Module No.	Module Title and Brief Details
1	Security Goals, Attacks, Services and Mechanisms, Techniques. Modular Arithmetic: Euclidean Algorithm, Fermat's and Euler's theorem. Classical Encryption techniques, Symmetric cipher model, mono-alphabetic and polyalphabetic substitution techniques: Vigenere cipher, playfair cipher, Hill cipher, transposition techniques: keyed and keyless transposition ciphers
2	Symmetric and Asymmetric key Cryptography and key Management Block cipher principles, block cipher modes of operation, DES, Double DES, Triple DES, Advanced Encryption Standard (AES), Stream Ciphers: RC4 algorithm. Public key cryptography: Principles of public key cryptosystems- The RSA Cryptosystem, The knapsack cryptosystem. Symmetric Key Distribution: KDC, Needham-schroeder protocol. Kerberos: Kerberos Authentication protocol, Symmetric key agreement: Diffie Hellman, Public key Distribution: Digital Certificate: X.509, PKI
3	Cryptographic Hash Functions Cryptographic hash functions, Properties of secure hash function, MD5, SHA-1, MAC, HMAC, CMAC.
4	Authentication Protocols & Digital Signature Schemes User Authentication, Entity Authentication: Password Base, Challenge Response Based Digital Signature, Attacks on Digital Signature, Digital Signature Scheme: RSA
5	Network Security and Applications Network security basics: TCP/IP vulnerabilities (Layer wise), Network Attacks: Packet Sniffing, ARP spoofing, port scanning, IP spoofing. Denial of Service: DOS attacks, ICMP flood, SYN flood, UDP flood, Distributed Denial of Service Internet Security Protocols: PGP, SSL, IPSEC. Network security: IDS, Firewalls
6	System Security Buffer Overflow, malicious Programs: Worms and Viruses, SQL injection

* Insert rows for more modules in the Course

Total

MSEI II III

chapter - 1, 4 chap - 2 chap - 3, 5, 6

PAGE NO:

Digital signature and digital certificate

when any client wants to verify authenticity of server, it needs digital certificate, specifically digitally signed certificate.

Digital certificate and signature creation

step 1 :

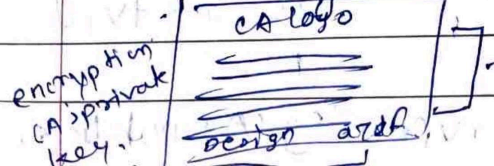
1) server creates Digital certificate [DC] & sends it to CA [certification Authority]
2) server encrypts the DC using servers private key.
* Any attacker can impersonate server & to avoid this use server. private key for encryption.
3) DC contains foll. info
   1) URL
   2) MAC add
   3) Port add
   4) Public key
   5) Expiry Date
   6) signature

step 2 :

1) CA Decrypts the DC using servers public key & if succesful that means DC is coming from valid server.
2) CA verifies all the details of the DC

Step 3 :

Assuming Details are valid



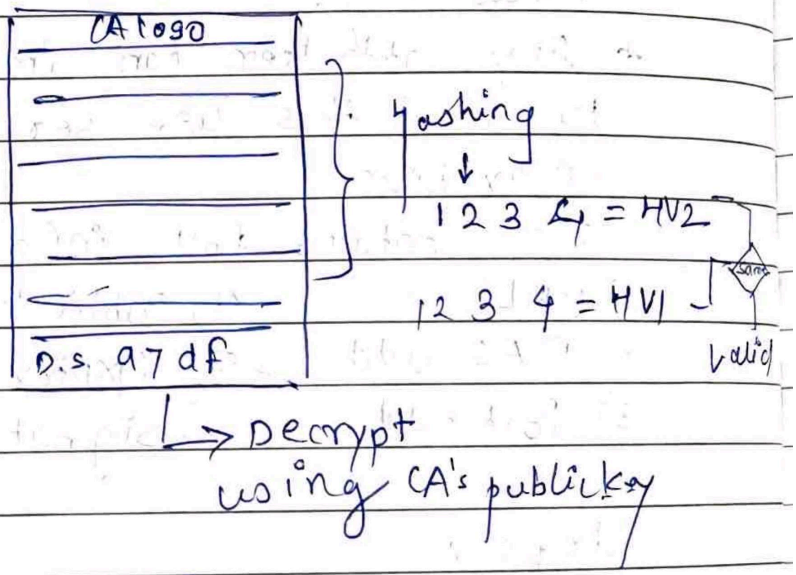
1) Take the DC and apply hashing
2) on hash value apply encryption using CA's private key & that is the D.S
3) CA sends DSC to server

Digital signature and certificate Verification

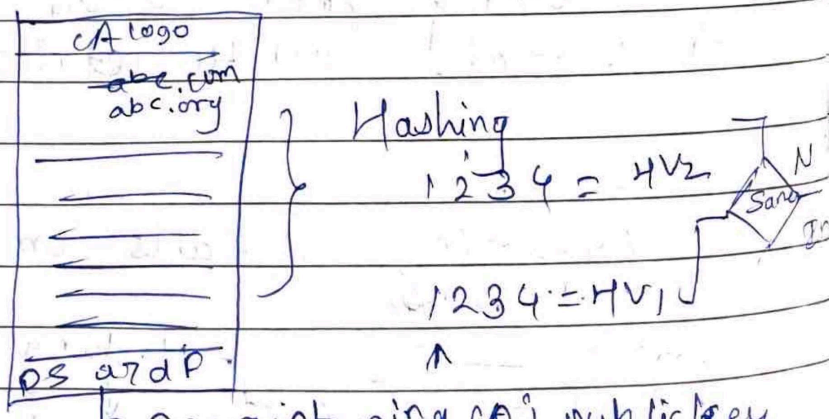
Step1: whenever client wants to verify authenticity of server, it request DSC from the server

Step2: It performs verification in the following manner.

Case I] Valid Case

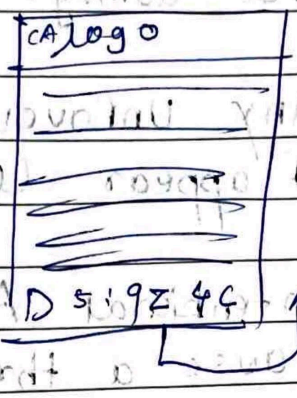


Case II] Tampering of digital certificate after signature



PAGE NO.:

Case III] fake certificate (Signed by Attacker)



step 3 :
CA verifies futhere with CRL (certificate Revokation List) database. It contains those certificate Revoke because of Expiry date, cybercrime case on server. private key of server compromise.

Step 4 :
If step 2 and step 3 turns to be positive then client creates virtual termina l with the server using symmetric key encryp -tion.
(Explain symmetric key with example)

valid

Chapter 4.

PAGE NO.:

Risk :- Any Uncertain event which may cause damage to one's well-being

Threat :- Any unfavourable situation which if appear causes risk

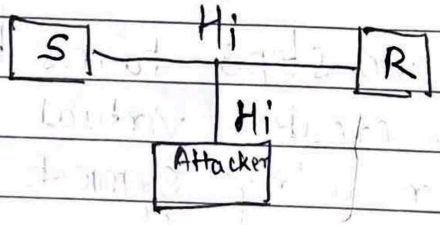
Attack :- Intentional Action which may cause a threat

Types of Threats :

1) Types of Threats

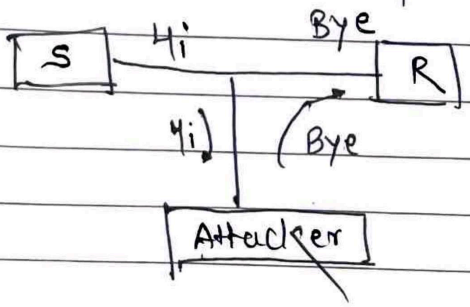
A] Interception :

An action of overhearing the communication between two parties (≡ eavesdropping)



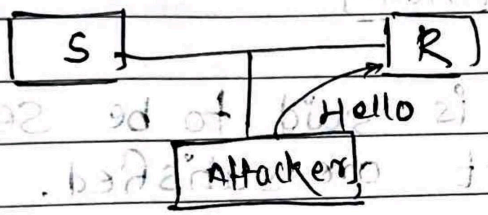
B] Modification :

An action of intentionally modifying the communication two parties



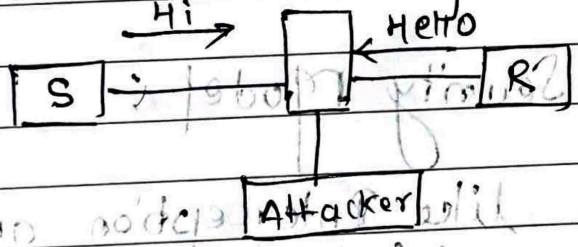
3) Fabrication

An action of intentionally sending some random data in account of achy sender (Impersonation)



4] Interruption

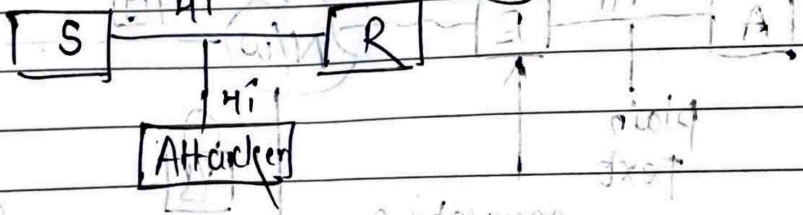
An action of blocking the communication.



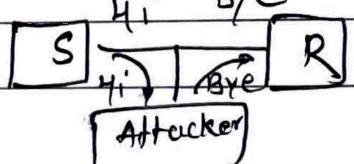
Goals of security / CIA Triad

Confidentiality : only Intended Receiver should receive the data.

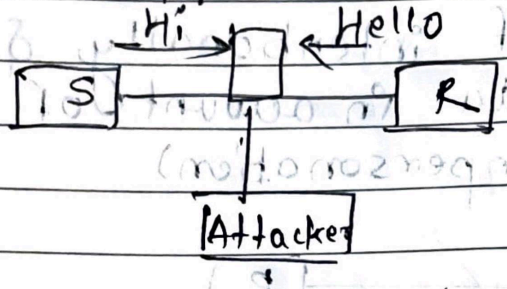
eg Loss of confidentiality



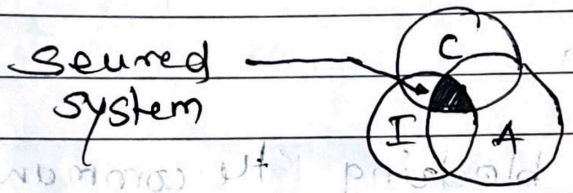
Integrity : The data sent should be delivered as it is



Availability : Any data sent has to be delivered

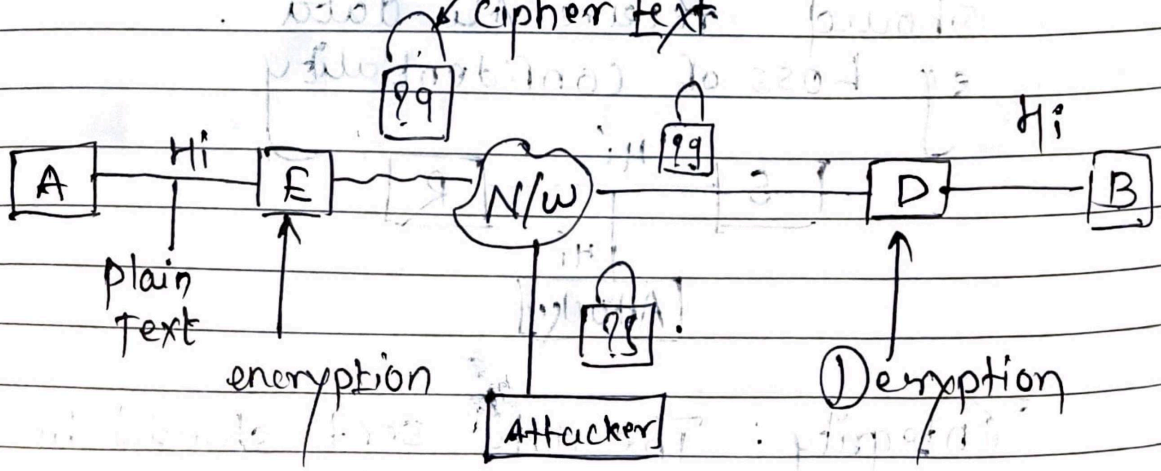


The system is said to be secured only if All 3 Aspect are satisfied.



Network Security Model :

The threats like Interception and modification needs some kind of security mechanism to protect data travelling in the network for which network security model is defined.



PAGE NO.:

Encryption and Decryption

It is a brach of crytography.

modular Arthematic

eg:) 4 mod 24 $\Rightarrow 4$

27 mod 24 $\Rightarrow 3$

331 mod 25 $\Rightarrow 6$

Note

finding mod operation

1097 mod 33

step 1 = $1097 \div 33$

$= 33.24$

step 2 = 0.24×33

$= 7.999$

≈ 8

$-6 \bmod 43$

$\Rightarrow 43 - (6 \bmod 43)$

$\Rightarrow 43 - 6$

$\Rightarrow 37$

PAGE NO.:

Caesar Cipher (Additive Cipher)

A	0
B	1
C	2
D	3
E	4
F	5
G	6
H	7
I	8
J	9
K	10
L	11
M	12
N	13
O	14
P	15
Q	16
R	17
S	18
T	19
U	20
V	21
W	22
X	23
Y	24
Z	25

for encryption

$E(x) = (x + k) \bmod 26$

for decryption

$D(x) = (x - k) \bmod 26$

If result $\Rightarrow$ [-] add 26

// eg Find caesar cipher for SECURITY for key value 5

Plain text	S	E	C	U	R	I	T	Y
Position	18	4	2	20	17	8	19	24
key	5	5	5	5	5	5	5	5
(P+k)	23	9	7	25	22	13	24	29
mod	26	26	26	26	26	26	26	26
	23	9	7	25	22	13	24	3
Ciphr	X	J	H	Z	W	N	Y	D

Decrypt

Cipher	X	J	H	Z	W	N	Y	D
plain tx	23	9	7	25	22	13	24	3
key	5	5	5	5	5	5	5	5
(P-k)	18	4	2	20	17	8	19	-2
								26 add
	18	4	2	20	17	8	19	24
	S	E	C	U	R	I	T	Y

PAGE NO.:

Q2 VI DYALANKAR k=13

Plain Text	V	I	D	Y	A	L	A	N	K	A	R	
Key	13	13	13	13	13	13	13	13	13	13	13	
Position	21	8	3	24	0	11	0	13	10	0	17	
(P+K)	34	21	16	37	13	24	13	26	23	13	30	
mod	26	26	26	26	26	26	26	26	26	26	26	
	8	21	16	11	13	24	13	0	23	13	4	

Cipher I V Q L N Y N A X N E

Decrypt

Cipher	I	V	Q	L	N	Y	N	A	X	N	E	
plasition	8	21	16	11	13	24	13	0	23	13	4	
key	13	13	13	13	13	13	13	13	13	13	13	
(P-K)	-5	2										

10 M

● Playfair Cipher

key - monarchy instruments

Step1: instruments Z

in

M O N A R

C H Y B D

E F G I K

L P Q S T

U V W X Z

cipher text ⇒ GA

step2: st

M O N A R

C H Y B D

E F G I K

L P Q [E][T]

U V W X Z

cipher tx ⇒ TL

step3: ru

M O N A [R]

C H Y B D

E F G I K

L P Q S T

[U] V W X Z

Cipher ⇒ MZ

step4: me

[M] O N A R

C H Y B D

[E] F G I K

L P Q S T

U V W X Z

cipher text ⇒ CL

step5: nt

M O [N] A R

C H Y B D

E F G I K

L P Q S [T]

U V W X Z

cipher text ⇒ RQ

Step6: SZ

M O N A R

C H Y B D

E F G I K

L P Q [S] T

U V W X [Z]

cipher text ⇒ TX

cipher text ⇒ GA TL MZ CL RQ TX

Jaipur

ZZ replaced by Z only

PAGE NO.

