Date: / /

INFORMATION ASSURANCE & SECURITY

Defn: Information Security – Protection of info and its critical elements, incl. the systems & hardware that use, store & transmit that info.

– Committee on National Security Systems (CNSS)

Security Goals :-

- Confidentiality –
- Integrity
- Availibility

- Confidentiality - Data should not be available to 3rd party users – only to authorized users/ entities.
- Integrity – While reading, data must not be altered.
  Any changes taking place must be made by authorized users.
  Whatever data is transmitted, the data should not be changed midway. If changed, there should be a mech to identify it.
- Availibility - Data should be accessible/avai whenever needed.

Attacks :-

- Threats to Confidentiality
  Snooping
  Traffic analysis

- Integrity
  Modification
  Masquerading
  Replaying
  Repudiation

- Availability
  Denial of service (DOS)
  Distributed DOS

→ Traffic Analysis – Predicting data by monitoring.
Ex: Suppose you have voltage 1001.
There will be a sequence – can be predicted
e. Monitor traffic through a router, identify packets, sequence of packets based on id, source, dest, can get complete data

→ Snooping – will be able to read data in un-encrypted form.

→ Replaying –

[client] —authentica"→ [Server]
[client] ↓ P1, P2, P3, P4

Authentication message contains, say 4 paramet

Attacker parallely redirects paramters to himself
Server replies to client and authenticates it
Client does work, session closed.
Now without modifying any content, attacker sends message to server.
Server doesn't know whether it is a replay attack or genuine messag.

How to prevent? Server must be able to identify if it is replayed message or original.
① Include one extra parameter – timestamp
↳ to verify if it is fresh or old.
↳ Timestamp should incl. date & time
↳ Disadv: Time synchronization bet" systems

(2) A no. at once - Extra Parameter - a no. can be sent only once.

→ Masquerading - Behaving / imposing as genuine user - given username & password.
Spoofing

→ Modification - login to genuine user acc & change data

→ Repudiation - Ex: send a mail to someone, he reads mail and deletes it & says he didn't recv it. But now, logs are available, everything is traceable.
If sender / receiver denies sending / receiving

→ DOS: service denied to genuine user bec system is engaged in checking a crafted message
Ex: Launch a DOS attack by a 3 way handshake
Sync
Sync + Ack
Ack
Attacker never completes handshake ∴ resources exhausted.

Services & Mechanisms

Security Attack: Any action that compromises the security of info owned by org.

Security Service ⊙ Mech / procedure to enhance the security of data processing systems & info transfers of an org.
⊙ Intended to counter security attacks

Date: / /

detect prevent recover

Security Mech → to used to find, if attack has occured detect prevent security attack, recover data from attack.

⇒ Security Services

① ↳ Data confidentiality – to prevent attacks possible for confidentiality

② ↳ Data Integrity – Att. To prevent attacks possible for integrity
- anti-change
- anti-replay demonstrating

③ ↳ Authentication – identity of user
(HTTPS) · peer to peer (peer entity) – connection oriented comm
(HTTP) · data origin = no need to prove identity – server verifies from which source data is originated

④ ↳ Non repudiation
- proof of origin
- proof of delivery

⑤ ↳ Access Control – Prevent unauthorized access of data

⇒ Security Mechanisms :-

① ↳ Encipherment – Encryption + decryption
→ for confidentiality of data

Defn ⊕ → To convert readable info/data into non readable format
This uses an encryption algorithm

(Readable) Plain text → Encryption algo → Cipher Text (Encrypted data Non Readable)

Encryption algo uses a key

At the dest machine, cipher text must be converted back to plain text : Decryption.

Decryption + key ⟶ Plain text

→ Encipherment is 5 tuples : {E, D, P, C, K}

(2) ↳ Data Integrity

- when data is retrieved, must maintain original form
- while sending data, data should be received the way it was sent. If there is any modification - receiver should identify it.
- any change to data must be made by authorized user only.

Ex use checksum mech to check if data is intact.

M + checksum : can verify if data integrity is maintained or not.
(Message)

Ex Hash Algorithm.

hash (plain text) ⟶ output (hashdigest)

- Attach hashdigest with message (original data)
- one way hash algorithm : not able to get back original plain text from hashdigest i.e impractical to get back original digest

→ M+ checksum is stored. } for stored data
h(m) ⟶ digest

→ M+ checksum sent to recv. } for transmitted data
recv calculates checksum and verifies if it is the same as the sent message. } — make sure both use same hash algo.

Date: / /

③ ↳ Digital Signature - Technique to identify authenticity. If same user is sending different messages with diff content - diff signatures generated for same user.
- Message + signature - sent from one entity to the other.
- This is applicable for public key cryptosys (Asymmetric)
- Every message has unique signature even if all messages belong to same user/entity
- M + Signature sent to receiver
- Receiver does not verify integrity
- Signature used to identify whether the sign belongs to message or not.
- Receiver identifies if signature belongs to particular message.
- checks if message has come from a certain entity
- one key used for identifying signature
- ——— " ——— user.

④ ↳ Authentication Exchange
- Mutual Authentication - both parties verify authentication of both users.
- One side authentication - only client has to verify authentication to server.

Some parameters are exchanged during authentication - called Authentication Exchange Parameters.

⑤ ↳ Traffic Padding
Add padding bits to message i.e useless/bogus bits to confuse traffic analyzer - mitigate attacks due to traffic analysis.

⑥ ↳ Routing Control

If a big message has to be transferred, file can't be accommodated in single IP packet ∴ split in diff packets → same original packet ∴ same source & dest.

use source routing concept i.e sender specifies path to be taken by each packet.

If all n packets have same path — attack poss.

If diff packets are sent using different paths, not poss for attacker to attack all routes.

⑦ ↳ Notarization

Third party is involved to send packet/msg from one party to another.

To avoid repudiation, third party introduced so that no party can deny.

⑧ ↳ Access Control

Servic	Mech
Confidentiality	①, ⑥.
Integrity	①, ③, ②.
Authentication	①, ③, ④.
Non-Repudiation	②, ③, ⑦.
Access Control	⑧.

Defns :

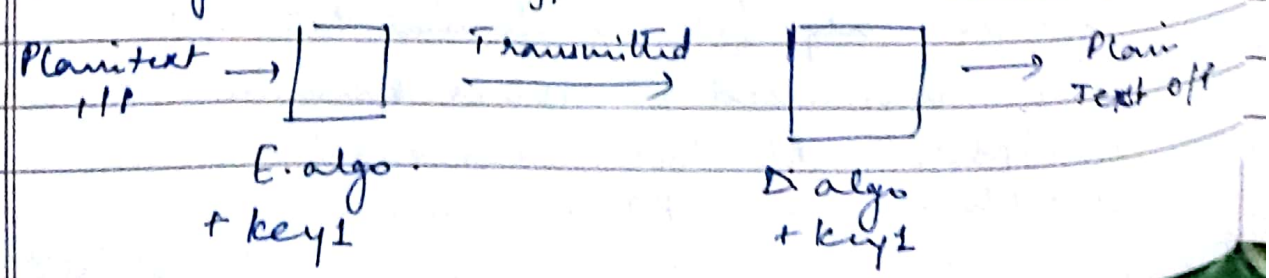
- Plaintext – original message / readable form
- Ciphertext – coded message (with E, K).
- cipher – algo for transforming P.T to C.T
- key – info used in cipher known only to sender, recv.
- encipher : PT → CT
- decipher CT → PT

- cryptography - study of encryption principle methods
- cryptanalysis (codebreaking) - procedure to derive PT from CT; study of principles of deciphering ciphertext without knowing key; finding what key is without C.T
- cryptology - field of cryptography + cryptanalysis

⇒ Cryptosystem
It is a five tuple (P, C, K, E, D) such that
↳ P :
↳ C :
↳ K :
↳ E :
↳ D :

⇒ Requests for secure use of Encryption Algo :
① • Strong encryption algo - opponent should be unable to decrypt C.T or discover key even if opponent has a no. of C.T + P.T that produce each C.T
② Secret key known only to sender & receiver - maintain secrecy.

⇒ Types :
① Symmetric Encryption
• conventional / private key / single key
• Sender & recv - common key.
• all classical encryption algos are symm.
• was the only type before 1970.
② Asymmetric Encryption



papergrid

Date: / /

• Same key used for encryption & decryption.

