SYSTEM SECURITY BY PROF. FARUK KAZI

System Security.

Attacker may attack:

1. DB
2. OS
3. Programs
4. Network

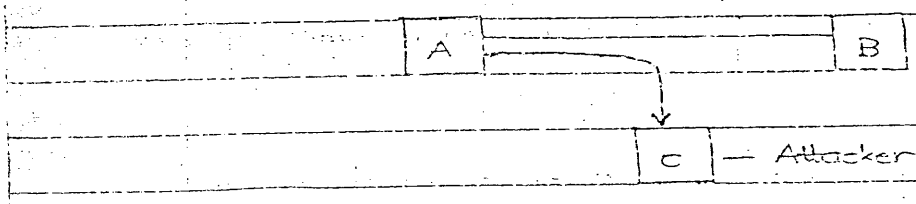
Security threats

→ Types of Attack:

1. Interception — A Passive Attack.
2. Interruption
3. Modification } Active Attack.
4. Fabrication

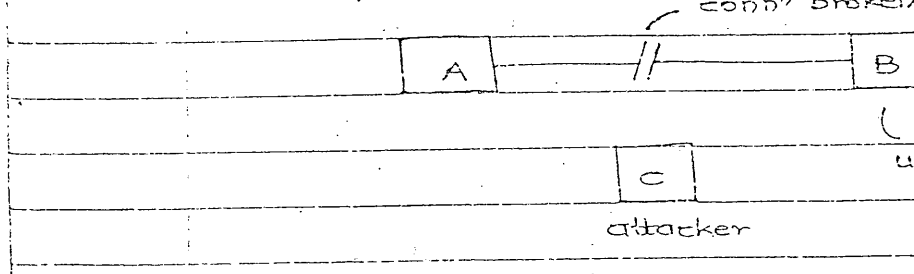
Spoofing
masquerades
session hijack
man in middle attack.

1. Interception:



(Attacker cause no harm)

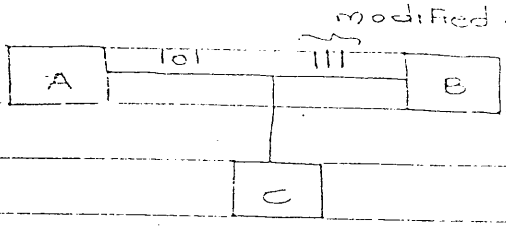
2. Interruption: / Denial of service



a) Break.

b) Congestion: Attacker overloads line with unwanted data (Attacker harms data)

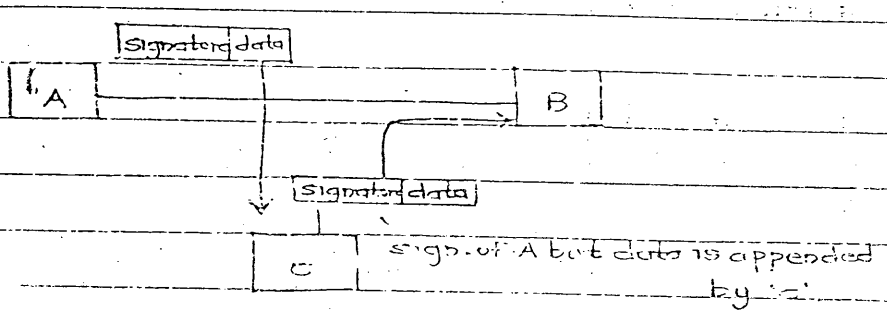
3. Modification:



- Attacker modifies the data

4. Fabrication: / Masquerade

- Attacker generate new data & send it



- Attacker resbuffles bit in such a way that new msg is generated.

* Masquerade:

One user pretends to be some other user.

→ Security:

1. Preserve confidentiality.

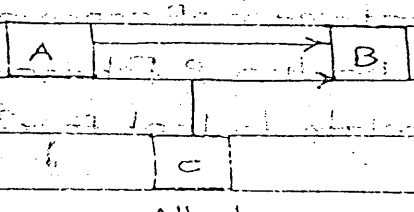
- data should be understood by intended user only.

- only B should receive & understand data send by A

※ - encrypt data such that only B can decrypt it

2. Authenticity:

- Only authenticate user should send data.



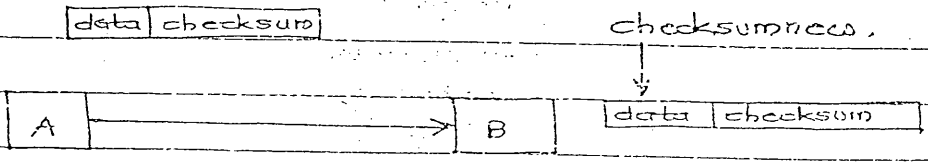
- B should be able make out that A is authenticated user & not C.

※ - Digital signatures used.

3. Integrity:

- data should reach user as it is, it should not be modified by attacker. In case if it modified, user should understand that

※ - checksum used.

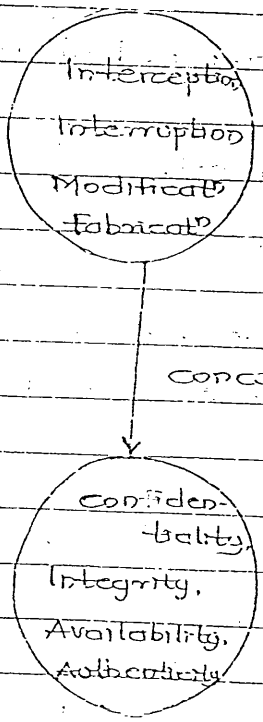


If checksum ≠ checksumnew ; data is modified.

A Availability:

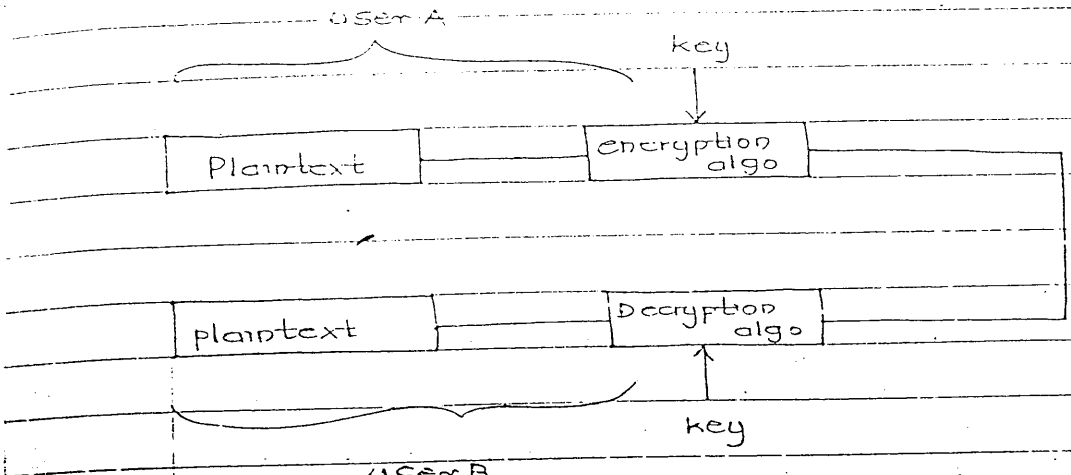
MVA
- If type of attack is Interception ⇒ confidentiality not preserved.
- Authe fabrication/Modification ⇒ Authenticity not preserved.
- Modification ⇒ Integrity.
- Interruption ⇒ Availability.

EQ May 2005 Q.3 (a)
Preserving confidentiality, Integrity & Availability of data is a restatement of concern over interruption, interception, modification & fabrication. How do the first 3 concepts relate to last four?



2. Cryptography. → Cryptosys → Cipher

→ Cryptography - Preserving confidentiality.



a. Symmetric cipher — single key.
b. Asymmetric cipher → Seperate key for E & D.

a. Symmetric cipher
- Substitutional technique
- Transpositional

2. Substitutional tech

I. Caesar cipher:

$c = (p+3) \bmod 26$

(to make it cyclic.

eg: plaintext = a b c d ... z
ciphertext = d e f g ... c

- each alphabet substituted by new alphabet

Generalized: $c = (p+k) \bmod 26$

Q. Diff b/t encryption algo & key

For Caesar cipher,

algo = $C = (p+k) \bmod 26$

key = k.

* For Caesar cipher,

P = a b c d

encry algo = $c = (p+k) \bmod 26$; key = k

C = d e f g

decryp algo = $P = (c-k) \bmod 26$; key = k

Both keys are same, hence, Symmetric cryptography.

- Sender & Receiver use same key.

VIVA Caesar cipher - substitution, symmetric crypto.

II Polyalphabetic cipher / Book cipher:

- Sender & receiver maintains same book

key \ Plaintext	a	b	c	d	.....	z
a	A	B	C	D	.....	Z
b	B	C	D	E	.....	A
c	C	D	E	F	.....	B
d	D	E	F	G	.....	C
⋮	⋮	⋮	⋮	⋮		⋮
z						

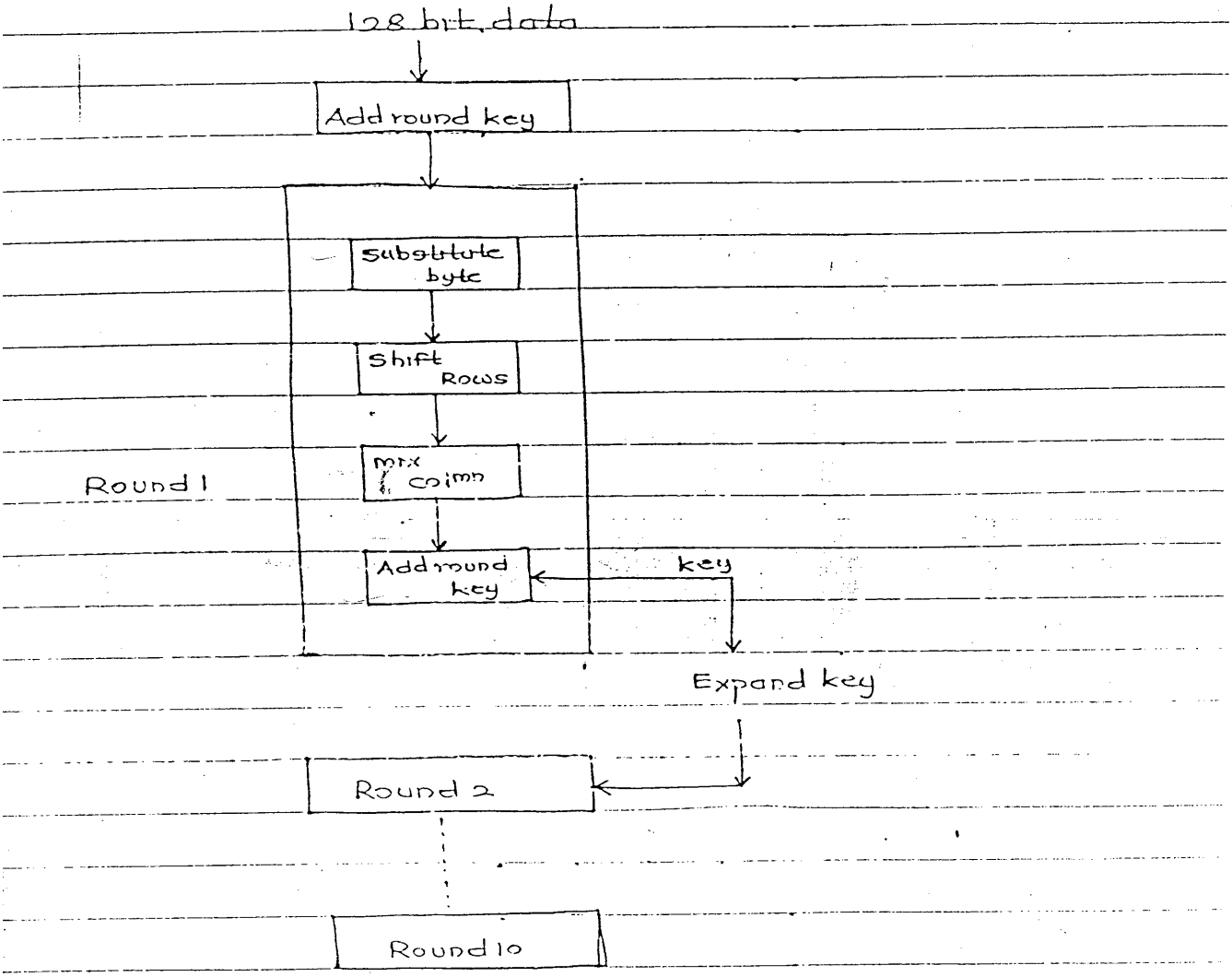
} ciphertext

- Book is part of algo, hence known to everyone, only key is secret.

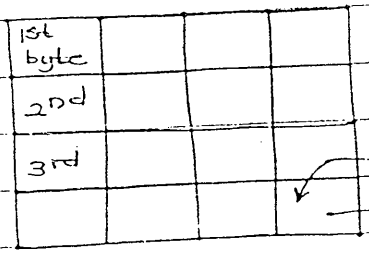
VIVA

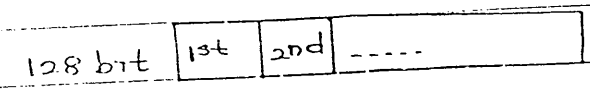
3. AES (Advanced Encryption Std).

- takes 128 bit data.



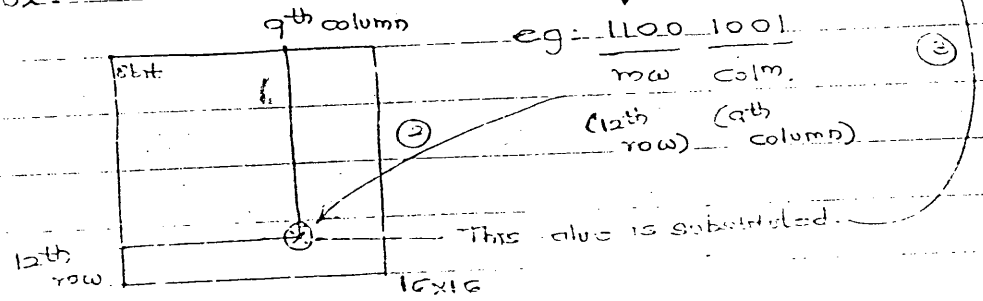
a. Substitute byte:



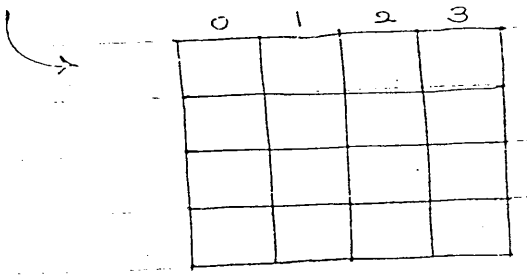


①

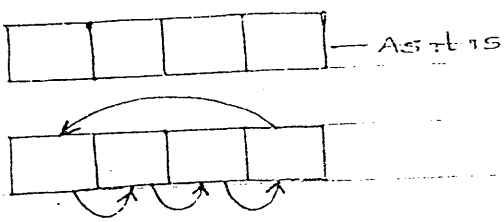
— S-box:



— Here, actual data bytes are substituted by the ones in S-box.



b. Shift rows:



eg: Sender , Receiver

key: ABCD <repeat the key> key: ABCD

plaintext: CAD BD AB ciphertext: CBF E

ciphertext: CBF E plaintext: CAD

III. One time Pad:

- one key used for sending one comm? msg.
- key/one time pad discarded after use/dynamic key.

eg: key b c d z j k l m n

plaintext a b c d

$$c = (p+k) \bmod 26$$

ciphertext = c e g

for key take position of alphabet

1) (p+k) = (2+a) = c

2) (p+k) = (3+b) = e

3) = (4+c) = g

a. Transpositional technique:

- alphabets/message remains as it is, the position of alphabets are shuffled.

eg:

Sender	Receiver
key: 4 1 3 2 5	key: 4 1 3 2 5
plaintext: A T T A C	plaintext: A T T A C
K P O S S	K P O S S
I B L E X (X to fill the blank. (for rx at Rx).	I B L E X

ciphertext: IPB ASE TOL AKI CSX

1 2 3 4 5

↑ arrange alphabets belong key 1

- Rx counts no. of alphabets in ciphertext, which is ÷ by no. of alphabets in the key. (equal division)

b. Asymmetric cipher / Public key cryptography:

	UserA	UserB
corresponding pair of key.	public key (avail. to everyone) = kUa	kUb
	private key = kRa	kRb

eg: kRb (encrypt) ⟶ kUb (decrypt)

kUc (msg) ⟶ kRa (decrypt).

U → Public

R → Pr

A → B

1. $E_{KU_b}[M]$ → $K_{Rb}[M]$ — confidentiality.

(encrypt 'M' using public key of 'B'

- Here, even attacker can encrypt the msg., hence, authentication not preserved

2. $E_{KR_a}[M]$ → $K_{U_a}[M]$ — authenticity

- Here, confidentiality not preserved, since, everyone has public key of A, however, other end sure that only A has encrypted the msg; hence, authenticity preserved

3. $E_{KU_b}[E_{KR_a}[M]]$ (① inner, ② outer) → K_{Rb} → E_{KR_a} → K_{U_a} → M — authenticity & confidentiality

- Both confidentiality & auth., both are preserved

※ - used in DIGITAL SIGNATURE

- logic used: R_a U_b R_b U_a

→ key Management:

- Apart from public & private key, session keys are used.
- session key (ks) used for one session.

* key distribution
  - a. public key distribution
  - b. Session key distn.

a. Public key distribution:

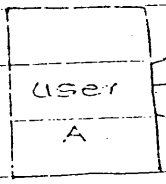
i. Public announcement of key

ii. Public discovery directory

* iii. Public key authority

* iv. Public key certificate → Digital Certificate

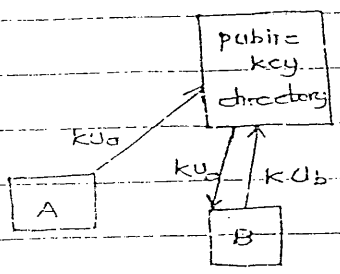
1. Public announcement of key:



- Each user announce key to everyone

disadv: - Authenticity not preserved - Attacker may pretend to be some user & transmit the public key.

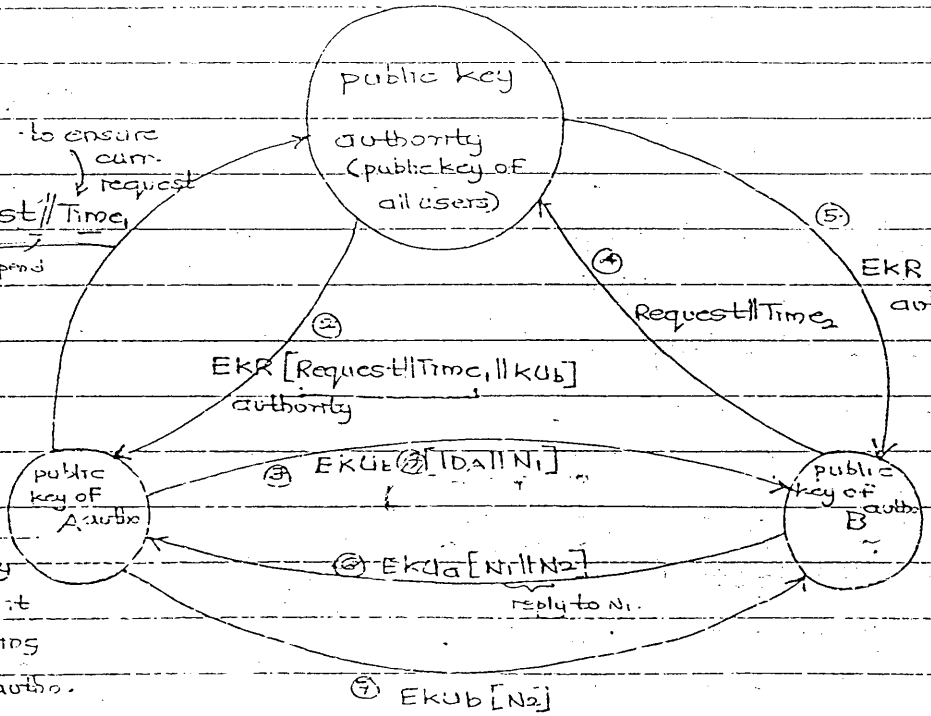
ii. Public key directory:



- Every user enters the public key in directory.

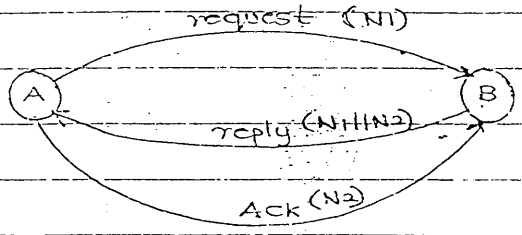
Disadv:- key can be modified by anyone ie: attack on directory.

iii. Public key Authority: ('A' wants to communicate with 'B')



(Establishment of conn? by getting public keys).

* Three way handshaking:-



ID_A : Identification of A

N : Nounce value

Sequence No (eg: TCP/IP) — Time → when attacker uses the pkt later, the time gets old — not useful

Public key Certificate → Digital Certificate.

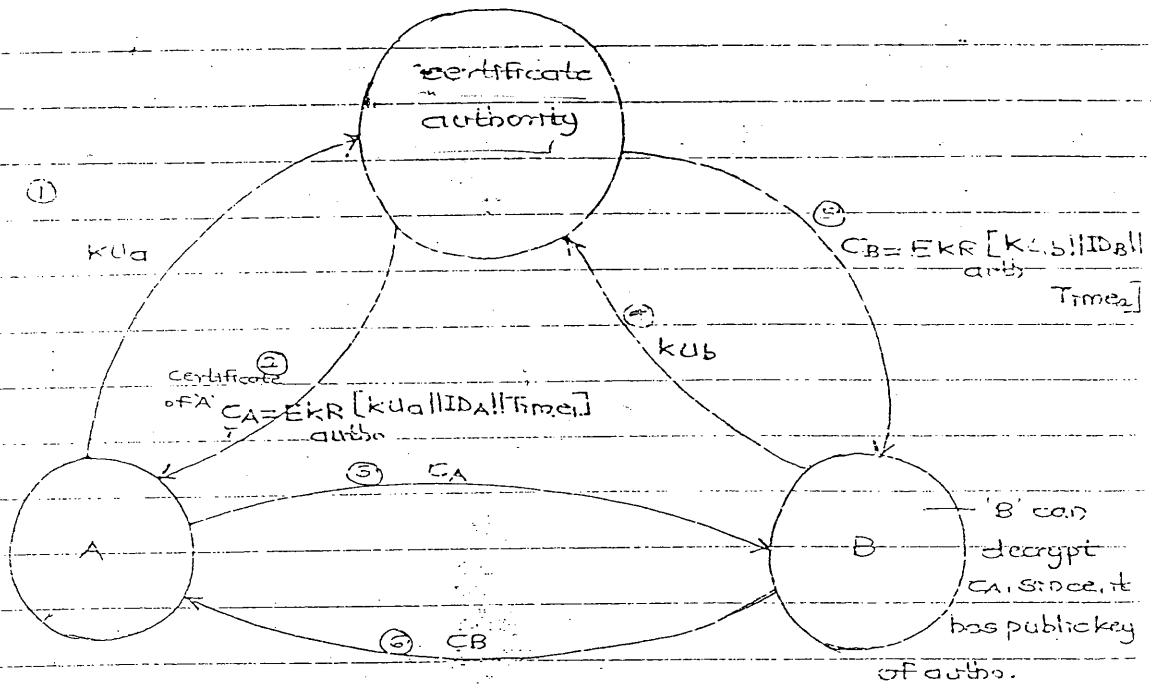
Disadv: – Attacker can acquire public key of authority.

* – Here, every comm? is encrypted except the Request msg., because, autho is sure that msg is coming from authenticated user., since it can look for IP address of user.