Keep B at 18

Q2 VIDYALANKAR key: MONEY

Step 1 VI DY A L AN KA R Z

M O N K E

Y A B C D

F G H I L

P Q R S T

U V W X Z

cipher text ⇒ XG

Step 2: M O N K E

T A B C D

F G H I L

P Q R S T

U V W X Z

Cipher text ⇒ YA

Step 3: M O N K E

Y A B C D

F G H I L

P Q R S T

U V W X Z

cipher text ⇒ DG

Step 4: M O N K E

Y A B C D

F G H I L

P Q R S T

U V W X Z

cipher text ⇒ BO

Step 5: M O N K E

Y A B C D

F G H I L

P Q R S T

U V W X Z

cipher text ⇒ OC

Step 6:

M O N K E

Y A B C D

F G H I L

P Q R S T

U V W X Z

cipher text ⇒ TW

whfefgcmgdtw

Cipher text ⇒ XGYADGBOOCTW

2. Vigenere Cipher
refer pdf

3. Vernam Cipher (one Time Pad (OTP))
Algorithm (refer pdf)

eg. VIDYALANKAR
Key : MONEY

1] plain text : VIDYALANKAR
2] position 21 8 3 24 0 11 0 13 10 0 17
3] key : MONEYMONEYM
Pos : 12 14 13 4 24 12 14 13 4 24 12
4] 33 22 16 28 24 23 14 26 28 24 2
- 26 26 26 26 26 26 26 26 26 26 26
7 22 16 2 24 23 14 0 14 24 3
H W Q C Y X O A O Y D

decrypt
7 22 16 2 24 23 14 0 14 24 3
- 12 14 13 4 24 12 14 13 4 24 12
-5 +8 3 -2 0 11 0 -13 10 0 -9
26 26 26 26 26
21 8 3 24 0 11 0 13 10 0 17
V I D Y A L A N K A R

PAGE NO.:

◉ Keyed Transposition : [check if occurrence]

eg. VIDYALANKAR
2401

encrypt

Plaintext	V	I	D	Y	A	L	A	N	K	A	R
Key	2	4	0	1	2	4	0	1	2	4	0
sort key	0	0	0	1	1	2	2	2	4	4	4
C.T	D	A	R	Y	N	V	A	K	I	L	A

C.T	D	A	R	Y	N	V	A	K	I	L	A
sort key	0	0	0	1	1	2	2	2	4	4	4
Actual key	2	4	0	1	2	4	0	1	2	4	0

V I D Y A L A N K A R

◉ key less Transposition - 10 mg - 7 Decrypt 3 encryption

encrypt TECMPNASEMSIX key=k=3

T			M			E			X	
	E			N			M			I
		C			A			S		

CT: TPSXEMNSMICASI

key = 3

decrypt step 1: filling row 1

T			P			E			X	
	-			-			-			-
		-			-			-		

step2: filling row no. 2

T				P			E			X
	E		M		N		S		M		I	
		-				-				-		

step3: filing row no. 3

T				P			E			X
	E		M		N		S		M		I	
		C				A				S		

step4: read Diagonally (zigzag)

TE CMPN A SIX

Steganography :- (refer pdf) 2nd Question

~~Modular~~ eeluds Algorithms

It is used for recursively finding GCD

```
int GCD(int x, int y)
{
    if (y == 0)
        return x;
    else
        G(1)
        return (y, x % y);
}
```

GCD(8,12)
= GCD(12, 8%12=8)
= GCD(8, 12%8=4)
= GCD(4, 8%4=0)
= 4

eg GCD 80, 105
= GCD (105, 80 % 105 = 80)
= GCD (80, 105 % 80 = 25)
= GCD (5)

What is relatively prime and co-prime.
→ when x and y have GCD as 1, it is called as relatively prime / co-prime relation.

Euler's totient function ($\phi(n)$)

$\phi(n)$ is set of all positive integers $< n$ and relatively prime to n.

eg. ϕ n = 6
$\phi(n) = \{1, 2, 3, 4, 5\}$
$\phi(n) = \{1, 5\} = 2$ //

If n is Prime then $\phi(n) = n-1$

n = 6
$\phi(n) = \phi(2) * \phi(3) = 1 * 2 = 2$ //

n = 9
$\phi(n) = \phi(3) * \phi(3) = 2 * 2 = 4$

n = 165
$\phi(n) = \phi(15) * \phi(11)$
$= \phi(3) * \phi(5) * \phi(11)$
$= 2 * 4 *$

1 = 1
2 = 1 ✗
3 = 1 ✗
4 = 1 2 ✗
5 = 1 ✗
6 = 1 2 3 6

PAGE NO.:

Euler's theorem

It works on principle of congruency ≡

$$x^{\phi(n)} \bmod n \equiv 1$$

ie when $x^{\phi(n)}$ is divided by n the remainder should be one.

- Fermates theorem:

It is also called modified eulers theorem which states $x^{\phi(n)} \bmod n \equiv 1$ if and only if prime.

In both eulers theorem & fermates x and n should be coprime.

eg $x = 3$, $n = 10$

step 1: checking coprime condition
(using eudides Algo) find GCD

$$\begin{aligned} g(3,10) &= gcd(3,10) \\ &= gcd(30, 3\%10) \\ &= gcd(10, 3) \\ &= gcd(3, 10\%3) \\ &= gcd(1) \end{aligned}$$

step 2: finding $\phi(n)$ using euler totient

$$\begin{aligned} \phi(n) &= \phi(2) * \phi(5) \\ &= 1 \times 4 \\ &= 4 \end{aligned}$$

step 3: find $x^{\phi(n)}$

$$\begin{aligned} &= 3^4 \\ &= 81 \end{aligned}$$

PAGE NO.:

step 4 : checking congruency

$x^{\phi(n)} \bmod n = 1$

$8^{1} \bmod 10 = 1$ is true

$x^{\phi(n)} \bmod n = 1$

10/11

RSA Algorithm

PAGE NO.:

It is an algorithm developed for public key and private key generation designed by rivest shamir adleman.

Algorithm — working

step1	select 2 large Primeno. P & Q	P=13, Q=11
step2	compute n	n = P * Q n = 13×11
step3	compute $\phi(n)$ $\phi(n) = \phi(P) * \phi(Q)$ by fermates theorem and Eulor's tobient function $\phi(P) = ?$ $\phi(Q) = Q - 1$	$\phi(n) = \phi(P) * \phi(Q)$ $= 12 * 10$ $= 120$
step4	select random public key E such that $0 \le e \le \phi(n)$ e & $\phi(n)$ should be co-prime. GCD should be 1.	e = 13
step5	calculate private key d $d = e \bmod \phi(n)$	$d = \frac{\phi(n) * i + 1}{e}$ i = 1,2,3,4... continue till answer is int $d = \frac{120 * 1 + 1}{13} = 9.3$, $d = \frac{120 * 2 + 1}{13} = 18$ $d = \frac{120 * 3 + 1}{13} = 27.8$, $d = \frac{120 * 4 + 1}{13} =$

PAGE NO.:

Shared with

step 6 public key = e, n public = 13, 143
Private key = d, n private = 37, 143

step 7 To encrypt
$C = P^e \bmod n$

step 8 To decrypt
$P = C^d \bmod n$

Find cipher text for P=5, e=13, n=143

⇒ P=5, e=13, n=143
$n = p * q$
$C = (5)^{13} \bmod (143)$
C = 70

RSA algorithm for digital signature

Step 1 to step 6 are same as RSA.

step 7 : Creating digital signature
$s = m^d \bmod n$
s is digital signature m = hash value of certific

step 8 : send digital signature & hash valu to the receiver (client)

step 9 : verification m is hash valu which cannot be decrypted back
Therefore $m' = s^e \bmod n$
if (m == m') then valid else invali

Next MSE

Authentication

PAGE NO.: 12/2/24

The process of checking whether any entity is valid / not

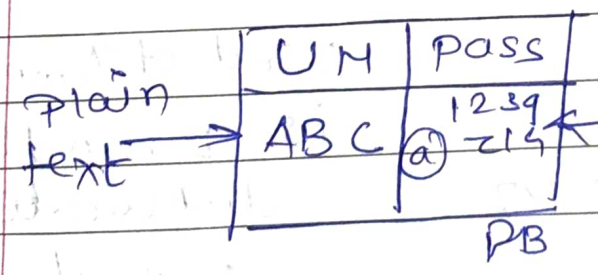
Types :

1] Entity Authentication
1.1 Password based
1.2 Hash based

2] challenge Response Authetication
2.1 Nonce based
2.2 Timestamp based
2.3 Bi-directional.

Entity Authentication :

Password based :
Server maintains database which contain user name & password.
The password is stored in encrypted form



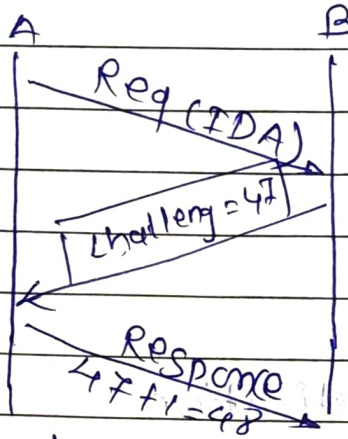
Hashing : (Same as explained before) creation only

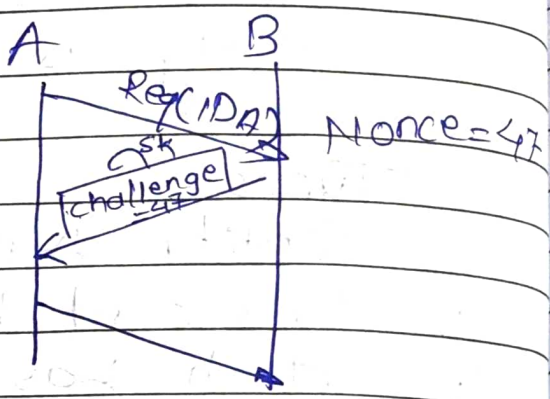
Nonce ⇒ Any randomly generated no.

PAGE NO.:

Challenge Response

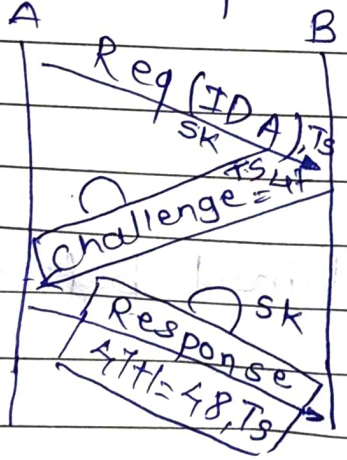
Nonce Based :





It is Assume that A is already shared session key, A now sends a request with its id
B creates random number called as nonce sends it as a challenge to A. encrypted through session key.
A decrypt the challenge, solves it and send the response in encrypted form.
Receiver decrypt the response and checks whether the result of the challenge is correct.