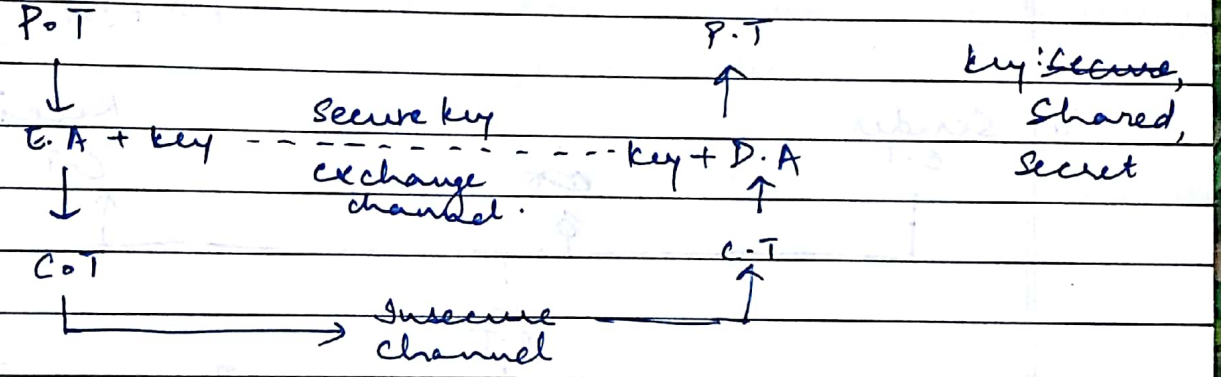
Note: All E. algos based on 2 principles:

i) substitution in which each element in P.T (bit/letter/group of bits or letters) is mapped into another element, and

ii) transposition: elements of P.T are rearranged. (no info should be lost)

(Most systems involve multiple stages of subs & transp.)

→ General Idea of Symm key C.S :-



Encryption: $C = E_K(P)$ Decryption: $P = D_K(C)$

$D_K(E_K(x)) = E_K(D_K(x)) = x$.

(Proof).

8/1/18

Block cipher: Encryp" & decryp" takes place in blocks.

Stream cipher: bit by bit.

⇒ Objective of attacking an encryption system is to recover key and not just derive C.T from P.T.

2 methods:

i) Cryptanalysis

ii) Brute force – try all keys available in domain.

Date: / /

Cryptanalysis

- Cryptography: Science & art of creating secret codes
- Cryptanalysis: Science & art of breaking those codes

Cryptanalysis attacks:- (To derive keys)

i) cipher text only - attacker only has set of C.T + algo details

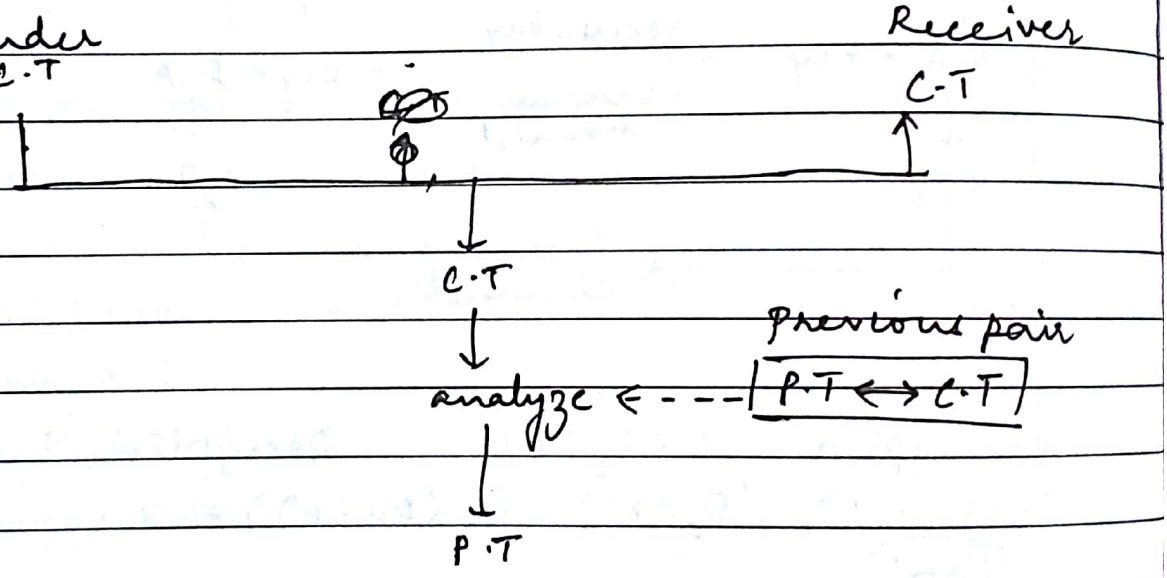
ii) plain text only - Attacker has P.T + corresponding C.T + algo details - can derive key.

iii) chosen plaintext - similar to (ii)

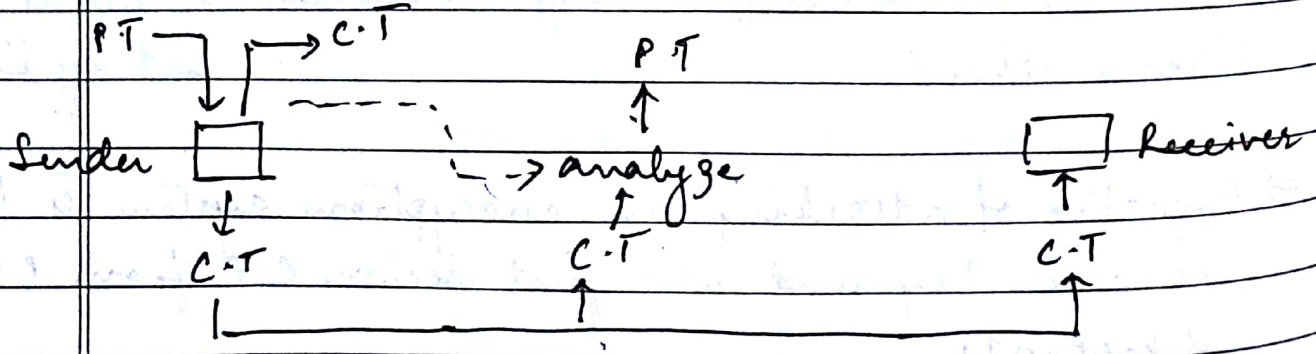
iv) chosen ciphertext - similar to (i).

attacker has to choose P.T + C.T

ii)



iii) attacker chooses some P.T, finds C.T for this I/P in some way (alias system)



Date: / /

iv) chosen cipher text analysis
↳ Pair created from chosen ciphertext

* Substitution cipher:
→ Each char of P.T substituted by another char.
→ Types:
i) Monoalphabetic } cyrum
ii) Polyalphabetic }

i) Relation bet" chars of P.T & C.T is one to one.
Ex: hello / blood ⟷ khoor (Since both 'l' subs by 'o')
ii) Relation bet" P.T char & C.T char is one to many.
⇒ autokey cipher. → polyalphabetic
↳ P.T char used as key
↳ to start, only one key value is used. then from P.T
↳ $P = P_1 P_2 P_3 \ldots$
$C = C_1 C_2 C_3 \ldots$
$k = k_1, P_1, P_2 \ldots$ (key value + chars of P.T)

⇒ Additive cipher: monoalphabetic, shift, caesar

@ Ex of autokey:
$C_i = (P_i + k_i) \% 26$ $P_i = (C_i - k_i) \bmod 26$.

P.T :	a	t	t	a	c	k
P value:	0	19	19	0	2	10
key stream:	12	0	19	19	0	2
C value :	12	19	12	19	2	12
C.T :	M	T	M	T	C	M

Papergrid
Date: / /

⇒ Vigenere Ciphers → POLY, stream cipher → encrypts char by char.
→ Divide P.T into a block of fixed size.
→ Block size depends upon key size
→ use key & block for encryption
Encryption : $C_i = P_i + K_i$
Decryption : $P_i = C_i - K_i$

Ex: She is listening → P.T
6 char keyword → PASCAL
∴ length of block = PASCAL = 6.

	s	h	e	i	s	l	i	s	t	e	n	i	n	g
P values:	18	7	4	8	18	11								
	P	A	S	C	A	L								
key stream	15	0	18	2	0	11								
Add → C.T values	7	7	22	10	18	22								
C.T →	H	H	W	K	S	W								

⇒ Hill Cipher
→ Block cipher
→ E.A takes m successive P.T letters and substitutes them for m cipher text letters.
Prerequisite → While choosing K, K inverse must exist (used for decryption) else decryption is not an inverse of encryption
→ uses matrix multiplication operation

key:

$$K = \begin{bmatrix} k_{11} & \cdots & k_{m1} \\ k_{21} & & k_{m2} \\ \vdots & & \\ k_{m1} & & \end{bmatrix}$$

Say, P.T domain size restricted to 26 characters.

C.T = K × P.T (multiplication of key and plaintext)

To check if K' exists:

K has an inverse iff det K is invertible in Z_{26} iff gcd (det K, 26) = 1.

$$K^{-1} = (\det K) \times \begin{pmatrix} k_{22} & -k_{12} \\ -k_{21} & k_{11} \end{pmatrix} \quad \text{where } K = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

$C_1 = (K_{11} P_1 + K_{12} P_2 + K_{13} P_3) \bmod 26.$

$C_2 = (K_{21} P_1 + K_{22} P_2 + K_{23} P_3) \bmod 26.$

⋮

$C = E(K, P) = KP \bmod 26$ (all matrices)

$P = D(K, C) = K^{-1} C \bmod 26 = K^{-1} KP = P.$

★ Transposition Cipher

→ characters moved around.

