By: Prof. Pandu Rangan Chandrasekaran

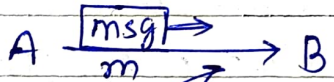
DATE: 16/8/22
PAGE NO.: 1

TUE

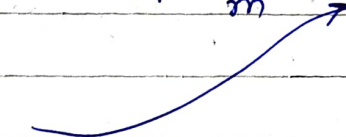
Introduction

1. Network security:
2. System security:
3. OS security:
4. Database security:

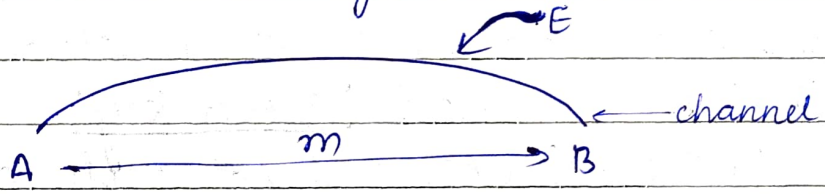
1. Confidentiality



(Msg should be accessible only to B)

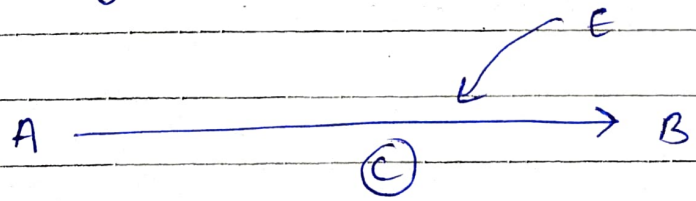


(adversary (or) attacker)
(who can cause damage)



⇒ We have to transform 'm' to smn else, o/w it will be attacked.

$E(m) = \boxed{C}$
(Encryption)



E can access c but it is already transformed.

$D(C) = m$
(Decryption)

- What if E could decrypt it? invert it?
→ It should be impossible for E to invert but allow B to invert.

- **E & D are known to the attacker.**

∴ Even if the attacker knows the D algo, he should not be able to ~~[illegible]~~ invert it.
This is the Kerckhoff's Principle
(Military Advisor)
→ First principle of cryptography.

→ Maybe all parameters aren't known to attacker.
∴ There must be some info known only to A & B & not E (attacker or rest of the world).
- only B can do AND E cannot do.
- This info is called 'secret key'
(only with A & B & not with any1 else)

∴ $E(k, m) = c$ & $D(k, c) = m$ (k: key)

- what if smn can find 'k' from 'c'?

Eg:
$$E(k, m) = km = c$$
$$= (2)m = c$$
(2 can easily be computed.)

DATE : / /
PAGE NO. :

- E is watching $C_1, C_2 \ldots C_{10}$
  - He can see the pattern & figure out the K.
  - If K can be computed by E, system is not secure.

* CRYPTOGRAM : puzzles (google it)

(set) $S = \{A, B, C, \ldots Y, Z\}$

Now we run encryption on S such that

$A \rightarrow D$
$B \rightarrow E$
$\vdots$
$Z \rightarrow C$

I A M P A N D U
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
L D P S D Q G X

$\therefore$ L D P S D Q G X ⟶ original word?

- This is called Shift cipher
  - can be done by computer (only 25 shifts max)
  - Shift cipher is easy to break
    (used by Julius Caesar)

- Transformation

(A, B, C, D, E) ⟶ (1, 2, 3, 4, 5)
⟶ (3, 1, 5, 4, 2) (permutation) ← secret key (random)
⟶ (C, A, E, D, B)

DATE: / /
PAGE NO.:

Total no. of permutations = 26!

- If our computer takes 1 sec for 1 million calc., for 26!, it will be 3170979 20 years.
- CRYPTANALYSIS
  - Don't really need 26! calculations

(m) → A B C D
(5, 21, 6, 8 - -)
↓ ↓
(c) → E R

- This is called Permutation Cipher

Eg: T G E E M N E L N N T

What is the original msg?

Solution

→ Frequency analysis

- E is the most frequently used alphabet.

Eg: N → E

Statistics of English character distribution

Frequency of letters:

E > A > R > I > O > T - - - > Z > J > Q

∴ We look at the frequencies & reduce the search space of permutations.

Guess & Try

DATE : / /
PAGE NO. :

⊙ Though we have 26! keys, the cipher text preserves the statistics of plaintext.
- Frequency analysis can help us get the original msg back.

⊙ Encryption algo preserves the statistical properties of plain text in cipher text.

★ Shannon - Information Theory
- Cryptography (Classical crypto)
(Formal crypto)
(Mathematical crypto)

• Shannon Theory
- Enc algo should not allow anyone to guess anything about the plain text.

Eg: Pick any no. from 1 - 100 [1-100]
(∴ P(1) → 1/100 ____ P(100) → 1/100
P(2) → 1/100
(Picked num) x + y = 40 ← (info given to us.)
(+ve/-ve/0)

- Now 'x' could still be anything
- We are just as clueless.
∴ This cipher text gave no clue on plain text.
∴ cipher text should have independent distribution for every key.

Eg: x + y = 40 ← could be hiding any 'y'.

∴ Perfect Secrecy

DATE : / /
PAGE NO. :

Perfect secrecy :

M is the message space $\{m_1, m_2, \ldots m_n\}$
k is key
c is ciphertext space
(Probability)
$\therefore P(m)$

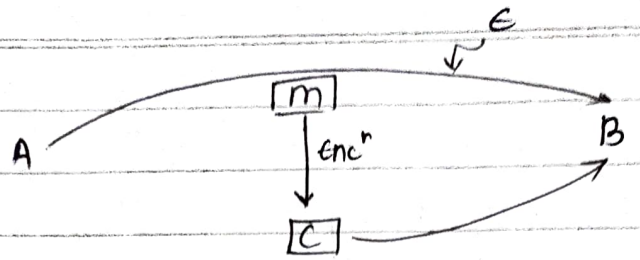
$\therefore P(m/c) = P(m)$ (If c is given, $P(m)$ remains same)

Even if plain text is biased, we are clueless abt it coz of c.