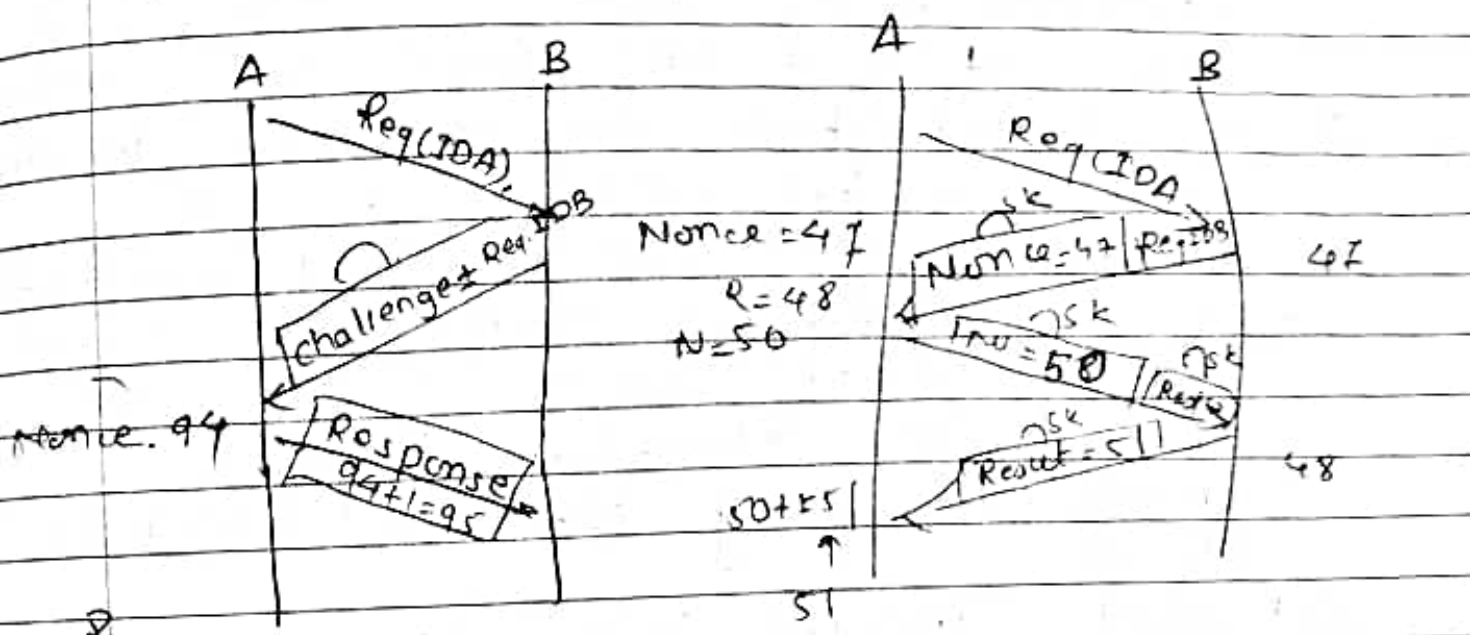
Timestamp Based



Nonce = 47

used of Timestamp is to handle duplicate Request

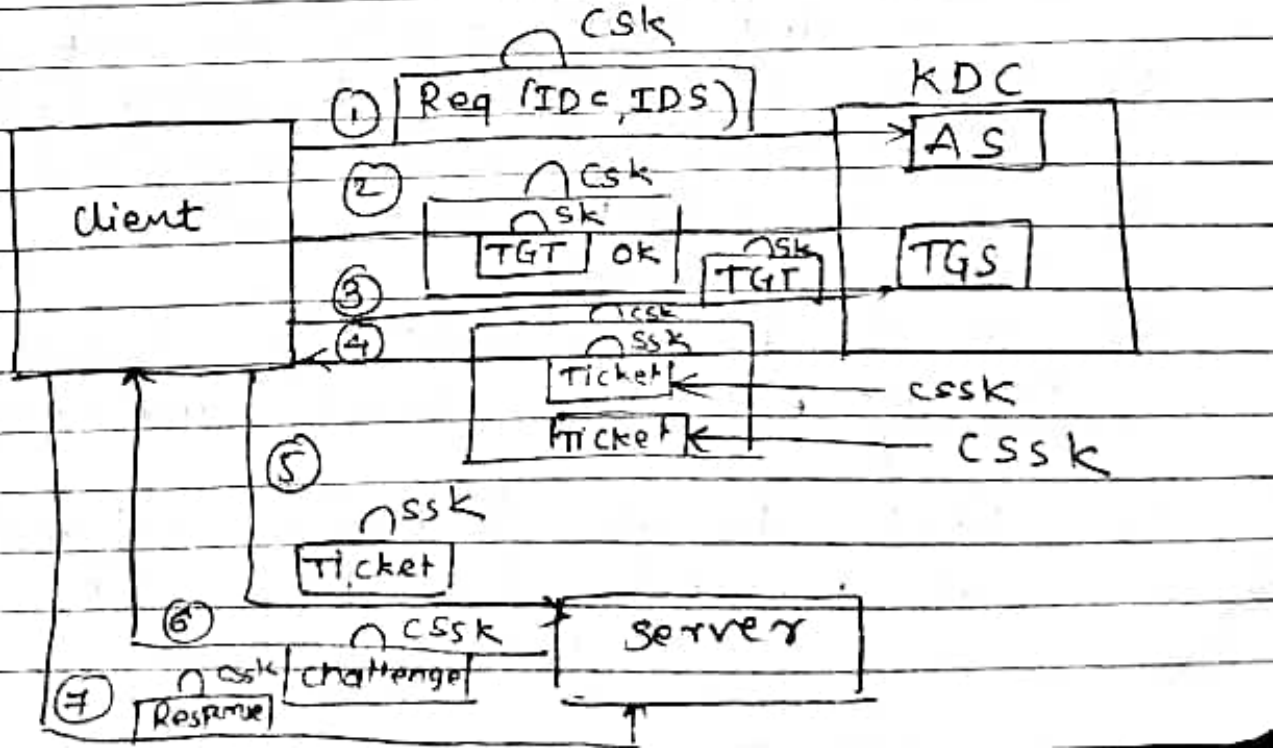
Bidirectional



Kerberos Algorithm

It is 3 level Authentication algorithm

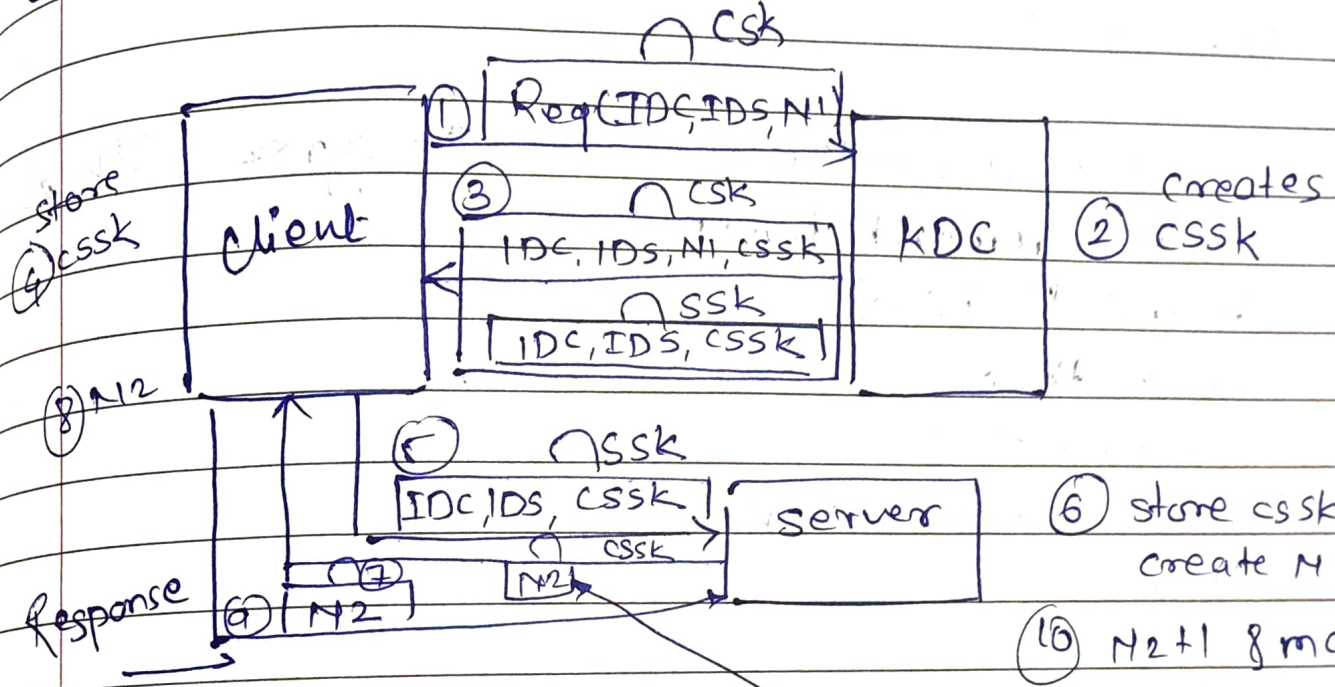
It is derived from a greek empire Security principle which safe guards the empire from 3 diff. types of Attack



1) Initially client create request packet encrypted using client session key (CSK) and send it to KDC (key Distribution center)
2) AS (Authentication service) of KDC decrypt request packet using CSK, if successfully decrypted client is authenticated, AS now creates TGT (Ticket granting Ticket) encrypted using SK along with acknoledgement (ok) encrypted using CSK, and it sent to client.
3) client decrypt this using CSK, gets acknoledgement and forwords encrypted TGT to TGS (Ticket Granting Service)
4) TGS decrypt the TGT using SK (session key), checks whether client has right to request server, if yes it generates client server session key (CSSK), puts that in a ticket (2 copies) 1] encrypted using SSK other using CSK. and both of this are sent to client
5] client decrypt ticket using CSK, get CSSK and forword encrypted to server
6] Server decrypts the encrypted tickets and gets CSSK.
7] server now create a challenge encrypt it using CSSK and send it to client.
8] client decrypt challenge, solves it and give response back to server.

If response to the given challenge correct server Authenticates client successfully.

- Needham Shroeder :-



N1: Nonce of client
N2: Nonce of server

Diffe Hellman algorithm.

This Algorithm creates an opportunity for sharing session key without KDC or even client sharing that with server. Assume there are following 2 parties that wants to communicate hence need session key.

eg A and B where A is sender (client) B is receiver (server)

A	B
Step 1: both will share a common random no. q and q is prime ~~Say~~ $q = 13$	it also agrees $q = 13$
Step 2 Consider a random value common between A and B i.e α such that i] Alpha $< q$ ii] α primitive root q $\alpha = 6$	It agree with $\alpha = 6$
Step 3: Private key generation (x_A) $x_A < q$ Assume $x_A = 5$	Private key generation (x_B) $x_B < q$ $x_B = 4$
Step 4 public key generations y_A $y_A = \alpha^{x_A} \bmod q$ $y_A = 6^5 \bmod 13$ $= 7776 \bmod 13$ $= 598.15$ $= 0.15 \times 13$ $= 1.95$ $= 2$	public key generation y_B $y_B = \alpha^{x_B} \bmod q$ $y_B = 6^4 \bmod 13$ $= 1296 \bmod 13$ $= 99.69$ $= 9.69 \times 13$ $= 8.97$ $= 9$

Generation of sessionkey

$K_A = Y_B^{x_A} \bmod q$

$Y_A = q^{x_A} \bmod 13$

$= 3.$

Generation of sessionkey

$K_B = Y_A^{x_B} \bmod q$

$K_B = 2^4 \bmod 13$

$= 0.23 \times 13$

$= 3$

Stream cipher vs block cipher

I/P 10101101 10100011

↓

Encryption

↓

O/P CT 11001110 01110111

I/P PT 10101101 10100011

↓ ↓

Plain [B1] [B2]

↓ ↓

Enc Enc

↓ ↓

cipher [B1] [B2]

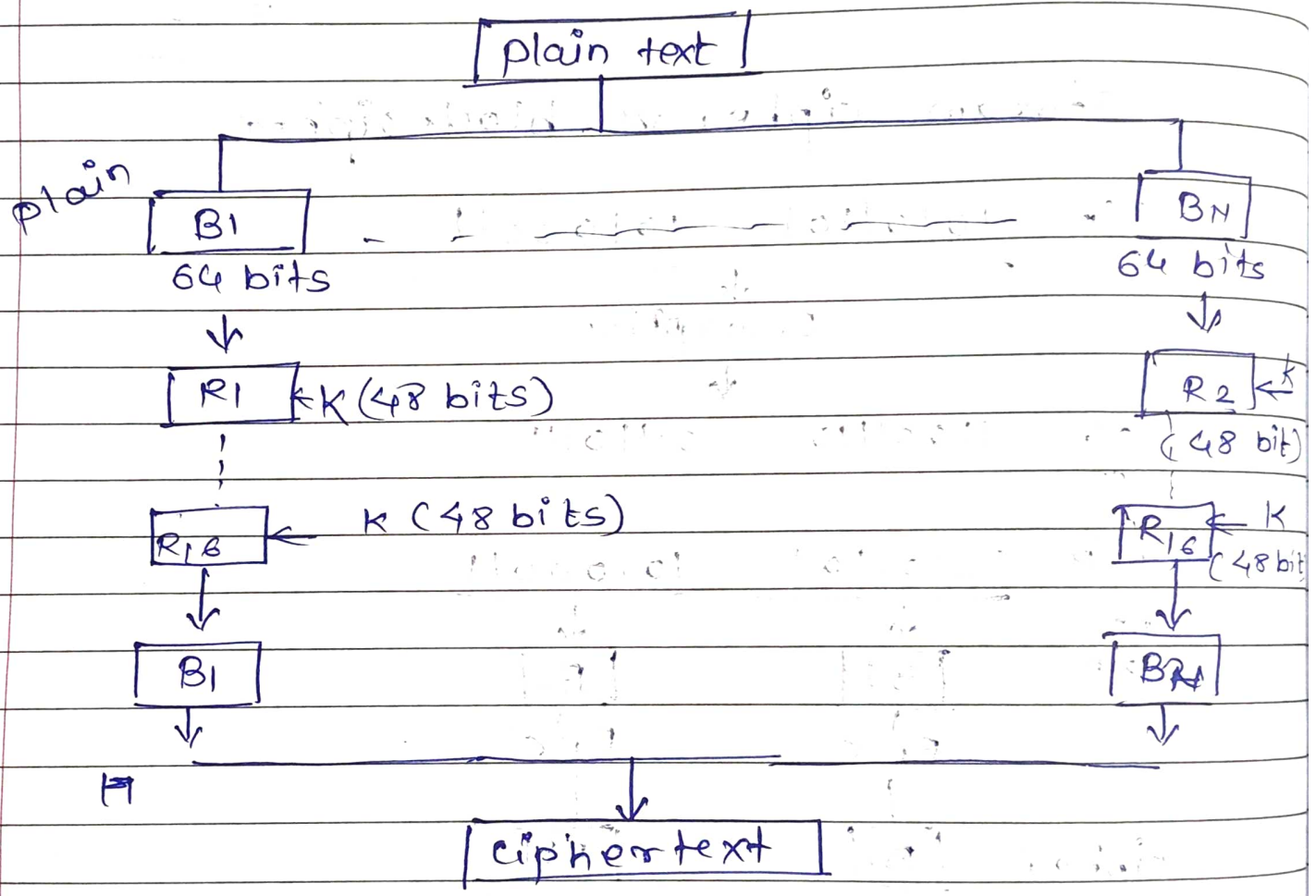
O/P CT 11011101 01001010

Data Encryption Standard

It is a block cipher technique developed by IBM

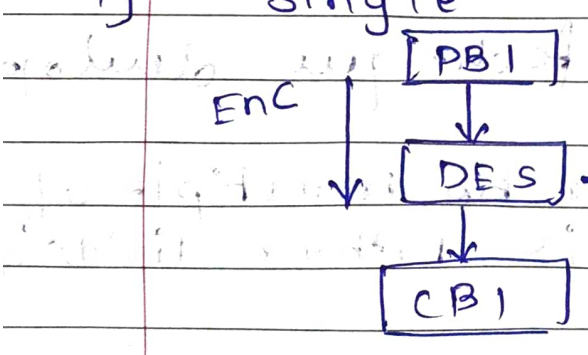
The plain text should be multiple of 64 bits. because it is futher divided into 64 bits.

on every block total 16 rounds of operations are carried with 48 bits of key.
The final crypted blockes are merge together as the final cipher text