→ reorders symbols

→ does not substitute one char for the other.

→ Types:

i) keyless – no key used

ii) keyed – key used.

iii) combining (i), (ii)

⇒ Rail Fence Technique.

→ Arrange P.T in zigzag way.

→ keyless

Ex:

"meet me at the park"

m e m a t e a k
e t e t h p r

→ C.T: memateaketethpr.

4. decide a block size → say 4.

m e e t
m e a t → mmtaeehreaektp
t h e p
a r k

CT ↑

01

- Divide plain text into blocks
- Block size = key size.

use key value:

Say:

	3	1	4	5	2	↑ D
E ↓	1	2	3	4	5	

3rd char in PT move to 1st etc

a	t	t	a	c	k	i	s	t	o	n	i	g	h	t
t	a	a	c	t										

If size of last block < block size → add padding character

Combining key & keyless approaches :-

- write row by row
- ~~read col by col~~
- shift / shuffle columns using key i.e 3rd col goes to 1st etc
- read col by col.

e n e m y
a t t a c
k s t o n
i g h t z ↑ extra

shuffle

e e m y n
t a a c t
t k o n s
h i t z g

key used: 3 1 4 5 2
1 2 3 4 5

Date: / /

C.T : et-th eak i maotycmzntsg

To decrypt : Reverse process.

If Since domain size is very small — brute force attack very easy ∴ can derive algo + key easily.
∴ Not possible for real world applications.

- DES : Data encryption standard.
- Double DES : Apply DES twice
- Triple DES

* DES

→ Symmetric key cryptosystem
→ block cipher i.e encrypts fixed size blocks
block size decided.

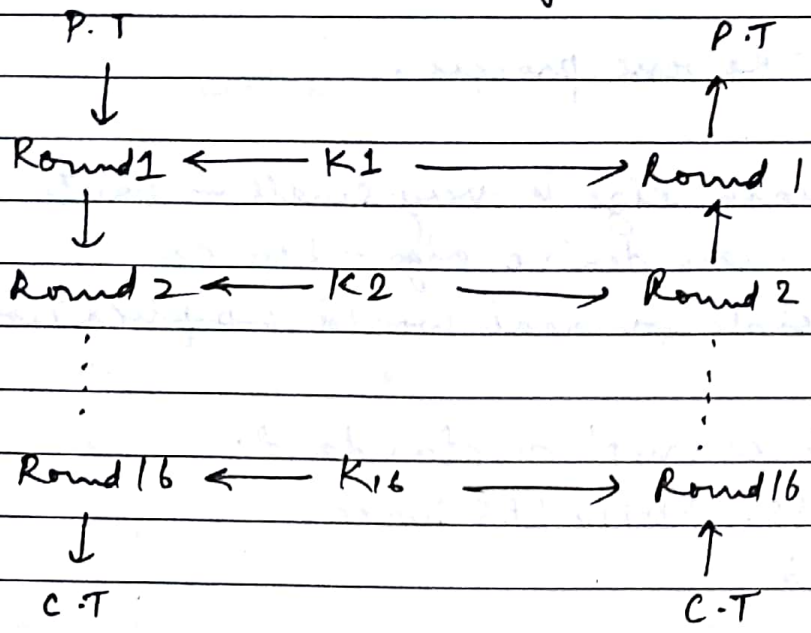
reduce brute force { → domain must be launched for key (key domain)
key domain : 2^{56}, key → 56 bits
P.T → 64 bits

→ P.T → many steps → C.T
all steps are same : performed numerous times
i.e same internal structure for all rounds

Ideal no = 16
→ More no. of rounds ⇒ Time to convert P.T to C.T is more. If no. of rounds ~~greater~~ less than 16 — possible to break in.

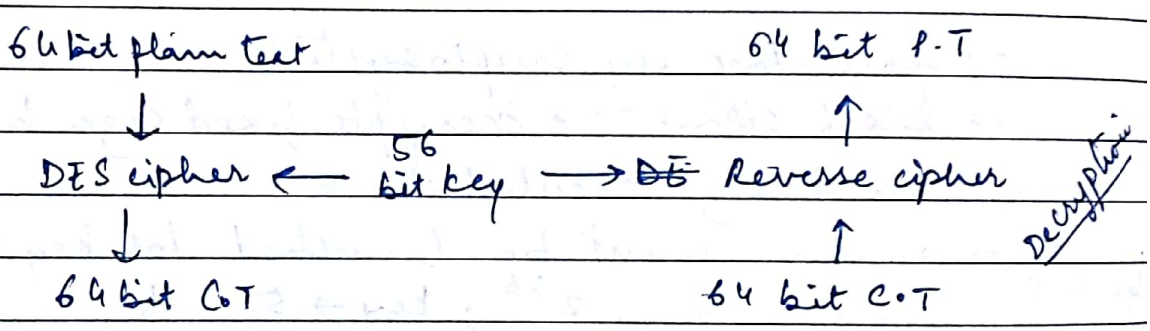
→ ∴ No. of rounds = 16 enough to secure data
→ ~~Different~~ O/P of 1st round → I/P of 2nd round.
O/P of 2nd round → I/P of 3rd round
→ In each round, key used is different
∴ Same procedure but diff I/P & key.
→ From 56 bit key, 48 bit round key derived. different for each round.

→ Cond^n: All round keys are different.



Internal structure of all rounds is same.

10/1/18

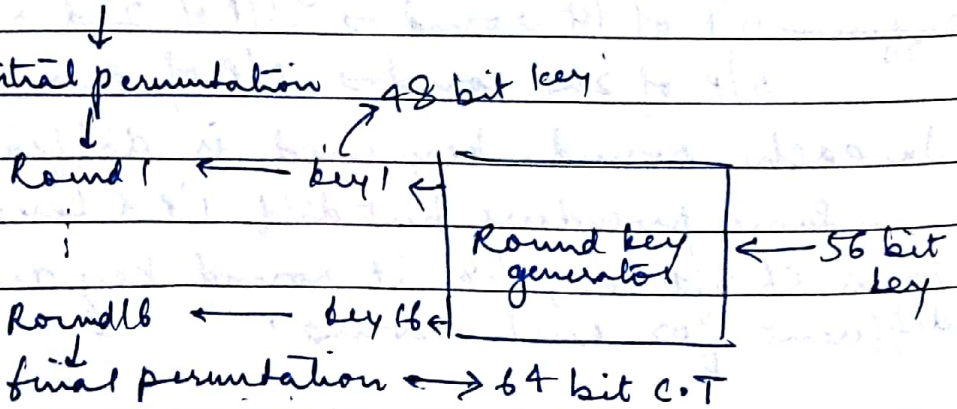


⇒ General structure of DES

↳ 2 permutations (P-boxes) called initial & final permutations - first bit/position change.

↳ final & initial permutation are inverse of each other.

↳ they do not add to strength of DES



→ Keys for each round are different - generated from 56 bit key - by round key generator, for each Round 48 bit key is generated from 56 bit key.

* Initial & final permutation steps

Initial: 58th bit → 1st bit (goes to)
1st bit → 58th bit
⋮

Final: 1st bit → 58th bit
58th bit → 1st bit

Reverse

Initial

Rows 1, 2, 3, 4 → even nos. Rows 5, 6, 7, 8 → odd nos.

Initial:

8		18	10 2
		20	12 4
		22	14 6
		24	16 8
		17	9 1
		19	11 3
		21	13 5
		23	15 7

Final: Alternate cols filled from bottom to top starting from col 2.

40	8	48	16	24
	4			
	3		11	
64	2		10	
33	1	41	9	17

Ex Find O/P of initial permutation box when I/P is given in hexadecimal as:

0x 0002 0000 0000 0001

Convert to binary → 64 bit in a bit stream only 2 bits are 1 & 15th & 64th bit.

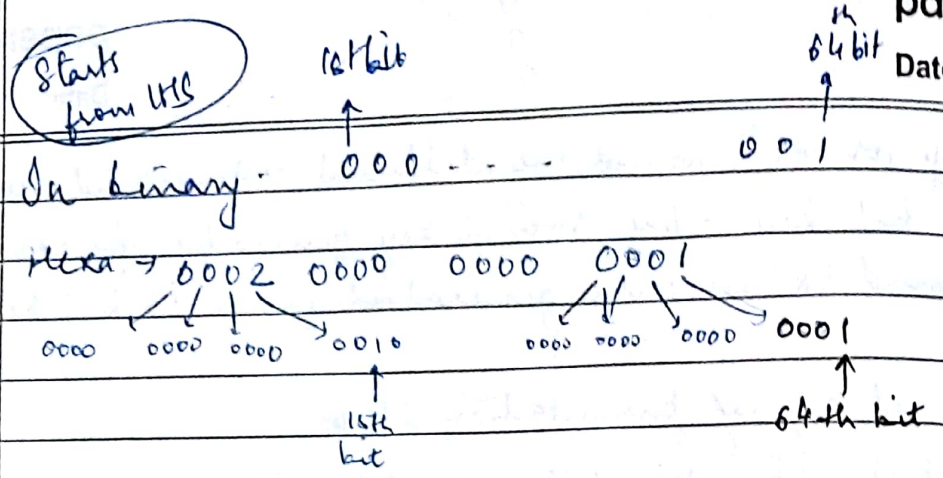
⇒ 64th bit in P.T becomes 25th bit after permutation in the 25th place.

