AUTHENTICATION FUNCTIONS

Authentication → verifying the users identity

Raman ⟶ John

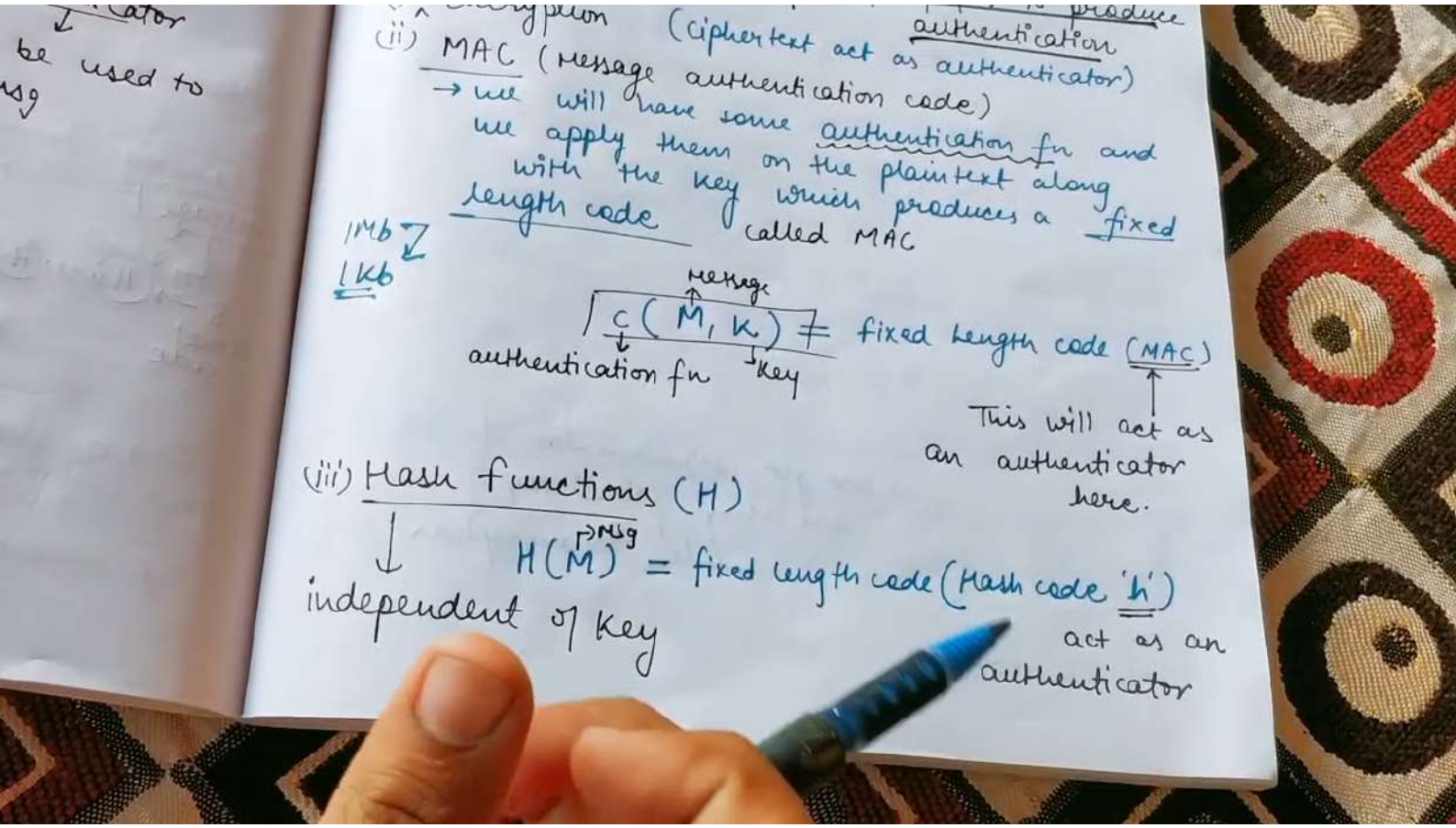
An authenticator must be there to authenticate the message.

Types of Authentication / Types of fns to produce authentication

(i) Message Encryption (ciphertext act as authenticator)

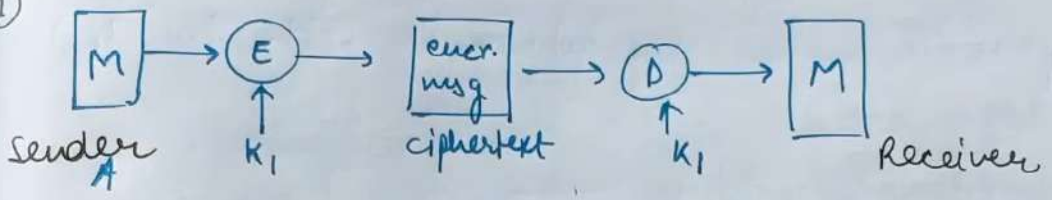
(ii) MAC (message authentication code)

→ we will have some authentication fn and we apply them on the plaintext along with the key which produces a fixed


be used to
(ciphertext act as authenticator)
(ii) MAC (message authentication code)
→ we will have some authentication fn and we apply them on the plaintext along with the key which produces a fixed length code called MAC
1MB
1KB
message
C(M, K) = fixed length code (MAC)
authentication fn
key
This will act as an authenticator here.
(iii) Hash functions (H)
H(M) = fixed length code (Hash code 'h')
act as an authenticator
independent of key