↑ Perfect secrecy.

DATE : 18/8/22
PAGE NO. :

Thursday



→ E should not be able to get m from c.

Goal : Attacker should not recover m from c.

M → message space
K → Key space
C → ciphertext space
E (k, m) → c
D (k, c) → m

} → known to attacker

What attacker DOES NOT know → The KEY.

How to assure that attacker won't succeed?

→ Know yourself - (Ramana Maharishi)

(Who am I)

→ Know your Enemy - Art of War -

- We need to put a password that can't be cracked
→ ∴ We lean on Probability Theory
- Gives you way to deal with uncertainty.
- Take help of Password creators.
- Yes / No → 1 bit of information.

DATE: / /
PAGE NO.:

Eg:

$M = \{0,1\}^{128}$ → 128 bit string

→ could be any string

→ how to express this?

$M = m_1, m_2 \ldots\ldots m_{2^{128}}$

$\frac{1}{2^{128}}, \frac{1}{2^{128}} \ldots\ldots \frac{1}{2^{128}}$ ← very vague

- $M = m_1, m_2 \ldots\ldots m_{2^{128}}$ } prior knowledge

  Prob. space $= p_1, p_2 \ldots\ldots p_n$

  → Discrete distribution

Q. $x \in [1,100] \rightarrow p_i = 1/100$

if $x + r = 40$, $r \in R \rightarrow p_i = 1/100$ still (no change)

$\therefore P(m/c) = P(m)$

(Prob of 'm given c' = Prob of m)

- For more than 1 variables, joint dist^r.

$P(M=m, C=c)$, $P(M=m/C=c)$

$\therefore P(M=m) = P(M=m/C=c)$

DATE : / /
PAGE NO. :

if $x+r=40$ and r is even → Given

∴ We can say x is even

∴ last bit of $x=0$ (in binary expansion)

$P(X=1)=0$, $P(X=2)=1/50$

$P(X=3)=0$, $P(X=4)=1/50$

X :	1	2	3	4	5	6	- - - -
P(X) :	0	1/50	0	1/50	0	1/50	

Q. How many ~~inf~~ questions you need to ask to know the answer, OR, how much info needed to be certain about the answer.

A. Information eliminates or reduces uncertainty

⊙ 'You do not understand what you cannot measure'
- Lord Kelvin

⊙ ~~Quantifying uncertainty~~

Shannon:

Entropy Function →

$$-\sum p_i \log(p_i) = \sum p_i \log\left(\frac{1}{p_i}\right)$$

{Measure of uncertainty}

- Reduction in uncertainty ~~is~~ gain in certainty

DATE : / /
PAGE NO. :

Distribution	Entropy/uncertainty
P	H(P)
Q	H(Q)

$\therefore$ $H(P) - H(Q)$ ← information available

↑ Reduction in uncertainty

let $P(x) = \frac{1}{100}, \frac{1}{100} \dots \frac{1}{100}$

$\therefore \sum_{i=1}^{100} \frac{1}{100} \cdot \log(1/1/100)$

$= 100 \times \frac{1}{100} \times \log(100)$

$= \boxed{\log 100}$

↑ 'log 100' uncertainty

If $x+r = 40$, $r \in$ even, $x \in [1, 100]$

$\therefore \sum_{j=1}^{50} \frac{1}{50} \times \log(1/1/50) = 50 \times \frac{1}{50} \times \log 50$

$= \boxed{\log 50}$ ← uncertainty

DATE : / /
PAGE NO. :

∴ Reduction in uncertainty = log(100) − log(50)
= log(100/50)
= log(2)
= 1

How much info you got ⟶ 1 bit of info
(i.e., last bit = 0)
even

If Reduction = 0, means no info leaked,
then the ~~inf~~ text is completely secure
→ Perfect secrecy achieved.

(a, b, c, d, e) —shifted by 2→ (c, d, e, a, b)
req: 45, 20, 70, 25, 100 ⟶ 45, 20, 70, 25, 100
↓ ↑
cipher text analyses

∴ In cipher text
P(c) = 45/250, P(d) = 20/250 — — — P(b) = 100/250

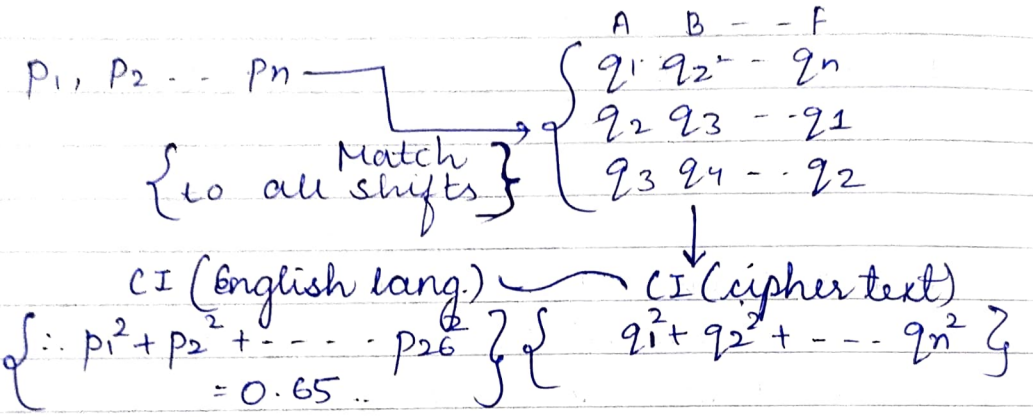
We know, in general english lang.
P(a) = 45/250, P(b) = 20/250 — — — P(e) = 100/250

∴ We can say mapping looks like: (a, b, c, d, e) plain
↓
(c, d, e, a, b) cipher

Coincidence Index (CI) = $\sum_{i=1}^{n} (P_i)^2$

HOMEWORK

① Get distribution
② Do calculations & match count



Approaches

let $p_1 \ldots q_3$
$p_2 \ldots q_4$
$p_3 \ldots q_5$
$\vdots$
$p_{24} \ldots q_{26}$
$p_{25} \ldots q_1$
$p_{26} \ldots q_2$

Shift = 2 ← Encr algo

$\therefore p_1 q_3 + p_2 q_4 + p_3 q_5 + \ldots p_{26} q_2$

$= 0.65\ldots$ (close)

$\sum_{i=1}^{26} |p_i - q_i|$ ← do for every shift, find min

Prob 1.1 & 1.2, Submit in a week.