Since in the table it is

& 15th bit becomes 63rd bit since it is at position 63 in matrix.

papergrid
Date: / /

Starts from LHS



Now check table : 15th bit becomes 63rd
64th bit 25th

* ~~Also~~ & In each round - algo same, key & I/P changes for each round.

In each round →
64 bits divided into 2 parts : 32 bit size of each part
↳ left part & right part
Left block involved in exclusive operation
Right " ~~possessed~~ processed with ~~sub~~ round key
O/P of right block now is processed with left block
Then swap left block with right block.
Now in next round, left block is processed with new round key.

Expansion

Key is 48 bits and right block is 32 bits → can't exor
∴ Expansion is done, then exor.
Again there is exclusive operation with left block
This time 48 bit key is mapped to 32 bit so that it can exclusively by or-ed

To compress 48 bit → 32 bit : use S-box. i.e a set of S boxes. Each S-box takes 6 bit I/P and produces 4 bit outputs
∴ 8 S-boxes required.
48 bit key divided into chunks of 6 bits.
Each chunk goes to S-box.

{ Right + key
{ Right + Left
Swap

papergrid
Date: / /

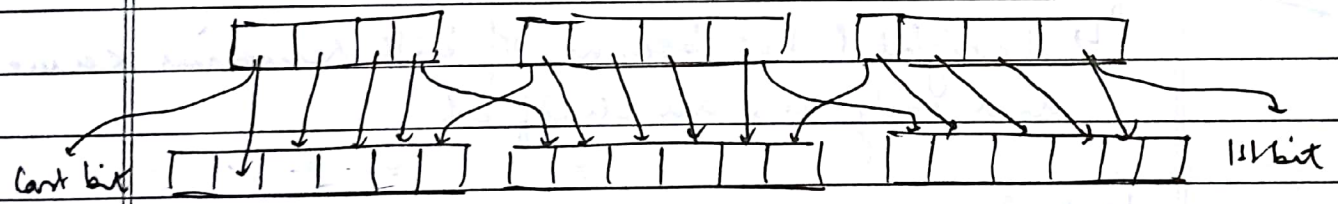
How does S-box work?
Each S-box maintains its own table & converts
6 bit → 4 bit .
Then it goes to permutation box - no compression
or expansion .
→ Concentration is our right block.

Expansion: How 32 bits → 48 bits?
Arrange 32 bits in blocks of 4 bits.

Expansion Permutation Box'

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20.	21

Each block of 4 bits expanded into 6 bits.



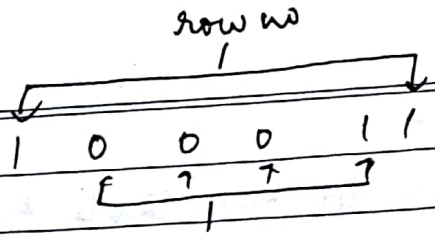
S box : Compression 6 bit → 4 bit

To identify any row in table → 2 bits reqd
col 4 bits

- Each S-box maintains its own table.
- Takes 6 bit I/P, 1st bit & last bit represents row no.
  Remaining 4 bits represent col no.
- In every row, values range from 0 to 15

Ex: S box I/P is 100011

Date: / /



11 → row no. = 3

0001 → col no = 1

Intersection of row & col → o/p of s-box

s-box table starts from 0.

	0	1	2	3	... 15
00					
01					
10					
11		O			

12 → o/p i.e 1100

For 011011, row → 01 = 1
col → 1101 = 13
} 5 (from table).

And 5 : 0100

11/1/18 After S box : P-box, permutation box,

↳ straight P-box bec. no. of bits remains same. and only bits are shuffled.

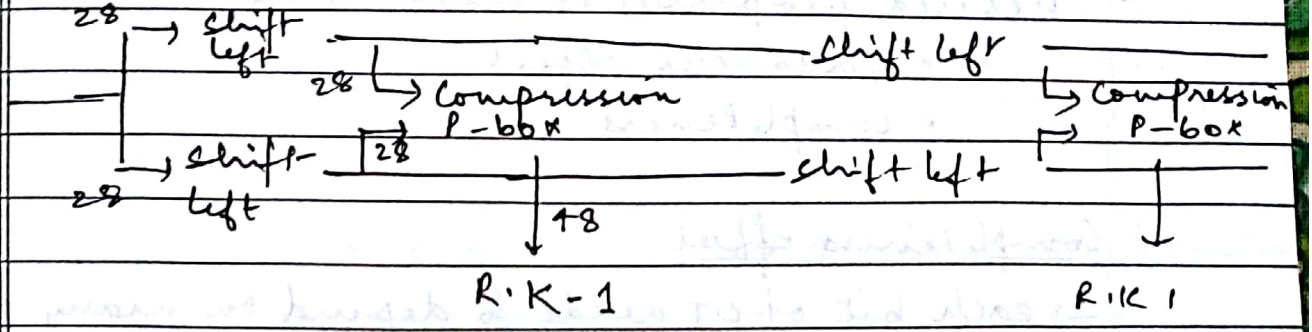
o/p of final permutation is cipher text.

Decryption operation:

- starts from Round 16 - key 16 used first i.e outer encryption removed first
- initial & final permutation cancel each other
- expansion P box and straight P-box ~~work~~ - same for all rounds.

* The 56 bit key (original) → 48 bit key (new)
flow ?

Divide 56 bit key into 2 parts of size 28 bits each.
In each round perform ~~a~~ left shift, ~~right~~ left shift on both parts
↳ compression P-box which generates 48 bit



Shifting :- In Round 1, 2, 9, 16 → bits are shifted by one position
other rounds → bits shifted by 2 positions.

After R1, Till round 8 → Total no. of bits shifted is even
After → odd
Round 16 → 28 bits totally shifted.

- Before ~~the~~ round starts, key goes through permutation choice table
- Permutation choice 1 :
- Refer initial permutation table to calculate PC-1

I/P: ~~58~~ ~~50~~ ~~42~~ ~~34~~ ~~26~~ ~~18~~

57 \ 49

Subtract i from i^{th} row of I/P
Table is 8×7
Only 1st 4 elements of 4th row of I/P are used
Continue with 5th row. ~~to~~ Add 6 to all elements.

Date: / /

★ Avalanche effect

- If a single bit changes in I/P or key, there is a significant change in C.T.
- Approx half bits change in C.T.
- Desired properties of block cipher:
  - avalanche effect
  - completeness

★ Completeness effect

→ each bit of C.T needs to depend on many bits of P.T.

→ The diffusion & confusion produced by P-boxes & S-boxes in DES shows completeness

⇒ Design criteria of S-box:-

- entries of each row are permutations of values bet" 0 to 15.
- single bit change in I/P of S-box will change two bits in O/P of S-box.

⇒ Design criteria of P-boxes:- Straight Expansion

B. 4 O/Ps from each S-box go to 6 diff. S-boxes...

56 bit key derived by removing parity bit from 64 bit key.

papergrid
Date: / /

⇒ Weakness in keys

Domain of keys = 2^{56}, some keys should not be used.

i) Weak keys - set of keys that produce same ~~similar~~ round key in all 16 rounds.

Ex: 00000000 00000000 (keys divided into 2 halfs).
00000000 FFFFFFFF

⋮

Note: O/P of 1st round of decryption = O/P of 15th round of encryption

If weak keys are used, O/P after every round of decryption = P.T.

(4 weak keys)

ii) Semi weak keys. - Creates 2 diffrent round keys are generated from 56 bit key and each of them is repeated 8 times (in a random order)

- first & second key in pair - RKs generated by both are the same, but order is ~~diffrent~~ may not be same.

Total no. of semi weak keys = (12)

iii) Possible week keys - (48) such keys.

- produces 4 diffrent round keys.
- 16 RKs are divided into 4 groups & each group is made of 4 same RKs.

Ex: Assume RK1 & RK2. What is the prob. of selecting a weak, semi-weak or poss weak keys ~~if 56 bit key is semi weak.~~

Soln:

RKS

$$\frac{4+12+48}{2^{56}}$$

RKS

Date: / /

Ex 000...00 00...0 — one of the possible weak keys.
28 bits 28 bits

17/1/18 56 bits key | Brute force attack | : Try one key at a time from key domain.
Attacker only has cipher text.
He has to find out P, K.

