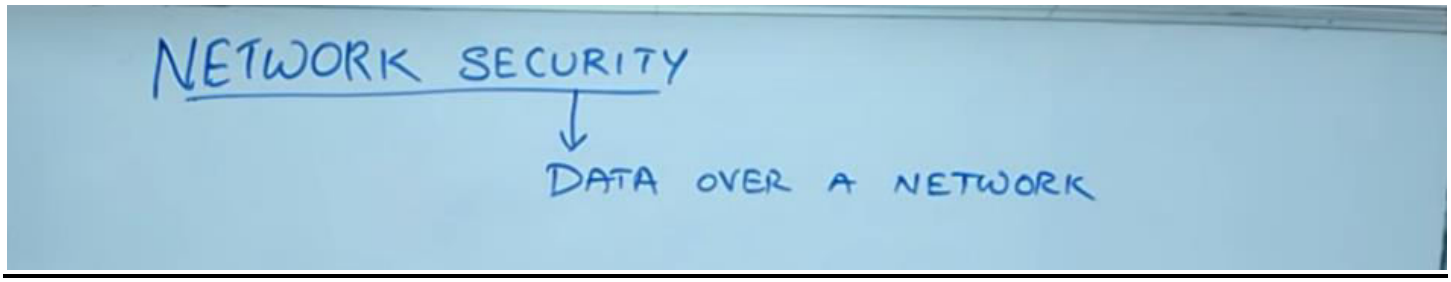
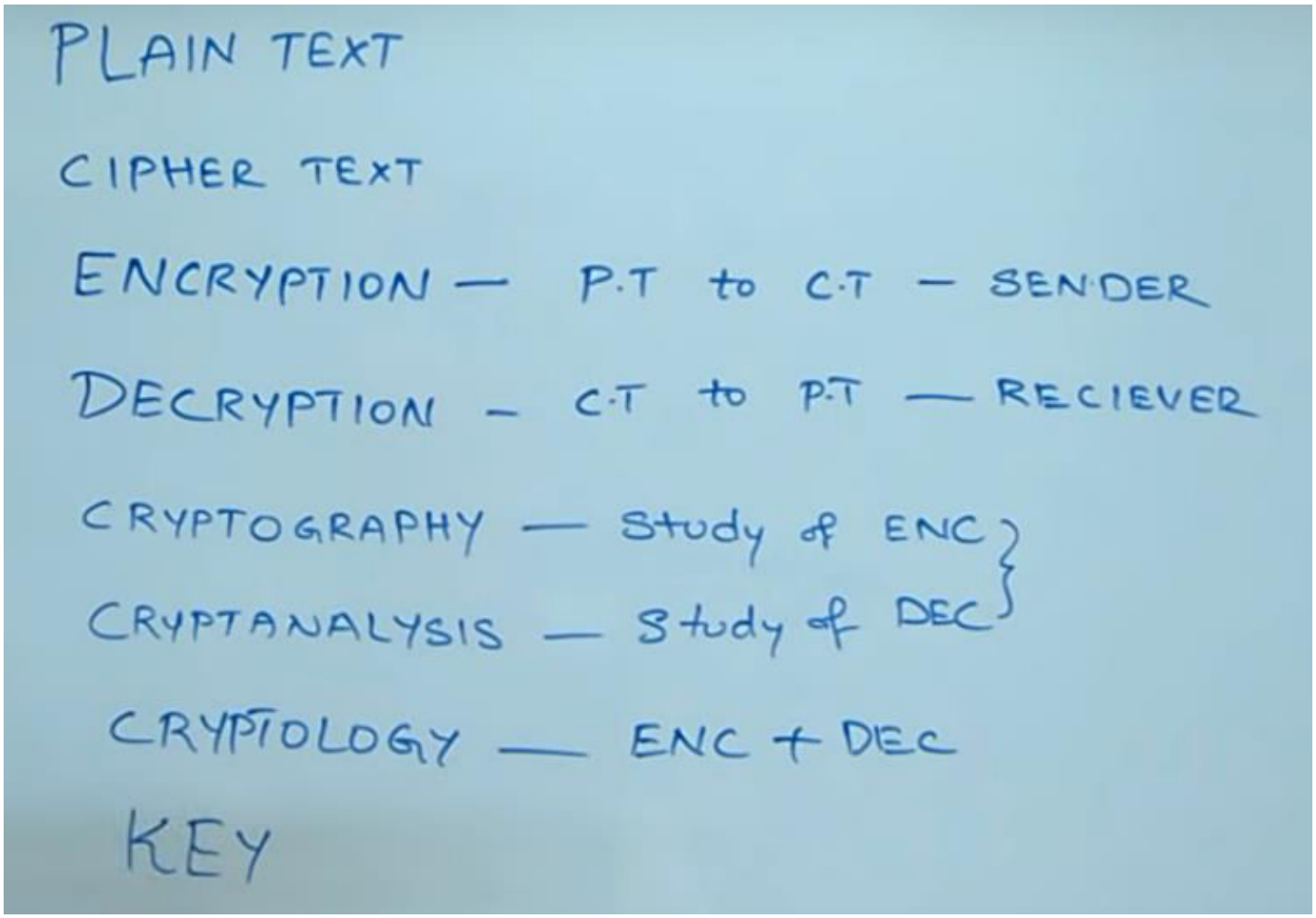
https://www.youtube.com/playlist?list=PLFD_vgVULUbDiMsLvHWxMjubOsEuvCvNn

http://bit.ly/cyptovid

Cryptography and network security

#1 NETWORK SECURITY - BASIC CONCEPTS





STREAM CIPHER — Bit by Bit

BLOCK CIPHER — Block by Block

↓

Group of bits

ENCRYPTION

/

SYMMETRIC ENC

Key — ENC, DEC

SECRET KEY

K_S

Ex1. User-A

KU_A

KR_A

ENC – KU_A

DEC – KR_A

ENC – KR_A

DEC – KU_A

ASYMMETRIC ENC

→ Two independent Keys

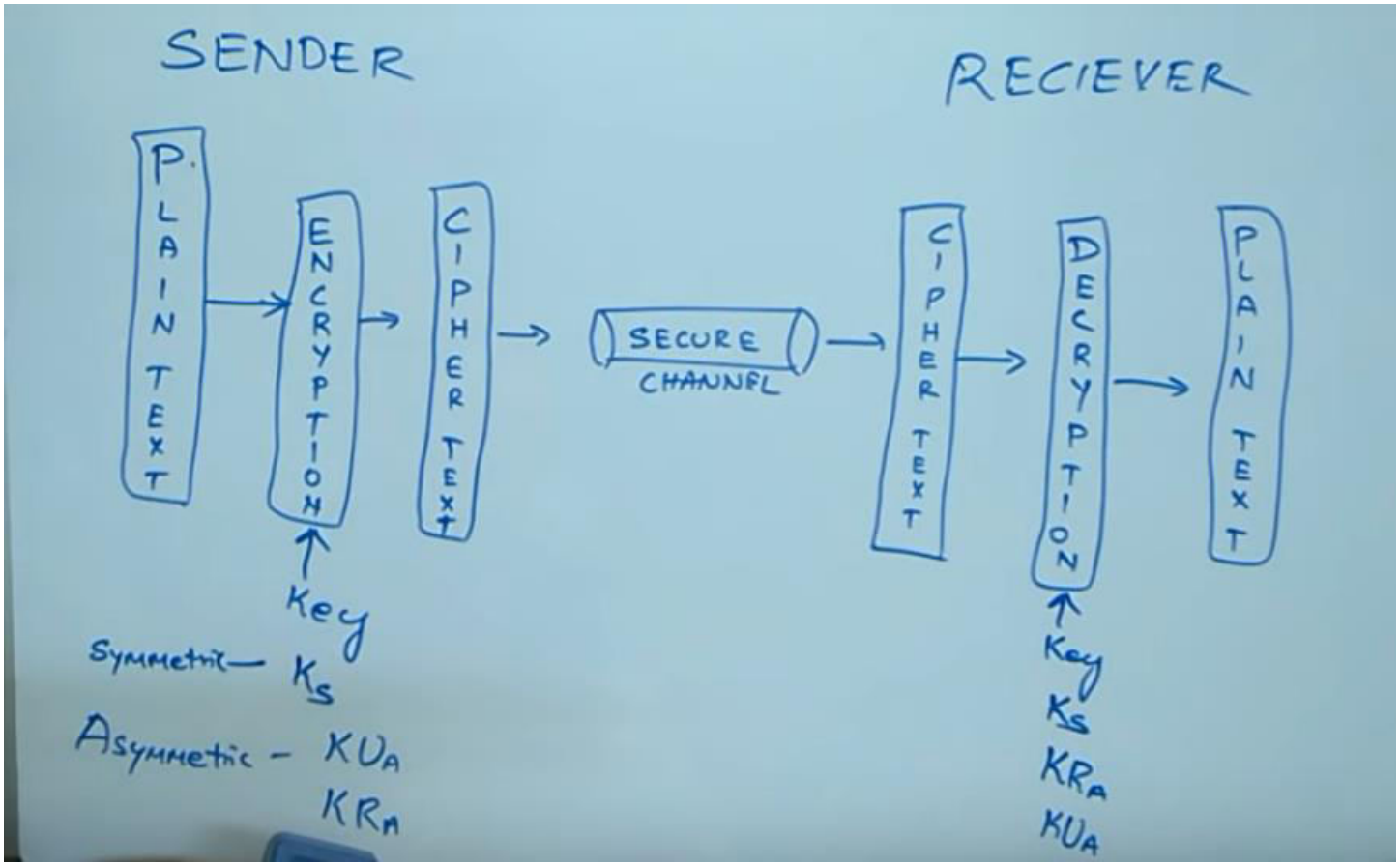
→ Pair of Keys

* Public Key (KU)
* Private Key (KR)

→ Every User – pair of Keys

→ One Key – Encryption

Second Key – Decryption



Plain Text

Cipher Text

Encryption

Decryption

Cryptography

Cryptanalysis

Cryptology

Block Cipher

Stream Cipher

Key

Secret Key

Public Key } Pair

Private Key

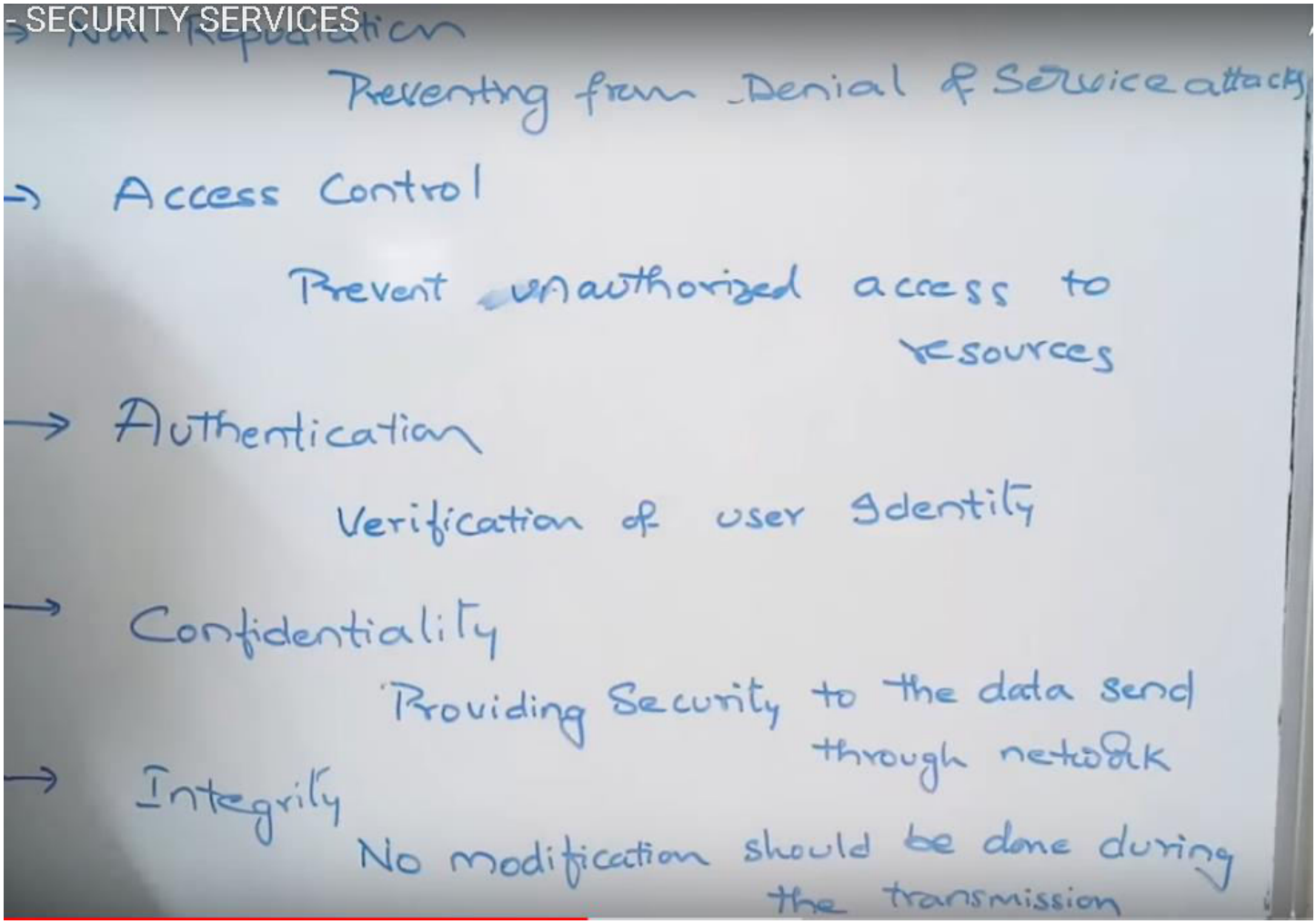
Symmetric Cipher

Asymmetric Cipher.

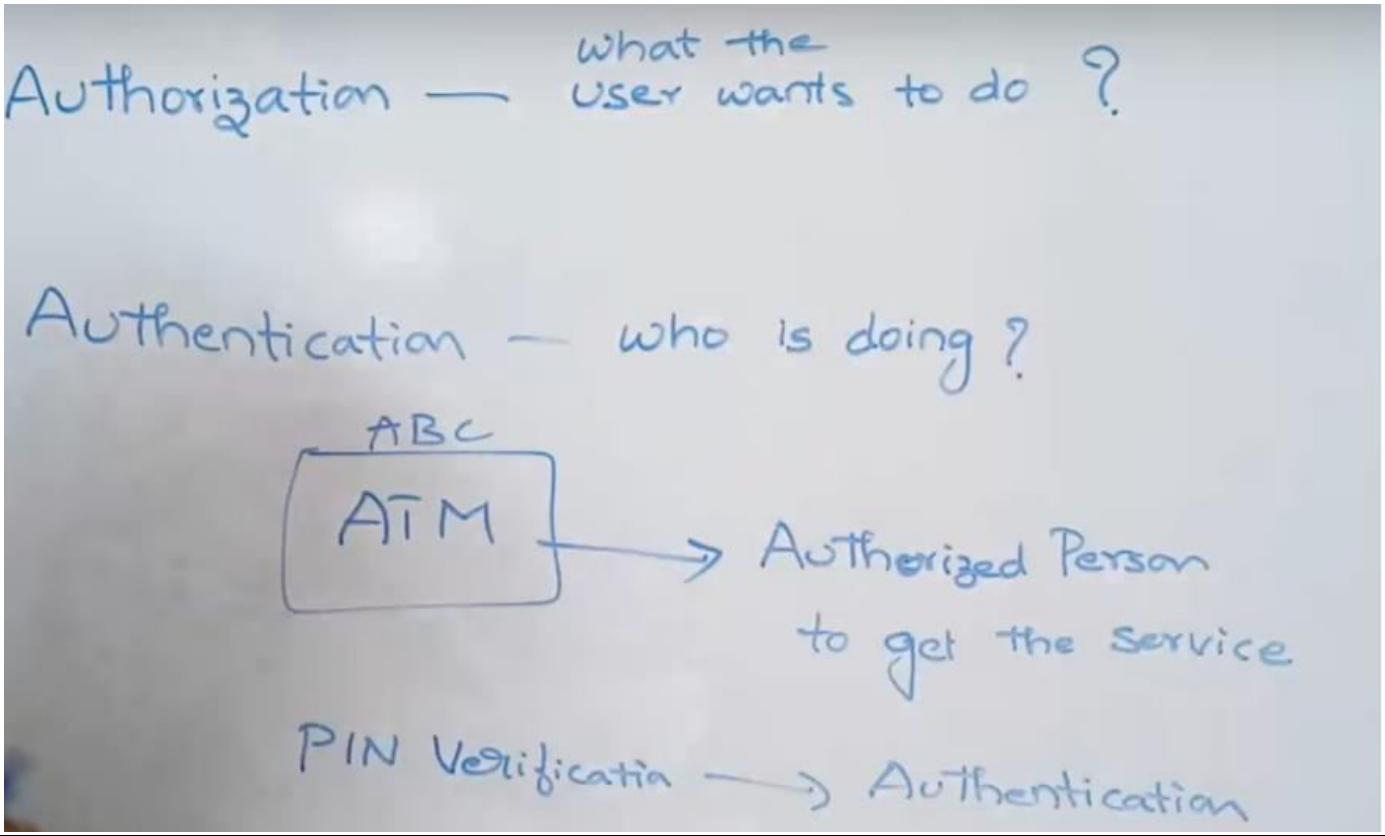
#2 NETWORK SECURITY - SECURITY SERVICES

Attack : gaining access of data by unauthorized person

I. Active attack : modification of mssg (4 types)
II. Passive : no modification only assess (release of mssg content, traffic analysis)

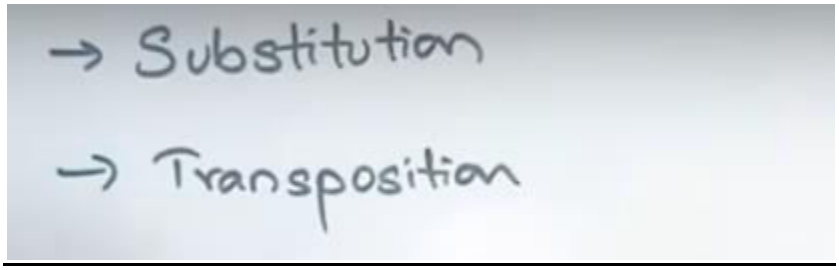


Authorization : access permission. (yes/no)
Authentication : verify user identity. (double check whether user is whom he pretends to be)

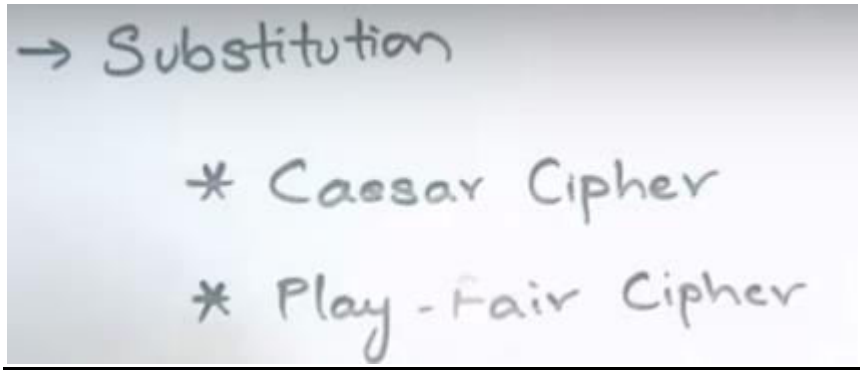


#3 NETWORK SECURITY - SUBSTITUTION TECHNIQUES

Simple symmetric encryption techniques :

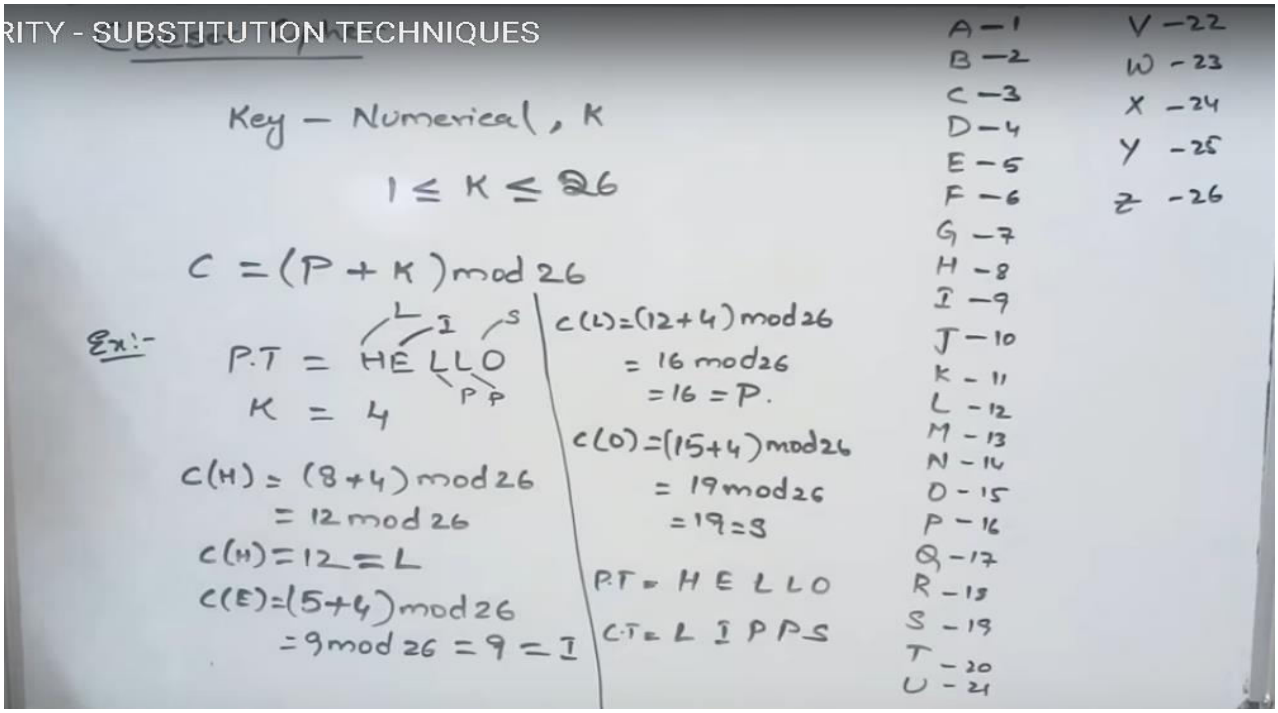


Substitiution : Complete plain text is replaced by cipher. (both in characters)



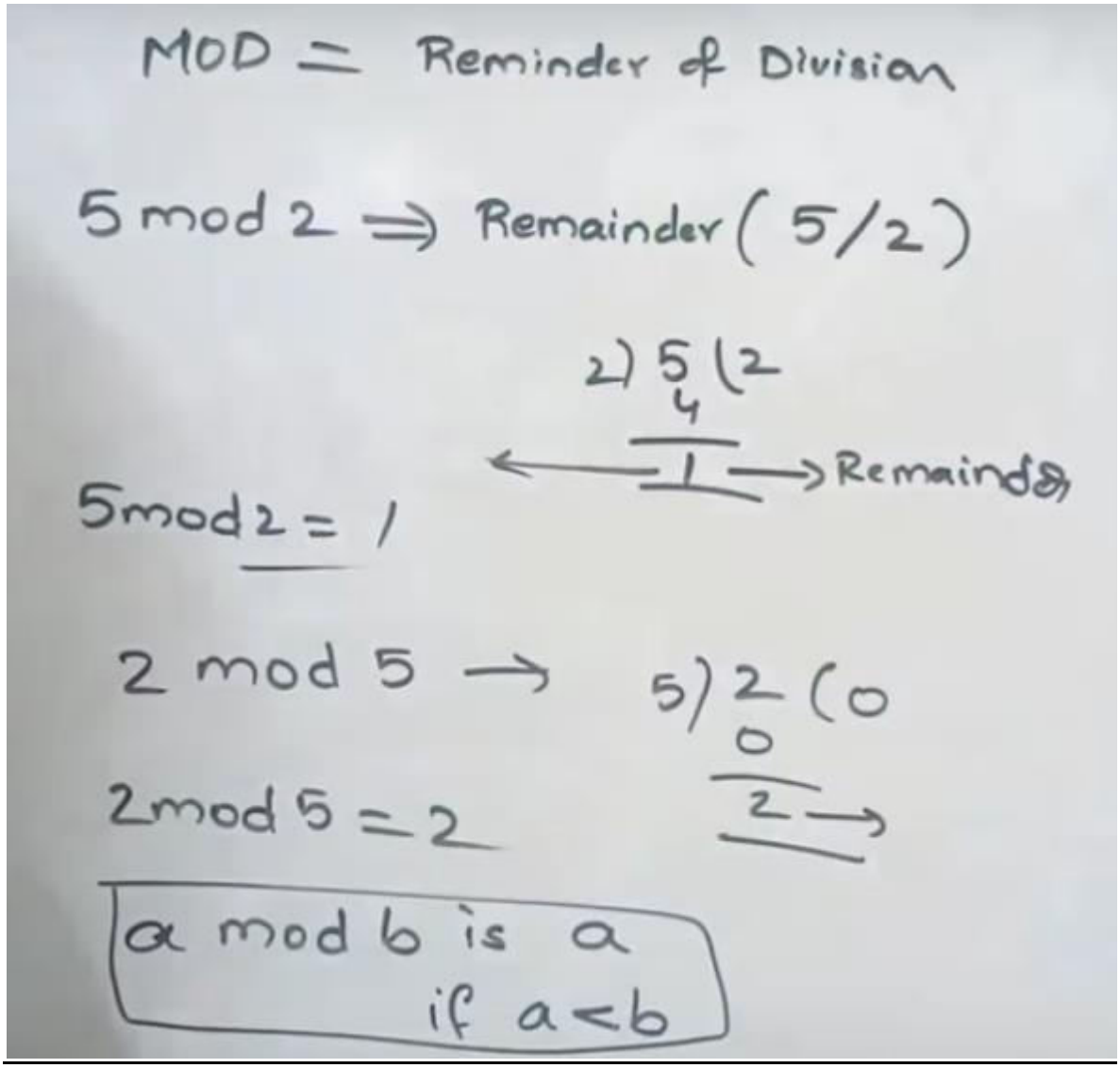
Drawback : Both symmetric techniques are used only for short text and are vulnerable to attacks.

Ceaser cipher



1. Number alphabets
2. Key is number bw 1 to 26, same used in enc & dec
3. C = (p+k) mod 26
4. Get corres char

Mod keeps new char in a to z.



PLAY FAIR CIPHER

1. First make 5*5 table
2. Key is string (fill in table, without repetition)
3. i/j in one cell
4. make pairs from begin in p.t., separate repeated by dummy, add dummy to single last char (dummy = x)

5. replace
    a. ROW – adjacent Right (circular if needed)
    b. COLUMN – beneath
    c. Different – diagonal (same row, other column as in pt)

P.T → HELLO

Key → NETWORK

C.T → ?

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

5×5 → 25 letters

→ Divide P.T to pair of letters

→ Differentiate Repeated letters in the pair with dummy letter

→ If pair of P.T letters are in Same Row

Replace them with Right most letters

→ If the P.T letters are in same column

Replace with beneath letters

→ P.T letters are in different Row & Column (diagonal)

P.T → HELLO

Key → NETWORK

C.T → ?

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

5×5 → 25 letters

HE | LL | O

differentiate with 'x'

HE | LX | LO

HE → WF

LX → UP

LO → NS

HELXLO → WFUPNS

BSTITUTION TECHNIQUES

P.T → BALLOON

Key → NETWORK

C.T → ?

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

5×5 → 25 letters

BA | LL | OO | N

BA | LX | LO | ON

BA → CB

LX → UP

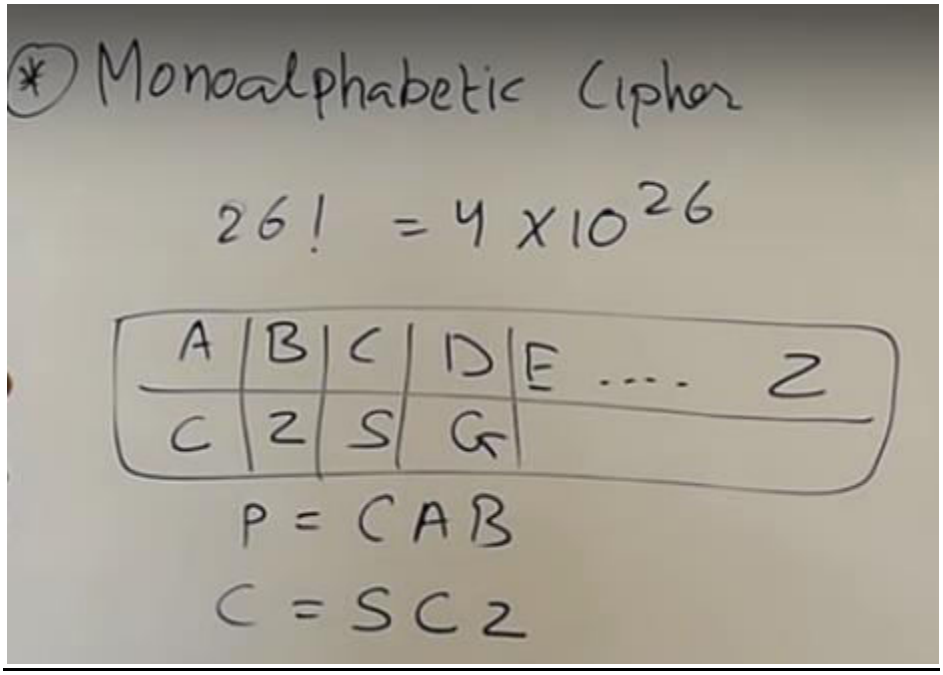
LO → NS

ON → NE

#4 Monoalphabetic Cipher

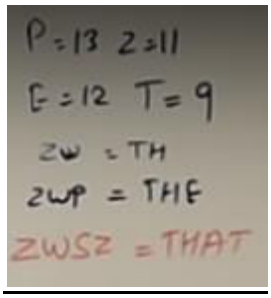
Adv : brute force not possible due to large no of keys.

Assumption : Both sender and receiver have substuton table



Drawback : Frequency of English alphabets is used.

1. Single : E, T
2. Diagram : TH
3. Triplet : THE
4. Four : THAT etc.



#5 Hill Cipher Encryption 2by2 Matrix

I. Key matrix is given (2*2, hence pairs are made)
II. Matrix multiplication is used.
III. Make pairs of p.t.
IV. **N.B. : INDEX STARTS FROM 0**
V. Repeat for other pairs.

⊛ HILL CIPHER (Encryption)

$26 \overline{)480}$

1

$26 \overline{)33}$

26

07

$P = \underline{HE}\underline{LP}$ $\quad K = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}$

$C = K \cdot P \bmod 26$ $\quad 2\times2$

$P = HE = \begin{bmatrix} 7 \\ 4 \end{bmatrix}$ $\quad LP = ?$

$= \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 7 \\ 4 \end{bmatrix} \bmod 26$

$= \begin{bmatrix} 3*7 + 3*4 \\ 2*7 + 5*4 \end{bmatrix} \bmod 26$

$= \begin{bmatrix} 33 \\ 34 \end{bmatrix} \bmod 26 = \begin{bmatrix} 7 \\ 8 \end{bmatrix} \bmod 26$

$= \begin{bmatrix} H \\ I \end{bmatrix} = HI$

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

#6 Hill Cipher Decryption 2by2 Matrix

⊛ HILL CIPHER (Decryption)

$P = HELP$ $\quad K = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}$

$C = HI$ $\quad C = K \cdot P \bmod 26$

$P = K^{-1} \cdot C \bmod 26$

$K = \frac{1}{|K|} \operatorname{adj} K \quad |K| = 9$

$K = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \operatorname{adj} K = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

$\operatorname{adj} K = \begin{bmatrix} 5 & -3 \\ -2 & 3 \end{bmatrix}$

$= \frac{1}{9} \begin{bmatrix} 5 & -3 \\ -2 & 3 \end{bmatrix} = 3 \begin{bmatrix} 5 & -3 \\ -2 & 3 \end{bmatrix}$

$= \begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix} \quad = 3 \begin{bmatrix} 5 & 23 \\ 24 & 3 \end{bmatrix} = \begin{bmatrix} 15 & 69 \\ 72 & 9 \end{bmatrix} \bmod 26$

A	B	C	D
0	1	2	3
I	J	K	L
8	9	10	11
Q	R	S	T
16	17	18	19
Y	Z		
24	25		

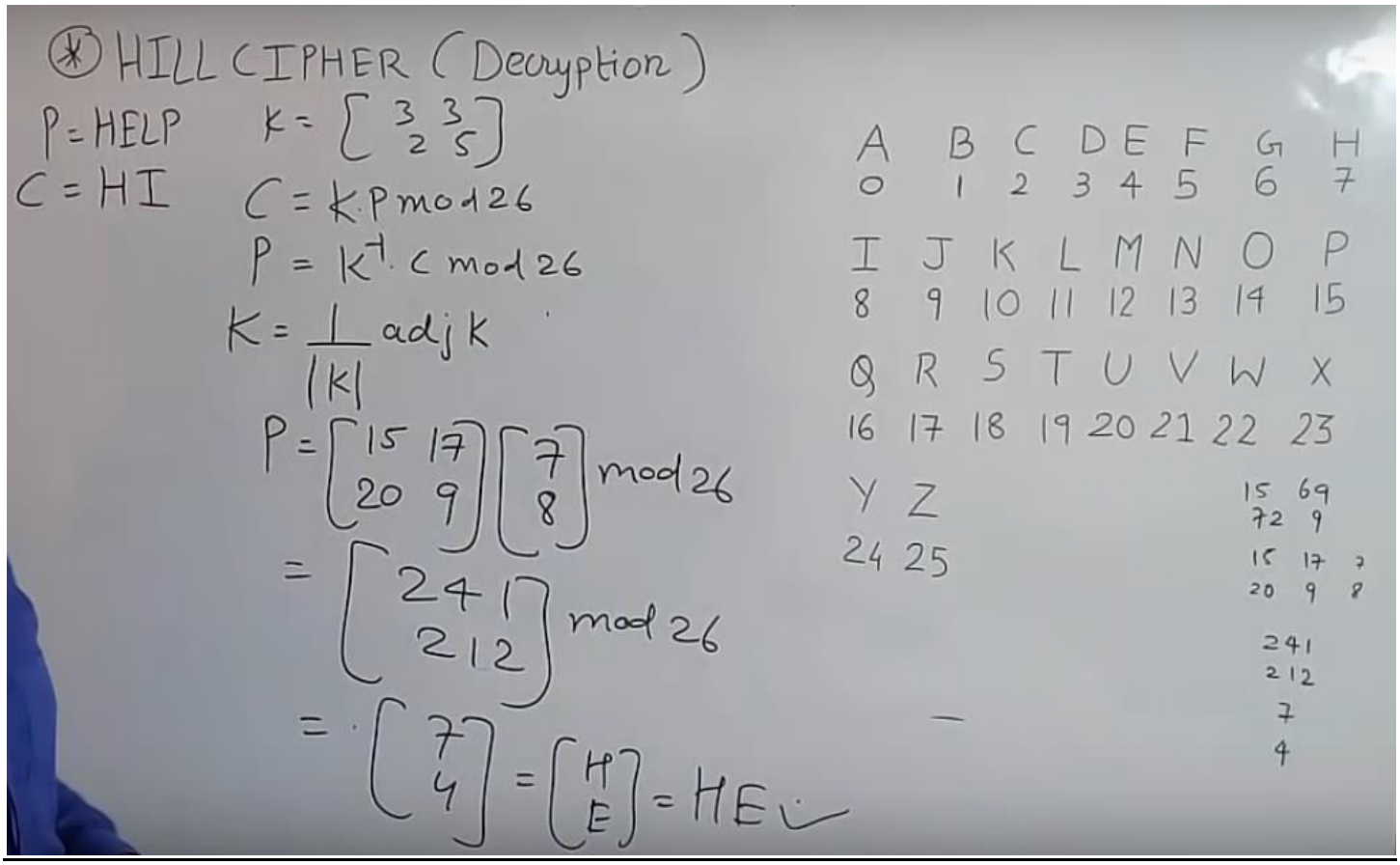
I. We take pair in c.t. if key matrix is 2*2. (triplet , if 3*3)

II. Find inverse of K mat.

   a. Determinant
   b. Adjoint
   c. Inverse mod of determinant
   d. Change negatives in adjoint
   e. Multiply
   f. Finally find modulus of resultant matrix

III. P = (k^-1).(C)MOD 26

   a. Multiply
   b. Find mod
   c. Find corresponding p.t.
   d. Repeat for other pairs



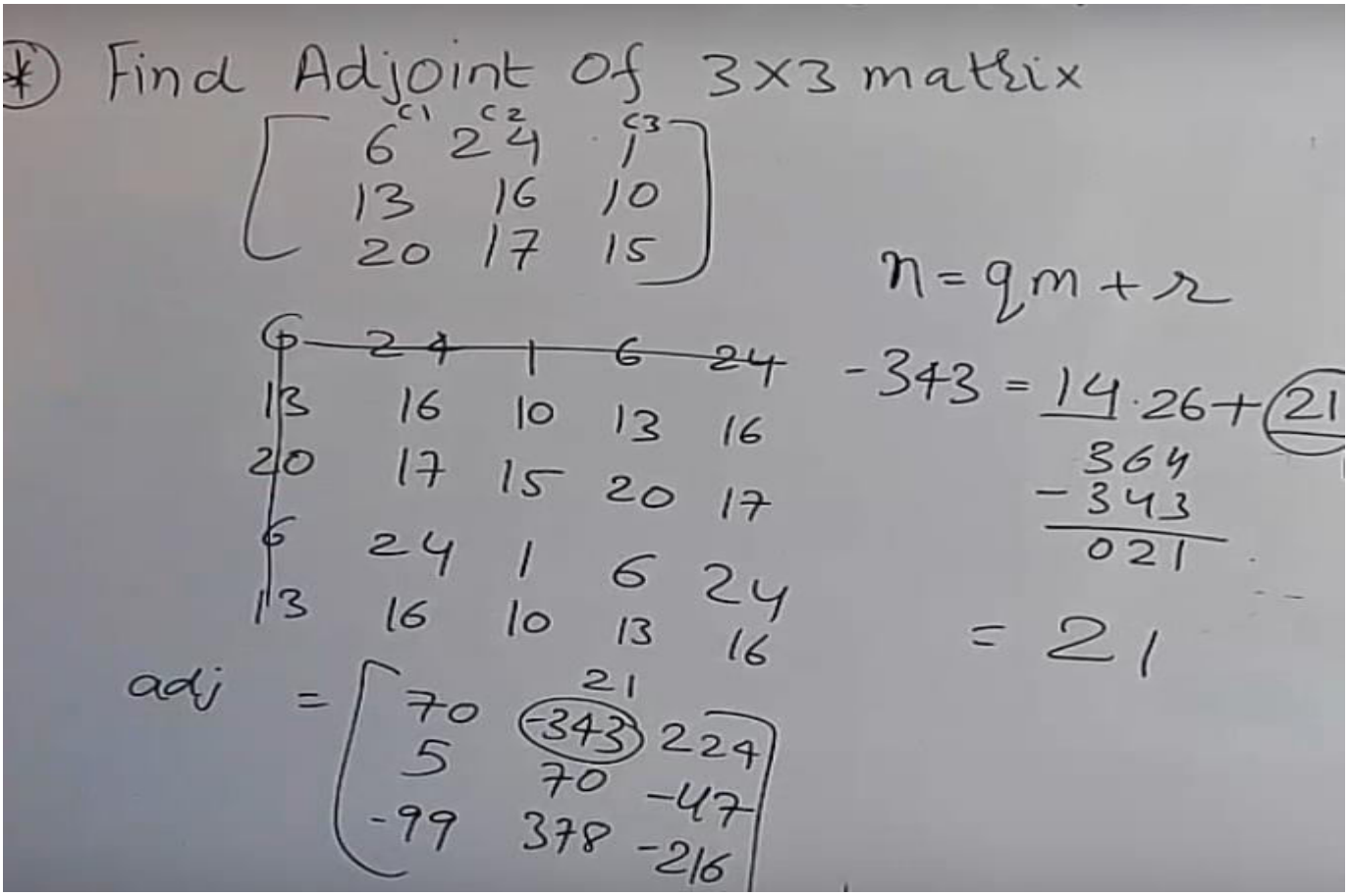
- Shortcut to find adjoint of 3*3 matrix

I. Repeat 1,2 row and col

II. Cancel 1st row and col

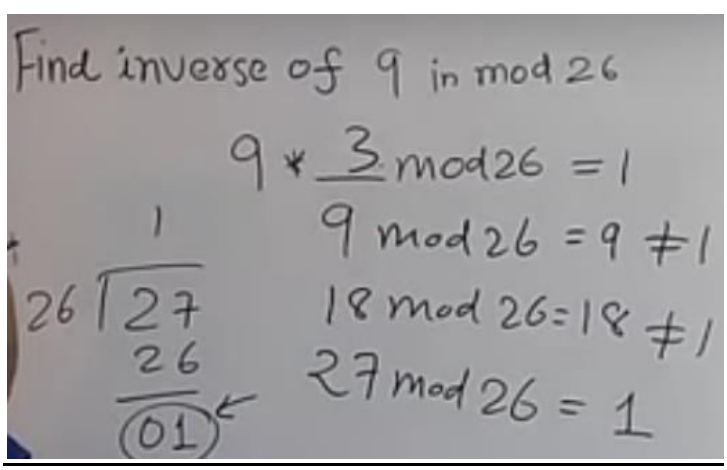
III. <u>Find convolution and write in transpose order.</u>

IV. <u>For mod operation convert negatives to positive by adding a multiple of 26 just larger than negative no in magnitude.</u>



- <u>find inverse in modular arithmetic-how to find inverse modulo m</u>

<u>Inverse modulo always exists bw 1 to 25</u>



#7 NETWORK SECURITY - TRANSPOSITION TECHNIQUES

Transposition Techniques

* Rearrange the order of Plain Text bits
* No replacement/substitution

* Rail Fence Cipher.