DATE : 23 / 8 / 22
PAGE NO. :
Tuesday

Basics:

$c = E(k, m)$ $\quad m = D(k, c)$

(E) knows: c, E/D Algos,
E doesn't know m, k

Eavesdr ← opper

Q: Is there any statistical pattern in c?

- Distribution of chars-

M ⟶ P(M=m)

↑ Msg ↑ Probability Dist°

Cipher Text (c) → $q_0, q_1 \ldots\ldots q_{25}$
(a) (b) (z)

Original Text → p_0 p_1 ⋯ ⋯ p_{25} ← freq distrn
(English lang.) (a) (b) ⋯ ⋯ (z)

if $a \to c$, $b \to d$
then $p_0 \to q_2$, $p_1 \to q_3$

$\therefore p_0q_2 + p_1q_3 + \cdots + p_{23}q_{25} + p_{24}q_0 + p_{25}q_1$
$= p_0p_0 + p_1p_1 \cdots + p_{25}p_{25}$
$= \Sigma p_i^2 =$ 0.065

DATE : / /
PAGE NO. :

$\alpha_0 : p_0q_0 + p_1q_1 + \dots + p_{25}q_{25}$
$\alpha_1 : p_0q_1 + p_1q_2 + \dots + p_{25}q_0$
$\alpha_2 : p_0q_2 + p_1q_3 + \dots$
⋮

See which $\alpha_{(K)}$ is closest to $\sum p_i^2 = 0.065$
↳ shift factor

---

① **Shift cipher** : $(x+a) \bmod 26$.
$(a, b, \dots z) = (0, 1, 2 \dots 25)$

Eg: If $a = 3$, z will map to c
($z = 25$, $25+3 = 28$, $28 \% 26 = 2 = c$)

② **Affine cipher** : $(ax+b) \bmod 26$.
$y = ax + b$
$y - b = ax$
$a^{-1}(y-b) = x$.

Inverse
Modulo world.
for n, $\{0, 1 \dots n-1\}$
operators : +, ×

Eg: ① $7+8 \bmod 11 = 4$
$\therefore 7+8 = 4$ (in mod-11)
② 13 $\{0 - 12\}$
$7+8 = 2$ (in mod-13)

∴ Inverse in modulo world.
In regular $5 + (-5) = 0$
↑ inv.

DATE : / /
PAGE NO. :

∴ In mod. world.

$8+3=0 \pmod{11}$

$3 = -8$ } inverses (additive)

$8 = -3$ }

Multiplicative Inverse

$3^{-1} \bmod 11$?

mult. inv.

$3x = 11k+1$.

$3 \times (4) = 12$

$8^{-1} \bmod 11$?

$8x = 11k+1$

$8 \times 7 = 56$. ∴ $8^{-1} = 7$

$7^{-1} = 8$

$8^{-1} \bmod 13$

$8 \times (5) = 40 = 39+1$ $8^{-1} = 5$

$5^{-1} = 8$

$443^{-1} \bmod 7761$? Does it exist?

$3^{-1} \bmod 12$? → does nt exist

$3x = (12k+1)$ ← not div. by 3.

↑

div. by 3 ∴ No such k exists

DATE : / /
PAGE NO. :

∴ $a^{-1} \bmod n$ exist only iff $GCD(a,n) = 1$

a & n should be coprime

Extended Euclid's Algo (See comment)

If $gcd(a,b) = d$,

$\exists\ x, y$ s.t. $ax + by = d$

$ax + by = 1$

∴ in mod-n,

$ax + ny = 1$

or

$ax = 1 \bmod n$

$\Rightarrow x = a^{-1} \bmod n$

In ~~append~~ affine cipher

$y = (ax + b) \bmod 26$

If $a = 3, b = 5$

$\therefore y = (3x + 5) \bmod 26$

$j = 10$

$y = (3 \times 10 + 5) \bmod 26$

$= 9 = (i)$

$j \longrightarrow i$

Enc Algos:

Shift, substitution, Affine, Vigenere

DATE: / /
PAGE NO.:

$a^{-1}(y-b) \bmod 26$ $\quad (a,26)=1$

$a \in \{1,3,5,7,9,11,15,17,19,21,23,25\}$

$\phi(26) = 12.$ $\quad a \in 12$, $b \in 26$ $\longrightarrow$ $12 \times 26 = 312$

- **Vigenere Ciphers**

same char may map to diff. chars in diff places.

let $m = 6$;

(key ~~word~~) → C I P H E R

(2, 8, 15, 7, 4, 17)

key

in Book.

see page 26-27

Transformation in block of 6 chars.

Plaintext (m): ~~thisciphe~~

thiscryptosystemisnotsecure

∴ t h i s c r y p t o s y s

19 7 8 18 2 17

add (mod 26) → Enc to: 2 8 15 7 4 17

(key)

21 15 23 25 6 8

↓ ↓ ↓ ↓ ↓ ↓

V P X Z G I

∴ thiscr ⟶ VPXZGI

DATE : / /
PAGE NO. :

Similarly
yptoSy → A X I V W P

We see
thiScr → Z & yptoSy → W

same char maps to different cipher alphabets

S (18) → 18+2 = 20 (W)
S (18) → 18+7 = 25 (Z)

(2, 8, 15, 7, 4, 17)

∴ Breaking is done in 2 stages
① Find (m) (size of keyword)

In given example
C_1, C_7, C_{13} ... → all shift by 2.
C_2, C_8, C_{14} ... → all shift by 8.
∴ m = 6

∴ look for repition of substrings

→ Prob 2.21 (Prog. Assn), Page 54, 55
(Challenge : solve 2.22 prob).