2^{56} possibilities for K: [K_1 K_2 ⋮ ⋮ ⋮] key → 56 bits.

Attacker tries to decrypt with K_1. If P.T does not make sense: drop K_1 and take K_2

$D_{K_1}[C_1] = P_1$ → meaningless

then $D_{K_2}[C_1] = P_2$

2^{56} Time ∝ = no. of attempts (max)
→ Depends on the no. of systems used, processing speed etc.
→ If a network system is used, DES could be broken
∴ Single stage / key DES is not secure.

Later it was proved, one need not try 2^{56} times.
They can try 2^{55} attempts (half)
This is done by using:
key complement concept.

* Key Complement Property of DES

Qs Given a bitstream x. Prove that

$DES_{\bar{K}}(\bar{x}) = \overline{DES_K(x)}$

$\bar{x}$ → denotes complement of P.T
$\bar{K}$ — key

Date: / /

⇒ Key complement property :-
In the key domain, definitely half the keys are complement of the other half.

$$C = E(K, P) \rightarrow \bar{C} = E(\bar{K}, \bar{P})$$

Proof :-

Properties of exor : ① $\bar{x} \oplus \bar{y} = x\bar{y} + \bar{x}y$
$= x \oplus y$

② $\overline{x \oplus y} = \bar{x}\bar{y} + xy = \bar{x} \oplus y$

Assume P.T is complemented & key is complemented. This complemented P.T is divided into two halves → L & R

Complemented R (exor) Complemented Key
= R exor key (same o/p - from prop ①)

Then complement of L is exor with o/p of R exor key.
from prop ② → o/p of this 2nd exor is complement of left ⊕ (right ⊕ key).

C_L - C.T of left block
C_R - C.T of right block

For 2nd exor $\{C_L, C_R = P_R, P_L \oplus F[P_R, K]$ Since swap takes place.
=

↑ I/P to function

Same steps apply for complement.

P_R, P_L complemented
K complemented

papergrid

from prop (1) Date: / /

$C_L, C_R = \overline{P_R}, \overline{P_L} \oplus F[P_R, K]$ — Same o/p as previous

$= \overline{P_R}, \overline{P_L \oplus F[P_R, K]}$

Same for all 16 rounds

Concl: If key is complemented & P.T is complemented, o/p is the same as if it is not complemented.

∴ One need not try 2^{56} combinations.
Just try 2^{55} for deriving key.

☆ **Double DES**

- Enhancement for DES.
- 2 DES
- Same rounds. (internal comp. of DES are same)
- encrypt given P.T 2 times
- o/p of 1st encryption, encrypt further - once more
- final C.T
- If key K_1 is used in both encryptions, there are chances that it may get cancelled.
- After 2nd DES — C.T
- If same key used in 1st ~~to~~ stage of DES and 2nd stage of DES — chances of getting cancelled.
  ∴ use different keys in 1st & 2nd stage. (K_1) (K_2)
- At time of decryption, k_2 used first, then K_1 is used.
- Meet in the Middle Attack :- (Known PT attack)

64 bit PT
↓
DES cipher
↓
64 bit IMP middle text
↓
DES cipher
↓
64 bit C.T

$k_1, k_2 \rightarrow$ 56 bits each ∴ thought complexity is 2^{112} (brute force)
However, the complexity ↑ only till 2^{57}
This was proved later

Date: / /

- For known P.T attack, attacker can break double DES in 2^{57} attacks.
- i.e attacker has P.T & corresponding C.T
- Attacker takes P.T and a key from key domain - Stores all poss C.Ts. middle texts after encryption.
- C.T is taken and decrypts it with key from key domain and stores decryption o/p in another table. (middle text)
- Middle text of encryption & decryption should be same
- ∴ Max encryptions for Middle text $= 2^{56}$
  decryptions $= 2^{56}$
- ∴ Total $= 2 \times 2^{56} = 2^{57}$
- Can derive K1 & K2 - both

$M_1 = E_{K_1}(P)$ $\quad M_2 = D_{K_2}(C)$ known PT & CT

- If $M_1 = M_2 \rightarrow K_1, K_2$ ~~same~~ found.
- sort tables in which M is stored. If they match, corresponding K1, K2 found.

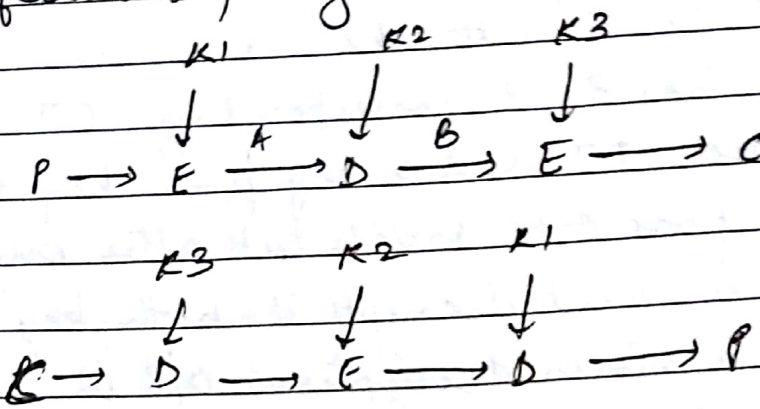
* Triple DES

- no modification in internal structure of DES
- no. of stages only varies
- 3 stages are: encryption - decryption - encryption
  (for encryption: K1, K2, K3)
  for decryption: decryption - encryption - decryption (K3, K2, K1)
- $C = E_{K_3}[D_{K_2}[E_{K_1}[P]]]$

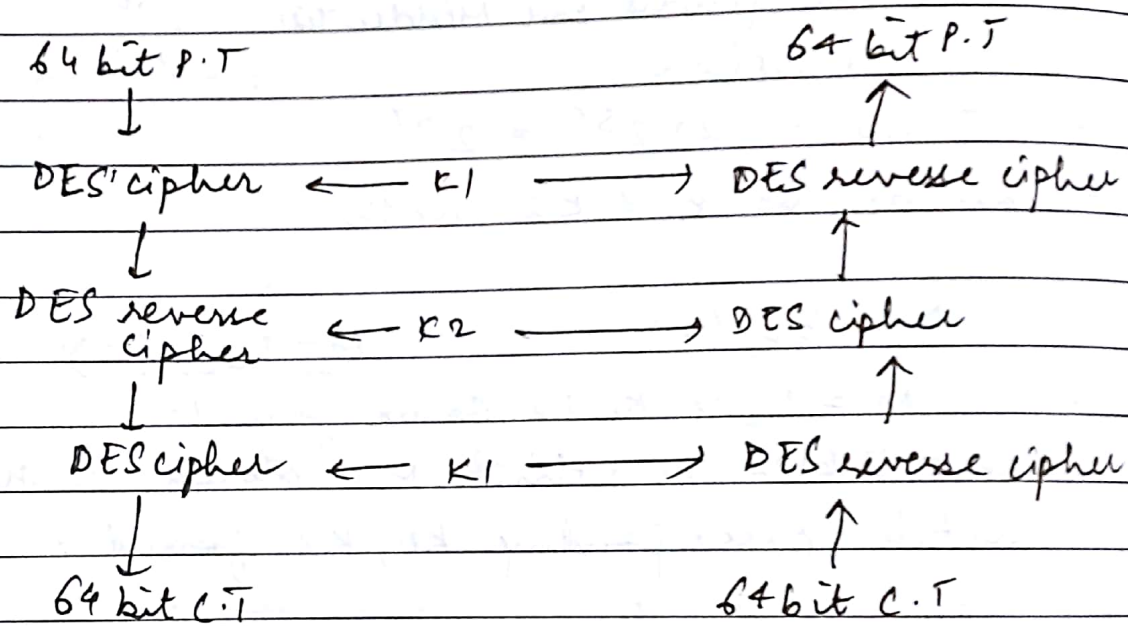
i) Triple DES with 2 keys
ii) ——"—— 3 ——"——

- effective key length = 168 bits

Triple DES encryption with 3 keys.



⇒ Triple DES with 2 keys :-



If K1 = K2 ⇒ Single DES bec. first 2 stages i.e encryption and decryption cancel out.
∴ K1 ≠ K2

Qs. How can an adversary take advantage of complementation property of DES for 3 DES with 2 keys ?

Soln $DES_k[\bar{X}] = \bar{C} = \overline{DES_k[X]}$

For triple DES with 2 keys:

$DES_{K1,K2}[\bar{X}] = \bar{C}$ → Prove. How?

$DES_{\overline{K1}}[\bar{X}] = \overline{DES_{K1}[X]}$ → o/p of 1st stage of encryption
↳ used in end stage I/P.

for decryption.

$DES^{-1}_{\overline{K2}}\left[\overline{DES_{K1}[X]}\right] = \overline{DES_{K2,K1}[X]}$
↳ o/p of 2nd round.