1. Message encryption → ciphertext is an authentication

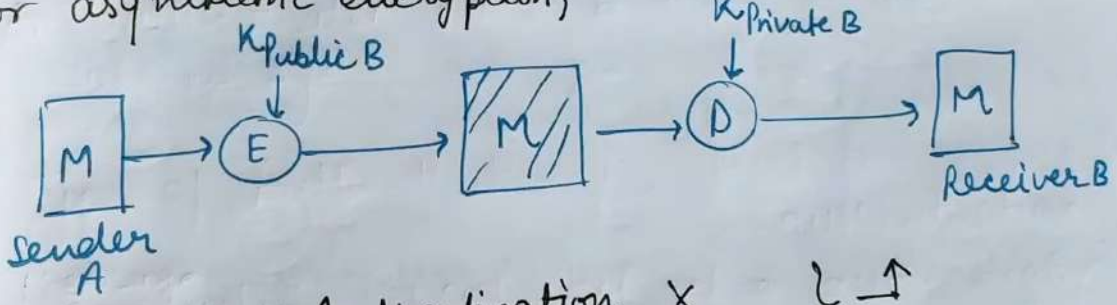
(1)



→ key K_1 shared only b/w sender & Receiver only.

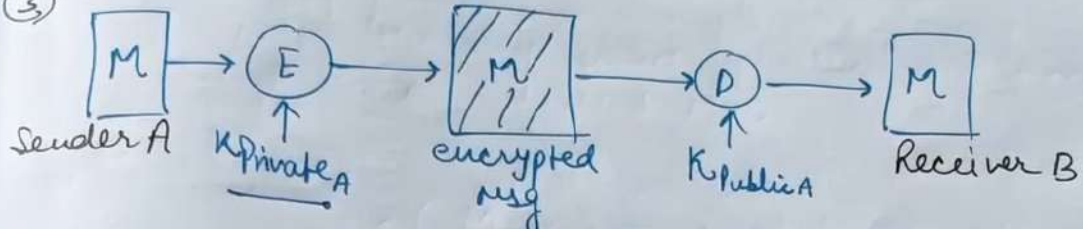
for asymmetric encryption,

(2)



Authentication ✗
confidentiality ✓

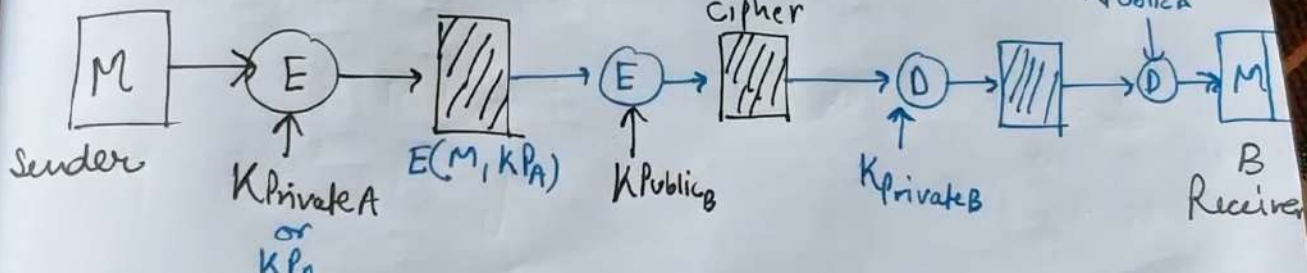
(3)



authentication ✓
confidentiality ✗

(4)

To get both, use dual encryption & decryption
and also signature



Block cipher modes of Operation

for different types of messages, we need different modes of operations.

5 modes of operation are:

(i) ECB electronic codebook mode
(ii) CBC cipher block chaining mode
(iii) CFB cipher feedback mode
(iv) OFB output feedback mode
(v) CTR counter mode

ECB (Electronic codebook mode)

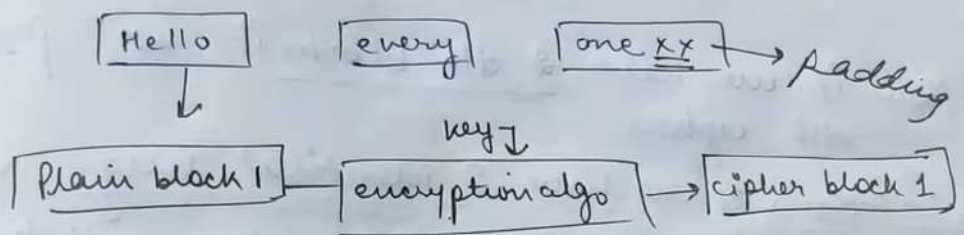
* simplest mode of operation
* plaintext is divided into a no. of fixed size block.
* If message is not a multiple of block size, then

padding is done ... block size, then

* Take one block at a time and encrypt it.
* Same key used for encryption and decryption.

eg Let block size = 5

Plain Text → Hello everyone



This will happen for all the [redacted] blocks.

Note → best for short amount of data, such as a key
→ Not secure for lengthy data
☹ If identical blocks appear, then this mode produces same cipher.

Payphone
one direction

→ for decryption process

∴ This will happen for every box.

