنی

- Toll quality Voice (کیفیت مخابراتی)
- hi fi (Hi Fi)
- CD Quality

کاربردی صوتی ←

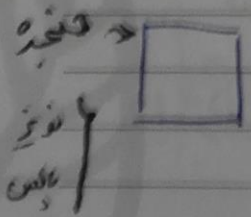
کیفیت برای همه‌ات Quality

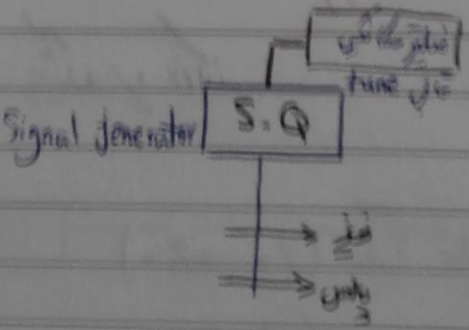
نرخ فشرده سازی

channel vocoder
LPC
} Low bit rate vocoder

آخرین نرخ فشرده سازی که توسط NATO انجام شده است ۱۰۰ bit/Sec باشد.

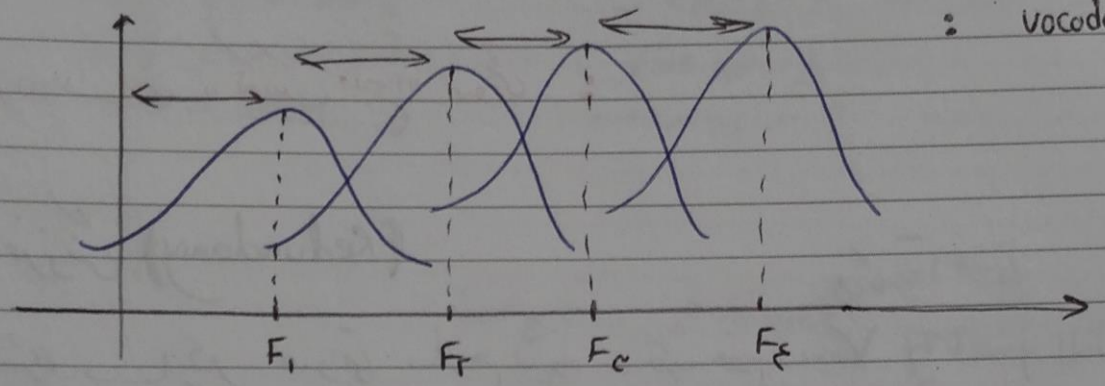
vocoder = مدل تولید صدای انسان



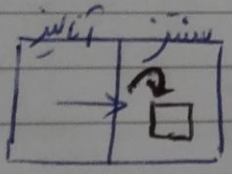


تفاوت سه مفهوم:
- voice هر صدا با مفهوم یا بی مفهوم
- speech گفتار
- audio مجموع انواع صداهایی که گوش می‌شنود [0-20 KHz] در پژوهش DC

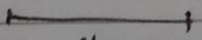
ساختار vocoder:



صوت دارای چهار Formant است. (طبیعی)



طبیعی ۲۰ ms ؟

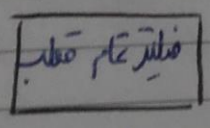


* پارامترهای صوت تقریباً ثابت باقی می‌ماند.

- Formant
- Pitch
- voice

روش فشرده سازی صوت LPC: (Linear Predictive Coding)

پارامترهای صوت را پیش‌بینی می‌کند.



$$A \over (Z-Z_0)(Z-Z_1)\cdots(Z-Z_n)$$

قطب‌های فیلتر $(Z_0 \to Z_n)$

p=33

در روش LPC به این صورت عمل کنند:

20–60 ms ← بازه مورد نظر را گرفته، در این بازه ضرایب فیلتر را محاسبه کنند.

برای { ۲۰ ms / ۶۰ ms } هرچه این زمان بیشتر شود، نرخ فشرده سازی کاهش پیدا کند.

این فشرده سازی با کیفیت پایین انجام می‌شود.

* اگر نمونه برداری یک سیگنال با تعداد بیت کم انجام شود، کیفیت آن کاهش می‌یابد « به دلیل نویز کوانتیزاسیون »

در نمونه برداری ۴ بیتی $Noise = \frac{۲}{۱۶} = \frac{۱}{۸} = ۰,۱۲۵$

در نمونه برداری ۱۰ بیتی $Noise = \frac{۲}{۲^{۱۰}} = \frac{۲}{۱۰۲۴} \equiv ۰,۰۰۲$

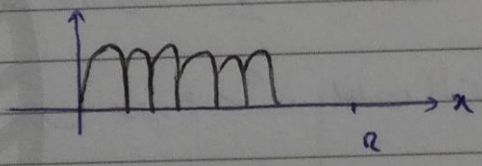
جاهایی که دامنه سیگنال زیاد باشد، تعداد بیت‌های بالا، فشرده خواهیم کرد.

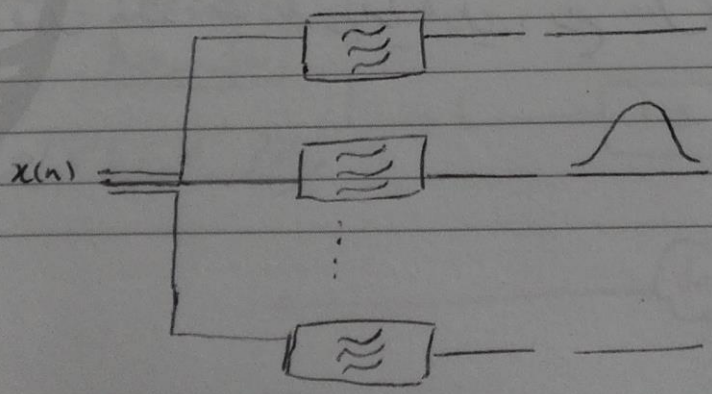
جایی که دامنه سیگنال ضعیف باشد « « کم فشرده خواهیم کرد.

فشرده سازی صدا با کیفیت بالا : (Hi Fi)

این روش معمولا در حوزه‌ی فرکانس پیاده سازی می‌شود. محدوده فرکانسی DC – ۲۴ KHZ

۲ تا ۱۶ زیر باند Subband به کمک بانک فیلتر (Filter Bank) « مجموعه از چندین فیلتر کنار هم » تقسیم بندی می‌شود.





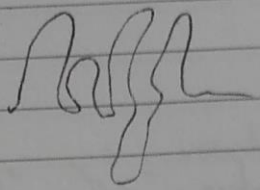
فیلتر بانک، سیگنال ورودی را به زیرباندها (Subband) ها تفکیک می‌کند.

معمولاً فیلتر بانک، به کمک DFT (تبدیل فوریه) ساخته می‌شود. p=34

"Discrete Fourier TranForm"

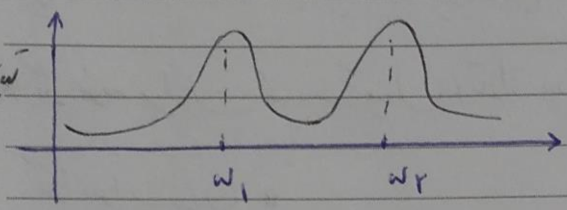
«این بحث را در DSP هم خواهیم داشت ولی اینجا هم توضیح می‌دهد»

$x(t) = A\sin w_1 t + B\sin w_2 t$ ⇒ به شکل درهم



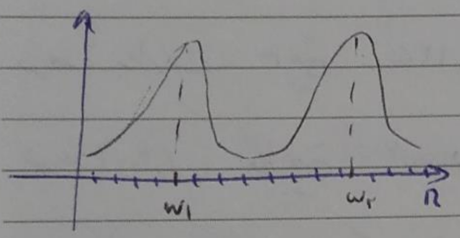
این مثال ناقص است و مبهم

تبدیل فوریه پیوسته



$X(e^{jw}) = \sum x(n) e^{-jwn}$

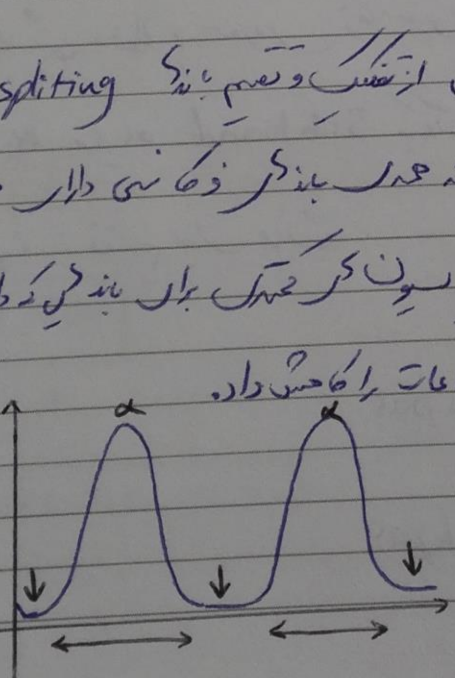
حال از این شکل حاصل، (مثلاً) ۱۴ تا نمونه برداری می‌کنیم.



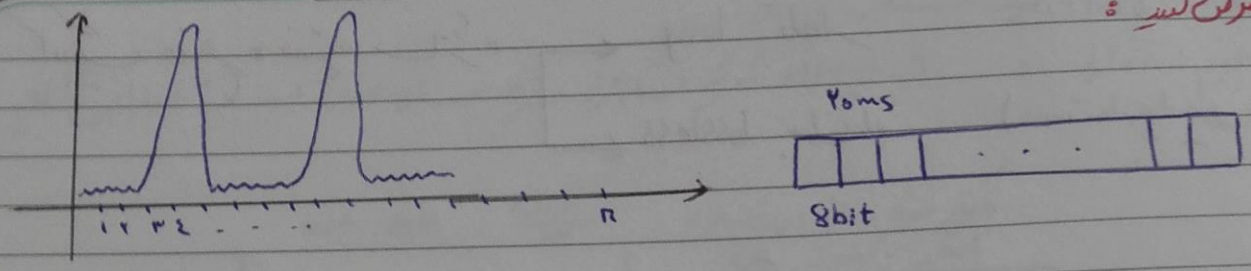
0	$0 - \frac{R}{14}$
A	$\frac{R}{14} - \frac{2R}{14}$
0	⋮
⋮	⋮
B	$\frac{5R}{14} - \frac{4R}{14}$
⋮	⋮
0	R

چون پیاده‌سازی دیجیتال DFT ساده است، معمولاً فیلتر بانک‌ها به کمک DFT (FFT) پیاده‌سازی می‌شود.

خلاصه: Sub Band Coding (SBC) از تفکیک و تقسیم باندها «band spliting» استفاده می‌کند. باید توجه داشت که اینگونه نیست که همه باندهای فرکانسی دارای محتوای انرژی برابر باشند لذا، می‌توان نرخ کوانتیزاسیون کمتری برای باندهایی که دارای انرژی کمتری هستند، بکار برد و به این طریق، نرخ تبدیل اطلاعات را کاهش داده.