$DES_{\overline{K1}}\left[\overline{DES_{K2,K1}[X]}\right]$

$= \overline{DES_{K1,K2,K1}[X]}$

With this, how many attempts are needed to break the system? 2^{110} ~~$2^{56} \times 3$~~

K1 · K2 · K1
↑ ↑ ↑
255 255 (no need to try complements)

22/01/18 (Problems)

Design criterias applicable for all S-boxes.

4th design criterias for S-box :- Take 2 I/Ps. In these 2 I/Ps only 3rd & 4th bit should differ (1st, 2nd, 5th, 6th bit same). For these 2 I/Ps, if o/p differs ~~only~~ atleast by 2 bits → design criteria 4.

Check 4th design criterion for S-box 2 using following pair of I/Ps :-

a) 001100 & 110000
↓ ↓ (middle 2 bits diff)
000000 111100

row → 00
col → 0110.

6th design criterion for S-box 5:-

32 ~~keys~~, 6-bit I/P pairs

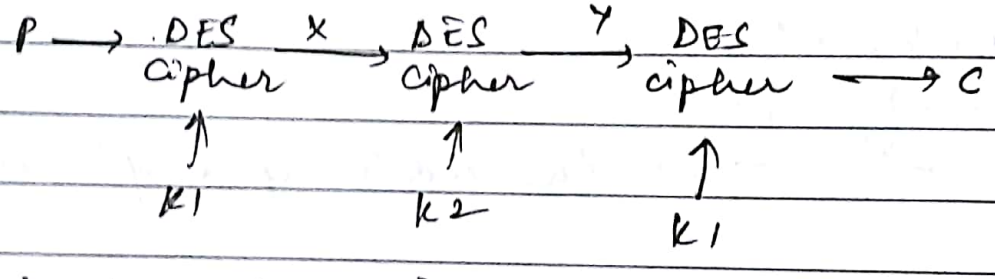
32, 6-bit I/P word pairs such that $x_i \oplus x_j \neq (000000)$

These 32 I/P pairs create 32 4-bit O/P word pairs

(For each pair compute O/P of S-box) ~~+f o~~

$d = y_i \oplus y_j$ → for all 32 pairs

Qs Devise a meet in middle attack for triple DES. with 2 keys.



Is it poss to derive K1, K2, K1 in 3DES?

Soln In 2 DES we need to know PT & CT for MIM attack.

Even in 3DES assume attacker has some pairs of PT & CT

For MIM attack with 3DES

X → guess

- find Y (first)

Then MIM for 2DES (remaining 2 DES ciphers)

Take random 64 bit-guess it is X

key domain → 2^{56}

Decrypt X with different keys.

* Cipher Block Modes

- diff ways to transmit data
- assume all blocks are 64 bits i.e message is multiple of 64.
- All blocks are encrypted with same key k.
- gives 'n' C.T blocks.
- If attacker can decrypt one block, he can decrypt all blocks.
- C.T depends on something else (beside key) which is diff each time.

Suppose message divided into 5 blocks — P1 to P5.

$E_k[P1] \rightarrow C1$, $E_K[P2] \rightarrow C2$, $E_K[P3] \rightarrow C3$.

$E_k[P4] \rightarrow C4$, $E_K[P5] \rightarrow C5$

Since same key is used for encryption, if 2 P.T blocks are same corresponding C.T blocks will be the same.

∴ The attacker only needs key, then he can decrypt rest of the blocks.

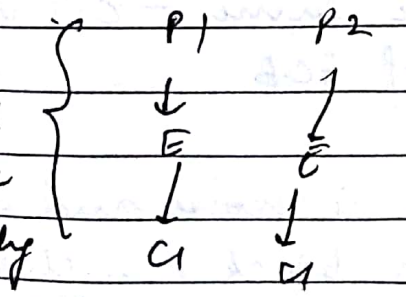
Modes of Operations categorized into:

① ECB - Electronic Code Book :- similar to telephone directory

Divide msg into blocks of 64 bits. — encrypt each block — transfer

Adv :- Parallel encryption (i) DES.

All $P_i \rightarrow C_i$ can be done parallely



(ii) one round key generation is enough.

(iii) If noise is there i.e bit inversion takes place in

~~decodes~~ one block. The effect will be there for only block - rest of them are unaffected

Disadv :- (i) If attacker desires key, the attacker can decrypt rest of the blocks.

(ii) 2 blocks of P.T same ∴ 2 blocks of C.T same

To overcome this → CBC

(2) **CBC** - Cipher Block Chaining :-

- One block attached to another.
- Chaining is applied to C.T.
- TO process block P2, o/p of P1 is needed and so on
- Dependant on C.T of previous block
- To encrypt first block, initialization vector (64 bit) ~~is~~ is used. (IV) → must be a secret like key.
- Take XOR with IV in round 1, then apply DES.
- ~~o/p~~ CT of round (i-1) is exored in round i with P.T i.

Encryption : $C_0 = I.V$

$C_i = E(K, P_i \oplus C_{i-1})$

Decryption: $P_1 = D(K, C_0) \oplus I.V$

$C_i = D(K, C_i) \oplus C_{i-1}$

- Same key used for encryption
- even if blocks of P.T are same - C.T is different
- Overcomes drawbacks of ECB.

Disadvantages :- (i) If there is noise and some bits are interchanged in block 2 - this affects ~~rest of~~ the blocks after that → only ~~one~~ two blocks are affected

Date: / /

i.e offsets that block & block after that

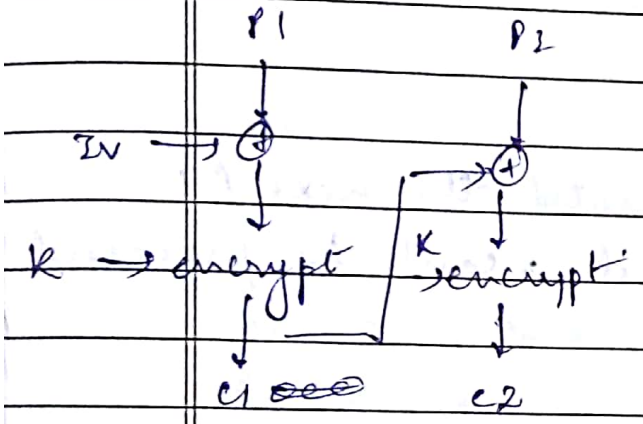
24/01/18 Prob 1 :- 64 octets of message - use DES in CBC mode - how many octets are sent now?

Soln : 64 octets = 64 × 8 bits.

Total data is divided into blocks of 64 bits

∴ No. of blocks = 8 → 8 blocks of C.T after encryption

IV also needs to be transferred.



IV is 64 bits for each block

⇒ 8 octets for P.T

∴ Total bits = 64 + 8 octets

= 72 octets.

Prob 2 : N 64 bit blocks

a) No. of DES encryption operations = N * 16

as DES must be performed on N blocks 16 times (16 rounds).

16 round keys for processing one block. Same round keys for processing ~~one~~ blocks.

⇒ 16 round keys to process one block and there are N blocks.

1 block → 16 rounds P_1 . . . P_N

↓ ↓

C_1 C_N

(16 * N)? refer ppt

Prob 3 : 32 octets = 32 × 8 bits

MAC - Message Authentication Code.

For integrity we add ICV.

32 octets ~~+ IC~~ D + checksum. (ICV).

data (ICV : integrity checksum value)

Date: / /

32 octets + MAC/ICV generated using CBC DES.
data

Data = 32 × 8 bits = 256 bits

No. of blocks = 256/64 ∴ 4 blocks

Each block encrypted using CBC DES

Blocks C1 C2 C3 C4 + MAC
(32 octets; last 64 bits)

MAC = 8 octets (last C.T block)
IV = 64 bits = 8 octets.

Drawbacks of CBC:

i) Till C.T of a block is generated, the next P.T block can't be processed ∴ can't be processed parallely.

⇓

New Modification: CFB

③ Cipher Feedback Mode

- IV used as shift register — no circular shift, just shift
- stream cipher i.e used to encrypt bytes of data, not blocks
- O/P of 1st encryption = (IV using key value)
- 64 bits in IV shifted by 's' no. of bits left, empty space occupied by O/P of previous block
- P.T is bit or byte
- A byte of P.T is exor-ed with byte O/P of encryption
- O/P of a round goes to next round.
- select most significant n-bits
- In CBC and CFB: even if P.T blocks are the same — C.T will be different (as O/P of previous is used & shift reg contents are different)

Drawback: If some bits are inverted during transmission in C.T, how many blocks are affected?

Suppose in shift reg. 8 bits are shifted and C.T is put in place of shifts in IV, how long does C1 stay in IV? — All those blocks are corrupted?

↳ 8 blocks

∴ 1 + 8 = 9 blocks affected.