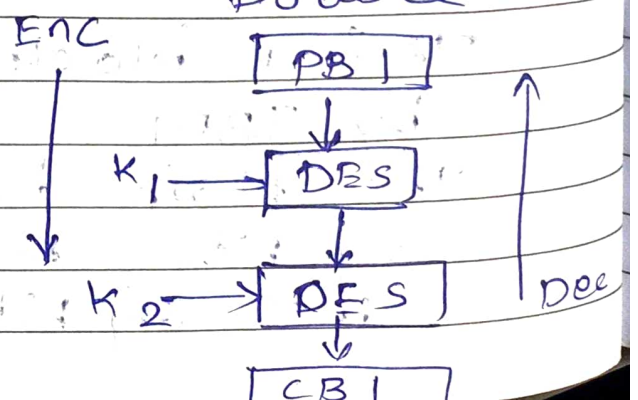


Types of DES

1] Single

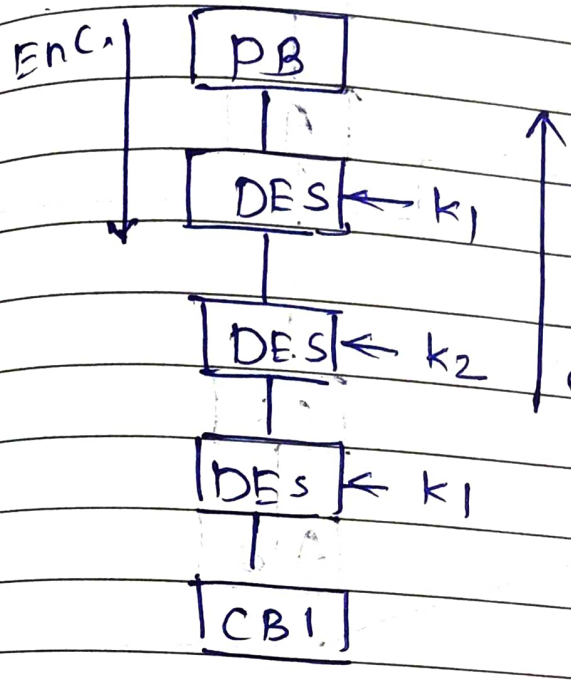


Double

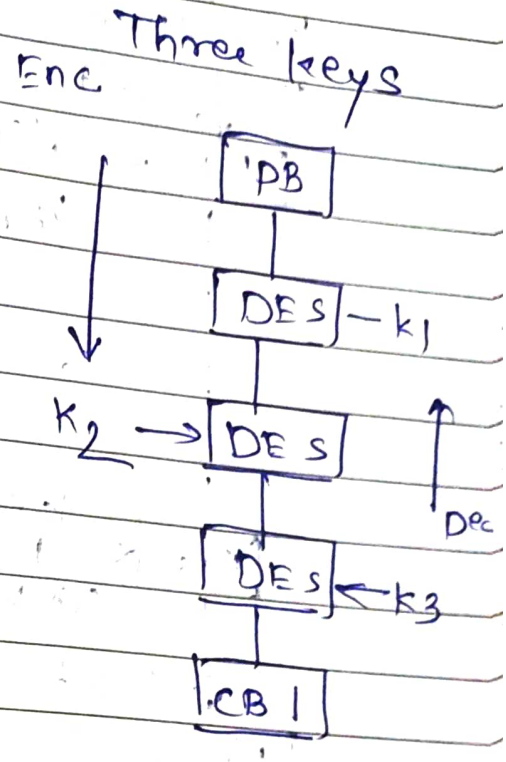


Triple DES

a. Two keys

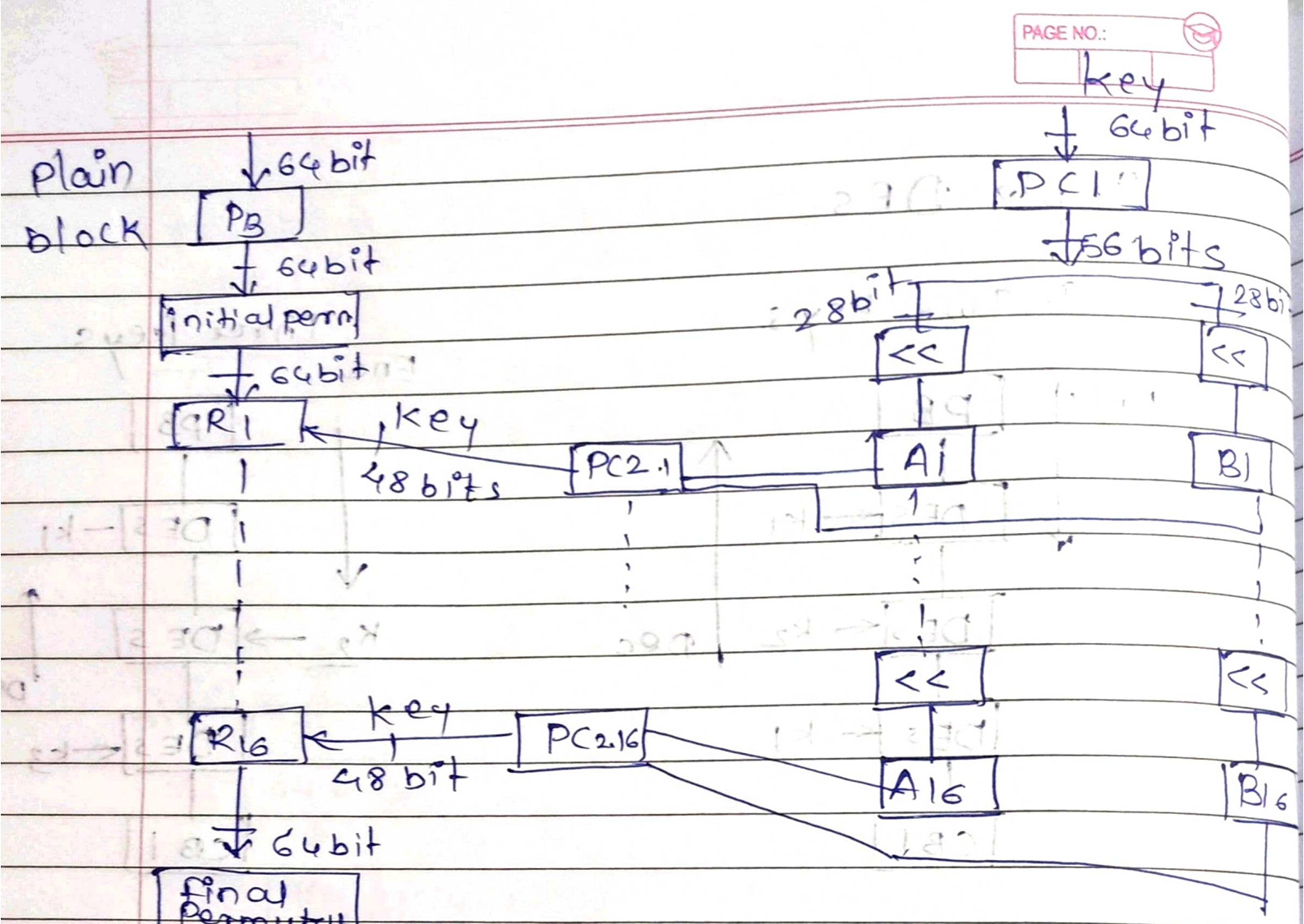


Three keys



block diagram of DES

on every block there is initial permutation which is shuffling of bit now total 16 rounds of operation will be carried.
after final permutation the cipher block will be ready.



PC1 Permuted choice

A1 - A16 & B1-B16 buffer

<< left shift

NOTE :

for 1, 2, 9, 16 L << by 1 bit

for others <<< by 2 bits

Key Generation

Initially the key is of 64 bits which is further compressed to 56 bits by permutated choice compresser. which will remove the every 8th bit

The 56 bits key is divided into 2 parts each of 28 bits and on 28 bits of key perform left shift operation and store that in the left and right buffer. (A and B) resp.

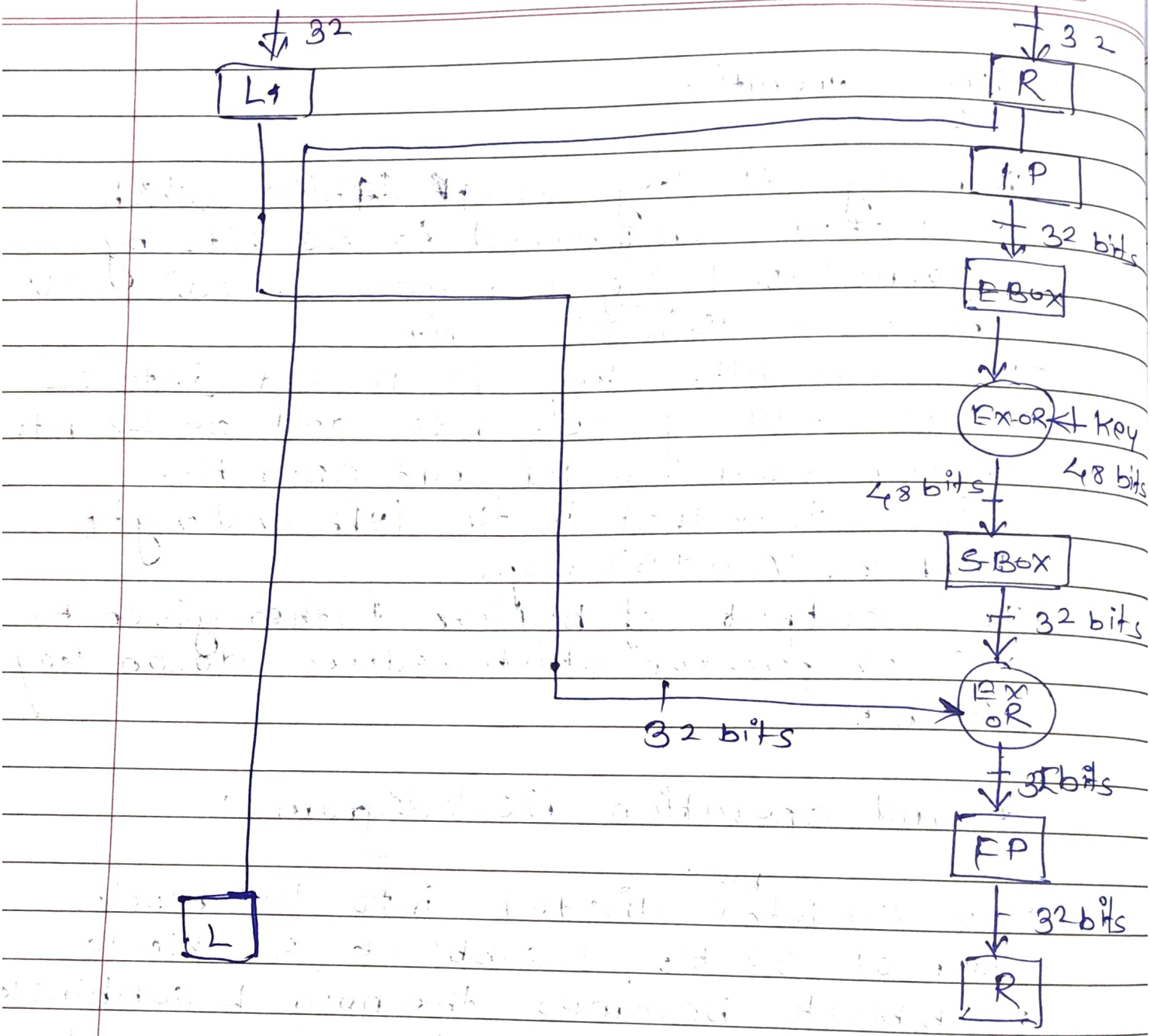
The output of buffers further given to PC compresser which reduce down key 2 48 bits

Round operation block digram :

The 64 bit block is divided into two blocks each of 32 bit L subblock, R subblock

RSub-block becomes the new L subblock but new R subblock following operation are performed

1) Initial permutation on R subblock
2) padding using Expansion box (EBox) (32-48)
3) Ex-or with key (48 bit)
4) compress using S-Box (48-32 bit)
5) Ex-or with L block
6) final permutation


32
L
32
R
I.P
32 bits
E Box
Ex-OR
Key
48 bits
48 bits
S Box
32 bits
EX OR
32 bits
FP
32 bits
R
L
32 bits

Q.1 AES

⟹ Stream Cipher:

1] RC4 Algorithm [Rivest Cipher 4/Ron code 4]

→ Developed by Roth Rive st

→ Steps:-

1] Key Scheduling:-

Algorithm:-

```
j = 0
for i = 0 to (n-1)
{
   j = [j + S[i] + T[i]] mod n;
   swap(S[i], S[j]);
}
```

S = State vector Array
T = Temp vector derived from key array
n = size of State vector array

2] Key stream generation:-

Iterations = size of key array

```
i, j = 0, L = 0
while (true)
{
   i = (i+1) mod n;
   j = (j + S[i]) mod n;
   swap(S[i], S[j]);
   t = (S[i] + S[j]) mod n;
   K[L] = S[t];
   L++;
}
```

Now key obtained will be used for Encryption & Decryption

3] Encryption & Decryption

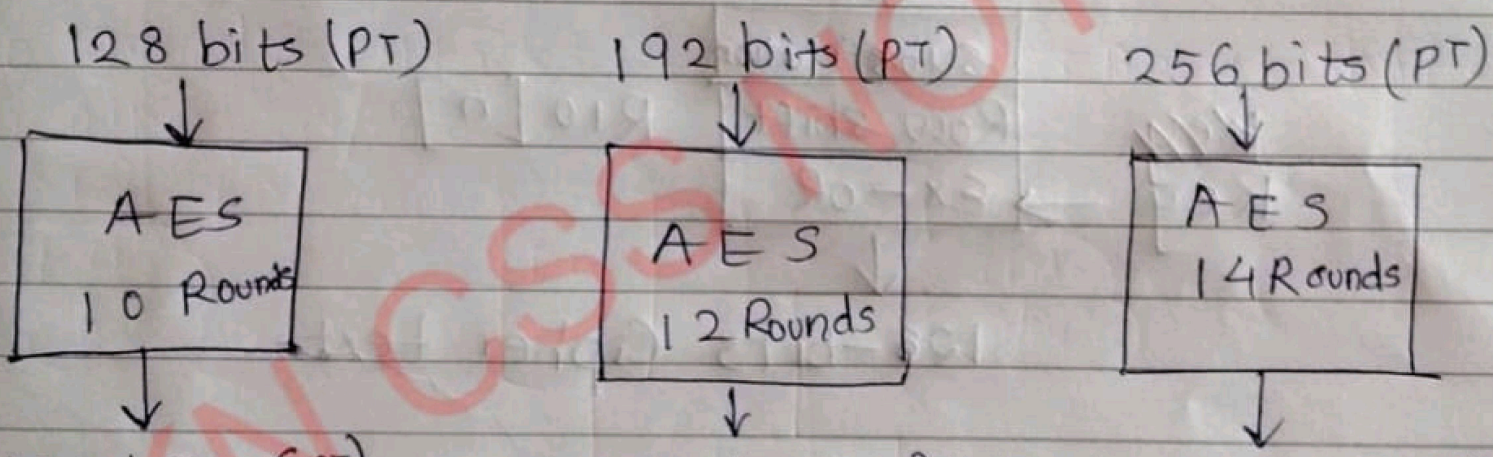
En = C.T. = P.T XOR key (new)

Dec = P.T = C.T. XOR key (new)