تعریف کلمه :



در ۲۰ میلی ثانیه ⟶ ۱۴×۸

در ۱ ثانیه ⟶ x

$x = \frac{1000}{20} = 50$

$\hookrightarrow = \underset{S.B}{50 \times 14 \times 8} = 4400 \ \frac{bit}{sec}$

حال در حالتی که سیگنال ضعیف است، بیت کمتری را در نظر می‌گیریم :

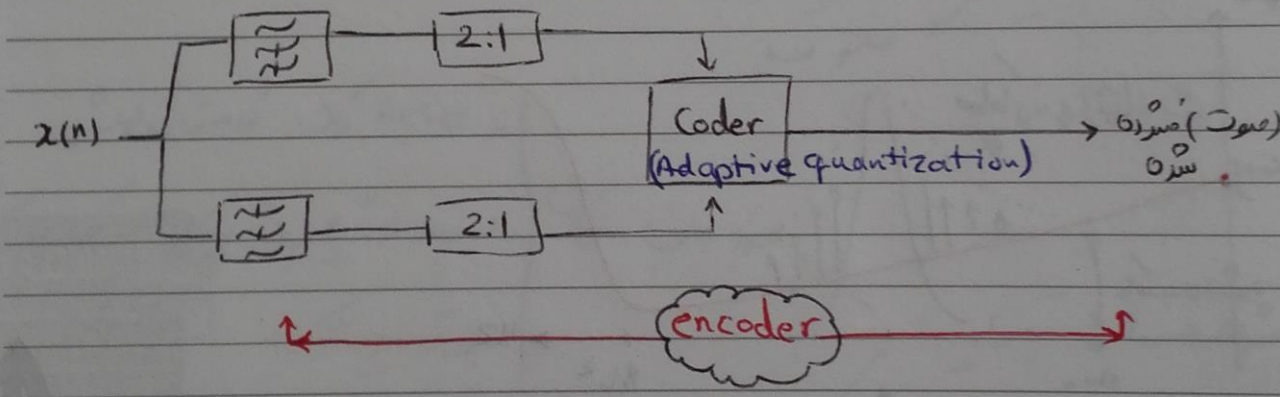
$\rightarrow (2 \times 8 + 14 \times 3) \times 50 = 58 \times 50 = 2900 \ \frac{bit}{sec}$

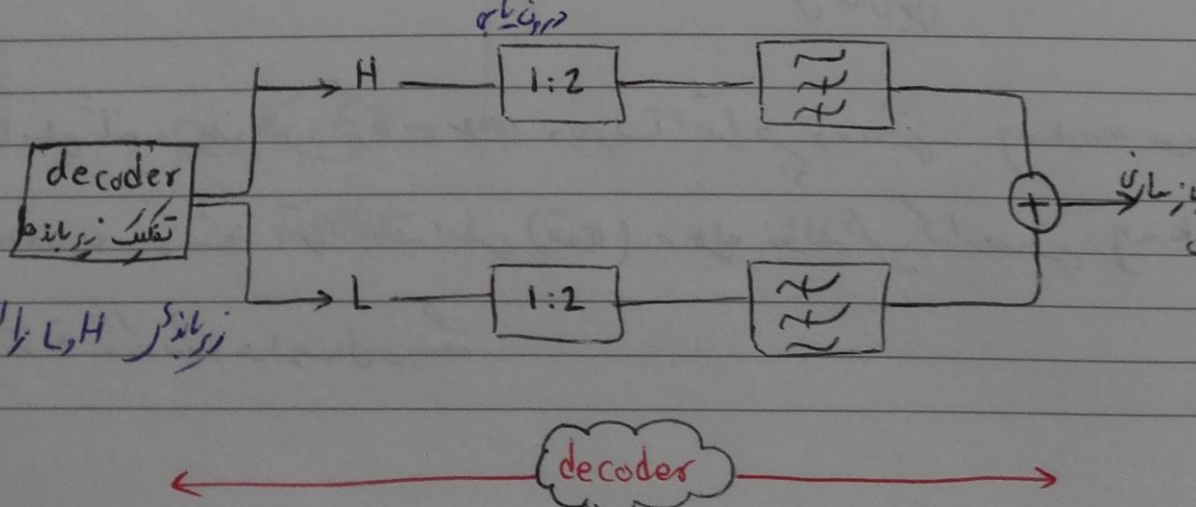
(2 ↑ bit) (14 ↑ bit)

که چشمگیر نرخ فشرده سازی کمتر شد.

بلوک دیاگرام فشرده ساز SBC :

DC - 3.4 KHz → تقسیم شده است SB به ۲





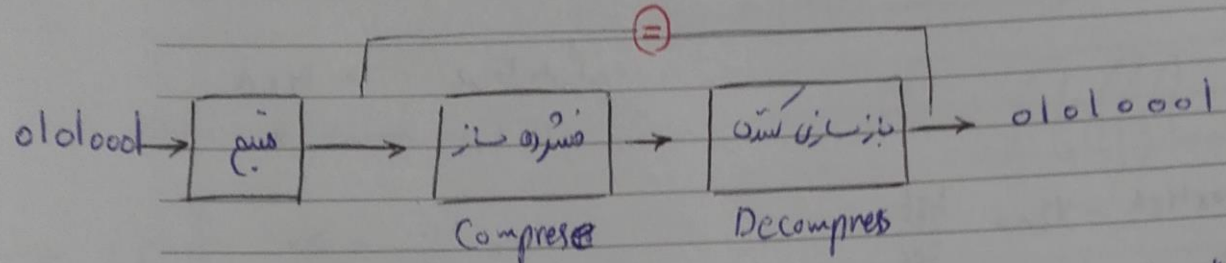
زیرباند کد H و L را استخراج میکند.

کدینگ منبع «فشرده سازی»

p=36

lossy تلفات ↓bitrate ↑تلف

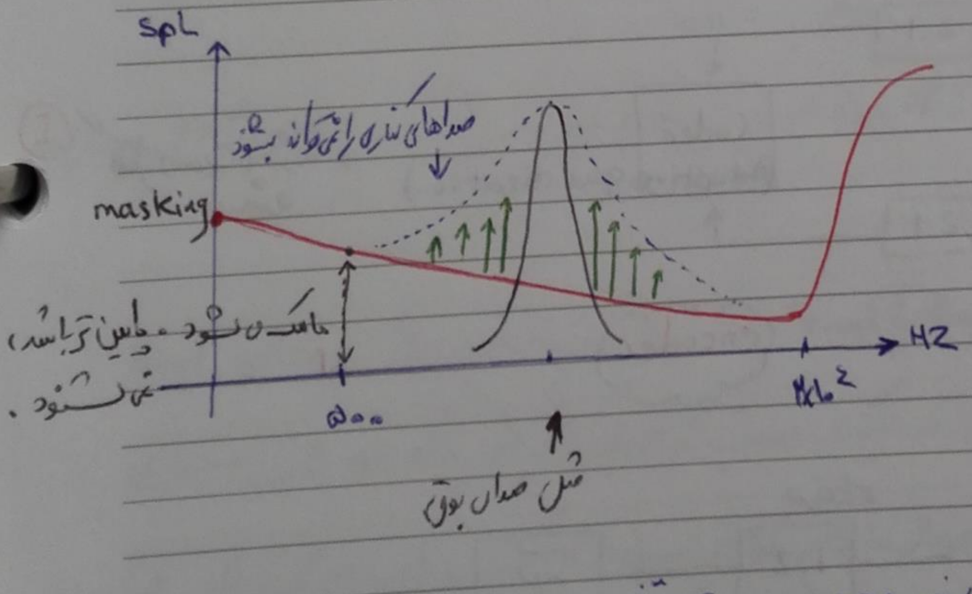
lossless بدون تلف (دوست داریم اینطور باشد)



وقتی که دقیقاً آنچه وارد کردیم تا فشرده شود، با آنچه بدست آوردیم برابر باشد، این نوع lossless است.

MPEG «Moring Picture Expert group»

گوش انسان دارای پاسخ درکی (preceptual Response) خاصی است.



هر ۲۰ میلی ثانیه، مؤلفه‌ها در فرکانس موجود در صورت استخراج و منحنی masking مربوطه update می‌شد. تنها مؤلفه اصلی (قوی) به همراه مؤلفه‌هایی که از منحنی masking بزرگتر هستند، کد شده و ارسال می‌شوند.

p=37

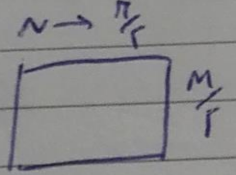
طبق استاندارد ، MPEG-1 که یک استاندارد فشرده سازی صوت است ، دارای ۳ لایه است .

	bit rate $\frac{Kbit}{sec}$	فاکتور فشرده سازی در مقایسه با تکنیک با کیفیت CD
Layer I	384	4
layer II	192	8
Layer III	128	12

بیشترین نرخ فشرده سازی برای لایه I است .

نرخ فشرده سازی CD است $1,4 \frac{Mbit}{Sec} \rightarrow \frac{٣٨٤}{١١٤} = ٤$ ↑

فشرده سازی تصویر 8



تصویر → عکس still
تصویر → تصویر متحرک video
تصویر → حالت صدا

- Resize کردن ، کاهش رزولوشن
- استفاده از شبیه بودن پیکسل ها در یک همسایه [illegible]
Run time coding
- برای تصاویر متحرک ، تعداد فریم + اضافه تصویر ارسال شود .

} حوزه زمان - مکان

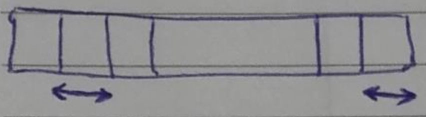
روش فشرده سازی تصویر :
- حوزه زمان - مکان
- حوزه تبدیل → FT ، تبدیل فوریه
wavelet

جلسه ۶

P=38

کدینگ و فشرده‌سازی Stringها (فایل داده) ZIP

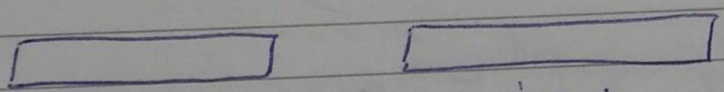
روش‌های کدکردن و فشرده‌سازی stringها کاملاً به روش‌های استفاده شده در صوت (vocoder) و نیز متن، متفاوت بوده و براساس حافظه منبع می‌باشد.



بخش‌هایی از String به صورت تکراری استفاده می‌شوند.

روش lempel-ziv

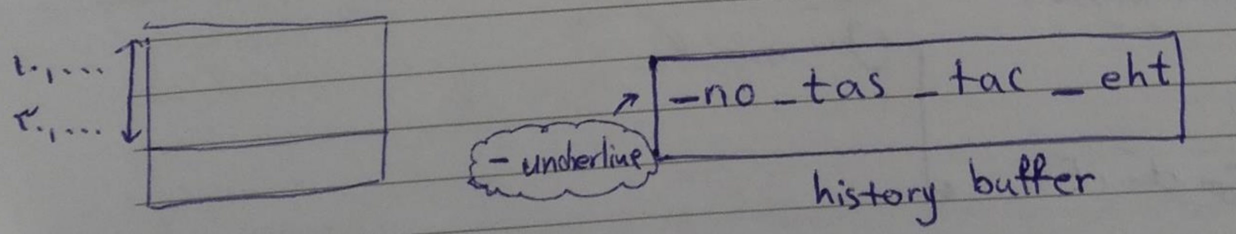
از یک بافر که مربوط به تاریخچه (History) داده‌ها می‌باشد، استفاده نموده و به دنبال تطابق‌ها (match) می‌گردد. هر وقت که یک تطابق (match) یافت شد، موقعیت و طول آن در بافر ارسال می‌گردد.



بافر (تاریخچه)

توجه: در روش lempel ziv، بافر ثابت است و از اول تا آخر داده‌ها را ذخیره می‌کند، ولی در روش lempel ziv welch (LZW)، بافرها ؟

مثال: پیام (string) زیر را که بافر تاریخچه داده شده، فشرده‌سازی کنید؟



history buffer

قبلاً استخراج و ارسال شده است.

فرض کنید پیام tam_eht را فشرده‌سازی کنیم. (و ارسال)

"مکان شروع خواندن string"

روز خبرنگار

p=39

وقتی شروع می‌کنیم از ابتدا به خواندن پیام می‌بینیم، _eht در بافر تطابقی وجود دارد، پس به جای ارسال 8،8،8،8 بیت یعنی ۳۲ بیت، یک word را ارسال می‌کنیم.

| Flag | | | | | | | |

⊛ | ⊠

⊛ آدرس تطابق پیدا شده

⊠ طول تطابق

Flag: اگر ارسالی تطابق با بافر دارد ①، اگر ارسالی تطابق با بافر ندارد ⓪

اینجا هم داریم آدرس می‌دهیم که همان بعد از اشاره‌گر در بافر است که آدرسش را ارسال می‌کند.

حال می‌بینیم که تطابق در آدرس 11 ام رخ داده است. پس در مکان آدرس ⊛ آدرس بیتی 11 را قرار می‌دهیم.

Flag

| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |

11 | ۴

یک می‌شود چون تطابق وجود دارد

طول تطابق هم ۴ بیت است.

شروع از:
- 0000 ← ۱
- 0001 ← ۲
- 0010 ← ۳
- 0011 ← ۴ ✓

آپدیت می‌شود

_eht_no_tas_tac

به کاراکتر بعدی که می‌رسیم، m است. از پیام tam_eht چون تطابقی پیدا نکرد، پس flag را صفر می‌کند و کد اسکی m را می‌فرستد.

| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |

کد اسکی m

حال به ta می‌رسیم، ta در دو جا آمده فرقی نمی‌کند آدرس کدام را ارسال کنیم.

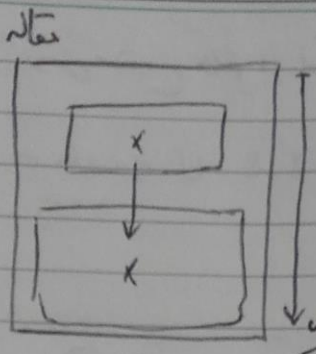
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

Flag | ۴ بیت | ۲ = طول

در کل ۲۵ بیت ارسال کردیم که اگر خود پیام را می‌فرستادیم، باید $7\times8=56$ تا باید ارسال می‌کردیم.

$$\frac{56}{25} \simeq 2.24$$

سؤال: چرا بافر را update می‌کنیم؟



احتمال کم تر

روش کدینگ String چون مبتنی بر search در History Buffer است، روش پیچیده و زمان بر است. «فشرده سازی online امکان پذیر نیست»

سؤال: آیا احتمال از دست دادن اطلاعات در بافر History وجود دارد؟ خیر چون lempel Ziv یک روش فشرده سازی بدون اتلاف lossless است.