Ex:- "Welcome to my session" – Plain Text

w l o e o y e s o
e c m t m s s i n

C.T → wloeoyesoecmtmssin

Vulnerable and only for short mssgs

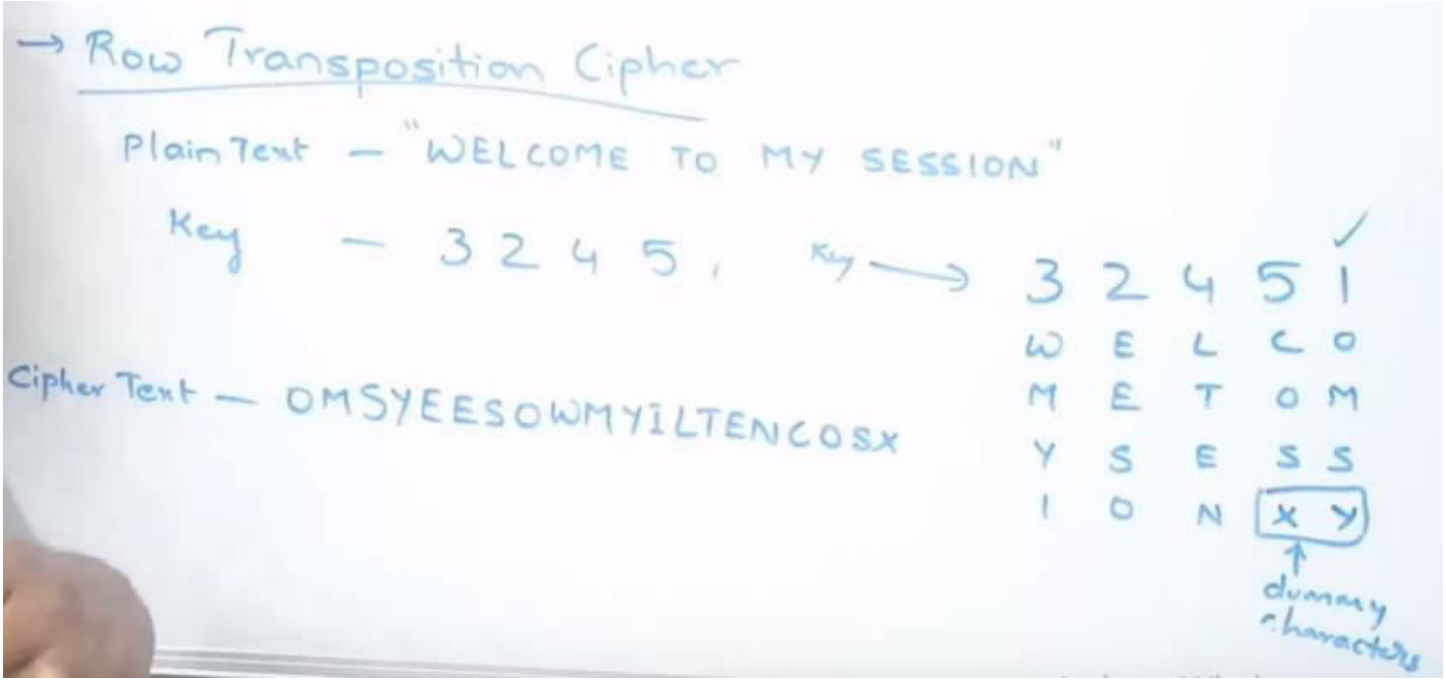
→ Row Transposition Cipher

Plain Text – "WELCOME TO MY SESSION"

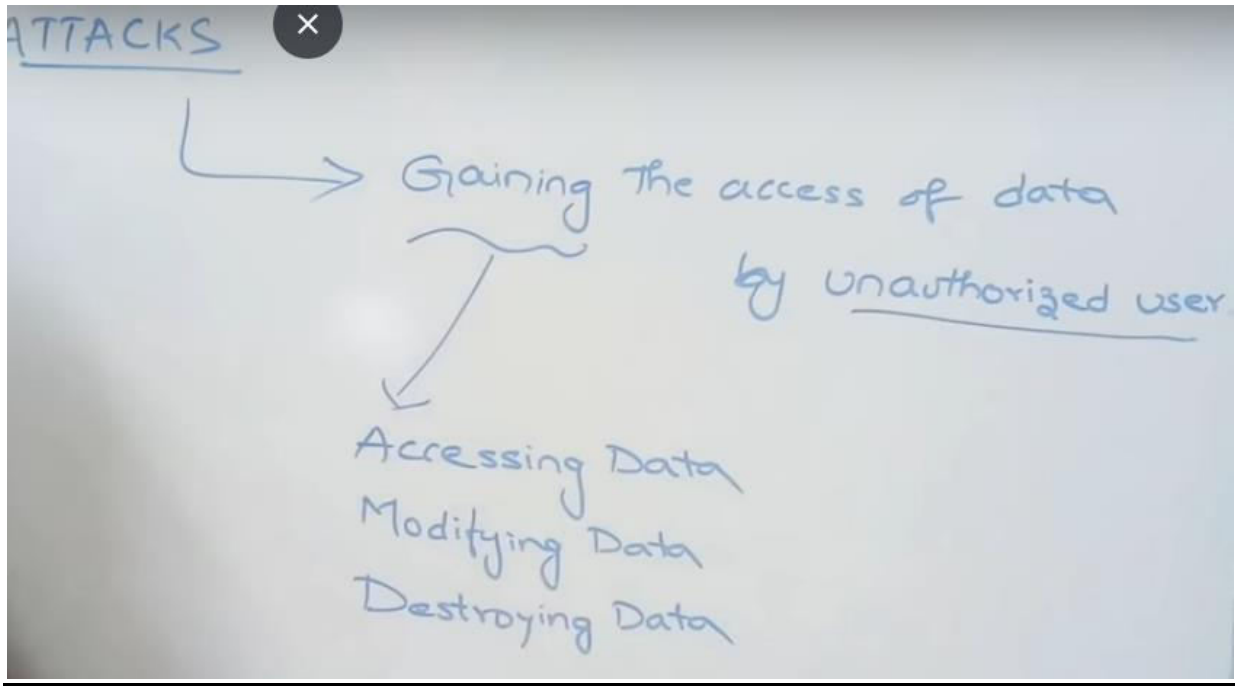
Key – Unique numbers should be considered 0 to 9 (3 2 4 5 1)

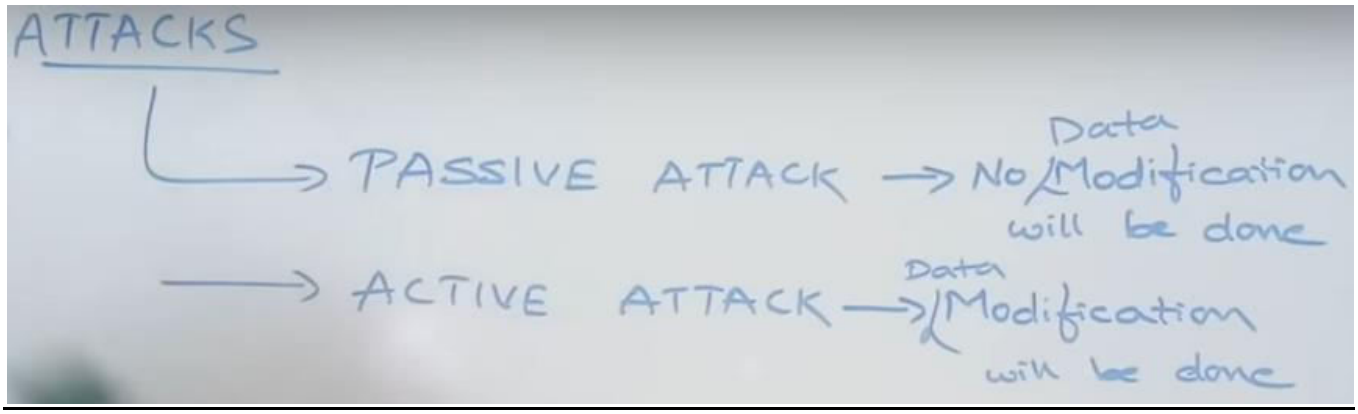
Fill all columns successfully with dummy chars.

If key length = n , make n*n table



#8 NETWORK SECURITY - TYPES OF ATTACKS

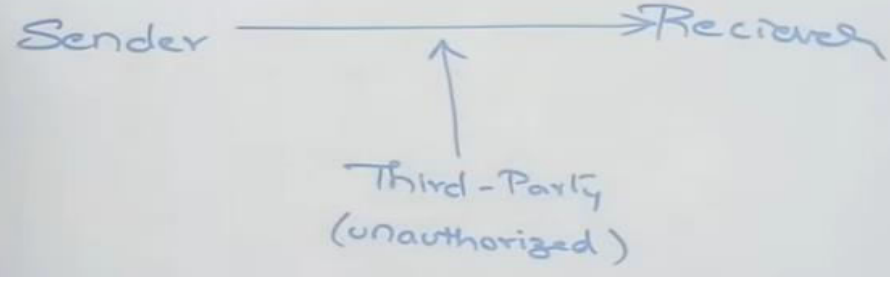




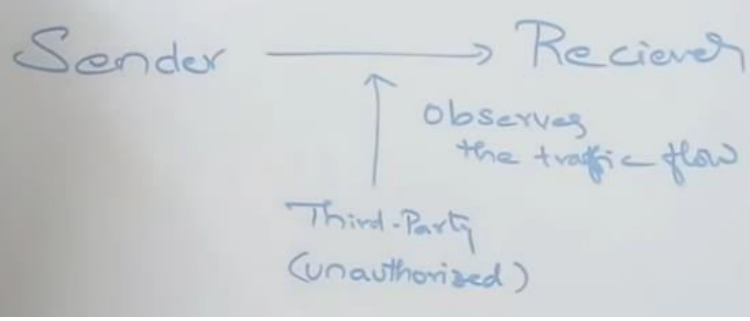
Passive Attack

Eavesdropping

→ Release & the content

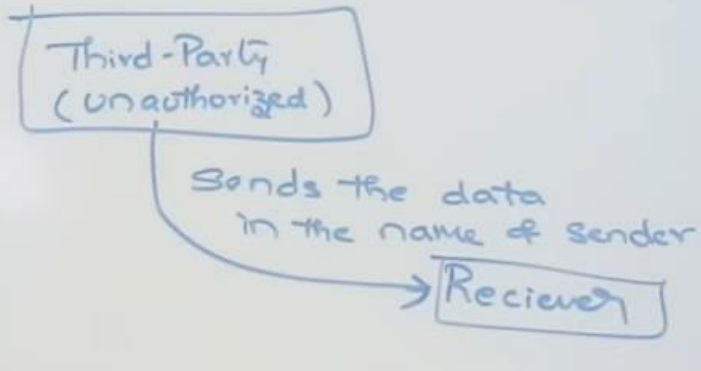


→ Traffic Analysis

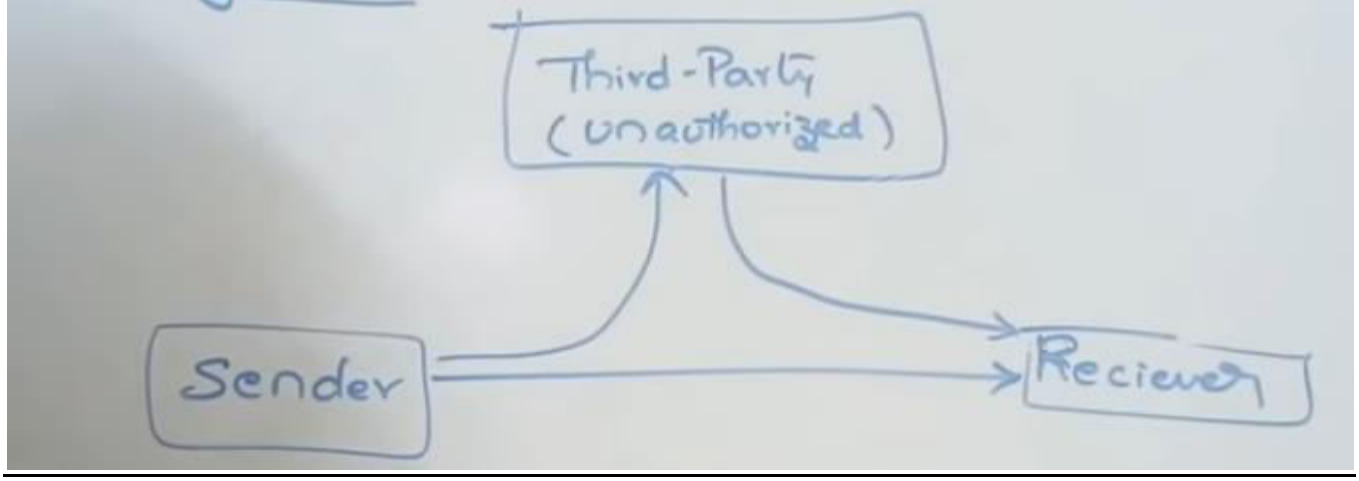


Active Attacks

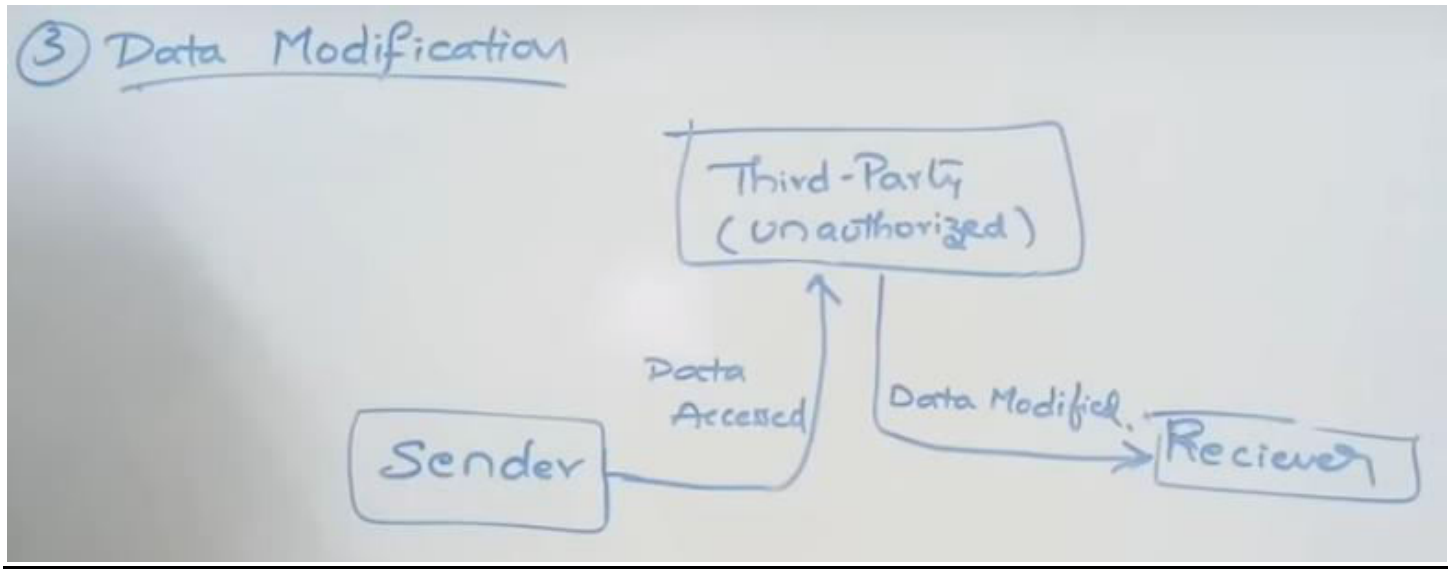
① Masquerade



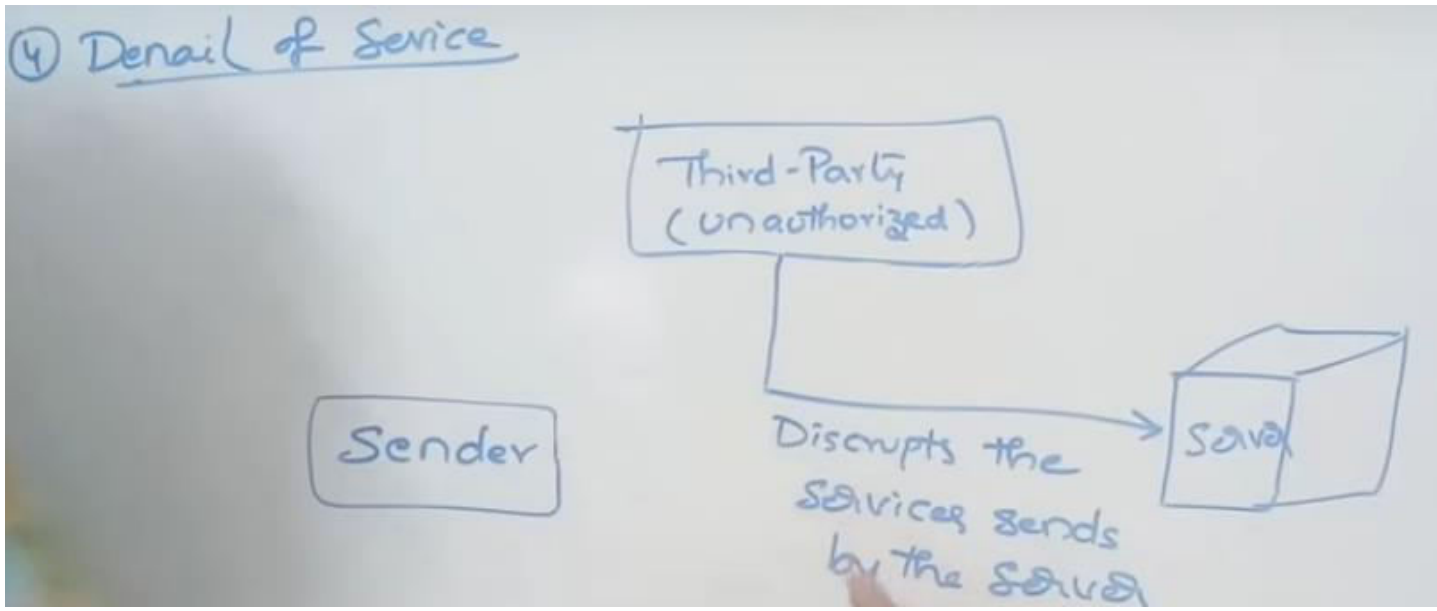
Replay Attack



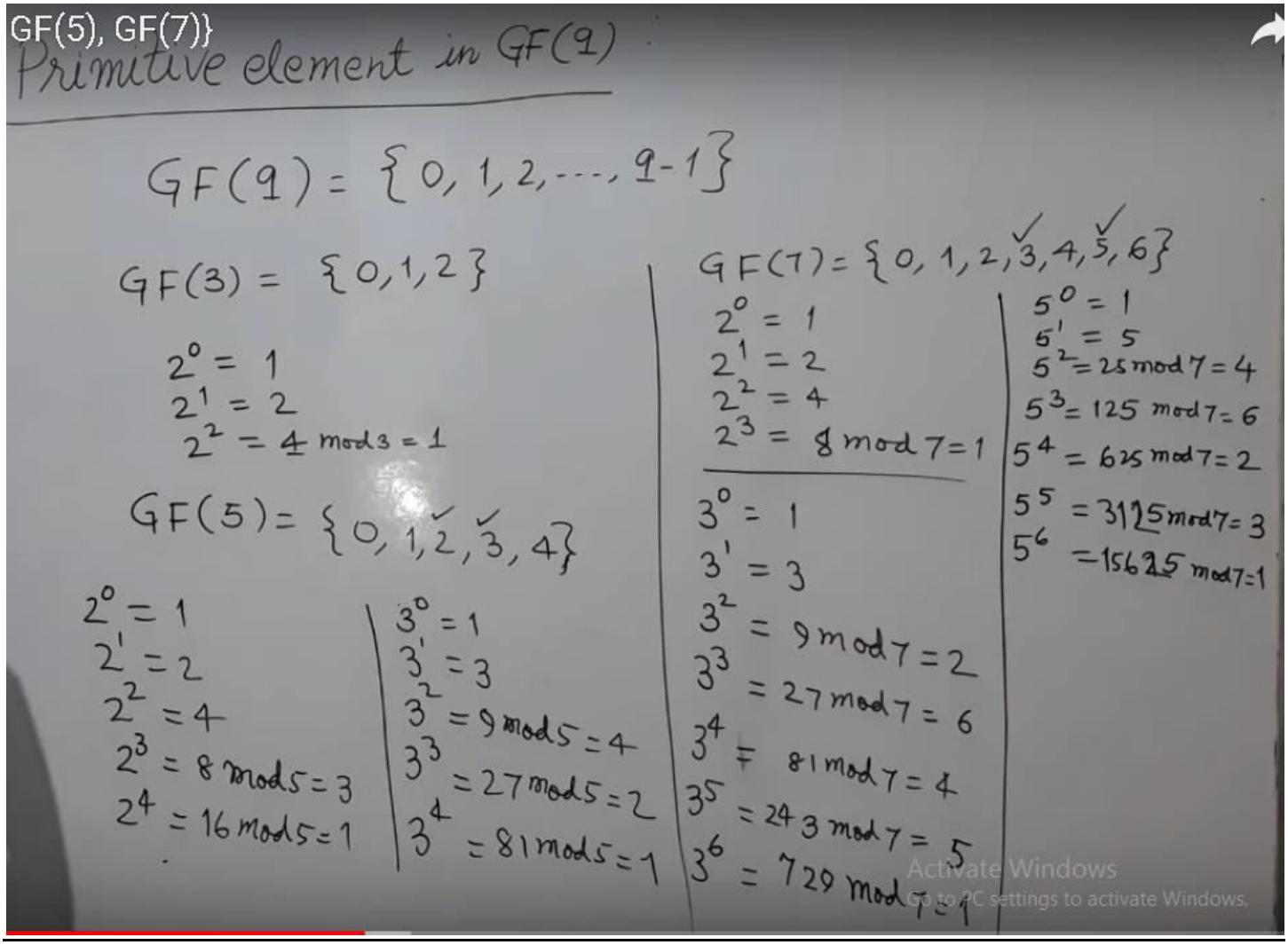
③ Data Modification



④ Denail of Service



#9 Galois Field {GF(2), GF(3), GF(5), GF(7)}



Primitive root/element = no in set which raised to diff powers give all other elements except 0.

Only check prime no.

Take mod n after power.

Prime no not primitive element will form cycle.

Multiplicative and additive table = take mod n

For additive inverse = see where all 0 lie

For multiplicative inverse = see where all 1 lie

GF(5)

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4		
3					
4					

GF(2)

+	0	1
0	0	1
1	1	0

*	0	1
0	0	0
1	0	1

GF(3)

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

*	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

#10 Extension of GF(2) to GF(4)

4 is a prime no, so cant find additive, multi inv directly . we need primitive polynomial (generator poly) [a prime poly of degree 2 (2^2 = 4)]

Prime poly = which can not be factorized.(check factors by putting values and if they give 0)

Check all the elements of gf(base) , here gf(2) must not be a factor. (also take mod base after getting result)

[Remainder of any poly divided by another poly of degree 2 will always be less than degree 2]

[for additive table, anything having 2 is made 0 due to mod 2]

[for multiplicative table, keep req on lhs, shift rest on rhs and make all signs +]

$GF(4) = \{0, 1, 2, 3\}$

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

$x^2 + x + 1 = 0$

$x^2 = +x + 1$

$x^2 + x = -1$

$x^2 + x + x + 1$

$x^2 + 1$

$p(x) = x^2$

$p(0) = 0$

$p(x) = x^2 + 1$

$p(0) = 0 + 1 = 1$

$p(1) = 1 + 1 = 0$

$\boxed{p(x) = x^2 + x + 1}$

$p(0) = 1$

$p(1) = 1 + 1 + 1 = 1$

$0, 1, x, x+1$

+	0	1	x	$x+1$
0	0	1	x	$x+1$
1	1	0	$1+x$	x
x	x	$x+1$	0	1
$x+1$	$x+1$	x	1	0

*	0	1	x	$x+1$
0	0	0	0	0
1	0	1	x	$x+1$
x	0	x	$x+1$	1
$x+1$	0	$x+1$	1	x

$x = 2$

GF(4) Addition table

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

Multiplication table for GF(4)

*	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

+	0	1	x	$x+1$
0	0	1	x	$x+1$
1	1	0	$1+x$	x
x	x	$x+1$	0	1
$x+1$	$x+1$	x	1	0

*	0	1	x	$x+1$
0	0	0	0	0
1	0	1	x	$x+1$
x	0	x	$x+1$	1
$x+1$	0	$x+1$	1	x

Activate Win
Go to PC settings

#11 NETWORK SECURITY - FEISTEL STRUCTURE

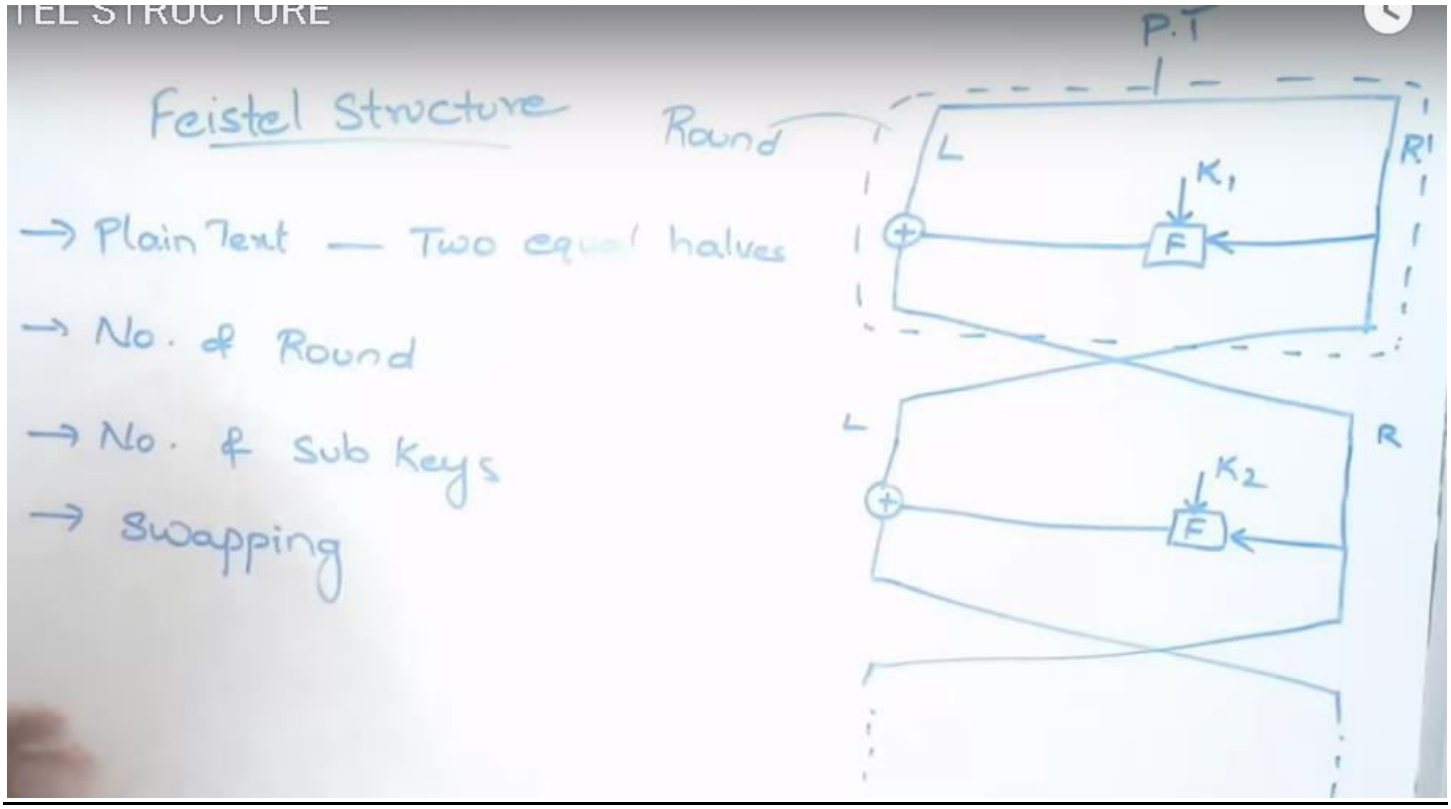
The no of rounds and function on right half depend on algo used.

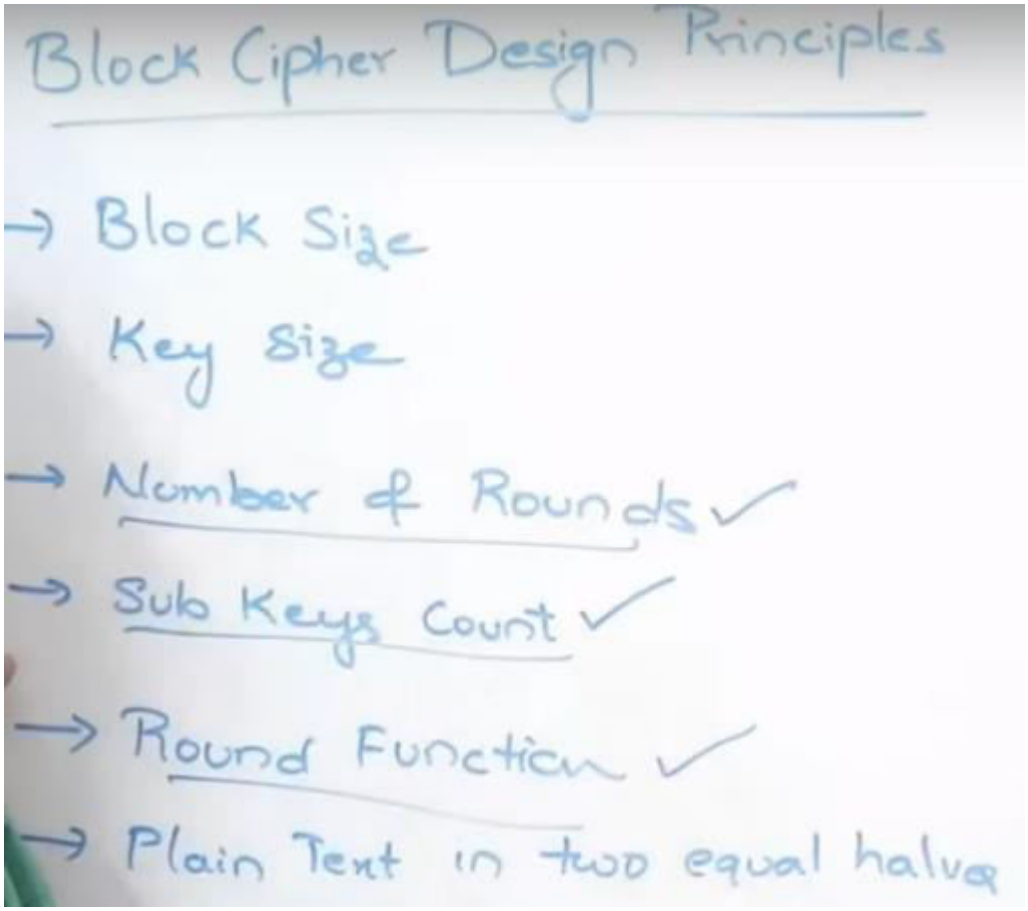
I. Divide in half
II. Apply a function on right half (each function uses sep key in each round)
III. Xor the result with left half
IV. Swap the result on left & original right for next round.

[keys are generated from master key]

Here security depends on

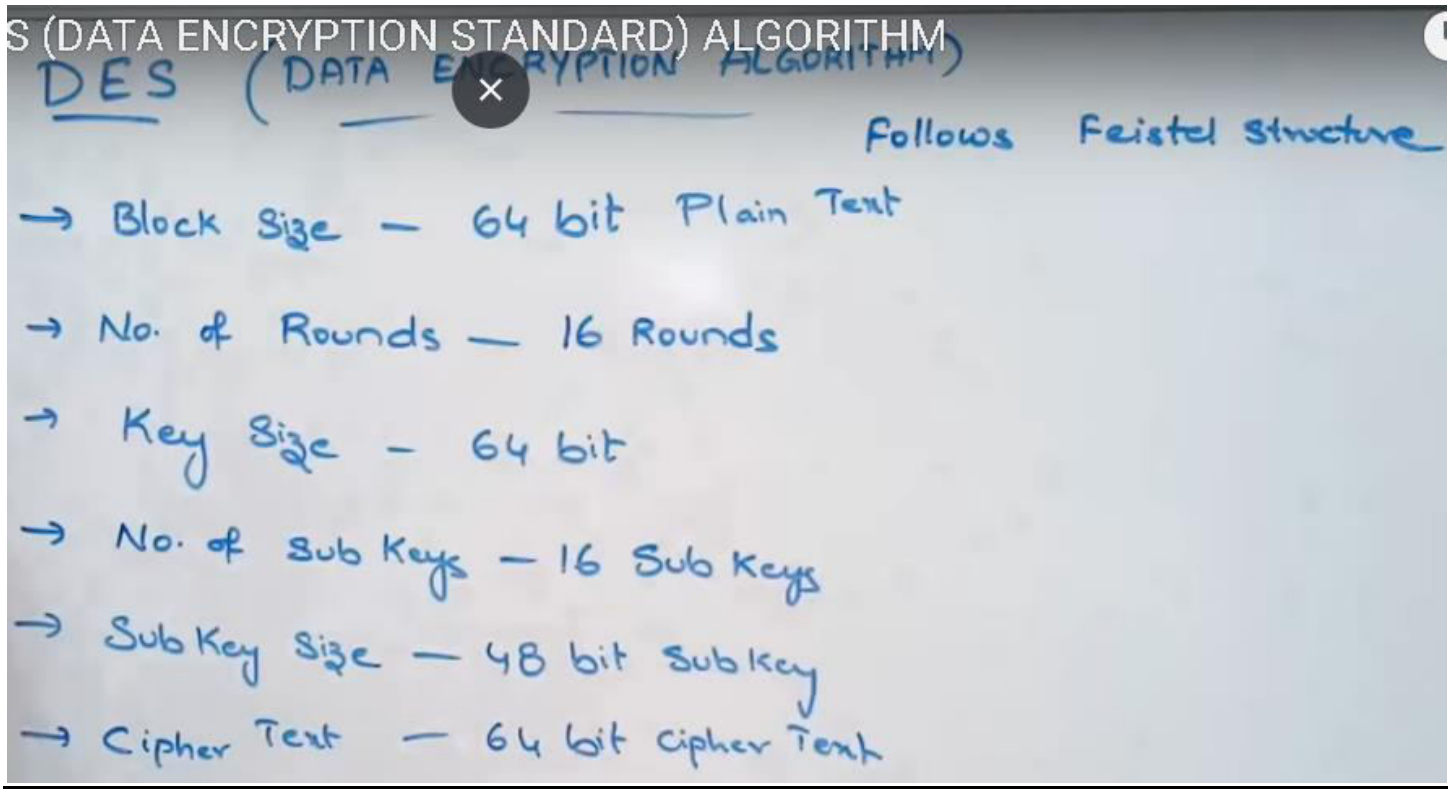
I. No of rounds
II. No of sub keys
III. Round Function complexity



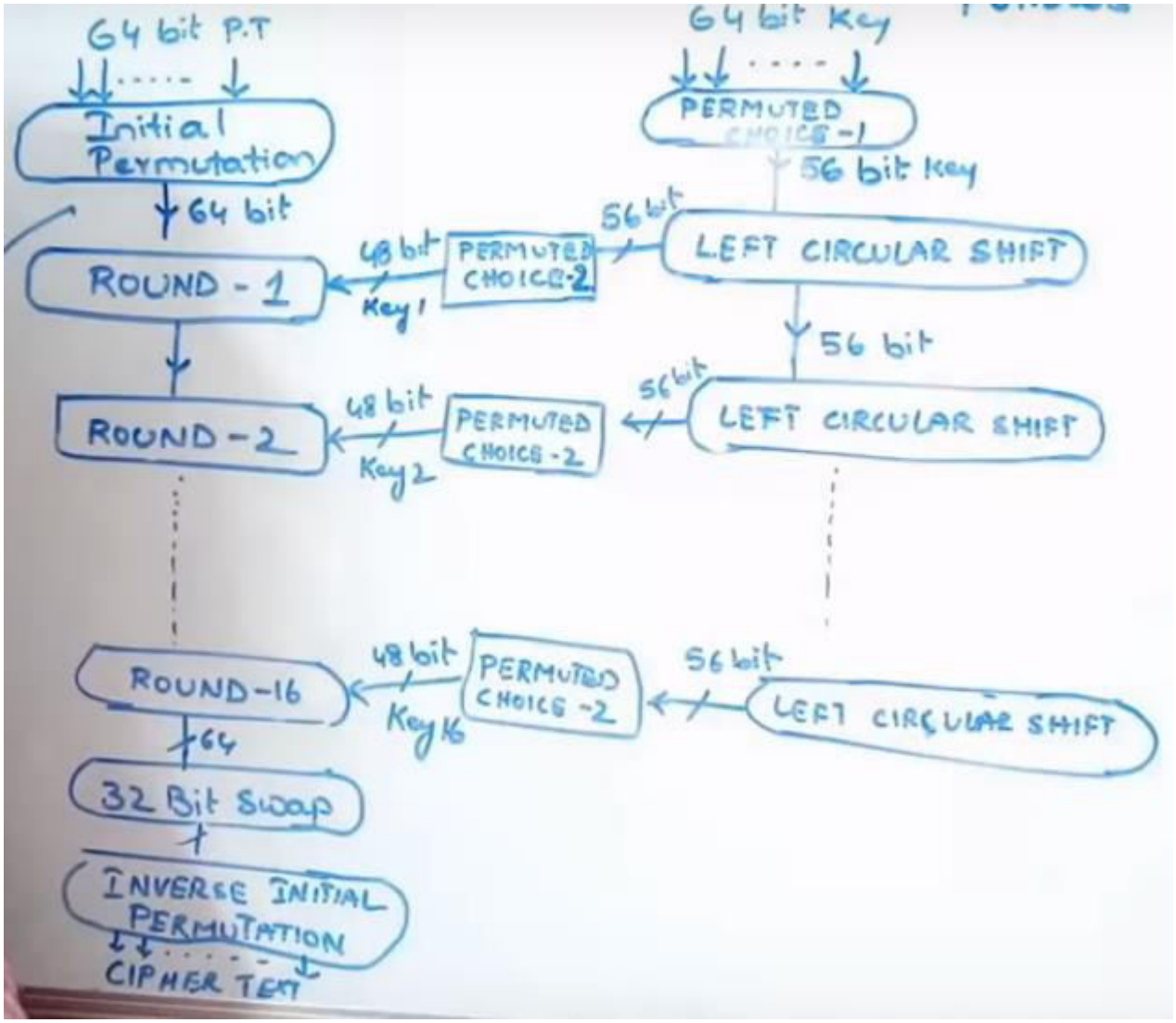


Block size = pt size

#12 NETWORK SECURITY - DES (DATA ENCRYPTION STANDARD) ALGORITHM



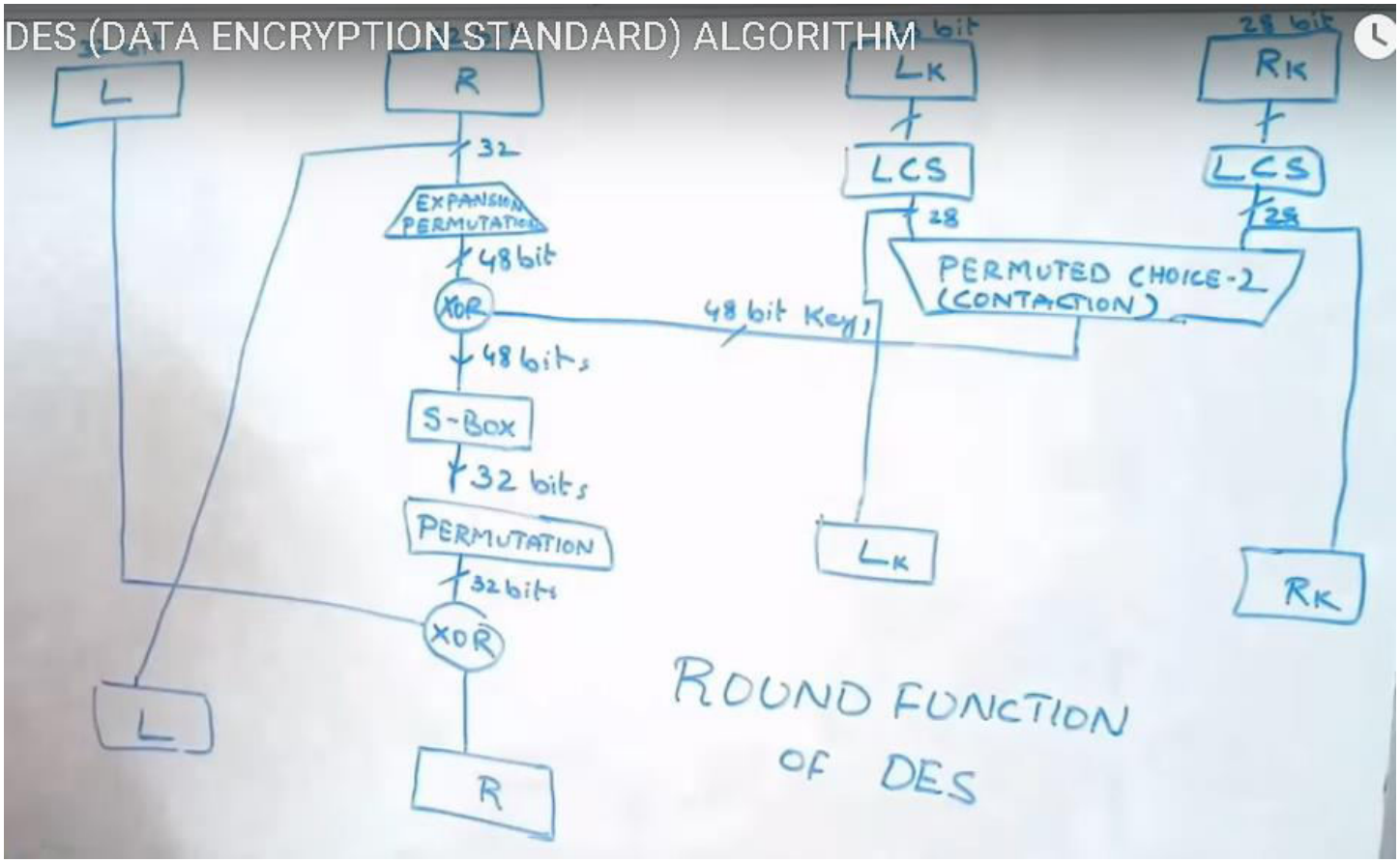
We use predefined transposition order to rearrange the bits. (also called permutation)