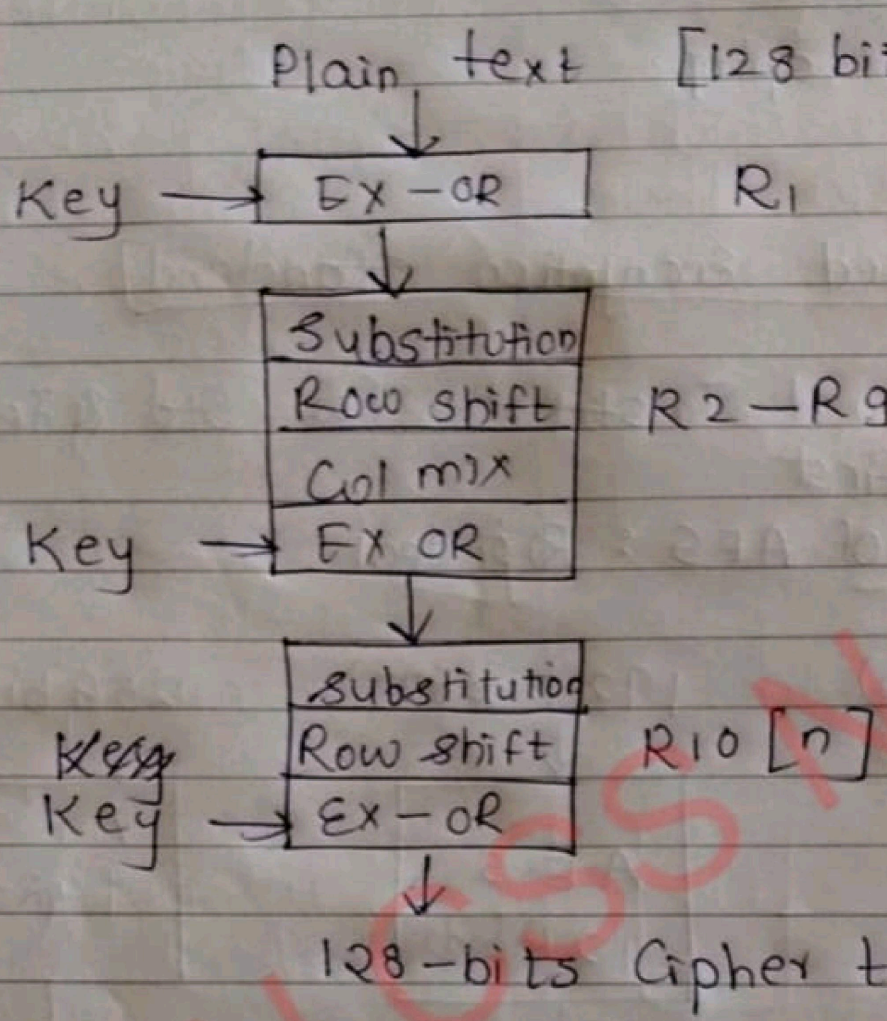
AES [Advanced Encryption Standard]

→ NIST [National Institute for Std & Tech.] published first.

→ Developer of AES : Rijndeal.



→ Block Diag:-



- As shown above, 128 bits of block of plain Text in Round 1 Ex-OR with Key
- From Round 2-9, 128 bits block will be mapped to 4x4 state Array {At a time 16 bits} will be processed with substitution, row shift & mixing of col elements, when all 128 bits are processed they will be EXORed with key.
- For last round only column mixing is skipped.

AES Refer PDF (Blocked level cipher)

RC4 Algorithm

(Algorithm refer notes)

S[] = [0, 1, 2, 3]
key[] = [3, 5]
plain text = [1, 4]

Solve using RC4 algorithm

n = 4 (Size of S[])
T[] = [3, 5, 3, 5]
T array derived from key array but size = S[] (n).

∴ Elements of key[] are repeated

step I :
NOTE : no. of iteration = n
i = j = 0 (iteration 1)
j = [j + S[i]) + T[i]] mod n

S[] = [0, 1, 2, 3] T[]
j = [0 + 0 + 3] mod 4
j = 3

∴ Swap S[i] with S[j]
S = [0, 1, 2, 3]
0 1 2 3

S = [3, 1, 2, 0]

PAGE NO.:

i=1 j=3

$j = [3 + 1 + 5] \bmod 4$

$j = 1$

swap [S[i], S[j]] S[] = [3, 1, 2, 0]

i=2 j=1

$j = [1 + 2 + 3] \bmod 4 = 2$ S[] = [3, 1, 2, 0]

i=3 j=2

$j = [2 + 3 + 5] \bmod 4$ S[] = [3, 1, 2, 0]

step 2 : key stream Generation

size of key array = 2

2 iterations

	0	1	2	3
S	3	1	2	0

	0	1	2	3
T	3	5	3	5

	0	1
K	3	5

iteration 1

$i = 0, j = 0, t = 0$

$i = (i + 1) \bmod n$

$i = (0 + 1) \bmod 4$

$\boxed{i = 1}$

$j = (j + S[i]) \bmod n$ S

	0	1	2	3
S	3	1	2	0

$j = (0 + 1) \bmod 4$

$\boxed{j = 1}$

Swap S[i] with S[j]

	0	1	2	3
	3	1	2	0

i j

$T = (S[i] + S[j]) \bmod n$

$= (1+1) \bmod n$

$t = 2$

$K[L] = S[t]$

$K[0] = 2$

$L = 0$

iteration 2

$i = 1, j = 1, L = 1$

$i = (i+1) \bmod n$

$i = (1+1) \bmod 4$

$i = 2$

$j = (j + S[i]) \bmod n$

$j = (1+2) \bmod 4$

$j = 3$

swap $S[i]$ with $S[j]$

3	1	0	2

$T = 2$

$K[L] = S(t)$

$K[1] = S[2]$

key

2	0
0	1

Step 3: Encrypt (plain text into cipher text)

00000001 00000100

XOR

00000010 00000000

00000011 00000100

C.T = 3 4

Knapsack Algorithm Publickey & Private ke

→ It is an asymmetric key algorithm which is similar to RSA it is even develop before RSA and con's
I] Public key hard knapsack
II] Private key is easy knapsack

step 1: creation of private key D (6 digits)
Randomly select 6 no. in ascendingord
d = {1, 2, 4, 10, 20, 40}
select a random variable m such that m is greater than sum of all elements in d
sum of elements in d = 77
Assuming m = 110

step 3: select a random variable m such that there is no common factor in m, n
n = 3

step 4: Creation of public key (e)
$E_i = [d_i * n \bmod m]$
$E_1 = [1 * 3 \bmod 110] = 3$
$E_2 = [2 * 3 \bmod 110] = 6$
$E_3 = [3 * 3 \bmod 110] = 12$
$E_4 = [4 * 3 \bmod 110] = 80$
$E_5 = [5 * 3 \bmod 110] = 60$
$E_6 = [6 * 3 \bmod 110] = 98$

steps: for Encryption $CT = \sum_{i=1}^{n} PT * E_i$

where

step 6: for Decryption

Q. find the cipher text for the plain text (the plain text should be multiple of keys)

PT = { 110011000110 }

E = { 3, 6, 12, 30, 60, 10 }

CT1 :

	1	1	0	0	1	1
*	3	6	12	30	60	10

$3+6+0+0+60+10$

$CT_1 = 79$

CT2 :

	0	0	0	1	1	0
*	3	6	12	30	60	10

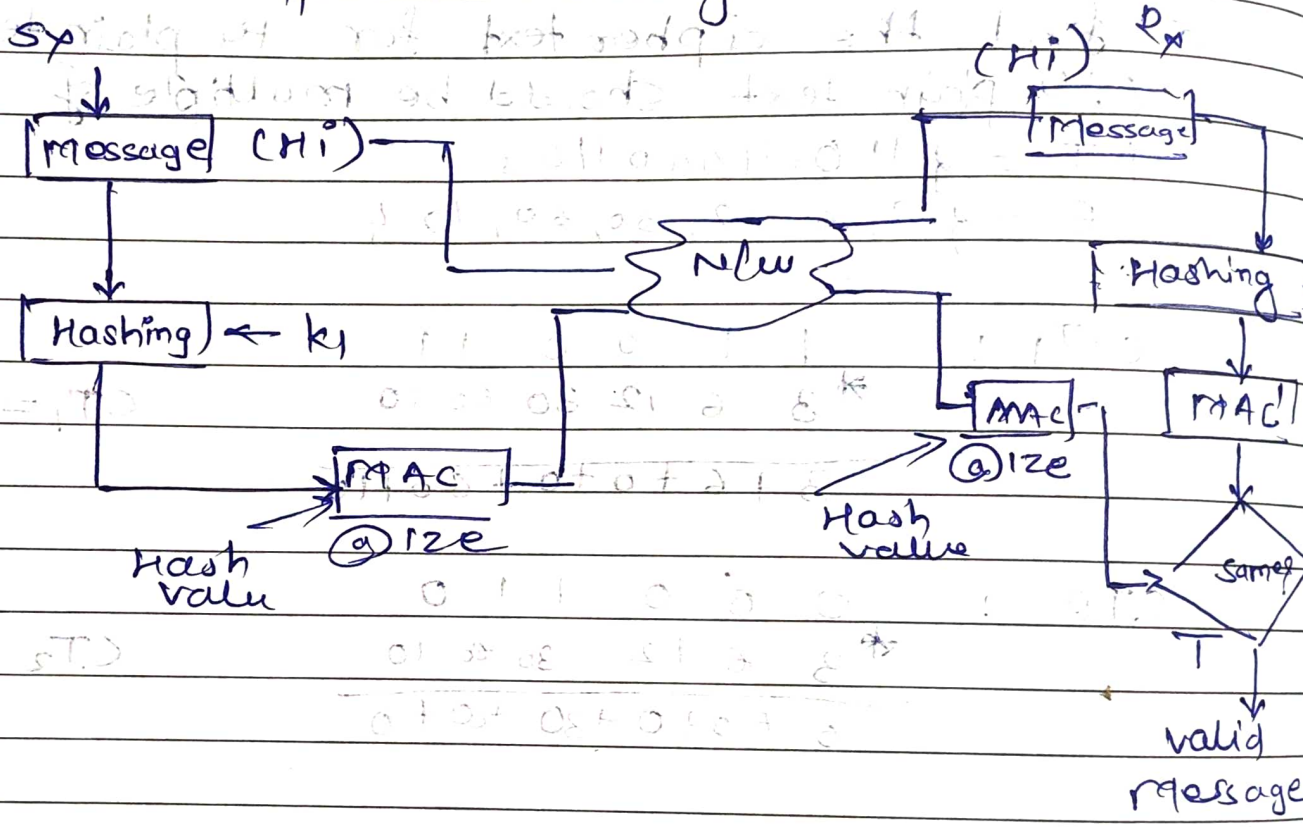
$0+0+0+30+60+0$

$CT_2 = 90$

MAC VS HMAC VS CMAC

These are the ways to authenticate the message to check the integrity

MAC
(Message authentication code) this is applie on simple text message



Algorithm :

step 1 : Apply hashing with a key on the message which will give MAC (Hash value)

step 2 : Send MAC and message

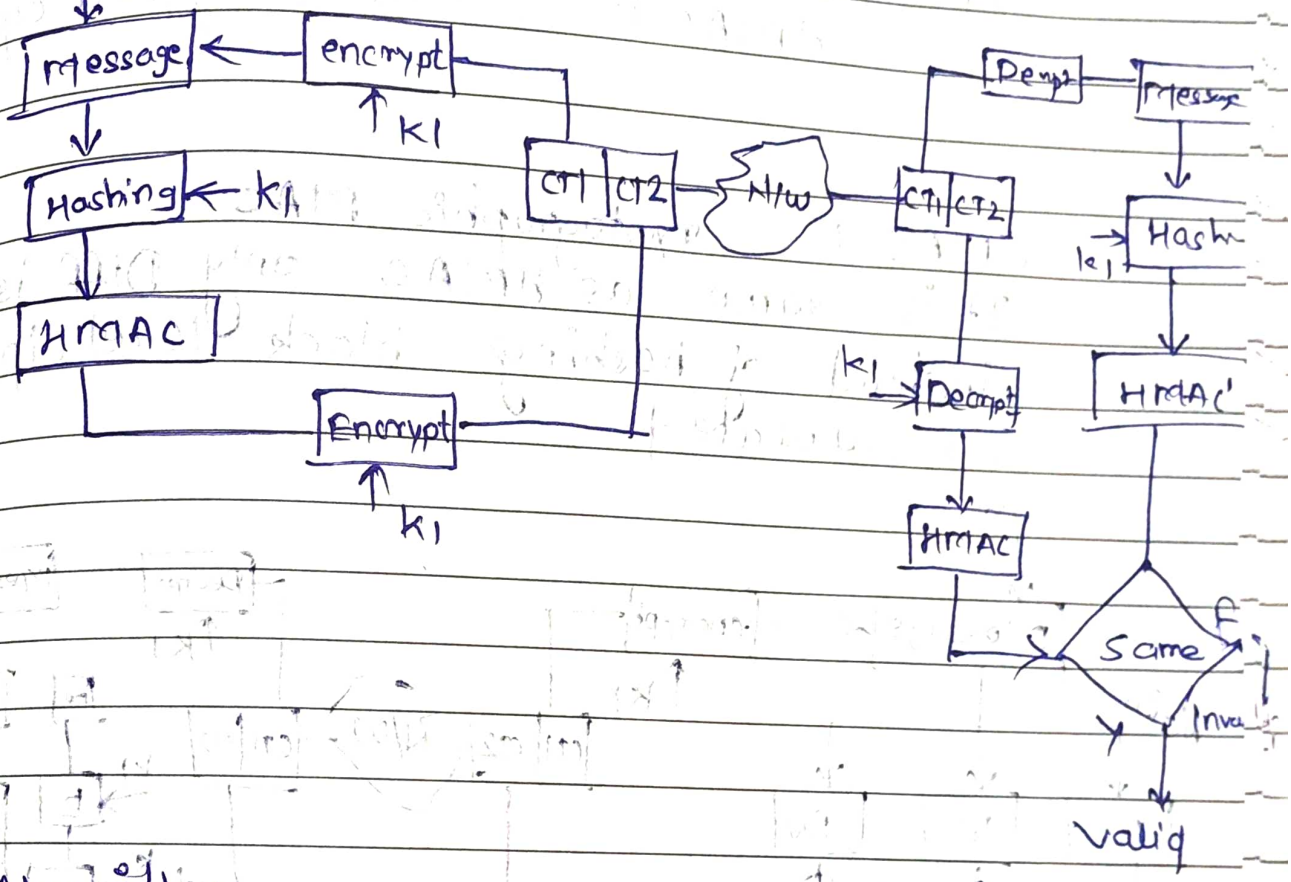
step 3 : Receiver gets message and MAC

step 4 : Receiver apply same hashing algo with same key as applied by sender

step 5 : Reciver now gets MAC', MAC and MAC' should be same

HMAC

It is the advance version of MAC, which can be applied on any kind of data and is more secured.



Algorithm:

Step 1: Apply hashing with a key on the message which gives HMAC (Hashval)

Step 2: encrypt message and HMAC using same key.