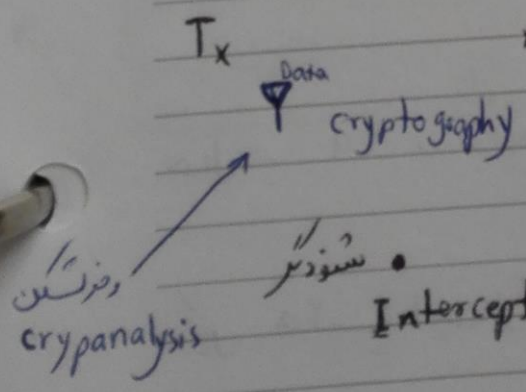
منبع جدید:

«رمزنگاری داده ها» «Data Encryption»

رمزنگاری با موضوعات زیر سروکار دارد:

① محرمانگی و خصوصی بودن «Secrecy & Privacy»

جلوگیری از دریافت اطلاعات پیام، توسط گیرندگان «دریافت کنندگان غیرمجاز»



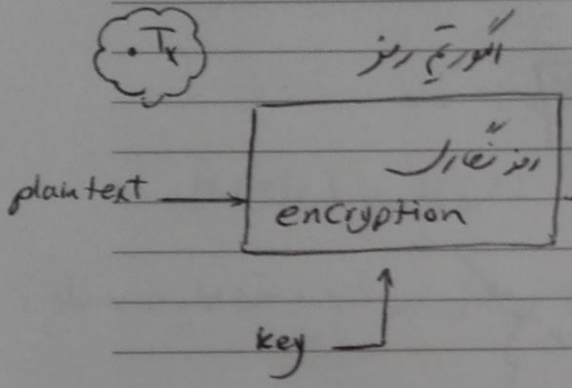
ادامه ←

۲) احراز هویت "Authentication"

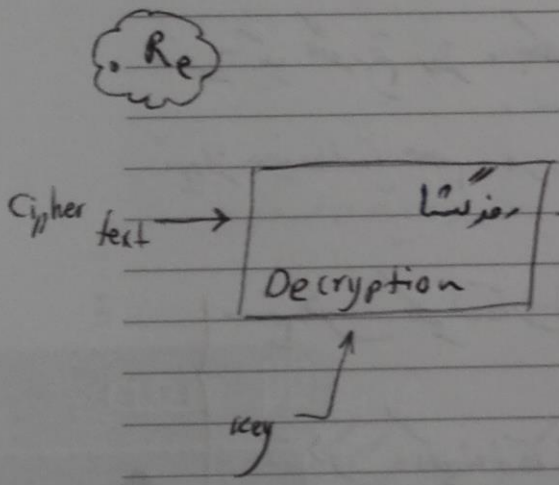
باید روشی در اختیار داشته باشیم که مبداء پیام را تایید کند.

۳) صحت Integrity

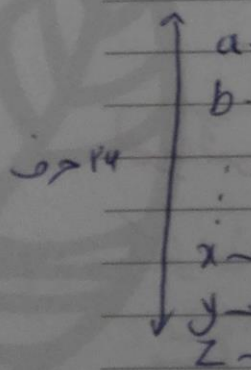
تضمین اینکه پیام دستکاری نشده است.



کلید key ها در رمزنگاری متقارن فقط در صورتی اعلام میشود که فرستنده و گیرنده برابر باشند.



رمز سزار: هر حرف را با سه حرف آن طرف تر تبدیل میکرد.



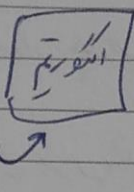
فرستنده → [+3] → رمزنگار

گیرنده → [-3] → کشف

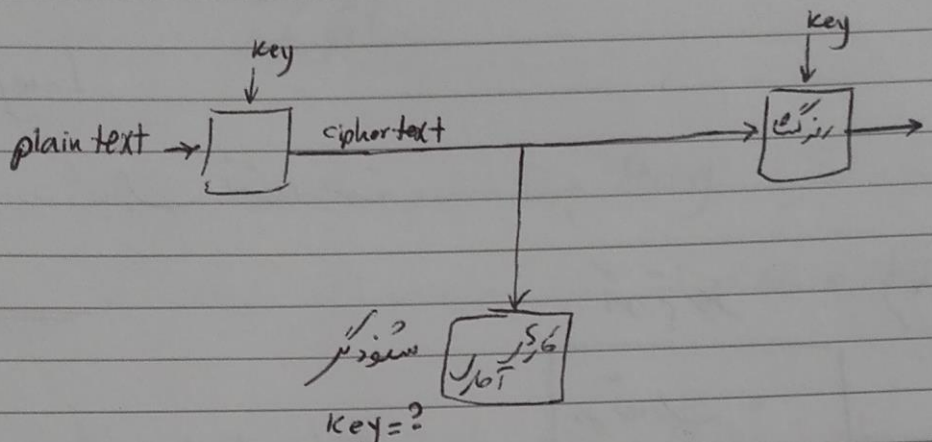
مثلا book → errn

رمزشکن ← الگوریتم رمزگشایی ندارد.

ابزار کار رمزشکن، استفاده از آمار است.



الگوریتم رمز مثل DES AES



مثلا

جبر رمز تکرار e → k جبر رمز تکرار

fghijk

کلید رمز با ۶ جابجا شده است

امنیت یک الگوریتم رمز، به کلید آن وابسته است.

یکی از موضوعات مهم در رمزنگاری، توزیع کلید است. "Key Distribution".

سیستم کلید متقارن "Symetric key" یا کلید خصوصی "private key"

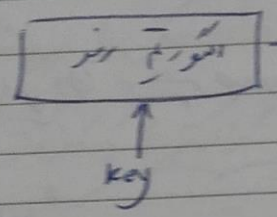
سیستم کلید نامتقارن "Asymetric key" یا کلید عمومی "Public key"

این سیستم‌ها مشکل توزیع کلید ندارند.

محرمانگی "Secrecy"

یک معیار برای محرمانگی، عبارت "قدرت رمزنگاری" عبارت است از: اطلاعات متقابل (mutual Information)

$I_m(T,C) = H(T) - H(T|C)$



$H(T)$ آنتروپی یا عدم قطعیت موجود در متن اولیه

$H(T|C)$ آنتروپی " " " " به شرط داشتن متن رمز شده.

محرمانگی کامل (perfect Secrecy)

وقتی است که متن رمز شده و متن اولیه کاملاً از هم مستقل باشند.

$I_m(T,C) = 0$ ⟹ $H(T|C) = H(T)$

↓

به این معنی است که: هیچ ارتباطی بین متن رمز شده و متن اولیه وجود ندارد.

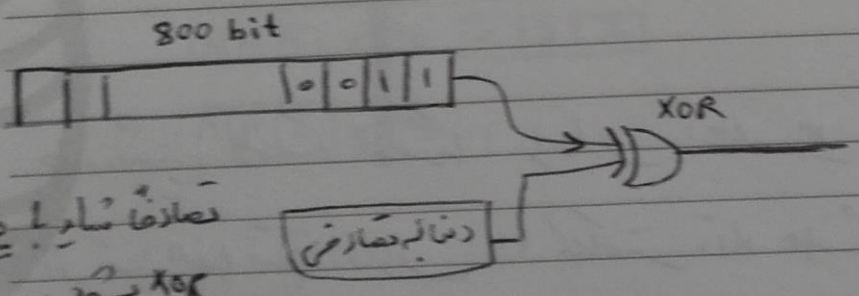
رمزهای one-time-pad

شرایط محرمانگی کامل را دارند (vernam Cipher)

هر کلید فقط یکبار استفاده میشود و بعد از استفاده کردن، دور ریخته میشود.

loo char ⟹ loo key

یک روش عملی برای پیاده‌سازی این الگوریتم، این است که کد باینری هر کاراکتر بیت به بیت با یک دنباله کاملاً تصادفی در modeul-2 جمع شود.

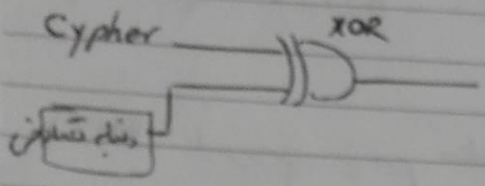


تصادفا شانس $1 = 0$ یا $1 \neq 1$ XOR شود.

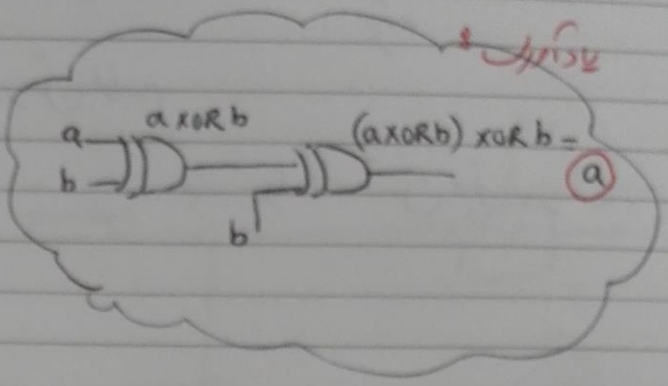
XOR	mod-2
1 1	0
0 1	1
1 0	1
0 0	0

ادامه ←

رمزگشایی آن، چه صورت است؟



یادآوری:



جلسه (۷)

هدف این جلسه: می‌خواهیم رمزشکنی را برای افراد مبتدی شروع کنیم.

جانشینی (substitution)

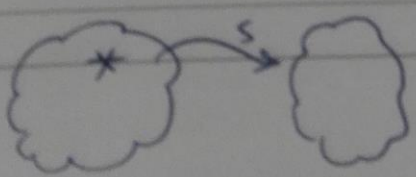
جابه‌جایی (permutation)

رمزشکنی، حملات Cryptographic Attact محور براساس تحلیل آماری متن رمزشده (Cypher text) است. لذا یک رمز مطلوب باید الگو و ساختارهای موجود را محو کند (absure) یا از بین ببرد (remove).

حد راه‌کار برای محو یا از بین بردن الگو:
① استفاده از کلیدهای صنعی
② استفاده از جابه‌جایی و جانشینی

جانشینی باعث محو شدن راحت‌تر آماری بین کاراکترهای (Cypher text) می‌شود.

اگر یک الفبا مثلاً دارای ۲۶ کاراکتر تعداد جانشینی‌های ممکن برابر است با:



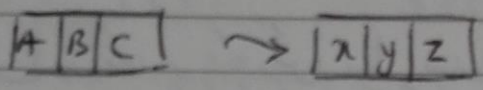
$$26 \times 25 \times \dots \times 1 = (26!)$$

Brut Force 8 عملی که در آن رمزشکن تمام حالت های ممکن را بررسی می‌کند.

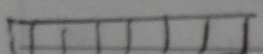
* این روش دیگر به این صورت است که به جای جایگزینی یک کاراکتر با یک کاراکتر دیگر، مجموعه‌ای از آرایه‌ای را جایگزین کند. در واقع در روش‌های پیچیده‌تر برای جانشینی معمولاً آرایه‌ها یا string ها، n کاراکتری را با آرایه‌ای n کاراکتری دیگر جایگزین کرد.

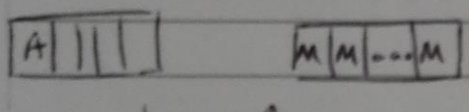
* فرض کنید الفبای مورد استفاده در plain Text دارای m کاراکتر باشد و آرایه‌ها n تایی در نظر گرفته شده باشد.

مثلاً



می‌خواهیم تعداد حالات را بدست آوریم:

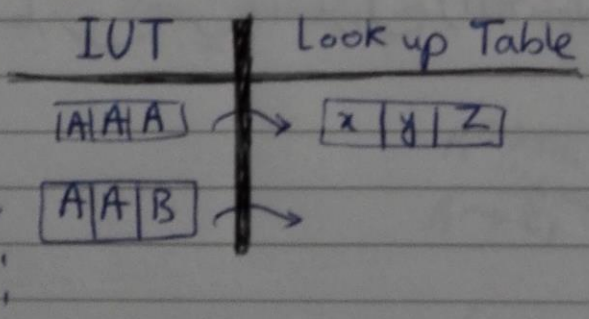
فرض کنیم  n تا از plaintext را با آرایه‌ای n تایی cipher، چون تکرار جایز است؛ map کنیم،



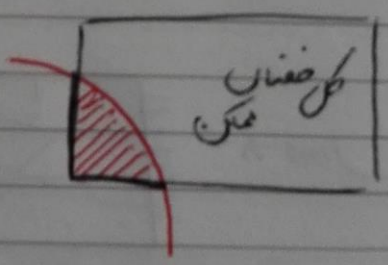
مثلاً $(26)^4$ که عدد خیلی بزرگی است و کار رمزشکن را مشکل می‌کند.