VIVA

iv. Public key certificate:

– CA issue certificates to users. User can use this certificate to prove its identity to another user.



b. Session key distri :

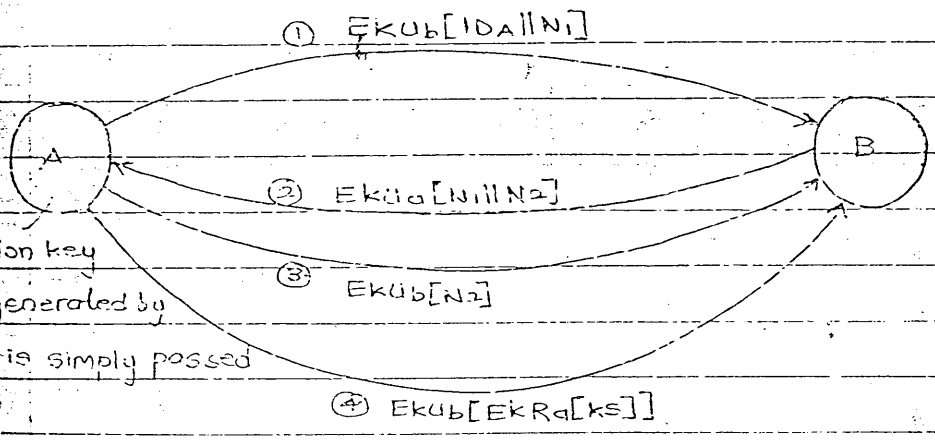
- Assumption: Public keys are already distributed.

VIVA * Advantage of session key:

To ensure authenticity & confidentiality, encryption is done twice.

eg: $E_{KU_b}[E_{KR_a}[M]]$

Now, if 'A' & 'B' know session key, msg encrypted just once ie: $E_{Ks}[M]$.



- confidentiality & authenticity preserved.

Steps 1, 2, 3 → Handshaking procedure,

Step 4 → Session-key is passed

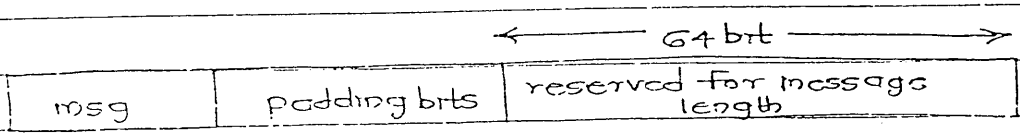
Henceforth, msg is passed as - $E_{Ks}[M]$.

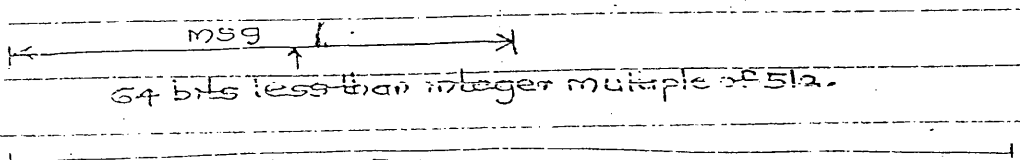
→ Cryptographic Checksums / Hash fn:

* Methods to generate checksum:
a. ~~Method~~ message Digest (MD5)
b. Simple Hash algo (SHA)

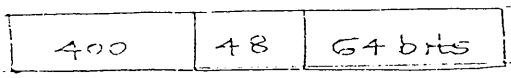
a. Message digest (MD5)

Step 1: Append padding bits

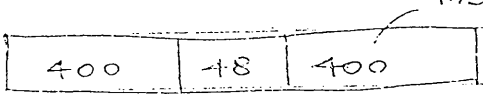




eg: The whole msg should be multiple of 512
If msg length = 400, 64 = reserved, hence
48 ~~pa~~ bits are padded.



Step 2: Append message length

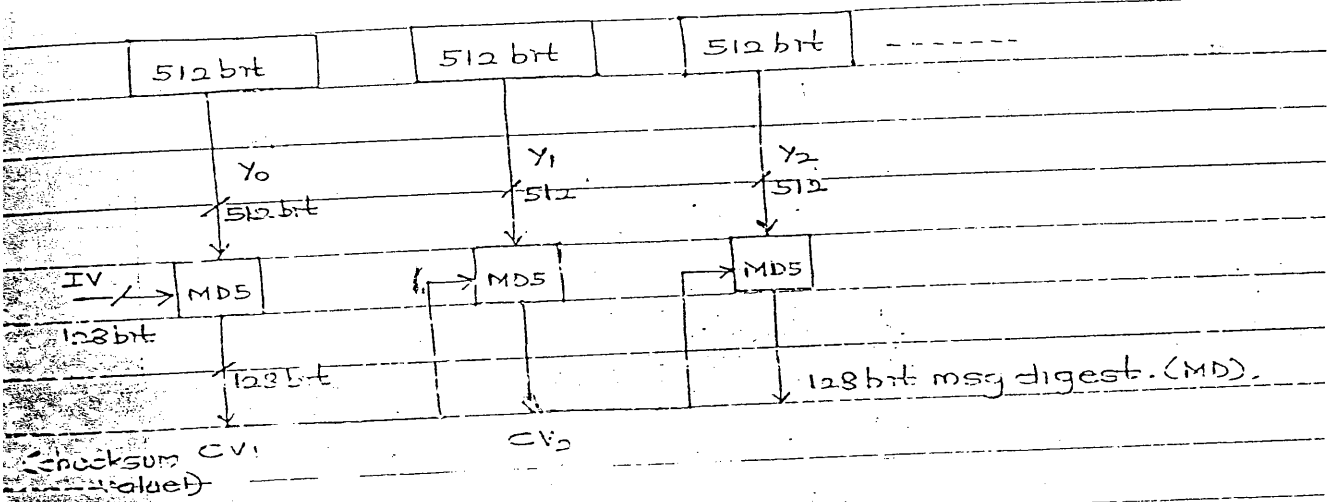


Step 3: Initialize buffers (Initial value (IV))
To put Some 32 bit value to A, B, C, D Buffer

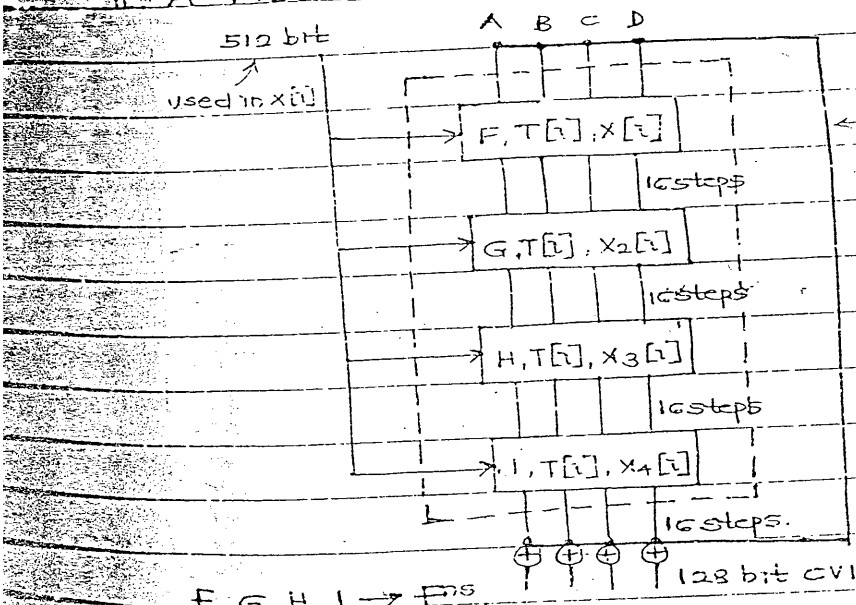
32 bit A = 67452301
32 bit B = EFCDAB89
32 bit C = 98BADCFE
32 bit D = 10325476
128 bit

stored in a little-endian format

I functional diagram:



II A MD5 Block:



F, G, H, I → F^5

T[i] → 16 table values (32 bit). (Fixed table).

Step 4: Process msg in 512 bit (i.e. 16 word) Bl

Step 5: Output

$a \leftarrow a + f(b,c,d) + x[k] + T[i] \lll s$ → circular shift of f.

a, b, c, d → four words of buffer.

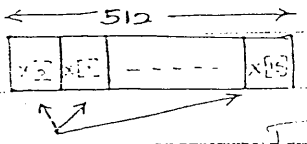
f → special fn.

$T[i]$ → Table value of 32 bit.

$x[i]$ = 32 bit word from msg.

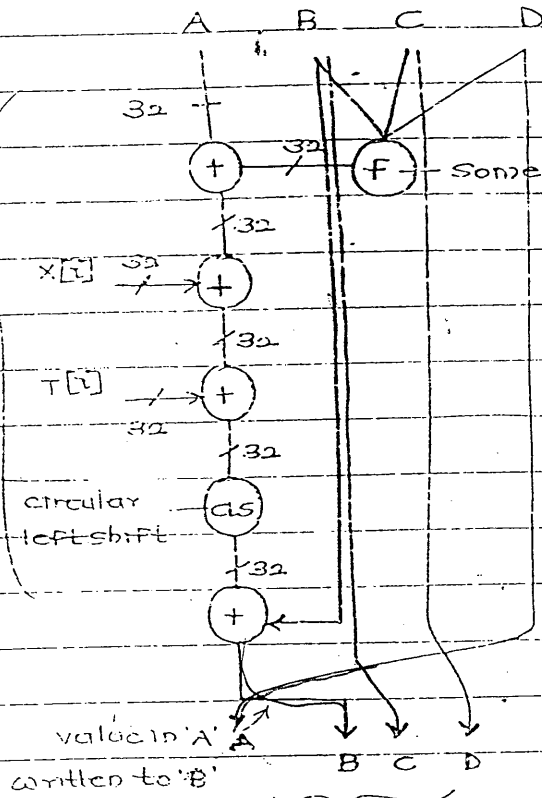
512 bit – message = 16 values.

32 bit – words



$x[k] = M[q * 16 + k]$

III. One step:



Values written as 'A to B', 'B to C', 'C to D' & 'D back to A'.

– This step repeats 16 times with diff X[i] & T[i] values.

Here msg is not changed/modified

– Buffer values are transferred from fn to fn.

$x_2[i] = (i+5i) \bmod 16$

for 1st fn ie: 'F', $x[i] = x[1], x[2], x[3], \ldots x[16]$ used,
for 2nd fn ie: 'g', $x[i]$ = random value from $x[i]$.

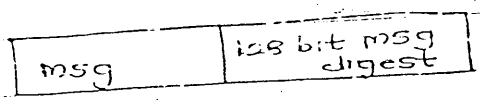
$x_2[i] = i=1;\ 6 \bmod 16 = 6$ ie: $x[6]$ used, } diff. sequence
$i=2;\ 11 \bmod 16 = 11$ ie: $x[11]$ used, }

$x_3[i] = (5+3i) \bmod 16$

$x_4[i] = 7i \bmod 16$

Hence, msg remains unaltered while buffer keeps changing.

* Advantage:



- At Rx, if attacker sends a modified data, the MD generated (cmt msg) will be different, since MD depends on the msg.

* Advantage:

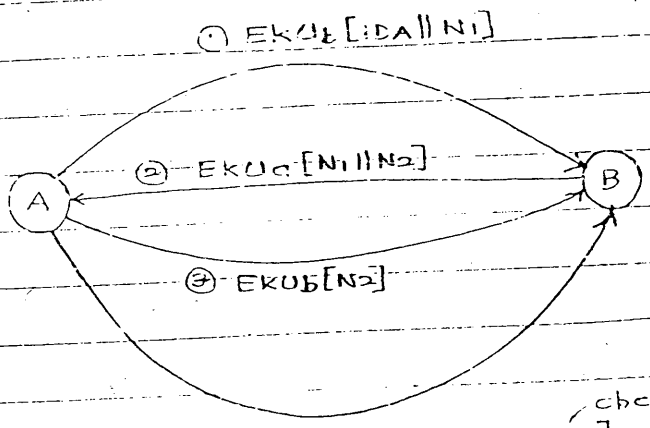
1. In MD5, 4 complex fns are used. Each fn performs 16 steps, so total of 64 steps are performed to generate 128 bit MD. This complexity makes MD or checksum unbreakable.
(procedure to generate checksum is difficult, otherwise the attacker will change/modify msg as well as checksum).
2. In MD5, for every step msg bits are used that means

checksum is dependent on msg. Hence, it is difficult to break checksum.

b. Simple Hash Algorithm (SHA):- (Explained latter).
- 160 bit MD.
- makes checksum more complex, thus unbreakable.

(May 05) 2. a) Public keys are distributed using public key authe. (Explain the procedure).

b)



- Steps ①, ② & ③ are for conn? establishment, where conn? is established betn A & B using 3-way handshaking procedure.
- In step ④, Hash value can be calculated on session key 'ks' using either MD5 or SHA.

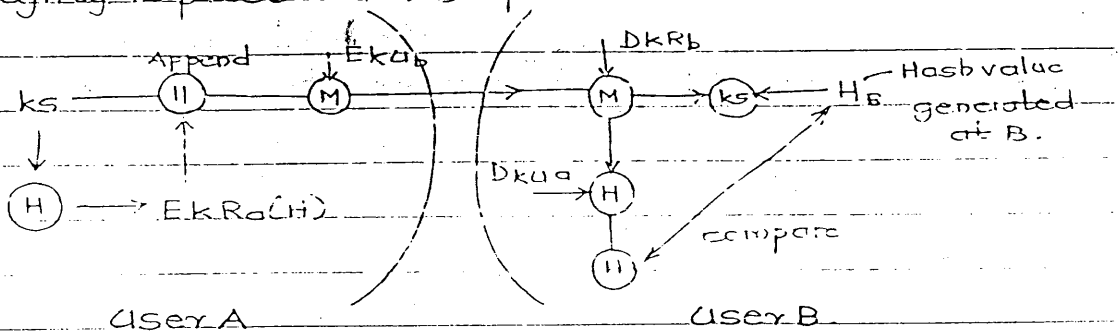
- This Hash value is then encrypted using private key of sender (or A). This is then appended with session key & the whole msg is encrypted using public key of receiver (or B).

a. Confidentiality:

Here, confidentiality is preserved as session key is encrypted using public key of B. So, we are sure that msg can be decrypted only by B.

b. Integrity:

Integrity is preserved in Step 4.



If $H_B = H$; integrity is preserved.

c. Authentication of Sender:

This is preserved in Step 4, as hash value is encrypted using private key of A.

d. Non-repudiation on Sender Side:

Here, sender can't deny of having sent msg as sender initiates comm^n by sending its identification (IDA) in Step 1. Further, Hash value is also encrypted using

private key of A.

Non-repudiation on Receiver side:
Here, receiver can't deny of having received msg as Rx replies back using nounce values N_1 & N_2

e. No possibility of Reply Attack:
Here, reply attack is not possible as handshaking is done using nounce value which is time value. This value is going to differentiate between old & new packets/msgs.

→ Diffie-Hellman key exchange Algo:
- both Sender & Rx generate ks at same time

	A	B
globally known	1. prime nos: p, q	q
	α ; $\alpha < q$	α ; $\alpha < q$
	2. private vars: X_A	X_B
	3. $Y_A = \alpha^{X_A} \bmod q$	$Y_B = \alpha^{X_B} \bmod q$
	Y_A →	
	← Y_B	
	4. key $k_1 = (Y_B)^{X_A} \bmod q$	key $k_2 = (Y_A)^{X_B} \bmod q$
	$k_1 = k_2$	

oof: $k_1 = (Y_B)^{X_A} \bmod q$

$= (\alpha^{X_B} \bmod q)^{X_A} \bmod q$

$= (\alpha^{X_B})^{X_A} \bmod q$

$= (\alpha^{X_A})^{X_B} \bmod q$

$= (\alpha^{X_A} \bmod q)^{X_B} \bmod q$

$= (Y_A)^{X_B} \bmod q$

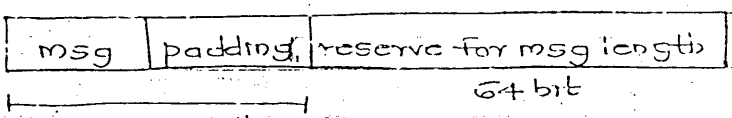
$= k_2$

b Simple Hash Algo (SHA): → developed by NIST (National Institute of Std & Tech)

Secure Hash Algorithm

based on model of MD5

tep1: Append padding bits.



tep2: Append message length

tep3: Initialize buffers.

32 bit A = 67452301

32 bit B = EFCDAB89

32 bit C = 98BADCFE

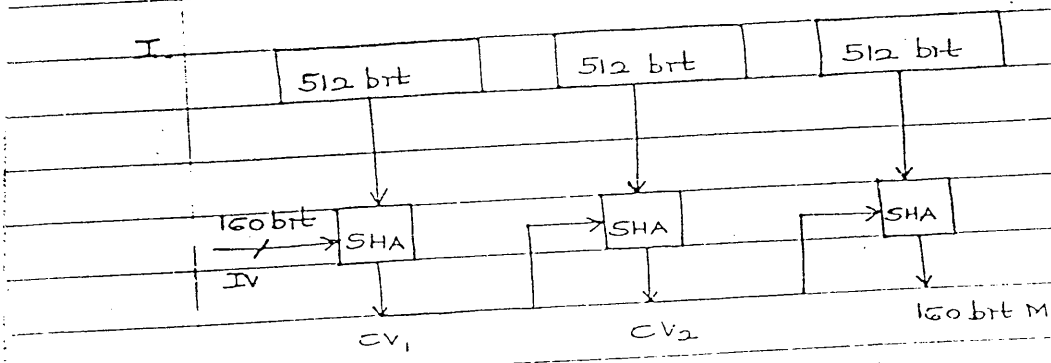
32 bit D = 10325476

32 bit E = C3D2E1F0

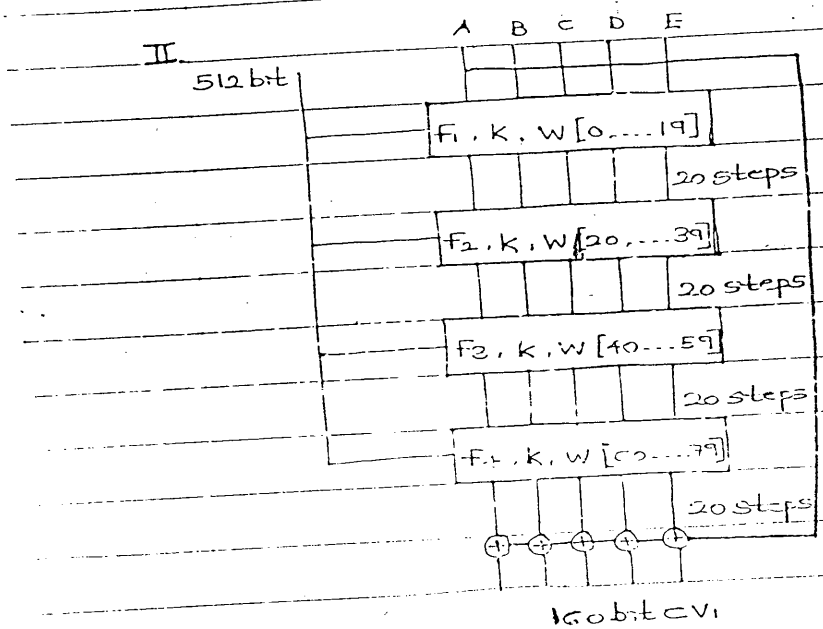
160 bit.

Values stored in a Big-Endian format

I.



II.



k – 32 bit word.

W[i] – 32 bit word from msg

(similar as X[i], but here, we have 16 values and 20 steps (4 values less)), hence, we use the foll expression –

$$W_t = S^1(W_{t-16} \oplus W_{t-14} \oplus W_{t-8} \oplus W_{t-3})$$

↑ left shift by 1

A<B<<CLDKE ← E + $f_t(B+C+D) + S^5(A) + W(t) + X[K]$

we have, W[0] ---- W[15] ie: 16 values,

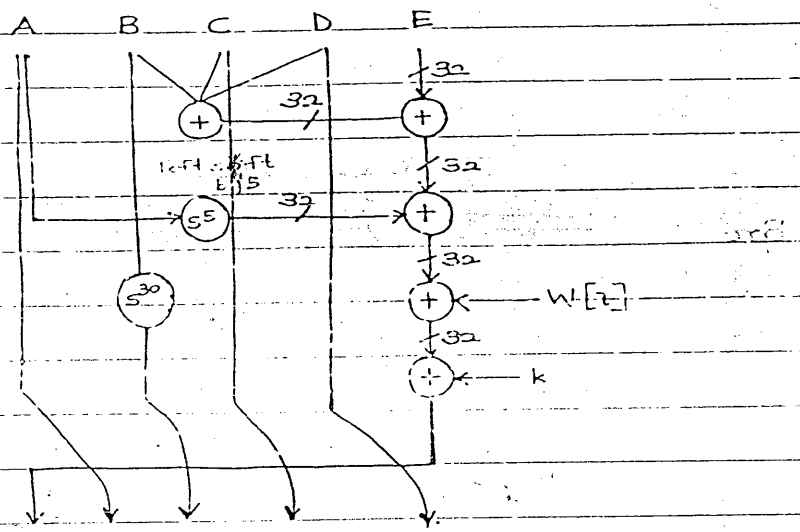
$W_{16} = S^1(W_0 \oplus W_2 \oplus W_8 \oplus W_{13})$ (old values used to generate new ones.)

= (32 bit)

$W_{17} = S^1(W_1 \oplus W_3 \oplus W_9 \oplus W_{11})$

= Similarly, W_{18} & W_{19} are calculated.

II. One step:



MD5	SHA
1. 4 x 16 = 64 steps	- 4 x 20 = 80 steps
2. 128 bit MD	- 160 bit MD
3. less difficult to break as compared to SHA	- more diff. to break as compared to MD5.
4. 4 buffers.	- 5 buffers.
- Buffer value stored in little endian format	- Buffer value stored as a Big-endian format.

→ cryptography:

* Attacks on ciphertext:
1. Cryptanalysis
2. Brute force attack

1. Cryptanalysis:
- Attacker tries to analyze encrypted data.
- first tries to find group of words which comes together.
  eg: ack bidjk ad mpp ack
      The t the
- Attacker interested in the message.

2. Brute force attack:
- Attacker is interested in knowing the key.
- More harmful than cryptanalysis since if key found, entire message can be decrypted.

* cryptography can be characterized by-

1. operations used:
   - substitution
   - Transposition

2. keys used:
   - symmetric
   - asymmetric (public-key crypt)

3. Way in which plaintext is processed:

- Stream cipher
- Block cipher

a. Stream cipher: continuous encryption whole msg is taken at once & each alphabet is encrypted.
(eg: All the algo. seen previously)

b. Block cipher: Blocks ~~te~~ of msg are formed., these blocks are encrypted at a time.

→ A pgm which is free of flaws

Program Security.

1. Secure programs → Secure pgm is the one which is difficult to attack in
2. Non-malicious program code
3. Malicious program code
4. Targetted malicious code
5. Controls against pgm attack

1. Secure programs:

Parameters based on which security is decided:

a. Fixing fault: If it is easy to fix a fault in pgm, it is secure pgm. Fixing fault is also called patching fault.

b. unexpected Behaviour: If unexpected behavrour is avoided or certain actions are prevented, pgm is secure.
eg: halting inbetn — unexpected behaviour, msg display when incorrect data entered.

c. Types of Flaws: (pgm)
i. Validation error: I/p data should be checked or validated & then processed.

ii. Incomplete or Non-existent authentication:
eg: username & pswd asked From user, but if username is entered as administration, access is provided with, pswd (Incomplete authentication) (out)

iii. Boundary cond? violation: user enters data which is not inbetn boundary.

Intentional — Non-Intentional

Malicious — Non-Malicious

→ Validation error
→ Serialization & Aliasing
→ Inadequate ID^n & Authentication
→ Improper Access Control Violation

2. Non-malicious program code:

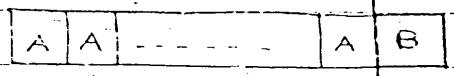
- may lead to/result into loss such as financial loss, data loss, modification, does not destroy the system.
- Malicious code may destroy user system.

Eg a. Buffer overflow
b. Incomplete Mediation.
c. Time to check to Time to use error./Serialization or Synchronization flaw

a. Buffer overflow:

```
eg: for (i=0; i<9; i++)
    {
        fun[i] = 'A';
    }
    fun[10] = 'B';
```

Buffer → memory space
→ has a max. capa
→ defined by pgmr
eg: Char Test[8]



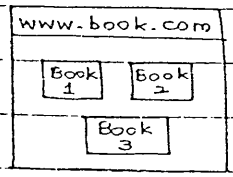
← code goes beyond space alloted to user 'A'.