* If message is
* padding is done
* Take one block
* Same key
eg] Let block siz
Plain Tex
Hello
Plain block
Thi
Note →

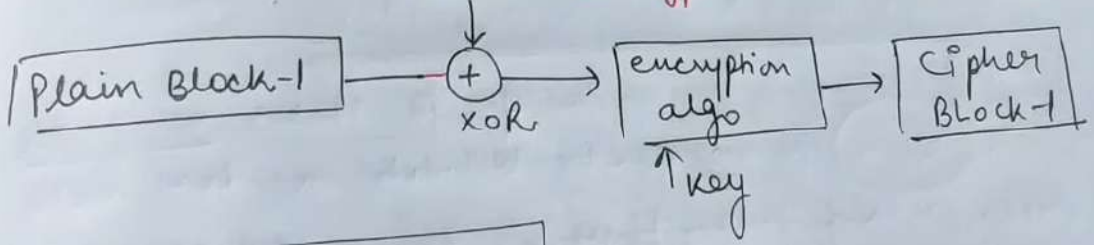
(ii) CBC — Cipher Block Chaining mode

* To overcome security issues of ECB mode (b/c in ECB if same blocks appear then ciphertext produced will be same).

* I/P to the encryption algo is XOR of the current plaintext block and the preceding ciphertext block. So, repeating patterns not exposed.

* Same key for encrypt + decrypt.

IV is data block of same size
Initialization vector is used in 1st encryption and 1st decryption

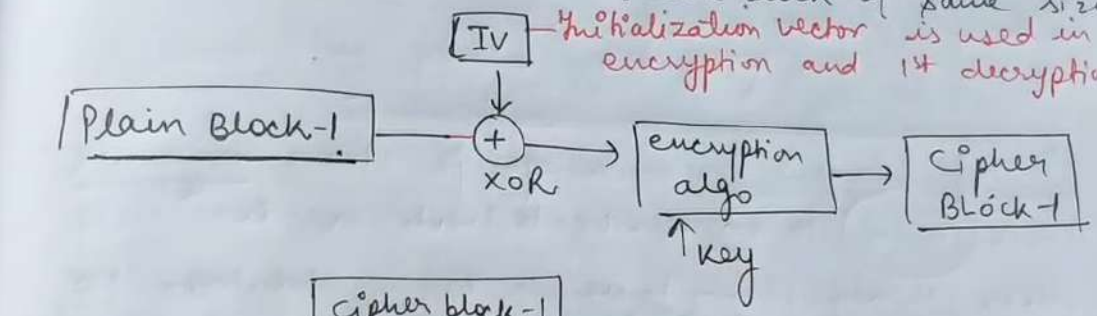


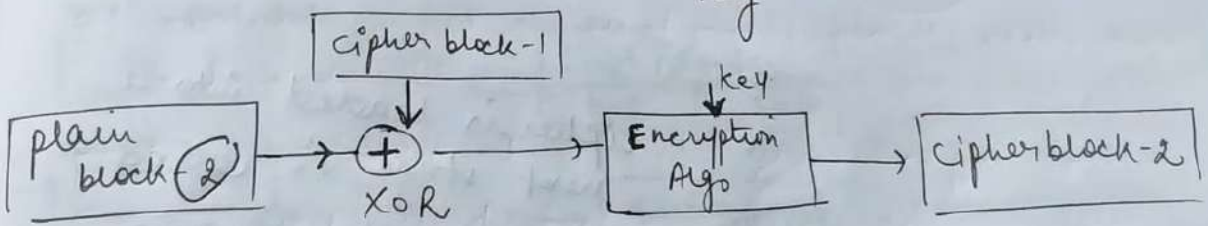
produced will be same).

(*) I/P to the encryption algo is XOR of the current plaintext block and the preceding ciphertext block. So, repeating patterns that exposed.

(*) Same key for encrypt + decrypt.

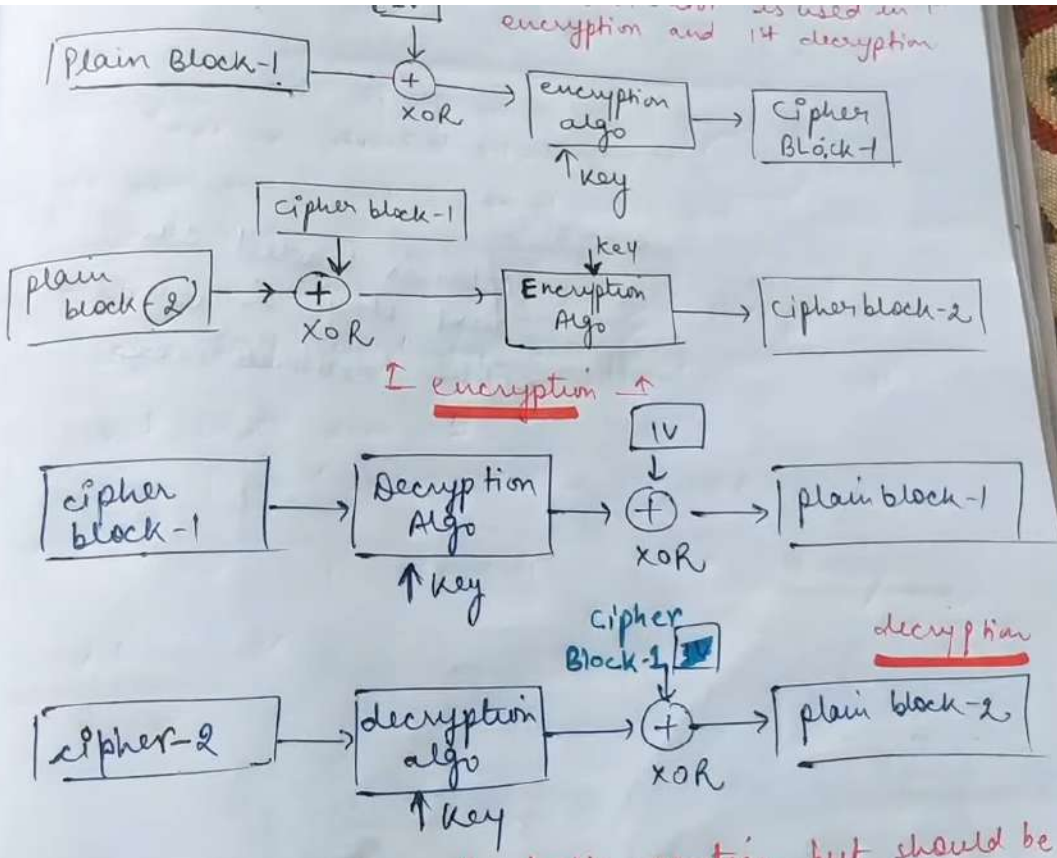
IV is data block of same size
Initialization vector is used in 1st encryption and 1st decryption.