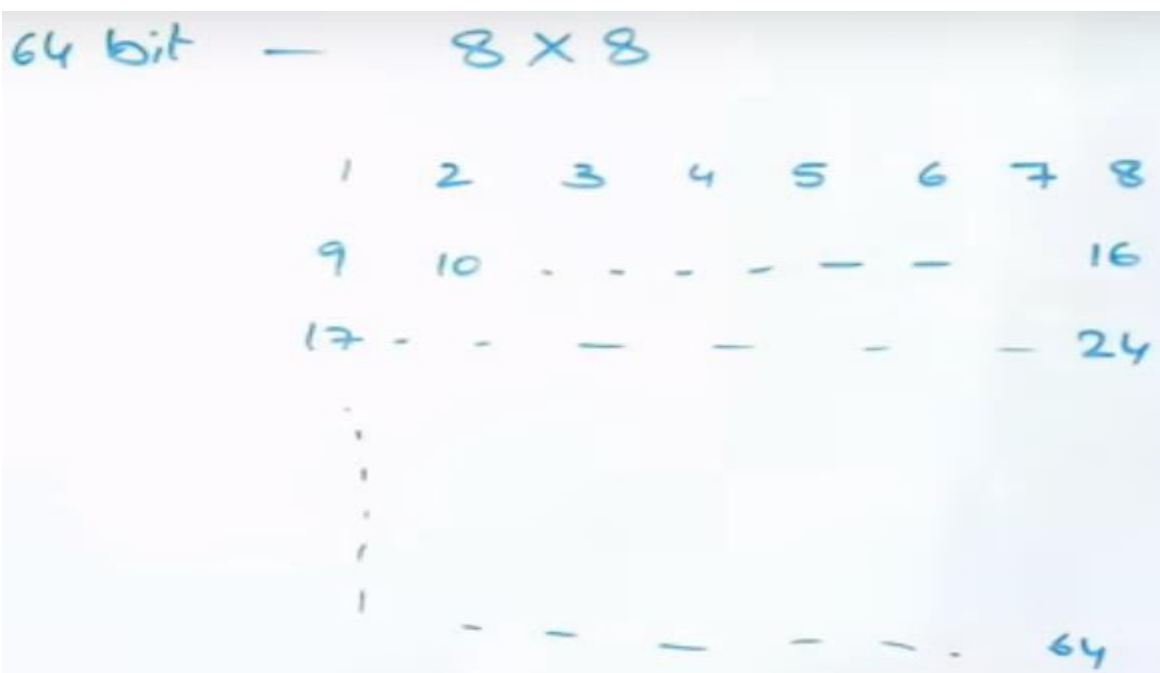
I. We begin with 64 bit pt and 64 bit key
II. Pt -> initial perm -> round 1-16 ->32 bit swap -> inverse initial perm -> ct
III. (64) key -> pc1 -> (56) -> lcs -> (56) -> pc2 -> (48) -> key n

Round function

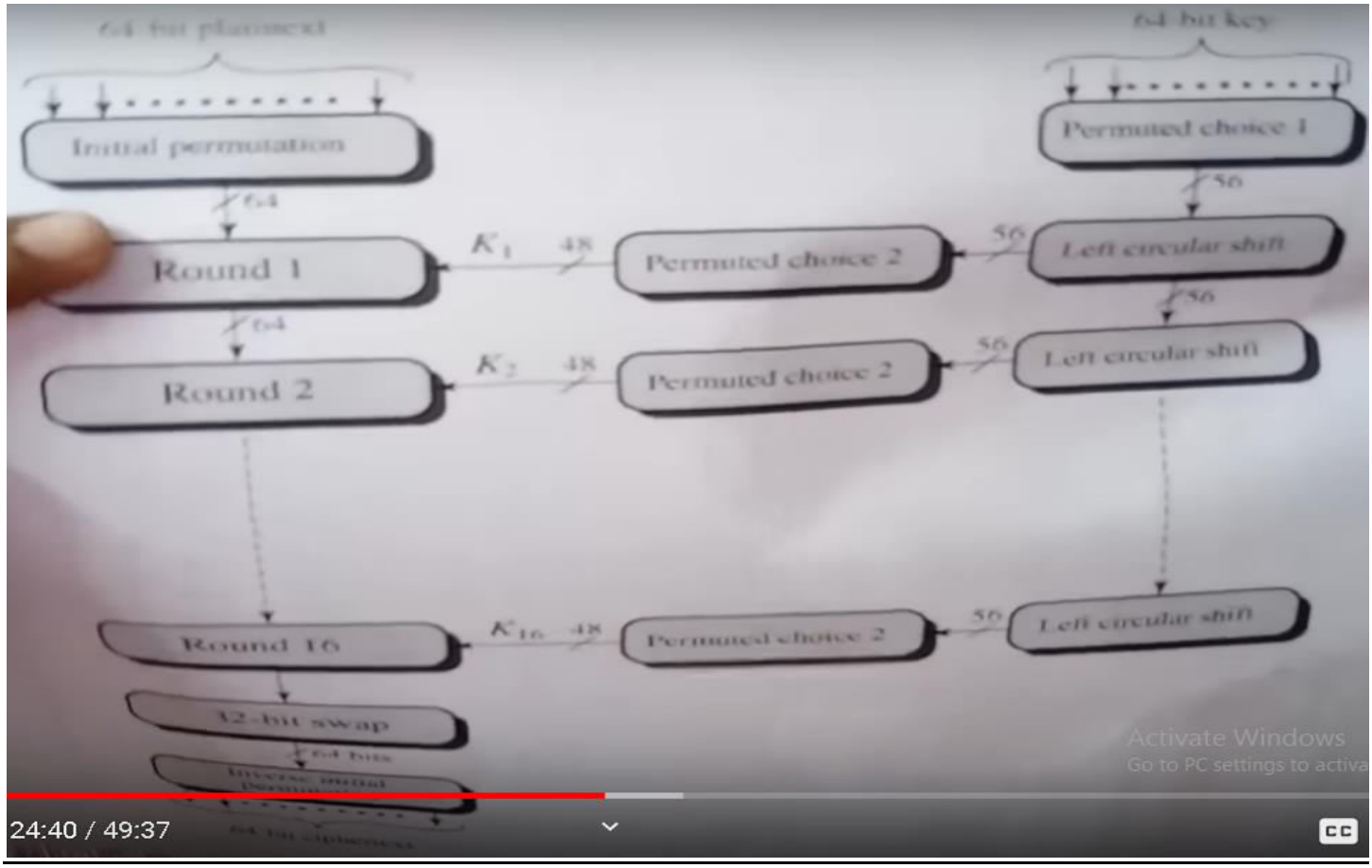
I. Left pt -> xor2 -> right pt next round
II. Right pt ->(32) -> expansion perm -> (48) -> xor1 -> sbox ->(32) -> perm [transposition order] -> xor2 -> right pt next round
III. Right pt -> left pt next round
IV. Left key (28) -> lcs -> left key next round
V. right key (28) -> lcs -> right key next round
VI. left/right -> lcs -> (28+28) -> contraction -> (48) ->xor1



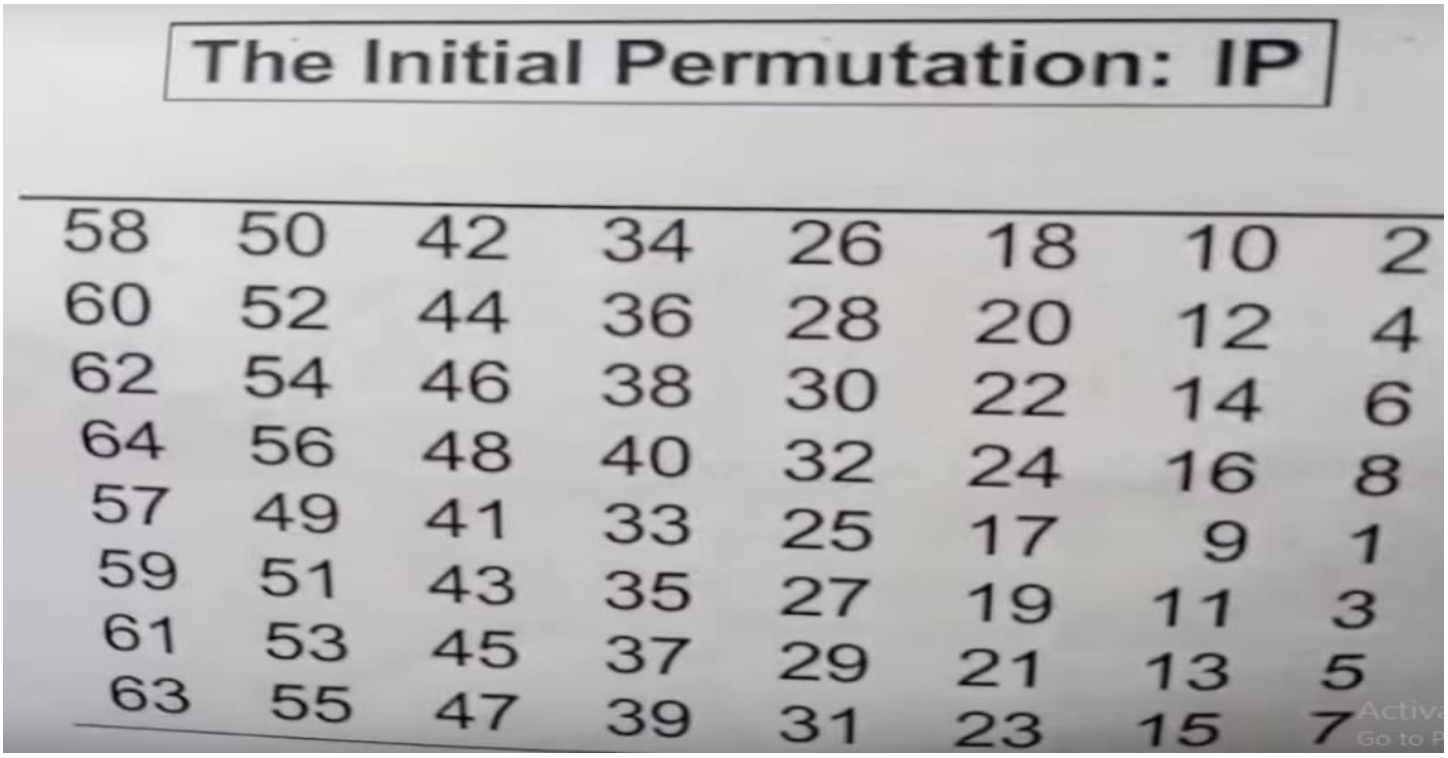
I. <u>pt input (8*8 form)</u>



II. input to initial per (des)



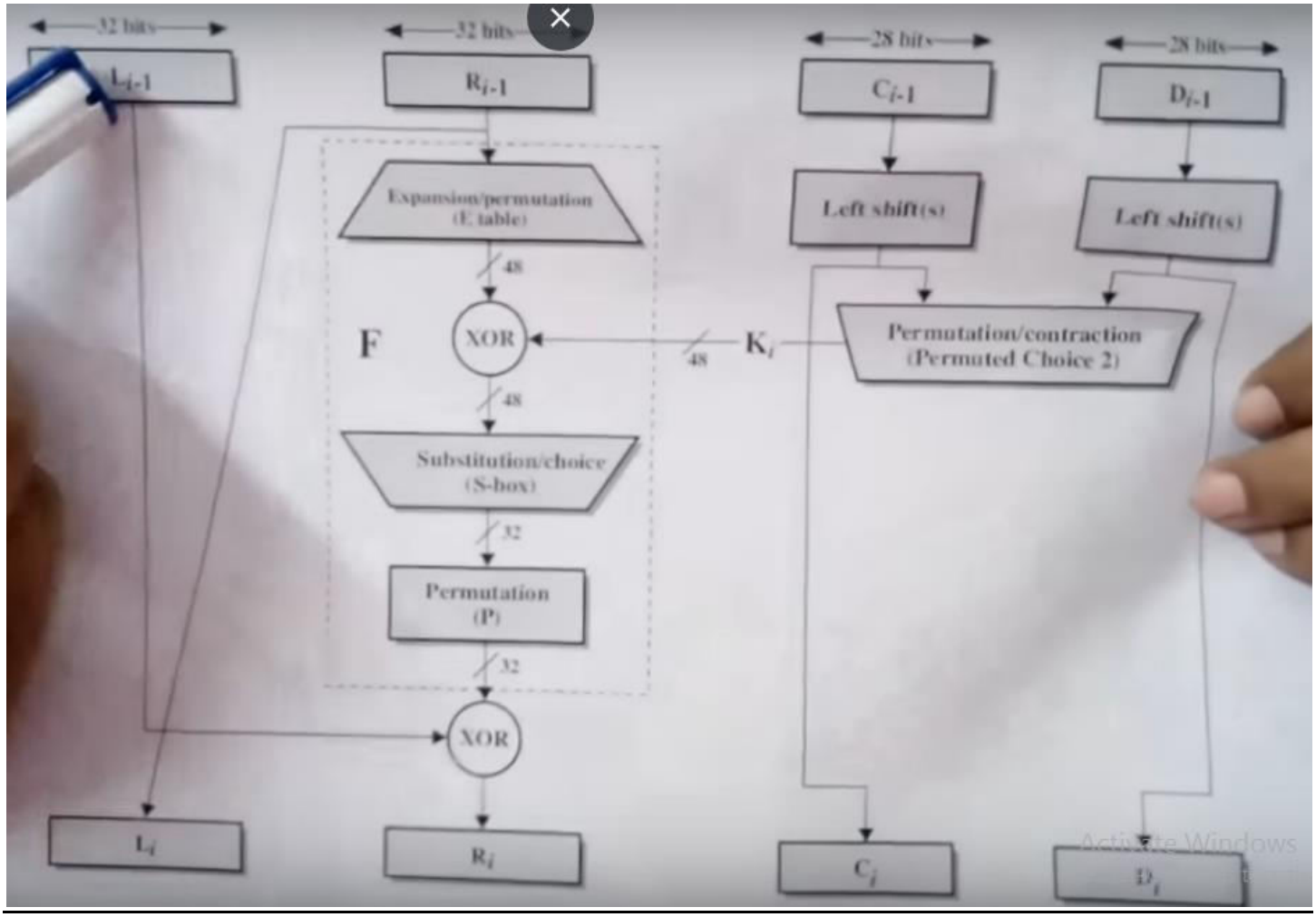
III. Initial per (transposition)



The Initial Permutation: IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IV. Rounf fn (process pt + key)



V. Expansion per (add 16 bits)

EXPANSION PERMUTATION

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

VI. Perm choice 1 (remove multiple of 8)

PERMUTED CHOICE 1

PC-1	57	49	41	33	25	17	9	1	58	50	42	34	26	18
	10	2	59	51	43	35	27	19	11	3	60	52	44	36
	63	55	47	39	31	23	15	7	62	54	46	38	30	22
	14	6	61	53	45	37	29	21	13	5	28	20	12	4

VII. Lcs (bits shifted depend upon round no)

Round	1	2	3	4	5	6	7	8	9	10
Bits Shifted	1	1	2	2	2	2	2	2	1	2

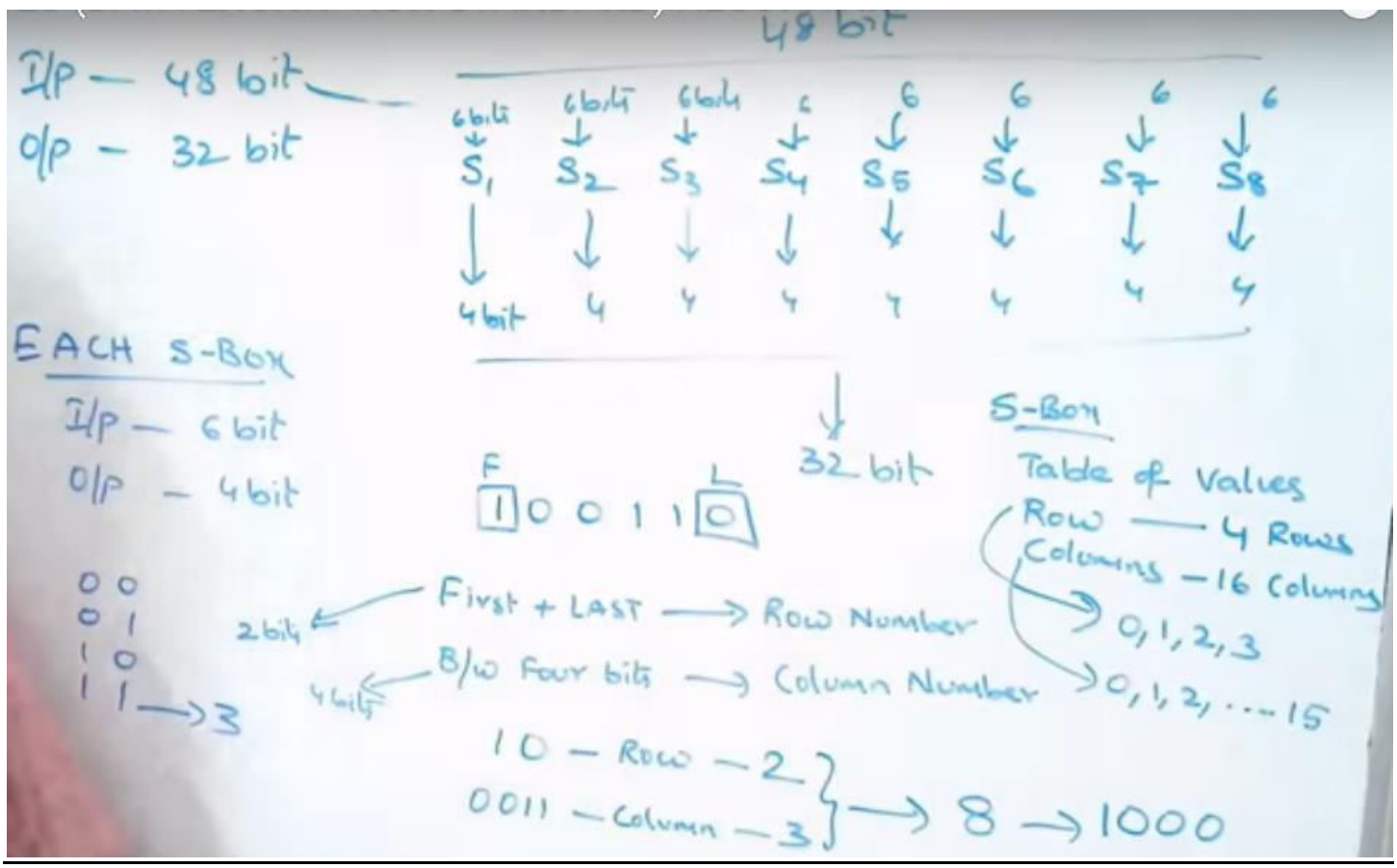
Round	11	12	13	14	15	16
Bits Shifted	2	2	2	2	2	1

VIII. <u>Contraction/perm choice 2 (some 8 bits reduced and rearrange)</u>

PERMUTED CHOICE 2

PC-2	14	17	11	24	1	5	3	28	15	6	21	10
	23	19	12	4	26	8	16	7	27	20	13	2
	41	52	31	37	47	55	30	40	51	45	33	48
	44	49	39	56	34	53	46	42	50	36	29	32

IX. <u>S boxes (8 s boxes)</u>



S box with 4 rows 16 cols

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

X. Permutaion func (rearrange according to given bit positions)

(d) Permutation Function (P)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

XI. Xor left & right data

XII. 32 bit swap

XIII. Final permutation (transpose bits in given order)

Final Permutation (IP^{-1})

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

#13 NETWORK SECURITY- AES (ADVANCED ENCRYPTION STANDARD) Algorithm

AES (ADVANCED ENCRYPTION STANDARD)

Block Size — 128 bit Plain Text (4 words / 16 Bytes)

No. of Rounds — 10 Rounds

1 → words
1
32 bit

Key Size — 128 bit (4 words / 16 Bytes)

No. of Subkeys — 44 Subkeys

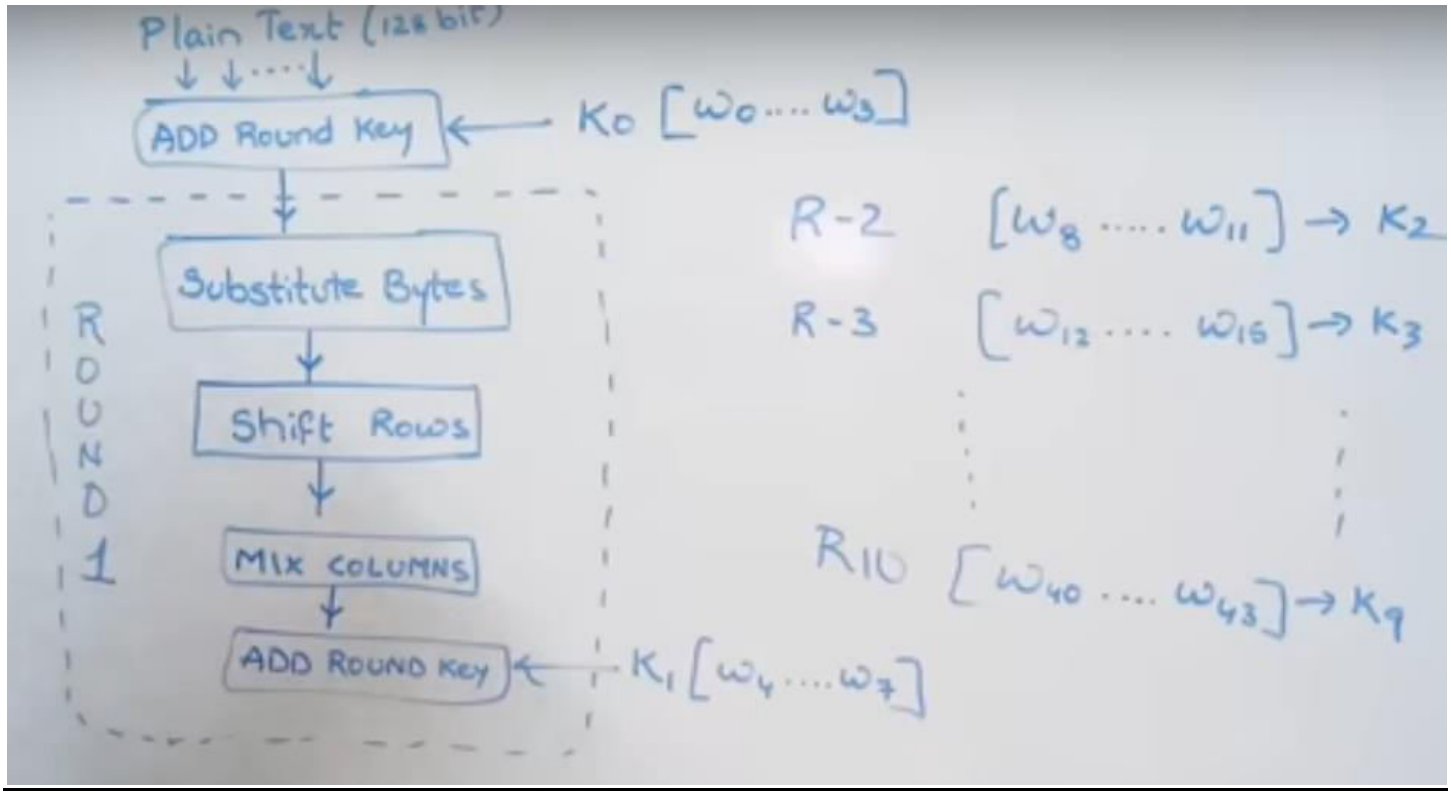
Each Subkey Size — 32 bit / 1 word / 4 Bytes

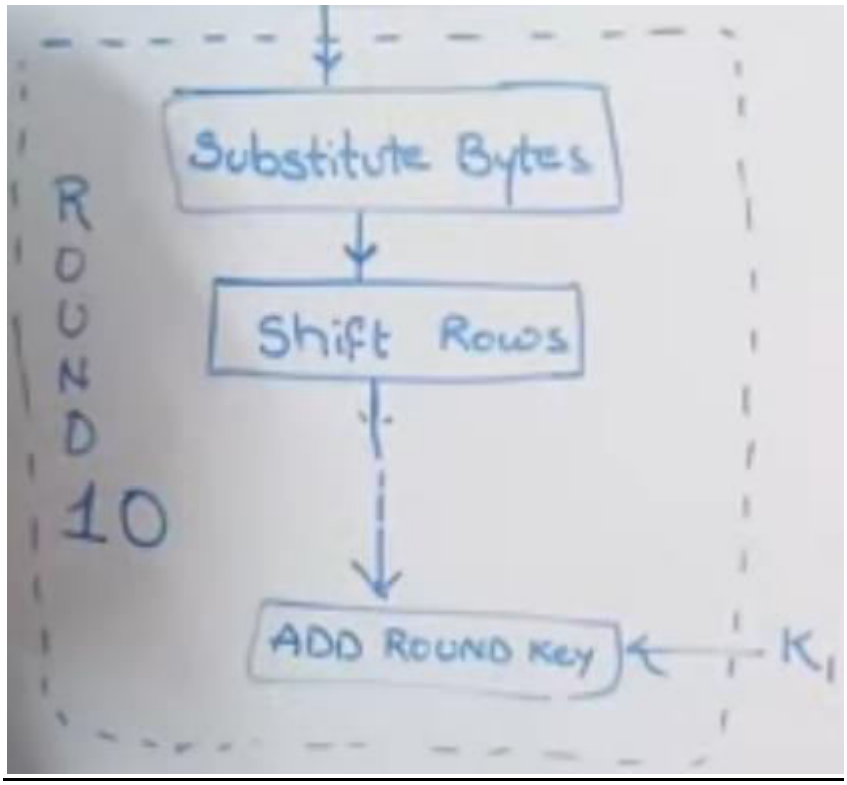
Each Round — 4 Subkeys (128 bit / 4 words / 16 Bytes)

Pre Round Caluclation — 4 subkeys (128 bit / 4 words / 16 Bytes)

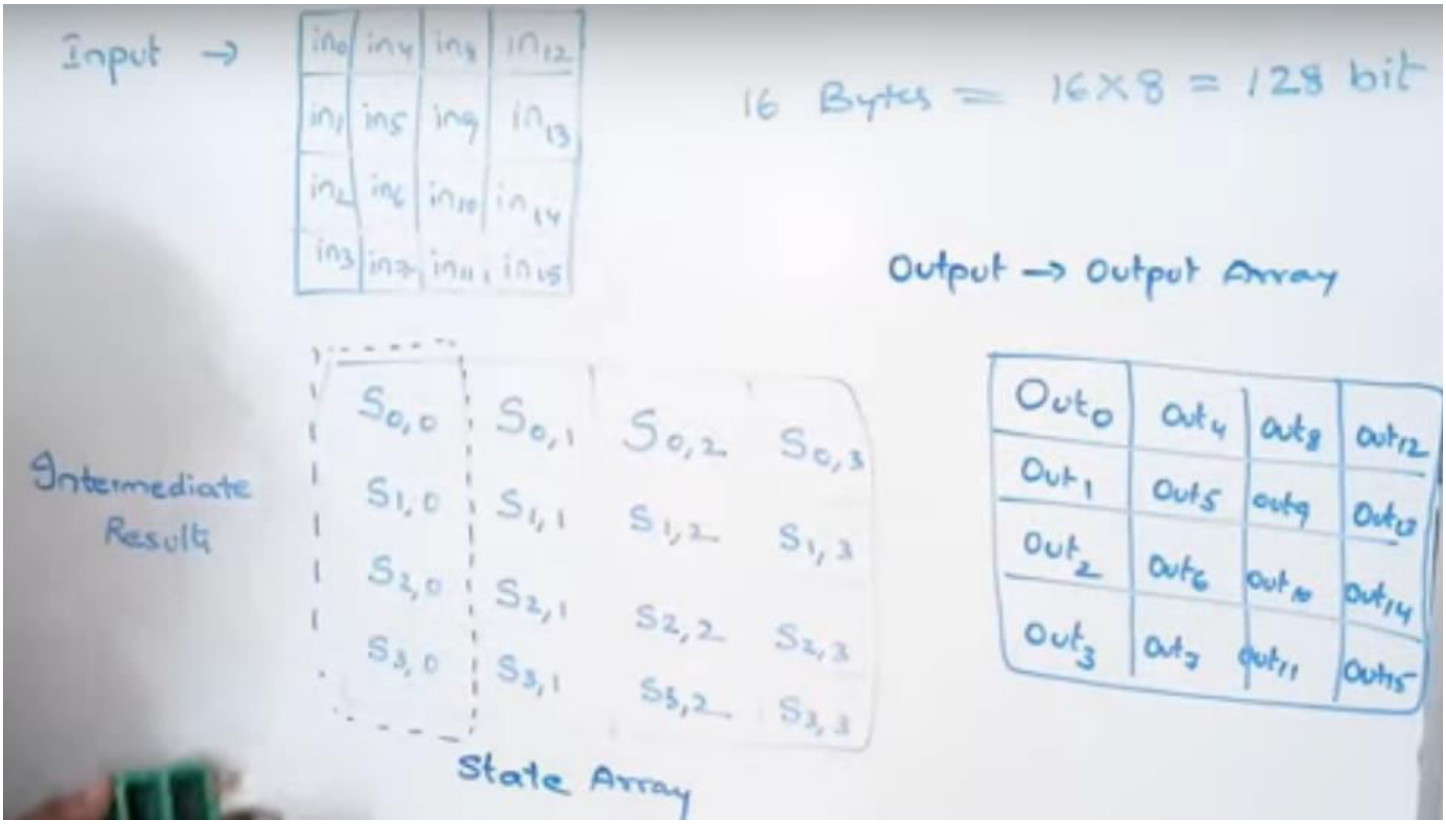
Cipher Text — 128 bit (4 words / 16 Bytes)

Activate Windows
Go to PC settings to activate Windows

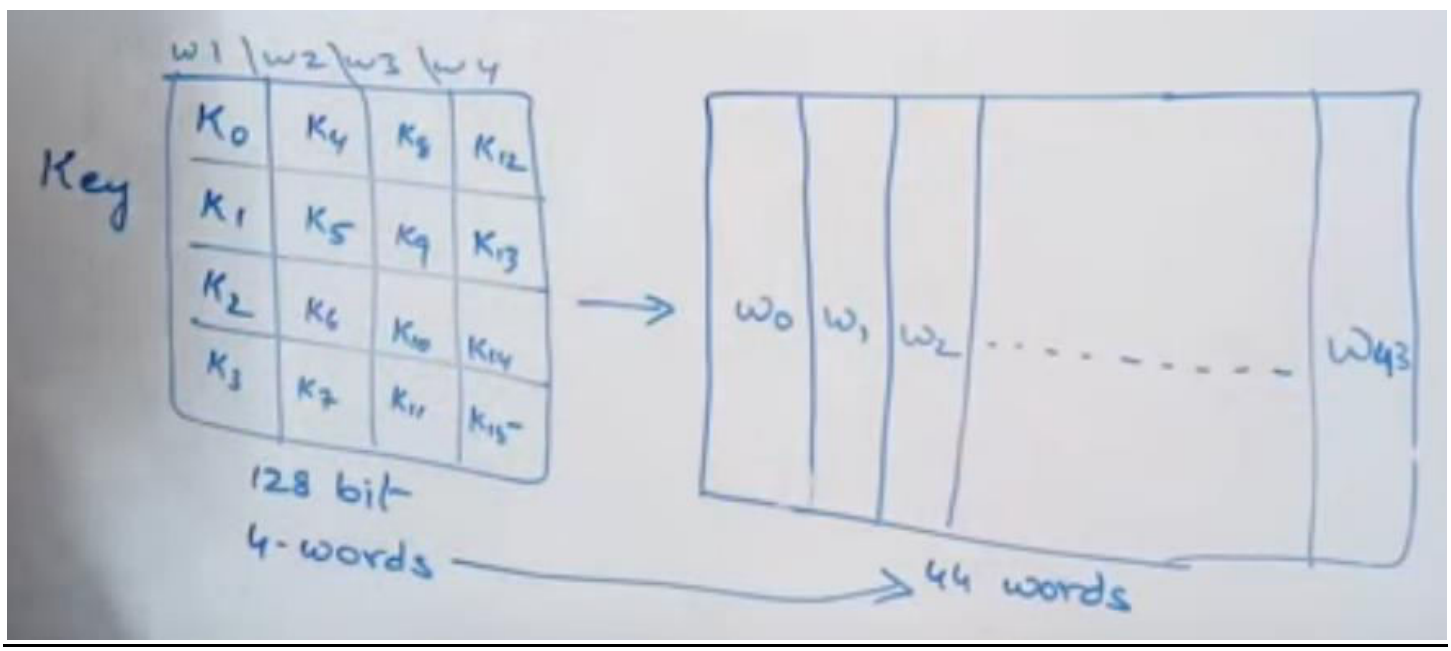

Plain Text (128 bit)
ADD Round Key
K0 [w0 .... w3]
ROUND 1
Substitute Bytes
Shift Rows
MIX COLUMNS
ADD ROUND KEY
K1 [w4 .... w7]
R-2 [w8 ..... w11] → K2
R-3 [w12 .... w15] → K3
R10 [w40 .... w43] → K9


ROUND 10
Substitute Bytes
Shift Rows
ADD ROUND KEY
K1

I. Input pt (4*4 table, each cell 1 byte)

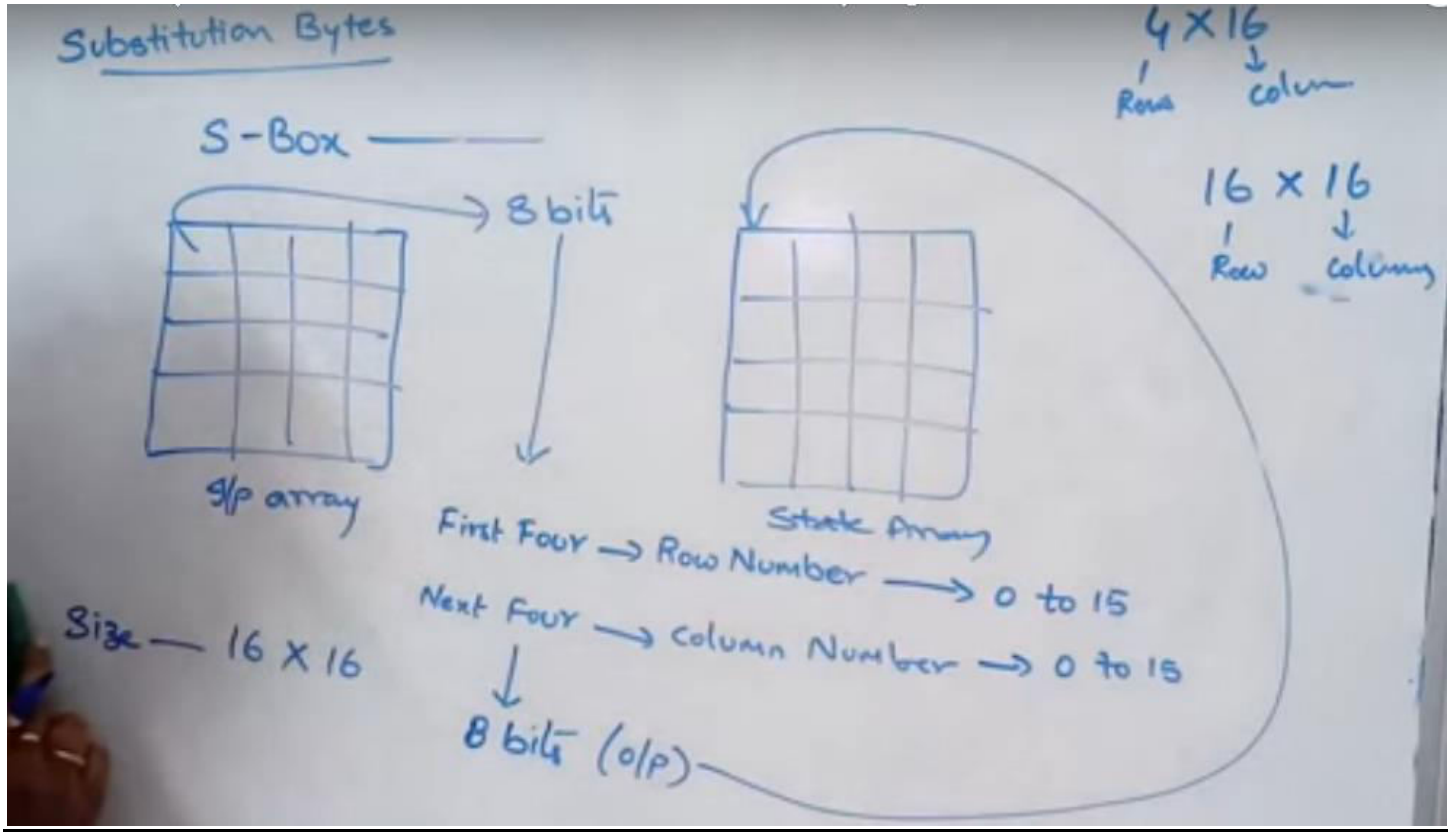


Intermediate results stored in state array (each col = 1 word) [ith byte of jth word]

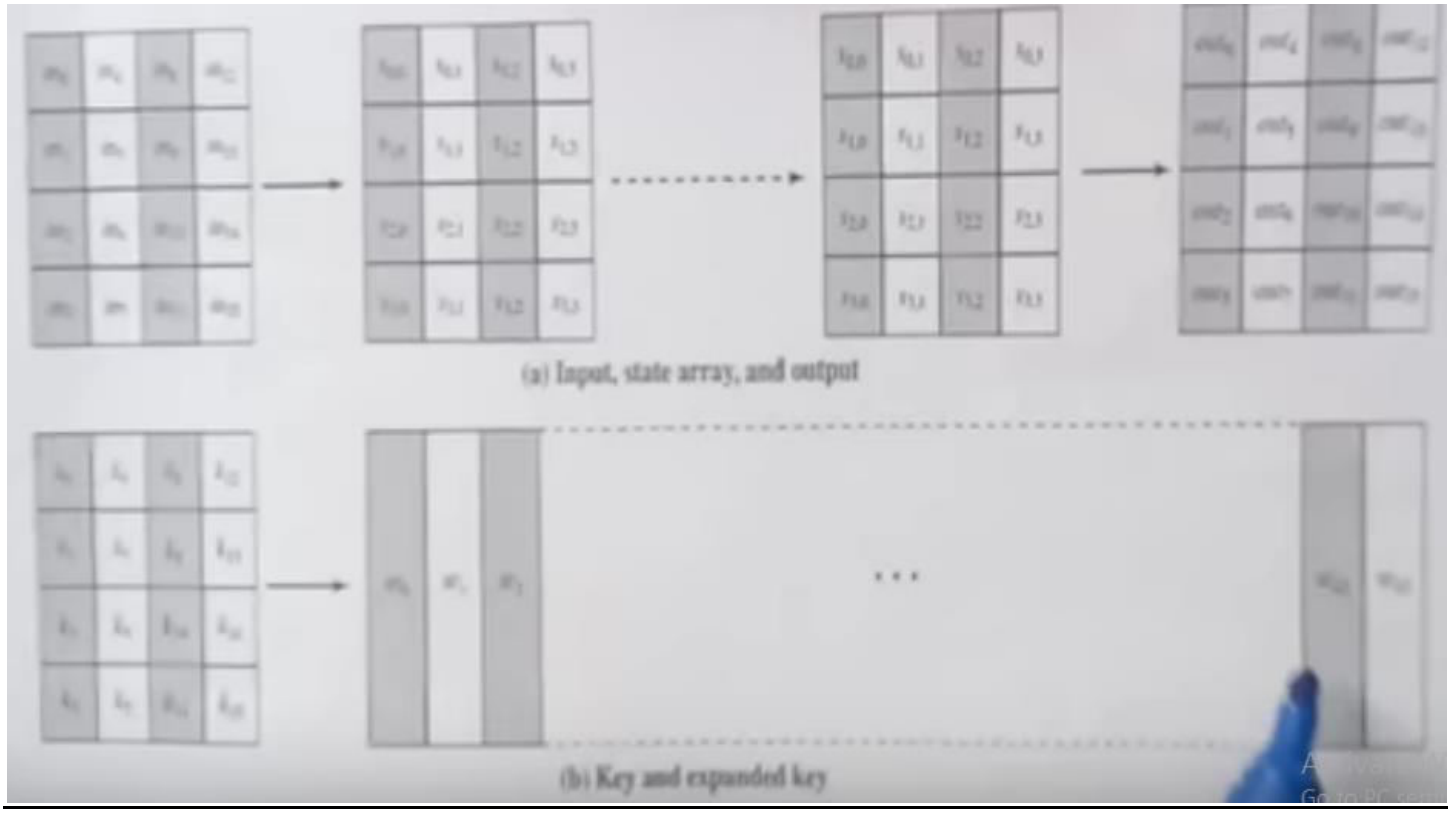


Key is of 4 words, it is expanded to 44 words (each subkey = 1 word)