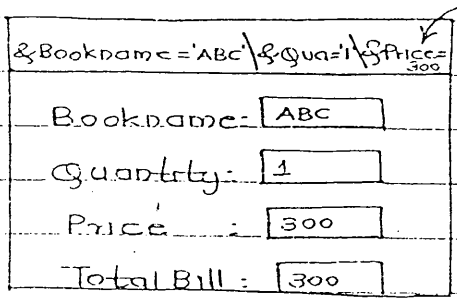
User A — User B (may overwrite data from user B space).

b. Incomplete Mediation:

Sensitive data is not protected/visible to user.

eg:





sensitive data visible to user. - can be modified by user

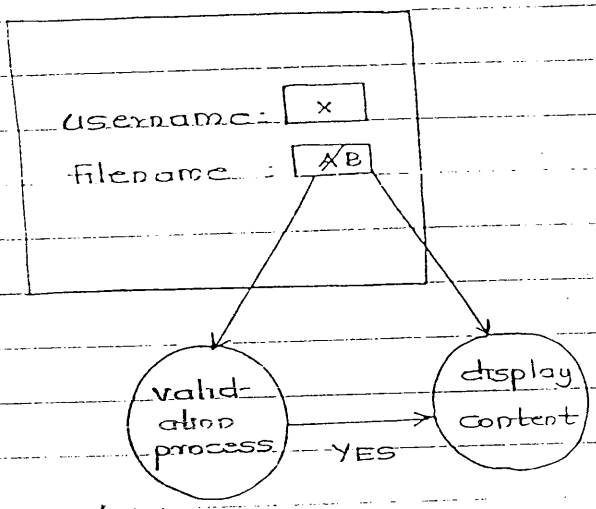
when Book 1 clicked

Soln: Sensitive data not displayed.

Time of check to time of use

c. To.c to To.u Error:

eg:



- By the time 'A' was checked for validation, if user enters filename 'B', then validation of 'A' is done while contents of 'B' displayed.

Sol?: Unless validation is over, user not allowed to modify filename.

3. Malicious program code:

eg: VIRUS.

→ * Types:

a. Virus
   - Transient (One time exec?) - virus executed not p
   - Resident - activated when that memo. is refere

b. Trojan Horse.

c. Logic Bomb.

d. Trapdoors.

e. Worm

f. Rabbit.

* Trojan Horse: MC that captures authenticated code.

eg: Login

Enter user_id: } Trojan Horse logins correctly.

Enter pswd: captures admin

Trojan Horse may attack 'login', endangering authentication.

* Logic Bomb: (Inbuilt malicious logic)

code is written such that activated at a specific time.

- Time
- Event eg: virus activated when specific file is accessed.
- Condn eg: whenever user passes info. on N/w or access N/w.
- Count eg: count on access of file.

* Trapdoor: Some fault in pgm thru' which attack is possible.

* Worm: spreads itself thru' N/w., it keeps replicating thru' N/w.

→ purpose: affect more users.

* Rabbit: Spreads thru' replication.

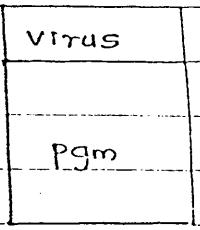
→ purpose: to block Resources of sys.

eg: replication in memo. to occupy memo resource, N/w resource, Io resources.

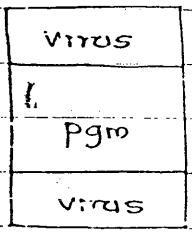
rabbits → replicates itself without limit to exhaust resources.

→ How virus attach:

I. Appended to pgm: pgm may not be executed.

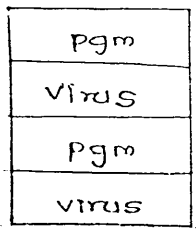


II. Surround pgm:



- code ω may be executed but may end unexpectedly or may not give result.

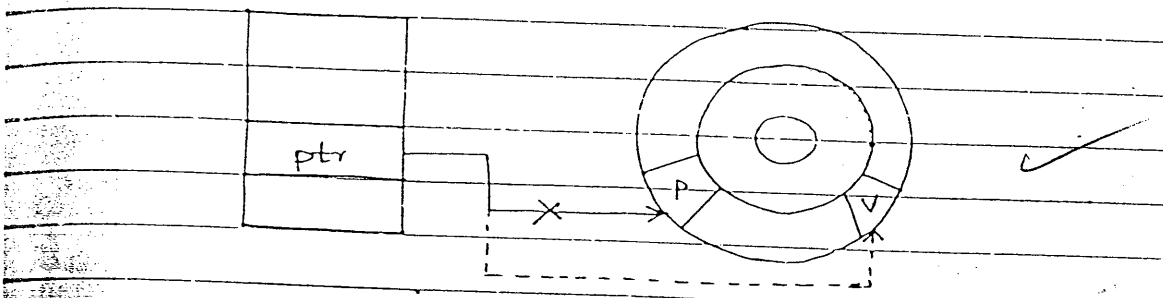
III. Integrated: inbetn the pgm code.



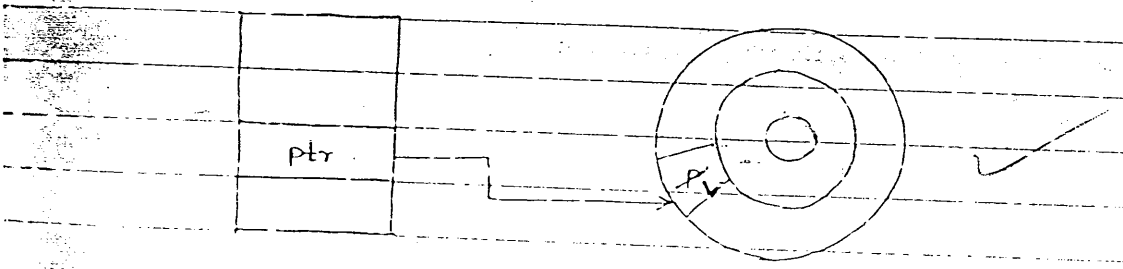
- pgm is not executed, since the pgm is affected.

→ How virus gain control: instead of pgm code, virus code should be executed.

a. virus change the ptr from pgm to virus code.



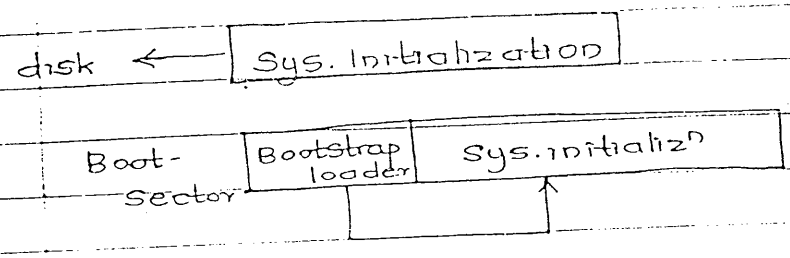
b. virus replaces the pgm code.



→ Homes for virus:

1. One time Execution: This virus does not store itself anywhere.
2. Memory: anywhere in memo.
3. Boot Sector:

- System initialization is a large code stored on disk.
- Bootsector is a part of memo containing Bootstrap loader, which loads the sys. initialization code in memo.



- If virus resides/attaches itself to Bootsector, sys. initialization not possible. (System does not start).

4. Libraries: They are header files.

→ Virus signatures:

certain patterns in virus codes.

- Virus scanner: pgm which checks / is able to locate virus. (thru' the signatures)
- patterns
  - Storage pattern = when virus is stored.
  - Transmission pattern = pattern observed when virus is on N/w.

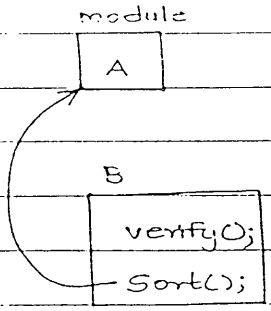
→ Controls:

a. Reliable source: To buy standard s/w.
b. Separate system: New s/w should be tested on separate system.
c. Attachment: opened only if sure of contents.
d. Back up:
e. Virus scanners:

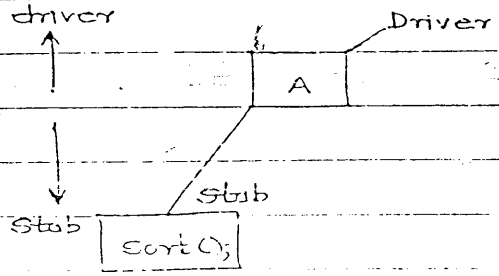
999.0 8645
999.086
(~ 9

* Targeted Malicious Code: Designed for a specific target
- Attacker first finds out deficiencies in a system. ie: find trapdoors
- Security measures: To find cause of trapdoor.
Major cause of trapdoor - Testing.
Stub: part of module, used for testing

eg:



- module B returns sort to A, so we form 'stub' ie: only Sort() part of 'B'



- If the sys. uses Stub instead of module B, sys. fails to verify();, this may lead to Trapdoor.

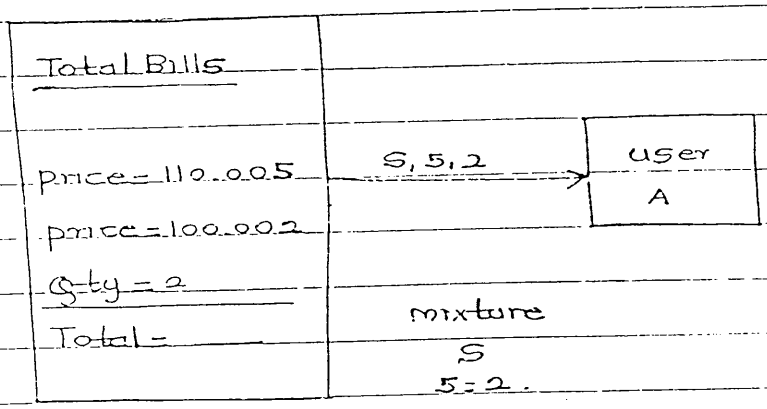
* Salami Attack:

eg: Interest 100.25
Interest 100.75

- Attackers steals small amt which would not make much difference to sys.
To avoid Salami Attack, sys must keep track of all the minute amounts as well.

* Covert channels: channels that leak info.

eg:

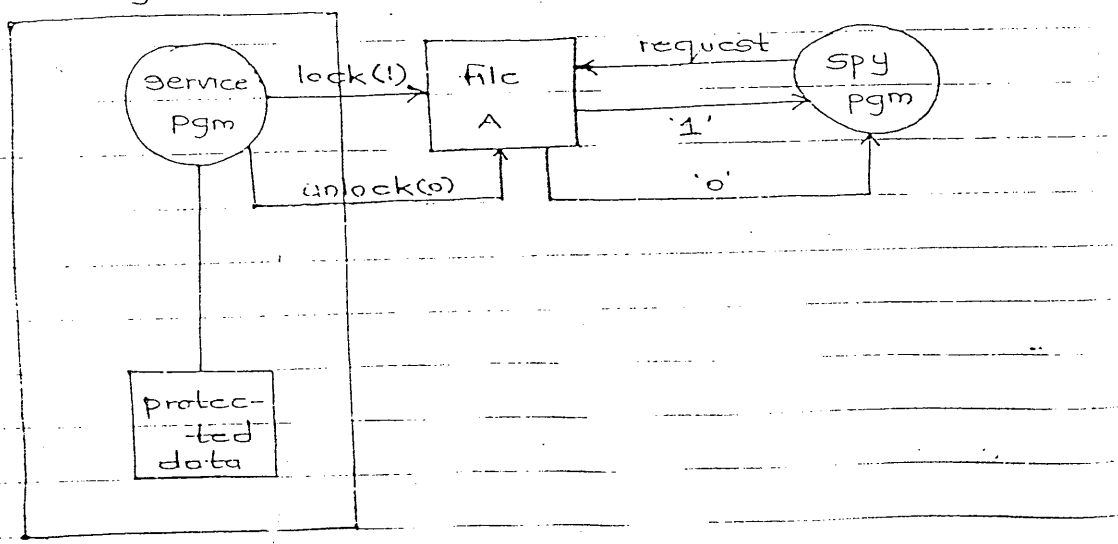


= loc^n of the leak is predecided, hence the other end comes to know abt the leakage.

- This leakage can be traded secret.

* Covert channels:
  - Storage channels
  - Timing channels.

a. Storage channels:



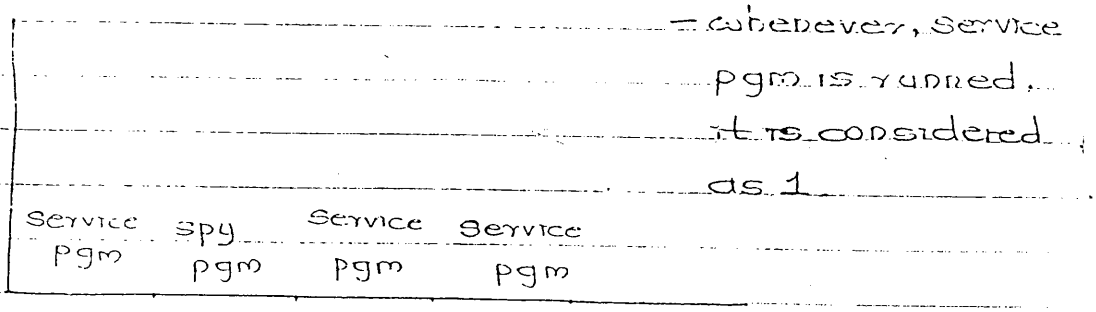
- Spy pgm running on sys but user/service pgm is unaware of it
- whenever service pgm uses file A, it locks it by making it '1', hence when spy pgm access it, it is replied with '1' (file is locked)

Info leaked:

1. when spy pgm gets '1', it understands that file is locked ie: service pgm is accessing file A.
2. The replies '1' & '0' bits received are stored by the spy pgm as Info.

b. Timing channels:

- every pgm or process assigned a timeslot.



- Every time slot, one info bit is conveyed.
- Spy pgm gets 1 info bit in each time slot.

- Spy observes only 1 pgm.

	serv:	serv1.	Spy
	Pgm1	pgm2	pgm
	1	0	0

- In one time slot, either one Service pgm may be running or not running.

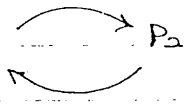
Info leaked:

1. Spy pgm comes to know when service pgm is running.
2. Bits info.

☆ Storage	Timing
1. Info passed when Spy pgm 'request'	- Info passed in every time slots

Q2. Limitations:

(GQ) 1. Deadlock: one process waiting for other process to release data & vice versa



Storage: eg: If service pgm locks file, then enters deadlock, any request from other pgm will be replied with 1s.

ie: only info received is 1.

- only 1s leaked for long time.

2. only OS may be leaked for a long time.

	SP1	SPY pgm	SP2	SP3	----
	1	0	0	0	

- Bcoz of load, SP1 is not runned/executed for long time.

→ Identifying Covert channels:

1. Shared Resource matrix
2. Info Flow

1. Shared Resource Matrix:

	processes	
	process 1 (service pgm)	process 2 (spy pgm)
pgm	R, M →	R, M
protected data	R ↑	R ↓

- Although spy pgm cannot directly Read the protected data, there is a channel to access it, thru' service pgm. (Info leak)

∴ Process 2 cannot access/communicate with Process 1.

2. Info Flow: keep of info flow

eg: A = B

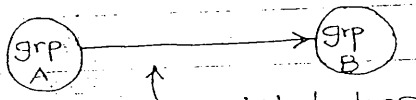
C = A (implies C = B) Indirect info leak

→ control Measures against Pgm threats:

1. System Development ctrl
- ctrls during development
2. OS ctrl
- using Audit log
3. Administration ctrl

1. System development ctrl: Steps to be followed when pgm is to be developed
a. Modularity: develop in modules
- Easy to locate modules faults

b. Peer Reviews:

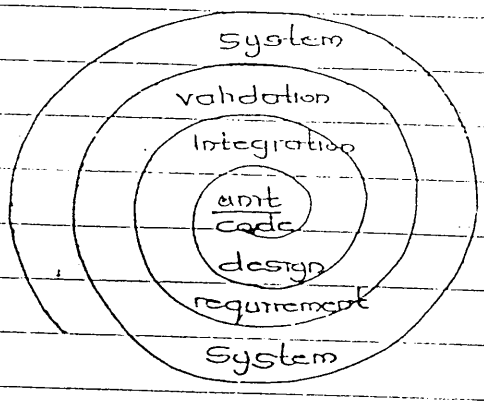


- Some faults go undetected by one grp may be detected by other grp

c. Hazard Analysis:
- FMEA (failure modes & effects analysis)
- Analyze the failures possible in future, alongwith their effects.
- The sys. can be developed such that these

predicted failures can be prevented.

d. Rigorous Testing:



e. Good Design: sys is fault tolerant.
eg: UPS does not lead to failure of entire system, even if there is no power supply.

2. OS ctrl:
Audit log: Log contains all transactions, while audit log contains user info as well.
In case of failure, we can locate the cause of failure as well as the user of the failed transaction.

3. Administration ctrl:
Diff individuals/employees assigned/distributed with security responsibilities.

Ch. 4

OS Security

1. Protected objects & Methods of protection
2. Control on General purpose objects
3. User authentication
4. Memory & Address protection

1. Protected objects & Methods of protection:

a. Objects:

i. Sharable Memory: one user should not overwrite other user's data

ii. Sharable Programs: one pgm may be shared bet' diff users with diff access rights

iii. Sharable Data:

b. Methods of Protection:

i. Separation:

ii. Protect' levels:

i. Separation: Pgm having same security requirements should be separated from others.

eg: If pgm A & B from all the pgms, are confidential then they can be separated from others.

- Data items having same security requirements are separated from others

- Separation
  - physical Separation
    - Data items having same security

Eg of Phy. Seperation : Diff. Processes are stored on diff disks which are accessed by a diff users with various access rights.

requirements are separated from others, by allocating separate resource to them.
eg: the confidential pgms A & B will have a sharable printer.

→ Temporal Separation:
Data items having diff security requirements are separated as per time.
eg: confidential data will be executed in some time period. eg: Two write processes are never executed at the same time to avoid inconsistancy.

→ Cryptographic Separation:
Data items having diff security requi. are encrypted using same algo.

→ logical Separation:

1. Eg of physical Separation:
In an organization, diff departments will have separate printers to keep data accessible only within a dept. Foreg: payroll dept will be printing salary slips every month, accounting dept may need to print balance sheets at the end of financial year.

2. Eg of temporal separation:
In a bank, employee info. is confidential & is accessible only to manager. Manager can access this before or after banking hours. This can be done by using time as additional login info.

	Confidential.	
Name	Salary	Performance Grade

- No Protection
- Isolating objects & shared all
- Shared All or Shared Nothing
- Shared via Access rights
- Shared by capabilities
- Limit use by an object

} Protection levels

username:

password:

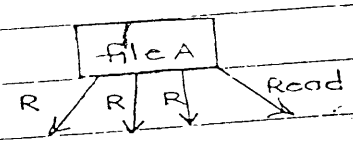
time: ← access given only if in specific time.

1. Protection levels:
   - share all or share nothing.
   - share via access rights.

↳ Share all or share nothing:

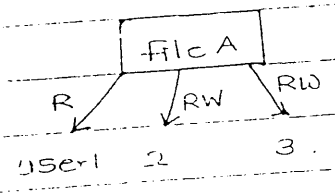
Public / Private

If the obj is shared, it is shared by everyone in same mode



↳ Share via access right

Every user share obj with diff access rights



2. Control of Access on General Purpose objects:

* General purpose objects:
  - memory
  - data
  - pgm
  - files.

- Subject = user.

Protection mechanism that are ~~need~~ used for the protection of the objects

* Control:
1. Directory.
2. Access ctrl list: (ACL).
3. Access ctrl matrix: (ACM).
4. Capabilities.

1. Directory: OS maintains a directory for every user.

User A

obj	Access mode
FileA	R
pgm	RW
FileB	R
⋮	⋮

(← Per subject wise)

adv: 1. Revocation of access.

Eg: 2 diff files with same name are not allowed.

If you have to modify the access, this modification should be done in all user's directories.

2. If user access more objects, length of directory ↑.

2. ACL: (per object wise).

FileA		
	UserA	RW
	UserB	R
	⋮	⋮

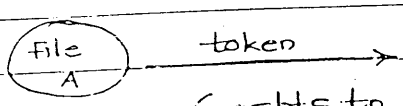
adv 1. List ↑ as more user access an object.

3. ACM:

object \ subject	user A	user B	user C	...
file A	RW	R	R	
pgm	R	R	R	
⋮				

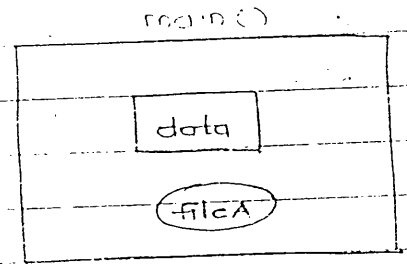
+ Capabilities:

user A — user B



- user 'A' can transfer file to 'B', it also gives permission to 'B' to transfer it some other.
- Here, whoever has a token, has a right to transfer the obj.
- This is usually done betn trusted users.

Program A



Token:

- Send token with expiry.
- Send a token stating, access right expired.

DIT PCL

sol Qs	per-subject wise	per object-wise	ACM	capability
1. Ease of determining access during exec^{n}.	Time consuming if list is long.	Time consuming if no. of user access-ing an object are more.	Easy (only one entry)	Difficult (os should know to whom all tokens are distributed)
2. Ease of adding access for new subject	Easy	Easy	Easy	Easy (one taken passed from owner to user or someone to user).
3. Ease of deleting access	Time consuming if list is long.	May be one time consuming if long list	Easy	Easy (one taken by send stating access expired)
4. Ease of creating new obj.	Difficult (Entries at diff. loc^{n})	Easy (Entries made at one place)	Easy (Entries made at one place)	Easy (owner pass taken to one user, user may send to others)

3. User authentication:

a. Use of password:

i. one time pswd

ii. Additional info

iii. Loose lipped sys

iv. Challenge response sys

i. one-time pswd: user for one time comm., small validity.

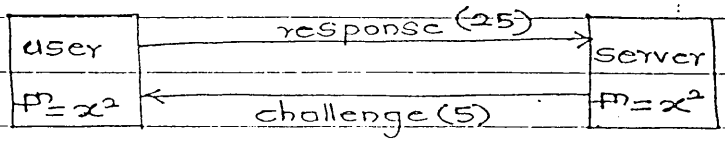
ii. Additional info: sys not only keeps track of pswd but also some additional info re time

iii. loose lipped sys:

sys 1	sys 2 (loose lipped sys)
username:	username:
pswd:	
Incorrect values	Incorrect username

- sys 1 preferred, since the attacker does not know whether username or id is incorrect.
- In sys 1 when both username & pswd entered, an error msg is prompted, hence, which value is incorrect is not known.
- In loose lipped sys, attacker can try many usernames to enter the sys.

iv. Challenge Response System:



- Both ends maintain a f(n), server places a challenge, user calculates the value using the f(n) & responses.

b. Attacks on pswd:

i. Exhaustive attack: Attacker tries all possi. values.

ii. Pswd likely for user: The attacker knows the user, attacker tries to judge the pswd.

iii. Plaintext password list: o.s checks pswd from the DB called 'pswd list'. The attacker can attack this list.

iv. Cryptographic password list: 'pswd list' is encrypted.

4. Memory and Address protection:

i. Fence

ii. Relocation.

iii. Base & Bound Register.

iv. Segmentation.

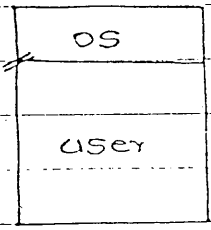
v. Paging.

* Protection:

i. overwriting: data should not be overwritten.

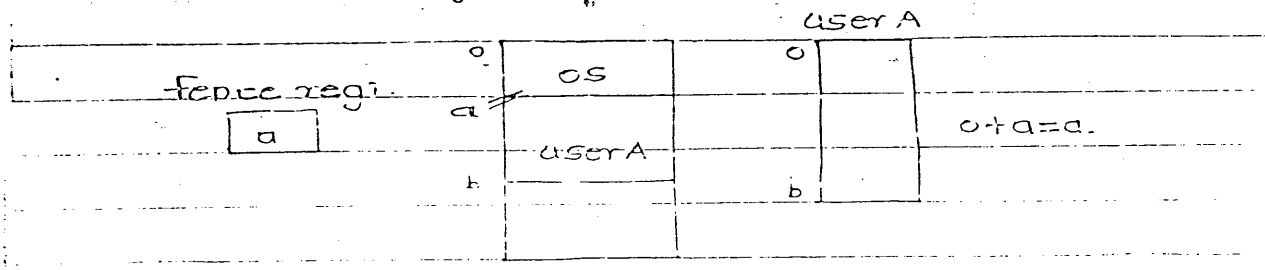
ii. Access ctrl: Access be given to authenticated user.