تعداد حالات M^n

* الگوریتم‌هایی وجود دارد که برای جانشینی آرایه‌ای n تایی وجود دارد، در واقع کلید رمزنگاری می‌باشند.



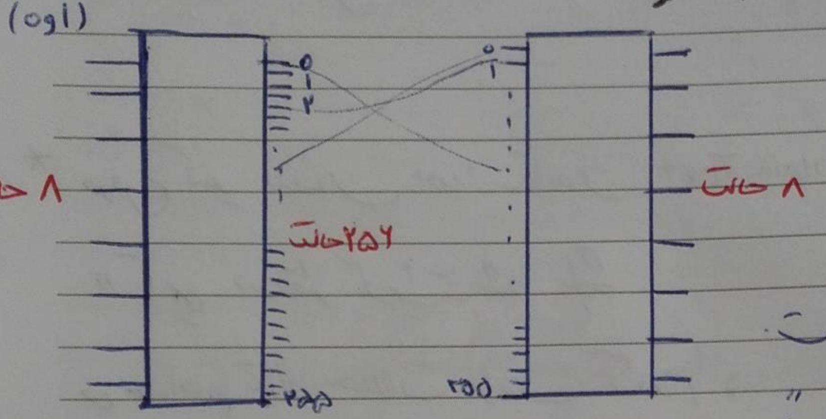
تعداد حالات $(26)^3$



به روش های رمزنگاری جانشینی اصطلاحاً S-box هم گفته می شود.

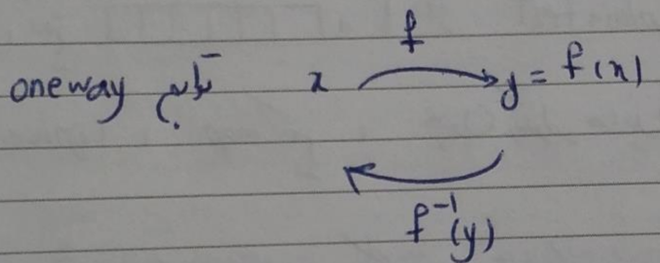
p=46

مثال: کل تعداد حالات ممکن برای یک رمز S-box که یک بایت را به عنوان ورودی دریافت می کند، محاسبه کنید؟



(۲۵۶)!

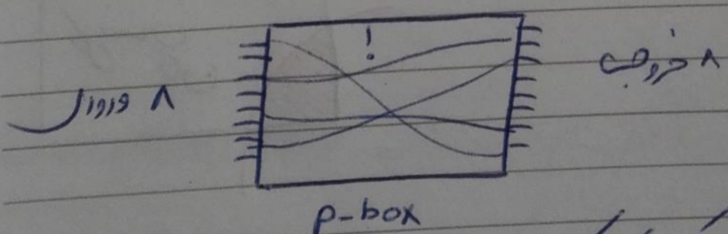
برای رمزنگار، کارش بسیار ساده است و بالا بودن پردازش سنگین لزوماً روی کار رمزنگار تاثیر ندارد.



روش دوم: جابجایی (permutation)

یا منظم کردن دوباره کاراکترها (rearrangement).

رابطه بین آمارگان (plaintext) و متن رمز شده (ciphertext) پیچیده تر شود.

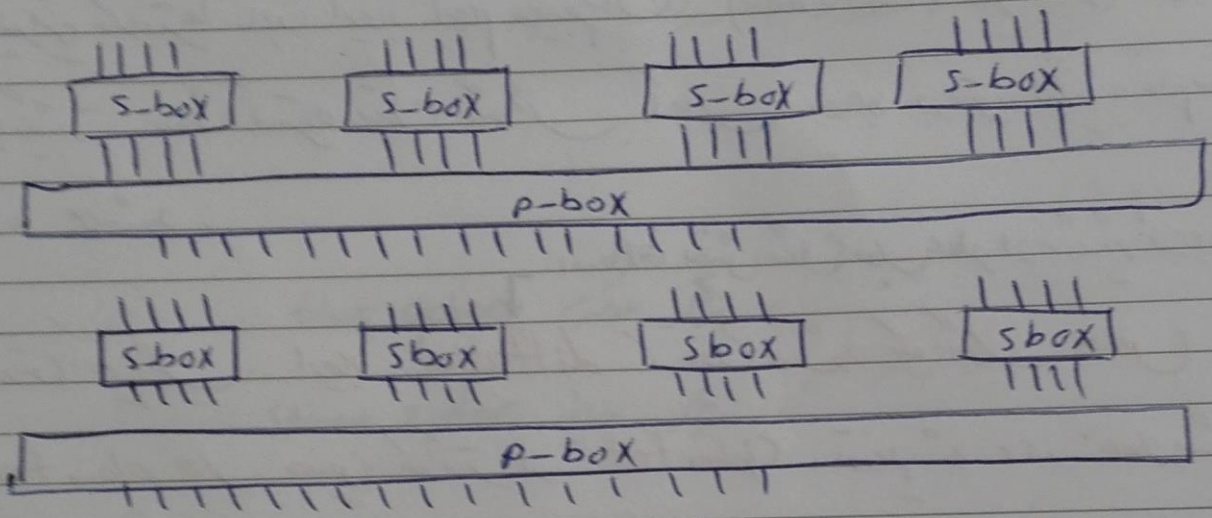


لازم به ذکر است، جابجایی آمارگان یک کاراکتر منفرد را تغییر نمی دهد ولی آمارگان مشترک بین کاراکترها تغییر می کند.

نمونه ای از یک سیستم رمزنگاری مبتنی بر جانشینی و جابجایی مثل:
(AES, DES, IDEA)

توجه: تعداد ورودی و خروجی های s-box و p-box باید برابر باشند.

sbox ها و p-box همگی ثابت و معلوم هستند. هر طبقه Raund نام دارد.



تذکر داخل پرانتز

plain text → [رمزنگار] → cipher text

دو حالت مختلف که به رمزشکنی وجود دارد:

① فقط متن رمزشده را درست داریم — Cipher text only

② ساده‌تر از بالا — know plain text

یعنی مثلاً
$P_1 \rightsquigarrow C_1$
$P_2 \rightsquigarrow C_2$

③ سیستم رمزنگار در دسترس است. فقط کلید را نداریم (Key=?)

تعداد طبقه‌های s-box و p-box بستگی دارد به پیچیده بودن حالت رمزنگار

برای ایجاد محرمانگی «از دیدگاه شانون»

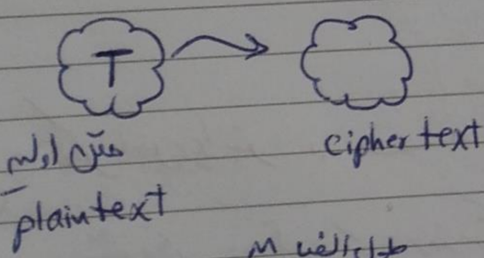
← ابهام ، درهمی (Confusion) ← s-box →

← گستردگی نشت (پخش) اطلاعات (diffusion) ← p-box →

جایگزینی کاراکترها باعث ایجاد ابهام و گیج شدن رمزشکن (Confusion) می‌شود. و جابجایی باعث گسترده شدن الگوی آماری موجود در متن اولیه و تمام متن رمز شده می‌شود.

اصطلاحاً پخش شدن بوی عطر از شیشه در باز ↓

یکی روش دیگر برای ایجاد diffusion که باعث می‌شود یک الگوی خاص در plaintext روی چندین کاراکتر از ciphertext توزیع شود، استفاده از روش زیر است:



| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |

$$c_n = \sum_{i=1}^{N} T_{n+i} \ (mod\_M)$$

طول الفبا M

$$c_5 = \sum_{i=1}^{3} T_{5+i} \ (mod\_M)$$ یعنی کاراکتر پنجم

مثال:

plain Text

5	7	10	12	0	9	4	7	2
				↑5	6	7	8	9

$M = 14$

$$c_5 = \sum = T_{5+1} + T_{5+2} + T_{5+3} \ (mod\_14)$$

$$= T_6 + T_7 + T_8 = 9 + 4 + 7 = 20 \ (mod\_14) = 4$$

N ← طول متوسط کلید. دست خوردن است و هرچقدر بزرگتر باشد بهتر است.

استفاده از Diffusion باعث می‌شود که حجم زیادی از Ciphertext برای استخراج کلید لازم باشد که این کار رمزشکن را مشکل می‌کند.

۱۵۲ / ۲۱۳

حداقل مقدار متن رمز که برای عملیات رمزشکنی لازم است، اصطلاحاً unicity distance گفته می‌شود. N_0

جلسه (۸)

انواع رمزها:
- رمزهای بلوکی: block cipher
- رمزهای جریانی: Stream cipher

در رمزهای بلوکی، متن اولیه plain به بخش‌های بلوک‌های با طول ثابت تقسیم می‌شود. و هر بلوک با الگوریتم رمز به صورت ترکیبی از sbox ها و p-box ها اعمال می‌گردد.

۱۵۳ / ۲۱۳

مثل AES و DES

* رمز بلوکی تابعی معلوم و Deterministic از بلوک ورودی است. یعنی چه؟

$B_1 \rightarrow$ [رمز بلوکی] $\rightarrow C_1$

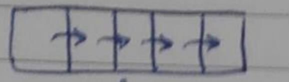
اگر دوباره همین بلوک B_1 ورودی اعمال گردد

$\Rightarrow C_1 = C_2 \rightarrow C = f(B)$

$B_1 \rightarrow$ [رمز بلوکی] $\rightarrow C_2$ خروجی‌اش خواهد داد که با خروجی اولی برابر می‌شود.

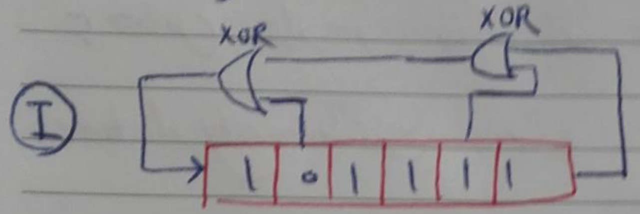
P=30

* در رمزهای جریانی، عملیات رمزنگاری روی بیت به بیت plaintext عمل می‌کند. شمای کلی یک سیستم رمز جریانی به این صورت است:



(در این نوع رمز، از یک شیفت رجیستر استفاده می‌شود.)

(LFSR)



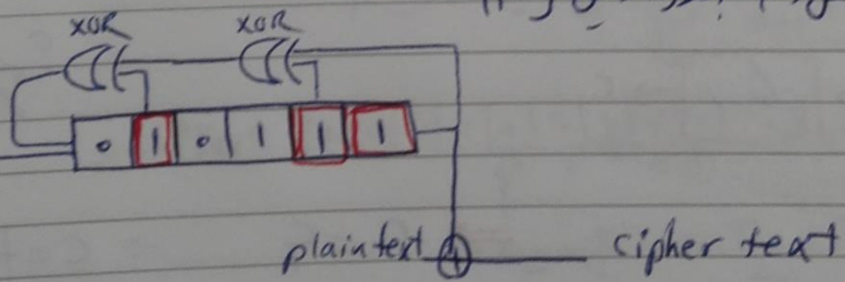
در بار اول انتقال خواهیم

010111

شیفت رجیستری که فیدبک خورده باشد، به آن linear shift register گفته می‌شود.

اعداد شبه تصادفی (Psaudo random)

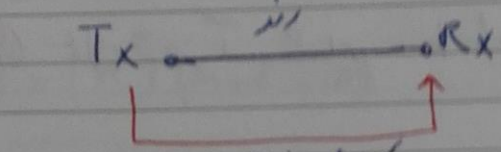
* ویژگی LFSR (Linear Feedback shift register) این است که اگر نحوه‌ی انتخاب بیت‌های SR که در حلقه‌ی فیدبک استفاده می‌شوند، مناسب باشد، اعداد شبه تصادفی تولید می‌کند. مثلاً: در شکل (I)، کلید LFSR، 110010 می‌شود. در شماره بیت‌هایی که Feedback داریم، بیت یک می‌گذاریم. اشکال دیگر به صورت دقیق‌تر !!



* سیستم‌های رمز بلوکی و جریانی هر دو از دسته‌ی رمزهای کلید خصوصی هستند (Private key) یعنی باید هم فرستنده و هم گیرنده هر دو از یک کلید ثابت استفاده کنند. که این کلید باید به صورت

←

محرمانه بین دو طرف باقی بماند.



کانال خوش اندازه است برای تبادل کلید

اینگونه سیستم‌ها در توزیع کلید دارای مشکل و پیچیدگی هستند.

* کانال باید خوش اندازه امن باشد. چون خود کلید را نمی‌توان رمز کرد. [در حضوری]

فلسفه ایجاد سیستم‌های رمزنگاری کلید عمومی این بود که مشکل تبادل کلید [در کانال ناامن] را حل کند.