Prob!: C.T bit corrupted — case 1
P.T bit corrupted — case 2

EBC - only that block is affected } case 1 - CT affected
CBC - affects 2 blocks

If P.T gets corrupted in CBC — All blocks are affected at receivers side.

For CFB, if shift by 8 bits — 9 blocks affected
2 bits — 33 blocks affected
$(\frac{64}{2} \rightarrow IV = 32 + 1)$ ↑ original

Drawback: Parallel not possible
⇓
Solution: OFB.

④ Output Feedback Mode

- output of encryption is taken (not C.T) and uses it in next shift register.
- OFB → O/P after encryption is taken in next block.

CBC, CFB → C.T used.
- stream cipher.

papergrid
Date: / /

- o/p of 1st encryption used as feedback for next encryption
- no dependancy.

(5) Counter Mode
→ IV is incremented by 1 or n from block to block automatically without depending on previous o/p
→ counter incremented for each block.
→ keep incrementing IV - no shift.

25/01/18

ASYMMETRIC KEY ~~CRYPTO~~ CRYPTOSYSTEM (PUBLIC)
(2 KEY CRYPTOSYSTEM)

(1) → In symmetric, both sender & receiver must have key - a major problem is distribution of key bet' sender & receiver.

(2) → In symm, not poss to generate digital signature or verify that message has reached intended user.
→ 2 keys used :- public key - encrypt
private " - decrypt
→ For digital signature :-
private key - to generate signature
public " " verify "

Ex RSA - used in many real world applications

→ ~~sender~~ receiver generate pair of keys - public & private
→ Recv. announces the public key i.e anyone can obtain it. (private key not shared)
→ Sender knows public key and encrypts data using this key.

Date: / /

→ If someone intercepts the message, they won't be able to decrypt.

→ Asymmetric – because diff keys are used for encryption & decryption

⊙ public key – encrypt & verify
private – decrypt & verify generati

→ sender can encrypt data, can not decrypt it.

→ only receiver can decrypt it.

→ anyone can verify signatures (since key is public)

Note: Two Algos depend on 2 keys :-

(i) → receiver's public key & private key are related to each other.

(ii) → It should be computationally feasible to perform encryption ~~key~~ / decryption by knowing public / private key

(iii)

~~Point~~ Limitations of symm → ① distribution of key
② signatures

3 categories of asymm:

1) Encryption / decryption
2) Key distn
3) Signatures

More Properties :-

→ Trapdoor secret

→ one way function – function such that ft
If $y = f(x)$, and if y is known, it is not possible to get back x i.e f^{-1} is not feasible.
i.e easy to compute $f(x)$ but not $f^{-1}(y)$
you can get x back from y by knowing some secrets i.e trap door secrets.

→ Secrets used to get back original message → trapdoor secret

papergrid

Date: / /

factorization prob. Eg: $n = p \times q$ (p, q: prime nos.) Easy to compute n. Not for p, q (esp. if they are big)

discrete logarithm problem Ex: $y \overset{\uparrow}{\equiv} x^k \bmod n$. → easy to compute y.
(≡: congruent to)

If we know trapdoor k' such that $k \times k' \equiv 1 \bmod \phi(n)$ then we can use $x = y^{k'} \bmod n$ to find x

RSA CRYPTOSYSTEM

- best for one way trapdoor functions
- DES takes characters and produces characters
- RSA takes integers integers
- public & private key required

Step1: Generating pair of keys

i) generate 2 prime nos. - p, q. (p, q - must be large)
   To check if generated integer is prime no. or not
   a) check divisibility by 2
   b) if not divisible by 2, use primality test algo (?)

ii) $N = p \times q$ (4 bit × 4 bit) ⇒ N is greater than 4 bit
   (512 512 ⇒ " " 512 "
   N must be large. When value of N is more, it leads to factorization problem. (time taken more)

iii) ~~Choose e and d~~ Compute $\phi(N)$ → Euler's quotient
   Euler's quotient: Take prime no, p. count from 1 to p.
   Find if each of those integers & p are co-prime or not

Ex) p = 7. Find GCD(1,7) → 1 ∴ 1, 7 are coprime
   GCD(2,7) → 1 2, 7 are coprime
   GCD(3,7) → 1
   :
   GCD(6,7) → 1

$\phi(N)$ ⇒ +ve integer lesser than N and co-prime to N.

papergrid

Date: / /

∴ for 1 to p we get p-1 co-primes.

Eulers constant: No. of co-primes is p-1 if p is prime and if N is non-prime

If p is prime no, $\phi(p) = p-1$

$\phi(N) = (p-1) \times (q-1)$ = ~~p.q~~ (∵ p, q are prime ∴ ~~odd~~ even)

↑ even ↑ even

[How many co-primes exist from 1 to N?]

iv) Generate pair of keys - public & private.

- generate an integer no (e) & verify if inverse (multiplicative) exists or not wrt $\phi(N)$

$e \cdot d \cong 1 \bmod \phi(N)$ ⇒ inverse exists.

(i.e $e \cdot d \bmod n = 1$)

∴ Now:

(e, N) → public key

(d, N) → private key.

29/1/18 Derive e & d → choose any 1 value ~~such~~ < $\phi(n)$

$e \cdot d \cong 1 \bmod \phi(N)$

If this cond^n satisfies, e is multiplicative inverse of d and vice versa.

(e, N) - public (d, N) - private

v) Announce public key. It is advised to destroy p, q, $\phi(n)$.

Public key announced as (e, n). Private key retained by entity.

Receiver generates signature and sends it.

vi) Encryption: Convert P.T to integers and

$C = P^e \bmod n$.

vii) Decryption: $P = C^d \bmod n$.

Ex $p=7, q=11$

$\therefore N = 77 \quad \phi(77) = (7-1)(11-1) = 60.$

Choose e, d such that

$e.d \bmod \phi(N) = 1.$

$e \rightarrow 13$ Then $d = 37.$

If $P.T = 5, \quad 13.37 \bmod 60 = 1.$

↳ encrypted : $5^{13} \bmod 77 = 26.$

$C.T = 26$

↳ decrypted : $26^{37} \bmod 77 = 5 \leftarrow P.T$

$\phi(N) = 60, \phi(N)^* \rightarrow Z_{60}^*.$

~~Problem~~

Why should e & d be less than $\phi(N)$. $M < N$

~~Suppose e =~~

If Message to be sent is less than N,

Ex : $M = 128 \qquad C = 128^{13} \bmod 77$

But $P \neq 128.$

∴ If you want to transmit something greater than N, divide M into blocks.

| 12 | 8 |

↑ ↑

$12 < 77 \quad 8 < 77$ ✓

→ Encrypt 12 and 8 separately.

Encry - $12^{13} \bmod 77 = 12 \qquad 8^{13} \bmod 77 = 50.$

Decry - $12^{37} \bmod 77 = 12 \qquad 50^{37} \bmod 77 = 8.$

Note : For alphabets - integer values are assigned bet^n 00 and 25. (two digit)

Ex : NO → N = 13

O = 14 ∴ P.T = 1314 ← concatenated

$1 < e < \phi(n)$.

papergrid

Date: / /

Qs If $e = 31$, $n = 3599$, $d = ?$

$31 * d \bmod 3599 = 1$

Extended ~~Extracted~~ Euclidean Algo provides inverse of no. given e, N?

↳ If $\gcd(\phi(n), e) = 1$, then inverse of e (i.e. d) exist.

else inverse does not exist.

If $\gcd(a,b) = 1$

and if we find 2 nos. such that $s \times a + t \times b = 1$

Then t is inverse of b.

$r_1 \leftarrow \phi(n)$ (a)

$r_2 \leftarrow e$ (b).

$s1 = 1 \quad s2 = 0$

$t1 = 0 \quad t2 = 1$.

```
while(r2 > 0)
{
  q ← r1/r2;
  r ← r1 - q×r2
  r1 ← r2
  r2 ← r

  s ← s1 - q×s2;
  s1 ← s2
  s2 ← s

  t ← t1 - q×t2
  t1 ← t2
  t2 ← t
}
gcd(a,b) ← r1
s ← s1
t ← t1.
```

After this, compute :-

$s \times a + t \times b = \gcd(a,b)$.

If this gives $\gcd(a,b) = 1$

then inverse of b exists

wrt a - t is inverse of b

If it is not 1, inverse of

b wrt a does not exist.

Q3 In RSA, public key of given user is $e = 31$, $n = 3599$. Find private key of user. $d = ?$

Ans Step 1: Compute $\phi(n)$ by $n = p \times q$.

~~Facto~~ Step 2: Factorize n into p, q.

$p = 59$, $q = 61$ } 59×61

$\therefore \phi(n) = ~~60 \times 5~~ \; 58 \times 60$

$= 3480$.

$ed \equiv 1 \mod \phi(n)$ $\quad\quad 31 \cdot d \equiv 1 \mod 3480$.

