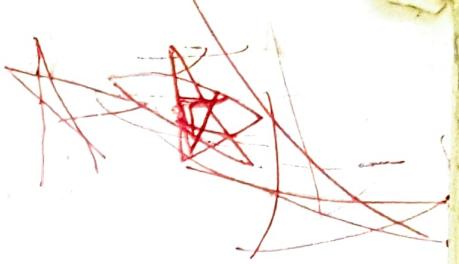


## h2: classical Encryption technique



### 2.2 Substitution Techniques

Caesar cipher: e.g.: replace each letter three places down

Plain: meet me after toga party

Cipher: PHTHW pH DIWHU WKH WRJSD SDUWB

$$c = E(3, p) = (p+3) \bmod 26 \quad // \quad p = \text{plain text}$$

$c = \text{cipher} //$

$$= E(k, p) = (p+k) \bmod 26 \quad // \quad 1 \leq k \leq 25$$

$$p = D(k, c) = (c-k) \bmod 26.$$

Brute-force cryptanalysis is easily performed.

Try with all 25 possible keys.

Sol'n: use large key space.

e.g.: 168-bit key for triple DES ( $= 2^{168}$  possible keys)

### Mono-alphabetic ciphers

cipher can be any permutation of 26 letters.

Then  $26!$  ( $= 4 \times 10^{26}$  possible keys, which is 10 times greater than DES) possible keys are there!

example page 37: The cipher given

WZQS ---

MQ

HJ

A powerful tool is -digrams.

Monocryptographic ciphers are easy to break ~~as for~~ ~~releas~~ as they reflect few of data of the original alphabet.

Playfair cipher Soln: letter e could be replaced with 16, 74, 35, 21 - homophone.  
Still breakable!

Playfair cipher multiple letter encryption cipher  
1854, WWII

Your message: SECRET MESSAGE

SECRET MESSAGE

Encoding

Cipher: ~~NORR~~ NO RD KU NK QZ PC ND

K	E	Y	W	O
R	D	A	B	C
F	G	H	I/J	L
M	N	P	Q	S
T	U	V	X	Z

Rules: 1. Split into pairs

2. for duplicated letters, insert x

3. for odd letters at end, insert x, ignore spin

Put each pair into a separate table

1. If in same column

→ Move each letter down ONE

→ Upon reaching end of table wrap around.

2. If in same row.

→ Move each letter right ONE

→ Upon ---, wrap around.

3. If it forms rectangle

→ swap the letters with ones on the end  
of the rectangle.

SB: for rule 3: S becomes N, B becomes O,

CR: rule 2: C || R, R || D

ET: rule 3: E || K, T || U

MB: rule 3:

SK: rule 3:

SA: rule 3:

GB: rule 1: G becomes N, E becomes D

$26 \times 26 = 676$  diagrams. → a bit difficult. freq analysis

is much more difficult.



## Polyalphabetic Ciphers

Vigenere cipher: (Following sum)

$k_0, k_1, k_2, \dots, k_{m-1}$ Key ( $m \in n$ )	3	4	2	4	
$p_0, p_1, \dots, p_{n-1}$ plaintext	22	4	0	17	
$c_0, c_1, \dots, c_{n-1}$ ciphertext	25	8	2	21	

$$c = c_0, c_1, \dots, c_{n-1} = E(K, P) = E[(k_0, \dots, k_{m-1}), (p_0, \dots, p_{n-1})]$$

$$= (p_0 + k_0) \bmod 26, (p_1 + k_1) \bmod 26, \dots, (p_{n-1} + k_{n-1}) \bmod 26$$

$$c_i = (p_i + k_i \bmod m) \bmod 26.$$

why?

Assignment: Make freq table for Bengali letters  
graph

letter freq information is obscured!

However, not all plaintext structure lost.

Assignment [Cryptanalysis of all ciphers]

use non-repeating key words  $\rightarrow$  See book

Vernam Cipher: 1918, AT&T

Works on binary data.

$$C_i = P_i \oplus K_i \quad // \text{ compare with 2.3 of Vigenere cipher}$$

$$P_i = C_i \oplus K_i$$

Plain Text = HELLO

Key : D6H3e

One-Time-Pad: giving randomness!

Sender send a non-repeating numbers to  
use RSA for reverse Tx.  
recv.

It works with same principle (as Caesar cipher)

- i.e. shifting letters. But with different key  
for each one!

$$\begin{array}{r}
 \text{HAPPY} \\
 / / / \\
 8 \quad 1 \quad 16 \quad 16 \quad 25 \\
 + 8 \quad +9 \quad +13 \quad +4 \quad +23 \\
 \hline
 16 \quad 20 \quad 29 \quad 20 \quad 48 \\
 \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
 = 3 \pmod{26} \quad = 22 \pmod{26}
 \end{array}$$

$\Rightarrow P \rightarrow C T V$

Note  $\Rightarrow T T$  maps for A P! pp maps to c T!

Once used, never reuse the same key!

Hence, pattern like letter freq is no more! Unbreakable

$$\begin{array}{l}
 p \Rightarrow (p-8) \quad (f-19) \quad (c-13) \quad (t-4) \quad (v-23) \\
 \hline
 H \quad A \quad P \quad P \quad Y
 \end{array}$$

8	19	13	4	23	20
11	10	41	---	---	---
ONE-TIME-PAD					

### Weakness:

- must keep the key!
- The keys should long enough (at least same as the plain text) [length limitation]
- Time consuming
- Exchange of one-time-pad.
- OTP must be <sup>potential</sup> random.

But random numbers generated by computer follow a pattern!

to Action for  
website  
1782/E

Two powerful properties:

1. Shift/key is random
2. Which is uniform freq distr.

What makes it so powerful?