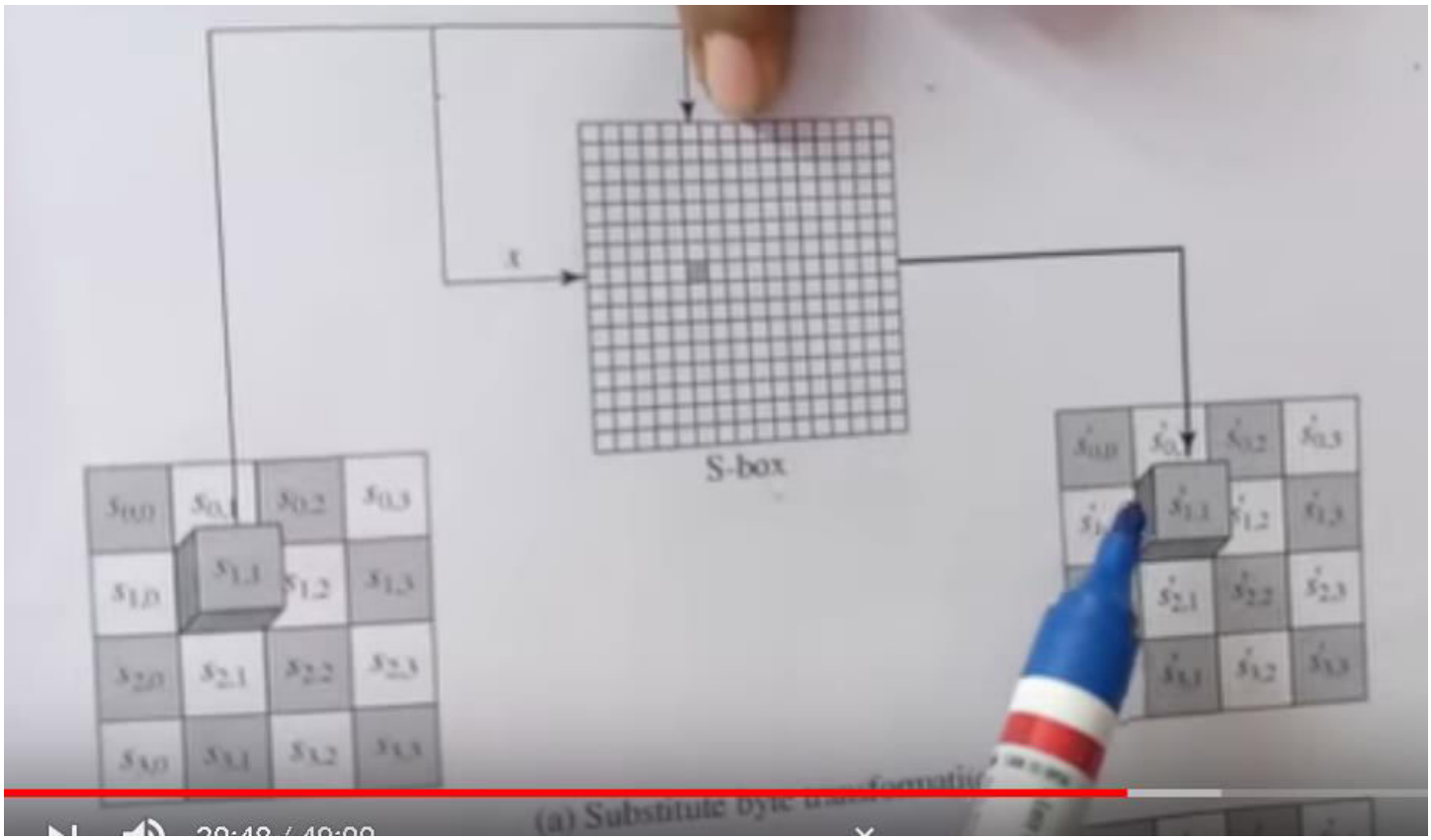
II. Substitution bytes (implementation of s box)



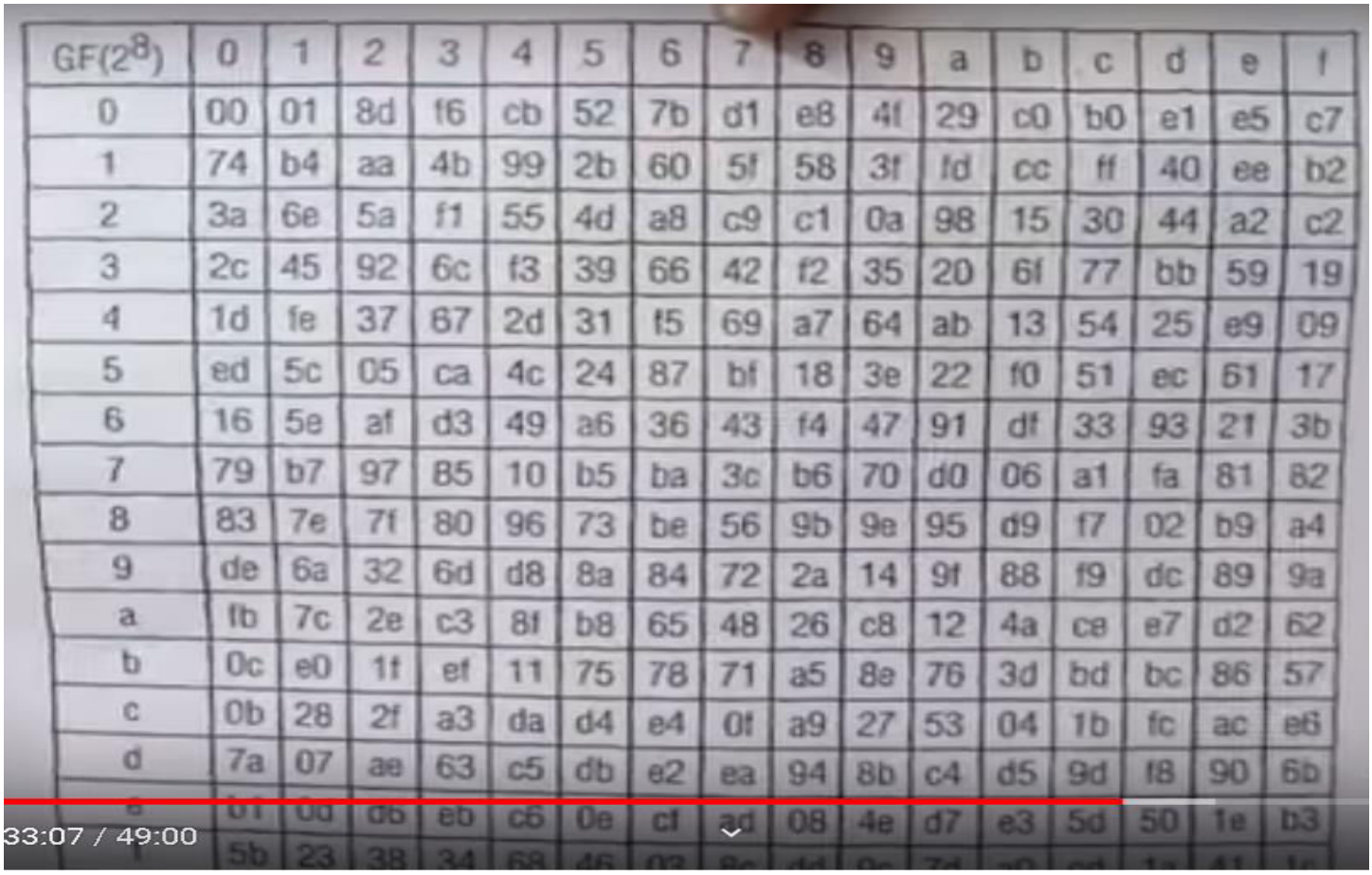
[AES = 16*16 S box, DES = 4*16]



[format of diff tables/array]



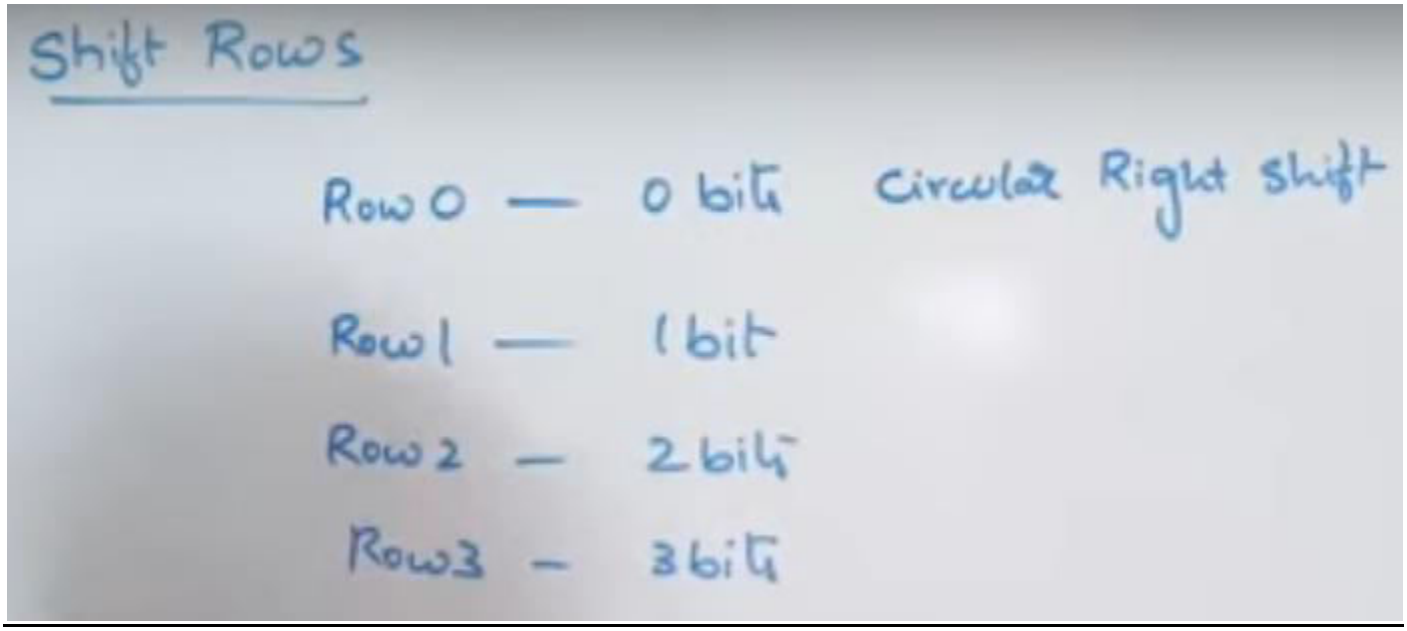
[s box use]

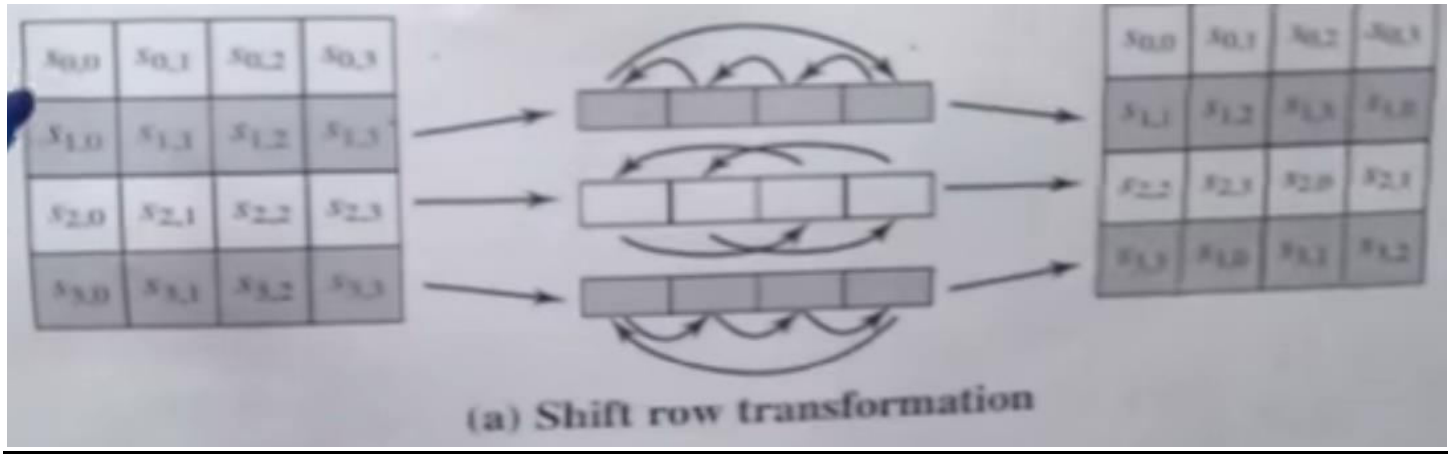


GF(2^8)	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	00	01	8d	f6	cb	52	7b	d1	e8	4f	29	c0	b0	e1	e5	c7
1	74	b4	aa	4b	99	2b	60	5f	58	3f	fd	cc	ff	40	ee	b2
2	3a	6e	5a	f1	55	4d	a8	c9	c1	0a	98	15	30	44	a2	c2
3	2c	45	92	6c	f3	39	66	42	f2	35	20	6f	77	bb	59	19
4	1d	fe	37	67	2d	31	f5	69	a7	64	ab	13	54	25	e9	09
5	ed	5c	05	ca	4c	24	87	bf	18	3e	22	f0	51	ec	61	17
6	16	5e	af	d3	49	a6	36	43	f4	47	91	df	33	93	21	3b
7	79	b7	97	85	10	b5	ba	3c	b6	70	d0	06	a1	fa	81	82
8	83	7e	7f	80	96	73	be	56	9b	9e	95	d9	f7	02	b9	a4
9	de	6a	32	6d	d8	8a	84	72	2a	14	9f	88	f9	dc	89	9a
a	fb	7c	2e	c3	8f	b8	65	48	26	c8	12	4a	ce	e7	d2	62
b	0c	e0	1f	ef	11	75	78	71	a5	8e	76	3d	bd	bc	86	57
c	0b	28	2f	a3	da	d4	e4	0f	a9	27	53	04	1b	fc	ac	e6
d	7a	07	ae	63	c5	db	e2	ea	94	8b	c4	d5	9d	f8	90	6b
e	b1	0d	d6	eb	c6	0e	cf	ad	08	4e	d7	e3	5d	50	1e	b3
f	5b	23	38	34	68	46	03	8c	dd	9c	7d	a0	cd	1a	41	1c

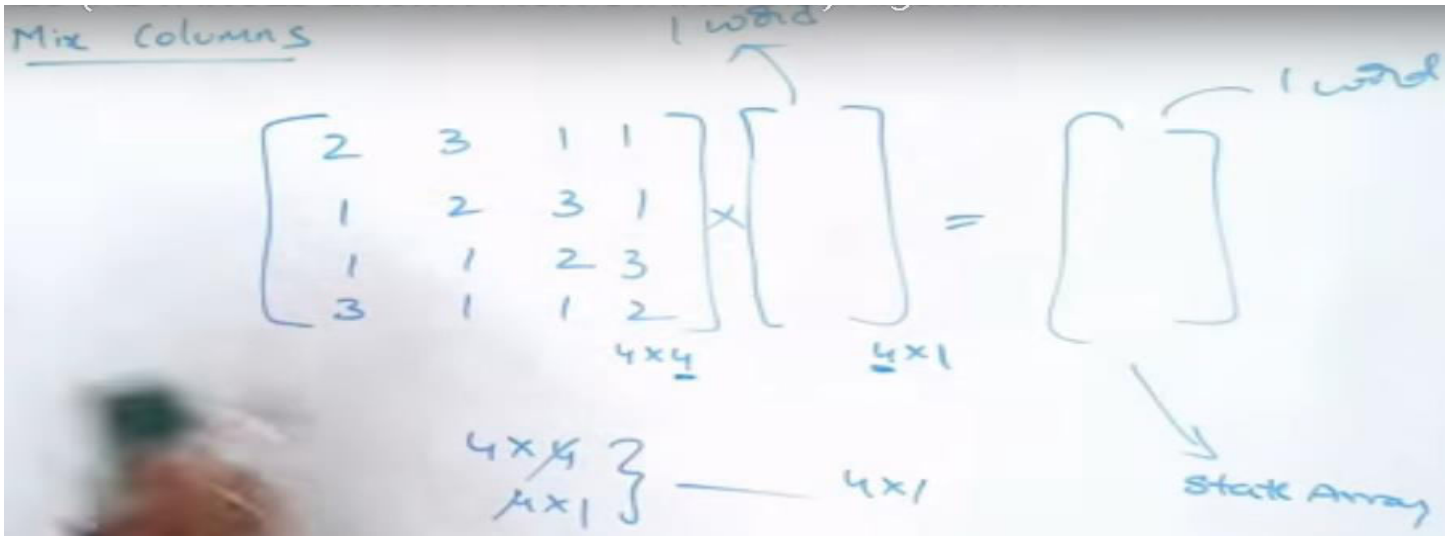
[s box format]

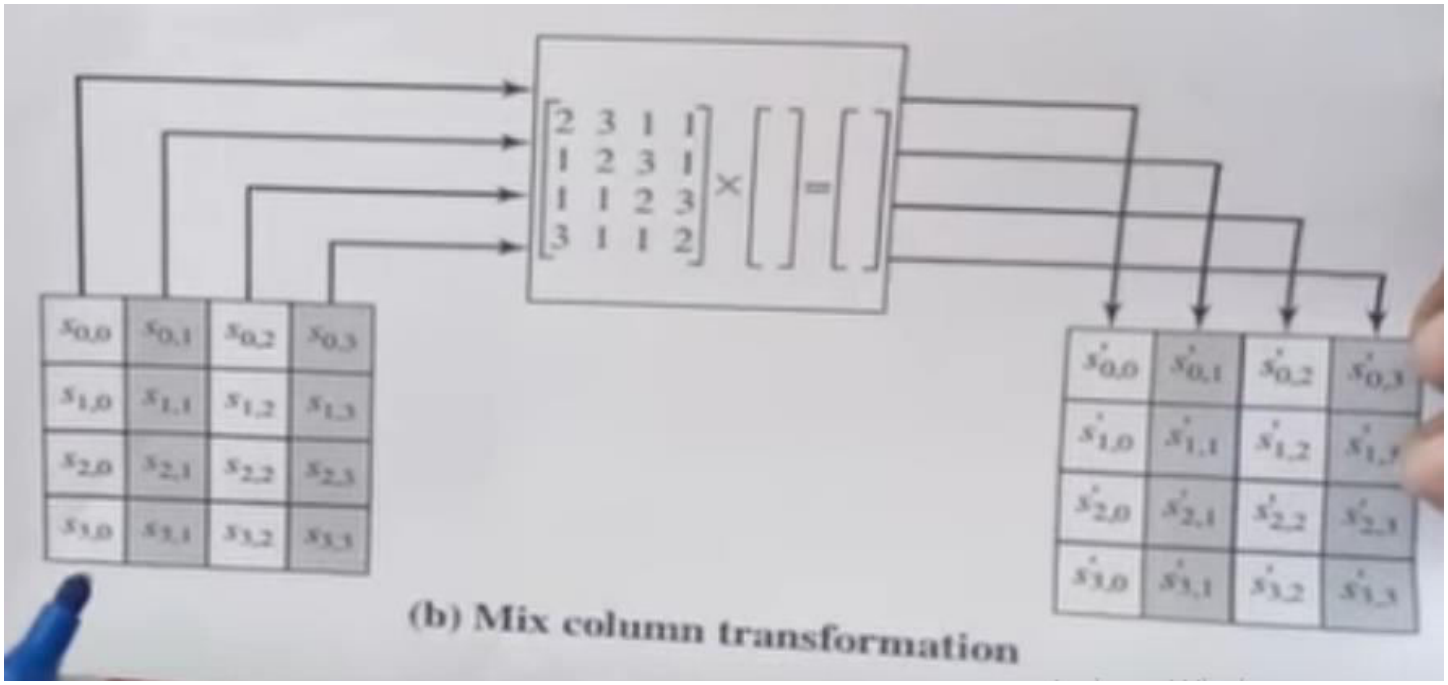
III. Shift rows (shift rows[right circular shift] in state array acc to row no)



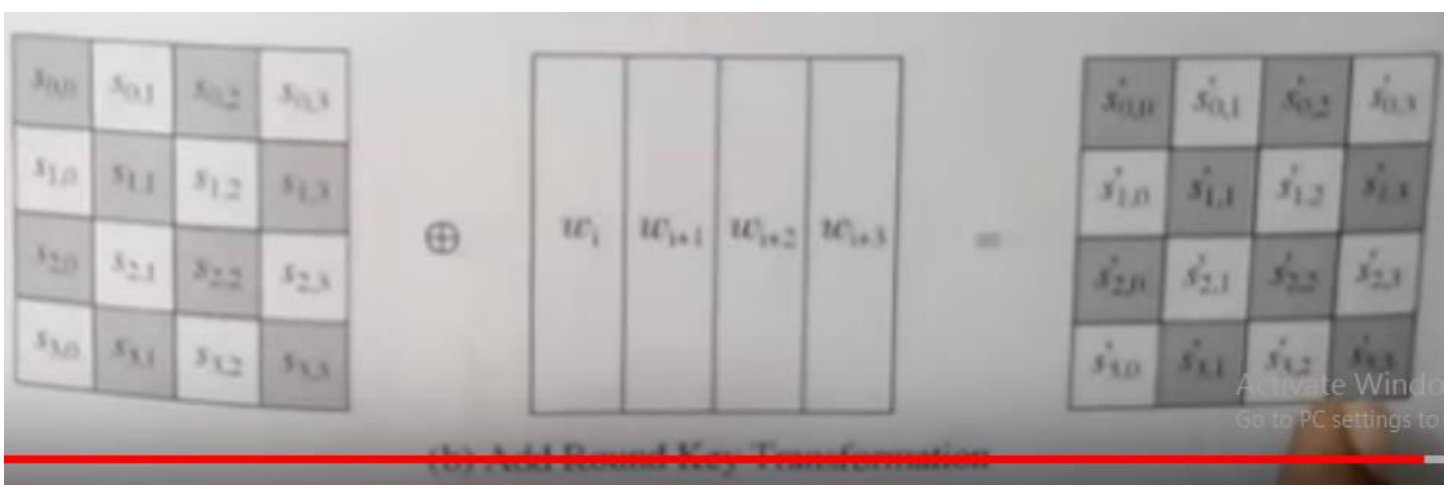


IV. Mix col transformation (matrix multiplication of each word in input state with predefined matrix]



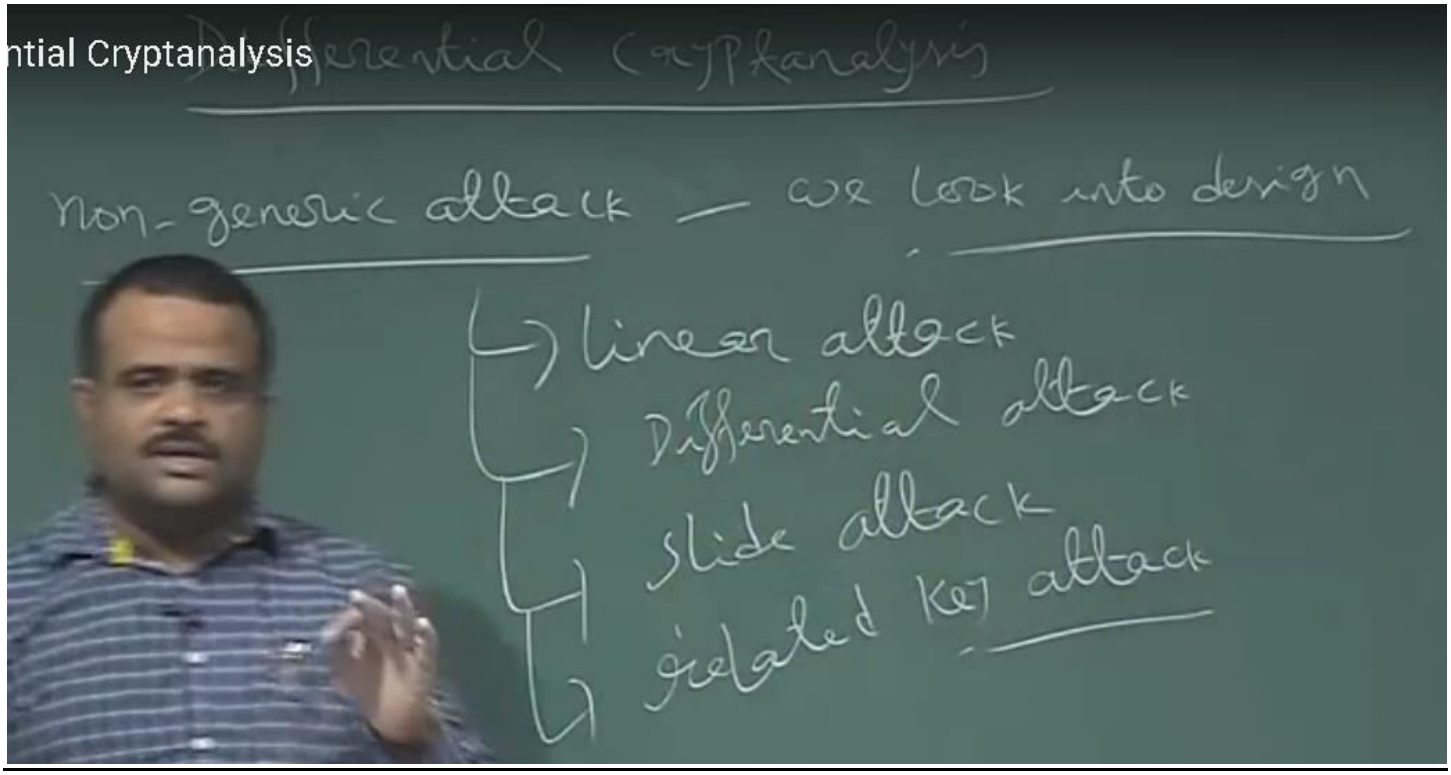


V. <u>Add round key (xor each col/word of state array with each word in key matrix)</u>

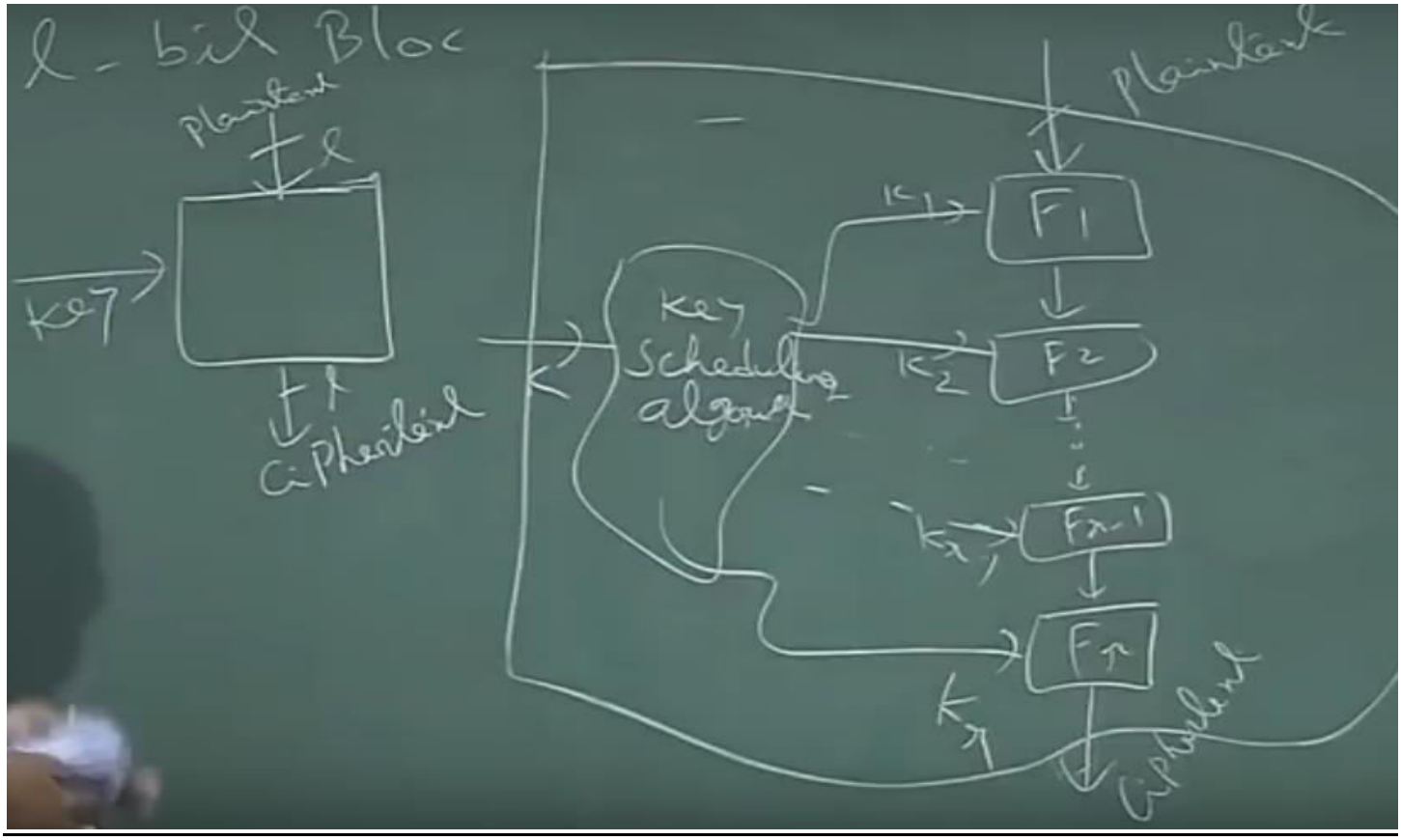


[don’t include mix columns step in round 10]

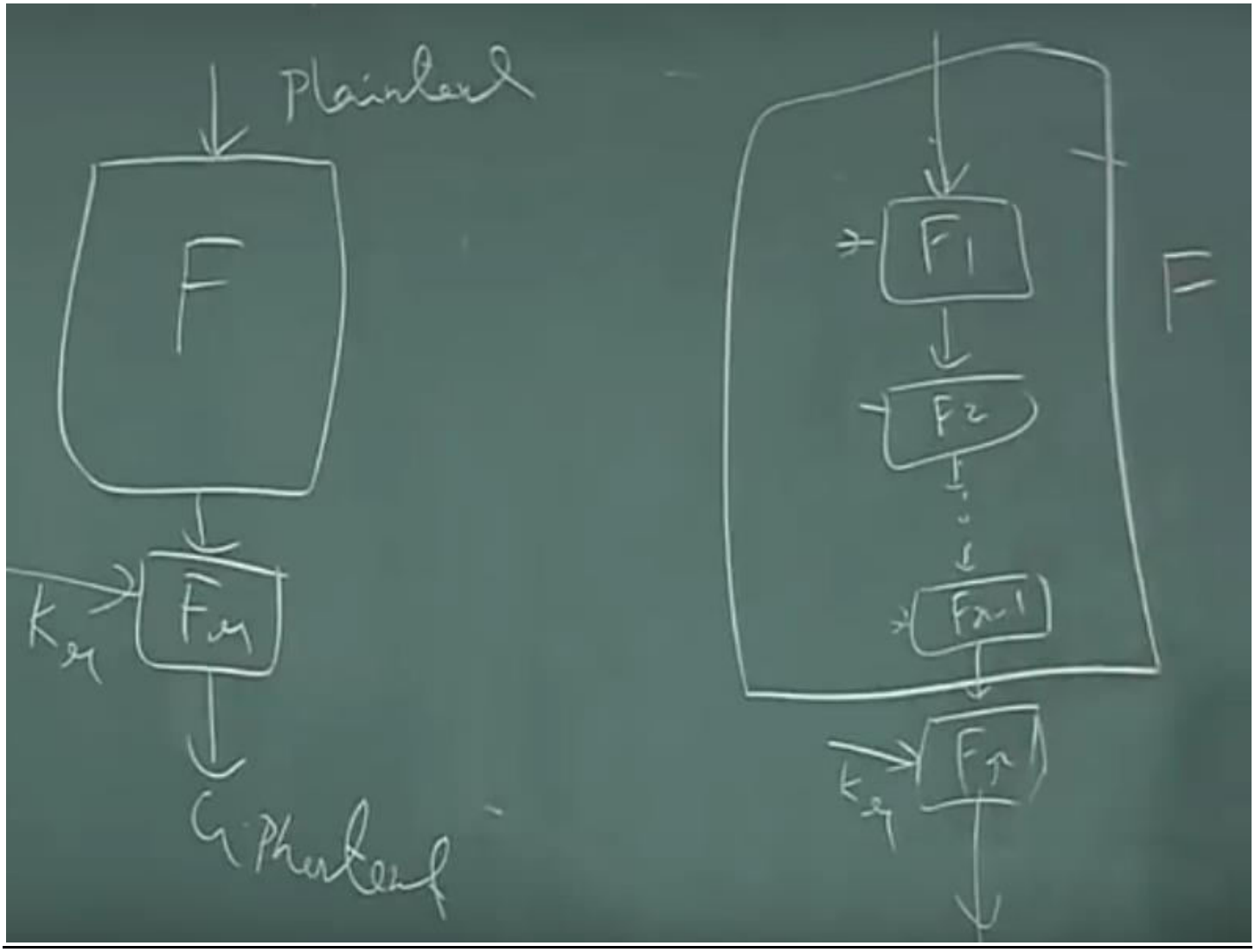
#14 Differential Cryptanalysis (for block cipher)



[attack specific to that particular cipher]



(R- round block cipher)



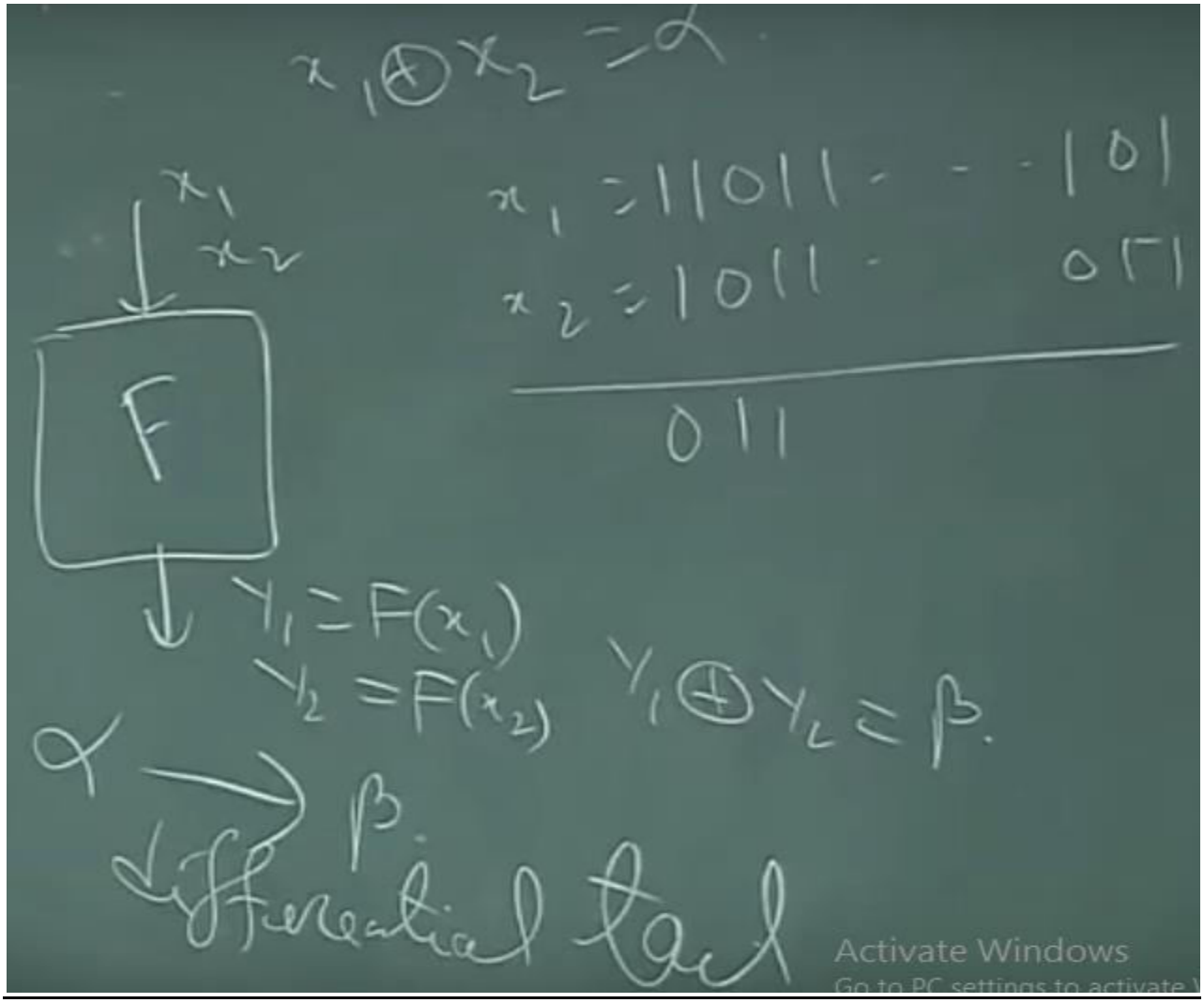
(in differential attack, attacker wants to get rth round key)

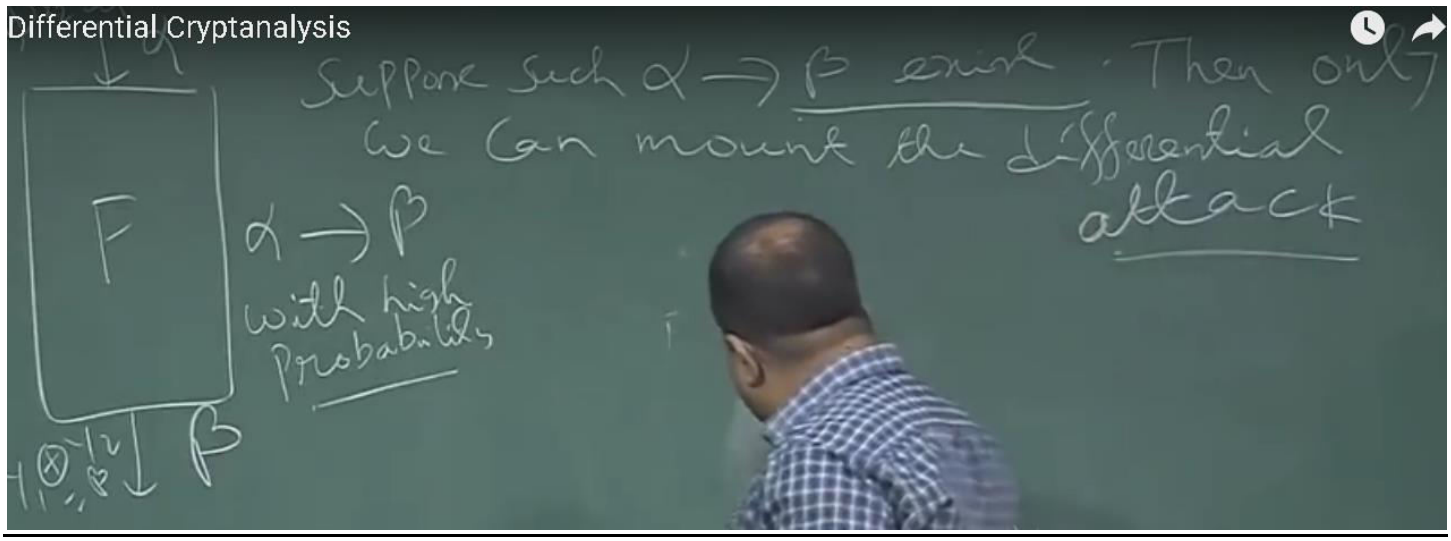
[we treat r-1 rounds as F fn]

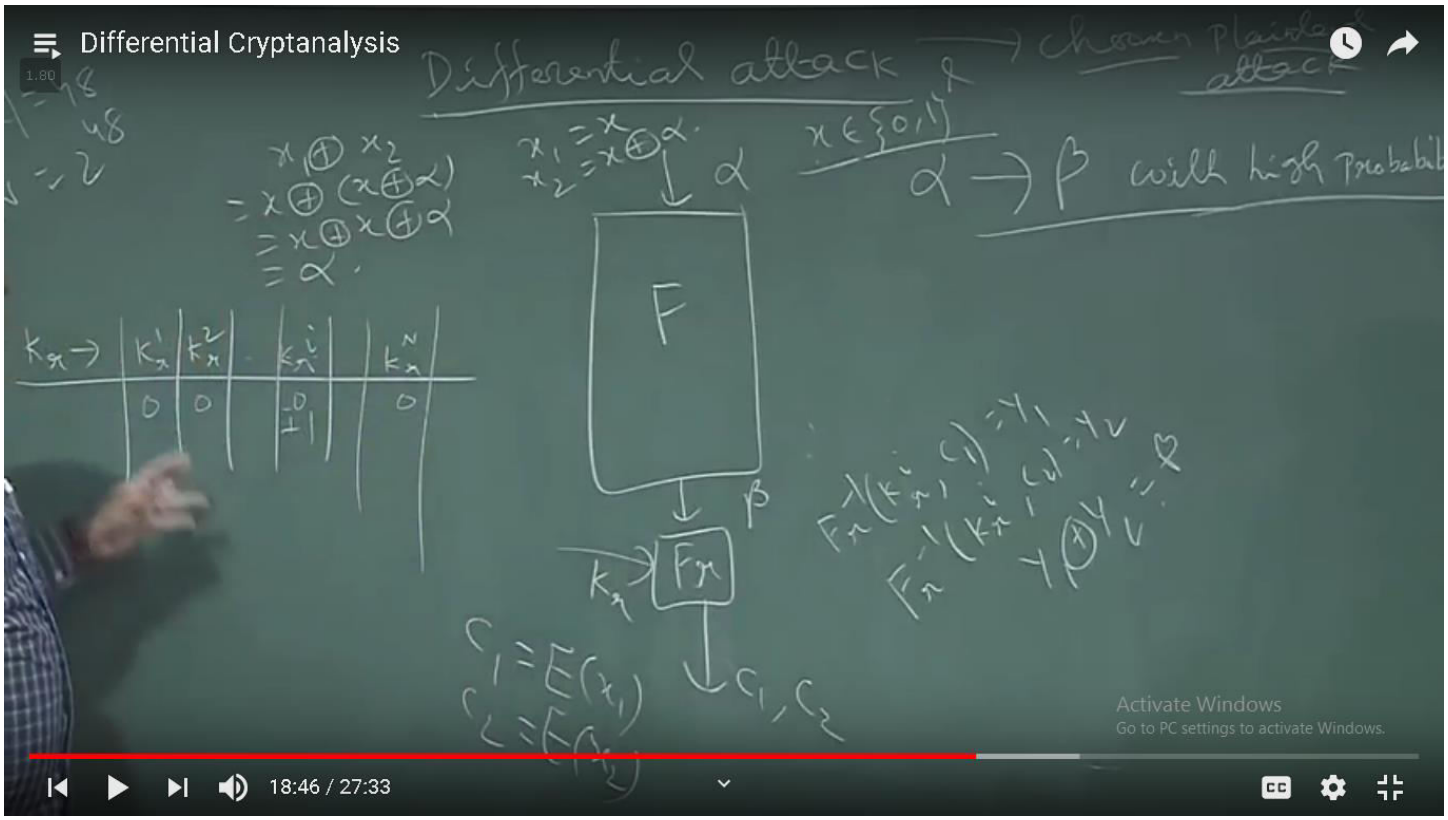
Differential tail : If we get the alpha, beta with high probability then only diff attack is possible