1. Fence: (Fixed Boundary) — Separate spaces given to diff users.



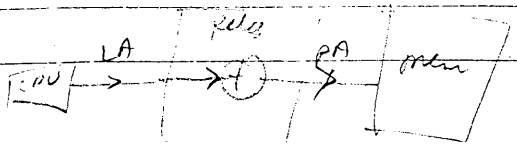
- less efficient since it indicates separation btw OS and user ie: only one user.

2. Relocation:
- uses fence register

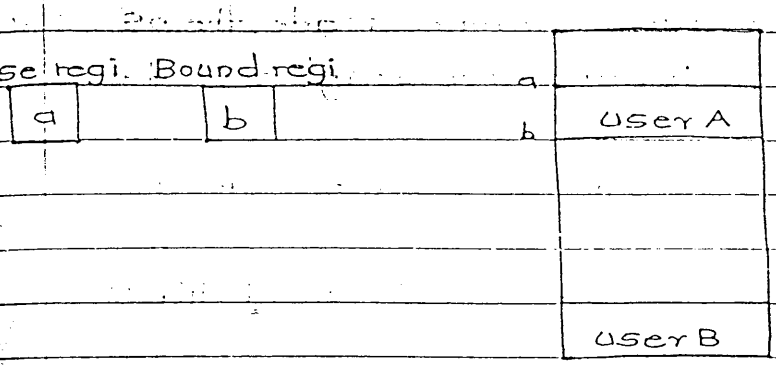


- The user A writes the code with starting addr. '0' (although addr. 0 is not avail.), this addr. is added with Fence regi. & relocated.
- fence regi. updated to 'b'
- IF, userB writes a new pgm, it starts with addr '0', then it is relocated to '0+b=b'
- fence regi. updated with size of user B pgm

Adv: user writes with starting addr '0', the pgm is relocated acc. to fence regi.

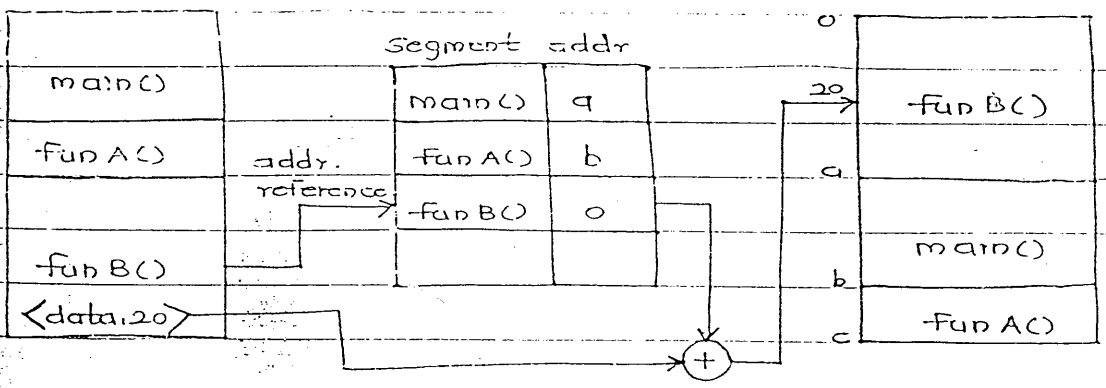


iii. Base & Bound Regr.: Specifies upper & lower limit.



iv. Segmentation:

- logical segments formed (ie based on logic of pgm).
- segments will be diff sizes.
- when the segments are loaded in the Main memory, they may occupy out of order locns

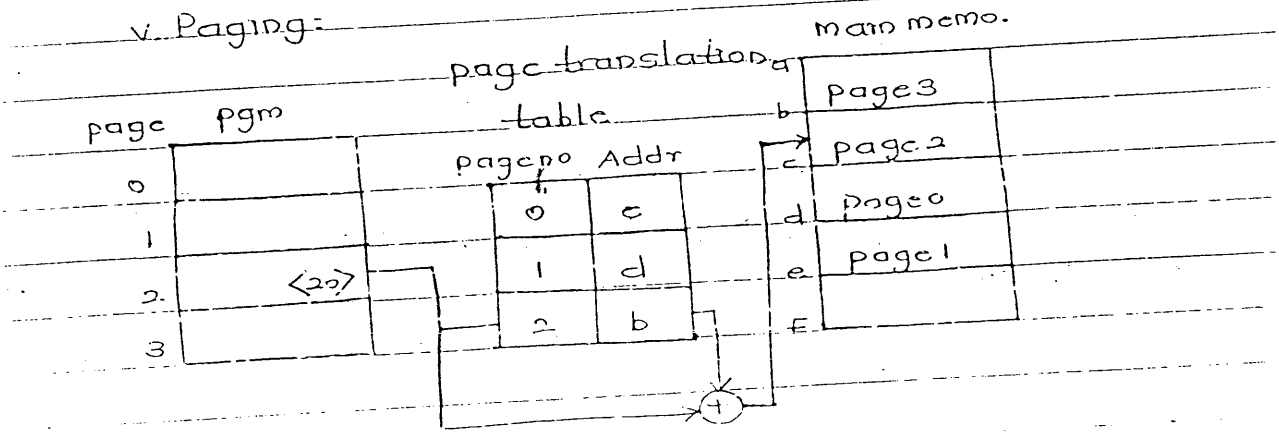


Adv: * Protection Mechanism:

1. Each addr reference is checked for access ctrl
   ie: whenever addr reference made, the OS
   can check for the access rights of the
   user
2. 2 or more users can share same segment

Disadv: fragmentation, bcoz segments are of diff sizes.

v. Paging:



Adv: same as segmentation

vi) Tagged Arch.

→ Tagged bits are used which contains an extra info^n about access rights of the mem. word for each

eg

Tag	mem. word
R	110
W	1210
X	0115
RW	1890
N	2150

Adv: Efficiency ↑

disadv: - HW cost ↑
- Cost ↑

Network Security.

1. Threats to network
2. N/w security controls.
3. Firewalls.
4. Intrusion detection sys.
5. Secure email.
6. IPSec
   → SSL (Secure socket layer).

→ 1. Threats to N/w:

a. Networks are vulnerable:

i. Anonymity: Attacker is anonymous, difficult to locate the attacker.

ii. Complexity of N/w: Huge N/w.

iii. unknown path: Many paths thru' which data pkts travel.

b. Motives:

i) Challenge:

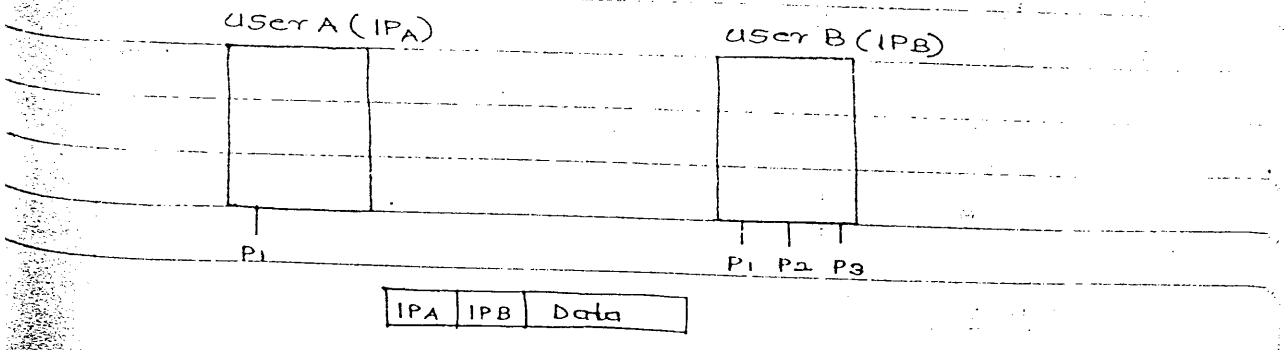
ii) Money:

iii) Fame iv) Ideology

c. Threat precursors: (Prior to attack).

- Before the actual attack, attacker tries to gain some info. about the system.

1. Port Scan:



- Process P_1 of user 'A' wants to transmit data to process P_1 of user 'B'. However, the pkt contains only the IP address, hence user B cannot determine whether the pkt send is for process P_1 or P_2 or P_3.
- to avoid such confusion, port nos are assigned to every process

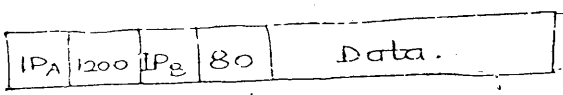
Port No: 0 – 1024 – used for particular/ Specific processed.

eg: port 80 – http
25 – email

- Now, IP addr & port addr are combined and then transmitted.

IP addr + Port Socket addr = Socket addr

- Hence, the pkt send is –



Socket addr.

--- Assume, port No of P_1 of user A = 1200, wants to comm. with webpages (ie: http = 80)

IP addr: user to user comm?
Socket addr: process to process comm?

* Port Scan:
- Attacker sends diff. msgs to port to find which process is running on a port.
- is a precursor, since attacker just finds out the process running, on which an attacker may plan an attack later.

ii. Social Engineering:

- attacker tries to obtain info from user. Can be telephone or personal meeting)
- on telephone attacker may pretend to be administrator or maintenance officer.

iii. Reconnaissance:

a. Dumpsterdiving: eg: Attackers refers the deleted items (Recycle Bin) to gain some info.

b. Eavesdropping:

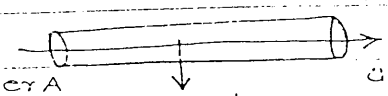
iv. Bulletin Boards: Attacker tries gaining info by communicating users thru' discussion forums.

v. Check for Available Documentation: manual or online.

→ Threats in transit:

1. wiretapping:

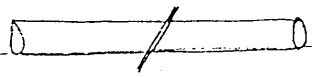
a. Cable:



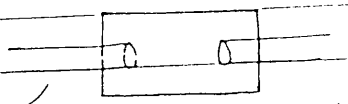
Inductance Method.

- confidentiality not preserved but Integrity is preserved.

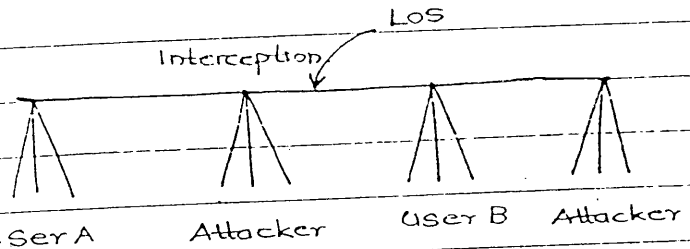
b. fibre optic:



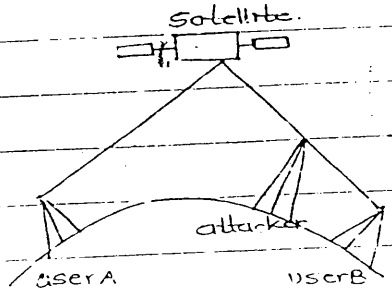
- Data pass thru' Glass, in form of light.
- is not 100% secure, Splicing → may be cause of leakage



c. Microwave:

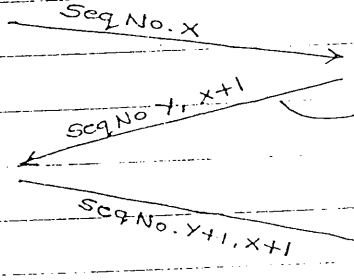


d. Satellite:



e. Protocol Flaws:

TCP: A B (3 way handshaking)



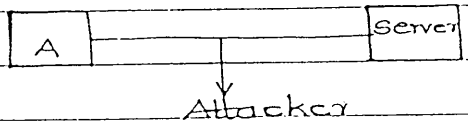
Attacker may attack the msg. & send the msg repeatedly, corr. the user A will continue sending the reply.

- Flaw: Replay ~~—~~ resulting into N/w ~~send~~ congestion.

-f. Impersonation: Attacker try to break authentication

i. Authentication failed by guessing = Attack guess pswd

ii. Authentication failed by wiretapping.



= while pswd was being verified (passed on comm^n line), attacker stole the pswd.

iii. Non-Existent Authentication = sys. may not have authentication.

eg: If username is administrator, no pswd, direct access

iv. Trusted authent^n =

eg: UNIX maintains file/directory of trusted users called 'rhosts'. Such users are not asked pswds.

→ g. Spoofing:

i. Masquerade

ii. Session-hijacking

iii. Man-in-the-middle attack.

---

i. Masquerade: Attacker pretends to be a particular user.

eg: orginal Bank website: www.ABCbank.com

Attacker creates new website: www.ABC-bank.com with similar appearance.

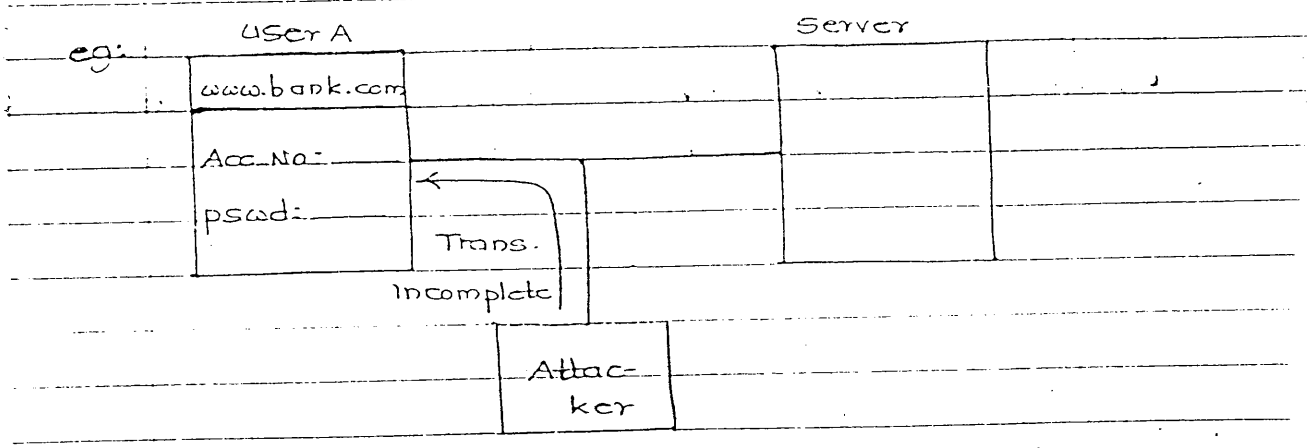
If a user enters to the fake site, attacker can get the account no. & balance details.

---

ii. Session-hijacking:

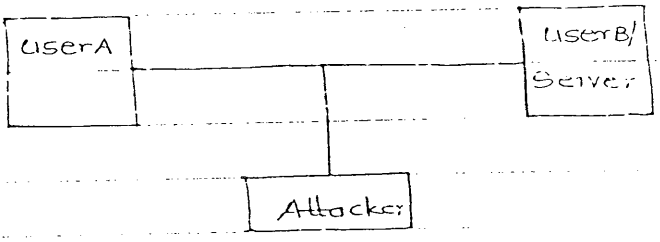
= comm^n b/w user & server forms a session. Attacker interferes only after session is established.

eg:



- Attacker enters a session, hijacks the data/ info. from send by user A to server & sends an error msg 'Transaction Incomplete', then, User A tries to re-enter the details.

iii. Man-in-the-middle attack:



- Attacker present even before the session is established betn the users.
- During development of session, the users exchange some keys (infact before session development), attacker may steal the keys as well.

→ b. message confidentiality threats:

i. Misdelivery of msg: Msg delivered to someone else than the intended receipent.

ii. Exposure of msg: Msg can be seen by attacker, by cryptanalysis or brute force attack.

iii. Traffic flow analysis: Attacker first finds out the sender & receiver & then analyze the msg.

→ I. Msg Integrity Threats:

i. Falsification of Msg: Modification of msg.

ii. Noise: on the comm^{n} line.

→ j. Websit: Defacement:

i. Buffer overflow.

ii. Incomplete Mediation:

→ k. Denial of Service: (broken ~~congestion~~ comm^{n} line, congestion)

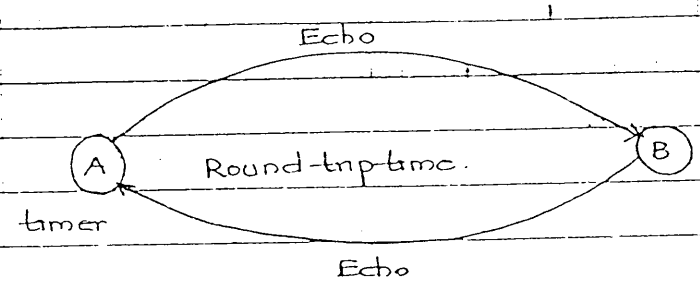
i. Transmission failure: broken comm^{n} line.

ii. Congestion (Comm^{n} flooding): Attacker sends unwanted pkts on comm^{n} line.

* Types of packets:
  - Echo chargen
  - Ping of death
  - synchronization packet

→ Echo charger: Attacker sends echo packets.

*. Use of Echo packet:

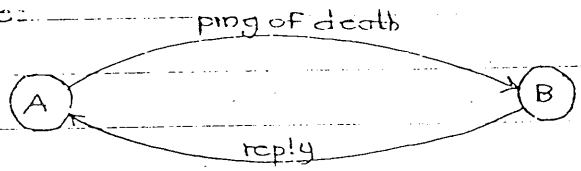


- user A sends echo pkt., user B replies with the echo pkt, user A timer gives the Round trip time.

- Attacker steals the echo pkts & congest the line by sending it repeatedly.

→ Ping of death:

*. use:



- user A sends ping of death pkt to B, B replies back implies B is still alive.

- Attacker congest N/w by sending the pkt repeatedly.

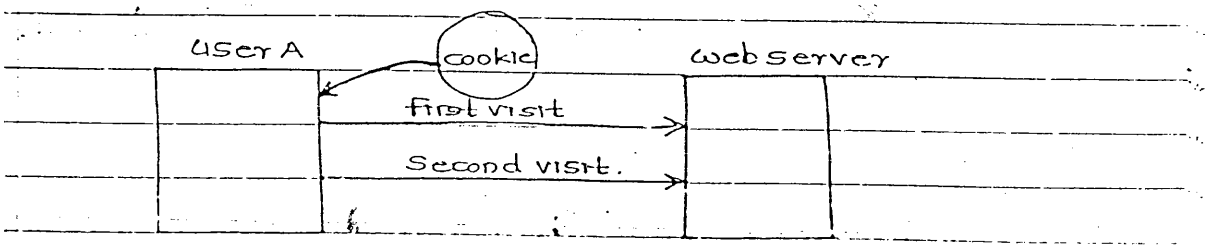
→ syn packet: Attacker captures the reply pkt & sends it continuously to the Rx, Rx continuously

sends the acknowledgement (in 3 way handshaking).

➢ 1. Distributed Systems:

- Attacker installs (virus) code in all the nodes of the distributed system. The code is activated thru' a single terminal and at same time. Thus, entire sys is attacked simultaneously. eg: time logic bomb.

➢ 2. Threat to Active code: moving code = cookies



- webserver stores user info in form of cookies, on the client M/c.

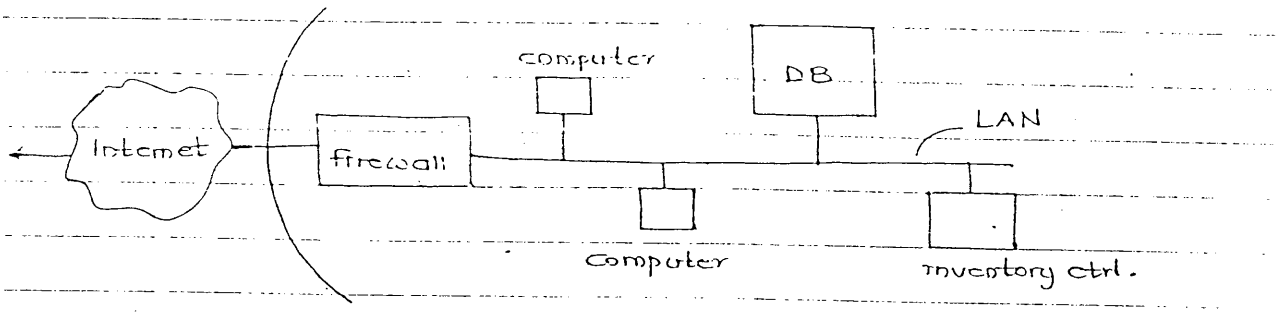
- loss of confidentiality

→ 2. N/w Security controls:

i. N/w threat analysis: Analyze possible threats

ii. Architecture:

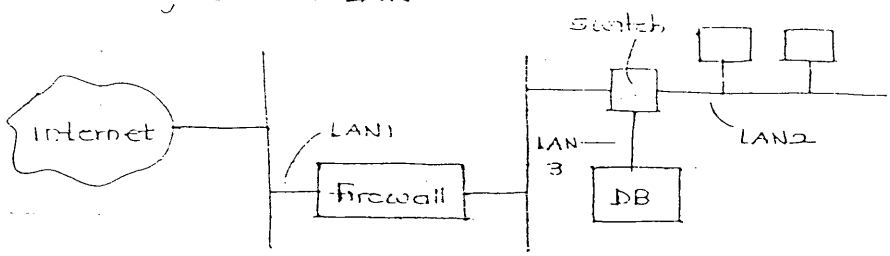
a. Segmentation:



System

- Single pt. of failure: If ~~failure~~ attack is made on this point, entire sys. may fail.
- Here, entire sys. is on same segment (LAN), hence attack on the LAN will allow attacker to access all the devices/nodes connected to it.

* Segmented LAN:



b. Redundancy:

→ Duplication of data.

→ Multiple paths.

III. Encryption:

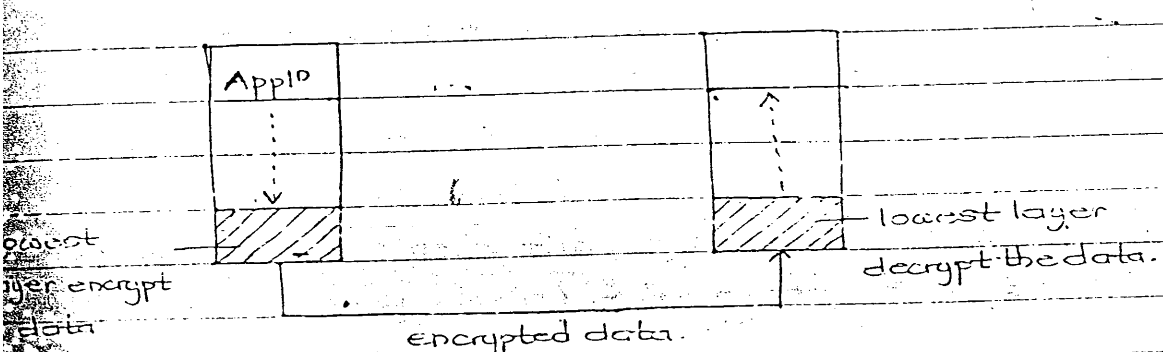
a. Link encryption.

b. End to End encryption.

c. V.P.N (virtual private N/ω).

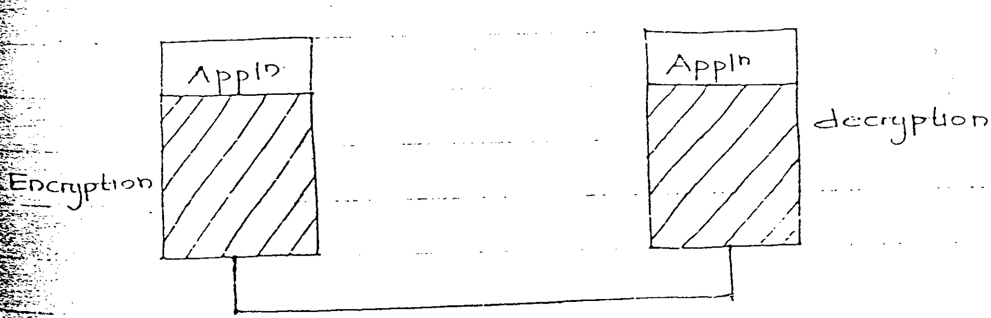
d. PKI (public key infrastructure) & certificates

a. Link encryption:



- Encryption done only on the commn link. If attacker attacks user terminal, attacker gets plaintext.