1. ALICE → Caesar cipher needs only 26 possibilities.  
using brute-force search

Assume ALICE - every letter is shifted different letters by One-time-pad.

$$\text{Hence } 26 \times 26 \times 26 \times 26 \times 26 = 12 \text{ million possible}$$

situation. Very hard to find.

The encrypted letter is equally likely distributed in 12m possibilities!

## Q.3: Transposition Techniques

1. Re-arrange, No substitution.

Rail Fence (meet me after the foggy party), with depth 2

m e t n a t r h t g p r y  
e t e f e t e o a a t y

See Book page 49

vulnerable, not good.

## Row Transposition Cipher:

Key  $\Rightarrow$  Unique numbers from 0-9.

Key: 4 3 1 2 5 6 7

Plain Text: a t + a c k p  
 o s t p o n e  
 d u n t i l t  
 w o a m [X Y Z]  
 ↓  
 dummy  
 char.

Cipher: start lowest column to highest in order.

t t n a, a p t m, t s n o, a o d w, c o i x, k o l y, p e t z.

Easy to break!

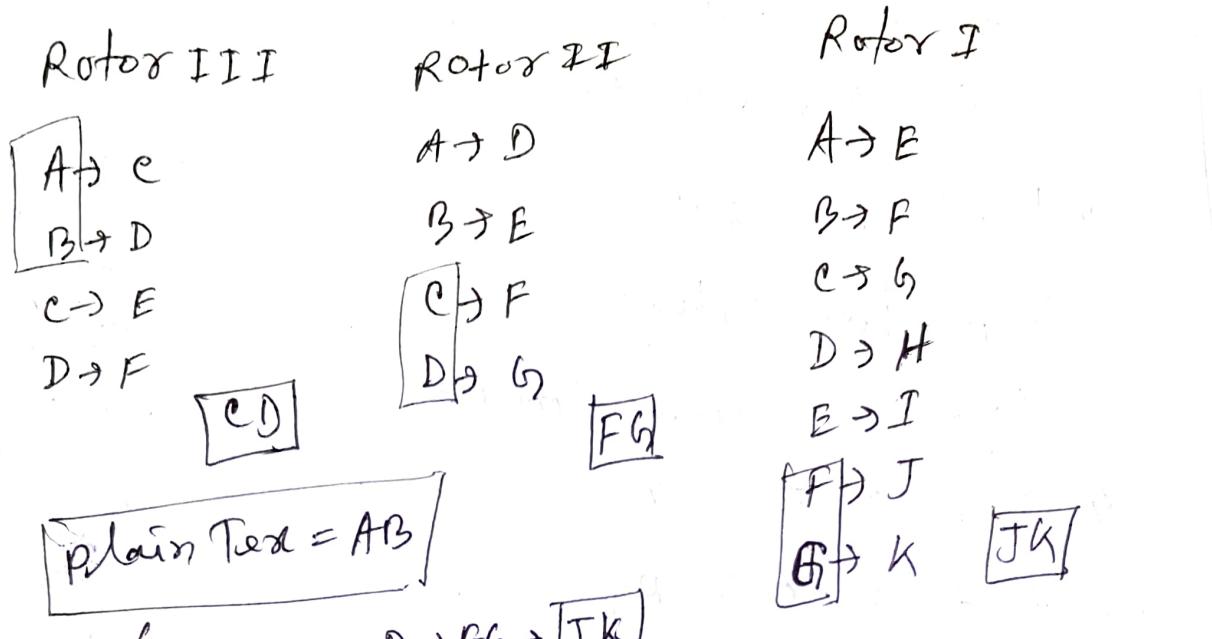
do more than one stage of transposition.

See in book example.

F	T	✓
Death	freq	
2	= 1	
y	x	
z	g y x w o ?	
2 0	[k o r] 1	
3	r a y r 2	
4	f r a y r 2	
5	w o r 2	
7	r a y	
3 + 2 = 2		
3 + 2 = 2 x		
T   F   choose		
②   ②   opposite		

## Rotor cipher

A rotor or electro-magnetic m/c to encrypt and decrypt - famous German Enigma



cipher  $\Rightarrow$  JK

strength of encryption depends on

$\rightarrow$  no. of rotors in system

$\rightarrow$  size of each rotor

$\rightarrow$  no. of rotor types  $\rightarrow$  can be a project

**Enigma Machine - Project**  $\rightarrow$  simple but ingenious invention

Explain

You tube : 1. The inner workings of an Enigma Machine.

1. Principles of Enigma machine

Enigma m/c was an encryption device in WWII by Germans

It had 1.58 quintillion combinations and it was thought to be unbreakable.

See YouTube: Building Enigma machine using principles of CS.

YouTube: Enigma II: Encryption Machine & puzzle - encode and decode cipher

### Chapter 3: Block Ciphers, DES

Feastal cipher/structure vs block cipher.

plain Text,  $p_i$  is divided into bytes.