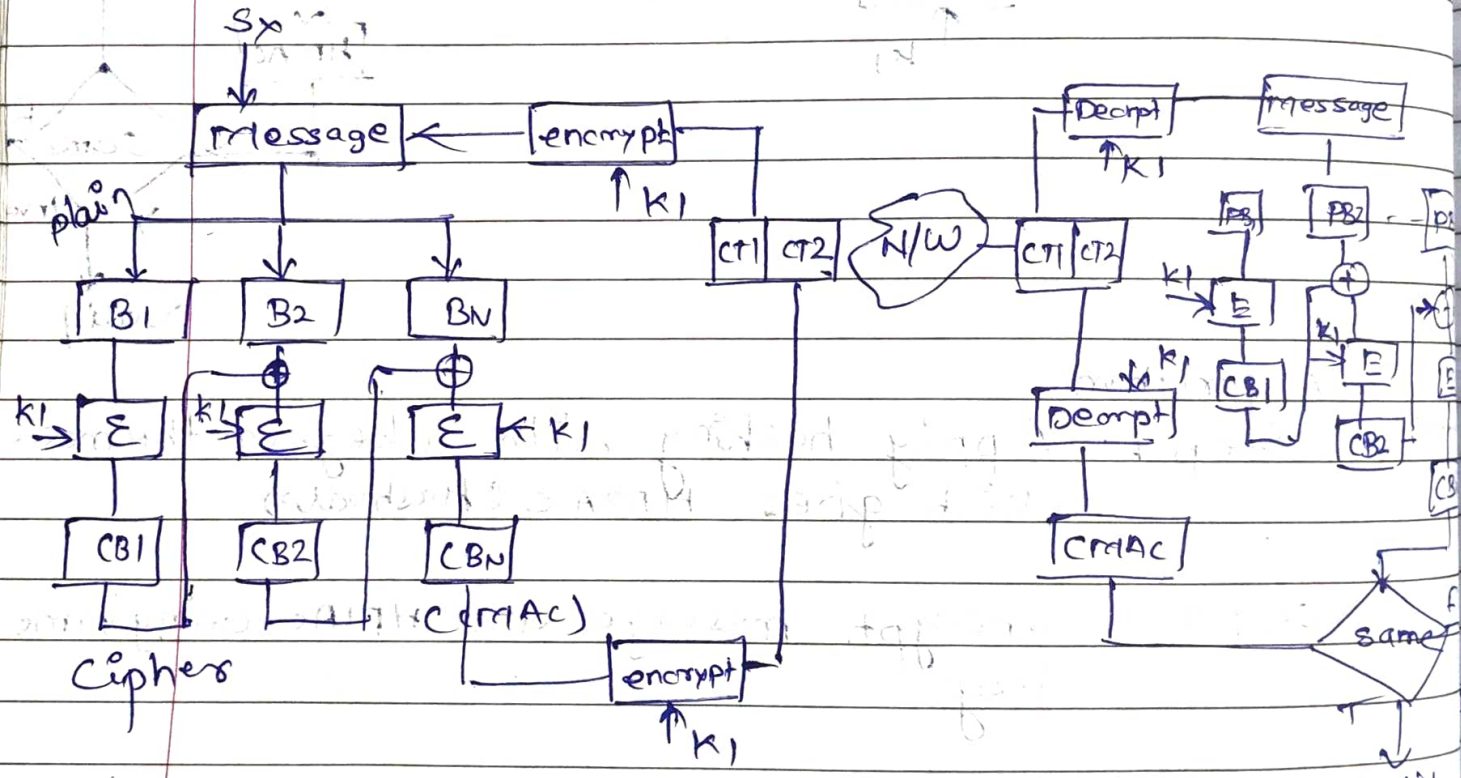
Step 3: send the 2 cipher text encrypted message & encrypted HMAC

Step 4: receiver receive 2 cipher text, decrypt both of them using same key used for encryption

step5: Apply hashing on ~~decrypted~~ message (same hashing & key asused by sender) this gives HMAC'

step6: HMAC & HMAC' should be same. (HMAC is second decrypted cipher text)

CMAC: Cryptographic MAC
It is same as HMAC only Diff is instead of hashing block level cipher is calculated



PAGE 2

Detailed Syllabus:

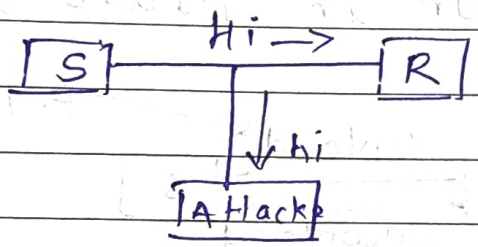
Authentication (entity, Nonce)
Kerberos
Needham schroder
Diffie Hellman
block cipher & stream cipher (

Key Generation (Knapsack sum)
Decryption
MAC, HMAC, CMAC
(No RSA)

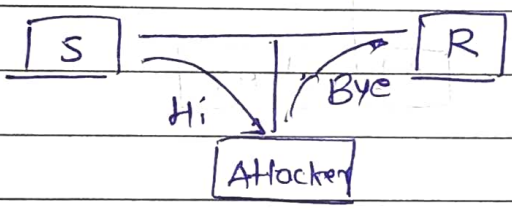
ke. Network security and attacks
Passive attack vs active attack

Passive Attack

Interception:



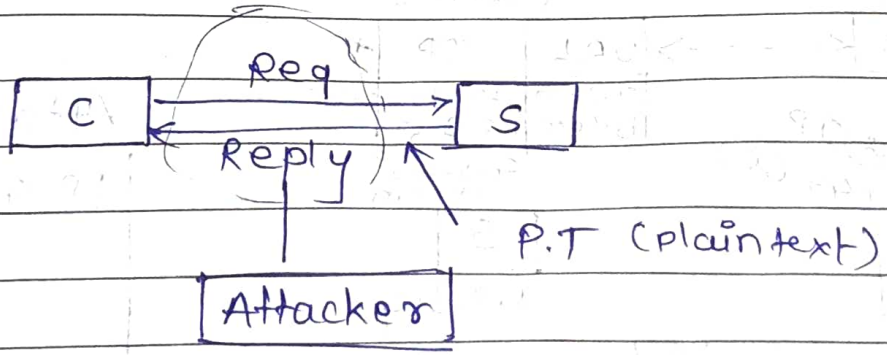
Active Attack :



~~DST~~ TCP/IP model Vulnalabilities

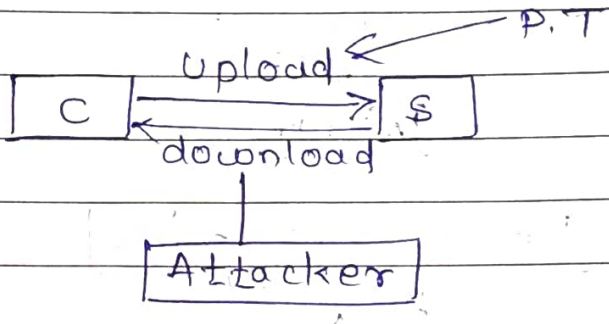
Application layer Attacks :

HTTP attack (Packet Sniffing)
any data send using HTTP goes in plain text format and hence there is possibility of man in middle attack



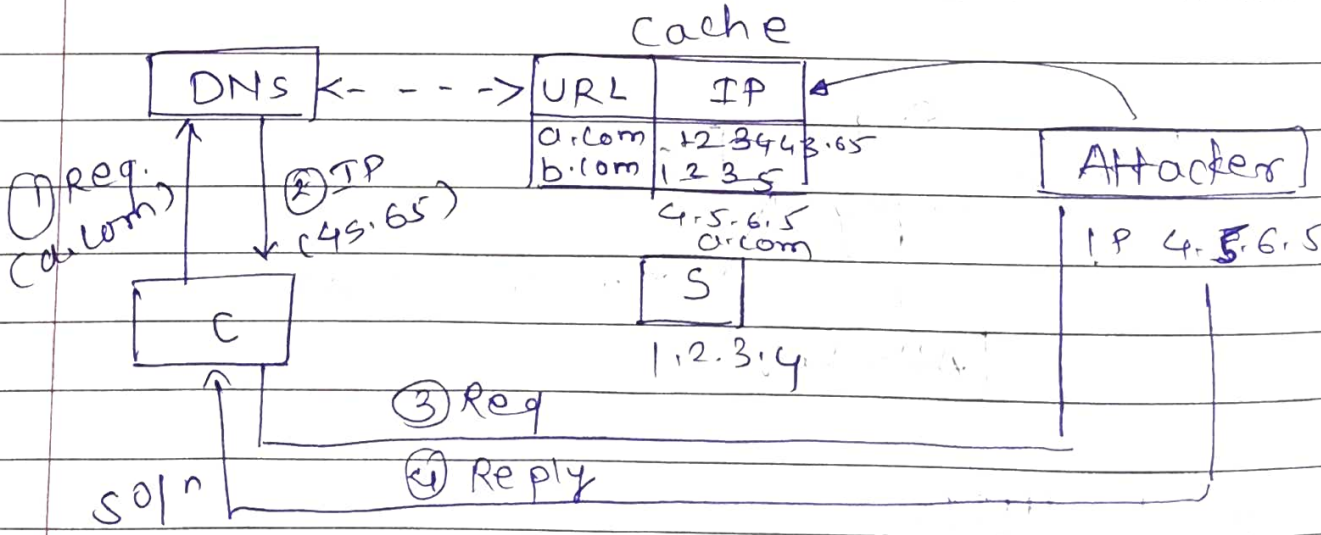
used HTTPS

FTP Attack (packet sniffing)
Any file uploaded & downloded using FTP. It performed using plaintext transfer process so there is possibility of man in middle attack.

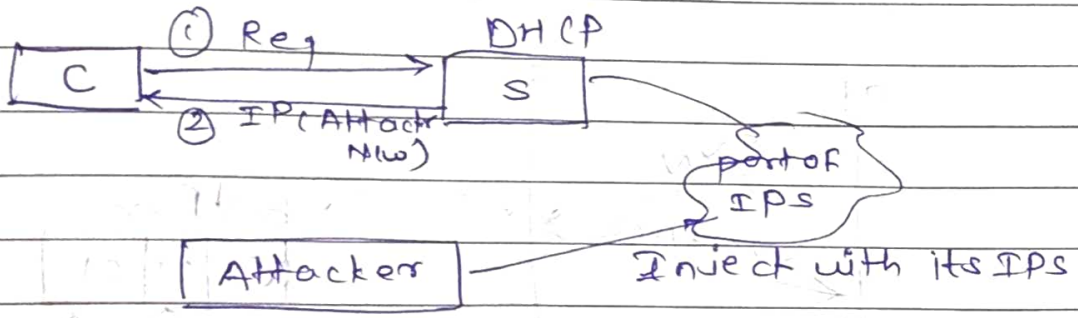


soln using SFTP

DNS poisoning
Attacker gains control over a DNS, modifies DNS cash by injecting its own set of IP address so when any client send request to DNS, it always get response ~~from~~ As IP of attacker.

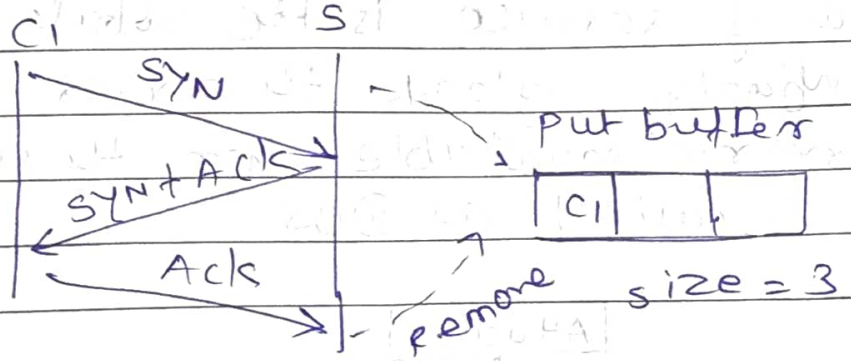


DHCP Spoofing.
Attack gets control over the DHCP server inject its own set of all address in the pool so when any client request for IP, it will always get IP from attackers network.

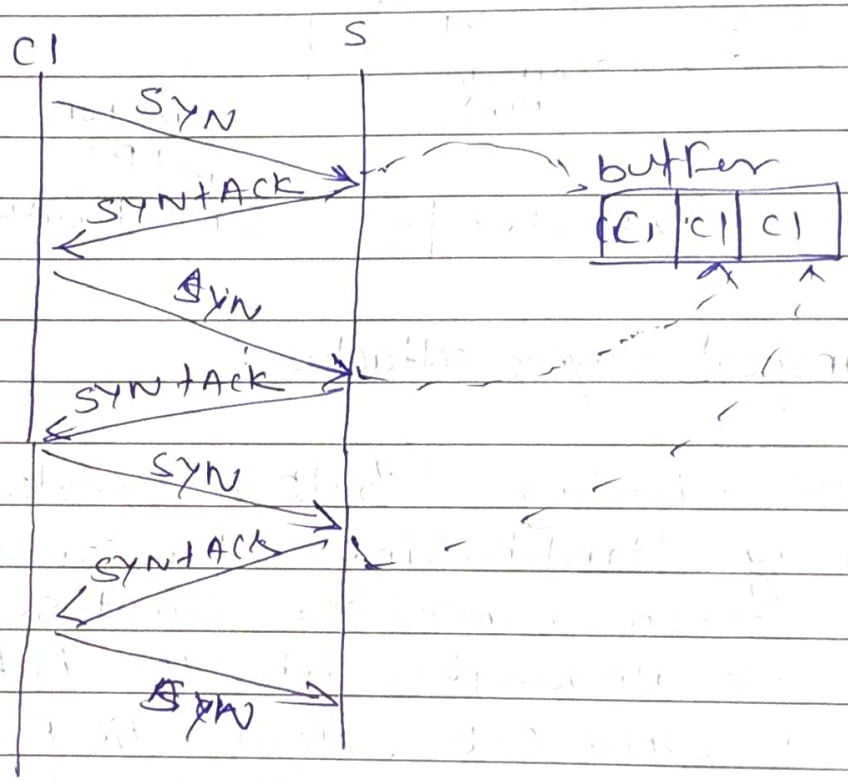


Transport Layer attacks:

SYN Flooding / Buffer Overloading
3 way Handshaking for connection establishment
As shown whenever client send SYN server enques this into the buffer and sends back SYN + Ack. Now client when sends Ack server removes client ~~the~~ details from the buffer.