↕ encryption ↑

IV



encryption and at decryption

the succeeding blocks

if

produce

& if

e same

Key to both parties, but should be

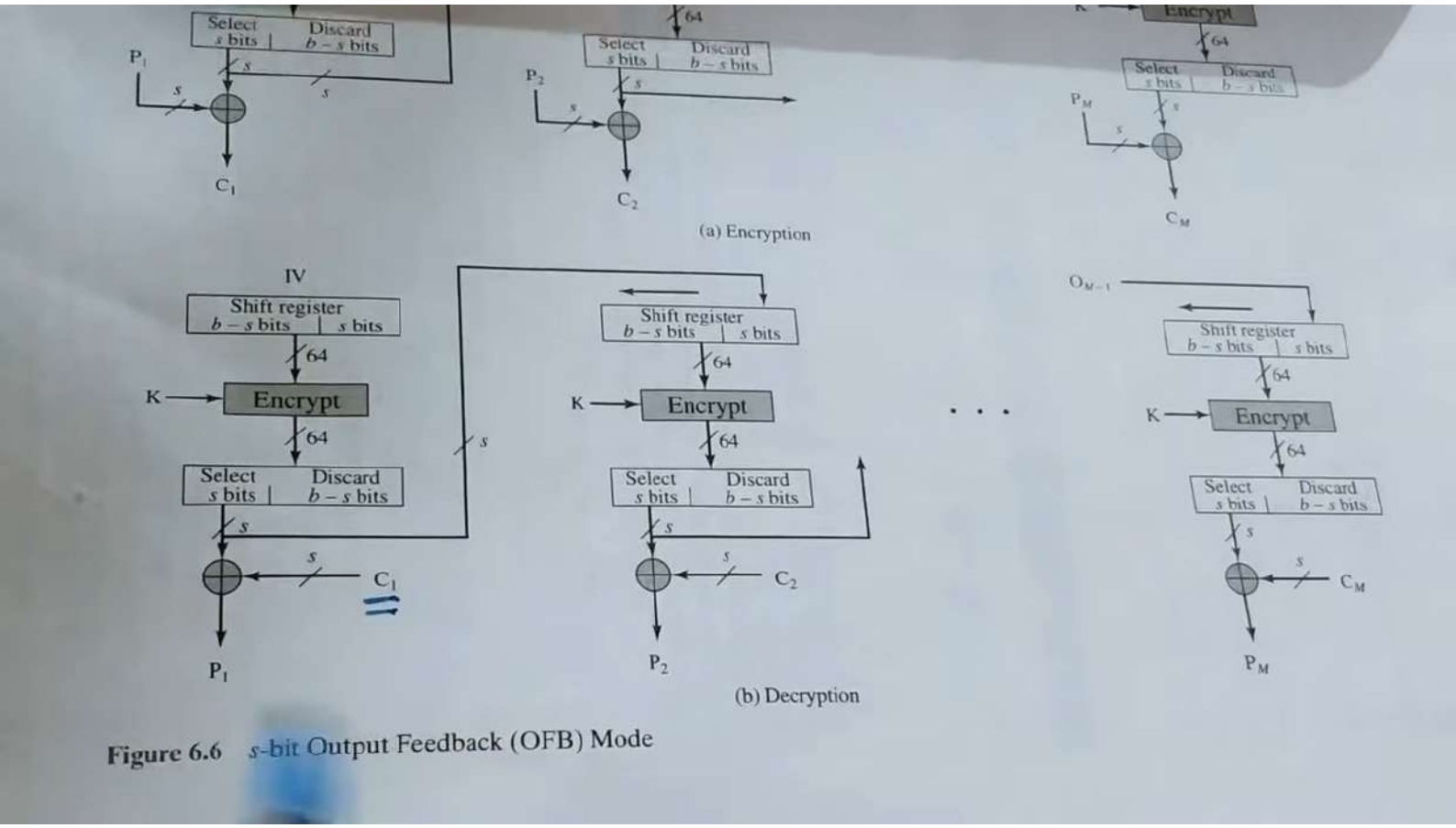
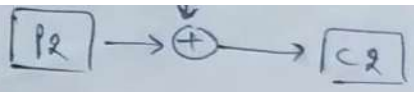