$p_i :$

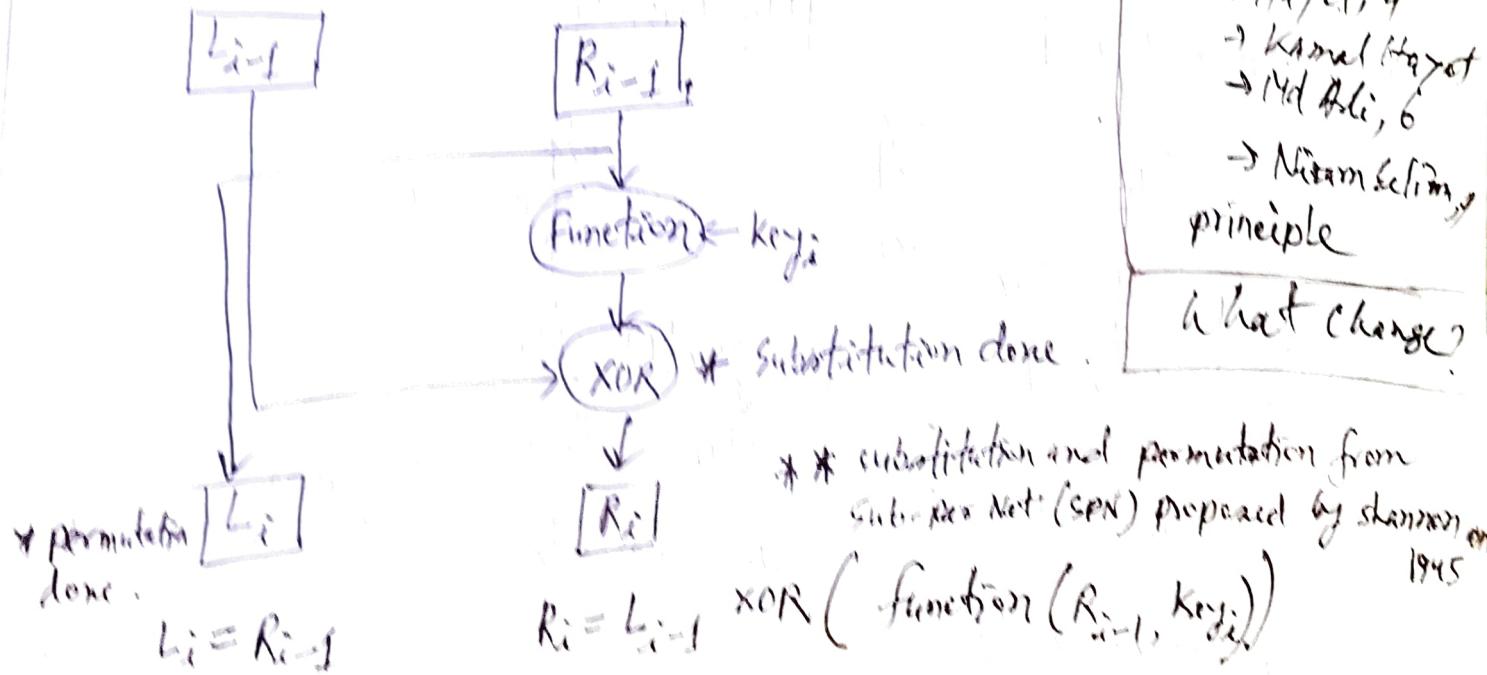
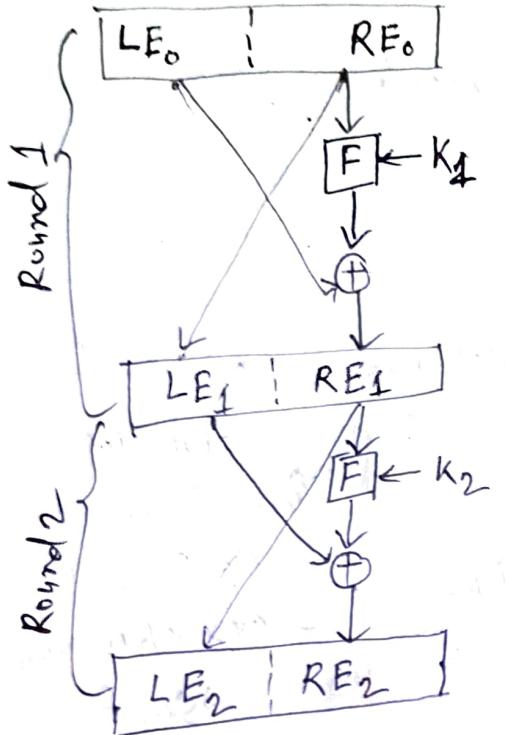


fig: I used for feistel structure.

Parameters: 1. No. of rounds (e.g. 16) 2. Block size (16 bytes)  
3. Key (e.g. 128 bits) 4. Subkeys for each round

Input (Plain Text)



Round 1

Round 2

Round 16

Output (Plain Text)

$$RD_{17} = LE_0 \quad LD_{17} = RE_0$$

$$LD_{16} = RE_0 \quad RD_{16} = LE_0$$

$$LD_{15} = RE_1 \quad RD_{15} = LE_1$$

$$LD_{14} = RE_2 \quad RD_{14} = LE_2$$

$$LD_1 = RE_{15} \quad RD_1 = LE_{15}$$

$$LD_0 = RE_{16} \quad RD_0 = LE_{16}$$

output (cipherText)

figure 3.3 Feistel Encryption and Decryption (16 rounds)