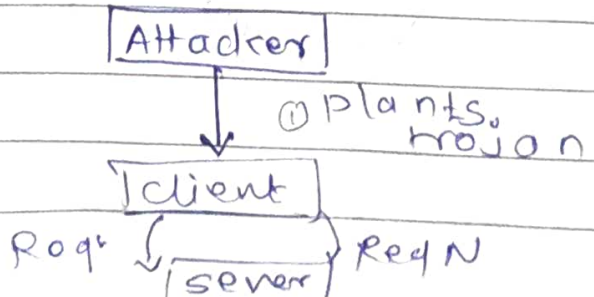


In case of an attack attacker keep sending SYN but never send the final ack which causes Attacker to intentionally overflow the buffer at the server ... now server will not be in position to accept request from any client thus leading to DOS (Denial of Server)



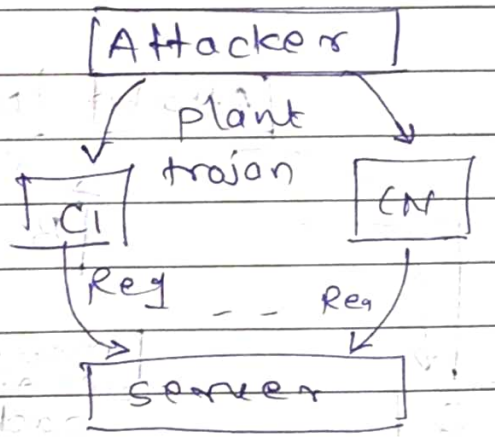
DOS and DDOS

Denial of service is the attack which intentionaly floods the server and make server unavailable for the service this is called as DOS



attacker plants a trojan (a virus file) called as zombie on any client and ~~with~~ attacker gives signal to trojan it start flooding server with multiple request thus causing server to ultimately crash

Soln: server will block the client attacker now plants the trojan in client and whenever it gives signal all those client send request to server ultimately causing the server to crash. now it is difficult to server to block because the DOS attack is distributed



(5) secure socket layer / TLS transport layer Secure / HTTPS working.
(same as ~~DA~~ verific & creation)

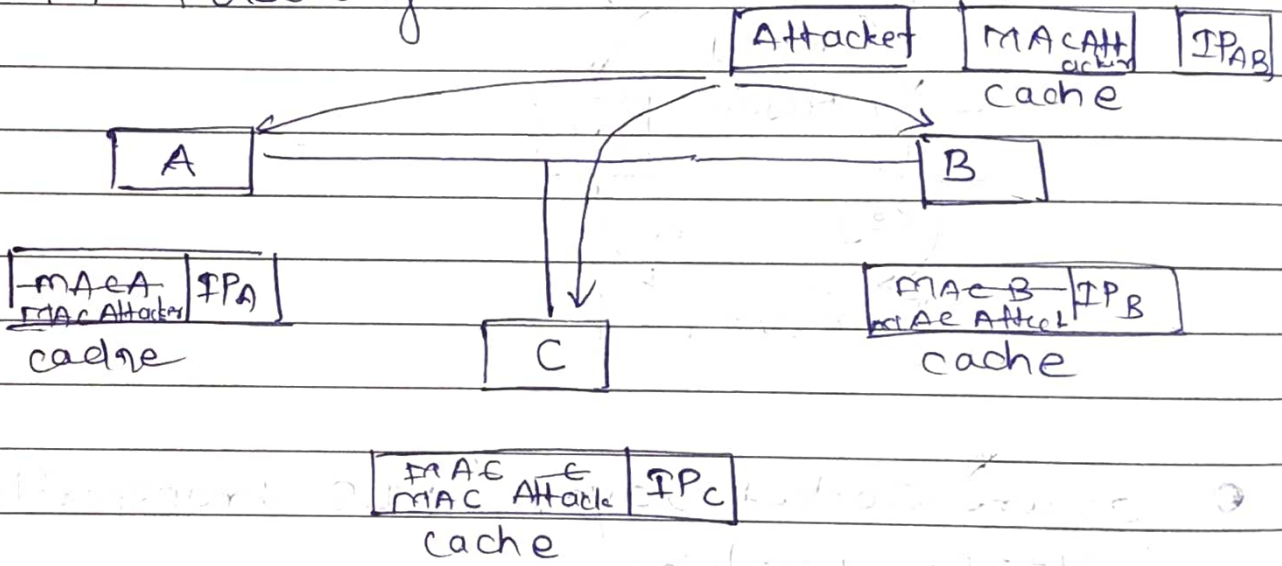
FireWALL

NETWORK LAYER ATTACK

1) IP spoofing (stealing)
The process of steabing IP address of any machine is known as IP spoofing

2) ICMP flooding (Ping of Death)
The process of continously sending ping to the server until the sever becomes overloaded and it crashes is called as ICMP flooding.

3) ARP Poisoning



Address Resolution protocal is responsible to give mac address for given IP address of the receiver.
As shown above attacker gains control over the machines and intentionally replaces mac address of machines in the ARP cache table by its own mac address so now ARP requst is made in reply it will be always getting attackers mac

Malicious Codes

1] virus
2] worms
3] Trojan
4] Logic bomb
5] Adware Malware
6] Spyware

} SELF STUDY

FIREWALL

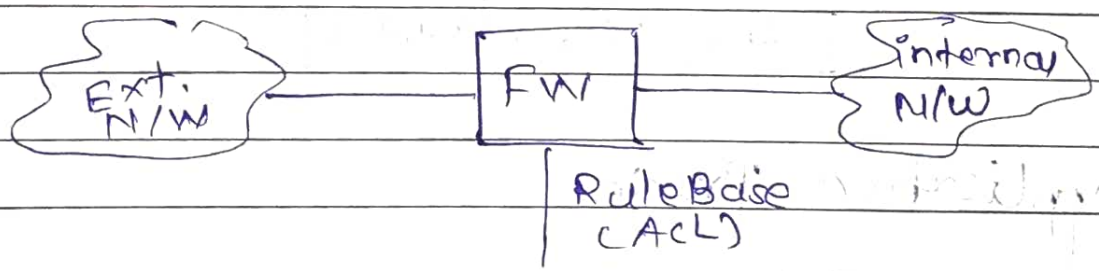
It is a security tool use for protecting a internal private networks from the external one.

Packet ⇒ examine out & incoming network packets based on rules to either forward / discard

Stateful inspection: monitor the state of active connections

Application proxies: provide most secure type of data connection because they can examine every layer of comm n including app. data

Packet filtering



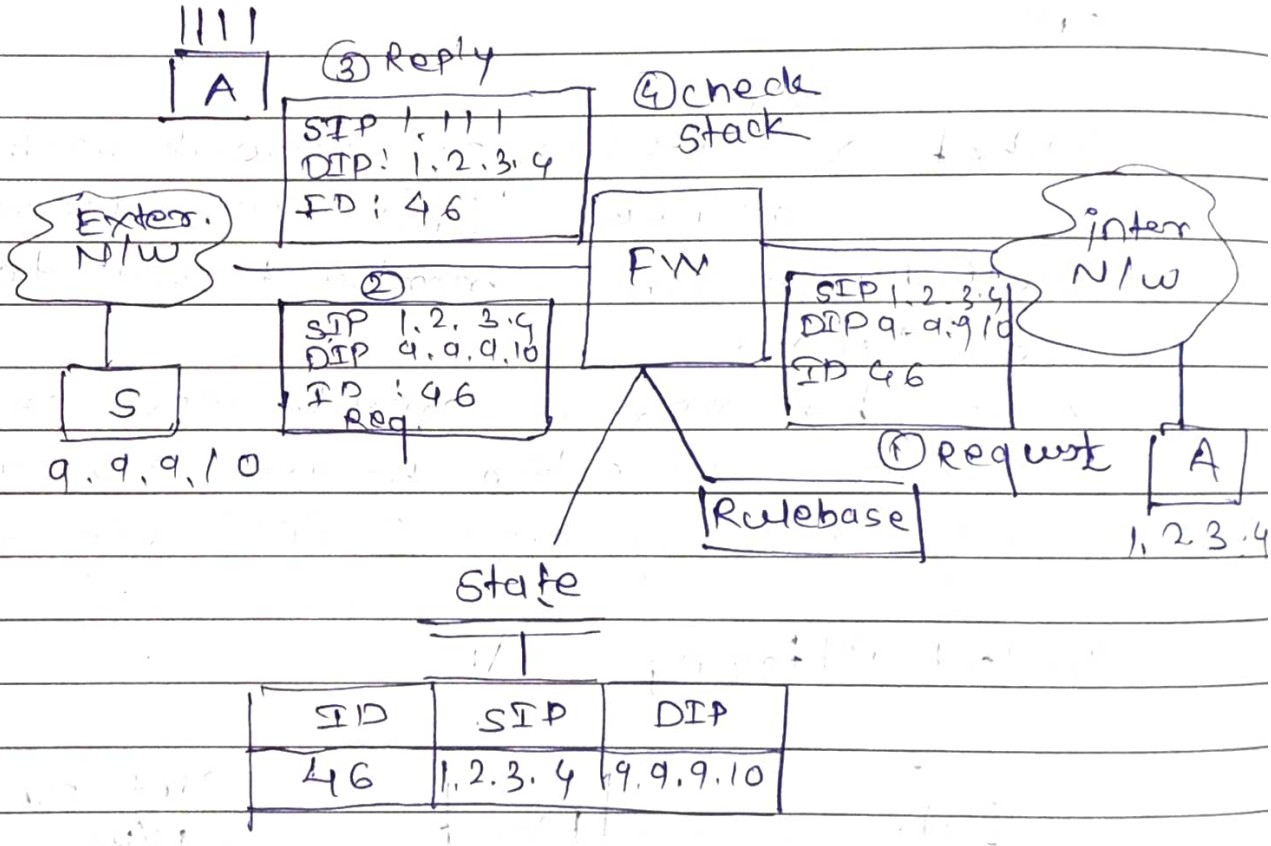
Direction	SIP	DIP	Sport	Dport	Protocol	Action
IN	Any	1.2.3.4	Any	80	HTTP	DROP
OUT	Any	9.9.9.9	Any	Any	Any	DROP

5th3 MAC HMAC CMAC
SHA MD5 VS SHA1 SHA2, SHA3
OSI Layer attacks
DOS & DDOS FIREWALL (FDS, IPS) Honeypot
~~SQL Injection~~ ESE

IN: INBOUND
OUT: OUTBOUND

In packet filtering Examine outgoing & incomin network packets based on rules to either forword / discard.
Problem: If there is attacker then SIP/DIP might change, which can cause problem as firewa

2] Stateful Inspection Firewall

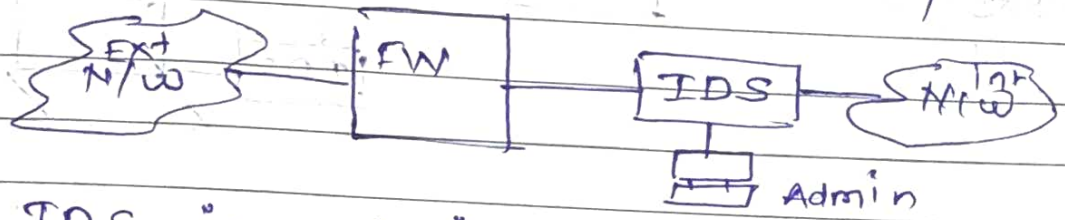


ID	SIP	DIP
46	1.2.3.4	9.9.9.10

3] Application Proxies

PAGE NO.:

- IDS : Intrusion Detection System :



IDS is a device that sits between firewall & internal network to analyze the details of packets which are missed by firewall.

Types :

1) NIDS (Network IDS)
IDS that monitor the entire network.

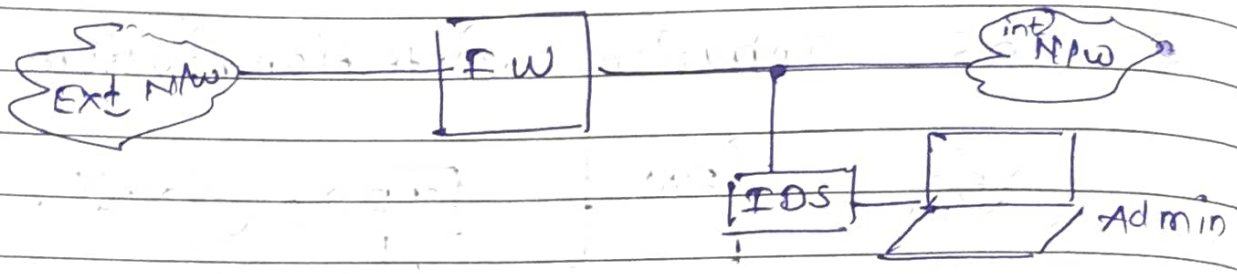
2) HIDS (Host IDS)
IDS that monitor specific Host in the network.

working principle
If IDS finds anything suspicious in the packet, only inform the admin to take action.

1) Signature IDS
IDS maintain the Database of various attacks signature & when encountered any of the signature in the packet it informs admin.

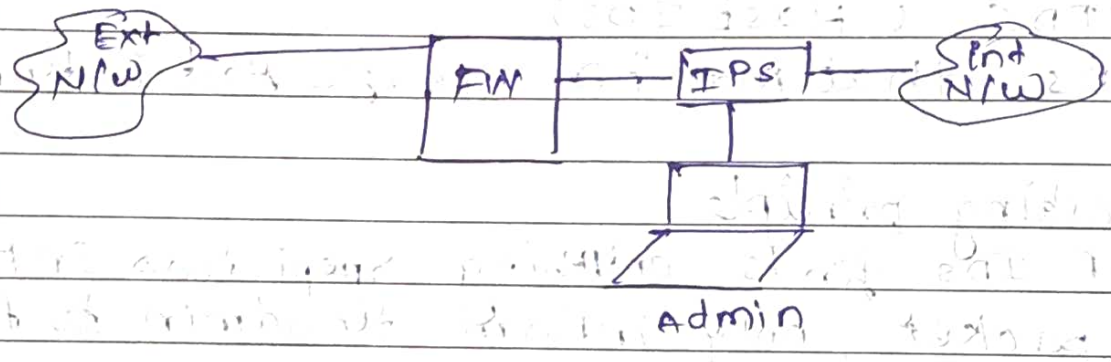
2) Anamoly / behavioral / heuristic IDS
It analyze everyday action of particular employee. & If it encounter some new behaviour or activity it informs admin.

Stealth mode IDS :



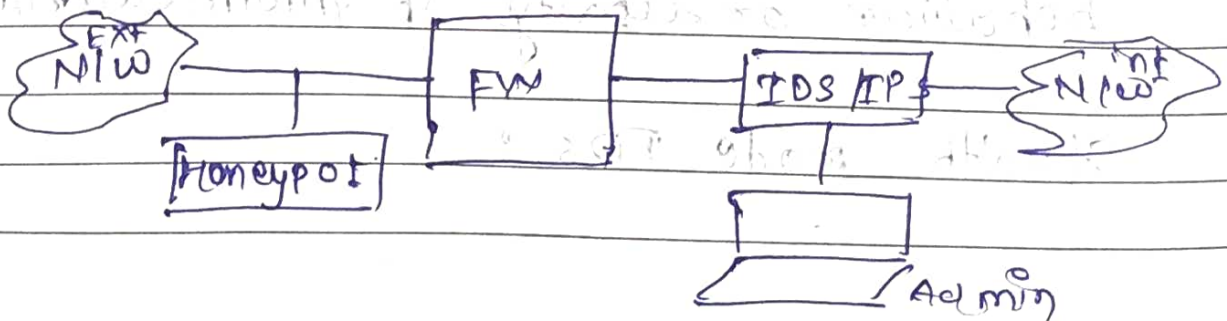
IPS Intrusion Prevention System

It is very much similar to IDS but it is more active in a sense it will block the packet then inform the admin.