DATE : / /

PAGE NO. :

Confusion & Diffusion

$Pr(C=c) = Pr(c/m)$ (cipher independent of msg)

Eg: 0 / 1

$P() : 3/4 \quad 1/4$

$K = \{0,1\} \quad \frac{1}{2} \cdot \frac{1}{2}$

$m \oplus k = c$

(XOR)

XOR:

	0	1
0	0	1
1	1	0

If $c=0 \longrightarrow m=0$ & $k=0$

OR

$m=1$ & $k=1$.

$Pr(M=0 / c=0) = 1/2$

$Pr(M=1 / c=0) = 1/2$

DATE: 25/8/22
PAGE NO.:

Thurs.

In vigenere cipher

ciphertext:

$y_1 y_2 y_3 y_4 \cdots$

→ $y_1, y_{m+1}, y_{2m+1}, y_{3m+1} \cdots$ → all shifted by k_1

→ $y_2, y_{m+2}, y_{2m+2}, y_{3m+2} \cdots$ → all shifted by k_2

∴ we need the value of m.

★ See the distances b/w repeating blocks
- Find GCD of those distances
- That will be = m

→ If you aren't sure, without any info
↳ its a uniform distribution
→ we get no clue.

Entropy

$p = (p_1, p_2 \cdots p_n) = H(p) = -\Sigma p_i \log p_i$

- Entropy is maximum when distribution is uniform.
- More bias → low Entropy → less randomness
→ less uncertainty
→ already some info is available that has reduced uncertainty.

- Information is eliminator of uncertainty

Info gained = Reduction in uncertainty

DATE : / /
PAGE NO. :

Eg: 32 students, 30B + 2G,

Info: There is a ~~gold medalist~~ prize-winner (1st Prize) in the class

space = 32

Info 2: It was a boy

space = 30 (2G eliminated)

or

Info 2: It was a girl

space = 2 (30B eliminated) } More info

→ If info eliminates more space, reduces more uncertainty.

$\frac{30}{32}$ → high probability event

$\frac{2}{32}$ → low probability event → removes high amt. of uncertainty

[also called 'surprise element']

Earlier distribution ① = $\frac{1}{32}, \frac{1}{32}, \frac{1}{32}$ ------ ×32

dist^r after info 2 ② = $0, \frac{1}{2}, 0 - - 0, \frac{1}{2}, 0 - - 0$

$$\text{Entropy}_1 = \sum_{i=1}^{32} -\frac{1}{32} \log \frac{1}{32} = \log 32 = 5$$

$$\text{Entropy}_2 = \sum_{i=1}^{2} -\frac{1}{2} \log \frac{1}{2} = \log 2 = 1$$

5 bits reduced

↖ low entropy - gained 4 bits of info

∴ for an attacker, entropy must remain same or increase ~~entropy~~, for him to get no information.

Let:

$H(p) = \sum p_i \log(1/p_i)$ $\quad \langle p_1, p_2 \dots p_n \rangle$

$H(q) = \sum q_i \log$

$= \sum \frac{1}{n} \log(1/(1/n))$

$= n \log n$ $\quad$ (n = size of sample space)

= Entropy of Uniform Distribution = $\log n$

$\therefore H(p) \leq \log n$

$M \rightarrow$ msg space – $P(M=m)$ (apriori info provided)

$K \rightarrow$ key space – $P(K=k)$

$C \rightarrow$ ciphertext space – $P(C=c)$

$Enc(M \times K) \longrightarrow C$

Que → When will you obtain a $c \in C$?

Ans → $Decr(c,k) = m$ and m was chosen as message & k was chosen as key.

$$P(C=c) = \sum_{k \in K} P(K=k) \cdot P(M = D(k,c))$$

$P(X,Y) = P(X=x) \cdot \& \; P(Y=y) \rightarrow$ 2 variable dist^n.

$\Downarrow$

$p(x,y) \; \forall \; x \in X, \; y \in Y$

DATE : / /
PAGE NO. :

~~dist~~

Assumption : dist^n of K is independent of dist^n of M.

$p(x,y) = p(x) \cdot p(y)$

$P(X=x)$ & $P(Y=y)$

$p(X=x, Y=y) = P(X=x) \cdot P(Y=y)$ (product)

for independent variables, we can multiply their individual probabilities.

$[(x,y)]$

$P(x) \longleftrightarrow$

$P(y) \longleftrightarrow$

$X = x_1, x_2, x_3, x_4 - - x_n$

$p_1, p_2, p_3, p_4 - - p_n$

$Y = y_1, y_2, y_3, y_4 - - y_n$

$q_1, q_2, q_3, q_4 - - q_4$

$P(X=x_i, Y=y_j) = P(X=x_i) \times P(Y=y_j)$

$= p_i q_j$

(when X & Y are independent random variables.)

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

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

$P(X=x_i, Y=y_j) = P(X=x_i / Y=y_j) \times P(Y=y_j)$

$$P(C=c \mid M=m) = \sum_{\substack{k \in K \\ \& \\ D(k,c)=m}} P(K=k)$$

Eg:

$m=0$ $\quad P(c=0/m=0) = 8/10$

$m=1$ $\quad P(c=0/m=1) = 2/10$

for $\boxed{c=0}$

$P(c=0$ hiding $0) = 8/10$ ← more

$P(c=0$ hiding $1) = 2/10$

∴ My guess : $\boxed{c=0 \rightarrow m=0}$

---

(1) → $P(C=c \mid M=m_1) = P(C=c \mid M=m_2)$

- in this case, we can't take a guess.
- attacker is equally confused.