\* Same alg<sup>n</sup> for Decryption  
but  $k_i$  in reverse order

At 16-th round

Encryption:

$$LE_{16} = RE_{15}$$

$$RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16})$$

Decryption:

$$LD_1 = RD_0 = LE_{16} = RE_{15}$$

$$RD_1 = LD_0 \oplus F(RD_0, K_{16})$$

$$= RE_{16} \oplus F(RE_{15}, K_{16})$$

$$= [LE_{15} \oplus F(RE_{15}, K_{16})] \oplus F($$

The XOR has the following properties:

$$[A \oplus B] \oplus C = A \oplus [B \oplus C]$$

$$D \oplus D = 0$$

$$E \oplus 0 = E$$

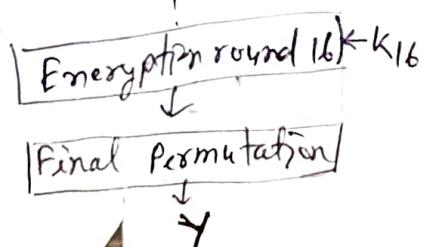
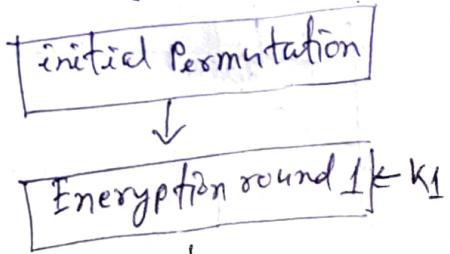
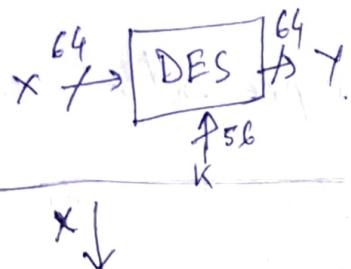
# Lecture 5: Data Encryption Standard (DES) (YouTube)

Few facts:

## 1st Part: DES Intro

- 1974 proposed by IBM with input of NSA.
- Big event in history of cryptography.
- 1977-1998 US standard; 80% of world used it. (e.g. e-passport with 3DES)
- unsecure today (key too short) but 3DES is very secure.

How it works?



Q. How do we build a block cipher?

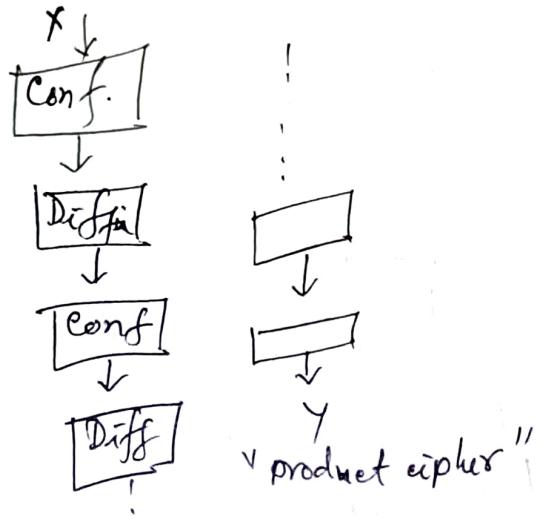
Shannon → proposed two atomic operations.

i) Confusion: Relationship between plain and ciphertext is obscured.  
e.g.: Substitution table (Caesar cipher)

ii) Diffusion: The influence of one each plaintext bit is spread over many ciphertext bits.  
e.g.: Permutation

Combine confusion and diffusion many

many times to build a strong block cipher.



$$\begin{aligned}x_1 &= 00101001 \\x_2 &= 00001011\end{aligned}\xrightarrow{\text{block cipher}} \begin{aligned}y_1 &= 10111001 \\y_2 &= 01101100\end{aligned}$$

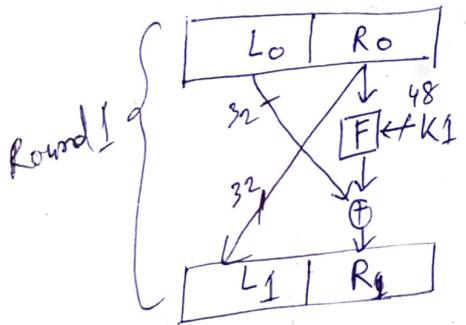
single bit flip      many bit flip  
(diffusion)

Most block ciphers ~~are~~ are product cipher, as they consists of rounds

## 2nd Part: Feistel Network

many of today's ciphers are feistel ciphers. (not all)

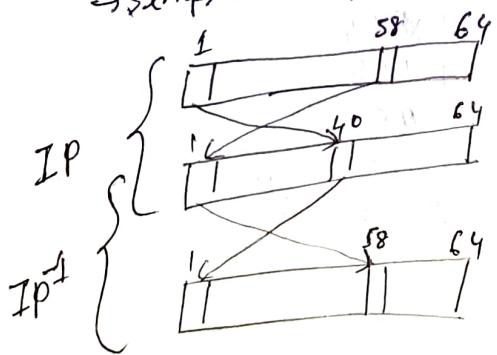
\* See fig in previous of feistel structure



## 3rd Part: DES internals

a) IP and  $IP^{-1}$  (Initial permutation)

→ simple bit permutation.



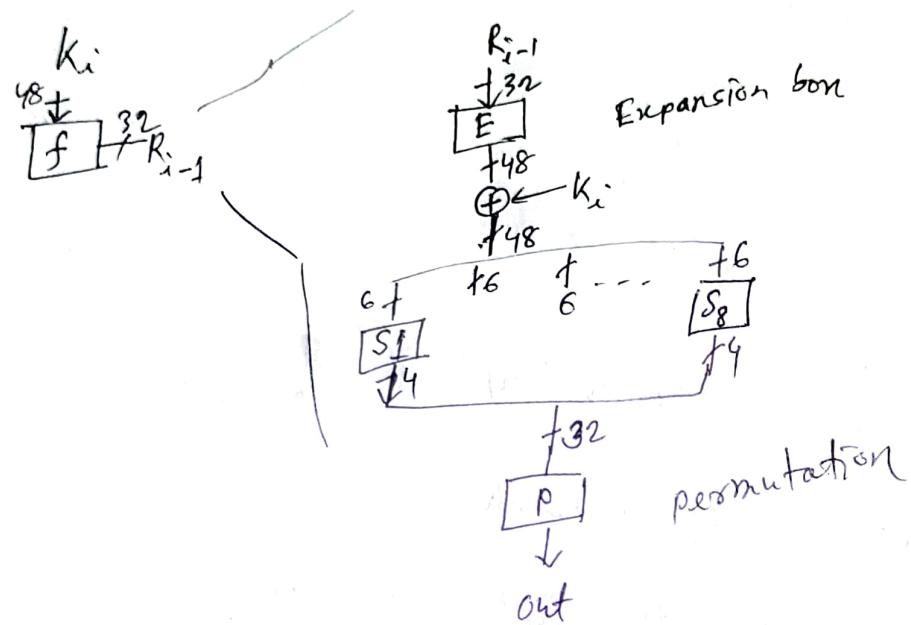
$$IP(IP^{-1}) = I(\text{original})$$

why  $IP, IP^{-1}$  needed?