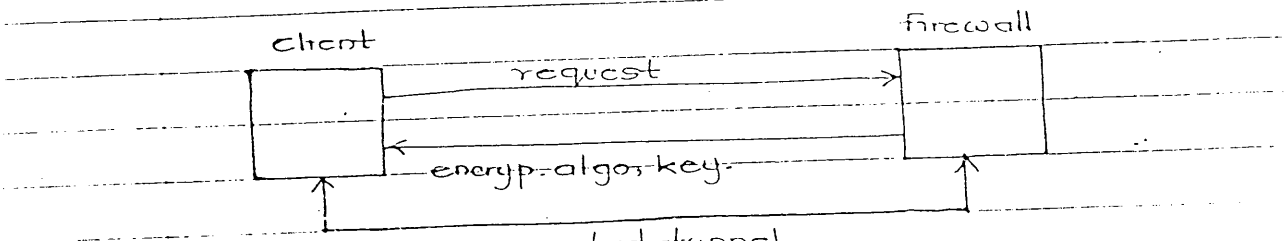
b. End to End Encryption:



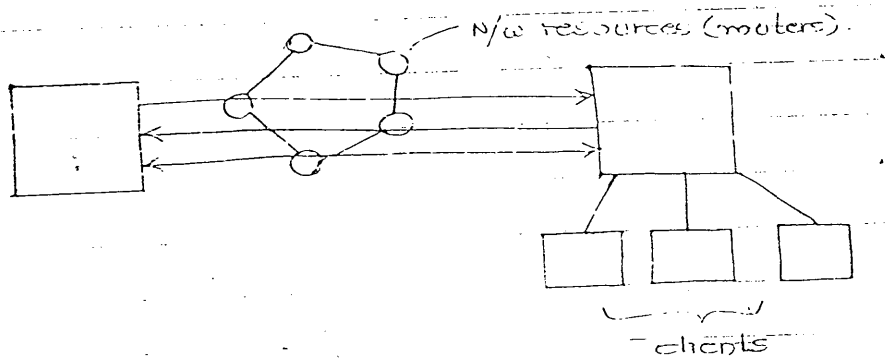
- In link encryption, user is not aware that data is encrypted. (Data encrypted by router / Networking device)
- In end to end encryp, user aware of encryption. (Data encrypted by users).

Comparison betn Link & End to end encryption

5. V.P.N:



- Client wants firewalls to give a secure channel.
- Firewall decides & sends algo & key's.
- Henceforth, any comm^n is thru' encrypted tunnel.
- use networking resources like Routers, also the firewall is connected to diff. clients — however, to client it appears as if a private N/w.



* - Follow link-encryption, since comm^n betn user & networking device.

d. PKI & certificates:
key distribution done thru' certificate authority.

GQ 18. Ans: To avoid single pt of Farlure.
(Explain archi. without segmentation & then with segmentation).

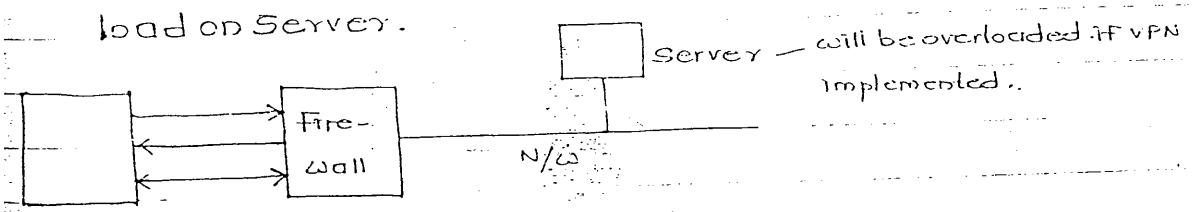
50. May 2005
7(b)

(a) 11. End to end & link encryption can be used on same comm^n for data that is highly confidential.

⇒ End to end encryption will prevent attack on terminal & link encryption will further encrypt already encrypted data. on the other side, data will be first decrypted by networking device & passed to user. user will further decrypt it to get actual data. If attacker attacks receiver end, plaintext is not available.

12. VPN uses link encrypt^n as encryption is between networking devices.

13. A VPN is implemented using a firewall & not actual server to reduce server load or to relief server from this job. Since, no. of clients are more & if every client wants to have a VPN, firewall needs to maintain corresponding decryption algo. & keys. If this job is shifted to server, it will cause extra load on server.



Client makes a request to firewall for providing private N/w, if firewall is already loaded, it may simply deny that request.

iv. Content Integrity Controls:

a. Error correction codes: using parity bits

b. Checksums: MD5, SHA

v. Access control: check access rights of users

- Access control list on
  - Router
  - Firewalls.

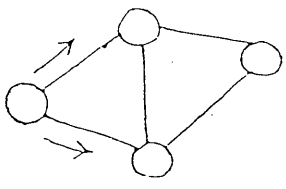
* Router:

- whenever data comes to router, it checks the access rights & then transmits.

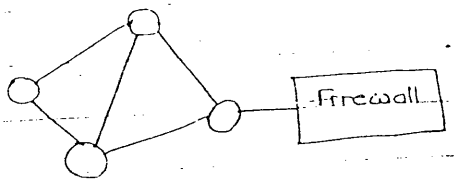
Disadv: Performance ↓ of N/w.

* Firewalls:

- all access checked at one point, hence performance will not degrade.



Router



firewall.

vii. Strong Authentication:

1. Challenge Response sys:

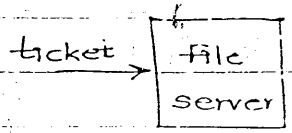
ii. kerberos:

Authentication Server (AS) (Sometimes called as kerberos server).

KS – TGS

① (Shared Secret key)

Ticket Granting Server (TGS)



TGS

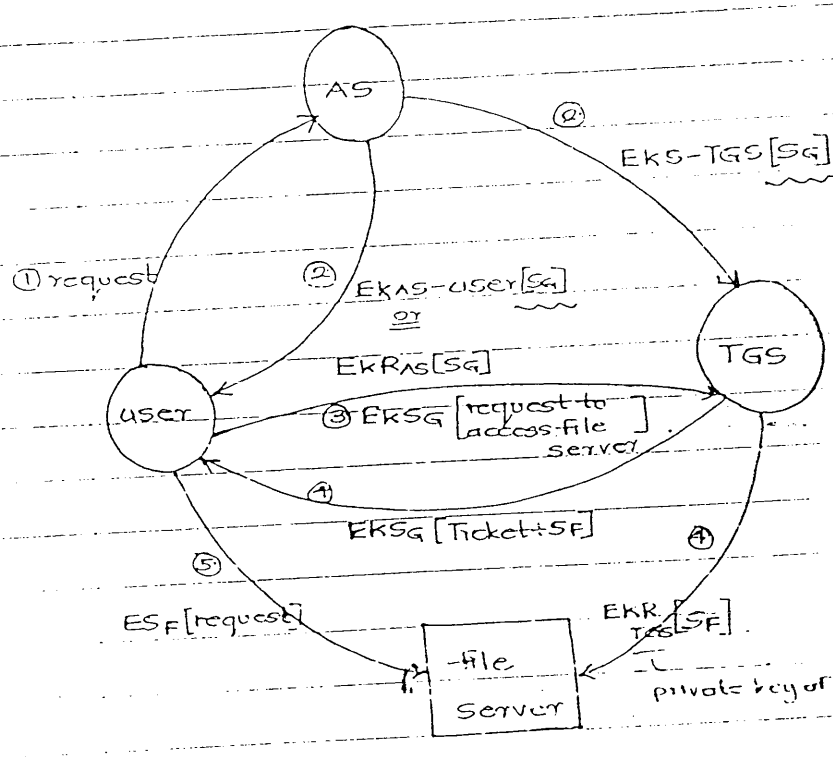
② SG

user

③ SF

file server.

They communicate using key called as shared secret key. user wants to communicate with file server. First, AS checks authenticity of user. If it is an authenticated user, AS tells TGS to grant ticket which is some kind of token. Once a user gets a ticket, user can access any file.

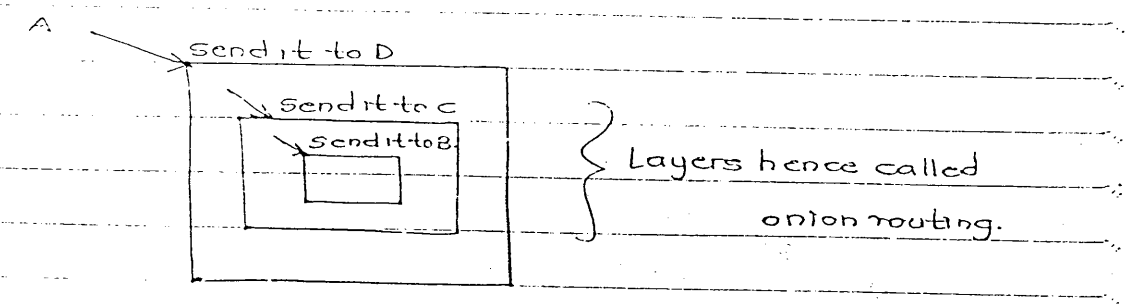


- SG: secret key generated by AS & shared with user & TGS.
- SF: shared secret key betn user & file server.

Step1:

vii) Traffic flow control: we want to hide the src & destn.
- onion routing.

eg: A wants to send pkt to B (A → B).



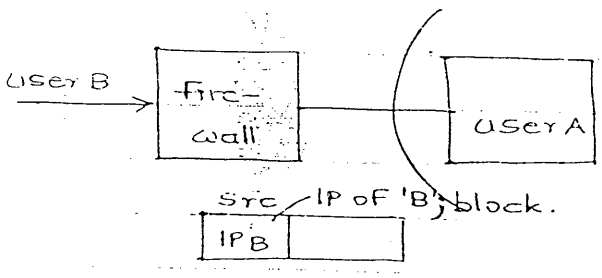
- Attacker thinks that 'A' is sending pkts to 'D'

viii) Firewalls: ctrl incoming & outgoing traffic (ID N/w).

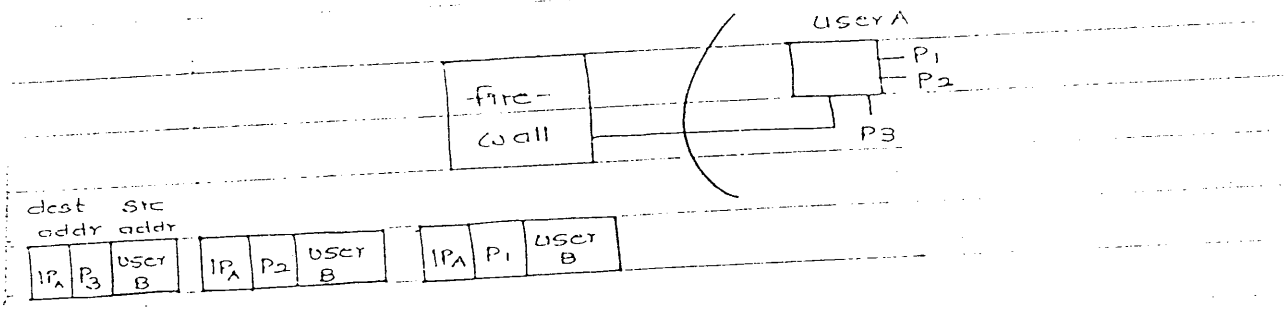
a. Types of firewalls:

1. Packet filtering firewall: Firewall blocks pkt depending on the IP address ; contents is immaterial.

eg: If user A do not want to receive data from B, firewall checks the IP addr, if IP is of B, block.



* Purpose:
1. Block traffic from all users to a particular user.
2. To prevent port scanning.



- Firewall finds that the same user sending pkts to all the ports (ie: scanning the ports); firewall discards the pkts.

Disadv: 1. It does not check the data, hence, virus code can attack/ enter the N/w.

II. Stateful Inspection firewall:

- Sometimes attacker does not send the virus code heavily thru' one pkt. he may distribute it in diff. pkts; so that the virus pattern go and tested.
- The virus is activated when all the pkts are combined.
- If firewall checks one pkt, virus pattern go undetected.
- Stateful Inspect? firewall combines some packets and checks/ inspects for virus signature.
- The scanner in firewall will map the signature with the signatures in the database.

Adv: 1. Even though virus code is distributed, it will be detected.

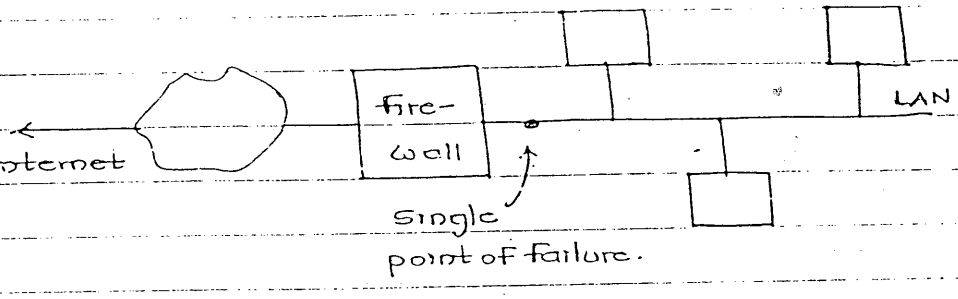
Disadv: 1. Since the firewall has to store the pkt, some extra storage space is reqd.

2. Virus signatures not in the DB go undetected.

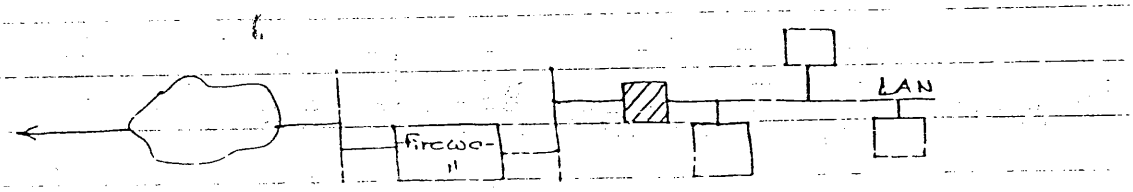
iii. AppID Proxy firewall: Made as per user requirements eg: organization wants to block certain websites or dont want to pass certain info on N/w.

b. Firewall Configuration:

i.

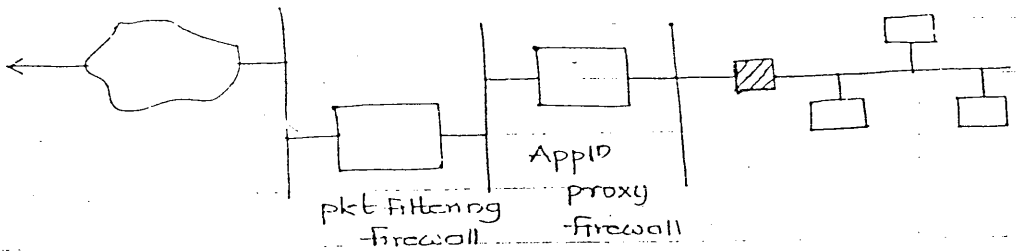


ii.



- If Attacker attacks firewall, data on the LAN will not be affected.

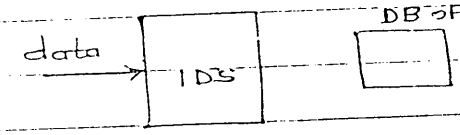
iii.



→ 4. Intrusion detection sys (IDS) - checks for virus.

a. Types of IDS:

i. Signature based IDS:



- Individual pkts are examined for signatures.

Disadv: 1. New virus goes undetected

2. If virus code is send discretely, virus goes undetected

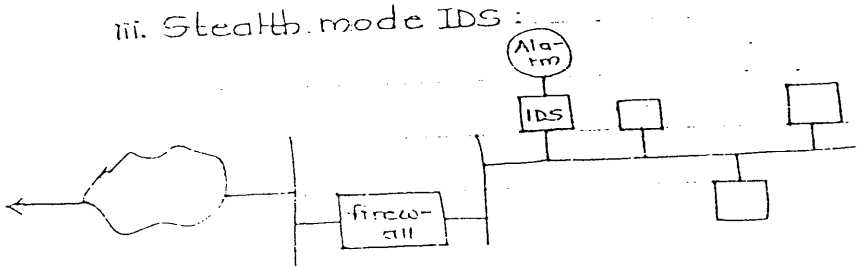
ii. Heuristic based IDS:

- Algo. uses its judgement to detect/find virus.
- Trial & Error method.

Adv: 1. New virus can also be detected.

Disadv: 1. Since it is heuristic algo.; a non-virus code can be judged or considered as a virus code.

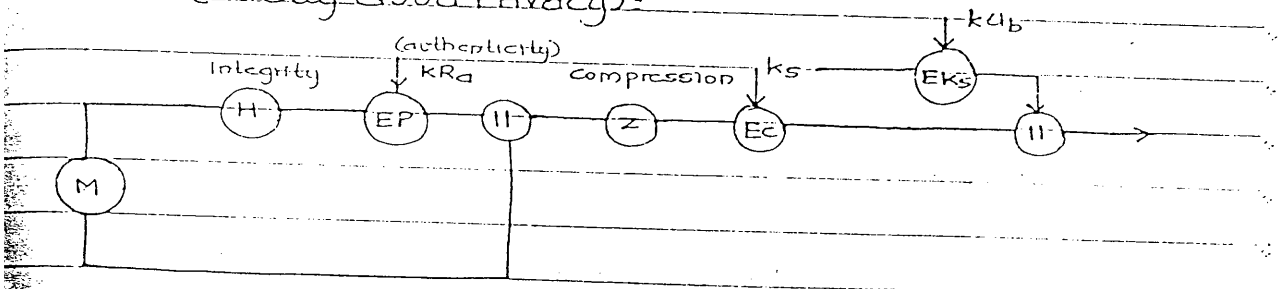
iii. Stealth mode IDS:



- An alarm is connected to IDS, so that everyone gets to know (the steal of data).

5. Secure Email:

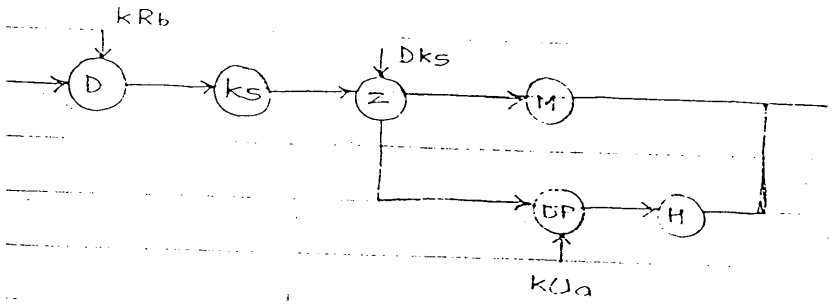
PGP (Pretty Good Privacy):



$E_{kUb}[ks] \,||\, E_{ks}[z[M \,||\, E_{kRa}(H)]]$

- M: msg ; H: Hash
- EP: encrypted plaintext ; Z: compression
- Ec: encrypt compressed message

→ At Rx, 'ks' is obtained by using private key of B, then the remaining msg is to be decrypted using ks.



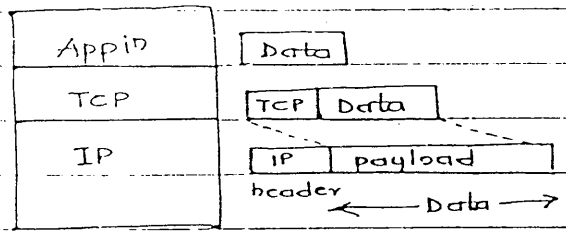
$E_{ks}(M)$

$E_{kUb}(KS)$

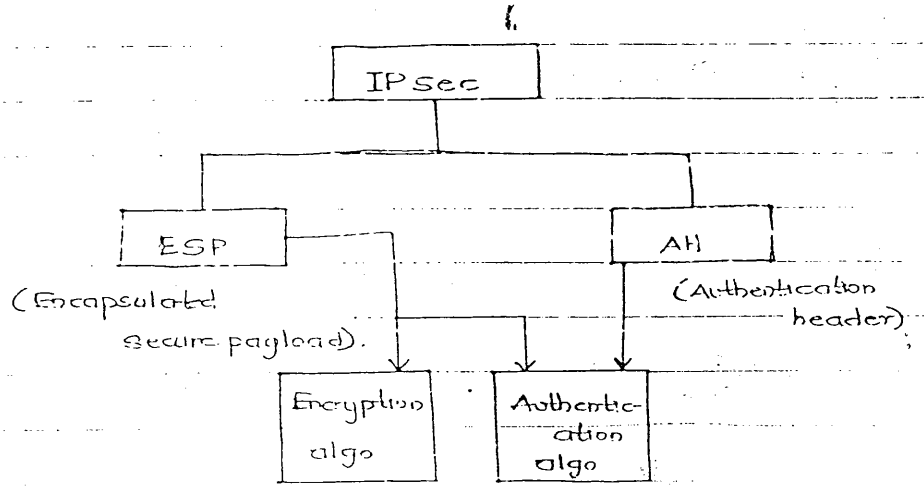
$E_{kRA}(H)$

→ 6. IPSec: Secure IP pkts

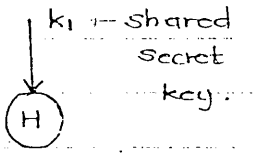
- IP = connectionless service, so pkts may follow diff. routes.



a. Architecture:



* HMAC (Hash Msg Authentication code)
- code generated using hash, uses shared secret key.
- Integrity + Authentication.



→ * Security Asso. (SA): It consist of certain parameters that should be discussed betn users before actual pkts or msg is passed.

1. Sequence no. counter: The sender sets the counter value, so that the receiver knows the no. of the pkt that the sender is sending.

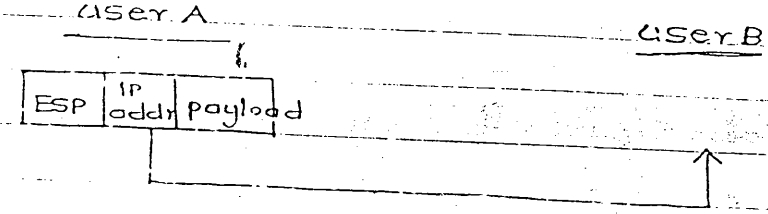
2. ESP info.: - Encryption algo.
- HMAC < MD5 / SHA

3. AH info.: - HMAC

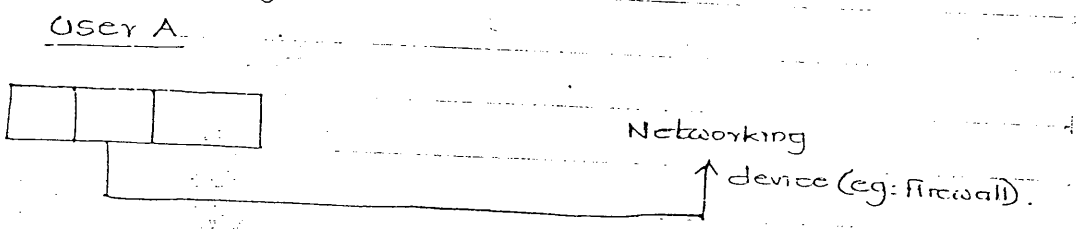
4. IP Security mode: - Transport mode
- Tunnel mode.

- Transport mode: b/w users.

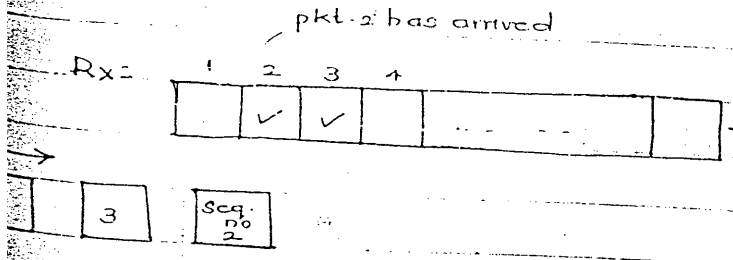
The security set by user A is seen/checked by user B.



- Tunnel mode: Encrypted data by user is decrypted by any networking device.



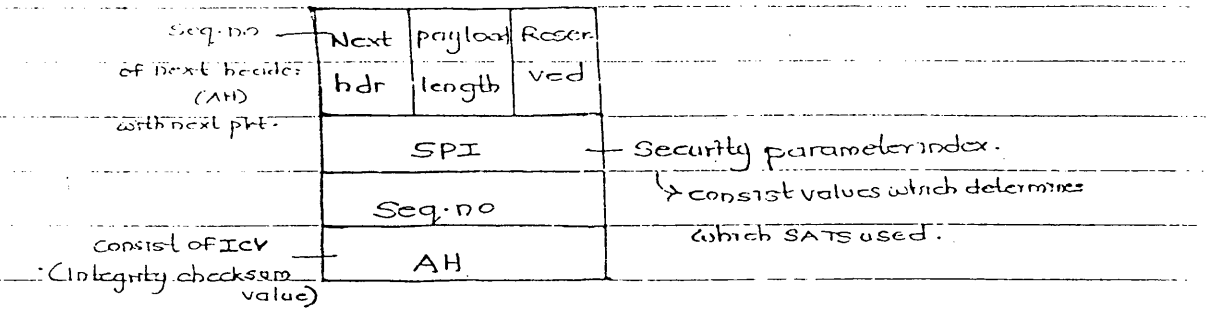
5. Anti-replay window: window size is communicated.