Figure 6.6 *s*-bit Output Feedback (OFB) Mode



encryption →

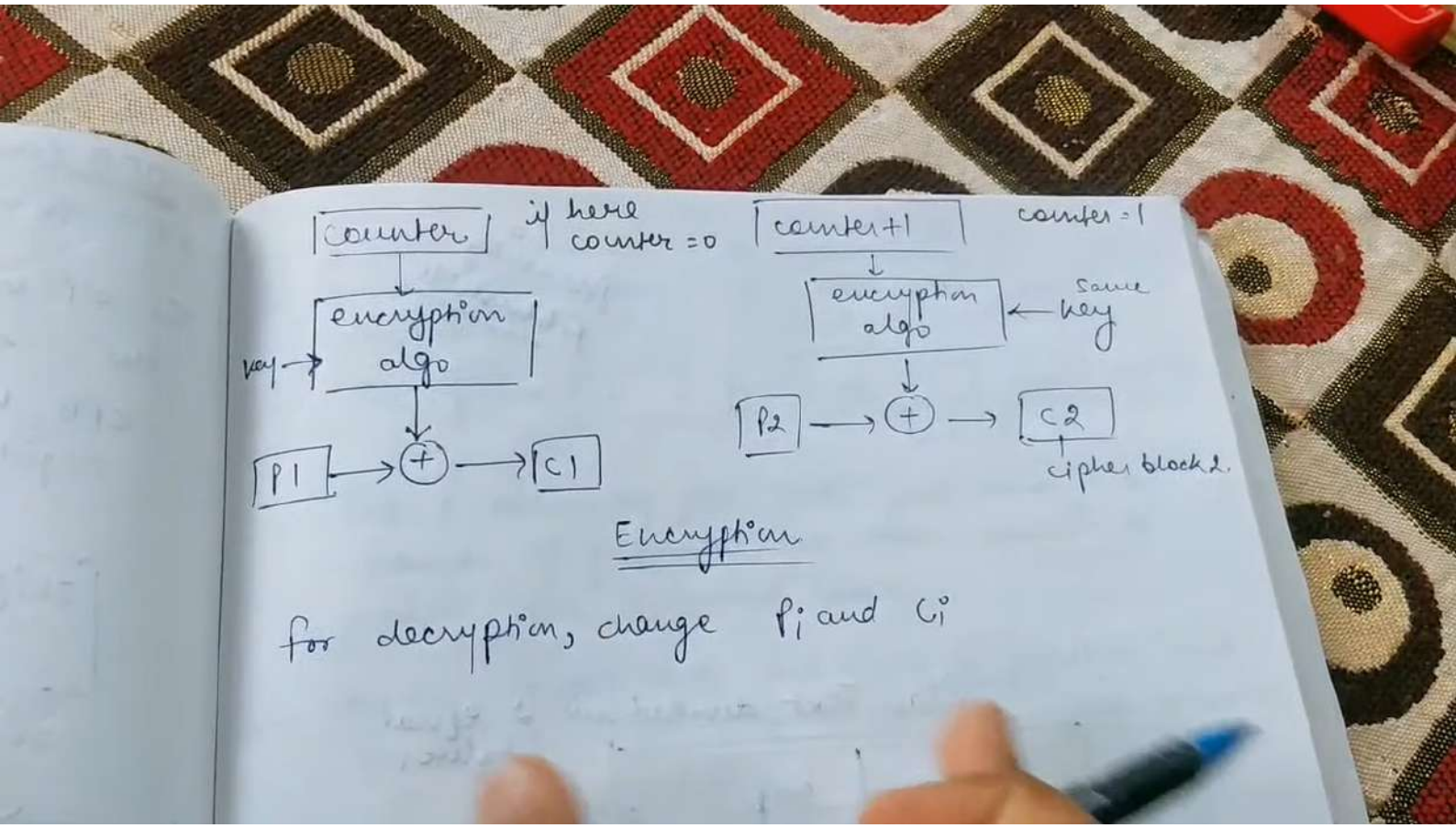
For decryption, c_j and p_j will be exchanged.

5) COUNTER mode [CTR]

→ Simple and fast

→ a counter, equal to the plaintext block size is used.

* counter is initialised to some value and then incremented by 1 for each subsequent block.



Encryption

for decryption, change P_i and C_i

BLOWFISH ALGORITHM in CRYPTOGRAPH & Network Security

* symmetric key algo.
* block cipher (64 bit) algo.

encryption technique designed by Bruce Schneier (1993)

* It is an alternative to DES encryption Technique & IDEA algo.