Alpha = input diif (bitwise xor, 64 bits)

Beta = output diff


$x_1 \oplus x_2 = \alpha$
x_1
x_2
F
$x_1 = 11011 \cdots 101$
$x_2 = 10110 \cdots 011$
011
$Y_1 = F(x_1)$
$Y_2 = F(x_2)$
$Y_1 \oplus Y_2 = \beta$.
$\alpha \rightarrow \beta$
Differential tool
Activate Windows


Differential Cryptanalysis
Suppose such $\alpha \rightarrow \beta$ exist. Then only we can mount the differential attack
F
$\alpha \rightarrow \beta$
with high probability
β

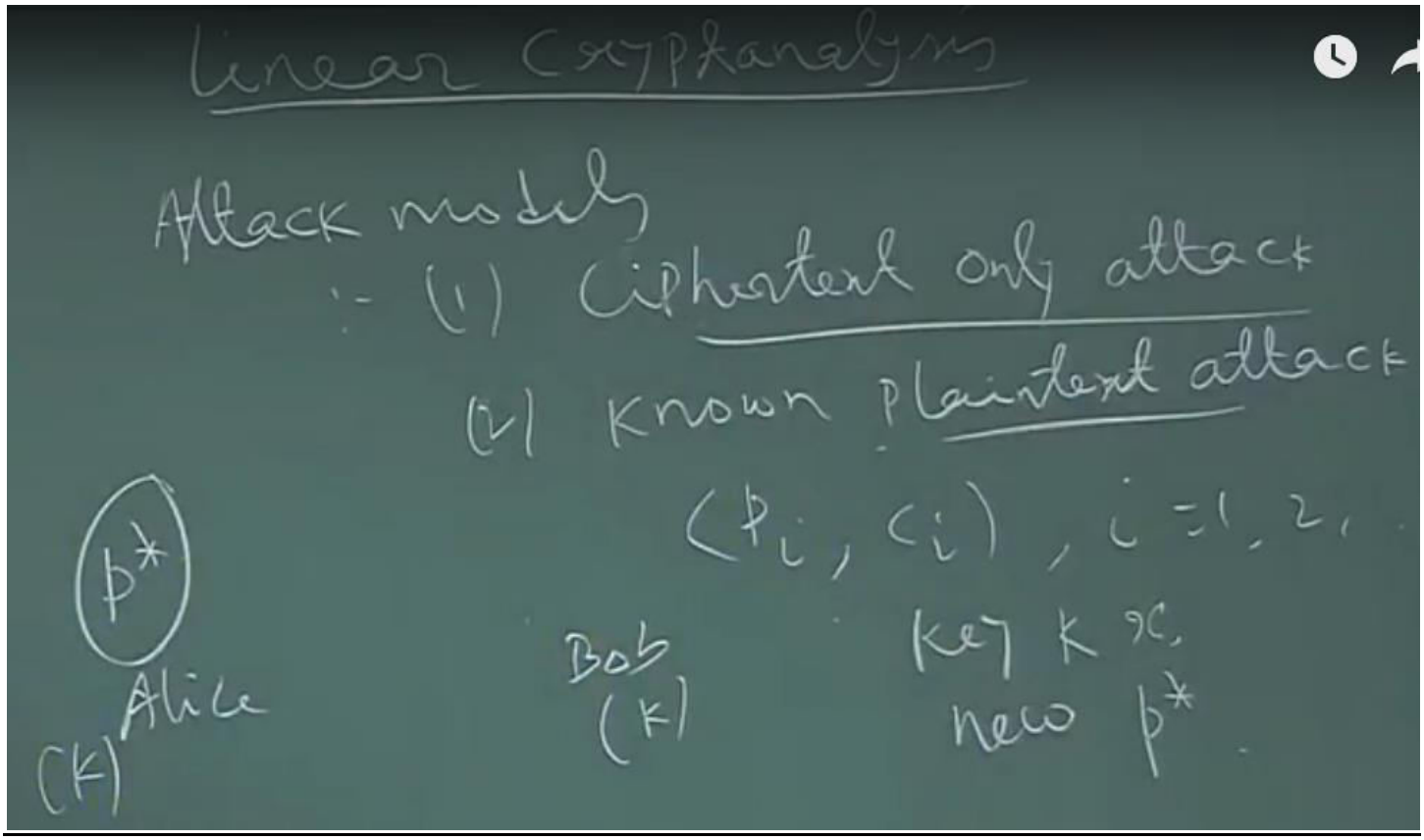


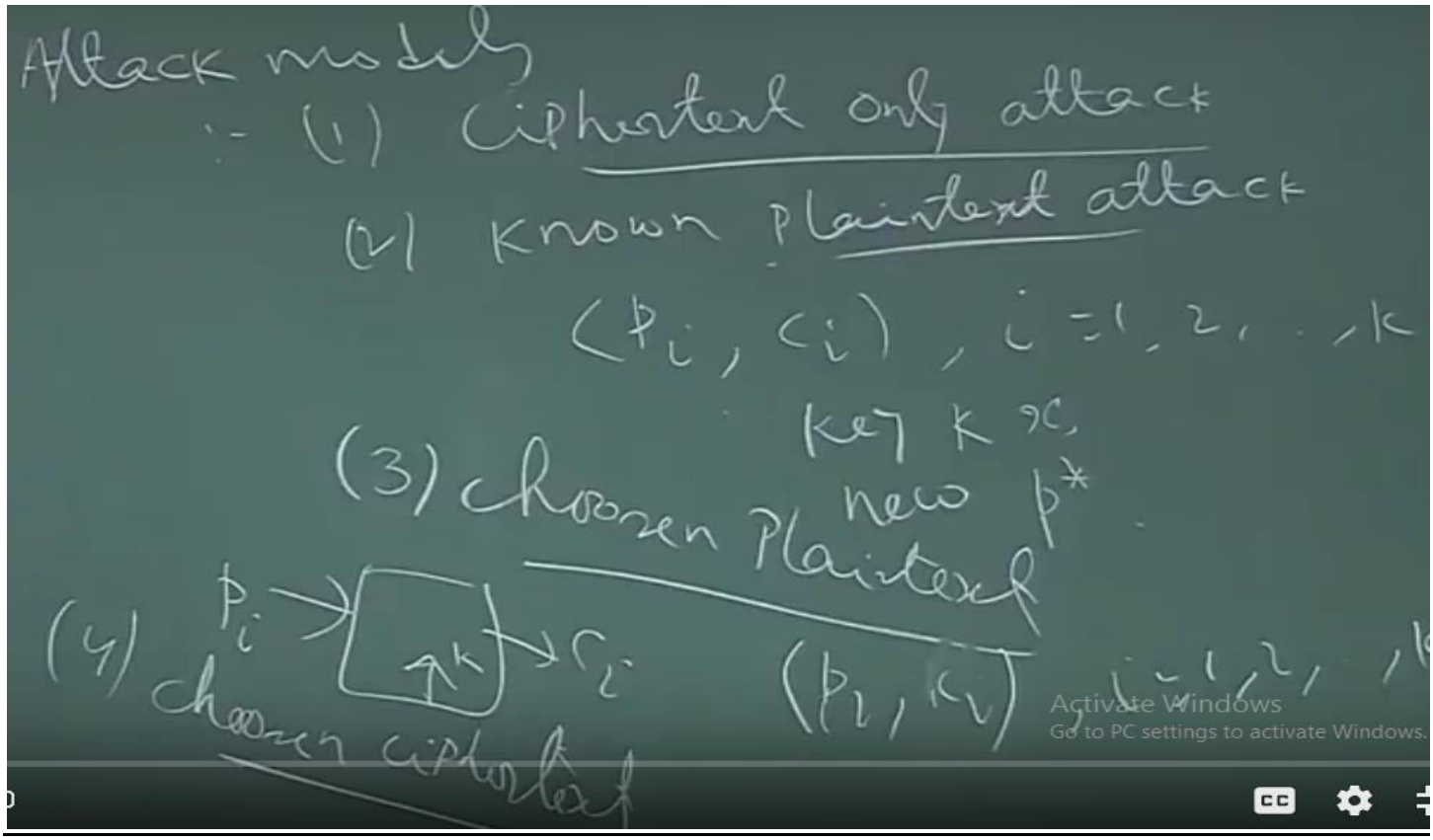
I. We choose x1 , x2 such that diff is alpha (chosen pt attack) [alpha,beta are known]
II. If x is l bit, there are 2^L options
III. We make a table of all possible b=values of Kr [Brute force attack]
IV. Initialize counter by 0.
V. If candidate key works for C1,C2 [gives beta after F^-1], we increment counter

#15 linear cryptanalysis

Linear Cryptanalysis

- A more recent development is linear cryptanalysis that was presented by Mitsuru Matsui at Eurocrypt '93.
- This attack is based on finding linear approximations to describe the transformations performed in DES (and other block ciphers).
- This method can find a DES key given 2^{47} *known* plaintexts, as compared to 2^{47} *chosen* plaintexts for differential cryptanalysis (it is therefore a known plaintext attack although it can also work as a ciphertext only attack).





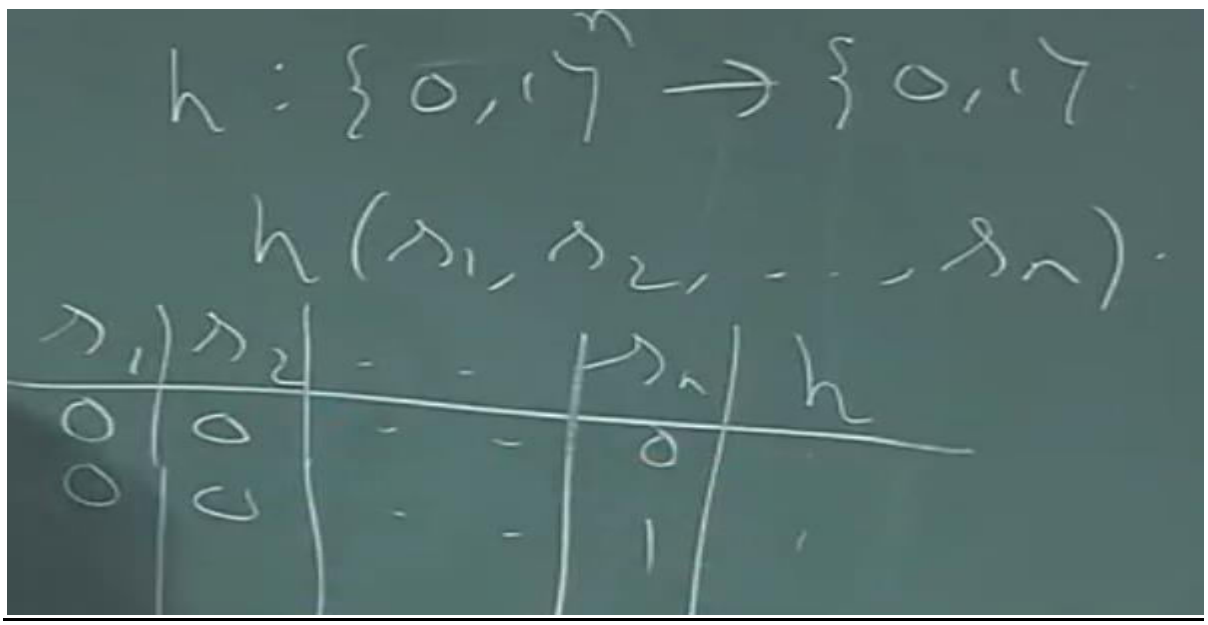
[DES = chosen pt attack, AES = known pt attack]

- To understand the attack we will define a few terms:

- A Boolean function $h : Z_2^n \rightarrow Z_2$ in n variables $s_1, \ldots, s_n$ is **linear** if it can be represented as $h(s) = a_1 s_1 \oplus \ldots \oplus a_n s_n$ for some $a_i \in Z_2 = \{0, 1\}, i = 1, \ldots, n$. The set of all linear Boolean functions in n variables is denoted by

$$L_n = \{h : Z_2^n \rightarrow Z_2 | h = a_1 s_1 \oplus \ldots \oplus a_n s_n\}$$

- A Boolean function $f : Z_2^n \rightarrow Z_2$ is called **affine** if either $f(s) = h(s)$ or $f(s) = h(s) \oplus 1$, for some $h(s) \in L_n$. The set of all affine Boolean



(Boolean function)

function in n variables is therefore:

$$A_n = L_n \cup \{h \oplus 1 | h \in L_n\} = L_n \cup \overline{L_n}$$

- In other words, A_n consists of all linear functions and their negations.

Summary of Attack

- For a cipher with n-bit plaintext and ciphertext blocks and an m-bit key, let the plaintext block be labeled $P[1], \ldots P[N]$, the cipher text block $C[1], \ldots C[n]$ and the key $K[1], \ldots K[m]$.
- Then define

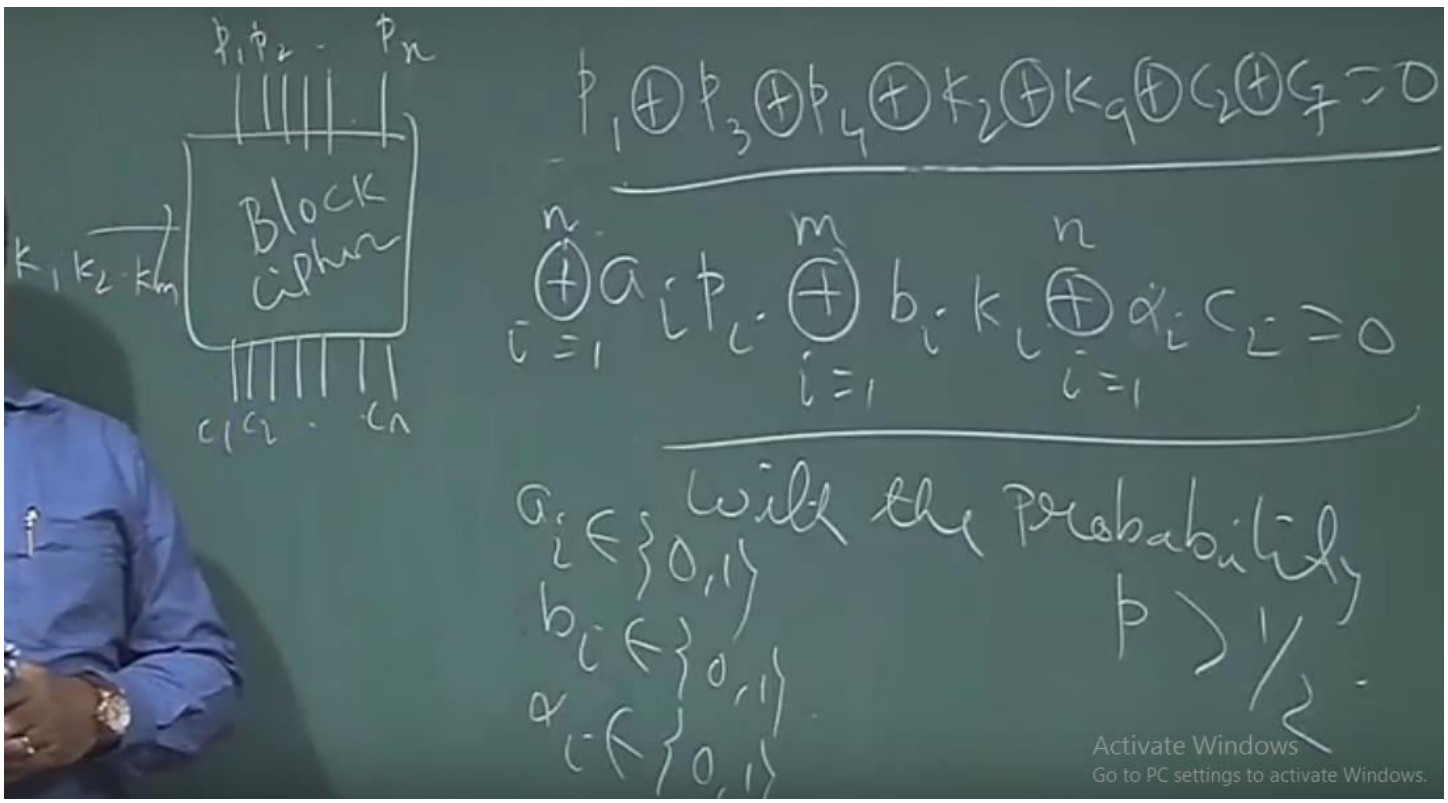
$$A[i, j, \ldots, k] = A[i] \oplus A[j] \oplus \ldots \oplus A[k]$$

- The objective of linear cryptanalysis is to find an effective *linear* equation of the form

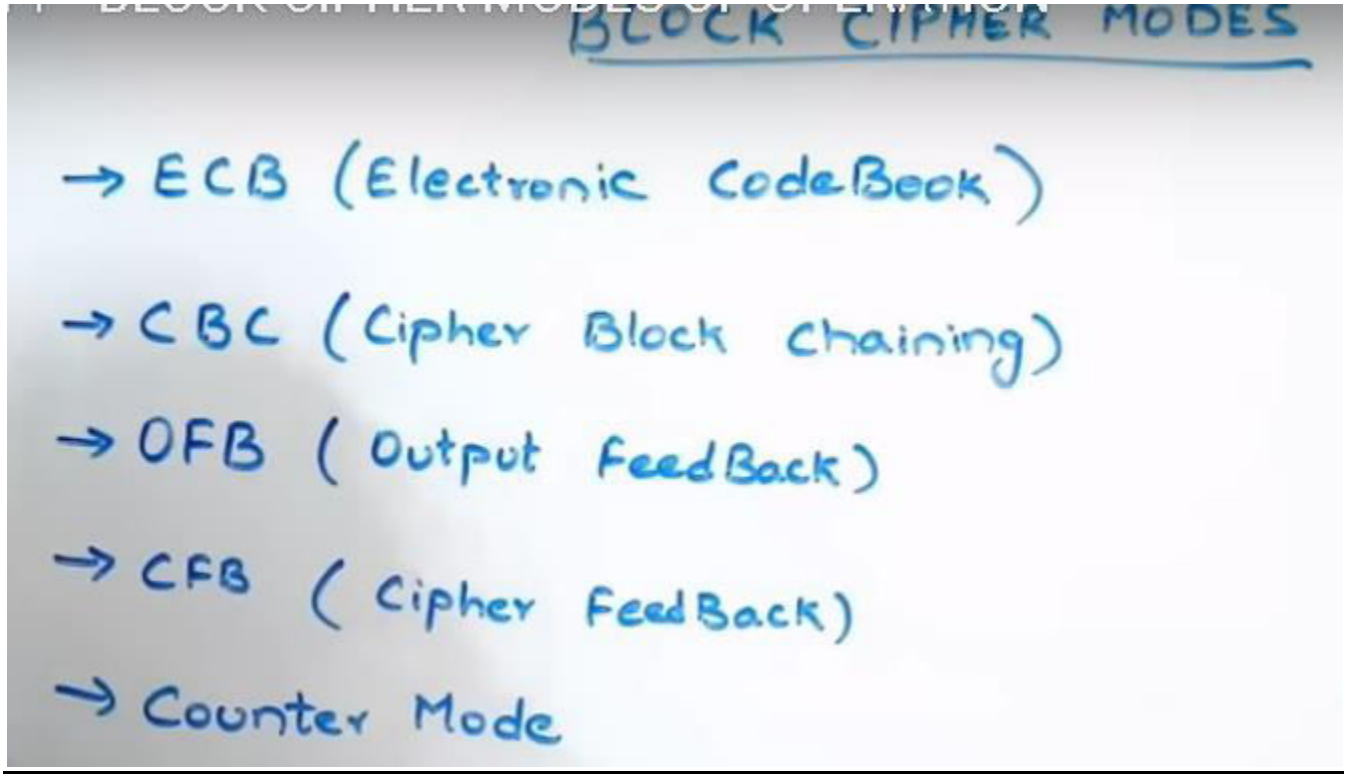
$$P[\alpha_1, \alpha_2, \ldots, \alpha_a] \oplus C[\beta_1, \beta_2, \ldots, \beta_b] = K[\gamma_1, \gamma_2, \ldots, \gamma_c] \qquad (1)$$

that holds with probability $p \neq 0.5$.

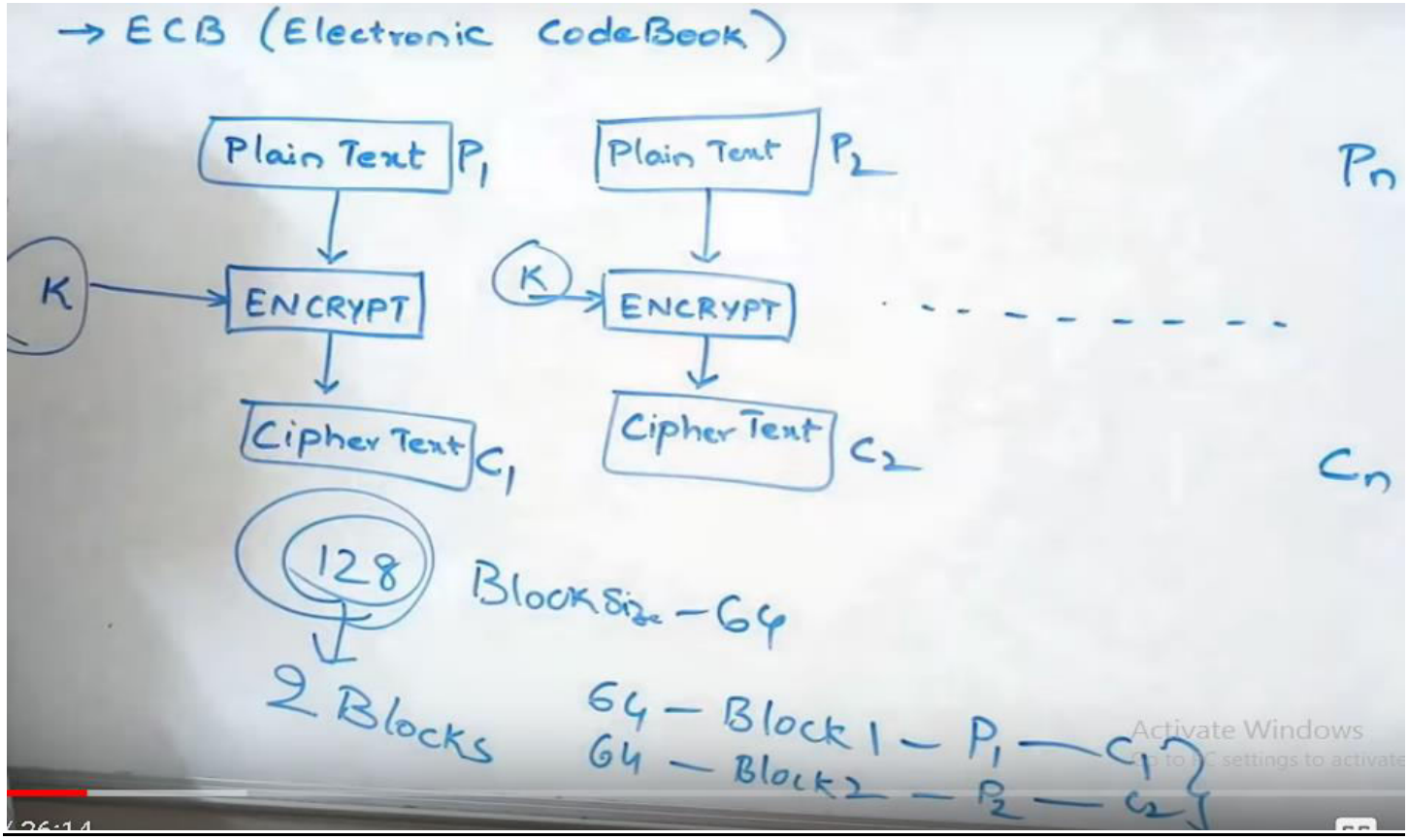
[we want to see some linear relation between pt, ct and key space]



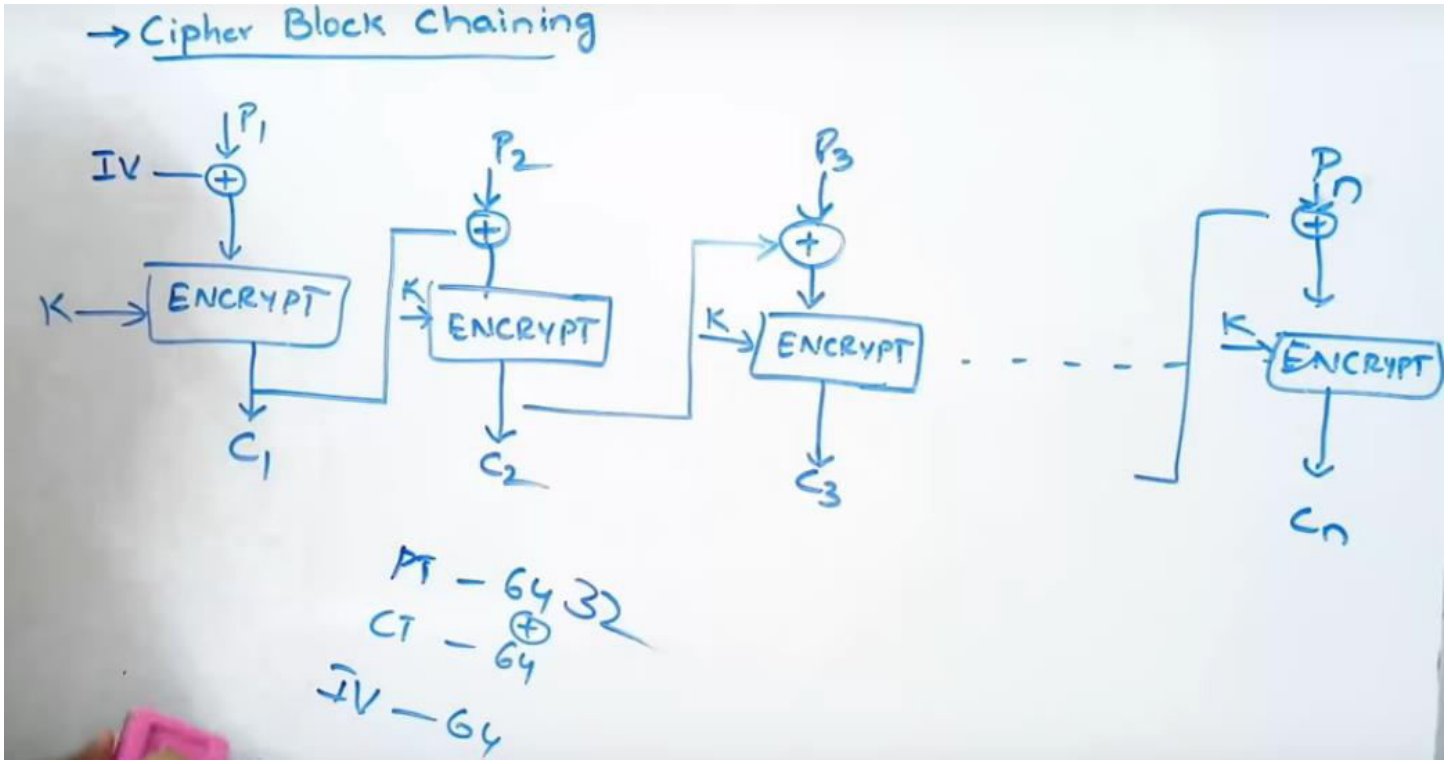
#16 NETWORK SECURITY - BLOCK CIPHER MODES OF OPERATION



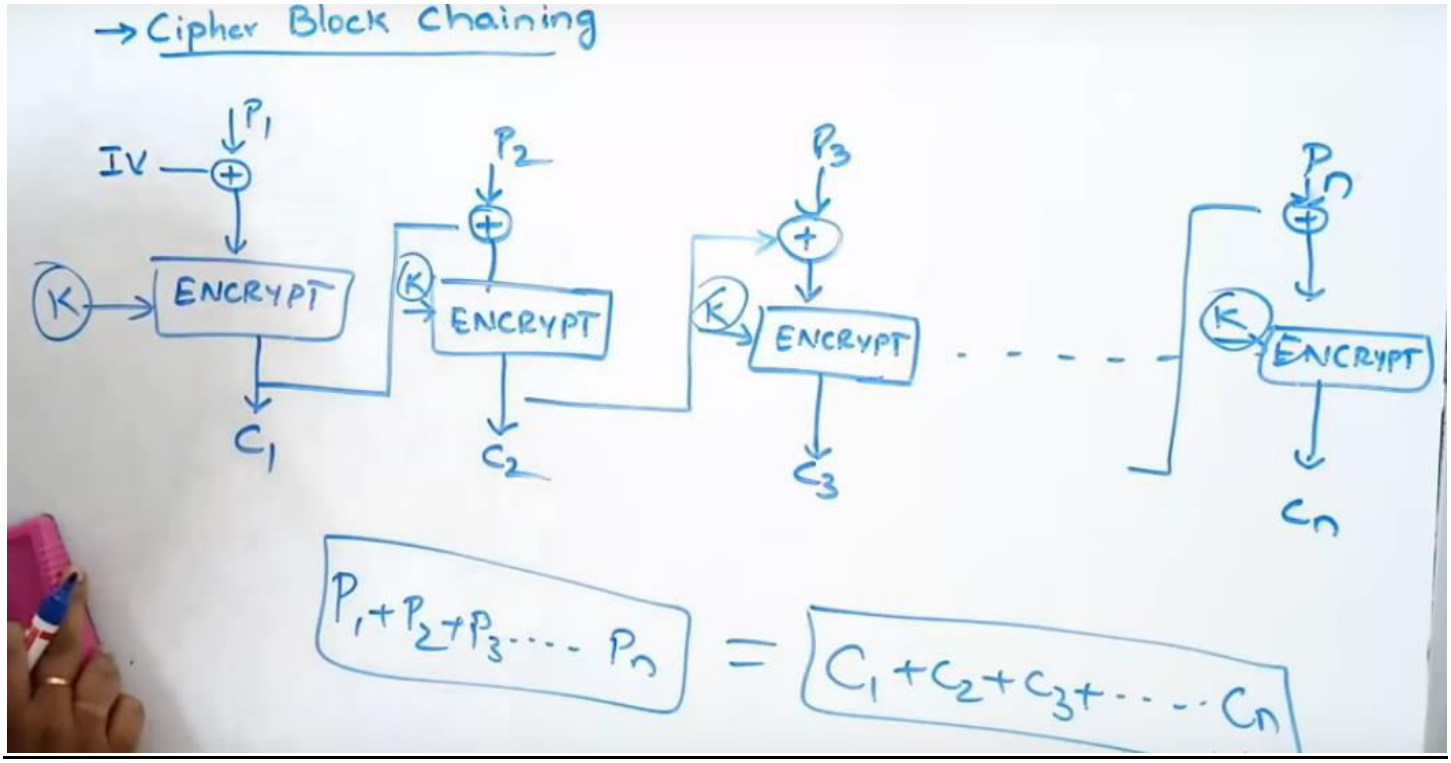
[block size depends on enc algo]



[same key used in all blocks]



[initialization vector (with same size as pt and ct) is xored with 1st pt block]

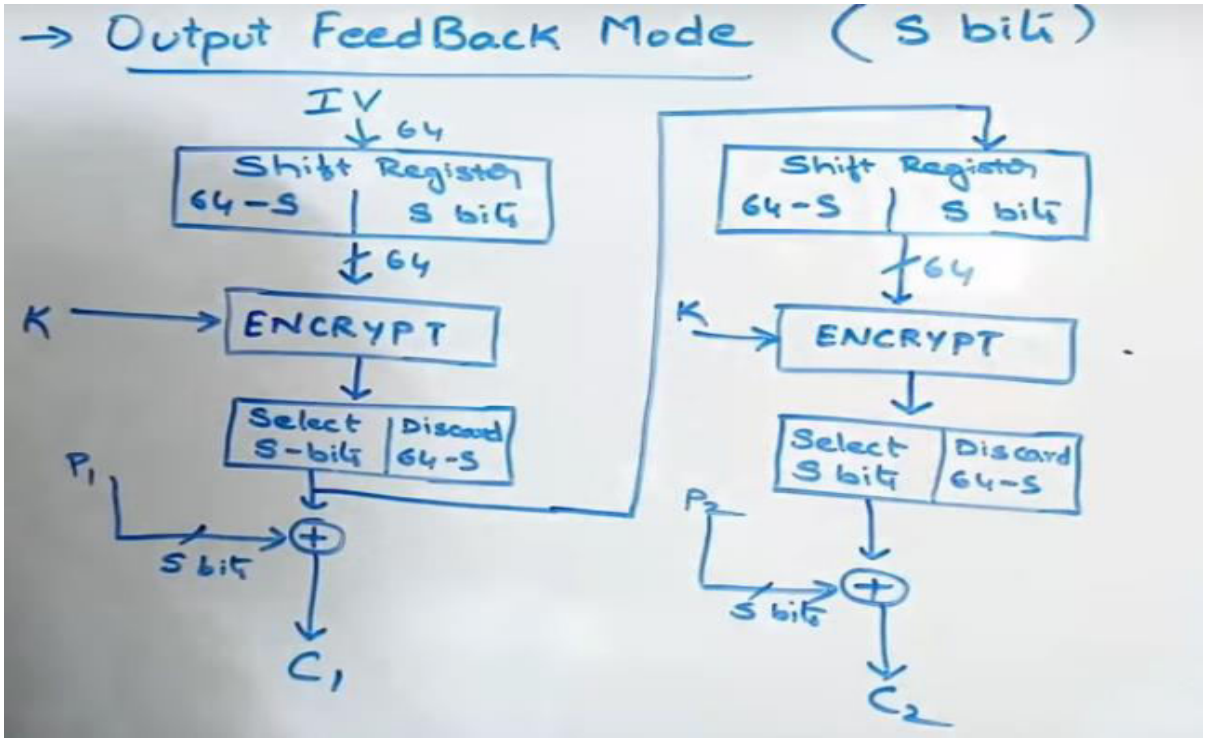


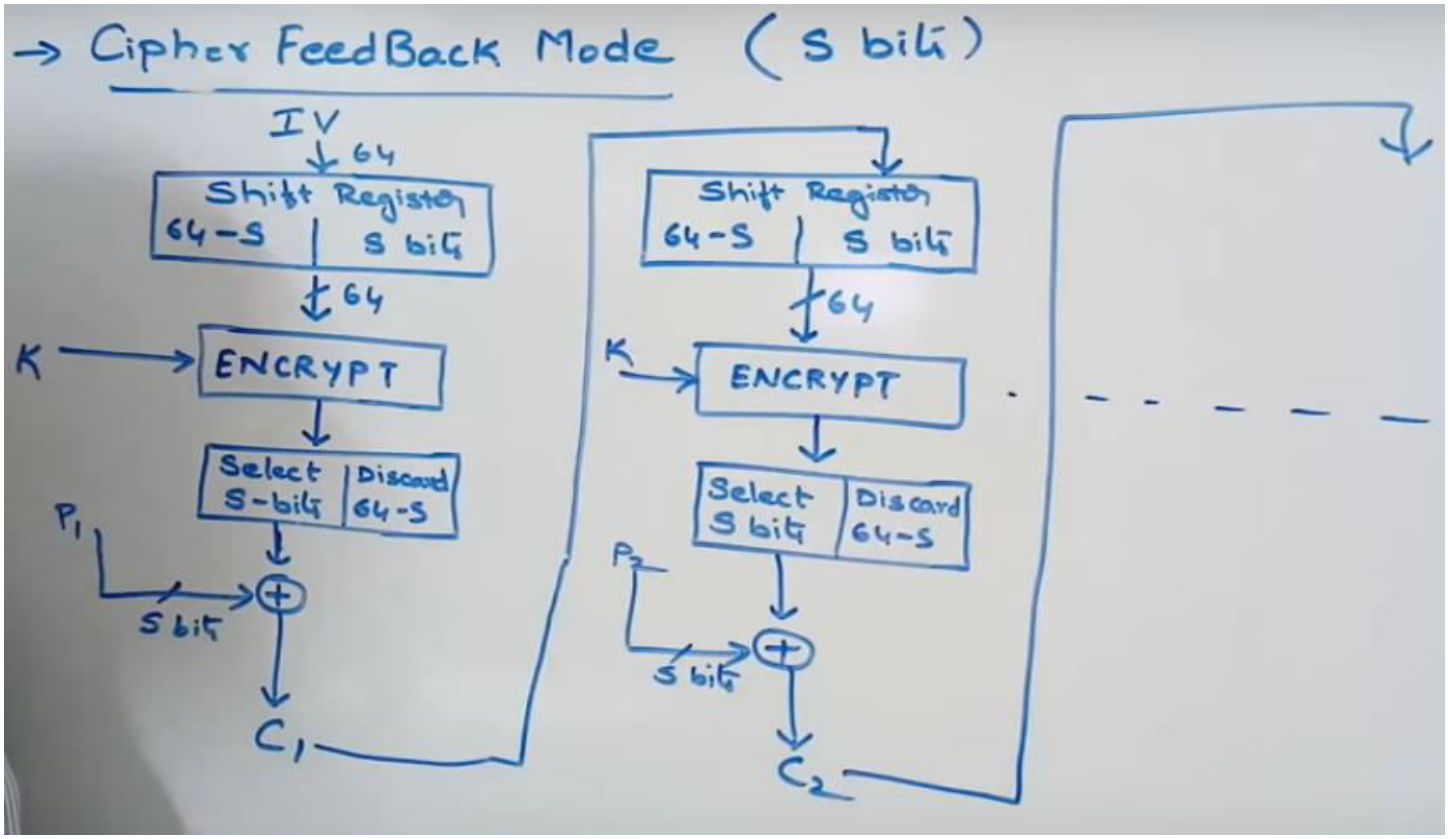
[same key, add pt/ct]

I. Process s bits initially [s = pt size] (default s = 8)

II. Selected s bits used for feedback to next shift reg . (IV for 1st pt block)

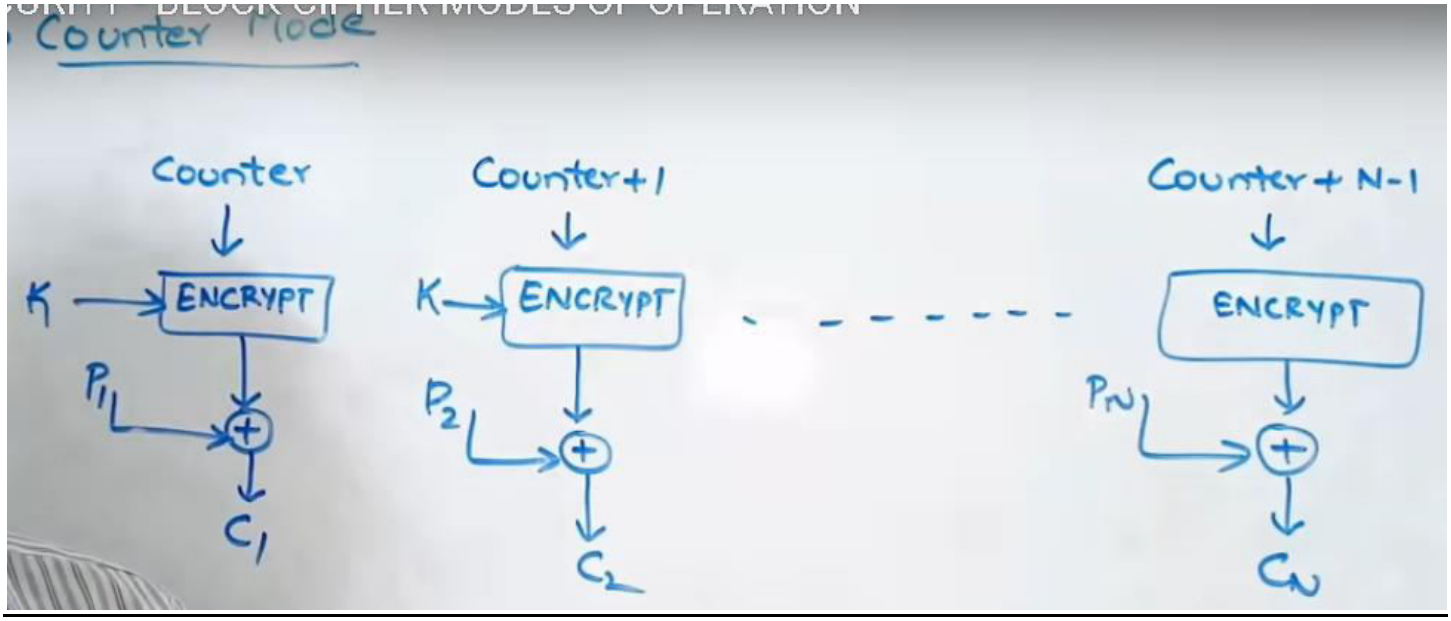
III. In last 3 modes same enc algo is used for dec as well.