- If attacker sends the pkt over & over again, Rx knows it's a replay (hence - antireplay window).

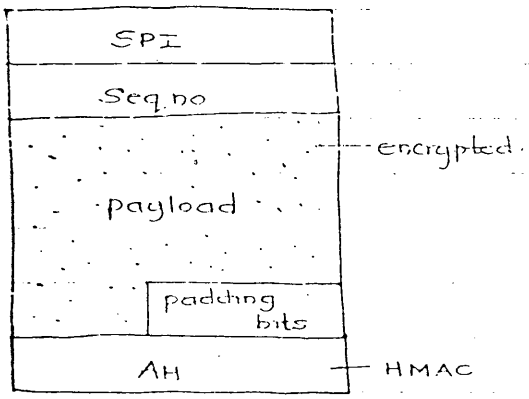
6. Life time of security asso:- for how long the parameters of SA are valid.

* AH:



- hash value either MD5 or SHA.
- HMAC

* ESP header:

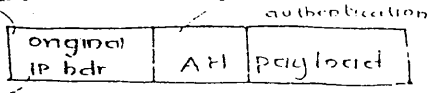


- encrypted data present in the header.

* Transport mode:

IP addr. kept outside, because routers see the IP addr then route a pkt.

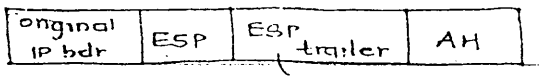
used for authentication



— AH

i. consist of src & dest IP

ii.



— ESP

delimiter.

* Tunnel Mode:

2.

New IP hdr	AH	original IP addr	payload

AH

IP addr. of networking device. (eg: firewall)

data for Rx.

II.

New IP hdr	ESP	original IP hdr	ESP trailer	AH

1. To have security in connectionless service.
2. Anti replay window prevents replay attack, also this window can be used to accumulate the msg & transmit to upper layer (TCP) at Rx end.

~~IP comes out of order at~~

-- pkts comes out-of-order at IP (: connectionless service), but IP sends pkts in-order to TCP (conn? oriented service).

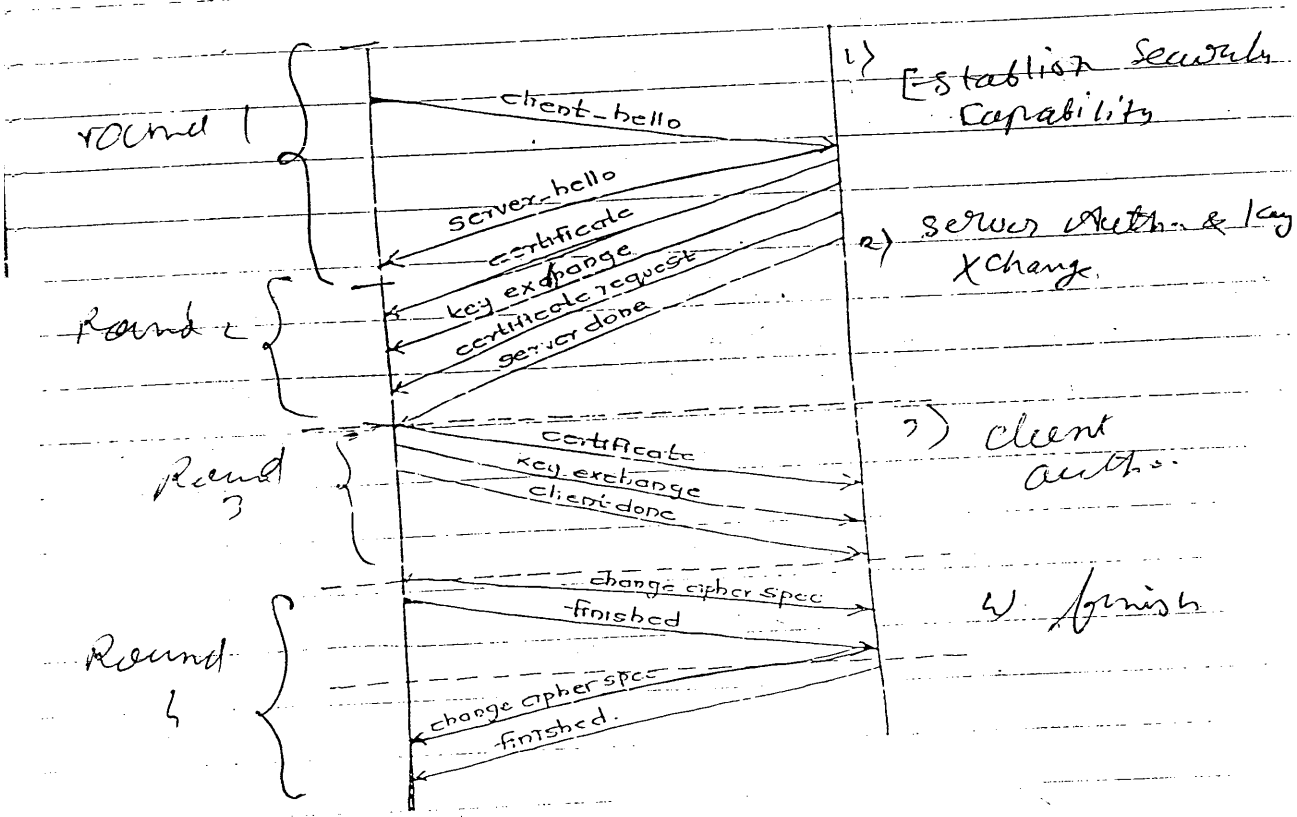
3. Security:
   - confidentiality: (ESP) - encryption
   - authentication: MAC
   - Integrity: Hash.
   - Prevents replay attack.

7. SSL (Secure Socket layer): Security at transport layer & above.

SSL Handshaking protocol	SSL change cipher specification	SSL alert protocol
SSL Record protocol		
TCP		
IP		

- Security is provided at the time of connection establishment ie: security parameters are exchanged before msg is passed to Rx.

1. SSL Handshaking protocol: used for conn^n establishment
   - sets up para: used for SSL Record prot^n.

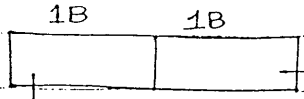


1 Byte	3 Byte	
Type of msg	message length	msg content

protocol

eg: | 1 | — | client-hello | Adv: with type of msg, user can identify the msg coming.

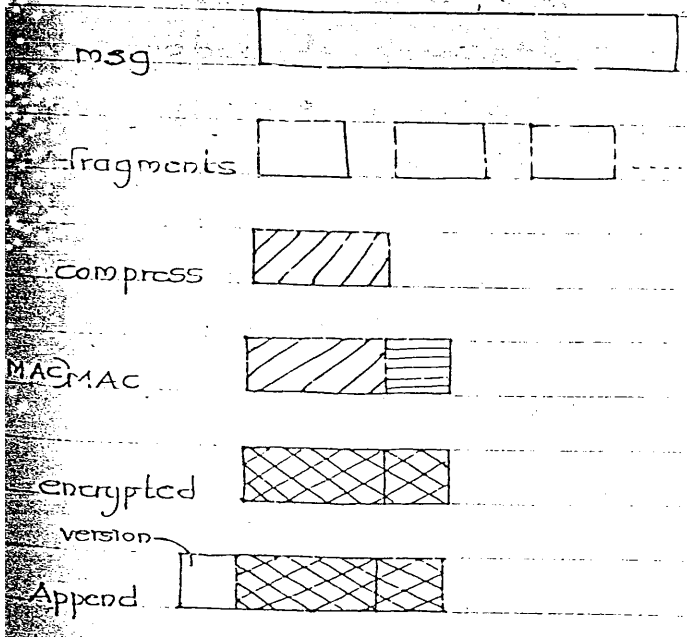
2. SSL Alert protocol: used to send error msg.



- type of error. (eg: misunderstanding of encryption).
- 0 : warning (eg: To slow the sender).
- 1 : Fatal (Severe error) (eg: misunderstanding of encryption)

3. SSL Record Protocol: provides actual security on msg.

- Divide msg into equal size fragments.
- Every fragment will be compressed.
- MAC attached to every compressed fragment.



provide confidentiality & msg integrity for SSL connection

4. SSL change cipher: contains value 1.

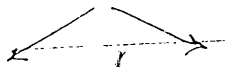
Single byte msg → Causes the state of the state to be changed from pending to current

- Server temporarily stores the contents send by the client, since it is not sure of continuous conn?
- when SSL change cipher send, it indicates that

conn^n is established & server should store the contents permanently.

(GB) 17. IP security implements Antireplay window which is used to prevent replay attack. This window keeps track of packets as per the sequence no.

18. ACL



router firewall

Situation where ACL is on Router :- Echo packets and ping of death packets can be checked by router so that if these pkts are for the same receiver, router will discard them. This requires checking of packets (addr of packet & type of pkt) by router which will slow down the speed or N/w performance is affected. This is tolerable if N/w capacity is loo. i.e. we are compromising on speed to prevent N/w congestion.

q. onion routing.

* DB is a collection of data & set of rules that organizes the data by specifying certain relationship among the data.

॥श्री गणेश॥

Database Security

- phy. DB Integrity
- logical DB "
- element integrity
- Authentication
- Auditability.
- Access control
- Availability

Database Security:

1. Security requirements.
2. Reliability & Integrity.
3. Sensitive data
4. Inference.
5. Multilevel DBS.

1. Reliability & Integrity: Methods to keep DB in consistent state.

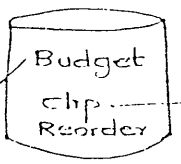
a. Two phase update:

- Intent phase : perform transc. (Collect infor)
- Commit phase.: update the DB.

eg:

organ. has to supply stationery materials to all the dept. org. maintains budget for each dept.
Account dept. asks for 50 clips (say). Hence, 50 clips will be substracted from database value of clip.
From the budget allocated to account dept, cost of 50 clips will subtracted.

budget = Rs. 500
cost = cost of 50 clips



clip = clip - 50

Budget = Budget - cost.

↳ Intent: perform transactions

1. check if commit flag is set — ie: someone else is updating DB.
   If not set then proceed

2. TCLIPS = ONHAND − REQUISITION.
   (temporary vari. — available — requested)

   TBUDGET = BUDGET − COST.

3. IF clips < level, TREORDER = TRUE; (place order)

↳ commit: update DB

1. Set commit flag.

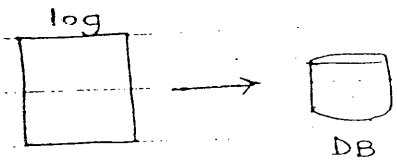
2. CLIPS = TCLIPS (DB vari.)
   BUDGET = TBUDGET
   REORDER = TREORDER

3. upset commit flag.

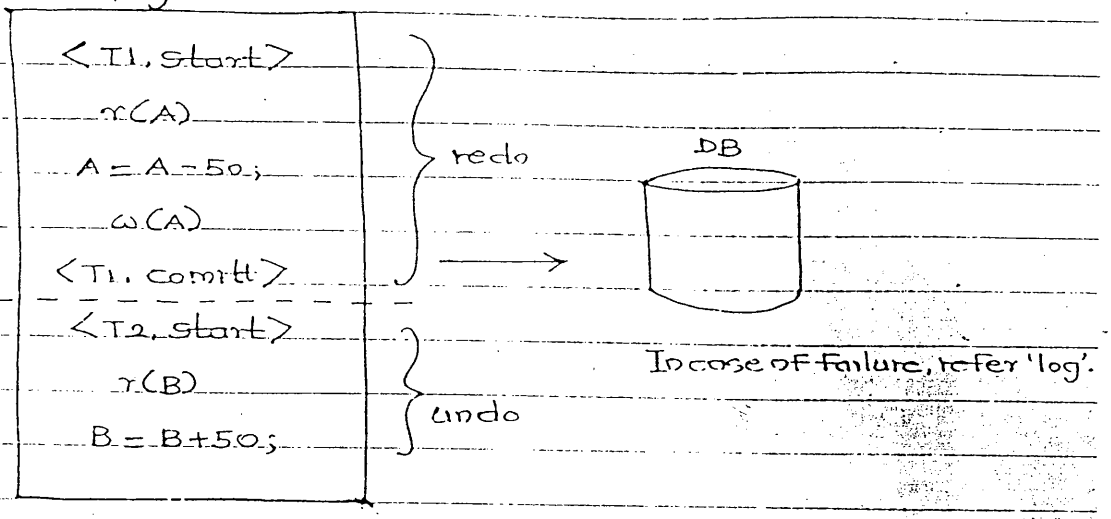
Appln: concurrent transc. sys.

b. Recovery Procedures: Recover failure such that the sys. is in consistent state.

→ log: stores all transc ; the values are then transferred



3. log



- Transc. 'B' is undone, since not committed; B is restored to prev. consistent value.
- Start + commit ⇒ Redo } idempotent
  Start ⇒ undo.

4c. UNDO(x) = x

'x' is already correct value, so UNDO(x) = x

b. UNDO(UNDO(x')) = UNDO(x) = x

Explain UNDO & REDO are idempotent.

Monitors:

i. Range comparisions
ii. State constraints
iii. Transition constraints

i. Range comparisons:

→ monitors implemented in User Interface.

eg: Year [ ] (range: 1800 - 2000).

↳ monitors implemented at coding level.

eg: create table account (
) checkamt>500.

II. State constraint:

- monitor checks state of a var.

eg: Two phase commit → where state of Flag is checked for proceeding.

III. Transition constraint:

eg: vacant=1 {

If new employee is registered,

if vacant=0, then employee DB is entered.

→ 2. Sensitive data: confidential data which we need to protect.

* Types:

a. Inherently sensitive:

eg: military records, profit plans of org.

b. Declared Sensitive:

eg: An artist wants his name to be kept hidden, so the DB contains only Painting No. & Year.

c. From a sensitive source:

eg: Data from military DB.

d. Data sensitive wrt previously disclosed info:

- A data value may not be sensitive, but previously avail. data makes it sensitive.

eg: If latitude is already known (prev. info), & longitude is requested, then actual pos? can be located.

3. Inference:

a. Direct Attack

b. Indirect Attack.

StudentName	Hostel	Sex	FinancialAid
—	A	M	—
—	B	M	—
—	C	F	—
—	C	F	—
—	C	M	—

- Financial info. of one particular student is not revealed.

eg: Query for financial aid of a student will not return a value, however, Query for financial aid for a hostel (i.e. group) is allowed.

- Attacker fires a Group Query & tries to infer some result.

a. Direct Attack:

- Attacker makes a long query so that it goes undetected by the security measure/mechanism.

eg: Select Studentname, Financialaid
from Student
where HOSTEL='A' and SEX='M' or
SEX≠M and SEX≠F or
meaningless info

- Some useless info. is added to query, so that the reqd data is obtained.

b. Indirect Attack: uses groupby.

eg:

	A	B	C
M	3000	2	4000
F	6000	2000	7000
	9000	2000	11,000

- Attacker generates the above table using GroupQuery

	A	B	C
M	2	0	1
F	3	1	4
	5	1	5

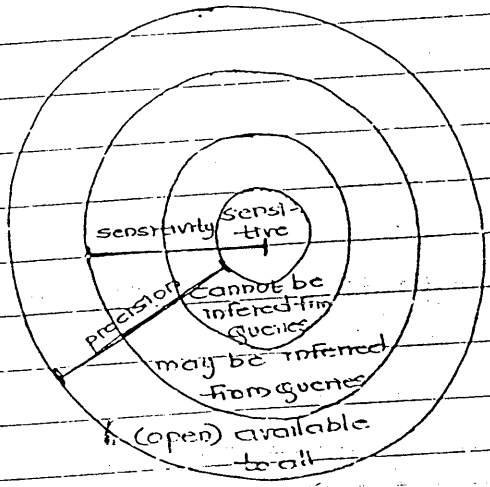
- From the above tables, Attacker infers that-

a. Hostel B contains only 1 female with Rs. 200/- as financial Aid.

b. Hostel C contains 1 male with Rs. 4000/- as Finan. Ai

Sensitive (admin data)

→ Sensitivity vs Precision:
↳ protect data. ↳ more & more data should be available to user



→ Multilevel DBs:

	E-code	Name	Dept	ph.no_no	Salary (sensitive)	performance-grade (sensitive)
(sensitive data)	101		admin			
	102		admin			
	103		admin			

- diff. parts of DB has diff. security requirements. or certain parts of DB are sensitive ie: security at diff. levels.

* Methods of implementing Security in Multilevel DB:

a. Partitioning: Data with same security requirements are separated from others.

eg: 1. (Ecode, Salary, perf_grade)
2. data of administrators.

1) Redundancy
2) Inefficient use of mem. Space (disk Space
3) updating same field at diff. location

Disadv: 1. We form small parts or partitions, the very purpose of DB is defeated. It is as good as traditional file sys.

2. (Similar as file sys). Data may be in inconsistent st

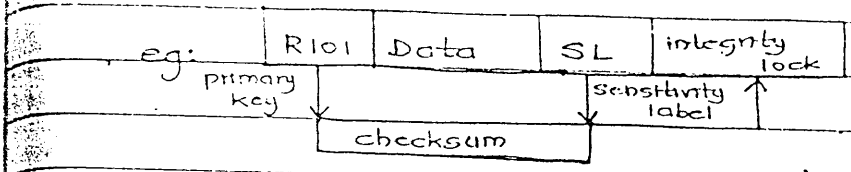
b. Encryption:
- encrypted the data to be protected. ie: encrypt sensitive records.
- Data items with same security requirement encrypted using same key.
- Sensitive within bounds is accepted, however, exact value of sensitive data is not accepted.

1. Select emp_name
   From employee
   where salary > 20,000 and < 10,000
   } accepted

2. Select emp_name
   from employee
   where salary = 10,000
   } rejected.

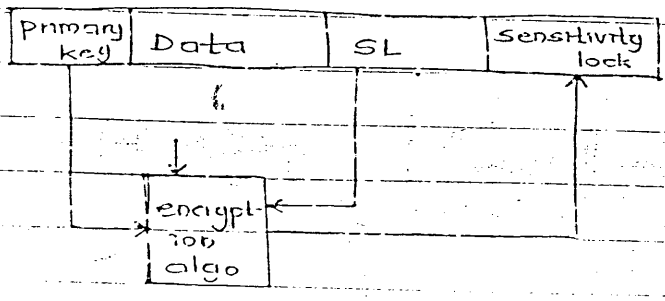
Disadv: 1. Disclosure of sensitive data within bound is accepted
2. Query Processing is slow, since, before the proce of query, data is to be decrypted.

c. Integrity lock: used to preserve integrity of sensi records.
- sensitive data is stored with checksum.



- SL: mark for one type of sensitive data.
eg: administrator data is given SL = A1.
- Authenticated modifies data as well as checksum.
- Attacker can only modify data.

d. Sensitivity lock:



- encrypted checksum is stored as sensitivity lock.
- In both the above cases, confidentiality is not preserved, only integrity is preserved.

Designs for Multilevel DBs:

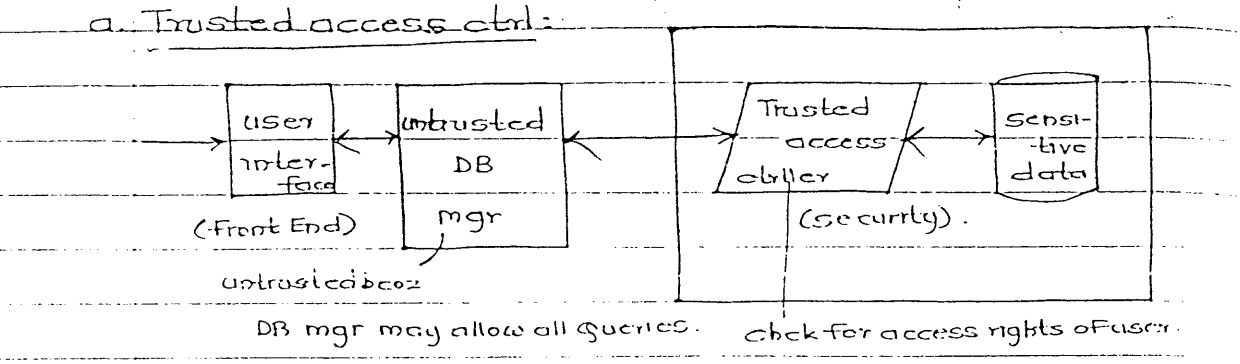
a. Trusted access ctrl.

b. Trusted front End

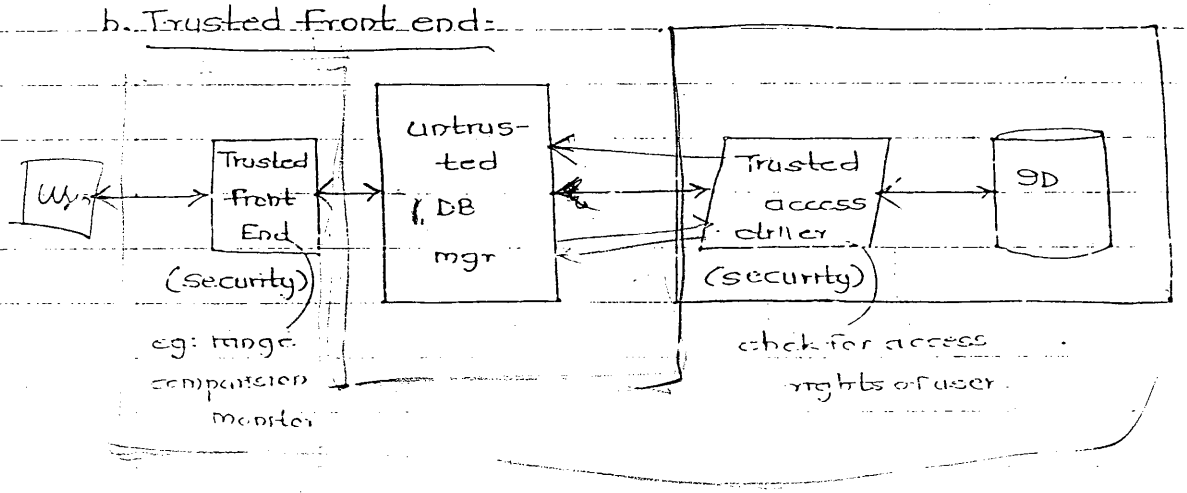
c. Commutative filters.

d. Windows/views.

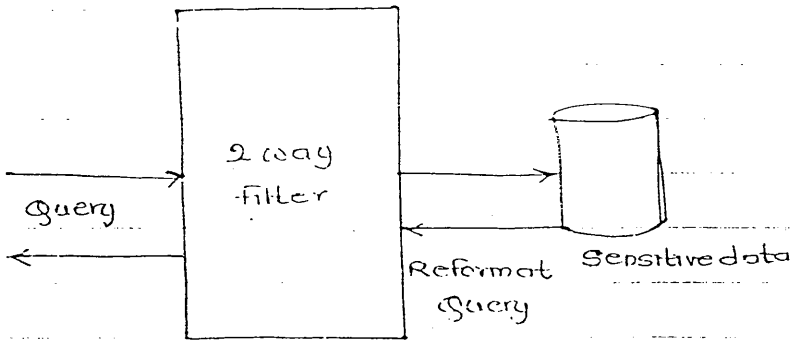
a. Trusted access ctrl:



b. Trusted front end:



c. Commutative filters:



eg: if query fired by user says name of empl. wrtb salary is 10,000. filter reformats it to - salary betn 10,000 - 20,000. It informs user of the data to be displayed.

Adv: 1. preserving sensitive data. Rejects some part of query

2. preserve precision (Some part of Query is accepted):

d. window/view:

- DB is kept at some place, some view/window is visible to user.

eg:

Airways	Departure	Arrival	Pilot	Staff

useful to passenger — useless for passenger.

5. Security Requirements:

a. Physical DB Integrity:

- losses related to physical assets
- Remedy: Backup

b. logical integrity DB:

- losses due to transactions.
- Remedy: Recovery mechanism using Log.

c. Element DB Integrity:

- values of elements should be modified/invalidated
- Remedy: Monitor → range comparison.

d. User Authentication:

- Remedy: password.

e. Access ctrl (Authorization):

keeping records of access rights.

V.N.A

: Authentication: valid / authenticated user should access the data

- Done to enter the sys / data.

: Authorization: Is user authorized to use a particular data or file.