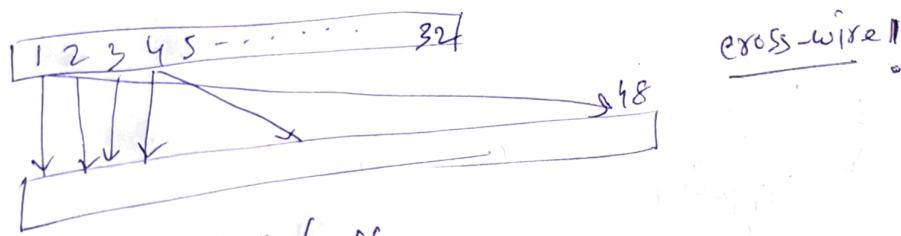
for electrical engg purpose.  
to integrated in H/w ckt.  
in s/w it cause slow down,  
critics say IBM wanted  
to sell H/w.

DES have permutation in  
every round, no need to  
have IP!

### 36: Details of the F-function

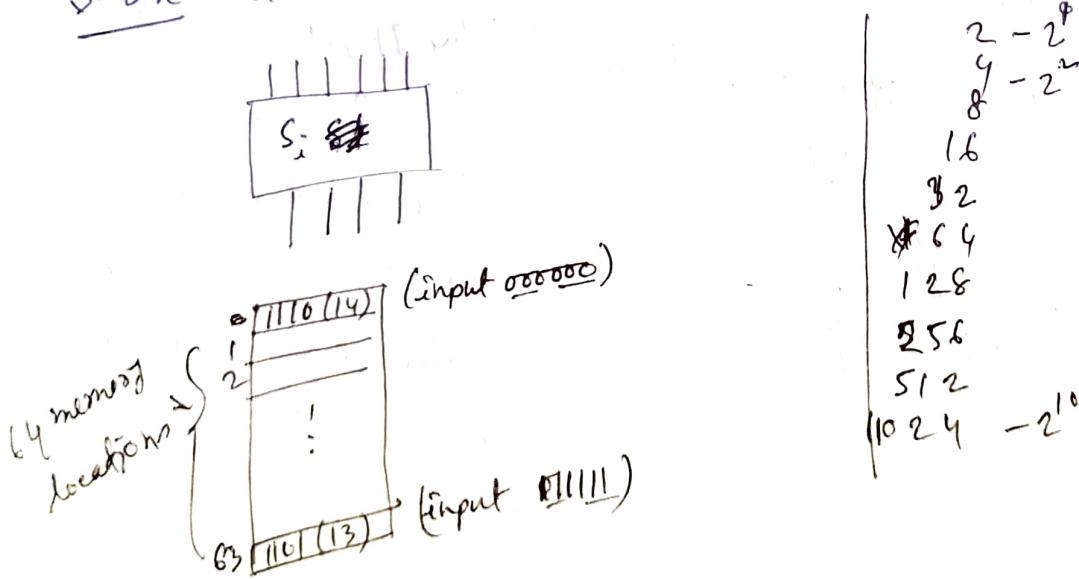


F-box: it provides confusion, diffusion.



16 connected to 16 -  
other 16 " to (16+16)

S-box: it is the heart of DES (it provides confusion).

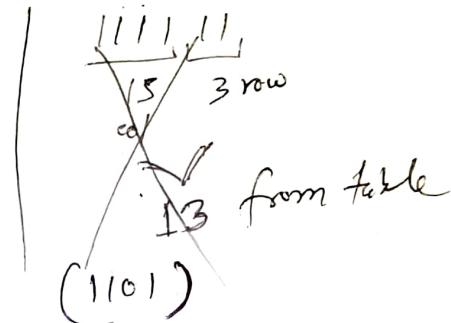
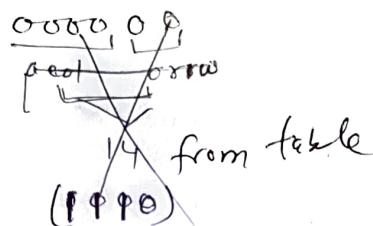
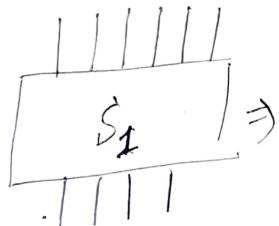


decoded from S-box

## unusual decoding of S-box tables

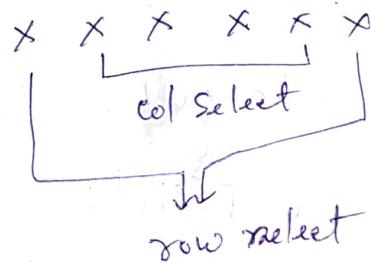
$S_1$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	03	10	6	12	5	9	0	7
1																
2																
3	15	12	8	2												13

$\Rightarrow [S_1 \text{ table}]$



## \* unusual decoding of S-box tables

input bits



Example: What is  $S_1$ -box representation of 37?

$$S_1(37) = S_1(10\underset{\substack{\text{col 1} \\ \downarrow \\ \text{row 3}}}{0}\underset{\substack{\text{col 2} \\ \downarrow}}{1}) = 08 = 1000$$

Why S-box?  $\rightarrow$  IBM says for security. These are 8-boxes for 8-S-boxes.. very hard to break.