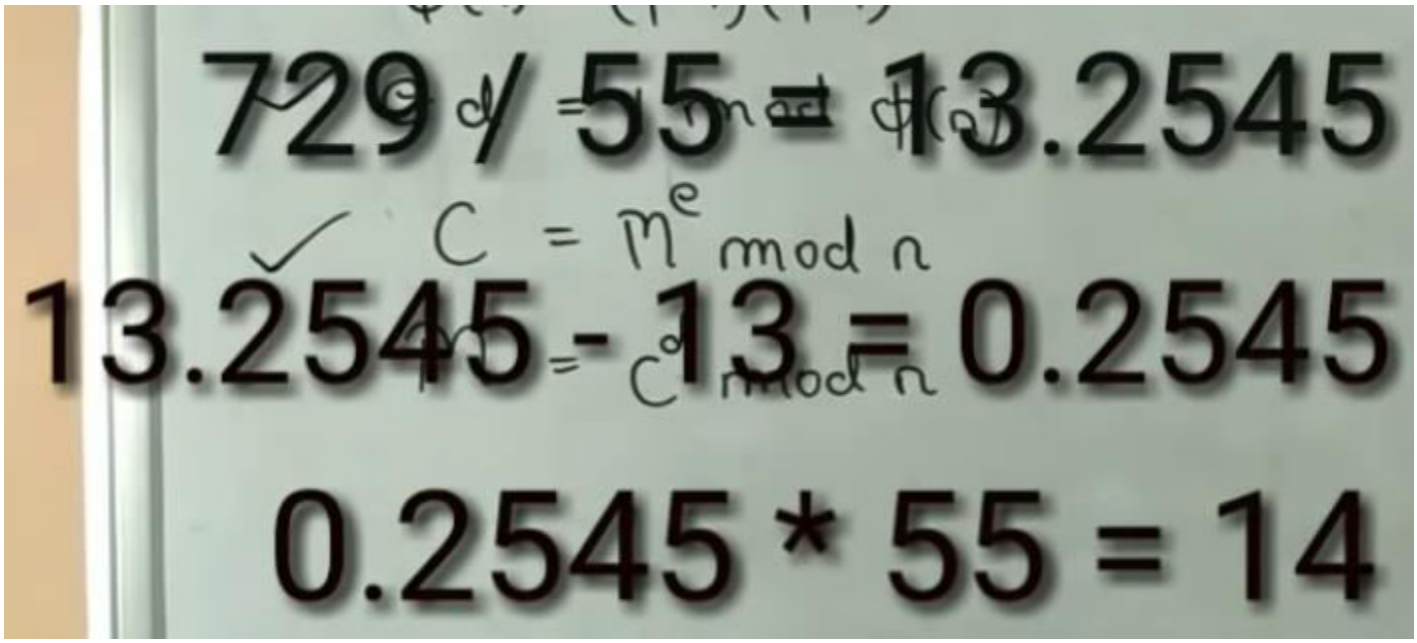
[HERE cipher used for shift reg]

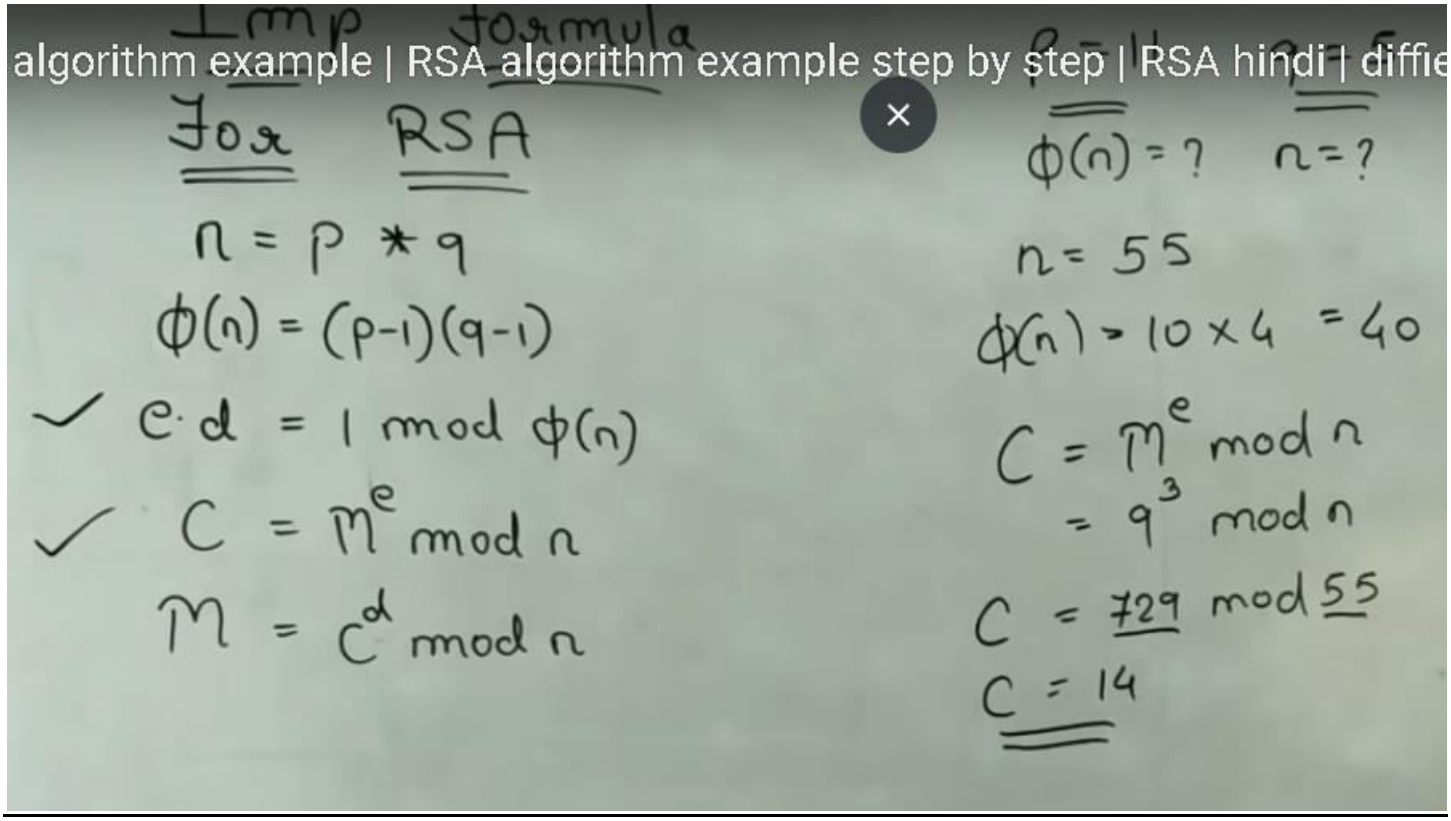
Length of counter and pt are equal

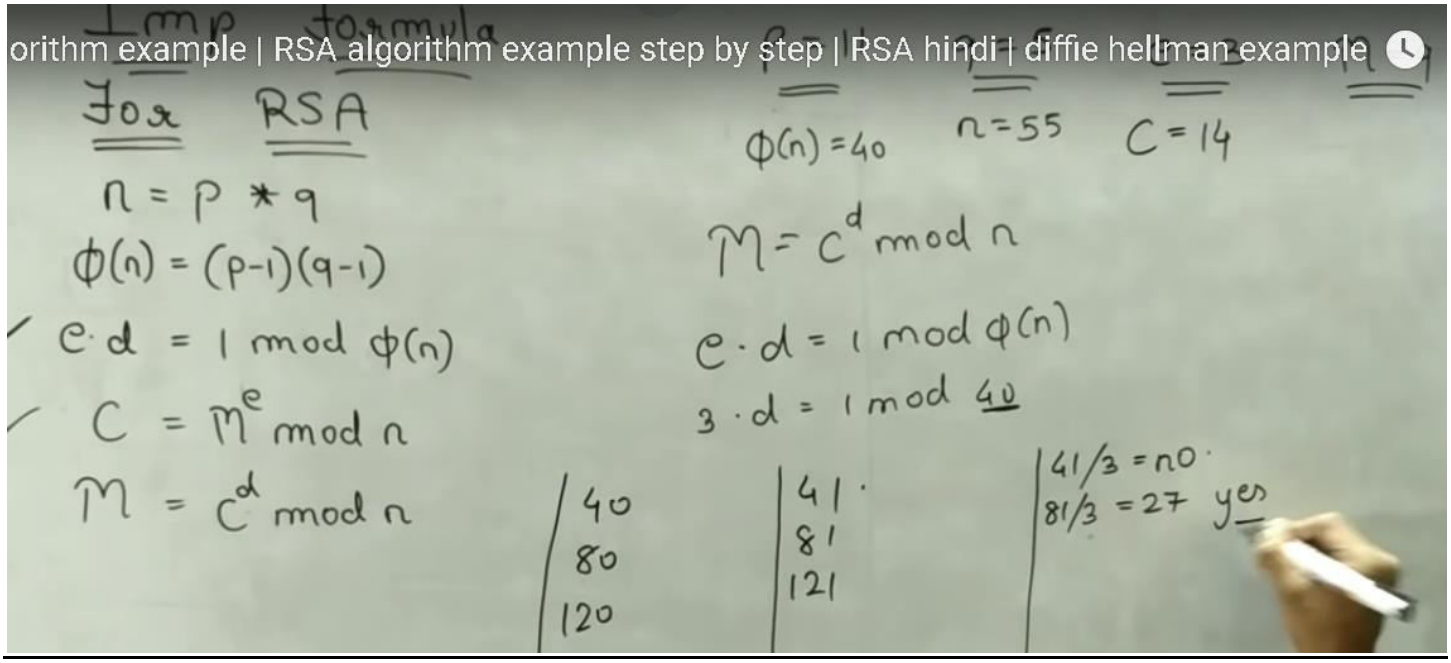


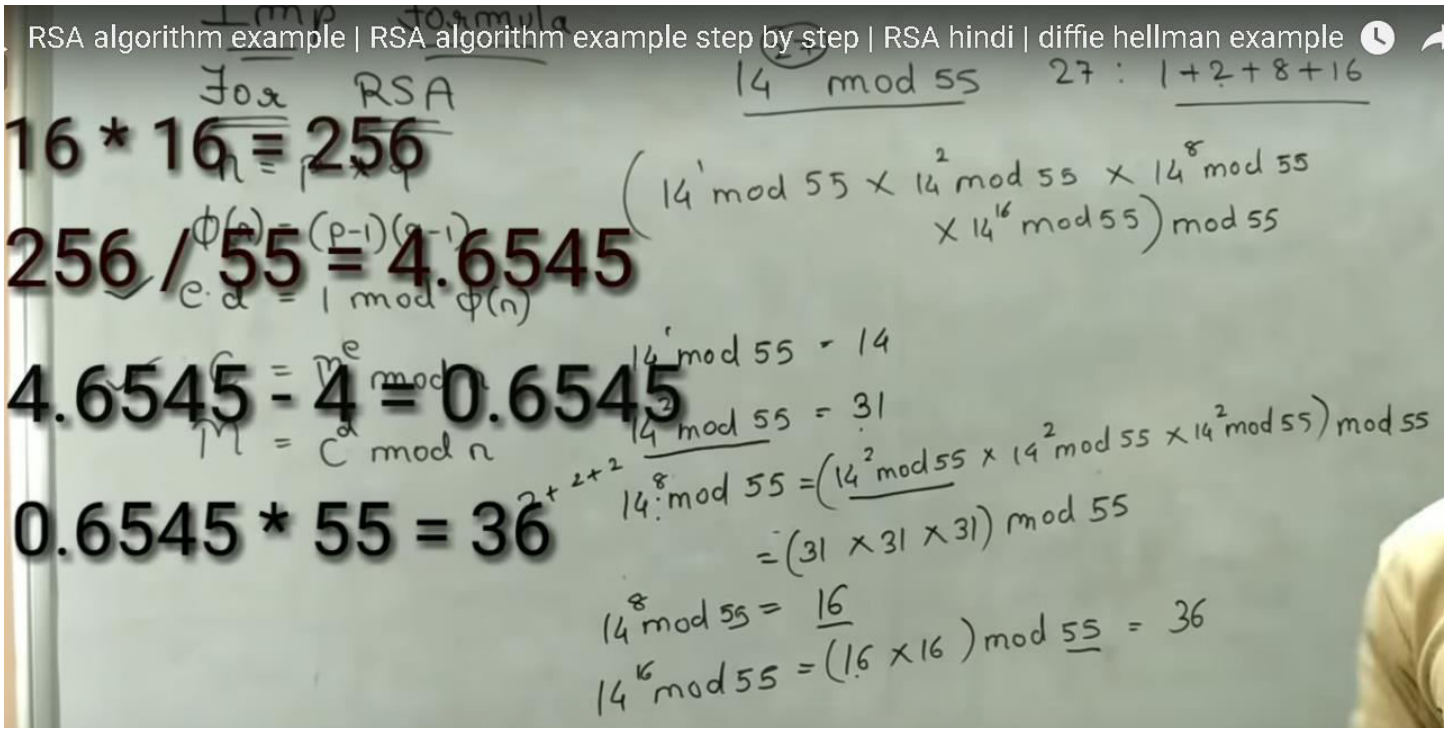
#17 RSA algorithm example step by step | RSA hindi | diffie hellman example

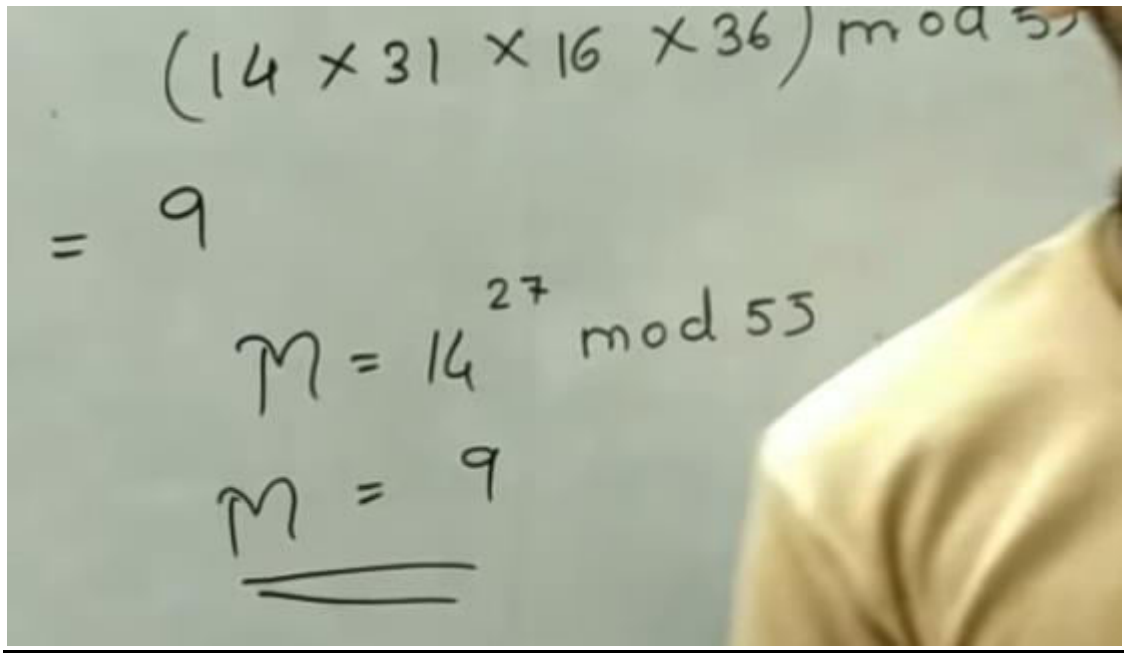
[m = pt, c = ct, e = enc, d = dec]



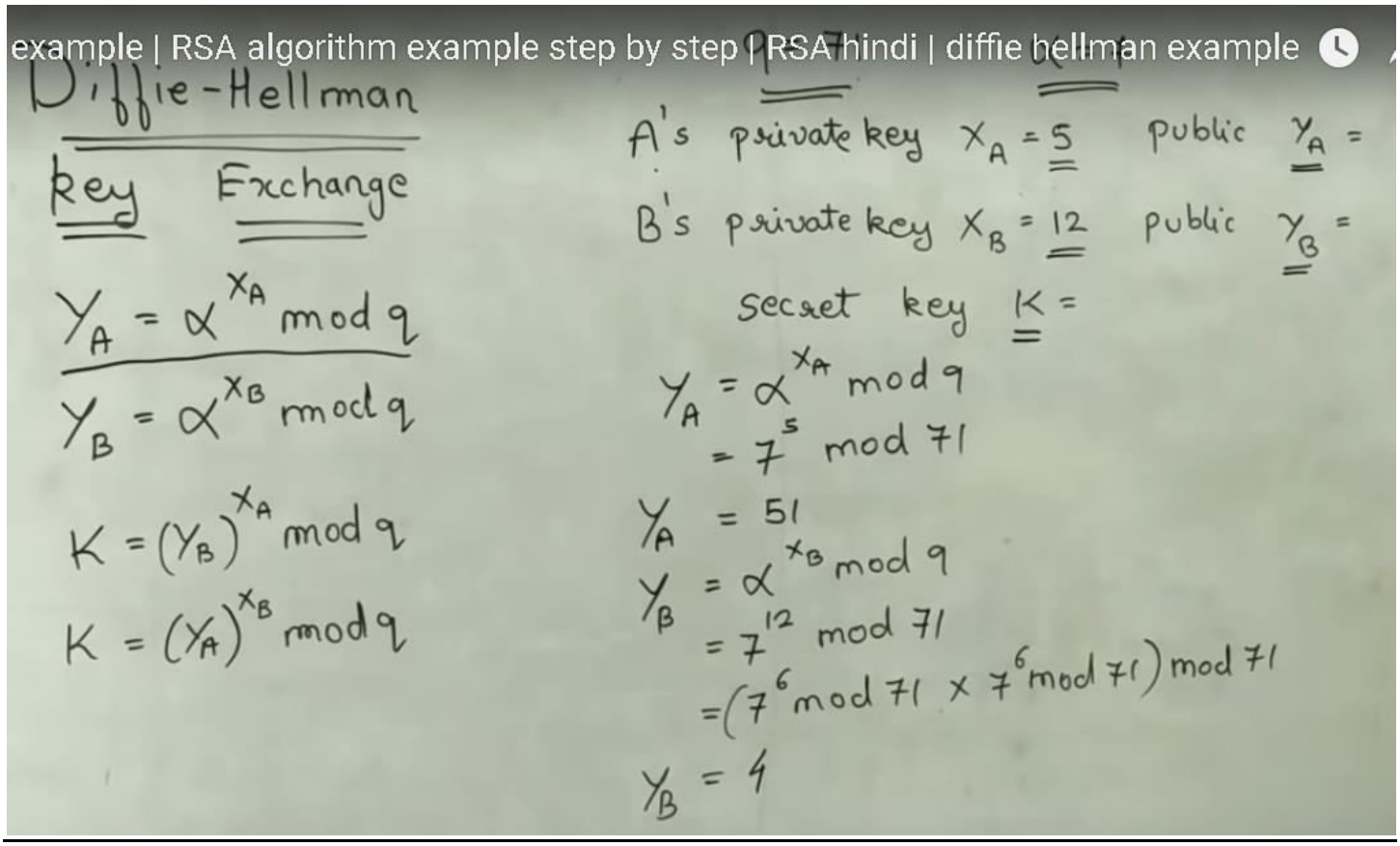

algorithm example | RSA algorithm example step by step | RSA hindi | diffie
For RSA
n = p * q
ϕ(n) = (p-1)(q-1)
e·d = 1 mod ϕ(n)
C = M^e mod n
M = C^d mod n
ϕ(n) = ?
n = ?
n = 55
ϕ(n) = 10 × 4 = 40
C = M^e mod n
= 9^3 mod n
C = 729 mod 55
C = 14

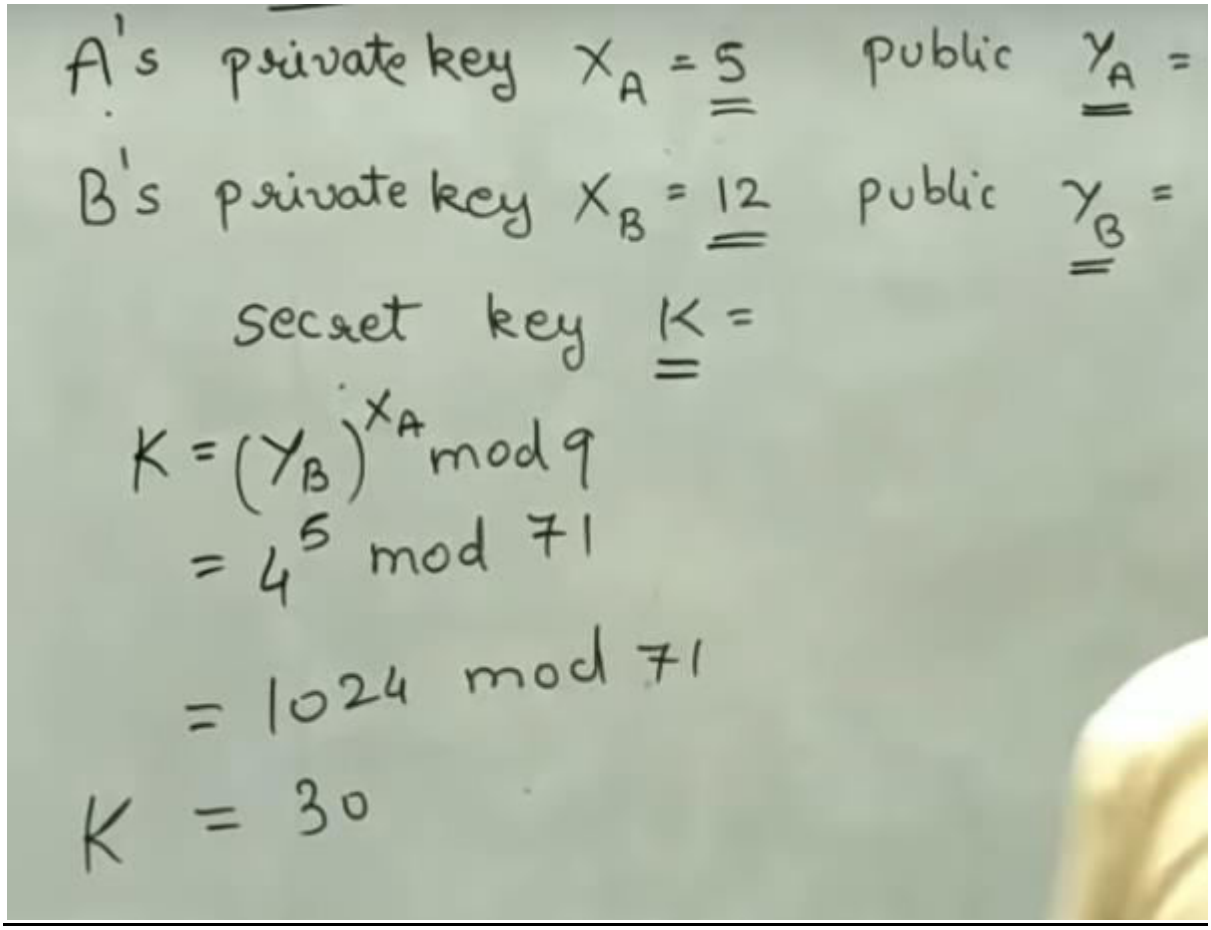

orithm example | RSA algorithm example step by step | RSA hindi | diffie hellman example
For RSA
n = p * q
ϕ(n) = (p-1)(q-1)
e·d = 1 mod ϕ(n)
C = M^e mod n
M = C^d mod n
ϕ(n) = 40
n = 55
C = 14
M = C^d mod n
e·d = 1 mod ϕ(n)
3·d = 1 mod 40
40
80
120
41
81
121
41/3 = no
81/3 = 27 yes


RSA algorithm example | RSA algorithm example step by step | RSA hindi | diffie hellman example
16 * 16 = 256
256 / 55 = 4.6545
4.6545 - 4 = 0.6545
0.6545 * 55 = 36
14 mod 55
27 : 1+2+8+16
(14^1 mod 55 × 14^2 mod 55 × 14^8 mod 55 × 14^16 mod 55) mod 55
14^1 mod 55 = 14
14^2 mod 55 = 31
14^8 mod 55 = (14^2 mod 55 × 14^2 mod 55 × 14^2 mod 55) mod 55
= (31 × 31 × 31) mod 55
14^8 mod 55 = 16
14^16 mod 55 = (16 × 16) mod 55 = 36

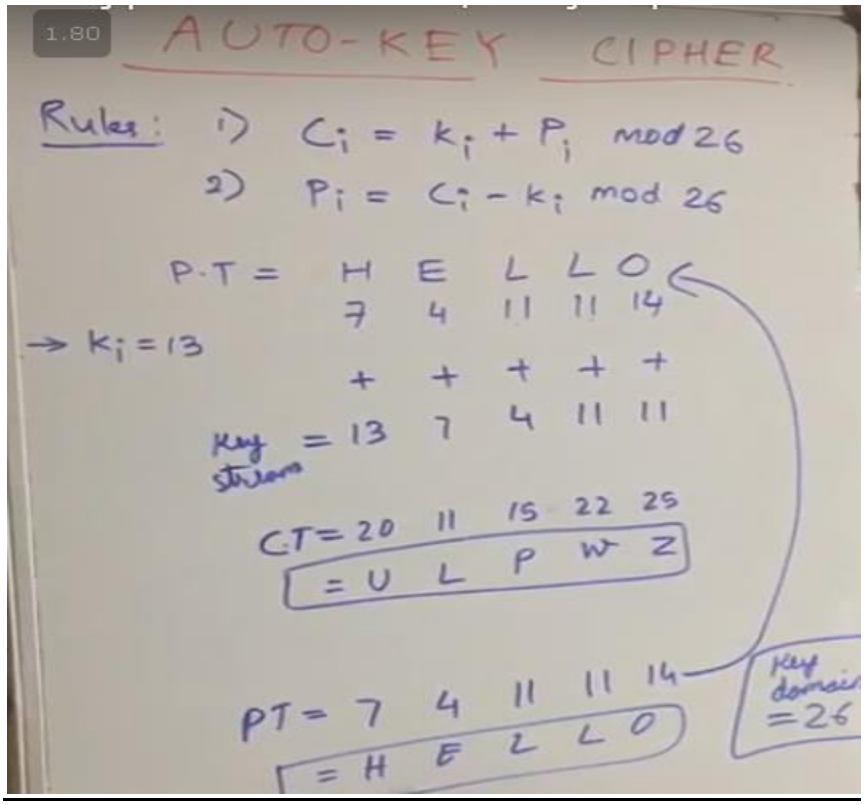

(14 × 31 × 16 × 36) mod 55
= 9
M = 14^27 mod 55
M = 9

[Diffi Hellman : alpha primitive root to find public key (Y)]

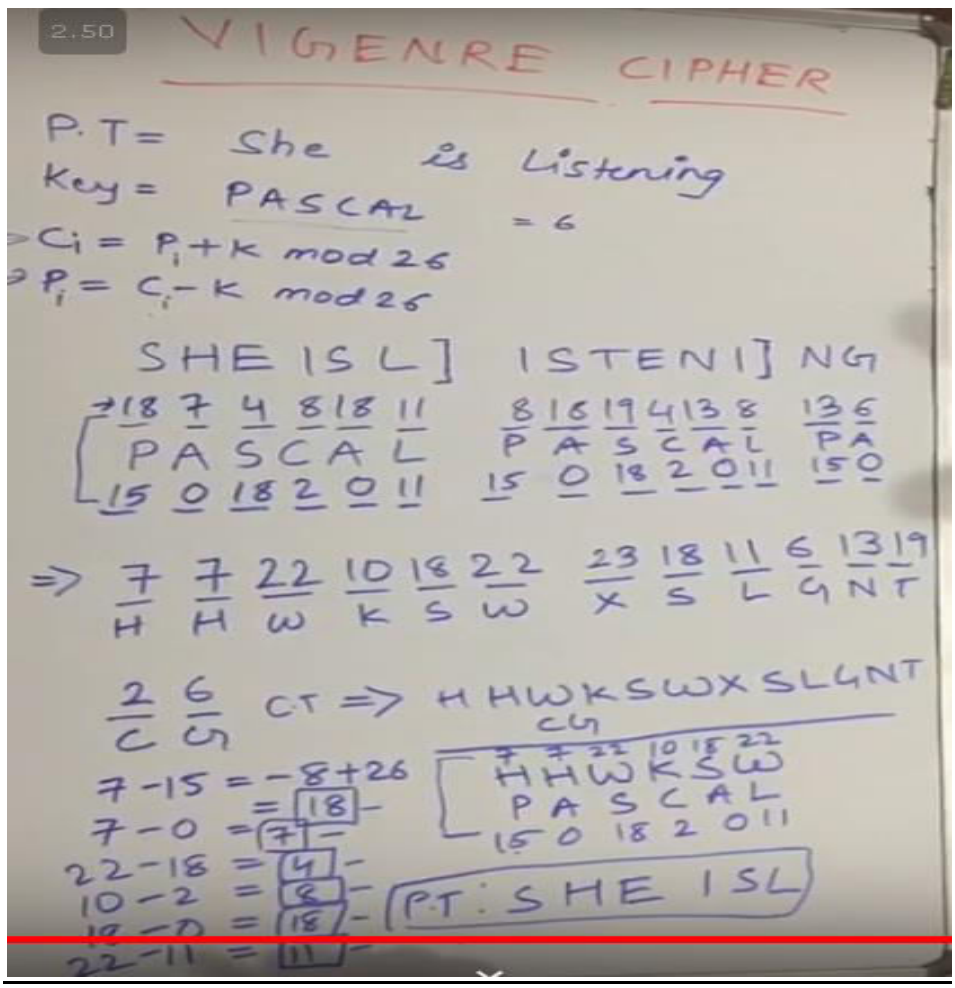




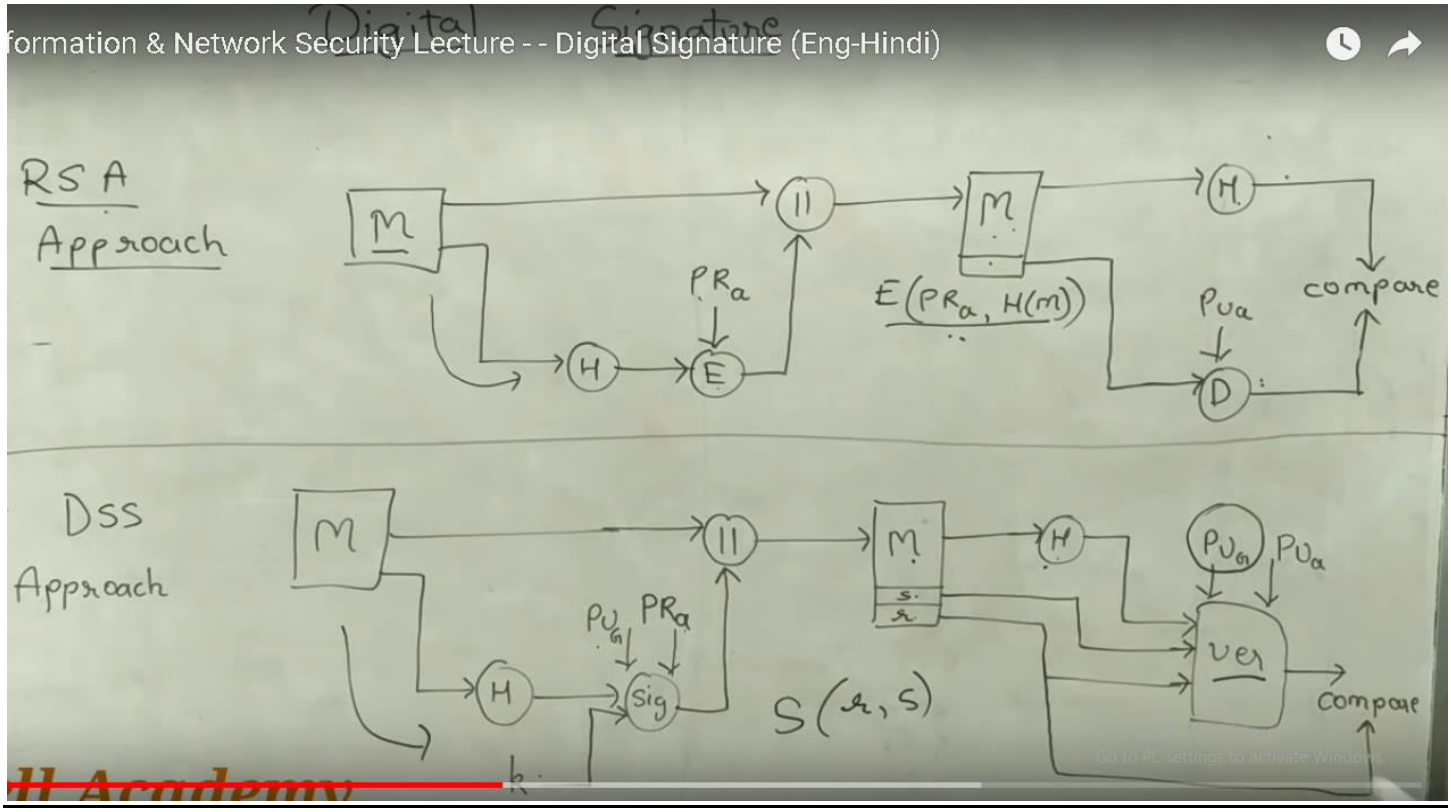
#18 Autokey cipher (polyalphabetic cipher [diff cipher for same pt char])



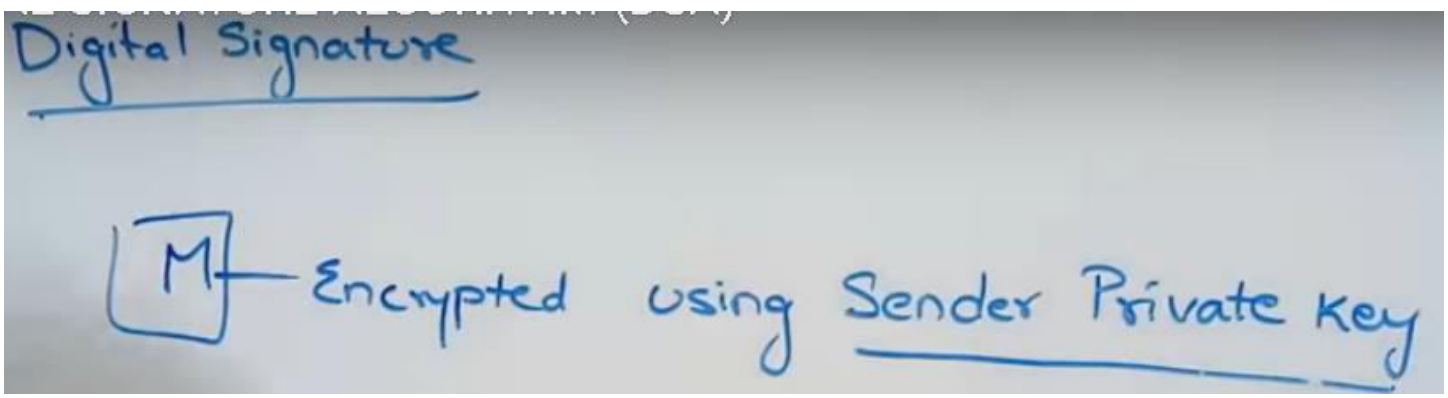
#19 vignere cipher [A = 0, Z = 25]

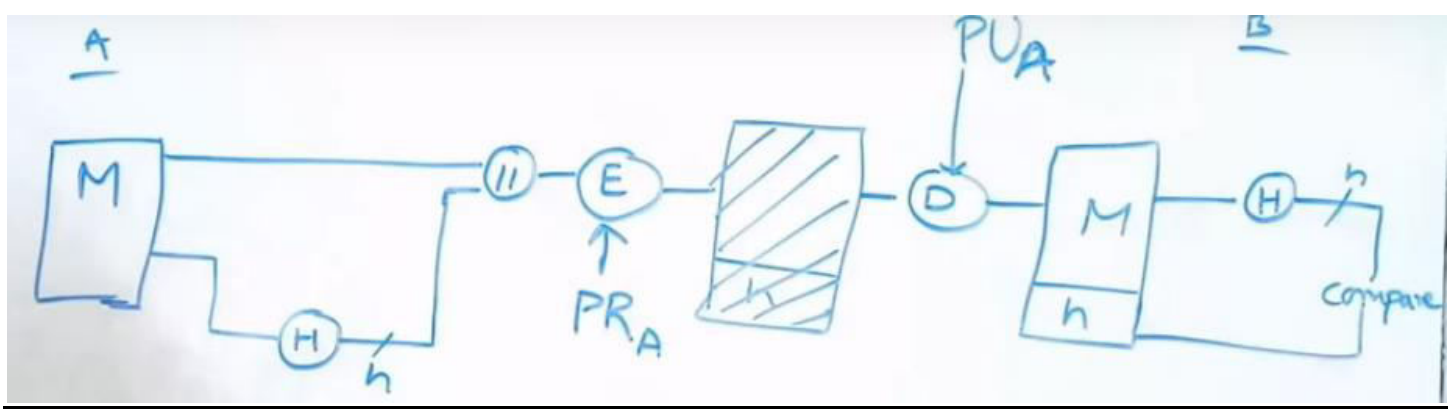


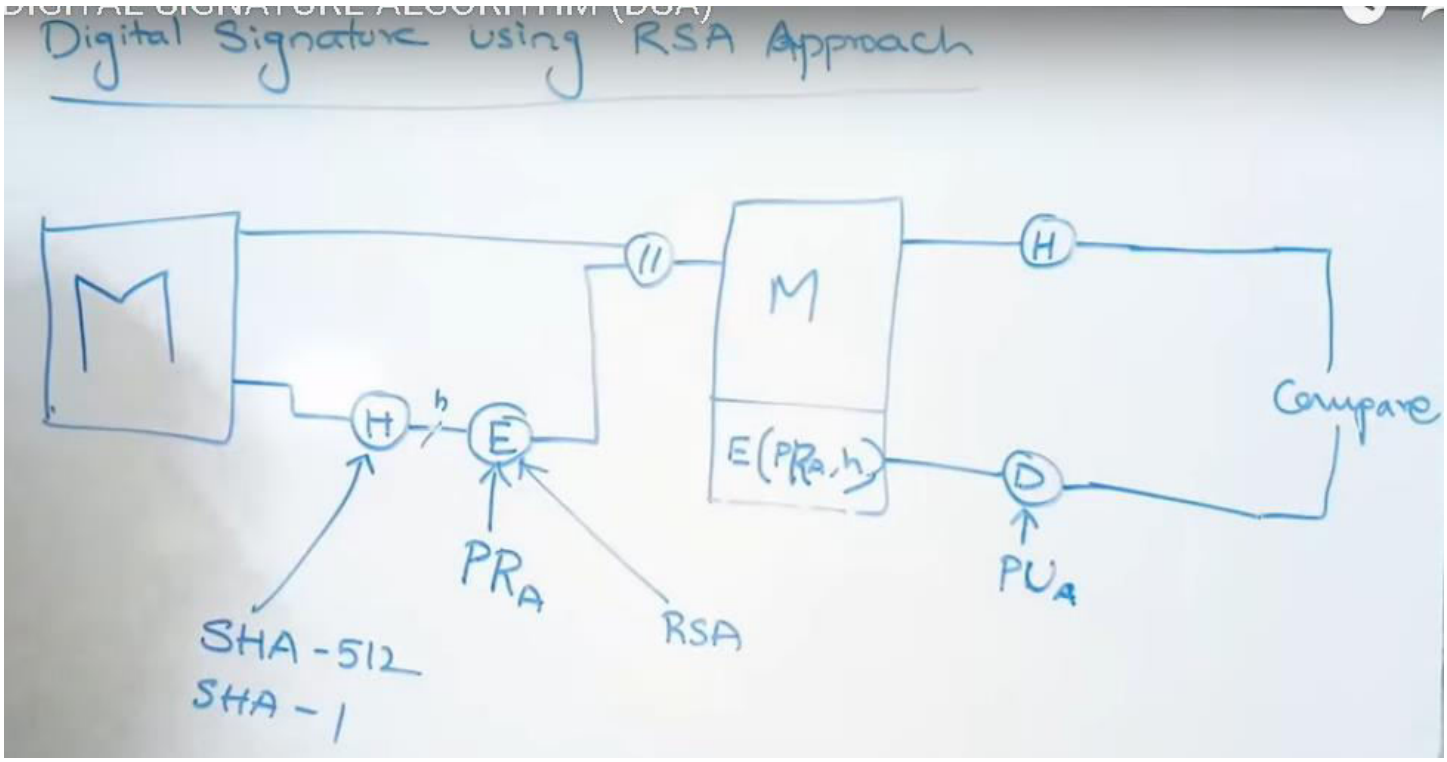
#20 Digital signature

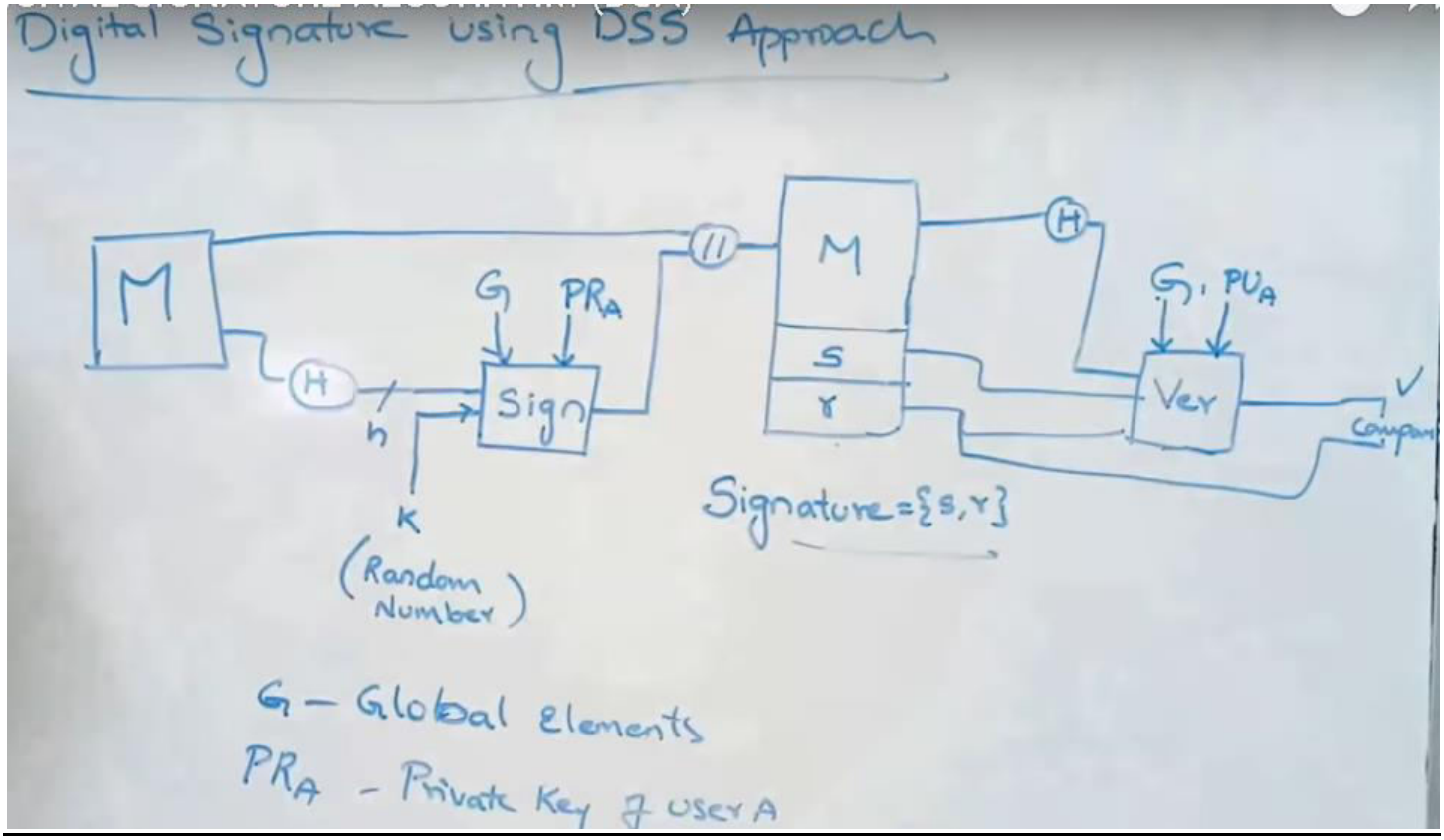


[M = pt, H = hash code, PR = private key, PU = Public key, PUG = public global key, s,r = signing components, k = arb no]