- $P(x/y) \cdot P(y) = P(x,y)$
  $= P(y/x) \cdot P(x)$

∴ $P(C=c \mid M=m) \cdot P(M=m) = P(M=m/c=c)\, P(c=c)$

→ $P(C=c \mid M=m) = P(C=c)$

(2) ∴ $P(M=m/c=c) = P(M=m)$

⇒ Even after seeing c, info didn't change

∴ It gave no clue.

Perfect Secrecy (Shannon's)

① $P(M=m)$, $P(K=k)$ (before encryption)

② $C = E(m^*, k^*)$
↑ seen by attacker

③ $P(M=m^* / C=c^*) = P(M=m^*)$
$P(C=c^* / M=m^*) = P(C=c^*)$ } Same.

Que: Do we have a system with perfect secrecy?

Requirements for Perfectly secret system

① Diffusion: eliminate all the prob / stat pattern in the plaintext.

//C & M are independent//

Computation implementation: small change in plaintext should cause huge change in ciphertext.

② Confusion: make rel^n b/w ciphertext (c) & key (k) as complex as possible

ONE - TIME - PAD

binary string:
bitwise XOR

$M \oplus K = C$
XOR

$M = m_1 \; m_2 \; - - - \; m_t$ → $P_i = 1/2^t$
$K = k_1 \; k_2 \; - - - \; k_t$

$c_i = m_i \oplus k_i$

$C = c_1 c_2 - - - c_t$

$(M \oplus K) \oplus K = M$

$C \oplus K = M$

$E(M \oplus k) = C$

$D(C \oplus k) = M$

- One-time-pad : use a key once only.
- Shannon showed : If key reused - perfect secrecy lost
  : key should be as long as the msg

- Attacker has ∞ computing power, so system needs perfect secrecy,
  ∴ → don't reuse key
  → key as long as msg

∴ Need Computational Security
We can't make it 'impossible to break' for attacker
∴ we can make it infeasible to break for him.

- In computational security, we use confusion & diffusion

Next class : Block cipher / stream cipher

DATE : 30/8/22
PAGE NO. : Tuesday

Symmetric Key Crypto system

$E(k, m) = c$ $\qquad$ $D(k, c) = m$

- A & B share the same key → Symmetric Key crypto
& E knows the Enc^n/Dec^n algo
& E does not know the secret key K.
& E has full access to the communication channel C.

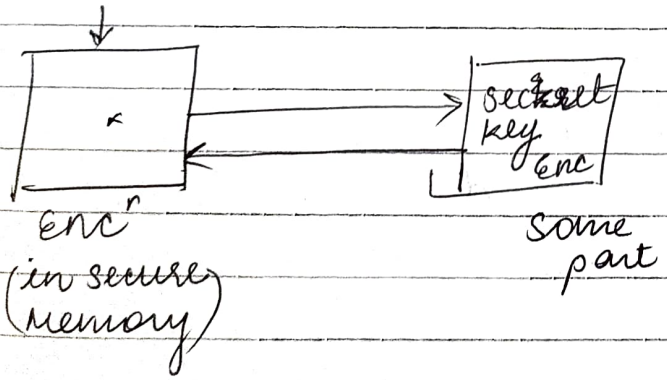
- **Adversary Model / Attack Model**

Let the E (attacker) know msg & corresponding cipher text.

→ (m_1, c_1), (m_2, c_2), (m_3, c_3), — — —

$\therefore c \xrightarrow{?} m$ (can he decrypt the c to m)

This is called the "known plaintext attack".



E has: (m_1, c_1), (m_2, c_2) - - -

chosen by Adversary

This is called "Chosen Message attack"

- tells you the kind of advantages he has with this cipher text.

(A) can he brk it

c

- Is the encryption algo robust?
- ciphertext only attack: only c known
- known plaintext attack
- Chosen plaintext attack
- Chosen ciphertext attack - (B)

Confusion:
→ Remove dependency of ciphertext & key.

Diffusion:
→ ~~Illusion eliminated in cipher text.~~
→ multiple chars shd influence 1 char.
↔ eliminating the dependency of cipher text & plaintext

① substitution:

a	b	c	d	e
↓	↓	↓	↓	↓
e	c	b	a	d

key - the permutation of letters.

Eg: t → x, h → a, e → c

∴ xac → the
ciphertext (ct) (pt) plaintext

∴ By looking at the repeating triplet, we can decrypt those letters

DATE : / /
PAGE NO. :

Transposition cipher

: positions are permuted.

pos. in plaintext : 1 2 3 4 5
pos in ciphertext : 3 5 4 2 1

(PT) (PT with pos jumbled)
∴ as the → ehats

↑ got this after decrypting,
: makes no sense

∴ Even more challenging for cryptanalysis.

c u v j g ← Enc
a s t h e ← P.T

jumble ↳

e h a t s

Decr → g j e v u

∴ c u v j g
→ shift 1 : b t u i f
→ shift 2 : a s t h e ✓ meaning ✓

Eg: g j c v u
① f e b u t
② e d a t s ✓

- Substitution → just a mapping
  → defined by a permutation
  $A \to A$

But if: PT: a b c d e
CT: x y z p t

⇒ specified separately, mapped in different set.
- this is not a 'permutation',
  in permutation, its mapped in same set.

- Transposition : depends only on position

1 2 3 4 5
→ 2 1 4 5 3

i a m p a n d u r a n g a n

(Transpos) ↓

a i a m p d n a u r n g n a n

(substitution applied) ↓

abcde...
txaz...

⇒ Substitution & Transposition,
- D-box (Table repr^)
- Doesn't substitute, only position change

Straight D-box : permutation → invertible

Expanding D-box:

(1, 2, 3)
↓ expanded
(1, 2, 2, 3, 1, 2, 3)

new positions are created

DATE : / /
PAGE NO. :

$\therefore$ (a b c) $\xrightarrow{\text{expanding}}$ (a b b c a b c)

D-box (transpos) ↓ substitution

(x y y z x y z)

- Expansion can't be contracted coz its 1 to many.