Block size / Plain Text in 1 block → 64 bit
Key size → variable (32 to 448) bits
No. of subkeys → 18 (P-array) $P_0, P_1 \cdots P_{17}$ → 32 bit each
No. of rounds → 16
No. of substitution boxes → 4 S-boxes having 256 enteries ↓ each is of 32 bit

Note → It follows feistel structure

1) Generation of subkeys (or P[1] to P[18] as per your convenience)

→ 18 subkeys (P[0] .. P[17]) are used for encryption as well as decryption

→ 18 subkeys are stored in a P-array with each array element being 32-bit entry.

eg P[0] = "243f6a68" } hexadecimal representation of each subkey.
P[1] = "8979abf3"
:

* block cipher (64 bit) algo. [illegible]

* It is an alternative to DES encryption Technique & IDEA algo. [illegible] as per your convenience

Block size / Plain Text in 1 block → 64 bit
Key size → variable (32 to 448) bits
No. of subkeys → 18 (P-array) $P_0, P_1 \dots P_{17}$ → 32 bit each
No. of rounds → 16
No. of substitution boxes → 4 S-boxes having 256 entries each is of 32 bit

1) Generation of subkeys
→ 18 subkeys (P[0] -- P[17]) are used for encryption as well as decryption
→ 18 subkeys are stored in a P-array with each array element being 32-bit entry.

eg P[0] = "243f6a68" } hexadecimal representation of each subkey.
P[1] = "8979abf3"
:
P[17] = "c97c5odd"

Note → It follows feistel structure

fast, compact, simple, Secure
compact → executes in less Memory
simple → XOR & add operation
Secure → variable length key

Confidentiality → refers to, protecting the information from being accessed by unauthorized parties.

In other words, only the people who are authorized to do so can gain access to the sensitive data.

Data Integrity → It is the overall completeness, accuracy and consistency of data, over its entire life-cycle.

i.e, data must not change during transit, and steps must be

completeness, accuracy and consistency of data, over its entire life-cycle.

i.e, data must not change during transit, and steps must be taken ~~to~~ ~~steps must be~~ to ensure that data cannot be altered by unauthorized people.

[redacted] Authentication → verifying the users identity.

Authenticity is used to make sure you really communicate with the actual person.

DES ANALYSIS

Properties

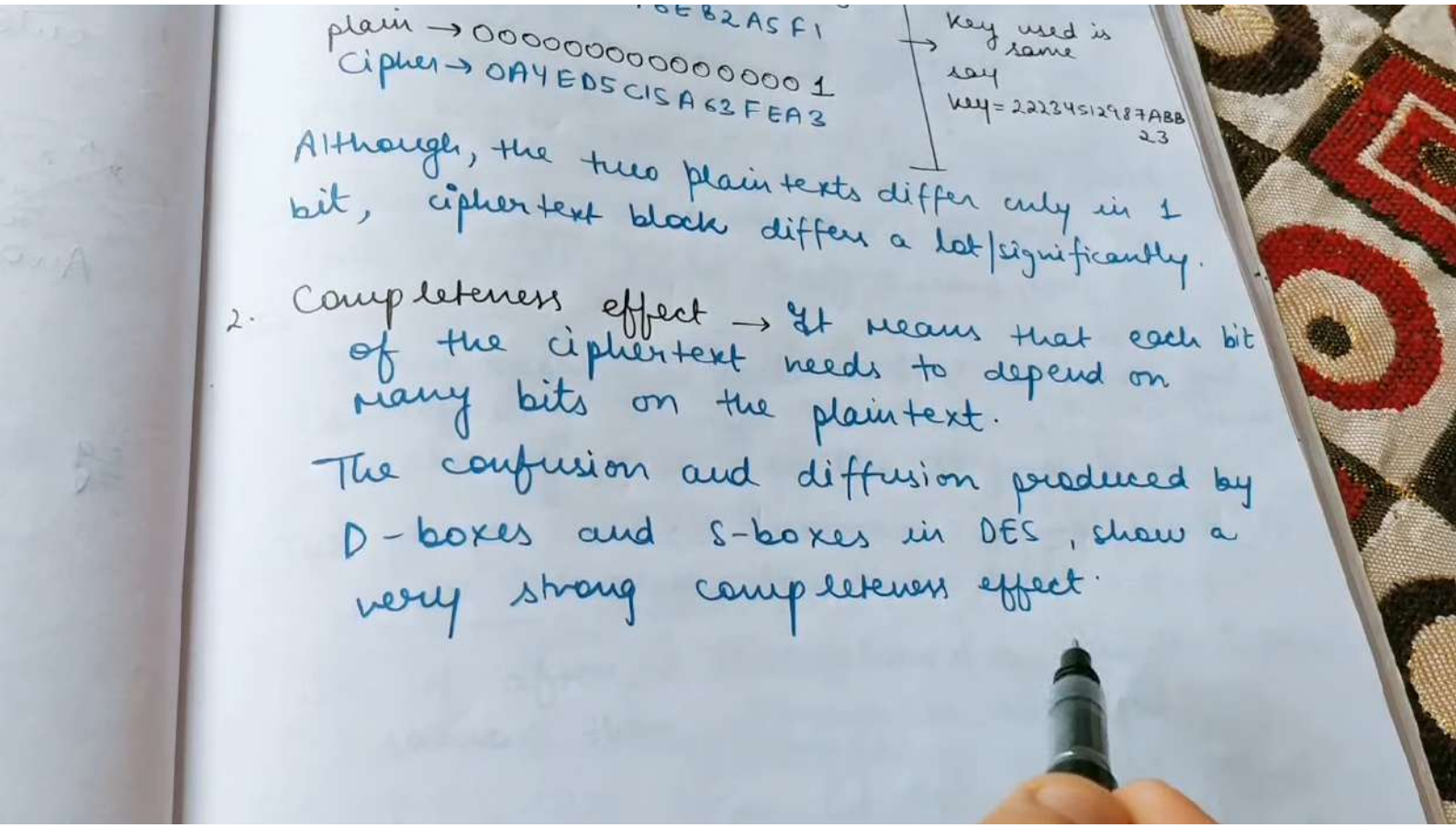
(i) Avalanche effect → It means a small change in plaintext (or key) should create a significant change in the ciphertext.