سیستم‌های رمزنگاری کلید عمومی "public key" تاریخ: 1973

اساس امنیت در سیستم‌های عمومی (RSA) مبتنی بر پیچیدگی تجزیه اعداد بزرگ به عوامل اول آنها می‌باشد.

مثلاً: تجزیه یک عدد ۳۰۰ رقمی به عوامل اول که چندین سال طول می‌کشد. پس این الگوریتم دارای امنیت خوبی است.

① دو عدد اول P و Q توسط گیرنده انتخاب می‌شود به نحوی که نسبت به $P-1$ و $Q-1$ نیز اول هستند. ($P-1 \nmid Q$, $Q-1 \nmid P$)

② $N = P \cdot Q$ کلید عمومی گیرنده است و عدد N اعلام می‌شود نه P و Q.

③ برای رمزنگاری و ارسال پیام رمزشده $C = T^N \pmod N$ متن رمزشده. (متن اولیه)

④ گیرنده مقدار "C" را دریافت می‌کند.

⑤ گیرنده یکی از این معادلات را حل می‌کند. $\begin{cases} PP' \equiv 1 \pmod{Q-1} \\ QQ' \equiv 1 \pmod{P-1} \end{cases}$ از حل این، P' و Q' را بدست می‌آورد.

P=52

۱۵۶ / ۲۰۹

۶) عمل رمزگشایی با استفاده از

$$\begin{cases} T = C^{P'} \ (mod\text{-}Q) \\ T = C^{Q'} \ (mod\text{-}P) \end{cases}$$

انجام می‌شود.

مثال: می‌خواهیم الگوریتم RSA که در آن $P=3$ و $Q=5$ است، برای رمزنگاری و رمزگشایی کاراکتر $T=2$ اجرا کنیم؟

۱) P و Q نسبت به هم اول هستند. P هم نسبت به Q-1 اول است (۳ نسبت به ۴) و Q هم نسبت به P-1 (۵ نسبت به ۲) اول است.

۲) $N = P.Q = 15$

۳) $C = T^{N} \ (mod\text{-}N)$ یعنی ← $C = 2^{15} \ (mod\text{-}15)$

$$= (16)^{3} \times 2^{3} \overset{15}{\equiv} (1)^{3} \times 8 = 8$$

۴) گیرنده مقدار $C=8$ را دریافت می‌کند.

۵) محاسبه معادله:

$$\begin{cases} PP' = 1 \ (mod\text{-}Q\text{-}1) \\ PP' = 1 \ (mod\text{-}4) \end{cases} \rightarrow 3P' = 1 \ (mod\text{-}4) \rightarrow \begin{cases} P'=1 & 3 \overset{4}{\not\equiv} 1 \ ✗ \\ P'=2 & 6 \overset{4}{\not\equiv} 1 \ ✗ \\ \boxed{P'=3} & 9 \overset{4}{\equiv} 1 \ ✓ \end{cases}$$

$$\begin{cases} QQ' = 1 \ (mod\ P\text{-}1) \\ 5Q' = 1 \ (mod\text{-}2) \end{cases} \rightarrow 5Q' = 1 \ (mod\text{-}2) \rightarrow \begin{cases} \boxed{Q'=1} & 5 \overset{2}{\equiv} 1 \ ✓ \end{cases}$$

پس $Q'=1$ و $P'=3$ بدست آمد

ادامه ←

۶) حل $T = C^{P'} \pmod{Q}$

$T = (۸)^{۳} \pmod{۵}$

$T = ۸^{۳} \overset{۵}{\equiv} ۳^{۳} \overset{۵}{\equiv} ۲۷ \overset{۵}{\equiv} ۲$ → $T = ۲$ متن اولیه بدست آمد ✓

۱۵۷ / ۲۰۸

(اثبات ریاضی مربوط T) : $C = (T^{PQ})^{Q'} = T^{PQQ'} = T^{P}$ (برعکس یک)

رمزنگاری دارای سه جنبه:
- محرمانگی
- احراز هویت
- صحت

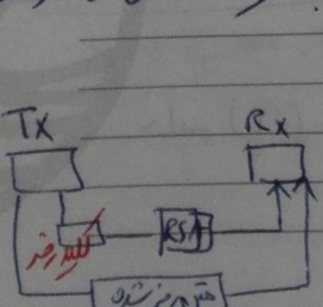
سیستم‌های رمزنگاری ترکیبی (هیبریدی):

کلید عمومی:
+ عدم نیاز به توزیع کلید
- پیچیدگی محاسباتی و سرعت کم

کلید خصوصی:
+ پیچیدگی کم‌تر و سرعت بالاتر
- مشکل توزیع و تبادل کلید.

⊕ ترکیب

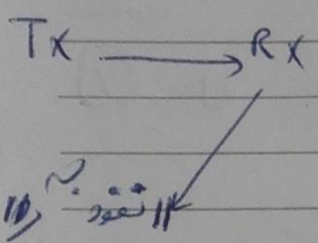
با ترکیب این دو سیستم رمزنگاری از مزایای هر دو می‌توان استفاده کرد. به این شکل عمل می‌شود که ابتدا کلید خصوصی با یک الگوریتم کلید عمومی رمز شده و به گیرنده ارسال می‌شود. و سپس متن رمز شده با کلید خصوصی توسط کانال ارسال می‌شود.



نیازی به کانال مجزا ندارد

رمز PGP یک سیستم رمز هیبریدی است که از روش فوق استفاده می‌کند.
(Prety Good Parity)

احراز هویت : (Authentication)



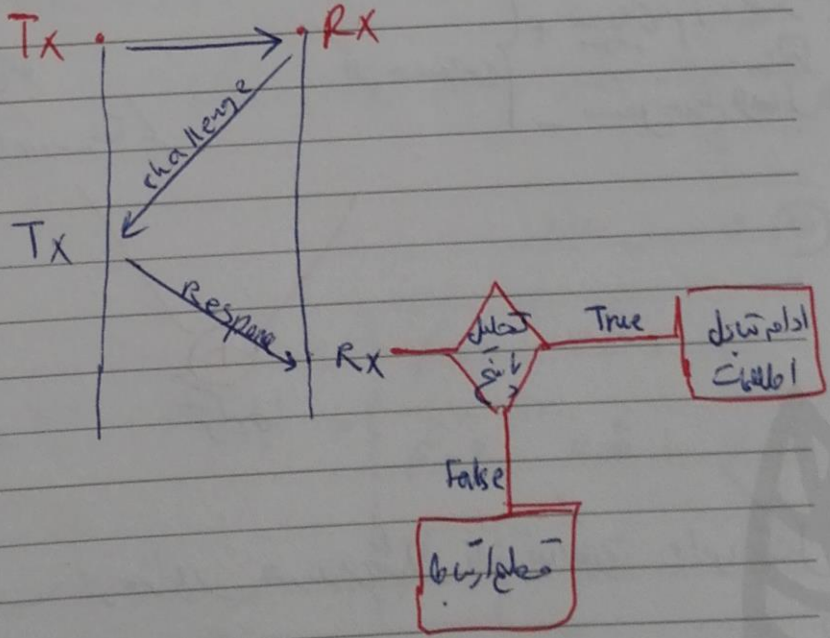
فرآیندی است که در طی آن منبع صدور پیام تعیین گردد.

شاید اینگونه برداشت شود که داشتن کلید رمز که فقط گیرنده و فرستنده آن را دارند ، برای احراز هویت کافی است ولی اینگونه نیست . زیرا پیام رمز می‌تواند ذخیره شده و در زمانی دیگر تکرار (repeat) شود.

(همیشه احراز هویت برعهده‌ی گیرنده است.)

سیستم‌های احراز هویت اغلب براساس سوال / پاسخ (Challenge / response) که در آن یک سوال یا چالش برای فرستنده مطرح شده و گیرنده از روی صحت پاسخ فرستنده هویت او را تشخیص یا احراز می‌کند.

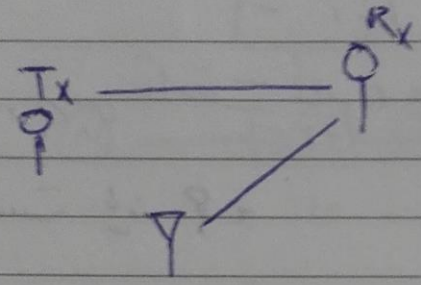
مثال : Captcha



جلسه (۹) پنجشنبه

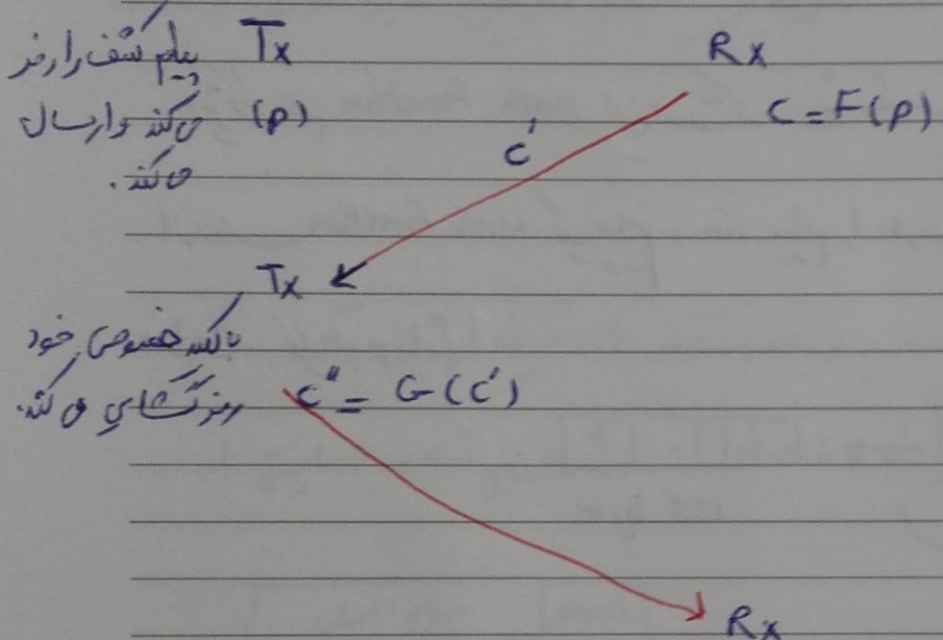
احراز هویت (Authetication)

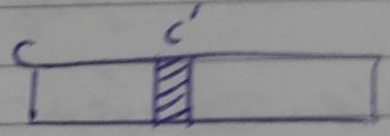
(منبع صدور پیام)



۱۵۹ / ۲۰۶

احراز هویت به کمک رمز کلید عمومی :





کلید عمومی $k_1 = p, q$ | کلید خصوصی p / q

$C''' = F'(C'')$ با کلید عمومی فرستنده رمزگذاری می کند.

در این صورت عمل احراز هویت درست انجام شده است.

۱۶۰ / ۲۰۵

توضیح :

گیرنده متن رمز شده ی C را دریافت می کند، بخشی از آن را برای فرستنده ارسال می کند. فرستنده با کلید خصوصی خود، متن را رمزگشایی می کند و "C را برای گیرنده مجدداً ارسال می کند. گیرنده پیام دریافتی را با کلید عمومی فرستنده رمزگذاری می کند ("C) بدست می آید. اگر C=C" باشد، فرستنده احراز هویت می شود و در غیر این صورت نه.

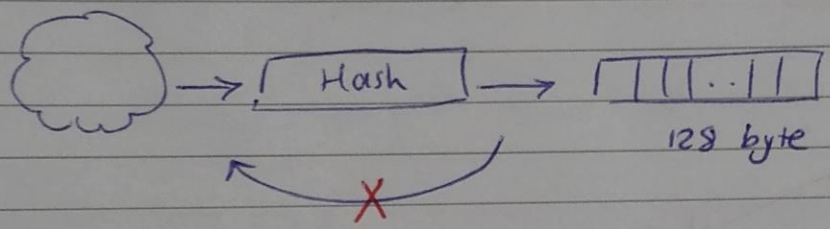
صحت و عاملیت پیام (Integrity):

در بررسی عاملیت پیام، هدف پاسخ به این سؤال است که «آیا متن دستکاری شده است یا نه؟»

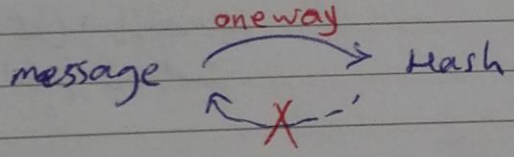
برای پاسخ دادن به این سؤال، از مفهومی به نام Hash Function استفاده می‌شود.

Hash Function اصطلاحاً به آن digest (خلاصه کردن) هم می‌گویند.

ویژگی مهم Hash Function این است که عملاً معکوس کردن آن غیرممکن است. یعنی نمی‌توان از روی Hash Function که پیام، خود پیام را بازسازی کرد. همچنین Hash Function باید خواص زیر هم را داشته باشد.