- 1997-1997 broke by differential cryptanalysis.

- IBM, NSA knew that the attack coming before researcher.

## 4-Steps in DES

1. Expansion E
2. XOR with round key
3. S-box substitution
4. Permutation.

Q: if 1 bit flips How many change in output?

- each of above 4-steps shows that lot of changes in a round.
- And that's done for 16 rounds!
- lot of change!

Avalanche Effect

A small change in either the plain text or the key should produce a significant change in the cipher text.

(A change in 1 bit in plaintext should produce a change off in many bits of the cipher-text)

## Today:

1. key schedule
2. DES decrypt
3. DES security
4. DES Alternatives.

### DES Key schedule:

Q. How to compute 16 subkeys  $k_1 \dots k_{16}$ ?

See fig 3.14 of Paar Book.

Key schedule consists of simple operations.

#### a) PC-1, Permutated choice-1

$\Rightarrow$  Drops bits 8, 16, 24, 32, 40, 48, 56, 64

$\Rightarrow$  Effective key length of DES:  $64 - 8 = 56$

#### b) $LS_i$ :

Left shift (in fact: left rotate) for each halves.  
( $i$  = round no.)

$$LS_i = \begin{cases} 1 \text{ pos shift}, i=1, 2, 9, 16 \\ 2 \text{ " " }, \text{ all other}, i=3, 4, \dots \end{cases}$$

Total number of bit position shift =  $1 \cdot 4 + 2 \cdot 12 = 28$

$$\Rightarrow C_{16} = C_0, D_{16} = D_0$$

#### c) PC-2, Permutated choice-2

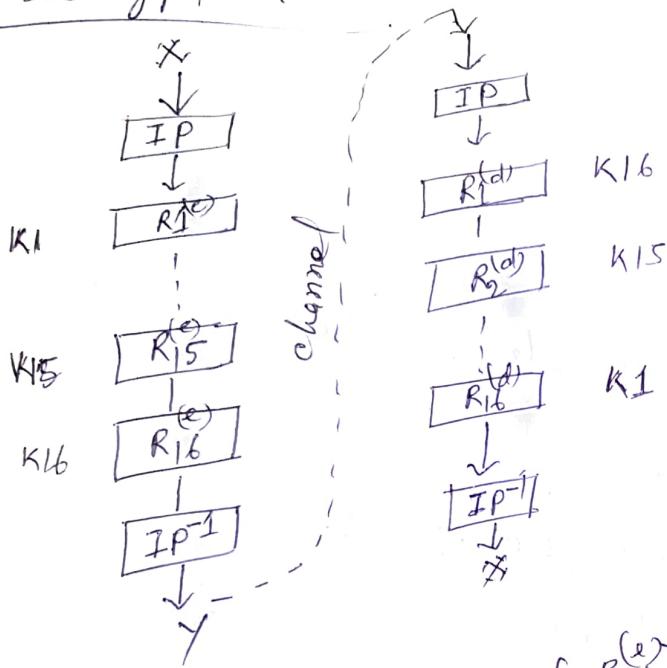


8 bits are dropped, remaining 48 input bits are permuted.

## Observation:

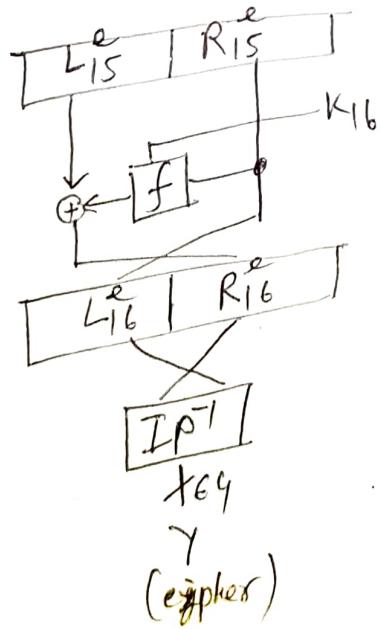
1. Each key  $k_1, k_2, \dots, k_{16}$  is merely a permutation of the original 56 key sets.
2. Key schedule is easy to implement hardware.

## Q: Decryption

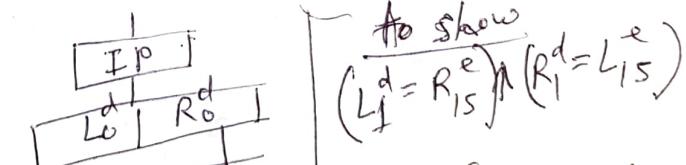


Note:  $R_1^{(d)}$  is inverse of  $R_{16}^{(e)}$  \* Reverse a round?  
 $R_{16}^{(d)} \parallel \parallel \text{ of } R_1^{(e)}$

end of encryption.