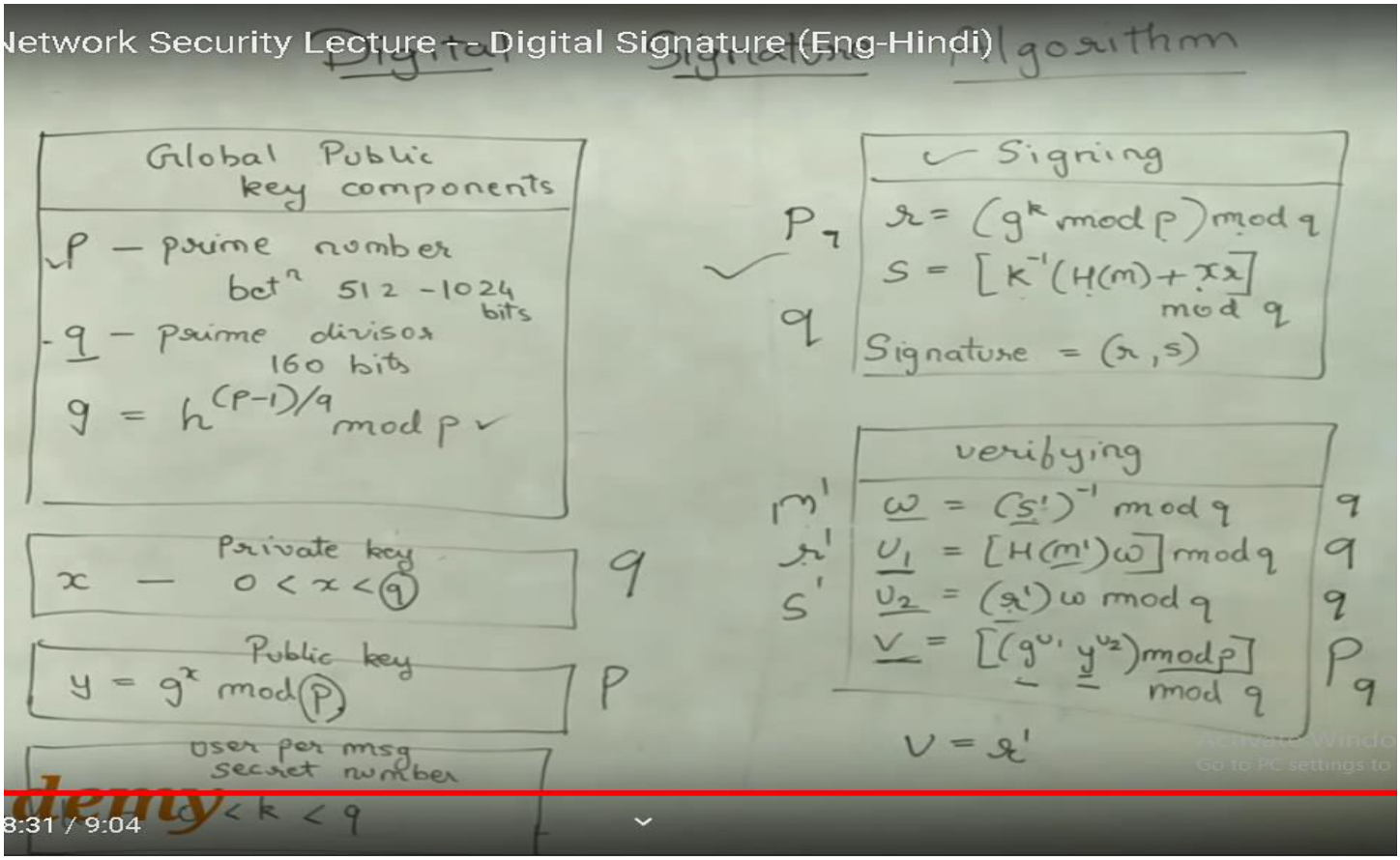


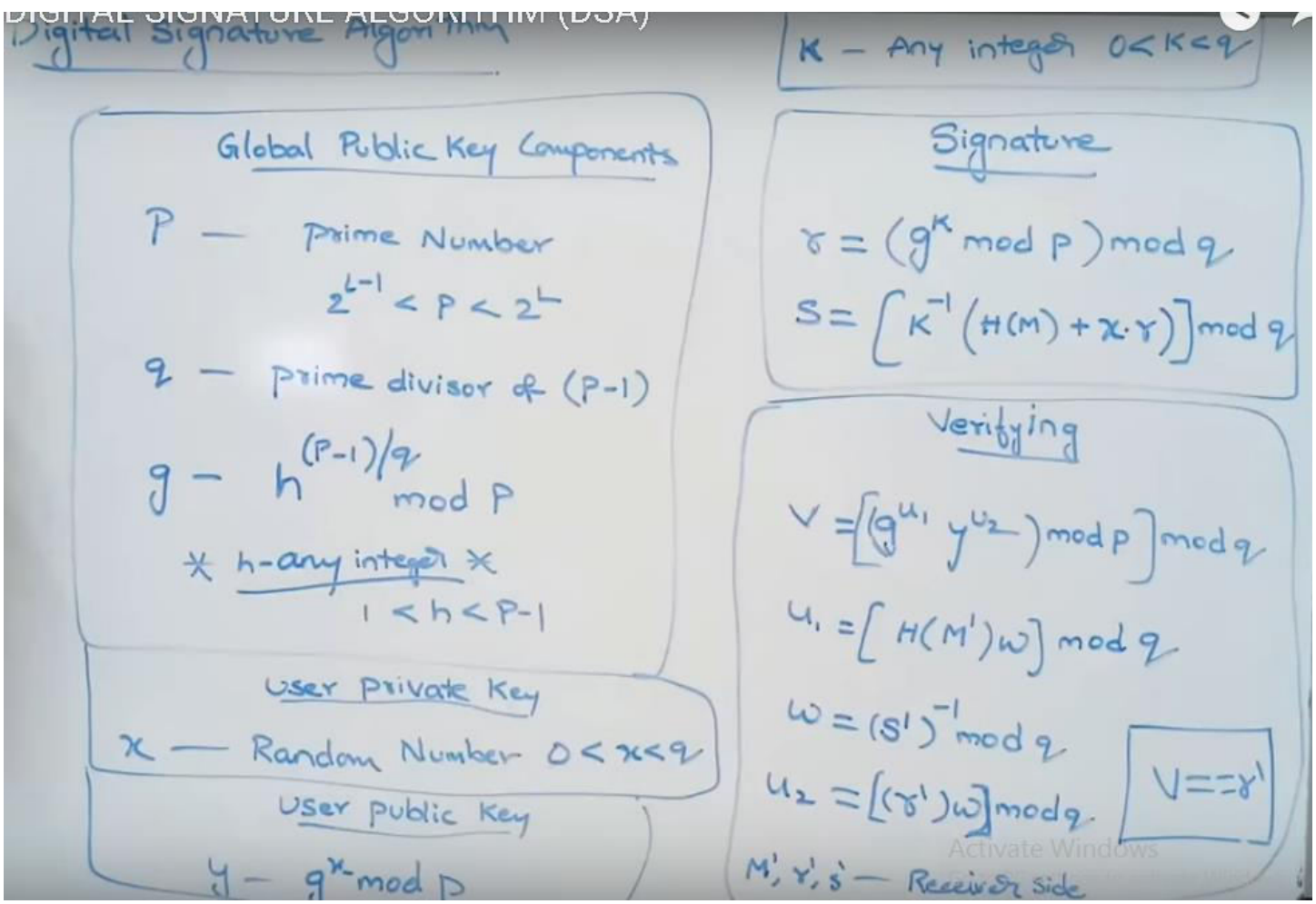




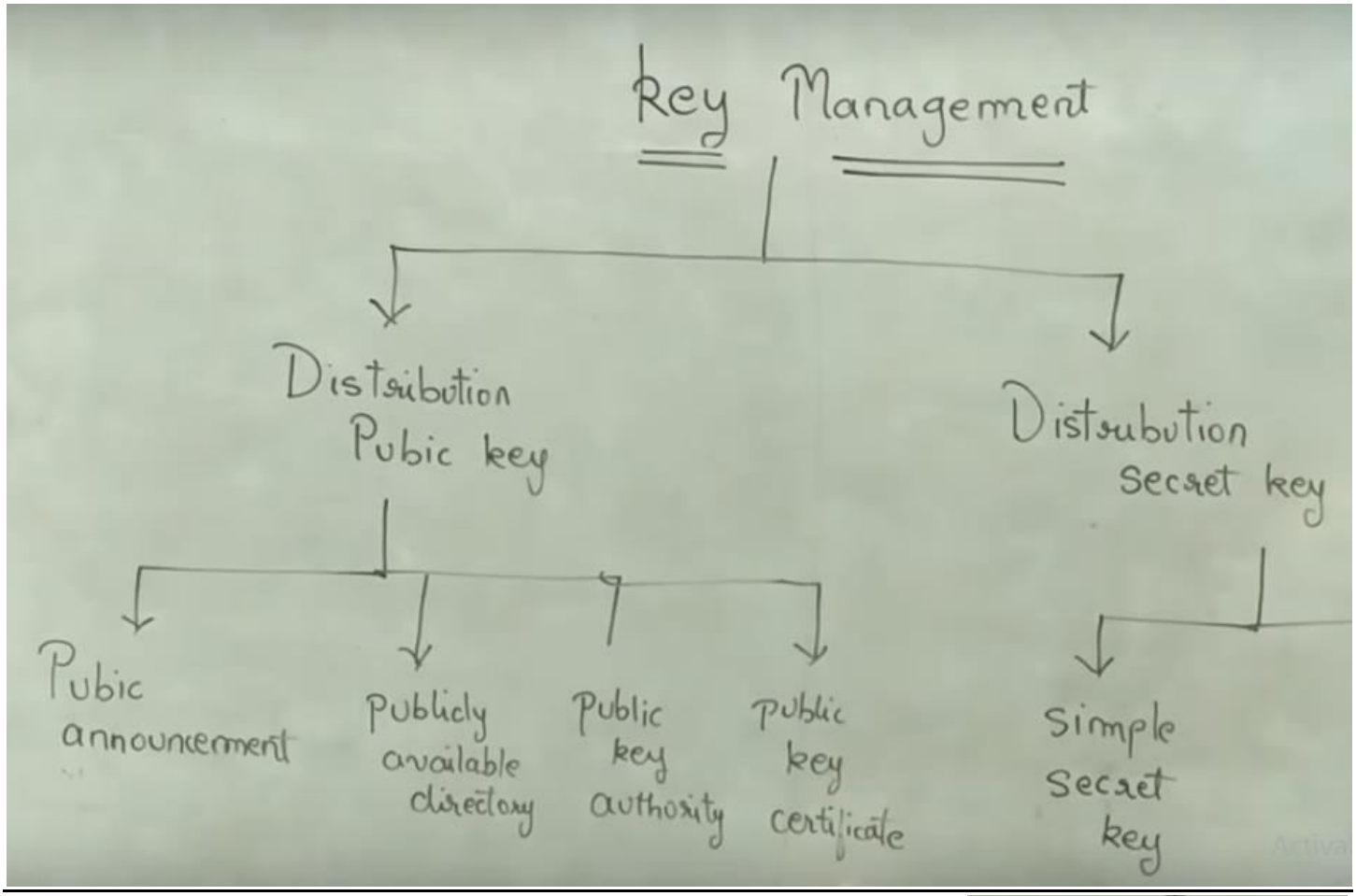


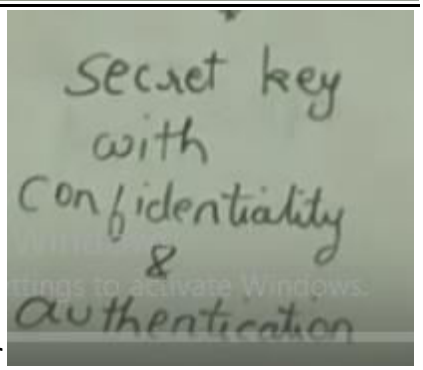
[l = length f bits , mul of 64

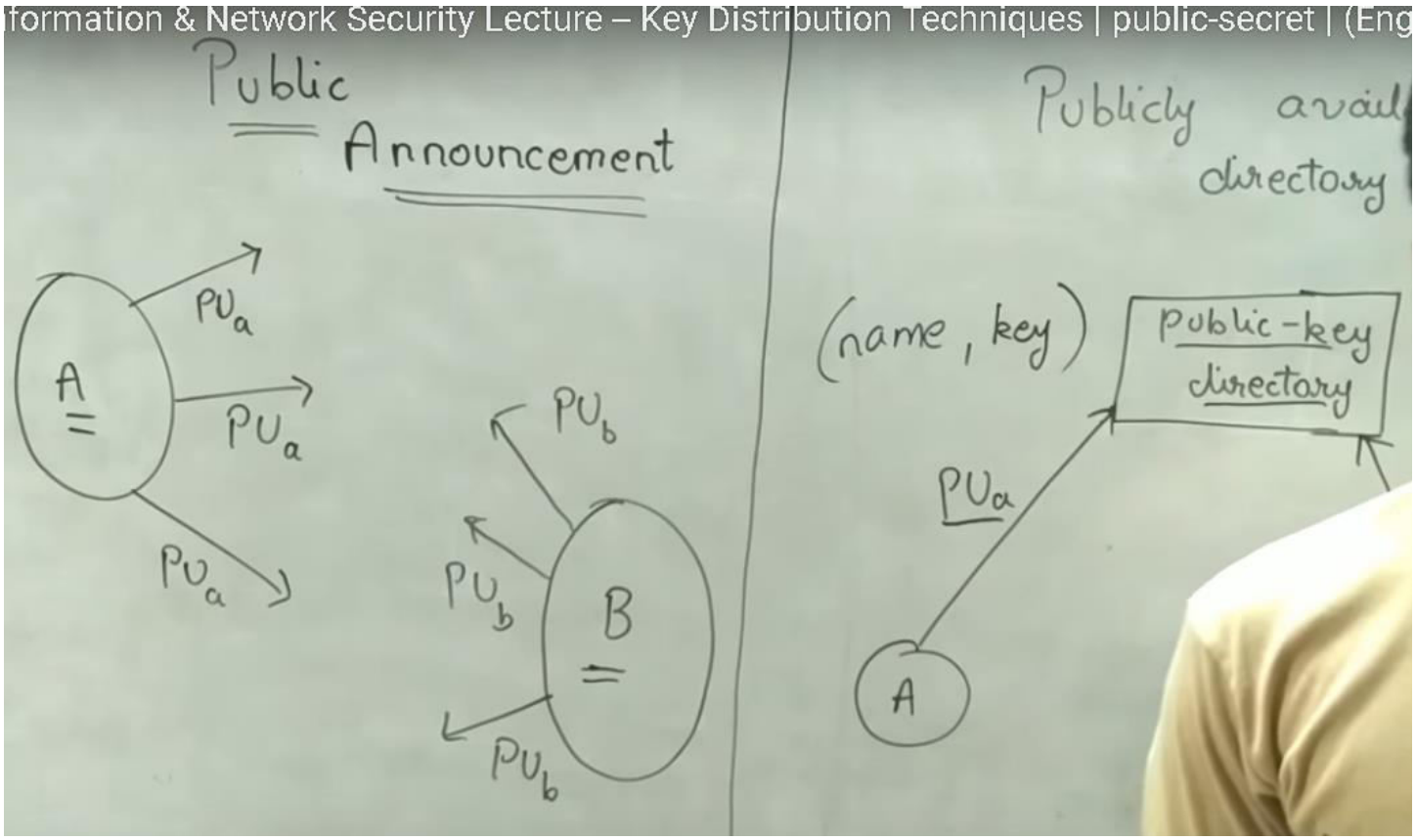

Network Security Lecture - - Digital Signature (Eng-Hindi)
Digital Signature Algorithm
Global Public key components
p — prime number bet^n 512 - 1024 bits
q — prime divisor 160 bits
g = h^((p-1)/q) mod p
Private key
x — 0 < x < q
Public key
y = g^x mod p
User per msg secret number
0 < k < q
Signing
r = (g^k mod p) mod q
s = [k^-1 (H(m) + xr)] mod q
Signature = (r, s)
verifying
w = (s')^-1 mod q
u1 = [H(m')w] mod q
u2 = (r')w mod q
v = [(g^u1 y^u2) mod p] mod q
v = r'
8:31 / 9:04

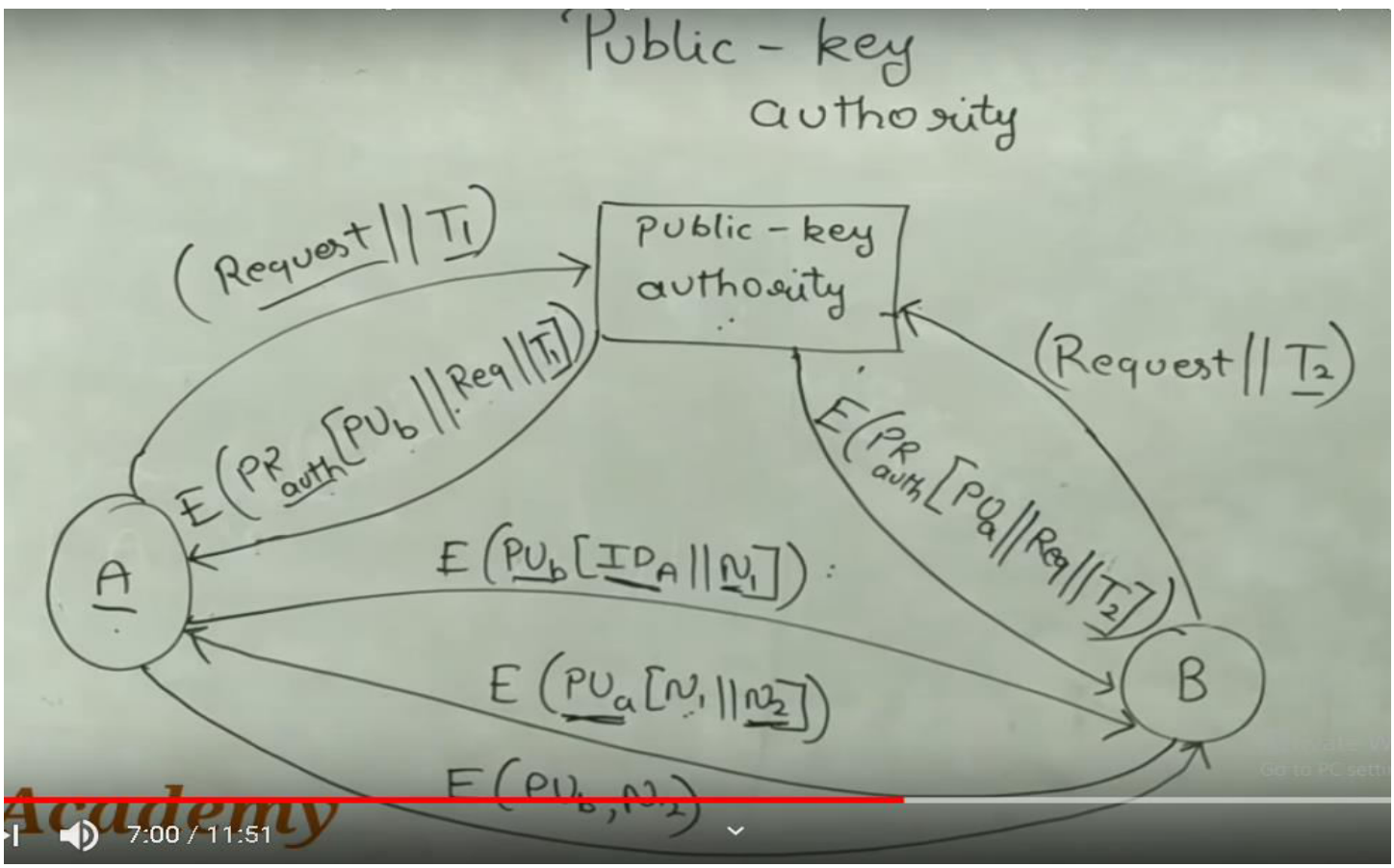

DIGITAL SIGNATURE ALGORITHM (DSA)
Digital Signature Algorithm
Global Public Key Components
P — Prime Number
2^(L-1) < P < 2^L
q — prime divisor of (P-1)
g — h^((P-1)/q) mod P
* h-any integer *
1 < h < P-1
User Private Key
x — Random Number 0 < x < q
User public Key
y — g^x mod p
K — Any integer 0 < K < q
Signature
r = (g^K mod p) mod q
S = [K^-1 (H(M) + x·r)] mod q
Verifying
v = [(g^u1 y^u2) mod p] mod q
u1 = [H(M')w] mod q
w = (s')^-1 mod q
u2 = [(r')w] mod q
v == r'
M', r', s' — Receiver side

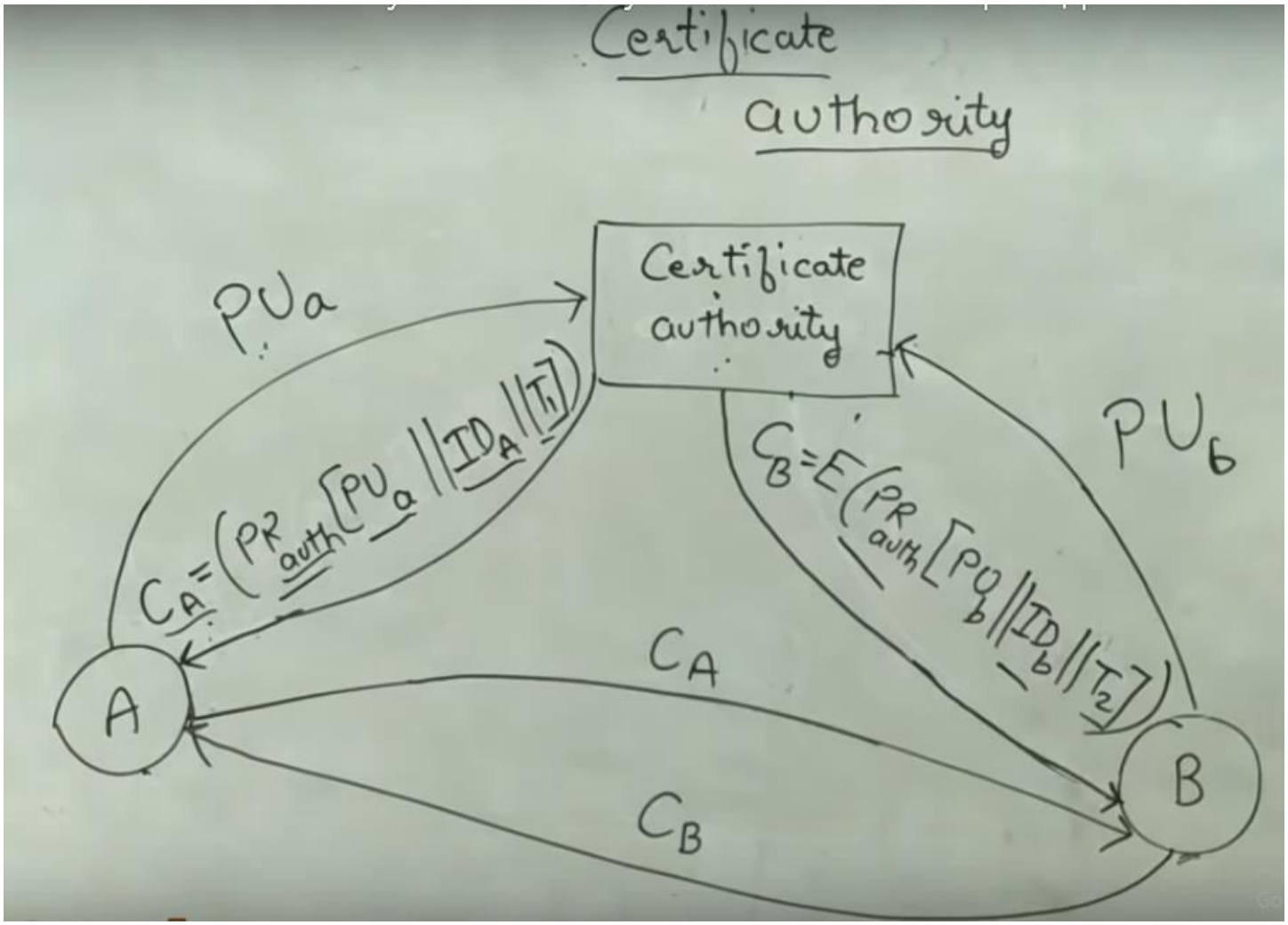
#21 key mgmt

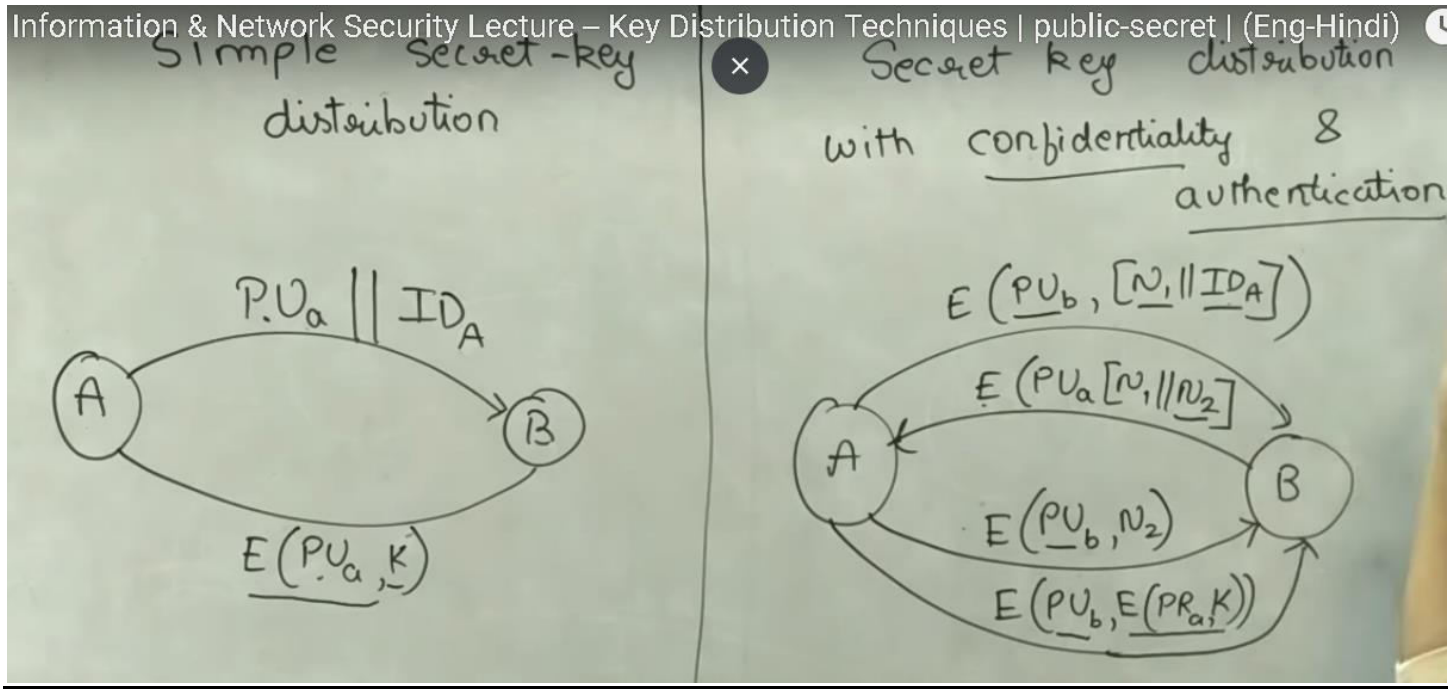

Key Management
Distribution Pubic key
Distrubution Secret key
Pubic announcement
Publicly available directory
Public key authority
public key certificate
Simple Secret key

r 
Secret key with Confidentiality & authentication


formation & Network Security Lecture – Key Distribution Techniques | public-secret | (Eng
Public Announcement
A
PUa
PUa
PUa
B
PUb
PUb
PUb
Publicly avail
directory
(name, key)
Public-key directory
PUa
A


Public-key authority
(Request || T1)
Public-key authority
(Request || T2)
E(PRauth[PUb || Req || T1])
E(PRauth[PUa || Req || T2])
A
E(PUb[IDA || N1])
E(PUa[N1 || N2])
E(PUb, N2)
B
Academy
7:00 / 11:51


Certificate authority
Certificate authority
PUa
PUb
CA = (PRauth[PUa || IDA || T1])
CB = E(PRauth[PUb || IDb || T2])
CA
CB
A
B


Information & Network Security Lecture – Key Distribution Techniques | public-secret | (Eng-Hindi)
Simple secret-key distribution
PUa || IDA
A
B
E(PUa, K)
Secret key distribution with confidentiality & authentication
E(PUb, [N1 || IDA])
E(PUa [N1 || N2])
A
B
E(PUb, N2)
E(PUb, E(PRa, K))

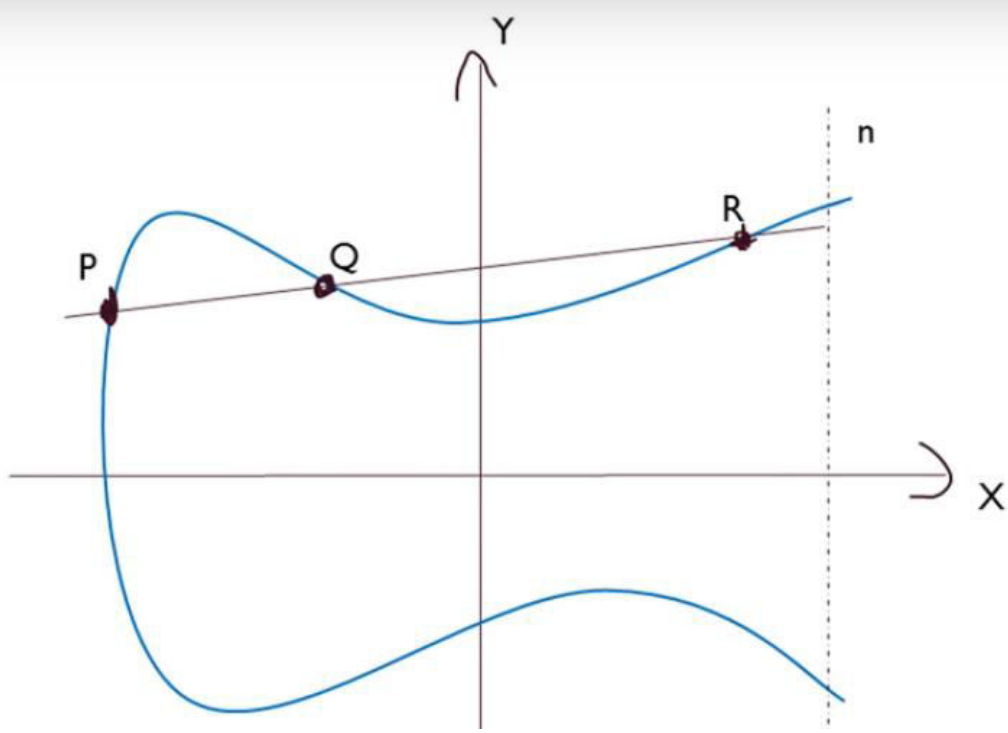
#22 ecc (used for enc-dec, key exchg, digital sign)

ECC

- Asymmetric / Public key cryptosystem.
- ECC provides equal security with smaller key size.
- ECC makes use of Elliptic curves.
- Elliptic curves are defined by mathematical functions - Cubic functions.
- eg:- $y^2 = x^3 + ax + b$

I. Symm to x axis
II. Any line touches max 3 pts
III. Limit is infinity

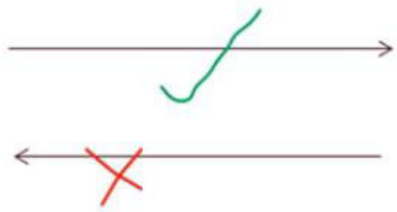
An Elliptic curve :



ECC

- Let E_p (a,b) be the elliptic curve .
- Consider equation, $Q = kP$

  where $Q, P \in E_p(a,b)$ and $k < n$
- It should be easy to find Q given k and P.
- But should be extremely difficult to find k given Q and P.



- Is a one way function - trap door function.
- Called the discrete logarithm problem.

ECC - Key Exchange

Global Public elements.

$E_q(a,b)$: Elliptic curve with parameters a, b & q
q is a prime or integer of the form 2^m

G : Point on elliptic curve whose order is large value n.

Alice key Generation.

Select private key n_A ; $n_A < n$

Calculate public key P_A ; $P_A = n_A \times G$

Bob key Generation.

Select private key n_B ; $n_B < n$

Calculate public key P_B ; $P_B = n_B \times G$

Secret key calculation by Alice

$K = n_A \times P_B$

Secret key calculation by Bob

$K = n_B \times P_A$

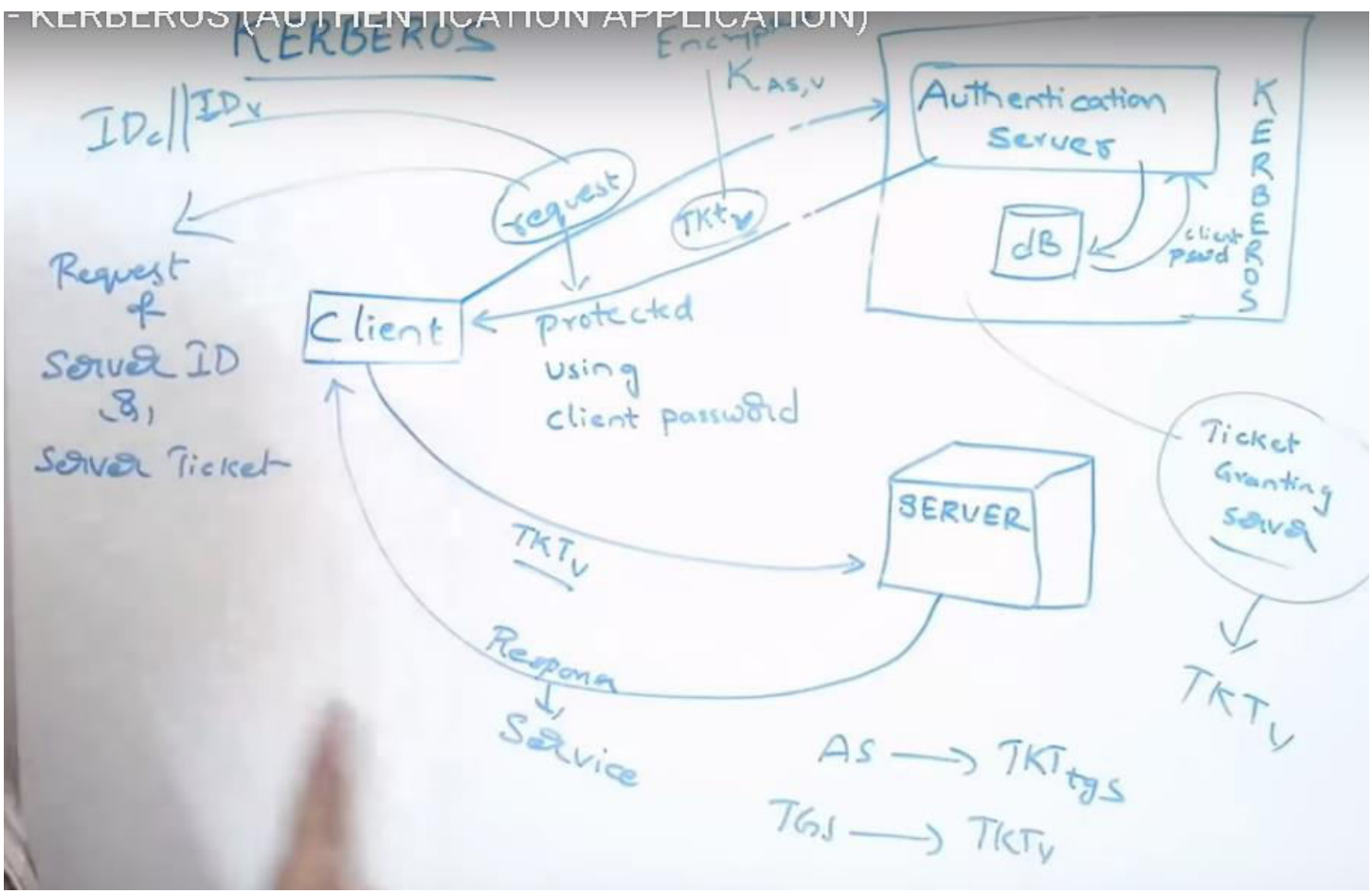
ECC - Encryption & Decryption

- Let the message be M.
- First encode the message M into a point on the elliptic curve
- Let this point be P_m.
- Now this point is encrypted.
- For encrypting choose a random positive integer k.
- Then $C_m = \{kG, P_m + kP_B\}$ where G is the base point.