$\therefore$ a b c d e $\xrightarrow{\text{subs}}$ 1, 2, 3, 4, 5 $\xrightarrow{\text{cont}^r}$ 1, 3, 5

↓

a c e

S-box : tells the substitution
- could be expanding, one-to-one etc - -

cyclic shift : a_1 a_2 a_3 a_4 a_5

cs_2 : a_5 a_4 a_1 a_2 a_3

→ a simple transposition.
→ it is invertible.

Operations : XOR : bitwise exclusive OR.
→ 2 '0' & 2 '1' → same amt. of 0 & 1.
(AND & OR etc are biased)

→ $x \oplus y = y \oplus x$ (commutative)

$(x \oplus y) \oplus z = x \oplus (y \oplus z)$ (associative)

$x \oplus 0 = x$ (unique)

$x \oplus x = 0$ (inverse) ← can tell you if some char is replaced with itself.

$m \oplus k = c$

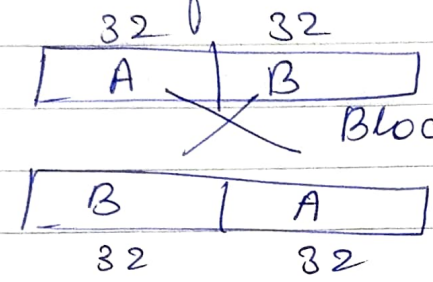
$c \oplus k = (m \oplus k) \oplus k$

$= m \oplus (k \oplus k)$

$= m \oplus 0 = m$

Block swap : Split

- Fixed block size, generally.
- Bit-oriented, not char oriented.
- Block size = 64, 128, 256, 512. - -
- Block = coll^n of bits



Block swap : clean transposition

Confusion, Diffusion

Bit level :

Composition :

- take 'm' —Encr.→ $E(K, m) = C_1$

↓ ← double encr.

$E(K_2, C_1) = C_2$

↓

send C_2.

→ could be applied to transposition as well.

Affine cipher : $x \rightarrow a_1x + b_1 = y$

$y \rightarrow a_2y + b_2 = C$.

No extra protection, just constant changed.

DATE : 1/9/22
PAGE NO. :
Thursday

Block Cipher

- one to one → invertible
  others → non-invertible
- ∴ for security, you want non-invertibility.

- **Self-invertibility**

Bitwise XOR → invertible (self) func^{n}

Reason: $x \oplus x = 0$

∴ x is inverse of itself

$(y \oplus x) \oplus x = y$

- Non-invertible function of a secret key.

$f(k)$ → (can be contraction / expansion func^{n})

(from $f(k)$, can't get k)

$m \oplus f(k) = m'$, say

m → msg, ⊕ → XOR

'k' is known to sender & receiver

∴ they can compute $f(k)$.

$m' = m \oplus f(k)$

$\therefore m = m' \oplus f(k)$

$m' \oplus m = m \oplus f(k) \oplus m = f(k)$

→ Rel^n b/w ciphertext(m') & $f(k)$ is obscure

→ k is not even in the picture, only $f(k)$.

→ This causes confusion

Confusion: obscuring the rel^n b/w ciphertext & key.

→ $f(k)$ → 1-way function.

→ non-invertible

→ arbitrary

- Diffusion: A small change (single bit) in k or plaintext should make huge change in cipher.
  ~~All bits should affect it~~

bit-oriented → block of bits – 64, 128, 256, 512.

Diffusion : statistical pattern removed.

- Difference b/w 2 bit-vectors is calculated by 'Hamming distance'

Eg : 1110 & 1111 → H.D. = 1

1110 & 1010 → H.D. = 1

1111 & 0000 → H.D = 4

1111 & 1111 → H.D = 0 (identical bit vectors)

$A = a_1 a_2 \dots a_n$, $B = b_1 b_2 b_3 \dots b_n$

Hamming Distance $(A, B) = |\{ i \mid a_i \neq b_i \}|$

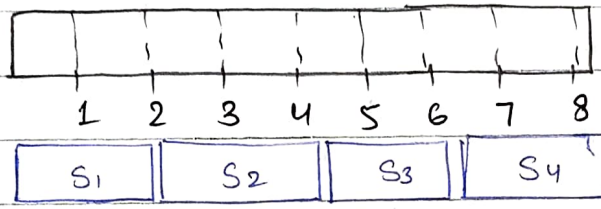
DATE : / /
PAGE NO. :

$a_1 a_2 - - a_n$ —encryp. as→ m

$a_1 a_2 - - a_n$ ——————→ m'

H.D (m, m') = large.

- D-box, S-box — use them over
- Cascade (composition, several rounds)
- Each round → diff^n key ($k_1, k_2, k_3 - -$)
- called 'key schedule' (= sequence of keys used)



← 8 bit plaintext

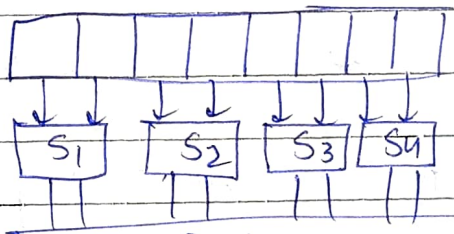
S-box substitution

whitening: obscuring plaintext with $f(k)$

$k_1 \oplus m$

(later we'll do $f(k_1) \oplus m$)

∴ whitened plaintext:



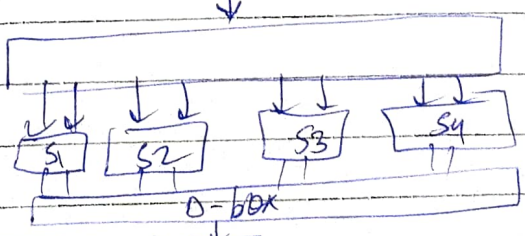
Transposition (Re-positioning) → D-box

↓

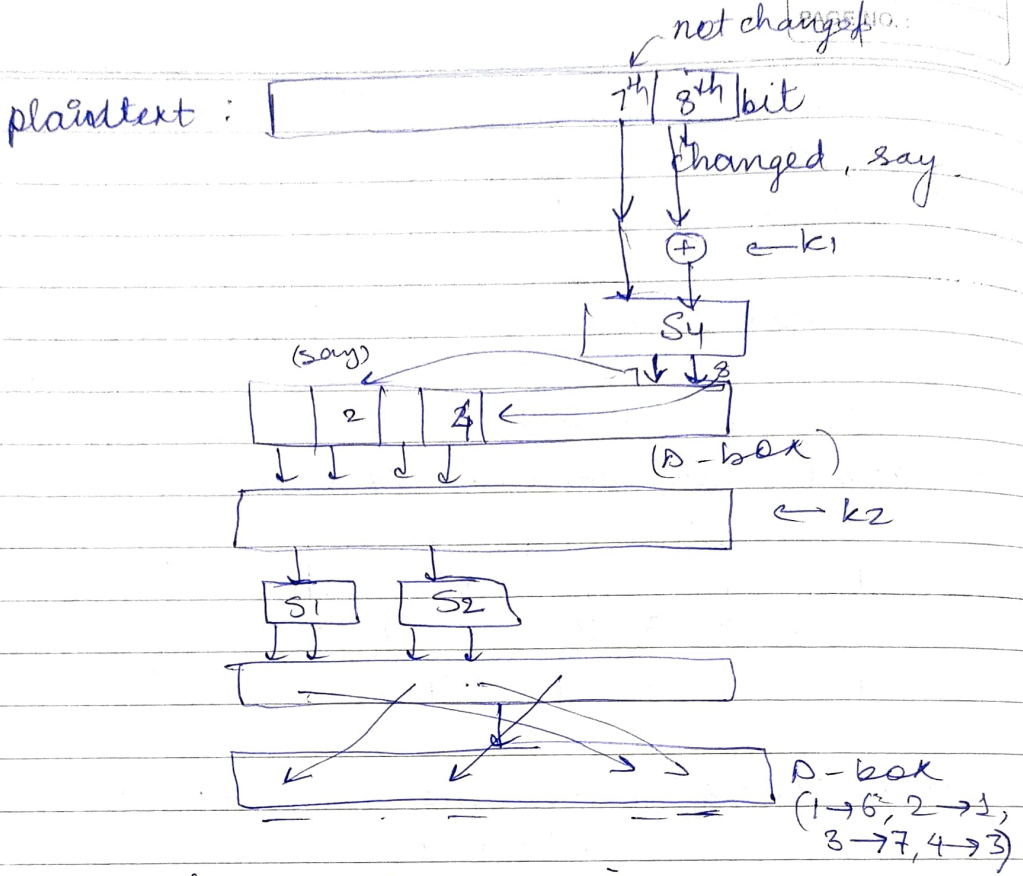
Middle text

↓ subject to further encryption with k_2 (whitening)

$k_2 \oplus$ midtext



cipher text



∴ changes seen in many places.

∴ cascading causes diffusion

★ one-to-one P-box = P-box
↑ permutation

working only on positions

★ S-box → substitution → one-to-one mapping

→ to same set ⇒ Permutation

→ to diff. set → arbit

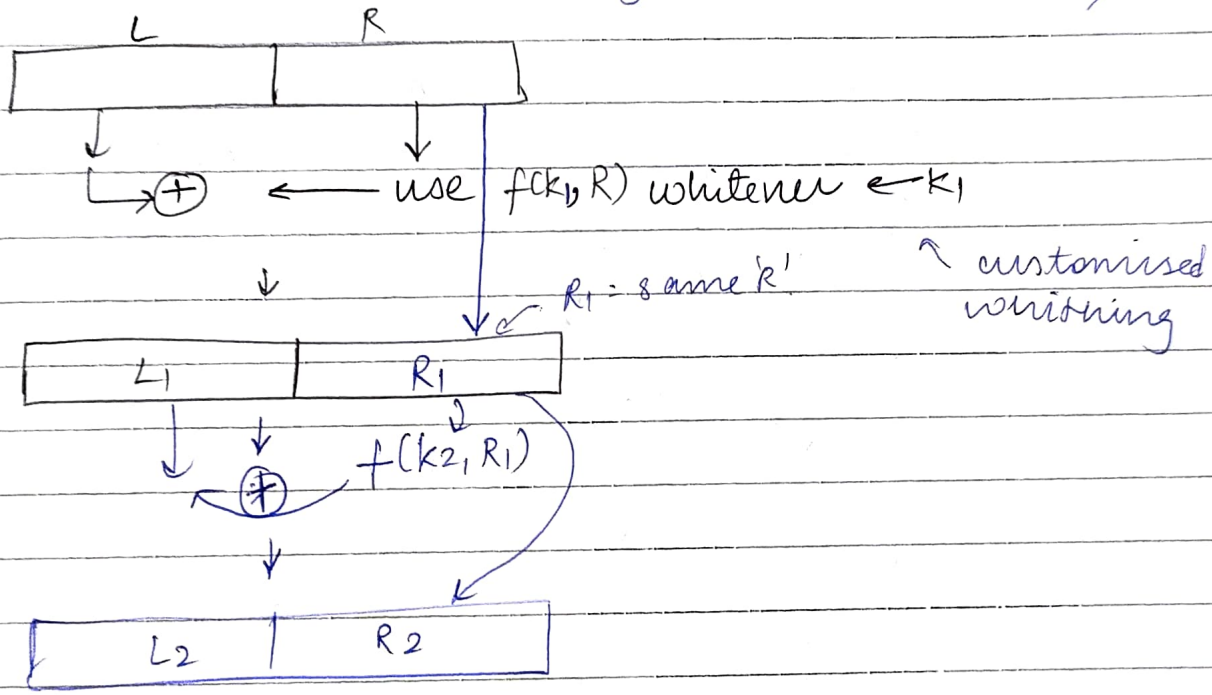
S-box → could be many to 1 (contract) or 1 to many (expansion)

DATE : / /
PAGE NO. :

Now, we'll whiten (Mixer) with $f(k)$.
i.e., $m \oplus f(k)$

Inversion
$= m \oplus f(k) \oplus f(k) = m$

Since XOR is self-invertible, ∴ although $f(k)$ is a non-invertible & we use it for mixing / whitening, we can invert the result (Reason: mixing done with XOR)



Weakness of this scheme : same R_0.
: attacker can figure out half the msg.