(۱) محاسبه Hash Function از روی پیام داده شده، ساده است. و عکس آن بسیار مشکل (غیرممکن) است.



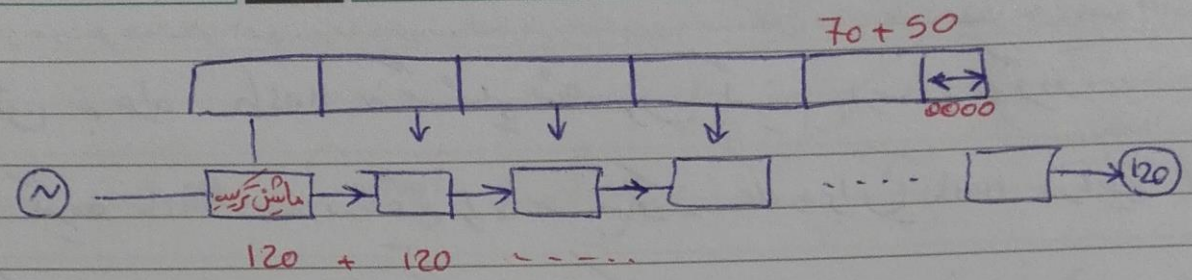
به این دلیل است که به این نوع توابع، توابع oneway «یک‌طرفه» گفته می‌شود.

(۲) تغییرات جزئی در پیام باعث تغییرات شدید در Hash Function می‌شود.

نمی‌توان که پیام را تغییر داد بدون اینکه Hash آن تغییر کند.

(۳) نمی‌توان دو پیام مختلف یافت که Hash یکسان داشته باشند. (از نظر آماری)

یکی از الگوریتم‌های Hash است: Merkle/Damgard



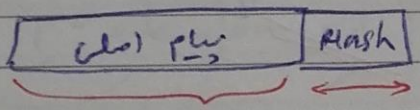
۱۶۲ / ۲۰۳

شبیه S-box , p-box

با خود الگوریتم مرحل کاری نداریم فقط از کاربردی Hash در تعیین احراز هویت و صحت استفاده می‌خواهیم کرد.

(چگونه؟)

پیام اولیه توسط فرستنده محاسبه و به صورت فشرده برای گیرنده ارسال می‌گردد. گیرنده پیام دریافتی را محاسبه و با Hash «رمزگشایی شده» دریافتی از فرستنده مقایسه می‌کند.



امضای دیجیتال (Digital Signature):

می‌خواهیم که سند یا محتویات را به یک فرد خاص نسبت بدهیم.

- احراز هویت
- تایید صحت و عاصیت

روشی که متعلق برای امضای دیجیتال است که کپی‌برداری می‌کنیم.

عمدهٔ سیستم‌های امضای دیجیتال، مبتنی بر سیستم‌های Public key است. (مثل گیرنده جدید)

① صباء «فرستنده» امضا شده Hash سند را محاسبه می‌کند و آن را با کلید خصوصی خود «که عملاً همان امضای دیجیتال است» آن را رمزگشایی می‌کند.

«P و Q از کجا به دست آمده است؟»

P=58

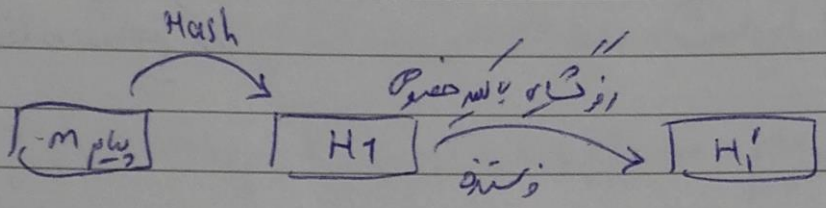
۱۶۲ / ۲۵۲

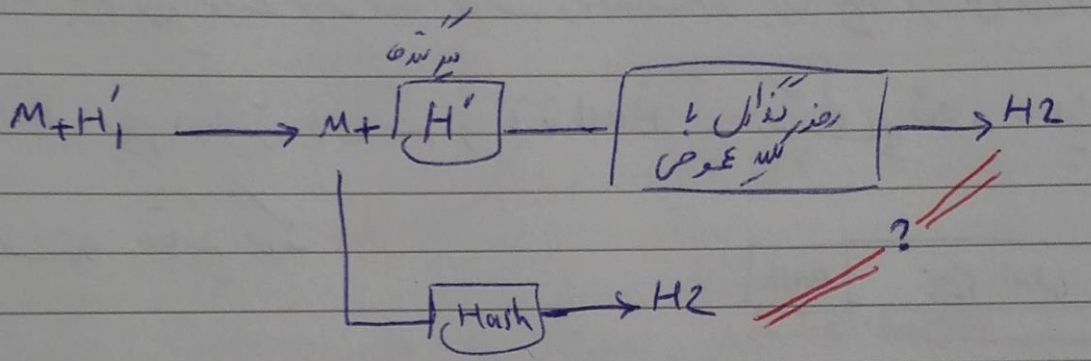
(۳) متن به همراه Hash رمزنگاری شده برای گیرنده ارسال می‌گردد.

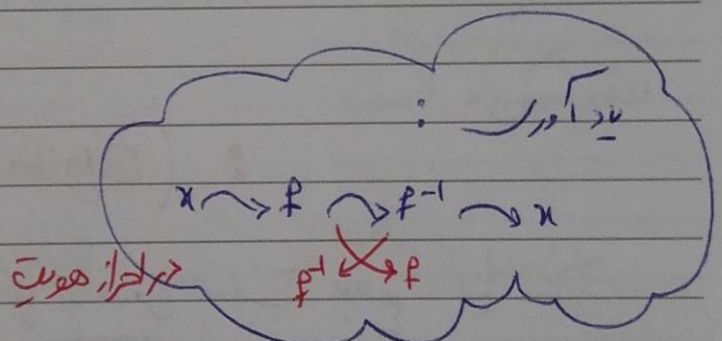
(۴) گیرنده Hash سند دریافتی را می‌سنجد علاوه بر این Hash دریافتی از فرستنده را با کلید عمومی فرستنده رمزگشایی می‌کند.

(۵) با تطبیق این دو Hash، اصالت فرستنده تایید می‌شود.

Schema







تمریناتی که باید حل گردد: ۹-۶، ۹-۷، ۹-۸، ۹-۹، ۹-۱۰، ۹-۱۱، ۹-۱۲، ۹-۱۳، ۹-۱۴، ۹-۱۵.

تا اینجا تئوری مطالب بود. از اینجا به بعد وارد بحث جدید تئوری و کدها می‌شویم

P=59

۱۶۴ / ۲۰۱

بخش جدید :

کدینگ برای کنترل خطا

"Error Control Coding"

Coding →
- wave Form coding — بحثها مربوط به کنترل و شکل دهی طیف که در مخابرات ۲ بحث می شود.
- source Coding — بحثهای مربوط به فشرده سازی و رمزنگاری
- Channel Coding / Error Control coding → ادامه بحث ما می باشد.

منابع یا Resource های در دسترس برای یک سیستم ارتباطی :

- توان (ERP) Effective Radiation Power
- زمان
- پهنای باند

از این منابع استفاده می کنیم تا یک ارتباط با کیفیت را برقرار کنیم.

"Trade off" بده بستان

↓

یعنی تصمیم می گیریم که زمان برای ما مهم است، پهنای باند را نسبت به آن متغیر و متفاوت در نظر می گیریم و یا ...

هدف کلی طراحی سیستم مخابراتی : رسیدن به حداکثر نرخ تبادل داده با حداقل پهنای باند در حالی که یک حد قابل قبول برای کیفیت تبادل رعایت گردد.

① احتمال خطای بیت P_b در گیرنده پایین‌تر گردد.

مثلاً $P_b=10^{-4}$: از هر ۱۰,۰۰۰ بیت، یک بیت خطا داریم.

پس باید کمتر از این باشد چون در کاربردی مثل صوت مشخص می شود. ←

مثلاً $P_b=10^{-6}$ یعنی یک خطا در یک میلیون بیت.

P=60

۱۶۵ / ۲۰۰

② نرخ تبدیل خطای بیت:

$$BER = P_b \times R_i$$

(R_i: نرخ تبدیل بیت (bit rate))

قضیه شانون:

برای هر کانال مخابراتی یک ظرفیت Capacity تعریف می‌شود که:

$$C = B \log_2 (1 + S/N)$$

که بیانگر حداکثر نرخ تبدیل داده‌ی بدون خطا برای یک کانال با پهنای باند B و نسبت سیگنال به نویز S/N است.

مثال: کانالی طراحی کردیم که پهنای باند آن $B = 1MHz$ و $S/N = 15$، حال ظرفیت این کانال را محاسبه کنید؟

$$C = B \log(1 + S/N) = 10^6 \times \log(16) = 4 \times 10^6 \frac{bit}{sec}$$

↓

بیشتر از این مقدار امکان ندارد حتی اگر هوش را بکشیم!!

* رابطه‌ی شانون یک حد (limit) برای ارتباطات بدون خطا ($P_b = 0$) را عرضه می‌کند. لذا اگر یک سیستم مخابراتی با نرخی کوچکتر از ظرفیت کانال، اقدام به تبادل داده کند، امکان رسیدن به تبادل بدون خطا وجود دارد و کدینگ کانال، ما را به این هدف نزدیک می‌کند.

P=61

* روش‌های کنترل برای آشکارسازی Detection و تصحیح خطا Correction استفاده می‌شود که اینکار را با اضافه نمودن بیت‌های systematic به هر پیام انجام می‌دهد.

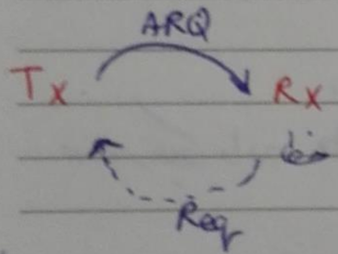
«آشکارسازی مقدم بر تصحیح است.»

دسته‌بندی روش‌های کنترل خطا «Channel Coding»

ARQ Automatic Repeat Request		FECC Forward Error Control Coding	
stop & wait	coutinious ARQ	Block code	convolution

Block code ← طول بسته ورودی و خروجی برابر

convolution ← طول بسته ورودی و خروجی برابر نیست. یک دنباله از داده‌ها وارد و دنباله‌ای از داده‌ها خارج می‌شوند.



باید مسیری دوطرفه بین فرستنده و گیرنده وجود داشته باشد.

اطلاعات اضافی «Redundancy»

شکل سیستماتیک داده‌ها اضافی می‌گردد تا هم خطاها آشکارسازی و هم بدون نیاز به ارسال مجدد، تصحیح شوند.

در ARQ، Tx هم فرستنده محسوب می‌شود هم گیرنده و Rx هم همینطور ولی در FECC، چنین نیست، Tx فقط و فقط فرستنده است و Rx هم فقط گیرنده.

«ادامه مطلب»

جلسه (۱۰) روز جمعه

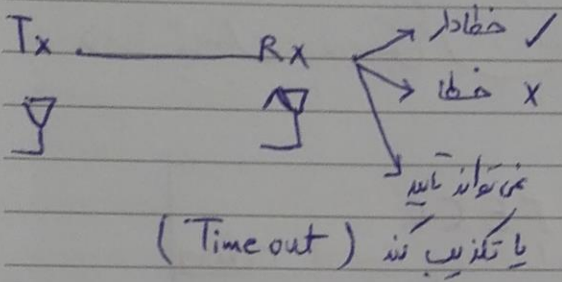
p-62

۱۶۸ / ۱۹۷

ARQ ساده‌تر از FECC است.

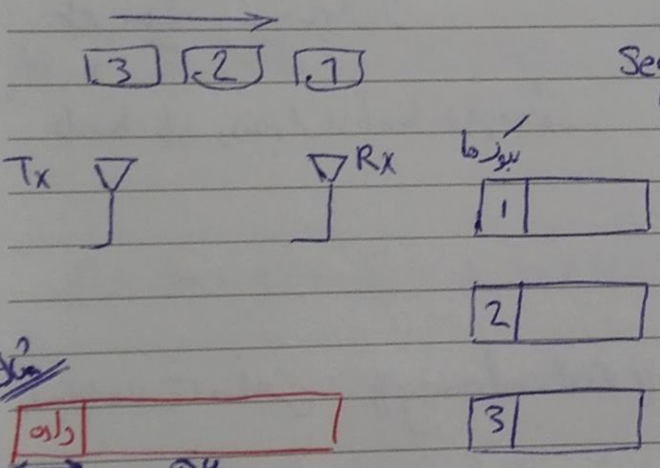
① S&W : هر بلوک ارسال شده، فرستنده توسط گیرنده تحلیل شده و صحت آن تایید یا تکذیب (عدم پاسخ) می‌شود.