- For decryption, multiply first point in the pair with receiver's secret key.
  i.e, $kG \times n_B$
- Then subtract it from second point in the pair.
  i.e, $P_m + kP_B - (kG \times n_B) = P_m + kP_B - (kP_B) = P_m$

Activa

#23 kerberos (authemtication application)



$C \longrightarrow AS$ Request – Encrypted using client Password

$AS \longrightarrow C$ $E_{tgs}[Ticket_{TGS}$ & Session Key $(C, TGS)]$

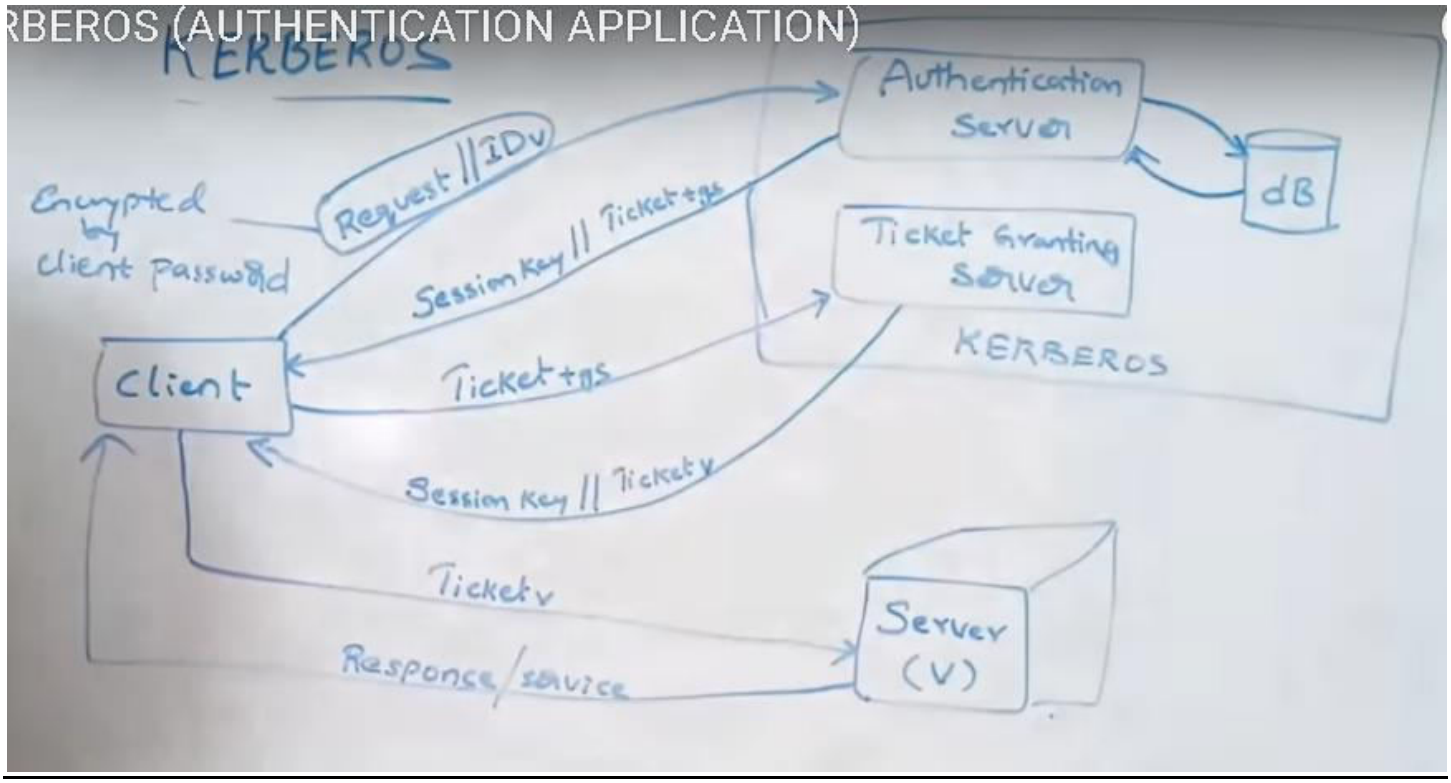
$C \longrightarrow TGS$ $Ticket_{tgs} \,||\, ID_V$

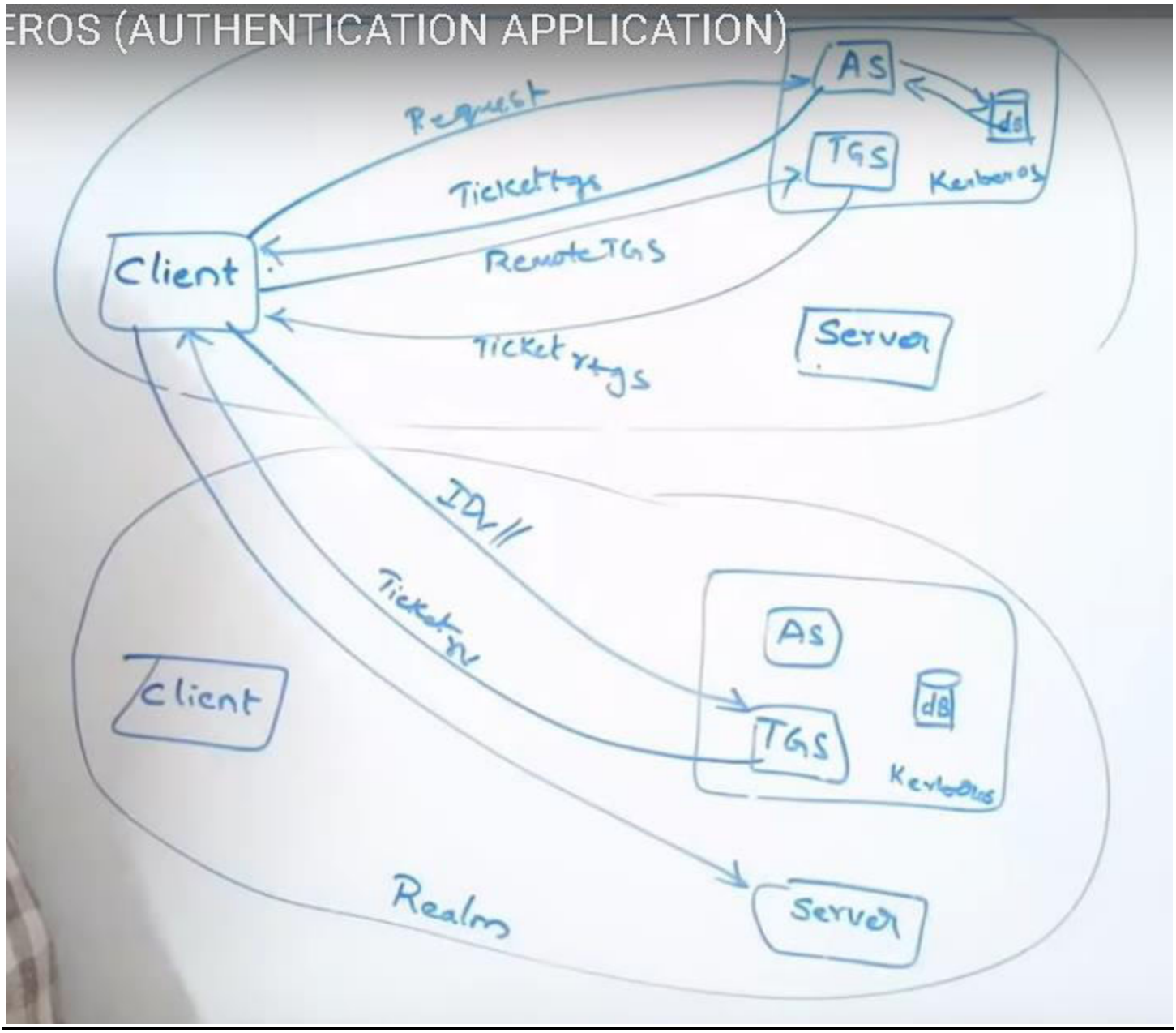
$TGS \longrightarrow C$ $E_V[Ticket_V$ & Session Key $(C, V)]$

$C \longrightarrow V$ $Ticket_V$

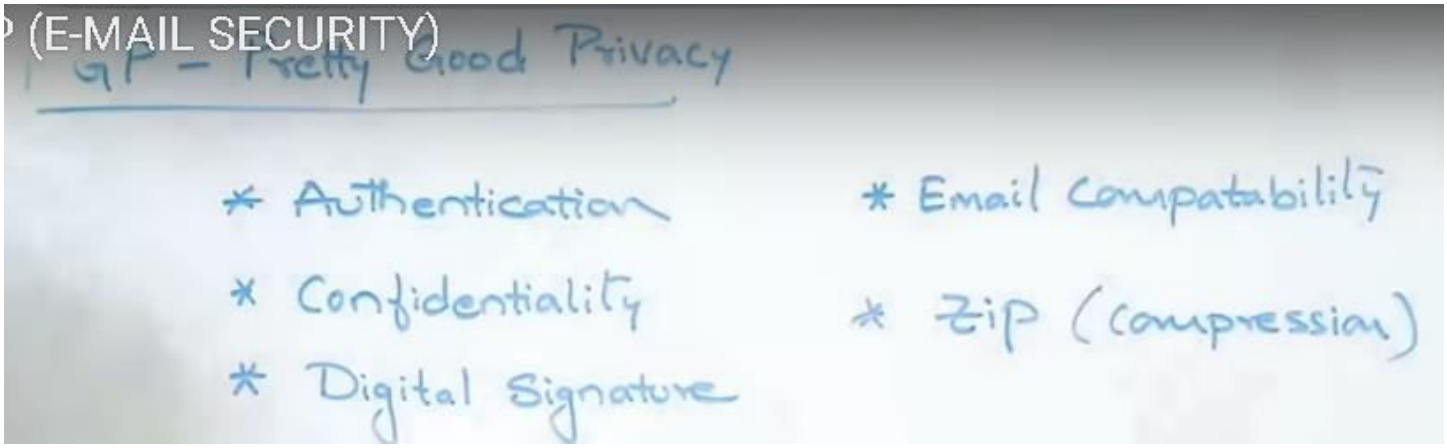
$V \longrightarrow C$: $E_{C,V}[\text{Time Stamp} + 1]$ – Mutual Authentication

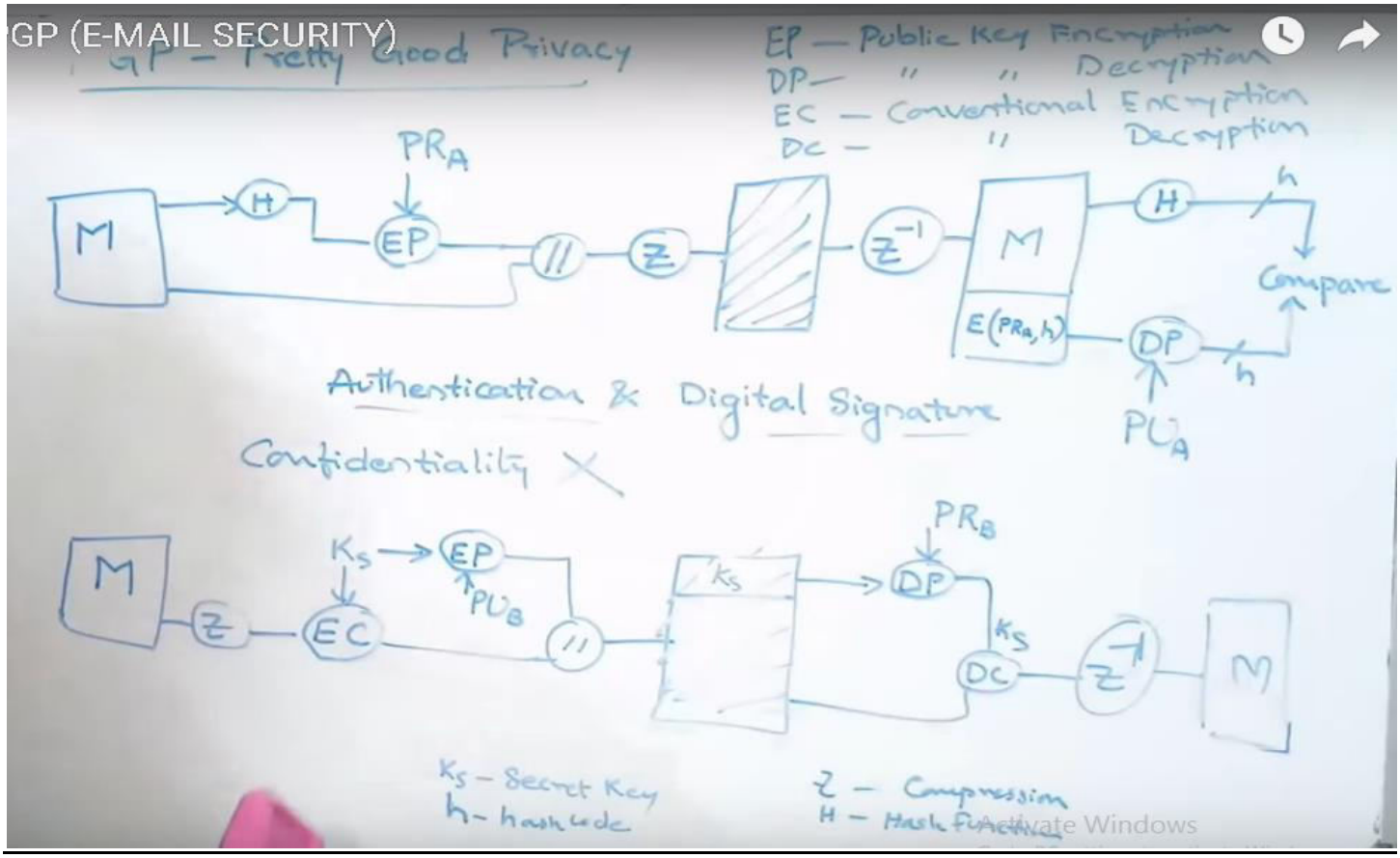
RBEROS (AUTHENTICATION APPLICATION)

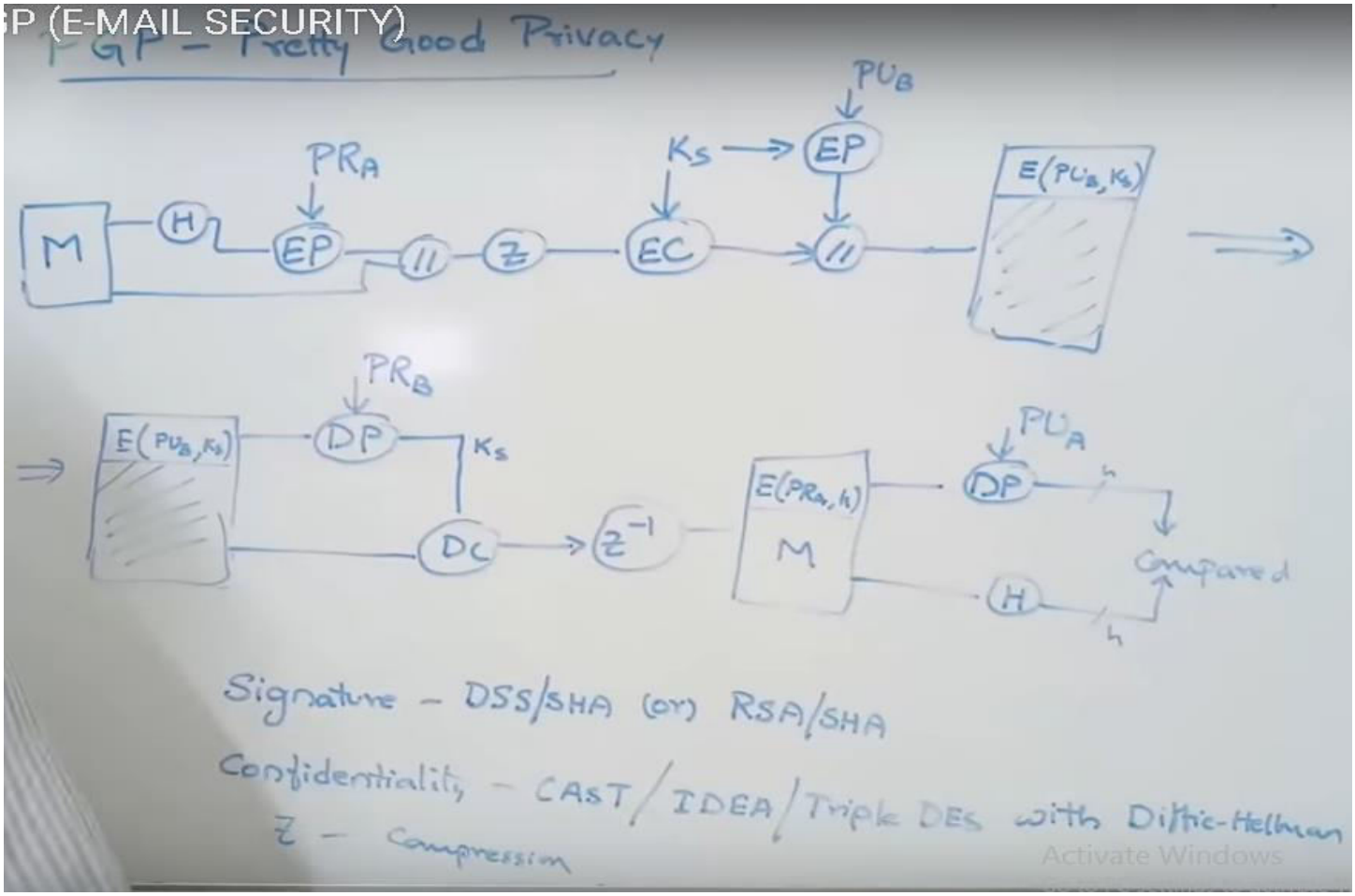



EROS (AUTHENTICATION APPLICATION)
AS
db
TGS
Kerberos
Request
Ticket tgs
Remote TGS
Ticket tgs
Client
Server
ID//
Ticket tv
AS
db
TGS
Kerberos
Client
Realm
Server

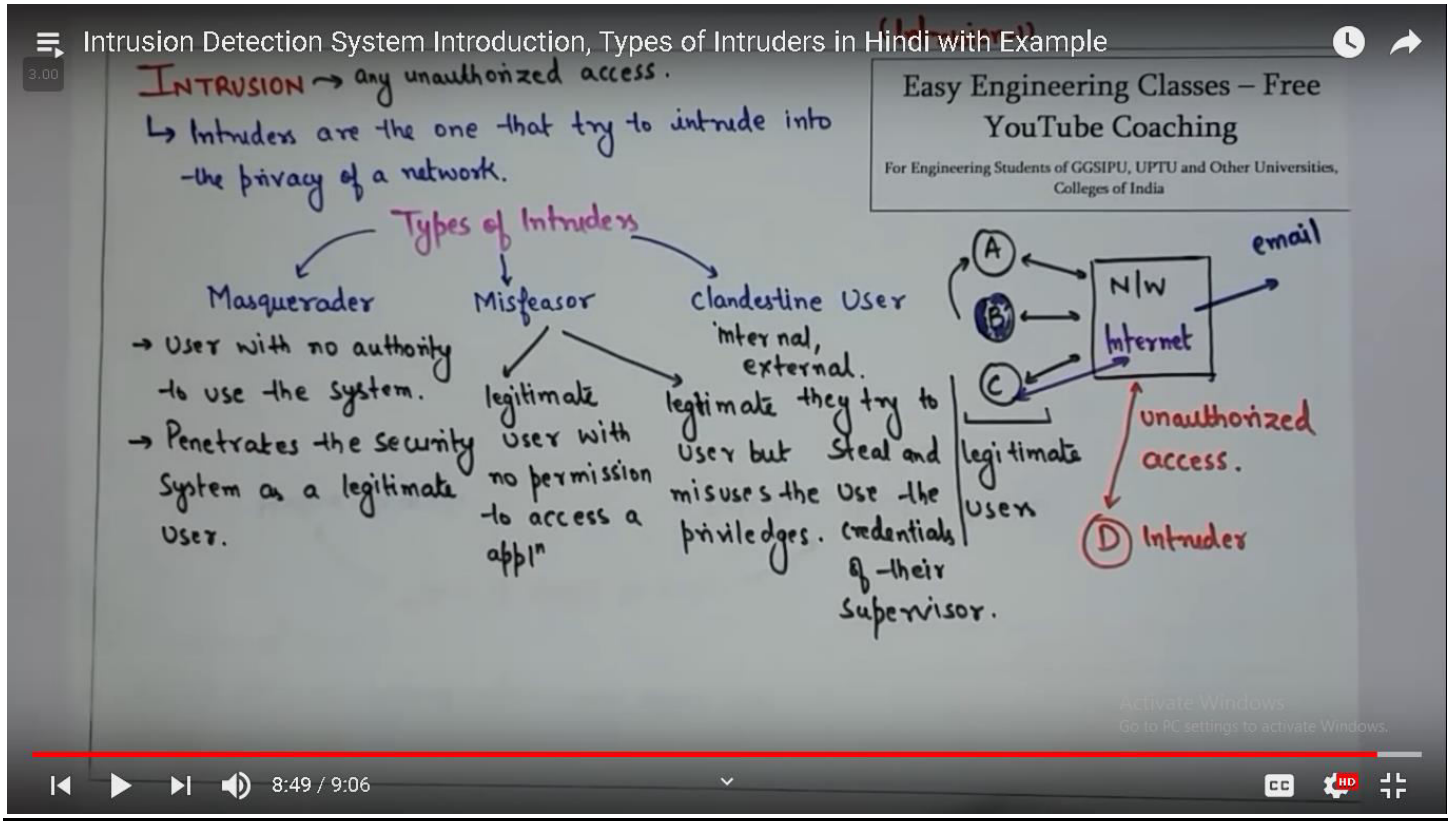
#24 pgp

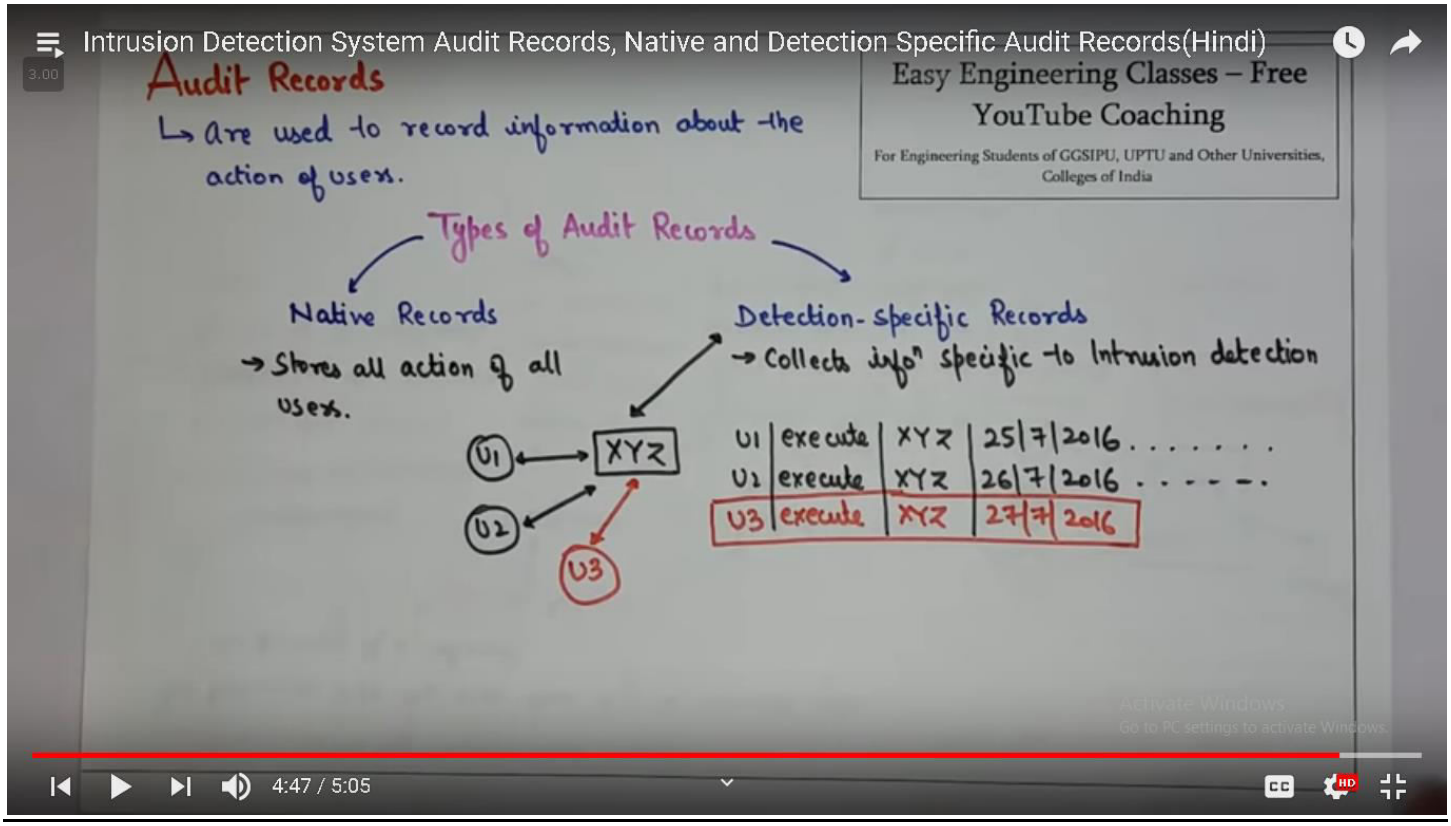

P (E-MAIL SECURITY)
GP – Pretty Good Privacy
* Authentication
* Confidentiality
* Digital Signature
* Email Compatability
* Zip (compression)

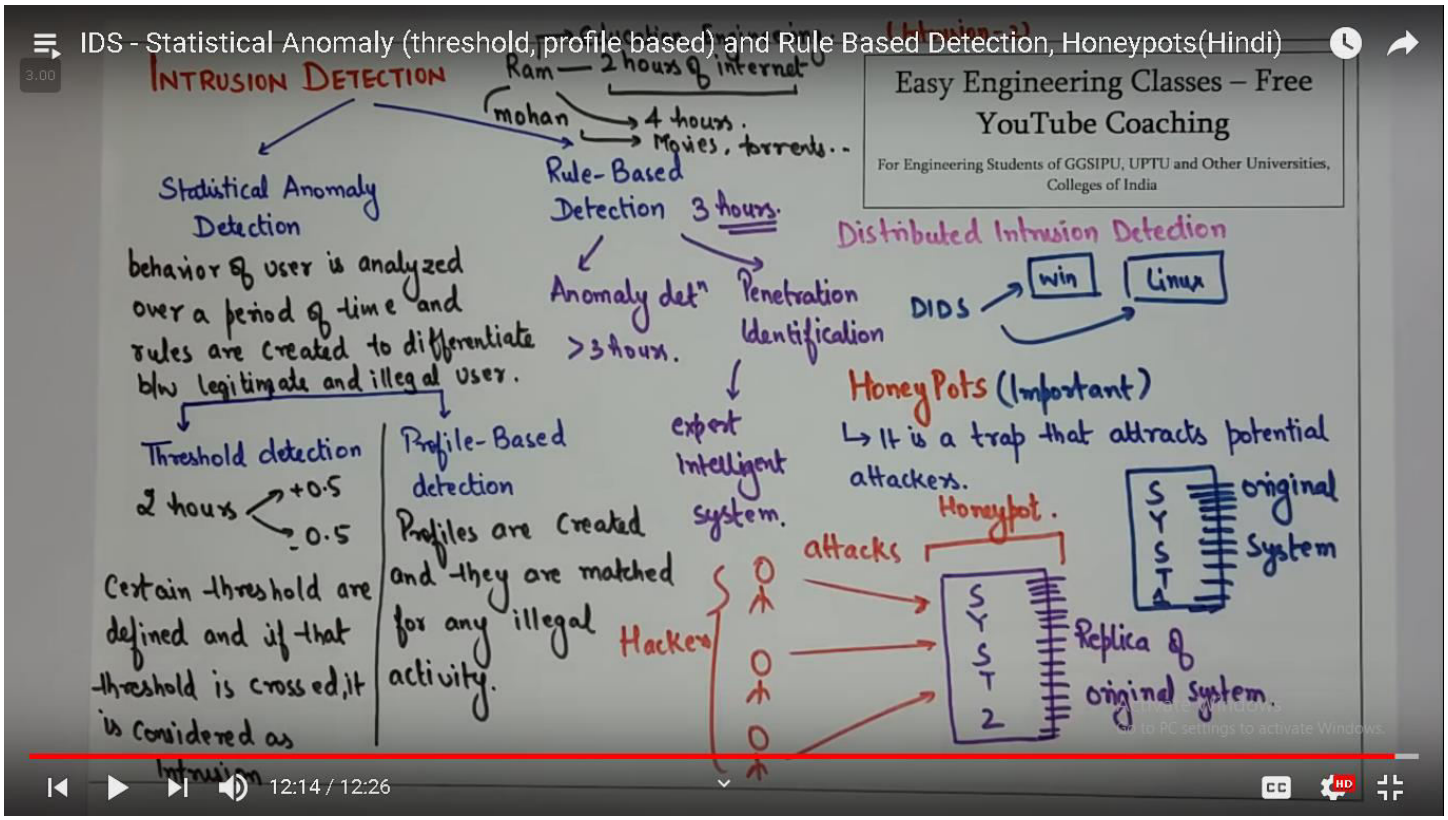

GP (E-MAIL SECURITY)
PGP – Pretty Good Privacy
EP — Public Key Encryption
DP — " " Decryption
EC — Conventional Encryption
DC — " Decryption
PRA
M
H
EP
Z
Z-1
M
E(PRA, h)
H
h
Compare
DP
h
PUA
Authentication & Digital Signature
Confidentiality
M
Ks
EP
PUB
Z
EC
Ks
PRB
DP
Ks
DC
Z-1
M
Ks – Secret Key
h – hash code
Z – Compression
H – Hash Function
Activate Windows


GP (E-MAIL SECURITY)
PGP – Pretty Good Privacy
PUB
PRA
Ks
EP
E(PUB, Ks)
M
H
EP
Z
EC
PRB
E(PUB, Ks)
DP
Ks
DC
Z-1
E(PRA, h)
M
PUA
DP
h
Compared
H
h
Signature – DSS/SHA (or) RSA/SHA
Confidentiality – CAST/IDEA/Triple DES with Diffie-Hellman
Z – Compression
Activate Windows

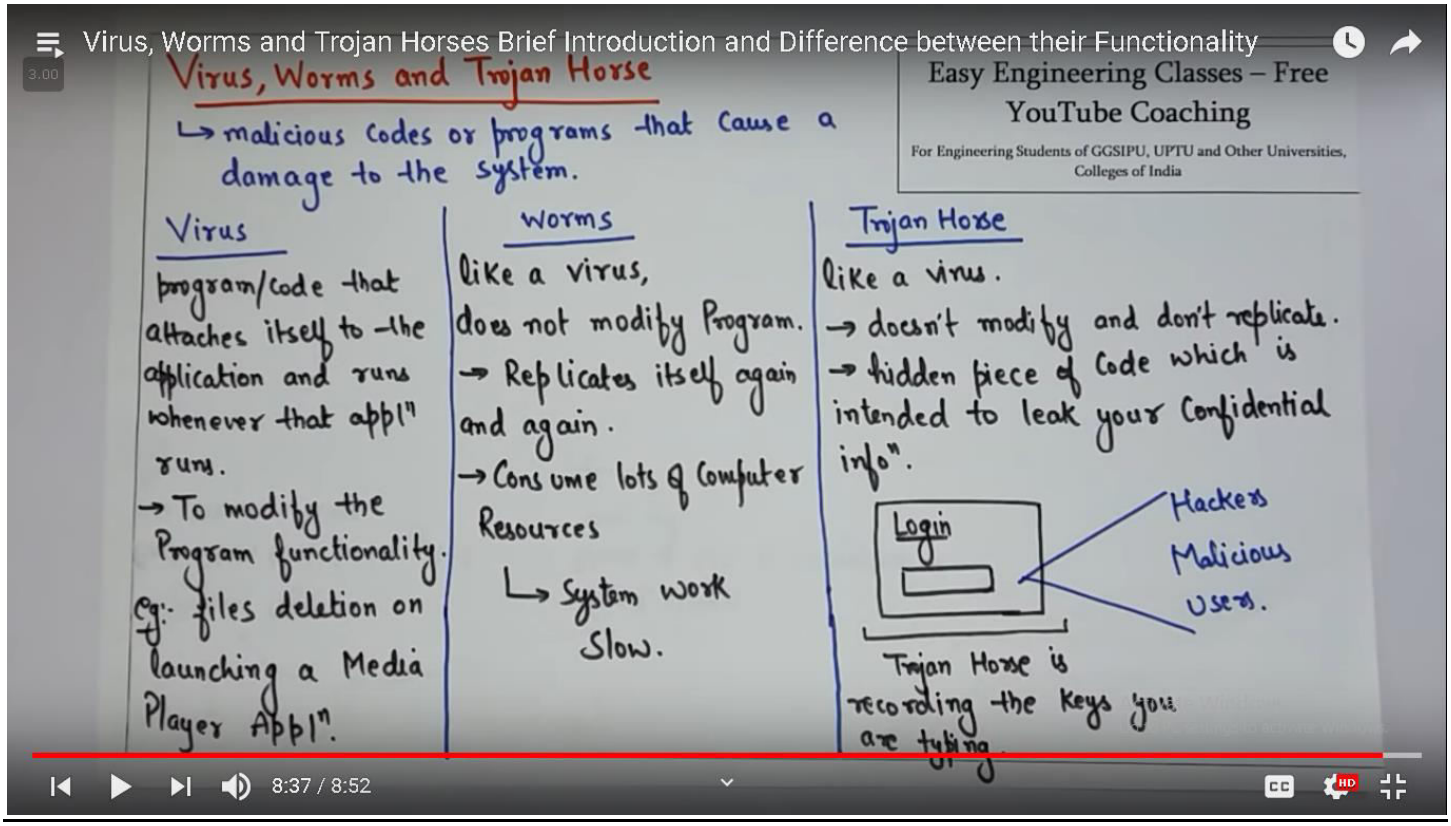
#25 itrusion


Intrusion Detection System Introduction, Types of Intruders in Hindi with Example
INTRUSION → any unauthorized access.
↳ Intruders are the one that try to intrude into the privacy of a network.
Easy Engineering Classes – Free YouTube Coaching
For Engineering Students of GGSIPU, UPTU and Other Universities, Colleges of India
Types of Intruders
Masquerader
→ User with no authority to use the system.
→ Penetrates the security system as a legitimate user.
Misfeasor
legitimate user with no permission to access a appl^n
legitimate user but misuses the priviledges.
Clandestine User
internal, external.
they try to steal and use the credentials of their supervisor.
A
B
C
N/W
Internet
email
legitimate users
unauthorized access.
D Intruder
8:49 / 9:06


Intrusion Detection System Audit Records, Native and Detection Specific Audit Records(Hindi)
Audit Records
↳ are used to record information about the action of users.
Easy Engineering Classes – Free YouTube Coaching
For Engineering Students of GGSIPU, UPTU and Other Universities, Colleges of India
Types of Audit Records
Native Records
→ Stores all action of all users.
Detection-specific Records
→ Collects info^n specific to Intrusion detection
U1
U2
U3
XYZ
U1 | execute | XYZ | 25/7/2016 . . . . . .
U2 | execute | XYZ | 26/7/2016 . . . . . .
U3 | execute | XYZ | 27/7/2016
4:47 / 5:05


IDS - Statistical Anomaly (threshold, profile based) and Rule Based Detection, Honeypots(Hindi)
INTRUSION DETECTION
Ram — 2 hours of internet
mohan → 4 hours.
→ Movies, torrents..
Easy Engineering Classes – Free YouTube Coaching
For Engineering Students of GGSIPU, UPTU and Other Universities, Colleges of India
Statistical Anomaly Detection
behavior of user is analyzed over a period of time and rules are created to differentiate b/w legitimate and illegal user.
Rule-Based Detection 3 hours.
Anomaly detn >3 hours.
Penetration Identification
expert intelligent system.
Distributed Intrusion Detection
DIDS
win
Linux
Honey Pots (Important)
↳ It is a trap that attracts potential attackers.
Threshold detection
2 hours +0.5 -0.5
Certain threshold are defined and if that threshold is crossed, it is considered as intrusion
Profile-Based detection
Profiles are created and they are matched for any illegal activity.
Hackers
attacks
Honeypot.
SYST2
Replica of original system.
SYST
original System
12:14 / 12:26

#27 virus

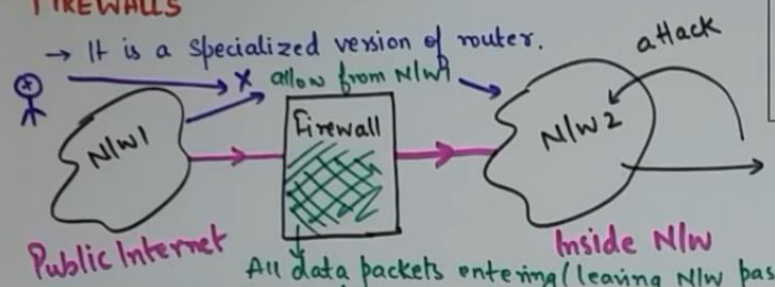

Virus, Worms and Trojan Horses Brief Introduction and Difference between their Functionality
Virus, Worms and Trojan Horse
↳ malicious codes or programs that cause a damage to the system.
Easy Engineering Classes – Free YouTube Coaching
For Engineering Students of GGSIPU, UPTU and Other Universities, Colleges of India
Virus
program/code that attaches itself to the application and runs whenever that appl^n runs.
→ To modify the Program functionality.
eg:- files deletion on launching a Media Player Appl^n.
Worms
like a virus,
does not modify Program.
→ Replicates itself again and again.
→ Consume lots of Computer Resources
↳ System work Slow.
Trojan Horse
like a virus.
→ doesn't modify and don't replicate.
→ hidden piece of code which is intended to leak your confidential info^n.
Login
Hackers
Malicious users.
Trojan Horse is recording the keys you are typing.
8:37 / 8:52

(Firewall)-①

Easy Engineering Classes – Free YouTube Coaching
For Engineering Students of GGSIPU, UPTU and Other Universities, Colleges of India

FIREWALLS

→ It is a specialized version of router.



All data packets entering/leaving N/W pass through firewall and after examining firewall decide whether to allow or not.

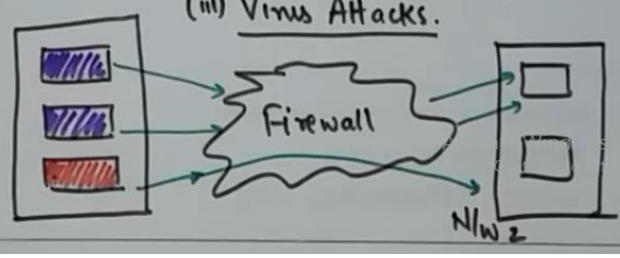
Characteristics of Good Firewall

↳ (i) All traffic must pass through firewall.
(ii) Only authorized traffic should be allowed to pass.
(iii) Firewall should be strong.

Limitations of Firewall

(i) Insider's Intrusion
(ii) Direct Internet Traffic.
(iii) Virus Attacks.

doesn't analyse the Signature or Contents of Packet/data.



(Firewall-2)

Easy Engineering Classes – Free YouTube Coaching
For Engineering Students of GGSIPU, UPTU and Other Universities, Colleges of India

Types of Firewall

Firewall
- Packet Filters
- Application Gateways

Packet Filters

Used certain Criteria to filter out the packet. ↳ IP address, Protocol, Port No., Various other Parameters.

ACL (Access Control List)

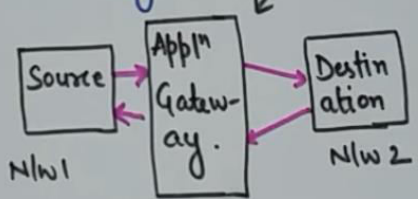
Rule → Selection Criteria (Cond^n) (if-else)
→ Action field → allow Packet
→ Block Packet

if ip_address = 127.0.0.1;
then : allow :
else Block.

Packet Filter
- Stateless: Simply on ACL Basis.
- Stateful: monitors the Ongoing traffic. → used in all modern firewalls.

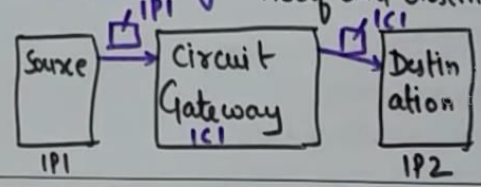
Application Gateways

↳ Acts as a relay node.
→ Function as Proxy Server.



Circuit Gateway

→ Creates a new Conn^n b/w itself and destination.



md5

HASH FUNCTIONS: [Compression func^n]

It is a mathematical func^n that converts a numerical i/p value into another compressed numerical value.

↳ o/p is always of fixed length.

(o/p) (4 bit) ← 10, 12, ⋮, 50 (i/p)

Message (Any Length) → Hash func^n → Hash Value < M size.

Size

Features:

(i) fixed length o/p

(ii) Compression func^n

(iii) Digest (Smaller rep^n of larger data)

Properties: (i) M ⟶ H (Easy) H ⟶ M (Very Hard)

(ii) M ⟶ H, ⋮, n - - - - > H } Same hash value for same message everytime.

(iii) $M_1 \longrightarrow H_1$, $M_2 \longrightarrow H_2$ } $H_1 = H_2$ → should not happen.

MD5 [MESSAGE DIGEST]

↳ developed by Ron Rivest.

↳ Fast and produces 128-bit message digests.

WORKING:

original message | Padding

↓

exact multiple of 512. { original message | Padding | Length

↓

512 bit Block1 - - - - - - - - 512-bit Block$_n$

(iv) Initialize 4-chaining variables. (32-bit, A, B, C and D)

(i) Padding is done such that total length is 64 bit less than exact multiple of 512.

1000 bits + 472 = 1472

512 × 2 = 1024

512 × 3 = 1536

64

1472

64 bit less than exact multiple of 512.

(ii) Append original length bef. padding (modulo 64)

1000] length mod 2^{64}.

(iii) Divide it in 512-bit blocks.

(v) Process Blocks

↳ Copy four chaining Variables into Corresponding Variables.
{A=a, B=b, C=c, D=d}

↳ divide 512-bit block into 16 (32bit blocks)

↳ Four Rounds.

512 bit Block → 32-bit, 32-bit, ..., 32bit (16)

16 Sub Blocks → One Round ← Constant (t)

One Round ← a, b, c, d

$a = b + ((a + \text{Process}, P(b,c,d) + M[i] + T[k]))$

<< Shift.

SECURE HASH ALGORITHM (SHA): (NIST)

SHA is a modified version of MD5.

↳ o/p is a message digest of 160 bits in Length.

Process Block → Copy of C.V
→ 512 → 16 Sub Block (32)
→ four Round (20 steps)

SHA Properties:-

(i) Generating original message from digest
(ii) finding two messages generating Same digest
} Infeasible.

Working of SHA:

(i) Padding [64 bit less than exact multiple of 512]
(ii) Append Length
(iii) divide the i/p into 512-bit blocks
~~(iv)~~ Five chaining Variables (A, B, C, D, E)
(v) Process Blocks.

} exactly same as MD5

SHA-2

Comparison b/w MD5 and SHA: (IMP)

		MD5 [Faster]	SHA [Secure].
(i)	Length in bits	128	160
(ii)	Attack to find orig. message	2^{128} operations	2^{160} operations
(iii)	Two Msg with Same MD	2^{64} operations.	2^{80} operations
(iv)	Succesfull Attacks	Some reported incidents of MD5 break.	No Such claim
(v)	Speed	→ Faster	Slower

#ssl

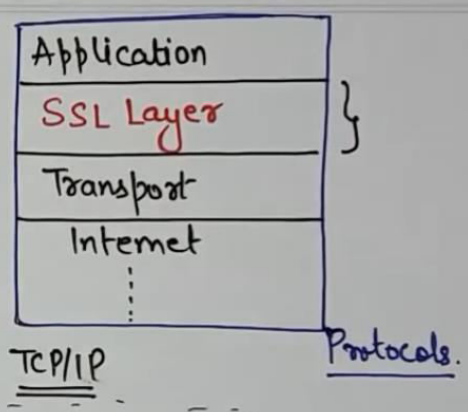
SSL-1

SSL (SECURE SOCKET LAYER):

↳ Internet Protocol for Secure exchange of info^n b/w browser and Server.

↳ Provides Security at Transport ... Layer.

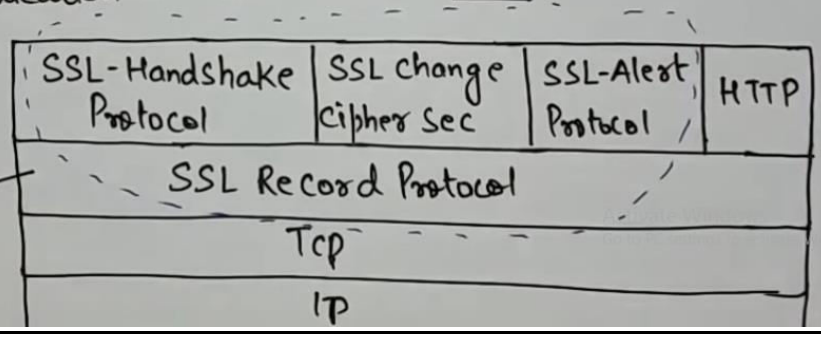
GOALS: → i) Confidentiality [C]
→ ii) Integrity [I]
→ iii) Authentication [A]



WORKING OF SSL:-

SSL Protocol Stack:

Offer Services to higher SSL-Protocols.



Asymmetric Key Vs. Symmetric Key Encryption

Easy Engineering Classes – Free YouTube Coaching
For Engineering Students of GGSIPU, UPTU and Other Universities, Colleges of India

Asymmetric Key	Symmetric Key
i) Different Keys.	i) Same Key.
ii) Encryption is done with the help of user's public Key. Decryption with private key.	ii) Both Encryption and Decryption are done with user's pvt./Secret Key.
iii) C.T may be of Large Size	iii) C.T Same or less in size
iv) No Problem of Key Exchange	iv) Problem
v) Confidentiality, Authentication, Non-repudiation	v) Confidentiality
vi) Process is Slow	vi Fast

Activate Windows
Go to PC settings to activate W