DES has been proved to be strong with regard to this property.

eg. Plain → 0000000000000000
cipher → 4789FD476E82A5F1

plain → 0000000000000001
Cipher → 0A4ED5C15A63FEA3

→ key used is same say key = 22234512987ABB23



Although, the two plaintexts differ only in 1 bit, ciphertext block differs a lot/significantly.

2. Completeness effect → It means that each bit of the ciphertext needs to depend on many bits on the plaintext.

The confusion and diffusion produced by D-boxes and S-boxes in DES, show a very strong completeness effect.

D.E.S

Data Encryption Standard

(i) block cipher
(ii) symmetric cipher (same key for encryption + decryption)
(iii) 64 bit plain text block
It encrypts the data in blocks of size 64 bit each
(iv) 16 rounds. each round is a fiestel round.

steps
(i) Initial permutation
(ii) 16 fiestel rounds
(iii) swapping / left right swap
(iv) Final permutation / Inverse initial permutation

Basic structure

64 bit plain text
↓
| Initial permutation |

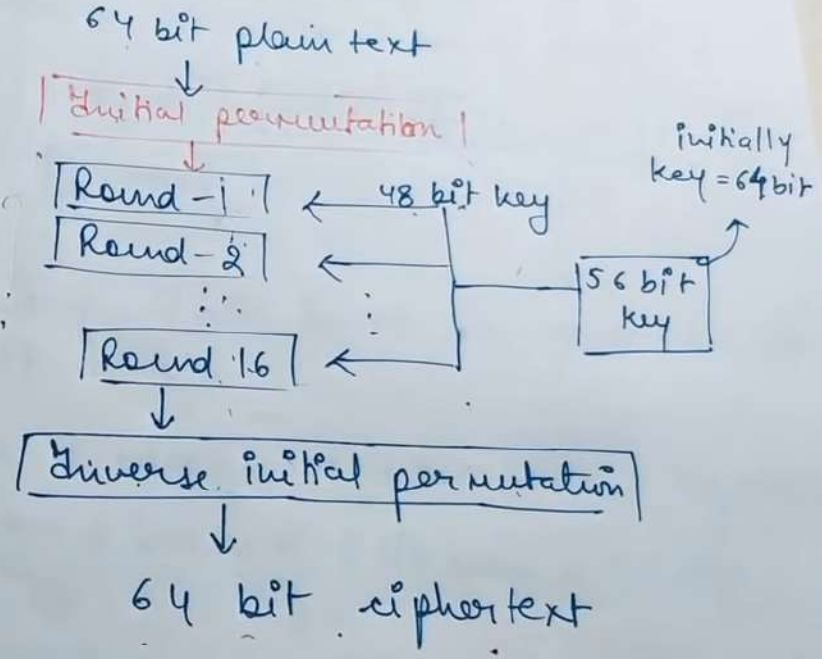
initially key = 64 bit

Round - 1

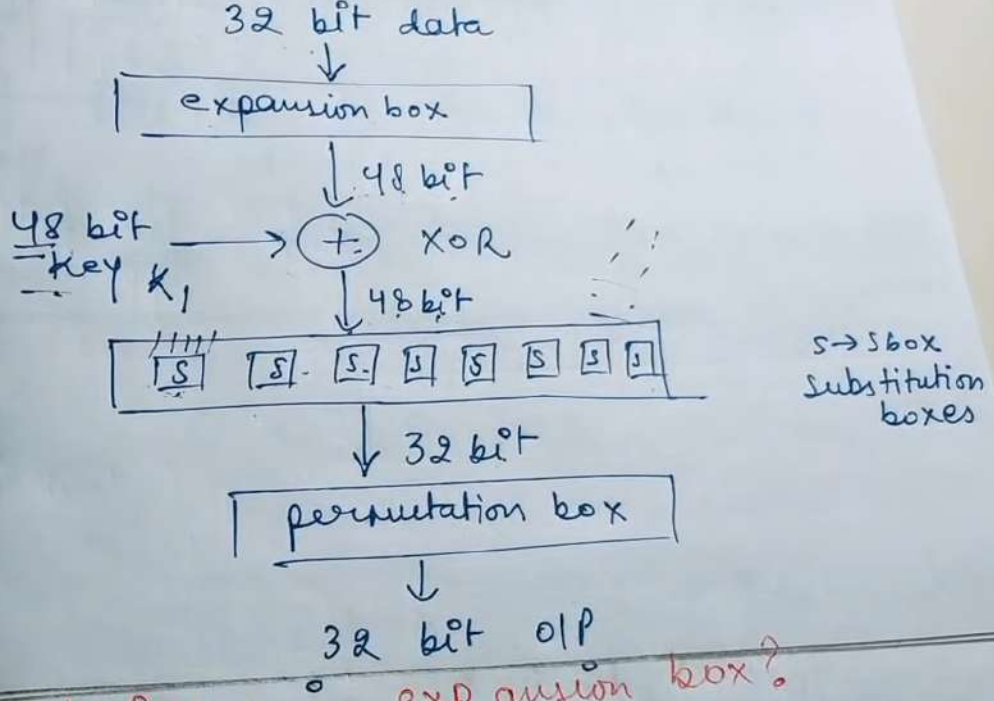
Round - 1

(iv) Final permutation | left right swap | Inverse initial permutation

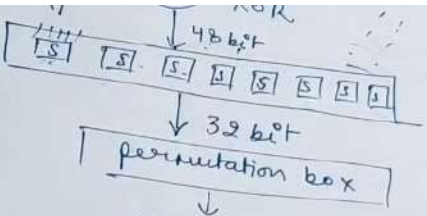
Basic structure



Function definition



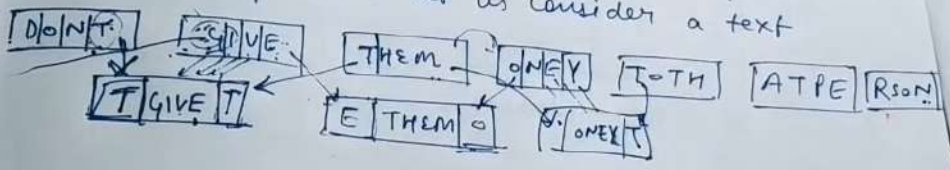
What happens in expansion box?



S → 8 box substitution boxes

What happens in expansion box?

32 bit data will be → 1's and 0's form but for explanation let us consider a text



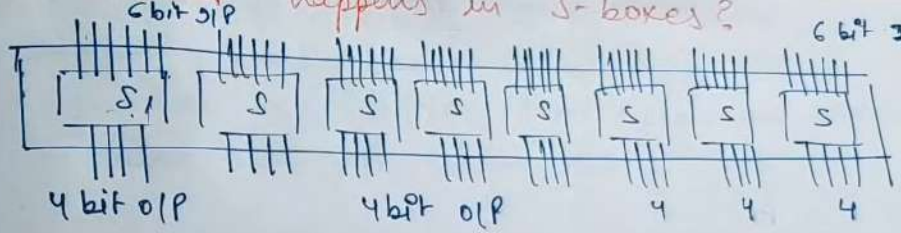
So here every 4 bit block is converted to a 6 bit block

There were 8 blocks of 4 bit each = 32 bit

Now, there are 8 blocks of 6 bit each = 48 bit

Now these 48 bits XOR with 48 bit key

Ques What happens in S-boxes?



O/P → 4×8 = 32 bits

These 32 bits will go into permutation box.

0 1 1 1

how 6 bit converted to 4 bit?

→ eg 0 0 1 0 1 1

↓ column

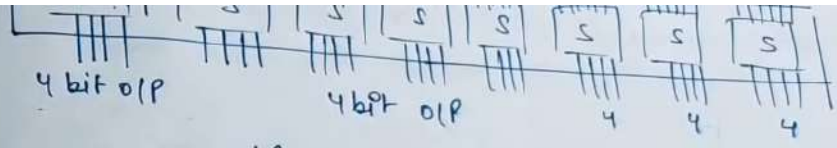
Row

0 1 → 1

0 1 0 1 → 5

S1 box - table 1

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



0 1 1 1 — o/p → 4 × 8 = 32 bits. These 32 bits will go into permutation box.

how 6 bit converted to 4 bit ?

→ eg 0 0 1 0 1 1

↓ column

0 1 → 1

0 1 0 1 → 5

Row

S₁ box - table 1