beginning of Decryption



To show  
 $(L_f^d = R_{15}^e) \wedge (R_1^d = L_{15}^e)$   
from figure, it  
is clear that  
 $L_f^d = R_{15}^e$

$$\begin{aligned}
 R_1^d &= L_0^d \oplus f(K_i, R_0^d) \\
 &= L_{15}^e \oplus f(K_{16}, R_{15}^e) \oplus f(K_i, R_0^d) \\
 &= L_{15}^e \oplus f(K_{16}, R_{15}^e) \oplus f(K_i, R_{15}^e) \\
 &\quad [R_0^d = R_{15}^e] \\
 &= L_{15}^e \oplus f(K_{16}, R_{15}^e) \oplus f(K_{16}, R_{15}^e) \\
 &= L_{15}^e \oplus \underbrace{\text{000...000}}_{32 \text{ zero}} \rightarrow \left\{ \begin{array}{l} \text{general: } L_i^d = R_{16-i}^e \text{ and } R_i^d = L_{16-i}^e \\ \text{at end of decryption: } \\ IP^{-1}(R_{16}^d, L_{16}^d) = IP^{-1}(L_0, R_0) = IP^{-1}(IP(x)) = x \end{array} \right. \\
 &= L_{15}^e \#1
 \end{aligned}$$

The remaining round reversals work the same:

Round 2<sup>d</sup> reverses round 15<sup>e</sup>



### 3. DES Security:

Two families of attack

a) Analytical Attacks (until 1999 it was unbreakable)

Differential cryptanalysis  $\vdash$  requires  $2^{47} (x, y)$  pairs.

Linear  $\vdash$  needs TB steps!

b) Brute-force attack:

given  $(X_0, Y_0)$

$$\boxed{\begin{array}{l} DES_{K_i}^{-1}(Y_0) \stackrel{?}{=} X_0 \\ i=0, 1, \dots, 2^{16}-1. \end{array}}$$

1998 - Deep Special-purpose DES hardware cracker  
\$250,000

2007 : COPACOBANA - \$10,000

⇒ DES can be broken in few days.

#### 4. DES Alternatives

Cipher	comment
AES	De facto world standard
3DES	Still very secure
AES-Finalists	

Hardware: Permutations such as E, P, IP and  $IP^{-1}$  are very easy to implement in H/W as they only require wiring. but no logic. S-box ( $6 \times 4$ ) also easily realizable in H/W. One S-box requires about 100 logic gates.

A single DES round can be done with 3000 gates. On Modern ASICs and FPGA (Application-Specific ICs, Field Programmable Gate Arrays)

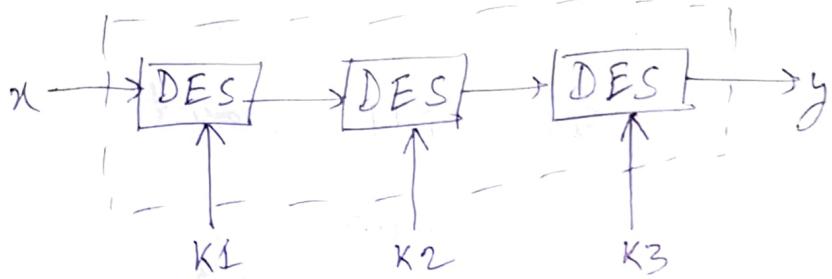
⇒ throughput rate 100 Gbit/sec possible.  
⇒ 3000 gates even fit on low cost RFID chips.

Software: E and P permutations are slow in software. S-box also slower.

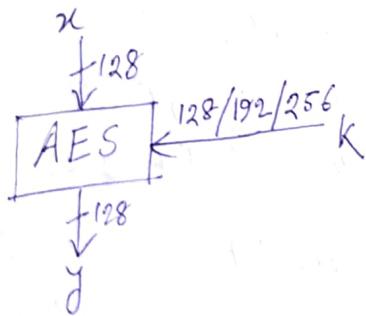
## \* Triple DES (3DES):

consists of three subsequent DES encryption.

$$y = \text{DES}_{K_3}(\text{DES}_{K_2}(\text{DES}_{K_1}(x)))$$



AES:



All internal op are based on Finite Field.

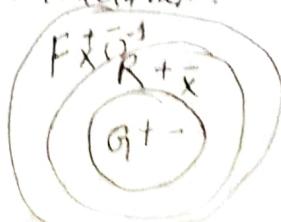
\* Introduction to Finite Fields:

02 Dec 2020

Initial, see - loppe chick (1-1.2kg), separate packet consisting  
of chick, twiga etc. legs all in a packet. shak, if available

Terminology: Finite Field (= Galois Field)

3 basic algebraic structures:



G: Group (+ -)

R: Ring (+ - x)

F: Field (+ - x (1))

See 4.3.2 of Book.

informally: "A field is a set of numbers in which we can add, subtract, multiply and divide".

we can add, subtract, multiply and divide".  
Example: Real numbers, complex numbers, natural numbers.

- in crypto we always need finite sets.

Theory 4.3.1: Finite Field only exist if they have  $p^m$  elements. ( $p$  is prime,  $m$  is integer)  
 $\begin{matrix} \text{smallest} = 2, \\ \text{smallest} = 1 \end{matrix}$

Example:

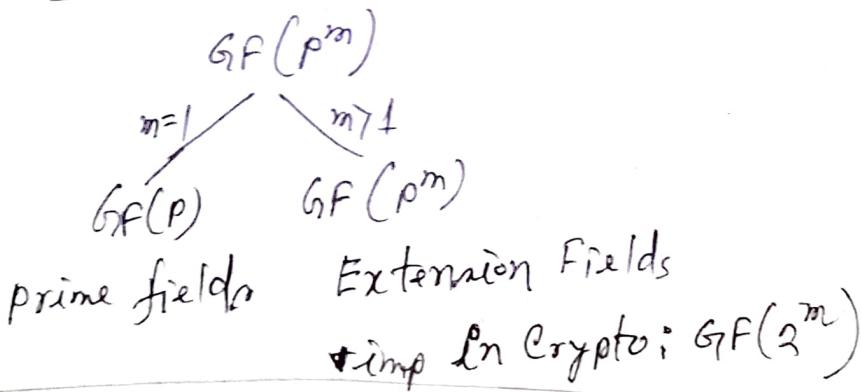
i) There is FF with ~~not~~ 11 element.  $GF(11)$

ii) " " " " 81 " :  $GF(81) = GF(3^4)$

iii) " " FF " 256 " :  $GF(256) = GF(2^8)$

iv) " " " "  $12 = 2 \cdot 3$ , NO FF "AES field"  
used in web browser

Types of FF



### 3. Prime Field Arithmetic

The elements of a prime field,  $GF(p)$  are the integers  $\{0, 1, \dots\}$

a) Add, subtract, Multiply:

Let  $a, b \in GF(p) = \{0, 1, \dots, p-1\}$

$a+b \equiv c \pmod{p} \rightarrow \left\{ \begin{array}{l} a+b \text{ may make tree elements out of range} \\ \text{but } (a+b) \pmod{p} \text{ make sure that it is with range of } (p-1) \end{array} \right.$

$a-b \equiv d \pmod{p}$

$a \cdot b \equiv e \pmod{p}$

Note that all conditions of fields set are satisfied with these computations.

b) Inversion:

$$a \in GF(p)$$

the inverse  $a^{-1}$  must satisfy  $a \cdot a^{-1} \equiv 1 \pmod{p}$ .

How to compute  $a^{-1}$ ?

Ans: Extended Euclidean Alg.

#### 4. Extension Field $GF(2^m)$ Arithmetic:

~~Arithmetic~~

a) Element representation:

The elements of  $GF(2^m)$  are polynomials:

$$a_{m-1} x^{m-1} + \dots + a_1 x + a_0 = A(x) \in GF(2^m)$$

Note: AES,  $m$  is 8, Hence the coefficient is:  $a_0, a_1, a_2$

$$a_i \in GF(2) = \{0, 1\}$$

Example:  $GF(2^3) = GF(8)$



$$A(x) = a_2 x^2 + a_1 x + a_0 x^0$$

$$= a_2 x^2 + a_1 x + a_0$$

( $a_0, a_1, a_2$  all have elements  $\{0, 1\}$ )

$= (a_2, a_1, a_0)$  3 bit vectors, each may 0 or 1  $\Rightarrow 8$  combinations.

$$GF(2^3) = \{0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1\}$$

Quest: How to compute with these elements?

$$\begin{array}{c|ccccc} & 0 & 0 & 0 \\ & 0 & 0 & 1 \\ & 0 & 1 & 0 \\ & 0 & 1 & 1 \\ \hline & 1 & 0 & 0 \\ & 1 & 0 & 1 \\ & 1 & 1 & 0 \\ \hline & 1 & 1 & 1 \end{array} \rightarrow \begin{array}{l} 0x^2 + 1 \cdot x + 0 \\ = x \end{array}$$

## b) Addition and Subtraction in $GF(2^m)$

[Def 4.3.3]

$\Rightarrow$  use regular polynomial add or subtraction, where the coefficients are computed in  $GF(2)$ .

Example:  $GF(2^3)$

$$\begin{aligned} A(x) &= x^2 + x + 1 \\ B(x) &= x^2 + 1 \\ A+B &= (1+1)x^2 + x + (1+1) \\ &= 0 \cdot x^2 + x + 0 \quad (2 \bmod 2 \text{ is } 0) \\ &= x \end{aligned}$$

see why

$$\begin{aligned} GF(2) &= \{0, 1\} \\ GF(3) &= \{0, 1, 2\} \\ 2^0 &= 1 \\ 2^1 &= 2 \\ 2^2 &\equiv 4 \bmod 3 \\ &= 1 \end{aligned}$$

Note: Add and sub in  $GF(2^m)$  are the same operations.

## c) Multiplication in $GF(2^m)$

Intuition: Just do regular polynomial mult.

Example:  $GF(2^3)$

$$\begin{aligned} A \cdot B &= (x^2 + x + 1)(x^2 + 1) \\ &= x^4 + x^3 + x^2 + x^2 + x + 1 \\ &= x^4 + x^3 + (1+1)x^2 + x + 1 \quad // 2 \bmod 2 \text{ is } 0. \\ &= x^4 + x^3 + x + 1, \text{ but it is not in the field!} \\ &= c'(x) \end{aligned}$$

recall prime fields

$$Ex: GF(7) = \{0, 1, \dots, 6\}$$

$$3 \cdot 4 = 12 \equiv 5 \bmod 7 \quad // \text{modulo reduction}$$

Solution: Reduce  $c'(x)$  modulo a polynomial that "behave like a prime".

These are called irreducible polynomials.

# Irreducible polynomial for GF(2<sup>3</sup>)

$$p(x) = x^3 + x + 1$$

$$\begin{array}{r} A \cdot B \\ \overbrace{(x^4 + x^3 + x + 1)}^{x^4} / \overbrace{(x^3 + x + 1)}^{p(x)} = x + 1 \\ \hline x^4 + x^3 + x \\ x^3 + x + 1 \\ \hline x^2 + x \end{array} \quad A \cdot B \bmod p(x)$$

The "AES irreducible polynomial"

$$p(x) = x^8 + x^4 + x^3 + x + 1$$

for every field GF(2<sup>m</sup>), there are several irreducible polynomials.

$$\text{Ex: } p(x) = x^3 + x^2 + 1$$

## d) Inversion of GF(2<sup>m</sup>)

The inverse  $A^{-1}(x)$  of an element  $A(x) \in GF(2^m)$  must

$$\text{satisfy: } A(x) A^{-1}(x) \equiv 1 \bmod p(x)$$

↓  
Ext Eucl. Alg.