⇓

$\boxed{d = 3031}$

gcd of 3031, 3480 = 1

* Attacks on RSA :-

① Factorization
- n small, attack possible. If n is large, diff to factor
- message domain is small.

② Chosen C.T attack
- objective is not to get 'd', but to get M from C.T.
- It is assumed that sender sends message to receiver and receiver sends an ack message back
- Suppose attacker intercepts and gets C.T for which publickey is (e, N)
- goal is to derive M from C
- Attacker takes random no. r : $C \cdot r^e \mod N$. E is found. ↳ inverse shd exist (↑ some variable)
- send this to receiver, decrypts using (d, N)
- Bob decrypts msg :

$$[C \cdot r^e]^d \mod N = C^d \cdot r^{ed} \mod N.$$

Actual message: $M^e \bmod N$

papergrid

Date: / /

$= M \cdot x \bmod N$. $C^d = M \cdot$ ed → inverse.

Receiver sends back this $M \times x \bmod N$

Attacker intercepts again and uses inverse:

$M \times x \times x^{-1} \bmod N = M \bmod N$.

Attacker gets message M. ✓

③ Cyclic Attack / → Plaintext Attack

Cyclic

- Given e, N, C it is easy to derive M. (C, e, N) → M
- Keep encrypting 'C' again & again to derive C again.
- $c = M^e \bmod N$.

$C_1 = C^e \bmod N$ If $C1 = C$, stop · else:

$C_2 = C_1^e \bmod N$ If $C2 = C$ stop else:

⋮

If $C_n = C$ it means C_{n-1} was M.

- Proved that this will happen.

Qs In public key cryptography, to intercept P.T, C = 10 sent to a user whose public key is e = 5, N = 35 what is the P.T?

Soln $C1 = 10^5 \bmod 35 = 5$ $\therefore M = 5$

$C2 = 5^5 \bmod 35$

$= 10 = C$.

31/1/18 Qs Launch/demonstrate a chosen C.T attack by making use of given data value of n = 33 and e = 7, C = 14.

Soln $33 = 3 \times 11$ $p = 3, q = 11$.

$\phi(n) = 2 \times 10$

$= 20$.

$7 \cdot d \bmod 20 = 1$. ⇒ $d = 3$

Date: / /

Choose r such that r^{-1} exists wrt mod N.
If r = 2, then $r^{-1} = 17$.
$(2 \cdot d \bmod 32 = 1)$.

Attacker sends $C \cdot r^e \bmod n$
$= 14 \cdot 2^7 \cdot \bmod 33 = 1792 \bmod 33$
$= (10)$.

Decryption takes place :- ~~$(14 \cdot 2^7)^3 \bmod 33$.~~

~~$(14^3) \cdot (2^{\cancel{8}} = 21) \bmod 33$~~ $10^3 \bmod 33$
↑ M. $= (10)$.
↓
Sent back.
~~This is M.~~

Multiply with r^{-1}
$: 10 \cdot r^{-1} \bmod 33$
$= 10 \times 17 \bmod 33$
$= (5)$ M

To verify $\rightarrow C = 5^7 \bmod 33 = (14) = C$ ∴ verified.

Attacks continued :-

④ Blind Signature Attack

M + Signature
↑
to be computed - uses private key.

Signature = $M^d \bmod N$. where (d, N) is private key
(Encrypt message using private key of sender).

S R

M + Signature
|
↑
M + Sign

Receiver has to verify signature i.e verify if signature corresponds to M.
This is done using public key of sender.

$\therefore (\text{Signature})^e \bmod N.$

$= (M^d)^e \bmod N = (M^{d \cdot e}) \bmod N$

$= M. \bmod N.$

$= M.$

Compare this M (from signature) with M sent by sender.

Verifies authenticity/signature, verifies that it is signed by ~~a~~ genuine entity.

Suppose, attacker want to send a message.
Attacker sends message in a way such that the receiver should think that message came from genuine sender and not attacker.
∴ He needs signature of sender.
└ Almost similar to chosen CT attack.
Objective is to find signature from sender
∴ Attacker generates a random no. such that inverse exists wrt N.

Attacker computes $M^* = r^e M \bmod N.$ → publickey

↑ for this he needs ~~M~~ signature

Attacker requests sender To sign M'.
If sender agrees to sign his private key

$(M')^d \bmod N.$

$= (r^e M)^d \bmod N = r^{ed} \cdot M^d \bmod N.$

$= r \cdot M^d \bmod N.$

M^d is the signature for M from sender.
∴ To obtain value M^d, multiply r^{-1}

Date: / /

$R \cdot M^d \bmod N \xrightarrow{\times R^{-1}} R^{-1} \cdot R\, M^d \bmod N$

$= 1 \cdot M^d \bmod N$

$= M^d \bmod N.$

Signature for M obtained ✓

∴ Now attacker can send message + signature to receiver

~~Plaintext Attack~~

to 01/2/18

Congruence of Squares - To compute factors of N.

To factor $N \rightarrow N = x^2 - y^2 = (x-y)(x+y)$

Suppose $x^2 = y^2 \pmod N \rightarrow x^2 - y^2 \equiv y^2 \bmod N$?

Then $x^2 - y^2 = k \cdot N$ | compute $(x+y)$ & $(x-y)$ ~~to find~~ gcd with N

2 simple methods if N is small

Ex: $34^2 = 8^2 \pmod{91}$

$x^2 - y^2 \equiv y^2 \bmod N.$

Ex: If N = 1649.

Step 1: Take some integer say 41.

find: $41^2 \bmod 1649 = 32.$ —①

Take another integer, say 43.

$\therefore 43^2 \bmod 1649 = 200.$ —②.

Step 2: Multiply 41, 43.

$32 \times 200 = 6400 = (80)^2.$

$(41 \cdot 43)^2 \equiv (80)^2 \bmod N.$ (in the form of $x^2 \equiv y^2 \bmod$

$32 = 2^5 \times 5^0$

$200 = 2^3 \times 5^2$ $\Big\}$ $(41 \cdot 43)^2 = 2^8 \times 5^2$

$6400 = 2^8 \times 5^2$ ②

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

$\therefore (41 \cdot 43)^2 \equiv (2^4 \times 5^1)^2 \bmod 1649$

papergrid

Date: / /

$32 = 2^5 \times 5^0$ take exponents 5 and 0 and apply mod operations with ~~bases~~ 2.
~~5 mod 2 and 0 mod 5.~~

$32 \rightarrow \begin{bmatrix} 5 \\ 0 \end{bmatrix} \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix} \pmod 2$. $200 \rightarrow \begin{bmatrix} 3 \\ 2 \end{bmatrix} \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix} \pmod 2$

⑤ Common Modulus Attack

For encrypting M - ~~to~~ use 2 public keys (e_1, N) and (e_2, N) → same N. (say it is for 2 diff users or same user with diff keys).
Is it poss to derive M from intercepted C.T ?

M — (e_1, N) public key1 — $\gcd(e_1, e_2) = 1$
(e_2, N) public key2 — [e_1, e_2 must be coprime]

$c_1 \equiv M^{e_1} \pmod N$
$c_2 \equiv M^{e_2} \pmod N$
} same message, same modulus, diff. public keys.

∴ there is possibility of deriving M.

Attacker must have 2 values in addition to c_1, c_2, e_1, e_2, N → x & y such that :

$x \cdot e_1 + y \cdot e_2 = 1$

Compute : $c_1^x \cdot c_2^y \pmod N$

$= (M^{e_1})^x \cdot (M^{e_2})^y \pmod N$

$= M^{e_1 x + e_2 y} \pmod N$

$= M \pmod N$ [$e_1 \cdot x + e_2 \cdot y = 1$.]

∴ M derived

⑥ Plain Text Attack - Short Message Attack

Assumption : Message domain size is very small.

Date: / /

Assume message = 4 bits
∴ Possible messages are: 0000, 0001, ...
(e, N) publically known.

$C = M^e \bmod N$.

Q2 If C is intercepts C, one can easily identify M by exhaustive search method. As you know C, e, N - since the poss. messages are less.
find $(M_1)^e \bmod N$ and see if it matches C
Keep doing this till message is found.

$(M_1)^e \bmod N = C_1$	Message size should be small.
if $C \neq C_1$,	
$(M_2)^e \bmod N = C_2$	To mitigate this attack: add bogus/padding bits. so msg not readable to attacker even if he decrypts
if $C \neq C_2$	
.	
$(M_n)^e \bmod N = C_n$	

Q3 Suppose Bob uses RSA cryptosystem with a large modulus N for which factorization can not be found in a reasonable amount of time. Suppose Alice sends a message to Bob by representing each alphabetic character as an integer betn 0 to 25 and then encrypting each no. separately using RSA with larger e and larger N.
Is this method secure?

Ans → short message attack
→ encrypt one char at a time (stream cipher)
→ each no. is encrypted separately ∴ M is a single character.
(M) → short message attack possible → 26 possibilities only