- Done after a user enter sys

f. Auditability:

- store transc & user info in Audit log.

Adv: - Attacker can be traced

g. Availability: 1.

- prcasion.

॥ श्री साई ॥

Administering Security

1. Security Planning } decide which assests requires
2. Risk Analysis } security.
3. organizational Security Policy
4. Physical Security.

- org. first decides the assets that needs to be protected & then design controls

→ 1. Security Planning:
   a. Contents of Security Plan.
   b. Business continuity plan.
   c. Incident Response plan.

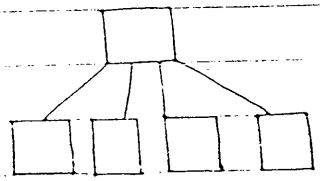
a. Contents of Security Plan: Changes to be implemented in future.
- may or may not be implemented, depends on organi resources.

i. Purpose:
   - goals of org
   - responsibilities (at what level).

- goal eg: To give max. data to user or to prevent sensitive data availability.
- ie: For some org. security is of prime concern, For some customer requirements are imp.

- responsibilities distribution: Decide method to distribute responsibility.
- eg: a manager assigned to N/w security.

- Some org. may keep responsibilities under management level while some under lower levels.



II. Current Security Status:

III. Requirements: What all is needed in the sys.
eg: a) DB, OS, pgm → may already be implemented
N/w security is reqd.
b) security for a particular dept.

* Characteristics:
Requirements:
- Correctness
- Completeness
- Consistency (no conflicting requirements).
- Verifiability (whether requirement can be fulfille
- Realism. (implementation should be possi).

IV. Recommended Controls: which controls are recommended for security.
eg: For N/w security requirement, controls can be firewall or IDS.
- implementation of these controls depends on the org. resources.

V. Accountability: Assigning responsibilities to individual
(actual assigning of responsibilities to mgr or subord

vi. Timetable: Schedule for how the plan will be implemented. eg: Gantt charts.

b. Business Continuity Plan: Plan is prepared to overcome catastrophic conditions like natural calamities (Flood, fire)

- used for Rare catastrophic failures.

i. Identify Valuable Assests:

ii. Develop plan: If catastrophic cond^n occurs, who will be given the responsibilty, who will decide the line of action.

c. Incident Response Plan:

- If there is an incident, who will be given responsibility to take line of action.
- Incident could be any small incident like failure.
- used for all types of failure.

a. Current Security Status: Current Security Status is considered so that new plan can be made to implement security that is currently not available.

b. Resources of org: Here we consider resources in terms of money available. Also, manpower resources are considered so that responsibilities can be assigned accordingly.

c. Future technology: While preparing plan, we should consider the possi. Future technology so that our security system can be enhanced in future.

2. a. Security requirement that is not realistic:

- To hide all data (since, min. data has to be communicated within org. & some data has to be made avail. to user).

b. Security requ. that is not verifiable:

= Reliability on outside N/w. (outside N/w is very complex, there is no check on it).

c. Security requ. that are inconsistent:

- i) Data Access to users Vs Sensitive data.

We want data to be avail. to user, at the same time we want to protect data = conflicting.

- ii) Security Vs Speed.

In order to keep comm? fast, we may have min. encryption. This will make data more prone to attacks as data will be passed on N/w or access will be thro' N/w.

→ 2. Risk Analysis:

- first, find the possi. risks, then their control & finally decide whether the controls be implemented or not.

→ *. Defn:

i. Risk Impact: Loss occured due to an event.

- Effect of risk on sys.

ii. Probability of Risk: chances of a particular risk occurence.

iii. Risk Exposure = Risk Impact * Probability.
(RE).

iv. Risk Reduction : controls to reduce risk.

v. Risk Leverage :

* Steps in Risk Analysis :

a. Identify assets : Assets where risk is possi.
- H/w
- S/w
- Data
- People

b. Identify vulnerabilities : which are the possible attacks on the assets identified.

Assets	Secrecy	Integrity	Availability
H/w			Theft
S/w			lost
Data	disclosed	damaged	destroyed.
people			

c. Calc. Probability :
- Delphi Approach
- frequency test.

Freq. Test : prob. judged on basis of past data / statistical data.

Delphi approach : expert opinion + statistical data.

d. Expected Loss:

e. Recommended controls:

* f. Project Savings:-

eg:- Risk: Loss of confidential data.

So, the org went for purchasing access ctrl S/ω which cost RS. 25,000/- with 60% effectiveness.

(i) - If the ctrl S/ω is not implemented, confidential data will be lost. So, org has to reconstruct the data →

- cost of reconstruction of data: 10,00,000 @ 10%.

Hence, actual cost of reconstn is: 1,00,000. (No ctrl)

(ii) - If the ctrl S/ω is implemented,

effectiveness (60%) = 60,000 (saving).

So, cost is: 1,00,000 - 60,000 = 40,000.

So, if S/ω is purchased, cost is 40,000/-, along with cost of S/ω = 25,000.

- So, total cost = 40,000 + 25,000 = 65,000.

→ Hence,

→ cost of reconstn : 1,00,000 (No ctrl)

10,00,000 @ 10%

→ Effectiveness (60%) : - 60,000

40,000

→ cost of S/ω + 25,000

65,000

→ So, Project Savings :

cost without ctrl - cost with ctrl

= 1,00,000 - 65,000

= 35,000.

- Project Saving decides whether recommended ctrls be implemented or not.

3. a. Natural Calamities : Flood, fire (impossi. to calc. prob).
b. System failure.
c. virus attack.
d. N/w failure.
} possi. to calc. probability but difficult

* Advantages of Risk Analysis:
1. It serves as basis for decision.
2. It promotes awareness. (about important assets & the possi. attacks on them).
3. Justify expenditure for security control.

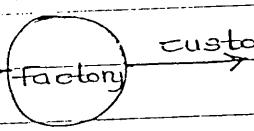
* Disadvantages:
1. It may lead to false prediction (since, it depends on probability).
2. It is difficult to perform. (we have to think of possi. attack and then their prob).
3. Risk Analysis Method is less accurate. (approx. values are calculated).

3. Organizational Security Policy: Org. prepares a Policy.
a. Purpose:
b. Contents:
- purpose
- protected assets
- methods of protection.

= Security Planning & Risk Analysis is done within org. The o/p of these steps are documented in a Security Policy.

c. Audience:

- owners = people within org.
- Beneficiaries = people related to org. eg: users



→ Purpose of Security Policy.

i. Clarifying responsibilities

ii. Promote awareness (of important assets)

iii. Guidelines for new employees

= Docu. the protected assets of organ., the methods of protn, which is avail. to everyone in org. This can be referred by the beneficiaries.

d. Characteristics of good Security policy:

i. Coverage: It should cover everything related to Security.

ii. Realism: Policy should match with actual implementn.

iii. Usefulness: Info. should be given such that it is useful to users.

iv. Durability: Whatever is implemented should be reflecte

in the policy (related to technology)

→ EXAMPLE

1. Data Sensitivity Policy.

Types of Data	Example
Sensitive	Profit Plan
Personal Info	Employee Info.
confidential	operating Plan / Balance sheets.
open	Company magazine

- Sensitive data remains only with the management, while confidential data remains within org. (managers and subordinates).

2. Internet Security Policy : listed by ISP.
- Policy is listed by listing clauses.

Clauses:

a. ISP will be responsible for passing the data to Rx. All Security requirements should be implemented by users.

4. Physical Security:
- Natural calamities:
  - Flood
  - Fire.
- Power Failure.
- Human Vandals.
  - Theft

Legal, Privacy & Ethical Issues.

→ Case Study:

1. Protected Information
2. Info laws
3. Rights of employers and employees
4. S/w failures
5. Ethical Issues

→ 1. Protected Info.:

a. Copyright:
b. Patents
c. Tradesecret

a. copyright:

i. Registeration for copyright (simple procedure)
   © copyrights reserved
ii. Data is accessible to everyone

b. Patents:

- used to mark a particular invention.
  eg: Person invents an engine with high efficiency.

i. Registration: (Complex)
   Registration office checks whether invention is actually new.
ii. The person wants his name be attached to his invention but the technology should not be revealed to anyone.
   - But the person has to reveal the technology to the registration authority.

c. Tradesecret:

- Every org have a tradesecret, which is not revealed to anyone.
- Registration not reqd.

eg: The proportion of contents in a softdrink is a trademark for the softdrink company.

1. Hardware ——— Patent
2. Object code ——— copyright
3. Source code ——— Tradesecret
4. Documentation ——— Copyright

e. Info. laws:

a. Characteristics of Info.:

i. Info. is undepletable:

- IF H/w is sold after some time, price obtained will be low, but, info does not grow old re: its value does not decrease with time.
- Info. pass as it is, no depreciation.

ii. Info. is replicable:

iii. Info. has less marginal cost:

- Marginal cost — cost of one product.

eg: Newspaper

- producing one or two copy of the document will not be costly.

→ * Info laws:

1. Criminal law
2. civil law
3. Tort law
4. Contract

- Cases under info are handled under these laws.

1. Criminal law:
- This law is applicable if case come thru' Government.
- eg: 1) Attacker attacks bank website & steals money - The Bank can lodge a case under criminal law.
- eg: Attacker tries to steal organizational data.

2. civil law: not like criminal law not req a high std proof of guilt.
- loss on smaller basis like organizational law.

Fraud is a Common or

3. Tort law: Comp. info" is perfectly suited for tort law.
- The case will be solved by referring previous case results.
- Tort law works on previously stored results.

eg: Taking info" from someone w/o permission & selling to someone as your own

4. Contract:
- betn 2 parties like org - employee or org - org.

eg:

A		B
employee or org	offer →	employee
	← acceptance	

Contract : Agreement bet^n 2 parties

- contract completed when offer is accepted by other party.
* Breach of contract: Any one of the parties goes against the terms of contract.
- A complaint will be compared with breach of contract.

3. Rights of Employer & Employees:
ownership of
- products
- copyright
- patent.

eg: If an employee makes a S/W, using company resources. Copyright is to be obtained/patent is to be obtained, then who is given the right?
- Sometimes it is decided while the contract is made.

a. Work by Hire: Employee works on employer's resources. Here, all the ownership right go to employer.

b. License: Employee is working, ownership is with employee. The employee can give license to employer to use the product.
- Here, resources used will either be of employee or employer, which is decided during contract.

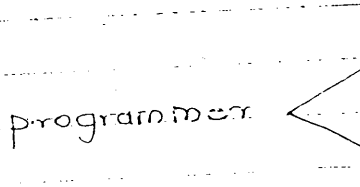
→ 4. S/w failures:

eg: S/w company wants to provide correct s/w to Bank →
However, in case of failure → company handles
failure by =

a. Replace S/w : New S/w.
b. fix a fault:
c. Refund money:

→ 5. Ethical Issues:

eg: Programmer is working of an organizational program
he also develops another program (not for org.)



- Unethical behaviour: (but not against law).
1. He is using company's resources to develop own pgm
2. original work (org. prgm) may get affected.

Ethics	Law
1. No fixed rule, varies from individual to individual.	1. Fixed Rules

- The unethical behaviour goes against law, if the company has signed a contract listing condns, then company can file a case.

Hence, unethical behaviour may or may not be against law.

→

Rule based	Consequence based
1. This grp says that rules are same for individual: all individual. universally the rules are same.	1. This grp determines a particular behaviour as ethical or unethical depending upon its consequences to individu eg: a person uses minimum/ negligible amount of resources, does not affec the company. (Ethical Behaviour). = consequence to universe.

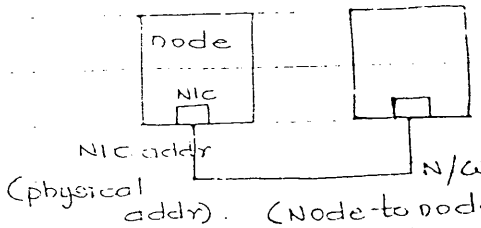
॥ श्री साई ॥

GS. Network Security.

1. OSI

- Appln = Data
- Presentation = Encryption
- Session = Handling msg sequencing error.
- Transport = Error detection & recovery (user to user) (connection establishment).
- Network
  - Error in pkts
  - Flow control (slow the rate of transmission).
- Data link layer
  - Error detection & recovery of frames (Node-to-node)
  - Flow control (By sliding window protocol).
- Physical = Signal.

2. (NIC - N/w Interface card.

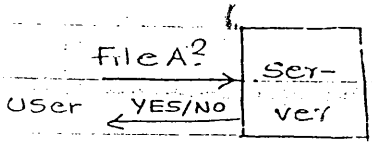


- whenever a node/terminal is to be connected to N/w, it is connected thru' NIC which has a unique addr.
- NIC addr. = phy. addr)

- If there is any error, msg can be send back to src, so that src can retransmit msg.
- In OSI, this takes place at Data link layer where error in frames can be handled using NIC addr.
((Sender sends data with IP, but the data is transmitted on N/w (node to node) with the phy. addr (NIC addr. When data reach destn, phy. addr is converted to IP addr). = using ARP (IP → phy) & RARP (phy → IP).
- (The phy. addr is given to the RARP server, to get a

unique IP addr).

3. (IF a huge file is to be transmitted/transferred, we want connection oriented protocol i.e: reliable comm^n, hence TCP used)
- UDP is used if very small msgs are to be send where sequencing is not reqd. Also error detection of msg is not needed.
Eg: user enquires existence of file A on a remote server. The answer to this will be either Yes or No. This query can be handled using UDP.



4. a) Sequencing. : protocol arrange msg/file in sequence
b) Encryption. : preserve confidentiality
c) Integrity. : preserve modification.

5. Social Engg attack is more likely to succeed over telephone as attacker may not be facing user & therefore, attacker can pretend to be someone else.
eg: Attacker pretending to be maintenance officer.
- In email, comm^n is not continuous, hence comm^n is not complete & time consuming. - may not succeed.

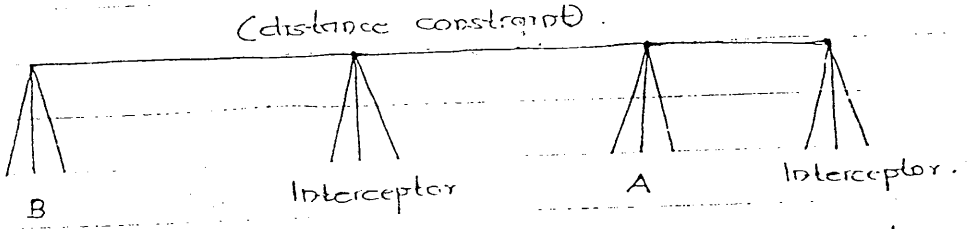
6. Suppose a file is being transferred between user A & user B. This will be done on some ports of A & B. Another user C may be interested in knowing what

comm^n is going on betn 2 users. In this case, the user may user port scanning to know that file is transferred. After this, the user C may decide to attack or not to attack.

May 2005 Q.6(b) 7. (Passive wiretapping: Attacker steals / looks at data. Active wiretapping: Attacker modifies / adds data.)

a. Copper wire: It is more prone to passive wiretapping as attacker can look for data using inductance mechanism. Also, active wiretapping is possible thru' impedance where attacker may add new data.

b. Microwave: Microwave comm^n follows line of sight principle, where two user antennas have to be aligned for comm^n. In microwave, passive wiretapping is possible where attacker may try to take data in betn.



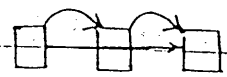
Microwave comm^n is less prone to active wiretapping as it is used to pass a large amount of data. Attacker generating such data and passing it is a rare case.

c. optical fibre: In this, info is passed in form of light thru' refraction mechanism. Therefore, it is ~~less~~ not prone to passive wiretapping as data will be lost if attacker intercepts. Also, there cant be active wiretapping on fibre optics. (Inserting data in form of light that follow the same parameters is very difficult).

d. InfraRed: This mechanism is used where comm^n is within a limited area.

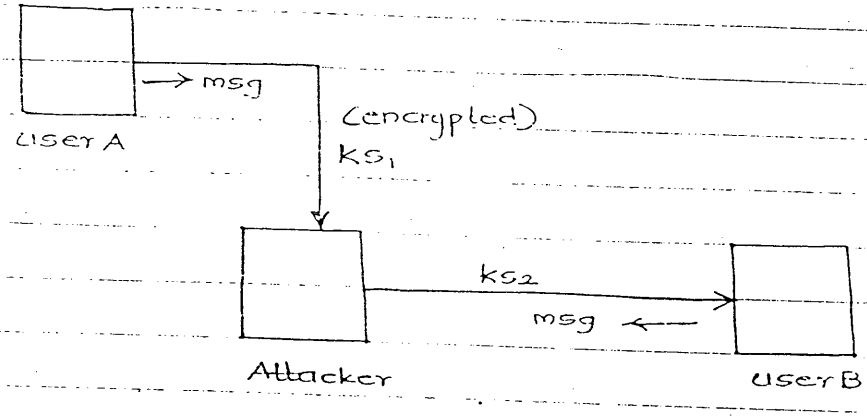
eg: employee connecting to company N/w.

- uses Radio freq. wave, no security, used for short comm^n
- It is prone to both active & passive wiretapping.
- Here, attacker blocks comm^n such that it receives the msg from sender. Then the attacker transmits / send the data to the Rx



e. Wireless: This comm^n is prone to passive wiretapping as signals will be avail. openly & can be received by an attacker. It is also prone to active wiretapping where info. can be altered and also because of DHCP (Dynamic Host config. protocol). Here attacker may acquire an IP addr., & sends msg on the N/w.

8.



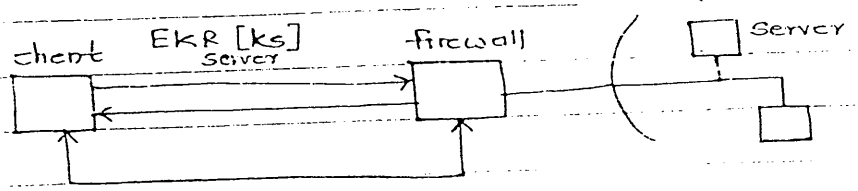
a. Means not requiring cryptography:

Man-in-the middle attack can be prevented by using strong one-time authentication where all users are distributed keys at the time of N/w formation. This can

be done by an independent server (keys distributed at the time of new conn? formed, hence, no key exchange reqd afterwards).

b. Means involving cryptography:
Man-in-the-middle attack can be prevented using Authentication process that involves cryptography.
eg: kerberos mechanism can be used where shared secret keys are passed by Authenticating server & TGS.

14. VPN uses symmetric encryption as communication is bet user & networking device. This comm? can be done using session keys or any shared secret key. To exchange this session keys, firewall may make use of private & public keys of server. But once the session keys are exchanged further comm takes place using session key. Therefore, it is symm. encryption as data will be encrypted & decrypted using same session key.

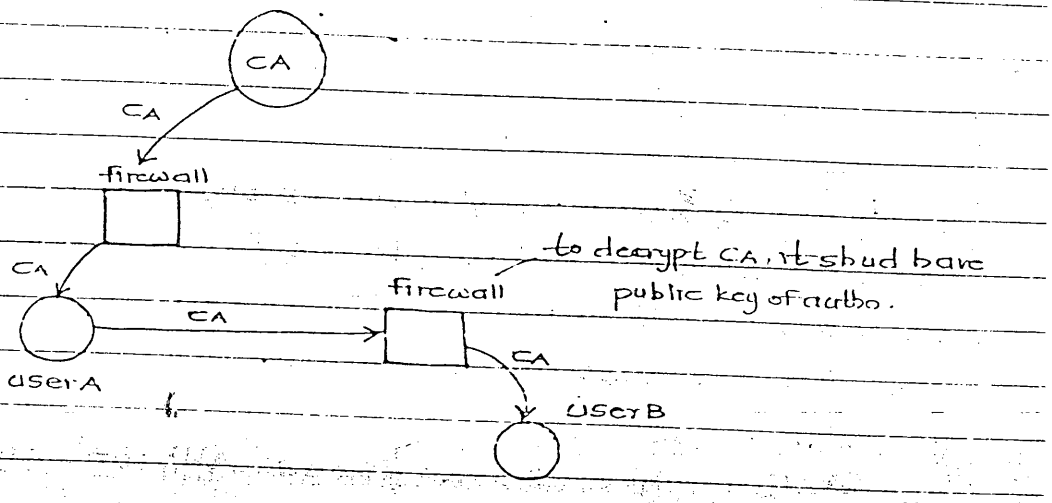


Sometimes session keys can be exchanged using Diffie Hellman key exchange algo.

15. Explain key distribution using certificate authority.

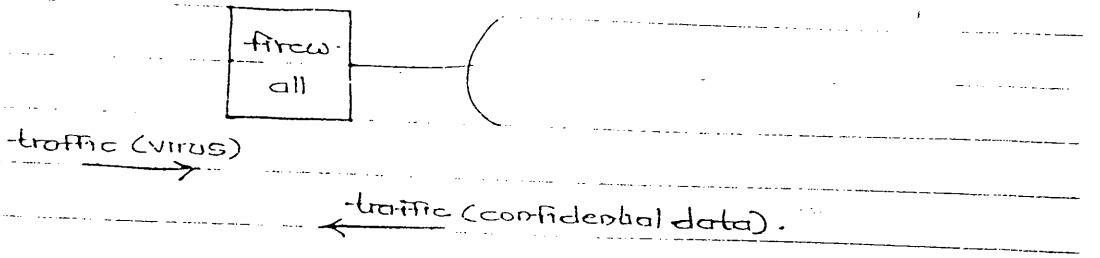
- unsymmetric encryption since the encryption is done using private key of authority while decryption is done using public key of authority.

16.



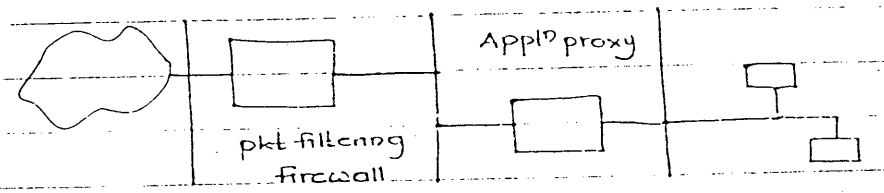
- PKI should not be supported on firewall as this requires checking and decrypting certificates using public key of authority. If firewall is used then it needs to store public key of authority to decrypt certificates. This wont be a secure mechanism as firewall is more prone to attack than any user on N/w. (public key should not be avail. with firewall, it is safe to keep public key of authority with user).

9. virus signatures / patterns.



- Not necessarily the rules of firewalls be symmetric. It could at times be one-sided.

21.



- pkt filtering firewall checks only for addrs, prevent
- Both port scan. Proxy looks for content.
- Both firewall handle diff. activities.
- Increase speed. & security.

22. Multipurpose, saves times. Although alarm is generated, management commands are not stopped (Data flow is not affected).

23. If 'A' wants to send msg 'M' to 'B',

$$A \xrightarrow{E_{KU_B}(M \,||\, H)} B$$

(confid. over M; integrity over H)

where, 'H' can be calc. using MD5 or SHA.

Ch: 5

(GQ). **Database Security**

1.	T_1	T_2
	Lock X(B)	
	r(B)	
	B = B - 50;	
	w(B)	
		Lock S(A)
		r(A)
		Lock S(B) — lock cannot be granted, since 'lock' in exclusive mode is not released.
lock can be rted, but waiting on T_1 to lease the lock on 'B'.	— Lock X(A)	

— Here, 'T_1' is waiting for 'T_2' & 'T_2' waiting for 'T_1' — deadlock.

* One user causing deadlock:

A user is handling/locking a var that may be needed by many other user. In this case, one user is causing **indefinate** postponement to all other users.

5.	T_1	T_2
	r(A)	
	A = A - 50;	
		r(B)
		r(A) ← T_2 reads the old value of A (DB value).
	w(A)	

* Integrity:

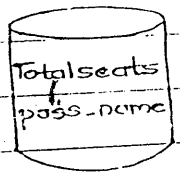
T_2 reads old value of 'A', which is immediately modified after T_2 oper^n.

* Nesting mechanism:

a) Nesting can be controlled by using 2 phase locking protocol where all the locks will be acquired in growing phase ie: at the beginning of trans and locks will be released in shrinking phase ie: towards the end of transc.

b) Time stamp based locking protocol can also be used where time stamp is alloted, so that the user knows whether it is reading old or new value.

2.



a. Intent phase:-

1. Check if commit flag is set
   If unset, then proceed.

2. TSEAT NO = '4C'.
   TPassenger_name = 'ABC'.
   TTotalseats = TTotalseats - 1;

b. Commit phase:

1. Set commit Flag

2. SeatNo = TSEATNO.
   Pass_name = TPassenger_name
   Totalseats = TTotalseats.

3. Unset commit Flag.

6. If two records are at separate places & if one is updated while the other is not, the records show two diff. values → inconsistency.
Hence, we cannot have 2 identical records without having negative effect on integrity of DB.