∴ Bring another self-inverting operation
"swap the half"
→ this transformation is self-inverting.

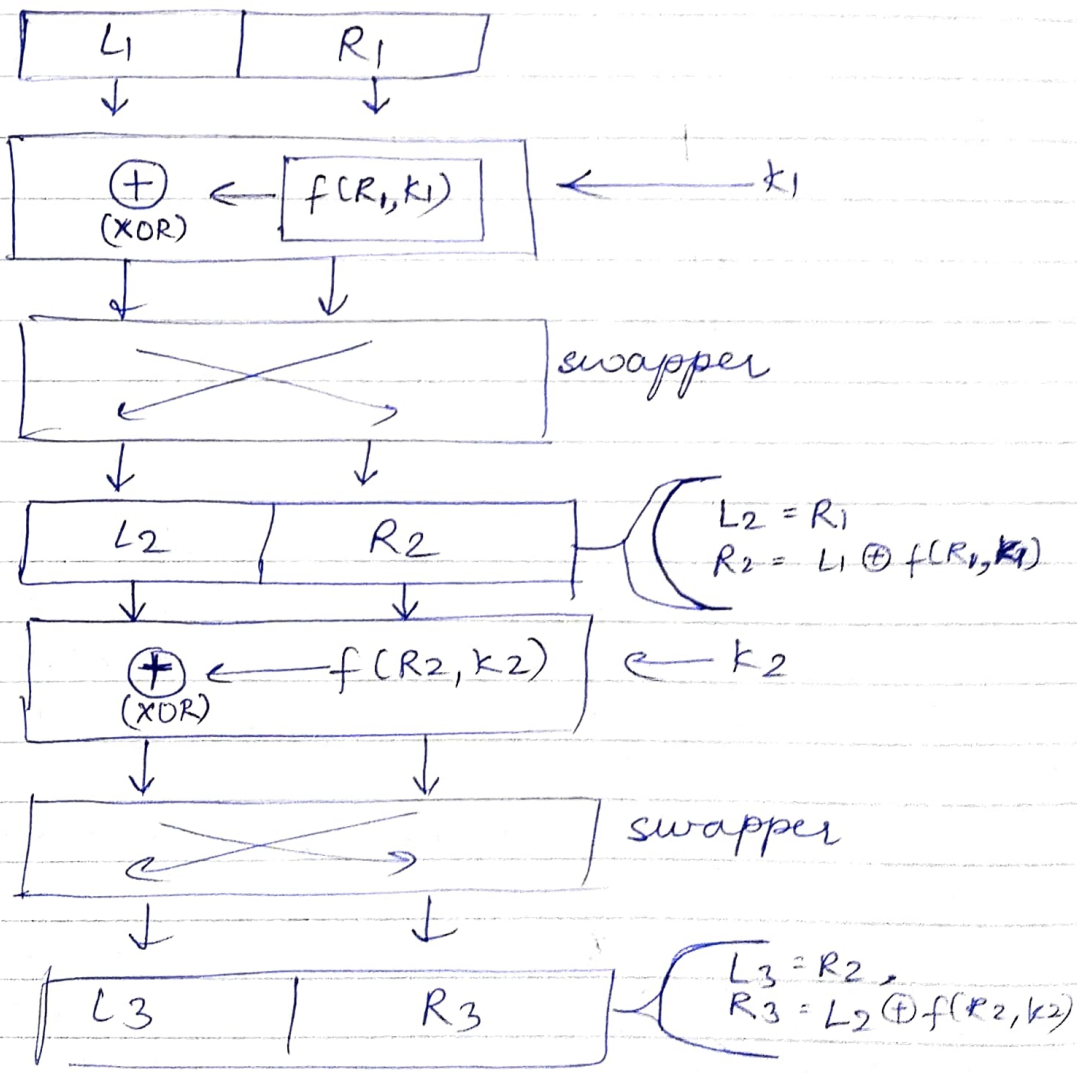
$x \oplus x = 0$

Why did nee not whiten the R?

- ~~We need R to~~
- Receiver needs R to go back like $L_1 \oplus f(K_1, R)$
- If R is whitened, there is no way to go back to R.
- We used R to make secret key, ∴ this R should be available to B to decrypt it

∴ We use 'swap the 2 halves.'

→ If we swap back, nee get the ~~##~~ original back



A person seeing $[L_3|R_3]$, go to $[L_2|R_2]$

$\therefore R_2 = L_3$

$L_2 = R_3 \oplus f(L_3, K_2)$

$\therefore$ Now we can trace back.

- swapper is a self-inverting D-box.

Eg: Blocksize = 64, Keysize = 128,

$\{0,1\}^n \rightarrow$ n bit string

$f(\{0,1\}^{32}, \{0,1\}^{128}) \rightarrow \{0,1\}^{32}$

$f(R_1, K_1) = \{0,1\}^{32}$ $\quad |R| = 32, |L| = 32$

$\oplus$

L_1

L_2

If blocksize = b, keysize = k

then

$f(\{0,1\}^{b/2}, \{0,1\}^{k}) \rightarrow \{0,1\}^{b/2}$