ARQ → ① ، ②

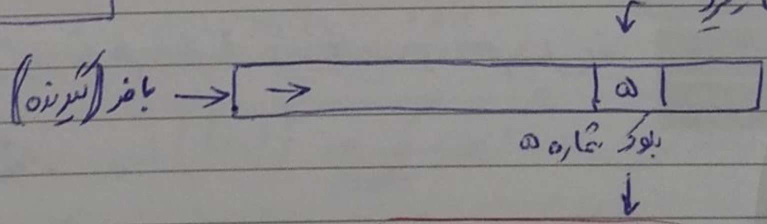


② Continuous → go-back-N ، selective

* هر بلوک داده دارای یک Sequence Number



گیرنده: داده‌ها را در یک بافر (FIFO) می‌ریزد



گیرنده متوجه می‌شود که در بلوک شماره ۵ خطا رخ داده است. حال، دستور go-back-N می‌دهد یعنی به فرستنده می‌گوید بلوک‌ها از ۵ به بعد را دوباره ارسال کند. که باعث کندی می‌شود. عیب این روش، باعث شده است که دسته‌ی دیگری به نام ←

selective موضع شود :

درروش selective فقط بلوکی که در آن خطا رخ داده است ، ارسال مجدد شود که مشکل کندی سرعت حل می‌شود.

برای آشکارسازی خطا بیت‌های اضافی در حالت parity به داده اضافه می‌شود و از طریق آن امکان آشکارسازی خطا ممکن می‌شود.

کدینگ ← کیفیت ارتباط را بهبود می‌دهیم.

① P_b : احتمال دریافت کوبیت به اشتباه

② BER : (Bit Error Rate)

$BER = P_b \times R_i$

R_i ← bit rate

مثال

RS232

پورت سریال کامپیوتر — 9600 bit/s

مثلاً $P_b = 10^{-4}$

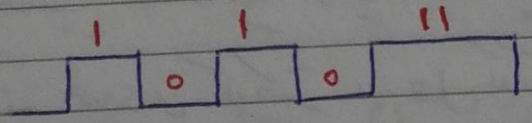
$BER = 10^{-4} \times 9600$

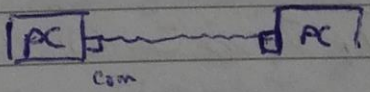
RS232

کیفیت ارتباط (P_b) به چه عواملی بستگی دارد ؟

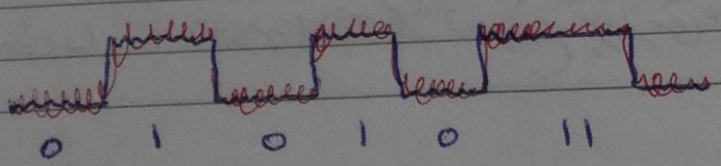
مثالی برای درک مثل قبل

1 0 1 1 0 1 1





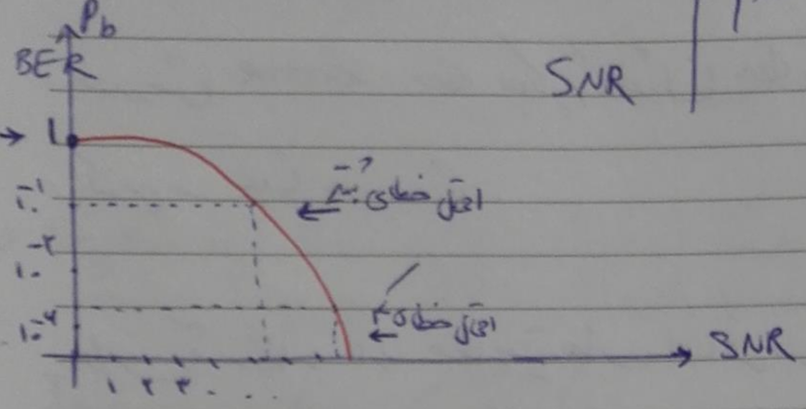
دامنه سیگنال رفته رفته کم می‌شود. مثل صوت و علاوه بر آن ممکن است آلوده به نویز هم شود.



(S/N)

رسم کیفیت: که بر حسب دو مفهوم { BER , P_b / SNR

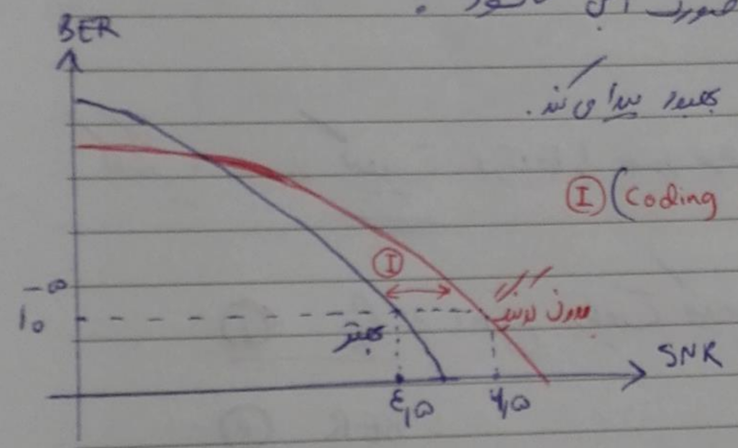
P=64



برای اینکه کیفیت بهتر شود، نمودار به صورت آبی می‌شود:

کیفیت اینجا با کدینگ (در اغلب موارد) بهبود پیدا می‌کند.

بهره (بهبود) کدینگ (coding gain) (I)



$P_b = 10^{-5}$ ← بدون کدینگ → SNR = ۶٫۵

بهبود یافته است ↓ ← با کدینگ → SNR = ۴٫۵

۶٫۵ − ۴٫۵ = ۲ (coding gain)

منحنی آبی (بهبود یافته) باید بالای منحنی قرمز بیفتد. به این پدیده آستانه گفته می‌شود.

پدیده‌ی آستانه: (Threshold phenomena)

از یک SNR ی کوچکتر، منحنی $\frac{P_b}{SNR}$ سیستم دارای کدینگ بالاتر از منحنی $\frac{P_b}{SNR}$ سیستم بدون کدینگ قرار می‌گیرد و این بدان معنی است که از یک SNR ی کوچکتر، عمل کدینگ، باعث تخریب کیفیت اینجا می‌شود و دلیل آن این است که در این محدوده تعداد خطاها به حدی است که هر نوع پردازشی باعث ایجاد خطاهای بیشتر می‌شود.

مثل ساختمانی که می‌خریم و از نقطه خراب است که اصلاح آن درست نمی‌ارزیم تا بهتر نشود.

P=65

۱۷۱ / ۱۹۴

کدهای بلوکی (block code) :

(n,k)

تعریف فاصله همینگ (Hamming distance)

* تعداد بیت‌ها یا digit هایی که بین دو کد واژه codeword متفاوت اند.

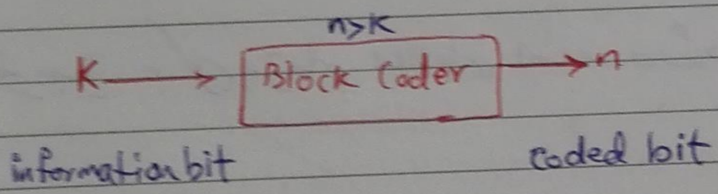
c_1 1 1 1 0 0

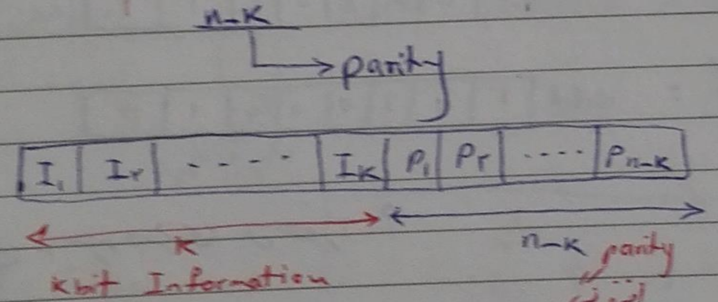
c_2 1 1 0 1 1

$d_H(c_1, c_2) = 3$

* وزن کد (wieght)

یعنی تعداد یک های کد واژه : $w(c_1) = 3$, $w(c_2) = 4$





$K = \frac{k}{n}$

$0 < \frac{k}{n} < 1$

به صورت همزمان تعداد بیت (n-k) تا به منظور کمک در آشکار سازی و تصحیح خطا به بیت های اطلاعاتی اضافه می گردد.

Block
- Systematic : اگر بیت های اطلاعات به صورت صریح در کد واژه موجود باشد
- Non systematic : " " " " موجود نباشد.

I_1	P_1	P_2	I_2	I_3	I_4	P_3	I_5

x_1	x_2	. . . . .

x_i ها Function های هستند از اطلاعات

به صورت صریح بدست نمی آید.

8 Single Parity check Code

K-bit information — یک بیت → even / odd

مثلاً: اگر $K=7$ و $P=1$ → $n=8$

نرخ $R = \frac{K}{n} = \frac{7}{8}$

یک بیت به کل بیت های اطلاعاتی اضافه می شود که معمولاً به صورت { even parity / odd parity } است.

مثال: even | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | : کل تعداد یک ها باید زوج باشد.

odd | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | : کل تعداد یک ها باید فرد باشد.

این روش به چه دردی می خورد؟

مثلاً: در RS232 میتوان even یا odd ست (set) کنیم.

parity: ☐ even ☐ odd

هرچه نرخ کدینگ (K) به یک نزدیک تر باشد، توانایی کدینگ در آشکارسازی و تصحیح خطا کمتر است.

مثال: | 1 | 0 | 1 | 0 | 0 | 1 | 0 | گیرنده اینطور دریافت کرده است. parity=3

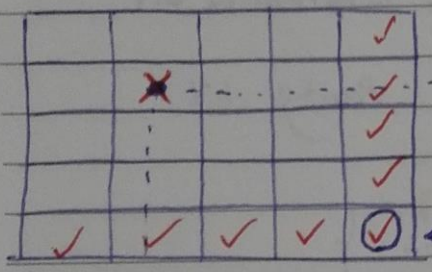
ولی فرستنده به این صورت ارسال کرده بود: | 1 | 0 | 1 | 0 | 0 | 1 | 1 |

$P=4$ می فهمیم که خطا رخ داده است.

این سیستم امکان آشکارسازی خطاهای مرتبه دو را فراهم می‌کند.

P=67

اگر Single Parity را به صورت دوبعدی ببینیم و فرض کنیم تعداد خطا دو تا باشد، در این صورت هم می‌توانیم آشکارسازی کنیم و هم تصحیح.



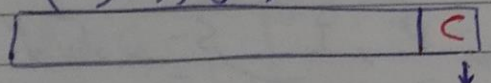
اگر صفر باشد، یک می‌کنیم و اگر یک باشد، صفر می‌کنیم.

این خانه چون باقیمانده هم هست، ←

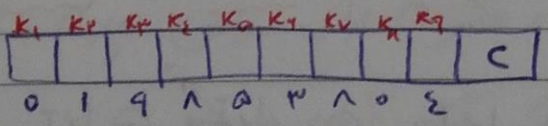
parity ستون را می‌گیرد ولی از نظر تئوری ثابت شده است که یکی درست می‌کند.