7. Before the os updates the DB, if other user uses the data, it will get old data.

9. To hide some data, it is encrypted.

(GQ) OS security. Ch: 4

3. Every pgm starts with '0' which is added with addr. of fence reg., to get relocated addr.

4. User A

fileB	R
pgmX	RW
⋮	⋮

- Modification is normally done by OS.
- Modification is not allowed by user since can attacker can modify a directory on behalf of the user.

6. Deleting is difficult since the data needs to be deleted from each user directory.
Soln: ACL (per object wise)
ACM.

7. So that only authorized users can change data or access data and it is not changed by anyone else. (This is how ACM preserves the integrity of obj).

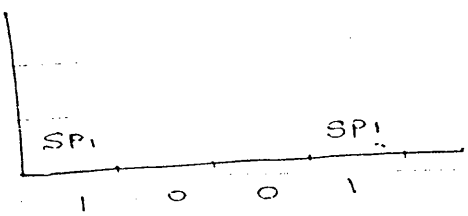
8. Situation where process transfer capability to other: In an org. mgnt wants to pass certain docu. to all the depts. This can be done by passing a token to one dept so that, that dept can further pass the docu. to other.

9. Disadv. of phy. separation:
a. More resources are needed.
b. overhead of having security on all resources.

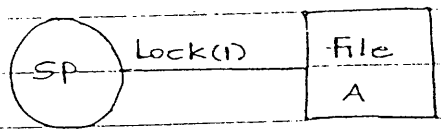
Disadv. of temporal separation:
a. Data open are in queue.
b. Need for an algo. for deciding priorities.

(GQ) **Program Security.** Ch:3

1. Covert timing channel passes the info. whenever service pgm is executed in a particular time slot. ie: info. bits will be passed during a particular time slot.



Covert storage channel passes info. whenever a certain obj is accessed by user. for eg: If user accesses file A, it provides a lock on that file which marks i.e. info is passed when a particular obj is locked & unlocked.



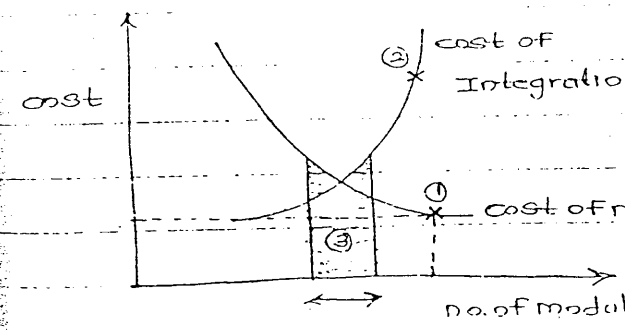
Ex May 2005 4(a)

→ Converting timing channel to storage channel:
Convert timing channel can be converted into storage channel by passing the info. only if service pgm accesses a particular obj. A service pgm will be executed on a time slot but no info is conveyed. Info bits are passed only when obj is accessed or locked.

module size is less then encode gets affected

(a) Approach 1: static approach
Every module should be assigned a logical fn. One major fn should be splitted into more modules.

(b) Approach 2:



Adv. of modularity is - Maintenance.
No. of modules ↑, cost of maintenance ↓ (easy to detect & correct errors).

② No of modules ↑, cost of integration ↑.

③ Actual no of modules should be within the shaded region, to have reasonable cost of maintenance & Integrity.

(8) 4. Audit logs contains (trans info + user info).

* Pieces of info in log that identify impostor.

a. Logins, Logouts.

b. Accesses or attempted accesses to files or directory.

c. Execution of pgms

d. uses of other devices.

(8) 5. eg: encrypted (ADD A, B)

↓ decrypt

A+B

↓ result encrypted.

A

- Instru to processor are in encrypted form. The result is encrypted as soon as generated. Hence pgm & data is not seen / visible to anyone. (visible in encrypted format).

* Design requirements:

1. Fast: Decrypt → perform oper^n → Encrypt.
2. Inbuilt encryption algo.: processor will not refer outside memo for algo, since it may ↓ speed/perfor
3. Inbuilt registers: to store the results before they are encrypted.

(Gg) System Security.

1. vulnerability = certain flaw/flaut in sys.
Threat = IF the flaw is detected, attack is possi.
control = preventive measures.

MAY 2005.

b.1) Consequence based approach: In some conditions, Dave's extra work may affect the batch productions that are done. These condns are=
→ on a particular day, if load on sys is more or there is more work, then Dave's work may cause delay as he has occupied a particular resource.
→ IF there is some failure in system because of which some M/cs may not work, so employees may have to look for additional resources. In this case, Dave's extra work will block resources.

With respect to above condns, consequences of Dave work are such that he is disturbing the actual sys. (Batch Prodn). Therefore, his behaviour is unethical.

OR

Rule based approach: As per this approach, Dave's behaviour is unethical as he is breaking the rules or code of conduct which normally says, An employee is to perform only the task assigned.

11) Ethics as ownership of resources: Dave uses org resources, hence ownership rights go to org.

Effect on others: Sometimes he may be affecting other & sometimes not.

Universalism: This principle is not applicable here, as Dave's behaviour affects only the org.

Possibility of detection & punishment: Punishment will depends upon the conditions of contract. If Dave works against the conds listed in contract, then org can file case against him.

Q] Authentication means ॥ श्री साई ॥
proving identities betⁿ entities which happen in diff. layers of ntw protocol stack for diff reasons.

May'06

Q-1(a) (i) Identify these entities & State them

(M) - 1a. Entities -
- process to process commⁿ.
- user to user.
- user to networking device.

→ process to process commⁿ: A process running on a particular port may need to identify itself to a process running on other user's port. This identification can take place thru' MAC (msg authentication code) or certificates. This is implemented in SSL.

→ user to user commⁿ: Here, one user may need to identify itself to other user. This can be done using certificates or HMAC. This is implemented using IPsec protocol (transport mode).

→ user to networking device: Here, user identifies itself to networking device (Firewall). This can be done using certificates as implemented in VPN. Also, set of public and private keys can be used in VPN. Authentication can also be thru' HMAC as implemented in IPsec protocol (tunnel mode).

Q-1(d) Five design principles of any Cryptographic Algo.

(M) a) cryptographic algo should perform more operⁿs in terms of permutation & combⁿs, so that it becomes difficult to break.

b) cryptographic algo should be such that if there is any error during transmission, then that error should not propogate thru'out the msg.

c) set of keys & algo should be free from any condⁿs for implementation.

eg: algo works if the msg contains more than 5 'A's ; cond^n like this should not be there.

d) key size in an algo should be such that when used with msg, complexity is increased so that, attack becomes difficult.

e) Size of cipher should be $\leq$ msg.

b 1. Private Public $X = E_{SK_1}(E_{PK_2}(M))$

Bubbly SK_1 PK_1

Bunty SK_2 PK_2

$E_{SK_1}[E_{PK_2}(M)]$

user C $\downarrow$ DPK_1

$[E_{PK_2}(M)]$

$\downarrow$ ESK_C

$E_{SK_C}[E_{PK_2}(M)]$

- Here, confidentiality is preserved but authenticity is not preserved
- Hence, correct form of encryption is-

$E_{PK_2}[E_{SK_1}(M)]$

- As no checksum used, Integrity is not preserved.

May 2006.

3 a. SQL allows different users to access diff relations in Read mode or R/W mode.
((Access control:
Select (Read mode)
update (R/W mode)))

1. grant select on branch (rel^n) to u1, u2, u3
2. grant update (amount (attri)) on loan (rel^n) to u1, u2
3. grant references (branchname (attri)) on branch (rel^n) to u1.
(u1 can create a new table with 'branchname' as foreign key).
4. revoke select on branch from u1, u2, u3.
(access right is taken back / cancelled).
5. grant select on branch to u1, u2, u3 with grant option.
(grant is given to u1, u2, u3, which in turn can allow other users to read).
6. revoke select on branch from u1, u2, u3 cascade.
(Grant cancelled for u1, u2, u3, also, read mode of those users who got the grant from u1 or u2 or u3 are cancelled).

revoke select on branch from u1, u2, u3 restrict.
(Grant cancelled for u1, u2, u3).

revoke grant option for select on branch from u1.
(Take back grant from u1).

P.N. 3-7

7b. Fields of digital certificate:

At minimum, each certificate contains:

i. owner's public key
ii. owner's name
iii. expiration date of digital ID
iv. Serial no. of digital ID
v. Name of certificate Auth (user knows the name of CA).
vi. Digital signature of DA (authentic)

Also, certificate may contain:

Address, email, basic registration info (eg. country, zipcode, age, gender).

DEC 05

7 b. Rule based approach:

If Patty writes program, this is unethical as she is doing wrong work which will harm org.

Consequence based approach:

If consequences of Patty's pgm are less. for eg. Much of the data in accounting files is not changed & further, even if it is changed, it is not a major loss to org. In this case, Patty's behaviour will be ethical. But if the effect of the pgm are disastrous ie: company may lose major amounts then, Patty's behaviour is unethical

Solution:

1. Patty can consult some person from higher level mgmt rather than immediate supervisor.

2. Patty can also try to implement security in pgms. This may involve extra efforts on her part & she may need to do more work than actually assigned.

3 a i. Digital Signature — Digital certificate.

→ Is of an individual — → Given by CA.

→ No expiration. — → may have expiry date.

→ Prove identity ie: Authentication. — → used to prove identities or for transmitting public keys.

→ (Explain mechanism)

ii. To trust a digital certificate, check for—
→ expiration date
→ digital signature of CA.

iii. a) CA provide certificate with limited validity.
b) Certificates can be taken for one-time comm°.
Adv: If the certificate is stolen, it will be of no use after the comm° session is over.

4 a ii. a) Password
b) challenge response sys. } user wants to authenticate itself to sys. (ie: user login to access pgm/dB).

a) Certificates
b) Digital signature
c) kerberos

} user proving its identity to another use.

iii. Flaws in user authentication process: Attacks on the authentication is possible.

eg: Password attack: anyone can login system.

iv. Controls:

password: long pswd, characters other than alphabets, change pswd periodically, avoid actual names.

challenge response sys: Both ends should have some complex eqns, so that it is difficult to guess.

CA: expiration date, digital signature of CA

Digital Signature: It should not be send as it is, it should be encrypted.

kerberos: (negligible flaws) no this reqd.

May 05 (Q.1)

1. RSA Algo :- (Rivest, Shamir & Adleman).

a. p, q → prime nos. Eg:- a) 17, 11

b. n = pq. b) n = 17 × 11

c. z = (p-1)(q-1). c) z = 16 × 10 = 160

d. Select key 'e', such that d) e = 7
e is relatively prime to z.
(e & z should not
have any common factor)

e. Select 'd', e) d × 7 ≡ 1 mod 160
$de \equiv 1 \bmod z$. $(160\times1+1)\ (160\times2+1)$
(161, 321, 481, 641, ...)

$d \times 7 = 161$
321
481
641
⋮

(Select 'd' which is
exactly divisible by 7)
∴ d = 23.

f. $C = P^e \bmod n$ f) eg: to transmit 'd', '4'
is send., a → 1, b → 2,

g. $P = C^d \bmod n$.

- asymmetric algo.
- Since, 'mod n' is reqd. for encrypt & decryption,
hence, public key $KU = \{e, n\}$
private key $KR = \{d, n\}$.

Q.1

MAY05-1

Alice	Bob
public {17, 321}	public {5, 321}
(e, n)	(e, n)

Hence, $n = 321$

$= p \times q$

$= 3 \times 107$

Then, $z = (3-1)(107-1)$

$= 2 \times 106$

$= 212$

a) $c = p^e \bmod n$

$c = 7^5 \bmod 321$

$c = (7^1 \times 7^4) \bmod 321$

$= [(7^1 \bmod 321) \times (7^4 \bmod 321)] \bmod 321$

$= (7 \times 154) \bmod 321$

$= 115$

b) To decrypt the msg send by Bob, Alice needs private key of Bob.

c) To find private key or 'd' : (of Bob)

$z = 212$

$d \times e = 1 \bmod 212$

$d \times 5 = 1 \bmod 212$

213

425

637

:

$\therefore d = 85$

can be calculated by Alice. Since, alice knows - $z = 212$, $e = 5$.

$\therefore$ Bobs private key is {85, 321}.

$(a+b)^4 = 1+2+3+4$

$= 1+2+2$

$= 1+4$

d) $z = 212$

$d \times 17 = 1 \bmod 212$

213
425
637

$d = 25$

∴ Alice private key is $\{25, 321\}$.

b) Alice will find out $\{d, n\}$ of Bob., which is $\{85, 321\}$.

$P = C^d \bmod n$

$P = C^{85} \bmod 321$

c) Bob will decrypt it using his private key $\{85, 321\}$.

$P = C^{85} \bmod 321$.

DEC 05

Integrity

1a) Protocol 1: only confidentiality is preserved & non-repudiation of Rx

ii) only confid. is preserved & non-repu. of Rx

Protocol 2: i) only confid. is preserved + non-repu. of Rx (non-repudiation of sender not preserved. since private key of sender not used).

ii) only confid. is preserved. + non-repu. of Rx.

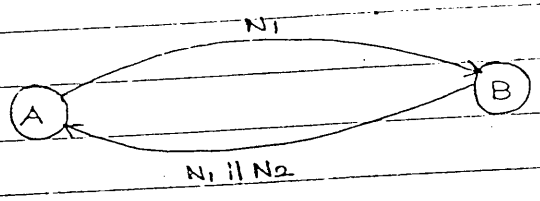
Integrity & authentication

b. Redesign:

1. $E_{KUb}[M \| E_{KRa}(H)]$

(Same dgm as PGP, without compression part)

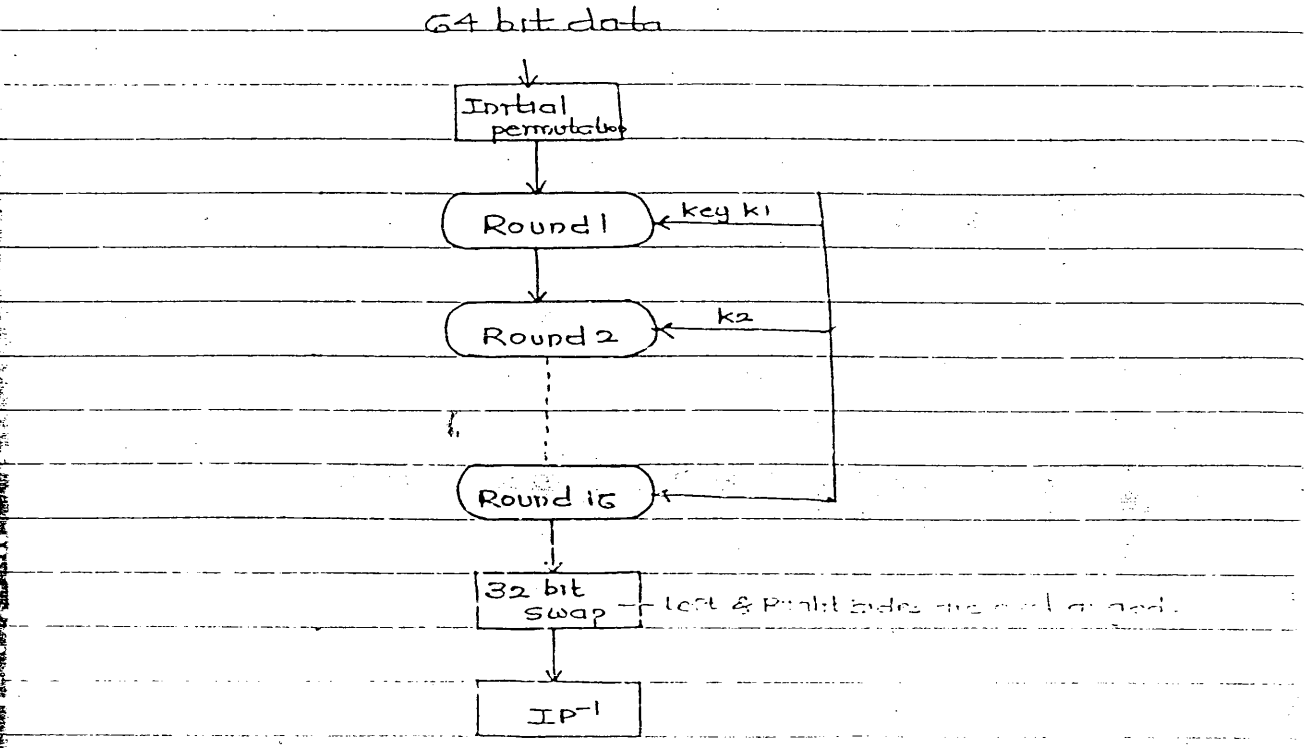
11. With whole msg, nounce value is used. (But comm^n should be 2 way, so that ack should be received)



- However, this is overhead, as every msg has to be replied.

2 Data Encryption Standard (DES):

- Block cipher.
- 64 bit block data.



Step 1:

a. Initial Permutation: (IP) (Std DES).

18	10	2		
20	12	4		
22	14	6		64 bits
24	16	8		
25	17	9	1	
19	11	3		
21	13	5		
23	15	7		

eg: 1100

2 3 1 4 permutation

1010. (posⁿ of bits are changed).
(data not changed.

Step3 b Inverse Initial Permutation: (IP^{-1}) (Std DES).

bits →								
40th bit comes first	40	8	48	16	56	24	64	32
	39	7	47	15		23		31
	38	6	46	14		22		30
	37	5	45	13		21		29
	36	4	44	12		20		28
	35	3	43	11		19		27
	34	2	42	10		18		26
	33	1	41	9	49	17	57	25

* Expansion permutation:

eg: 1100

4 bit

2 3 1 4 3 1 2 1 expansion permutation.

8 bit

10100111

* Contraction permutation:

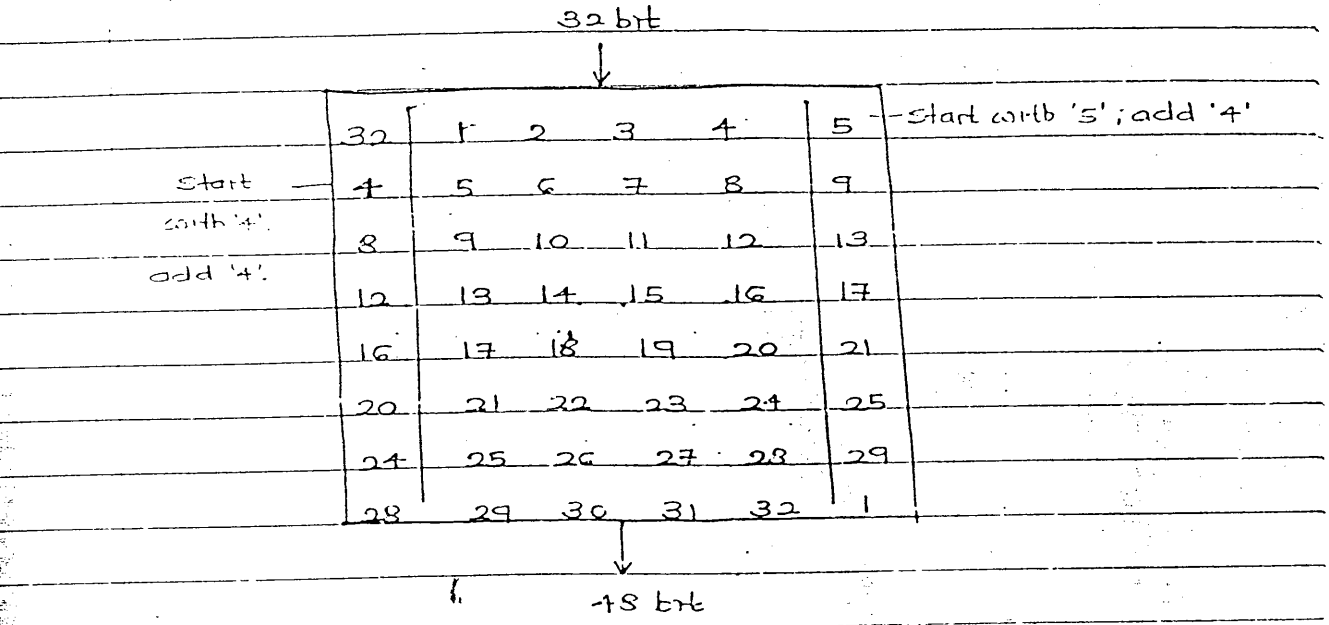
10101100

2 3 1 4 contraction permutation.

4 bit

0110

c. Expansion permutation: (Std DES)

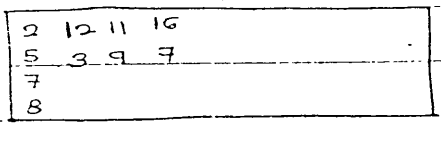


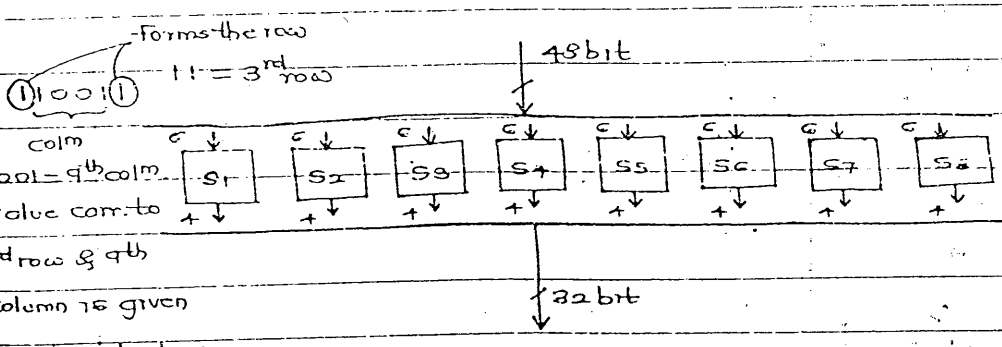
- To generate 48 bit o/p, certain bits are replicated.

d. Contraction permutation: (Std DES).

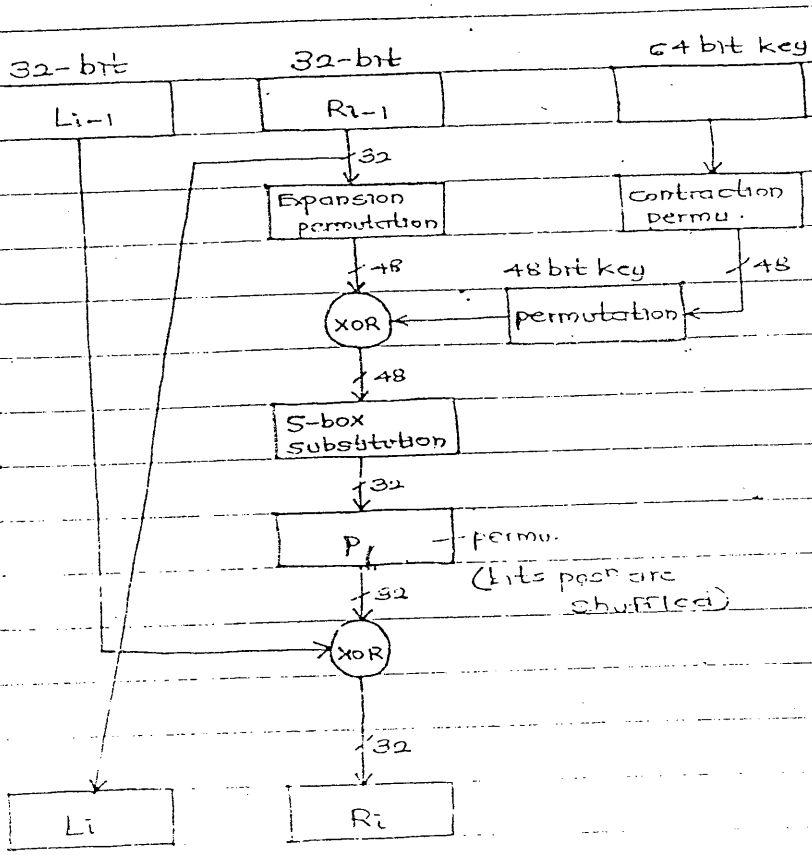
- block any 2 columns to get 32 bit o/p from 48 bit i/p.

e. S-box substitution:





cp2- * Steps performed during each round:



- In DES, expansion permu / contraction permu. & S-box substitution are Standard.
- Same key is used for all the rounds.