- Honeypot

It is device that mimics ~~it is a device~~ like the legitimate server but it is only showcasing limited information and is made to sit outside the firewall to collect all the threat vectors to analyze who is attacking, what is path, etc.



MAC HMAC CMAC basic diff.

stream ciphers block ciphers

PAGE NO.:

diff

MD5, SHA (refer pdf)

SQL Injection :

Select * from Users WHERE UserName='ABC' AND Password = 1234;

this query will fetch the entire database table of users and will display record of user ABC whose password is 1234.

this indicate query will only execute if where clause returns true.

SQL injection is the process of intentionally injecting malicious logic for fetching the data. in unauthorised manner. eg

select * from users where username='ABC' AND password ='1170=1 = 7 1

logic entered in where clause will always return 1. & will fetch details of ABC without the need of entering password.

How to avoid SQL Injection :

31/03/2022

⇒ SHA [Secure Hash Algo] :-

→ Hashing algo.
→ Published by NIST [National Institute of Stds & Technolgy]
→ O/p is fixed of 160 bits.

Step I] Step II] Same as MD5

Step III] Same as MD5

Step IV] Initialize Chaining var i.e. Buffers. There are 5 buffers each of 32 bits.

Total buffer length = 5 x 32 = 160 bits.

A = a
B = b
C = c
D = d
E = e

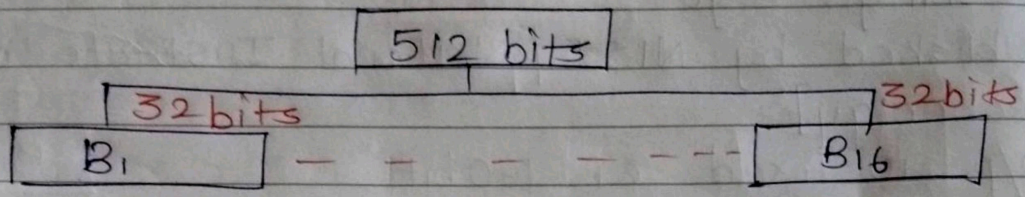
} Variables or Buffers.

Step V] Each Block of 512 bits will go for 4 rounds & every round has total 20 operations.

∴ Total Operation performed in Single Block = 20 x 4 = 80 bits

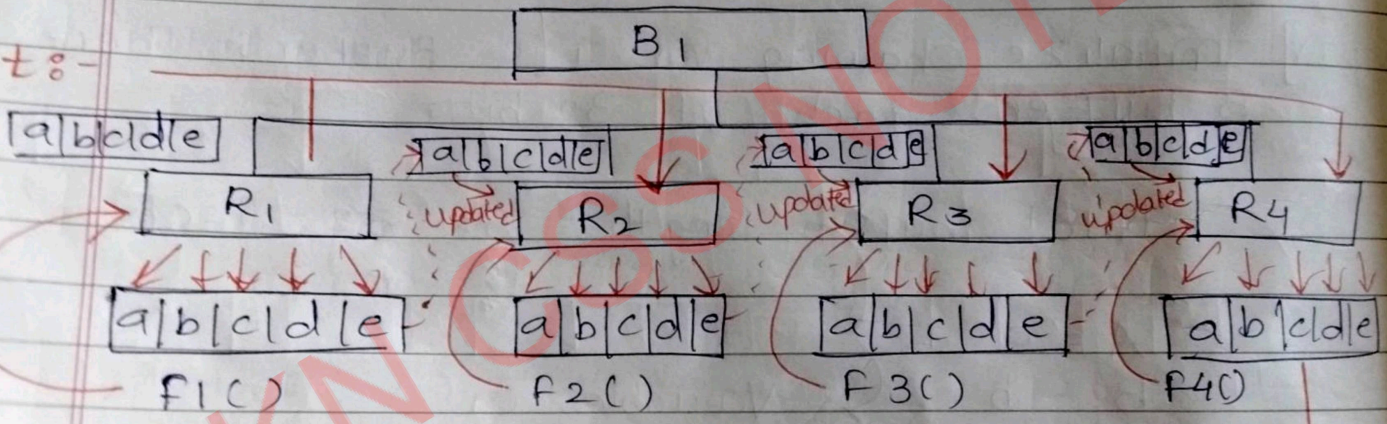
NOTE: All rounds have different operations

Step VI] Block Diag: Now 512 bits Block is divided into 16 blocks each of 32 bits.

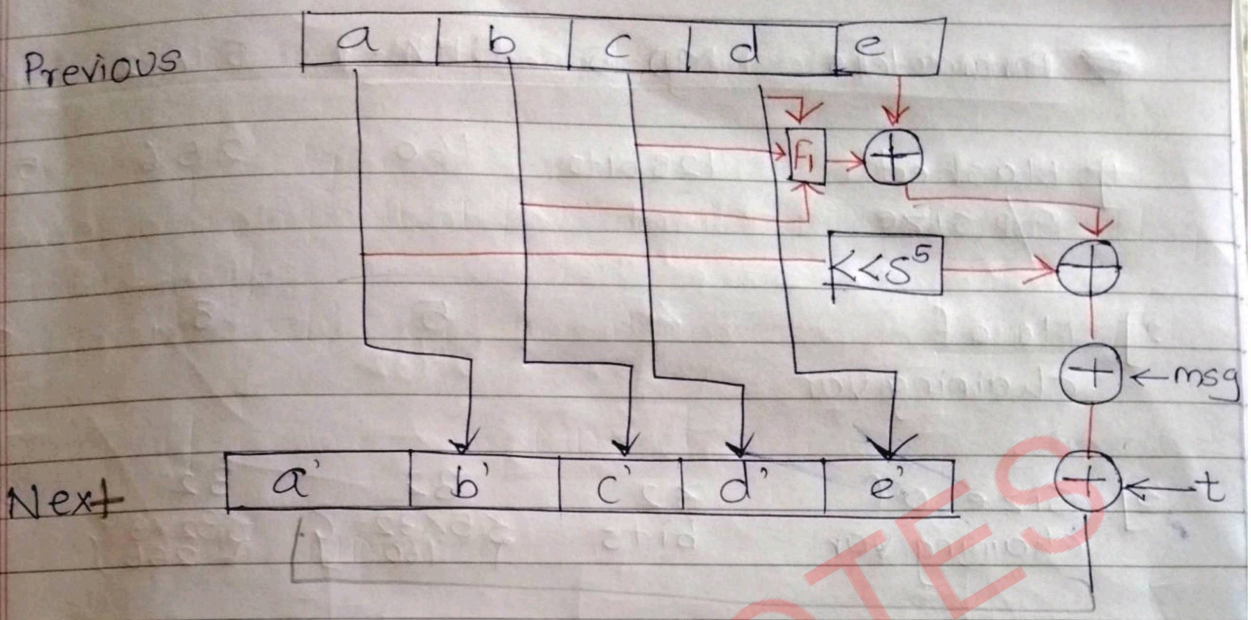


Now on each block 4 rounds of oper^{n} will be performed where chaining var will be updated by some Func(x)

t:-



Final Hashed string [SHA]



Operation performed on every Round:-

$b' = a$
$c' = b$
$d' = c$
$e' = d$

$$a' = \left[e \oplus f_1(b,c,d) \oplus [<<S^5(a)] \oplus msg \oplus t \right]$$

Where $\oplus$ → XOR

⇒

Parameters	MD-5	SHA-1	SHA-2	SHA
1] Hashed o/p size	128 bits	160	256	512
2] No. of chaining var	4 (128/32)	5 (160/32)	8	8
3] Size of chaining var	32 bits	32 {5×32 = 160}	32 {32×8 = 256}	64
4] Total Rounds in each block	4	4	4	4
5] Total oper^n per round	16 {4×4}	20 {5×4}	32 {8×4}	32 {8×4}
6] Total operation	64 {4×16}	80 {4×20} (Rounds × oper^n)	128 {4×32}	128 {4×32}

⇒ Message Digest - 5

→ Hashing algo
→ Ron Rivest is developer
→ o/p is fixed of 128 bits

I] Accept msg & do padding.

message	Padding

i.e. Total len of msg + padding + 64 = multiple of 512

For padding :-

Total Length of bits = should be multiple of 512 less 64

Eg: message = 1000 bits

1] 512 x 1 = 512 ✗
2] 512 x 2 = 1024 ✗
3] 512 x 3 = 1536 ✓

∵ msg bit is only 1000 & we would subtract 64 ∴ can't take 1024

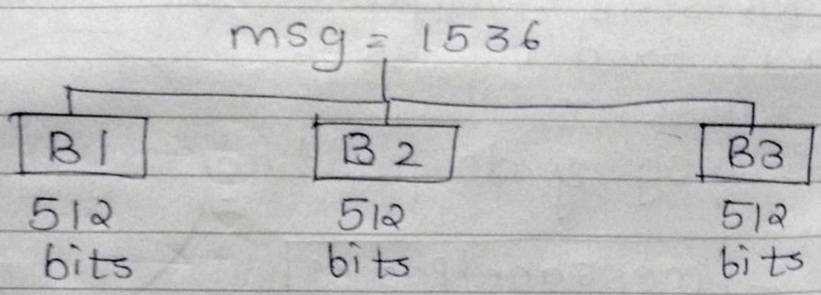
Padding bits = (Selected - 64) - msg
= (1536 - 64) - 1000
= 1472 - 1000
= 472

II] Append random string in msg appended with Padding, random string = 64 bits.

NOTE :- Why? To get Randomness

msg	padding	string
1000	472	64

III] Divide above msg block into sub-blocks each of 512 bits



IV] Initialize chaining var i.e. Buffers. There a 4 buffers each of 32 bits

Total buffer length = 4×32 = 128bits

A = a
B = b
C = c
D = d
} Var or Buffers

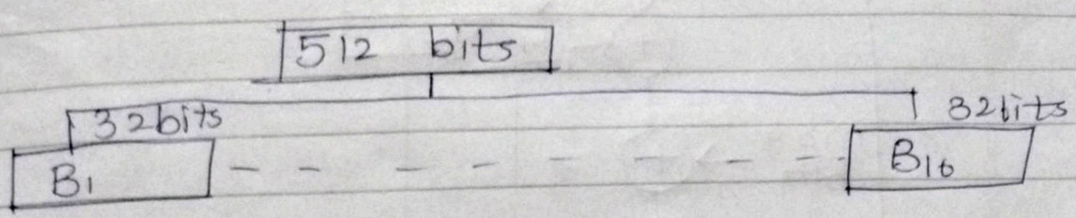
V] Round Explanation:

Each Block of 512 bits will go for 4 rounds & every round has total 16 operations.

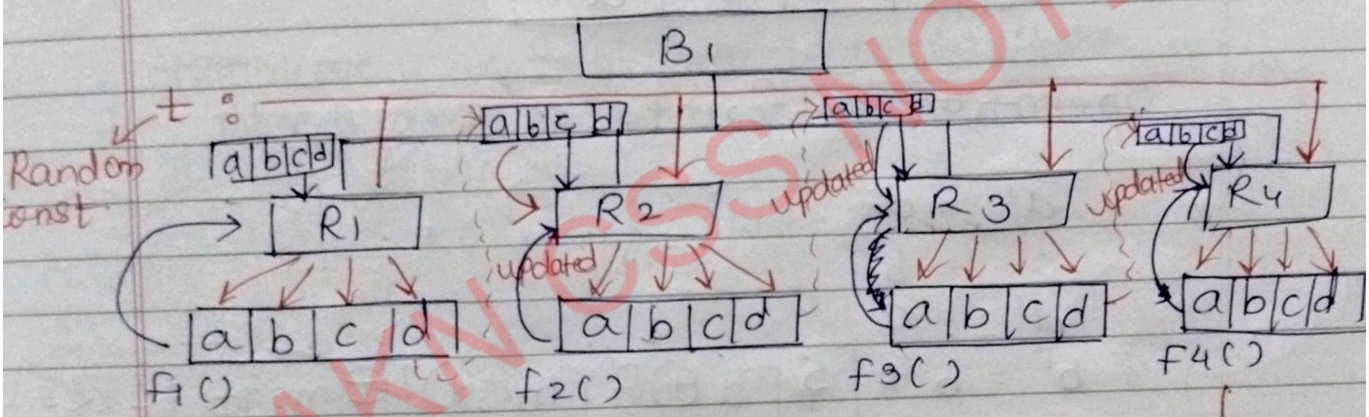
∴ Total operations performed in single Block = 16 × 4 = 64

NOTE: All rounds will have diff operation

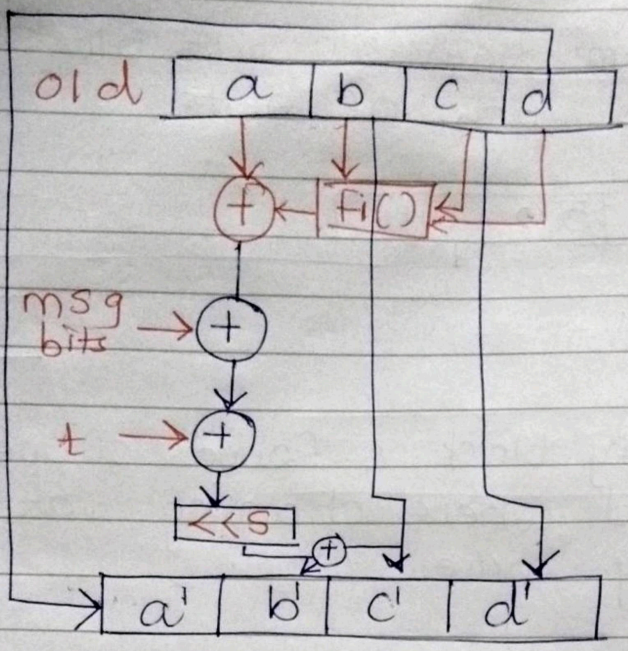
VI] Block Diagram: NOW 512 bits Block is divided into 16 blocks each of 32bits.



Now on every block 4 Rounds of operation will be performed where chaining var will be updated by some func.



512 bits.
↓
16 blocks
↓
Each block 4 rounds
↓
16 operation in every round.



Operation performed on every Round.

$$a' = d$$
$$c' = b$$
$$d' = c$$
$$b' = b + \left[\{[a \oplus fi(b,c,d)] \oplus msg \oplus t\} <<< s\right]$$

<< : Left shift by random bits 's'