شماره شابک کتاب (ISBN)

Inter Standard book Number



فرمول ISBN : $C = 11 - \sum_{i=1}^{9} (11-i)\,K_i \pmod{11}$



مثال: جواب

$$C = 11 - \sum_{i=1}^{9} (11-i)\,K_i \pmod{11} = 0\times10 + 1\times9 + 9\times8 + 8\times7 + 5\times6 + 3\times5 +$$

$$8\times4 + 0\times3 + 4\times2 = 35$$

$$\begin{array}{r|l} 35 & 11 \\ 33 & \\ \hline 2 & \end{array}$$

(mod) باقی مانده ۲

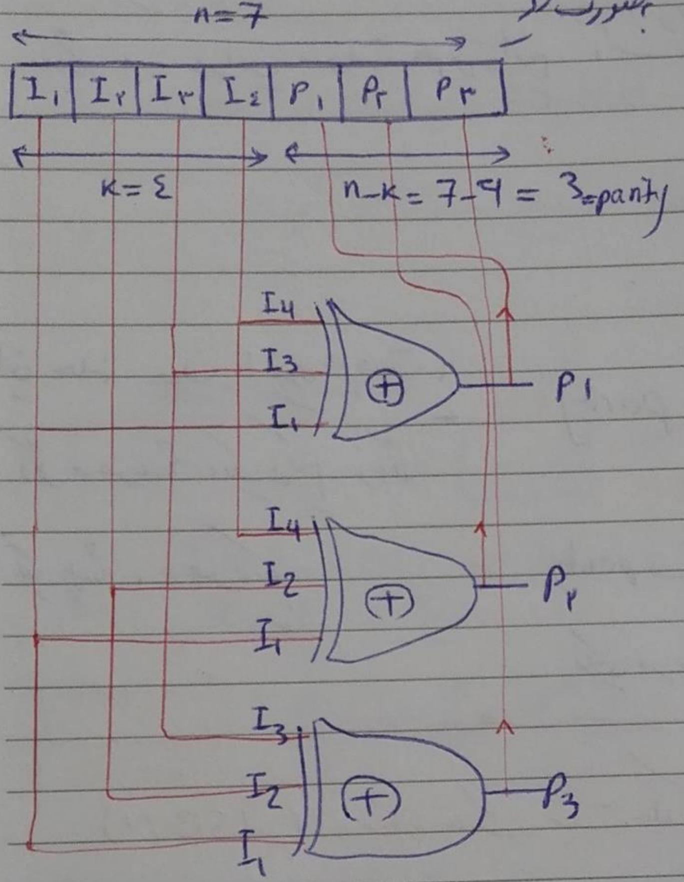
$$\Longrightarrow 11 - 2 = 9$$ ← رقم یازدهم

عید سعید فطر (تعطیل) ـ وفات آیت‌الله سید محمود طالقانی اولین امام جمعه تهران (۱۳۵۸ هـ ش)

جلسه (۱۱) «روز پنج شنبه» «۹۳/۹/۲۷» «این جلسه غایب بودم» p=68

۱۷۵ / ۱۹۰

مثال ۵ فرض کنید کد بلوکی (7,4) بصورت زیر



I : Information bit

P : Parity bit

if $I_1 = 0$, $I_2 = 0$, $I_4 = 1$ $\rightarrow P_1 = 1$

if $I_1 = I_2 = I_3 = 1 \rightarrow P_1 = 1$

if $I_1 = I_2 = I_3 = 0 \rightarrow P_1 = 0$

اگر تعداد بیت‌های I و P زوج باشد، یا تعداد بیت‌های I و P زوج باشد، آن را even Parity گوییم. همچنین کد سیستماتیک است. (محل بیت‌ها مشخص)

«مورد سؤال لغت»

$R = \frac{k}{n} = \frac{4}{7}$

خودمان بنویسیم →

$P_1 = 1 \times I_1 \oplus 0 \times I_2 \oplus 1 \times I_3 \oplus 1 \times I_4$

$P_2 = 1 \times I_1 \oplus 1 \times I_2 \oplus 0 \times I_3 \oplus 1 \times I_4$

$P_3 = 1 \times I_1 \oplus 1 \times I_2 \oplus 1 \times I_3 \oplus 0 \times I_4$

بصورت ماتریسی هم قابل نمایش دادن است.

ادامه ←

p=69

$$H = \left[\begin{array}{cccc:ccc} I_1 & I_2 & I_3 & I_4 & P_1 & P_2 & P_3 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

H ↓ parity matrix

Parity بخش (ستون‌های I) ، $I_{3\times3}$ (ستون‌های P)

۱. قطر اول چون P_1

" دوم چون P_2

" سوم چون P_3

* ماتریس I در واقع Parity را مشخص می‌کند.

کدهای خطی مبتنی بر گروه «Linear group code»

نظریه‌ی گروه «Group Theary»

اگر (G,o) یه عملگر (o) و یه گروه یا مجموعه داشته باشیم:
(o ↓ Operator)

$g_1 \in G$ ، $g_2 \in G$ ⟹

1. $g_1 o g_2 = g_3 \in G$ ‏G نسبت به عمل o بسته است.
2. $g_1 o g_2 = g_2 o g_1$ ‏G خاصیت جابجایی (آبلی) دارد.
3. $\forall g_1 \in G \,,\; \exists e \in G \quad g_1 o e = g_1$ ‏e: عضو خنثی
4. $\forall g_1 \in G \,,\; \bar{g} \in G \;..\; g_1 o \bar{g} = e$ ‏G دارای عضو قرینه (وارون) است.

* اگر یک مجموعه G و عملگر o دارای خواص چهارگانه ذکر شده باشند، آنگاه گفته می‌شود که (G,o) تشکیل یک گروه آبلی داده‌اند.

تعریف کدهای گروهی: کدهای هستند که شامل کدواژه‌ی تمام صفر (عضو خنثی یا کدی که تمام بیت‌هاش 0 باشد) بوده و بسته باشه به عبارت دیگر اگر c_i و c_j جزو کدواژه معتبر باشند، آنگاه $c_k = c_i \oplus c_j$ نیز یک کد واژه معتبر است.

مثال : یک منبع دارای الفبای ۴ تایی است a,b,c,d .

یک کد ۵ بیتی تعریف می‌کنیم (برای a)

p:70

	k
a	00
b	01
c	10
d	11

m = 5

00000

b را با کد 00111 ، c را با کد 11100 و d را با 11011 ، کد می‌کنیم می‌خواهیم بررسی کنیم آیا این کد یک کد گروهی است یا نه؟

$a \oplus b = b$ 00000 ⊕ 00111 = 00111 = b که ✓

$c \oplus b =$ 11100 ⊕ 00111 = 11011 , d

✓ هر کد دلخواهی را که ترکیب می‌کنیم، جواب، کدی می‌شود که در داخل فضای کدها قرار دارد.

پس این یک کد linear group است.

این کد سیستماتیک است زیرا :

بیت‌های قبل از کد شدن را در بیت‌های بعد از کد شدن می‌بینیم (تکرار شده) یعنی این سیستماتیک است.

	قبل از کد شدن		بعد از کد شدن (دو بار تکرار شده)				
a	0	0	0	0	0	0	0
b	0	1	0	0	1	1	1
c	1	0	1	1	1	0	0
d	1	1	1	1	0	1	1

تذکر : BCH و R.S جزء کدهای linear group است.

تخمین کارایی (Performance Prediction) :

روش تخمین کارایی یک کد، استفاده از فاصله همینگ است « منظور از کارائی : ایجاد شرایط بهتر برای تصحیح خطا »

می‌خواهیم بررسی کنیم کدام سیستم کدینگ کارایی بهتری دارد ؟

I		II	
0 →	1001	0 →	1001
1 →	0110	1 →	1000

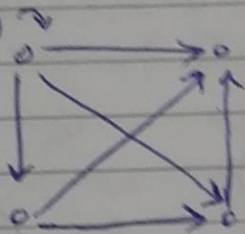
10^{-2}

احتمال خطا در سیستم II، 10^{-2} و در سیستم I، 10^{-8} است. (I کارایی بیشتری دارد)

فاصله همینگ در سیستم II، ۱ و در سیستم I، ۴ است.

اگر سیستم ۴ کد واژه داشته باشد:

(فاصله همینگ ۱)



در این فاصله احتمال تبدیل کد (بیت‌ها) بیشتر است؟ می‌گوییم زیاده احتمال خطا هم بالاست.

* معیار ارزیابی کارایی کد یک سیستم کدینگ حداقل فاصله همینگ (D_{min}) بین کد واژه‌ها است.

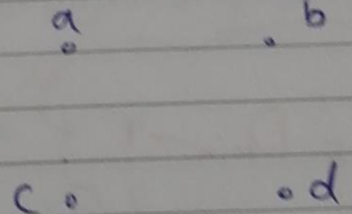
مثال: فاصله همینگ را بدست آورید؟

C_1 00000

C_2 00111

C_3 11100

C_4 11011



$\frac{4\times 3}{2}=6$

$\frac{n(n-1)}{2}$ = تعداد فاصله‌های همینگ

فاصله همینگ: C_1 با C_2، ۳ ، C_3 با C_4، ۳ ، C_2 با C_4، ۳ ، C_2 با C_3، ۴

C_1 با C_4، ۴ ← $D_{min}=3$

* بهترین حالت وقتی است که D_{min} (بیشتر) باشه.

(D_{min} وقتی کوچک باشد، قدرت تشخیص خطا کمتر میشه) زیرا کدهای کمتر داشتیم.

p=72

وزن کد c_1 برابر ۵، کد c_2=۳، کد c_3 برابر ۳، کد c_4 برابر ۴ است.

* برای یک گروه این فرمول صادق است:

$$D_{min} = wieght_{min} \text{ (کد غیر صفر)}$$

* اگر حداکثر قدرت و توانایی کد برای تصحیح خطا را t فرض کرده ثابت می‌شود:

$$t = int\left(\frac{D_{min}-1}{2}\right)$$

جزء صحیح

حوزه تصحیح کد: کد ← ۵، صحیح کد، $\frac{D_{min}-1}{2}$

مثال

۲ ۳ ۴ ۵

فاصله کد ۲ و ۵ = ۳

اعداد بین ۲ و ۵ = ۲ ← از این آمده $D_{min}-1$ 2 = 3 − 1

$$e + t = D_{min} - 1 \quad , \quad t \leq e$$

e: قدرت آشکارسازی خطا

t: قدرت تصحیح خطا

تذکر: هر کد نمی‌تواند به طور همزمان در هر دو حوزه آشکارسازی و تصحیح خطا کار کند.

در مورد مثال قبل:

$D_{min} = 3$ $e+t = 2$ ⟹ $e+1 = 2$

$t = int\left(\frac{3-1}{2}\right) = int(1) = 1$ $e=1$, $t=1$

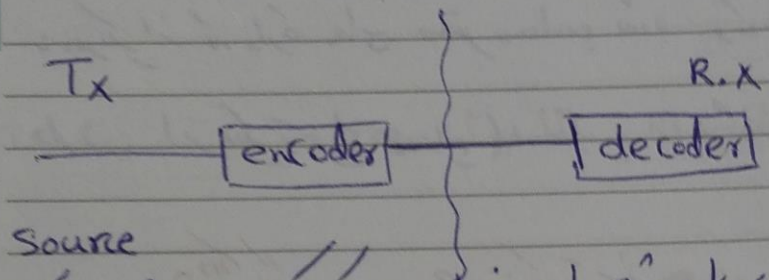
نمی‌توانیم بگوییم ۲ بیت خطا را آشکارسازی کرده و ۱ بیت تصحیح می‌کند!

←

می توانیم بگوییم یک خطا را آشکار کرده و همان را تصحیح کرده است.

P=73

روش دی کدینگ:

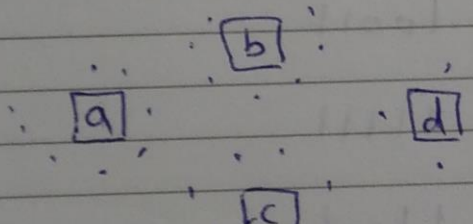


۱۸۵ / ۱۸۵

یا می رانکه به صورت کد شده به گیرنده ارسال شده را می خواهیم دی کد کنیم با روش حداکثر شباهت (maximum likehood)

* روش دی کدینگ به روش نزدیکترین همسایه N.N: Nearest Neighbour

در مثال قبل هرکه را با ۵ بیت کد، کدینگ کردیم، پس در گیرنده $2^5 = 32$ حالت از کدها وجود خواهد داشت: اطراف هر کد a, b, c, d، ۲۸ حالت از کدها وجود دارد:



۱۸۴ / ۱۸۱

با روش N.N می خواهیم که اصلی را تشخیص دهیم.

	کدکر اصلی	
a	00	00000
b	01	00111
c	10	11100
d	11	11011

فرض کنیم در گیرنده کدواژه ی 00010 را دریافت کردیم، ابتدا فاصله این کدواژه را با هرکدام از کدواژه های اصلی محاسبه می کنیم. فاصله با a برابر ۱، با b: ۲، با c: ۴ و با d: ۳ است. نزدیکترین فاصله کدواژه دریافتی با 00000 است پس گیرنده 00000

را انتخاب کرده و می‌گوید که فرستنده ۰۰ را فرستاده است (فاصله دنباله دریافتی با هریک از کدواژه های معتبر محاسبه شده و کدواژه‌های که حداقل فاصله را با دنباله دریافتی دارد (نزدیک‌ترین همسایه) انتخاب می‌شود.

روش لایه‌ای = روش نزدیک‌ترین همسایه :

در این روش یک جدول که در بالای آن کدواژه های معتبر وجود دارد. در زیر آن کدواژه های که دارای فواصل ۱و۲و... هستند، تشکیل می‌شود.

00000	11100	00111	11011
00001	01100	10111	11010
00010	10100	01111	10011
00100	11000	00011	11111
01000	11110	00101	11001
10000	11101	00110	11010
10001	01101	10110	01010
10010	01110	10101	01001

$D_{min} = 3 \quad t = 1$

حالت های را می‌نویسیم که در جدول وجود ندارند.

در کل ۳۲ حالت داریم.

(۲۴) تمام کد واژه های که فاصله را دارند نوشتیم، ۸ حالت باقی مانده مربوط به فاصله ۲ خواهد بود.

جلسه ۱۲، روز علم

حل تمرینات صفحه ۹۴، ۷، ۸، ۹، ۱۰، ۱۱، ۱۲، ۱۳، ۱۴، ۱۵

اتمام درس کدینگ