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

10 → 1010

↓ numbers will be filled.

will have a dif table.

DES Weaknesses

- **Key size** — 2^{56}

Critics believe that the most serious weakness of DES is its key size of 56 bits.

B/c, with today's technology (like parallel processing and v. powerful processors) it can easily be cracked.

$\Rightarrow$ 2^{56} keys (brute force attack)

→ ∴ we use triple DES (3DES) with two keys (112 bits) or triple DES with 3 keys (168 bits).

(ii) **Weak keys** → four out of 2^{56} keys are called weak keys. after the parity drop operation is the one which consists of

→ ∴ we use triple DES (3DES) with two keys (112 bits) or triple DES with 3 keys (168 bits).

2^{56} keys (brute force attack)

(ii) Weak keys → four out of 2^{56} keys are called weak keys.

A weak key is the one which consists of (after the parity drop operation) all 0s, all 1's or half 0's and half 1's.

The disadvantage of using a weak key is:
If we encrypt a block with a weak key and subsequently encrypt the result with the same weak key, we get the original block.

The process creates the same original block if we decrypt the block twice.

decryptions, if the result is the

weak keys.

A weak key is the one which, after the parity drop operation consists of all 0s, all 1's or half 0's and half 1's.

The disadvantage of using a weak key is:
If we encrypt a block with a weak key and subsequently encrypt the result with the same weak key, we get the original block.

The process creates the same original block if we decrypt the block twice.

So, if after 2 decryptions, if the result is the same then, attacker is successful.

(iii) Semi weak keys → Six key pairs are called semi-weak keys.

(refer book pg 154)

A semi-weak key creates only two different round keys, and thus each of them is repeated 8 times. (show in book pg 155).

key → K_1 but
key → K_2 1

(iv) Possible weak keys → 48

There are (48) keys that are called possible weak keys.

A possible weak key is a key that creates only 4 distinct round keys, in other words, the 16 round keys are divided into 4 groups and each group is made of 4 equal keys.

(iv) Possible weak keys → 48

There are 48 keys that are called possible weak keys.

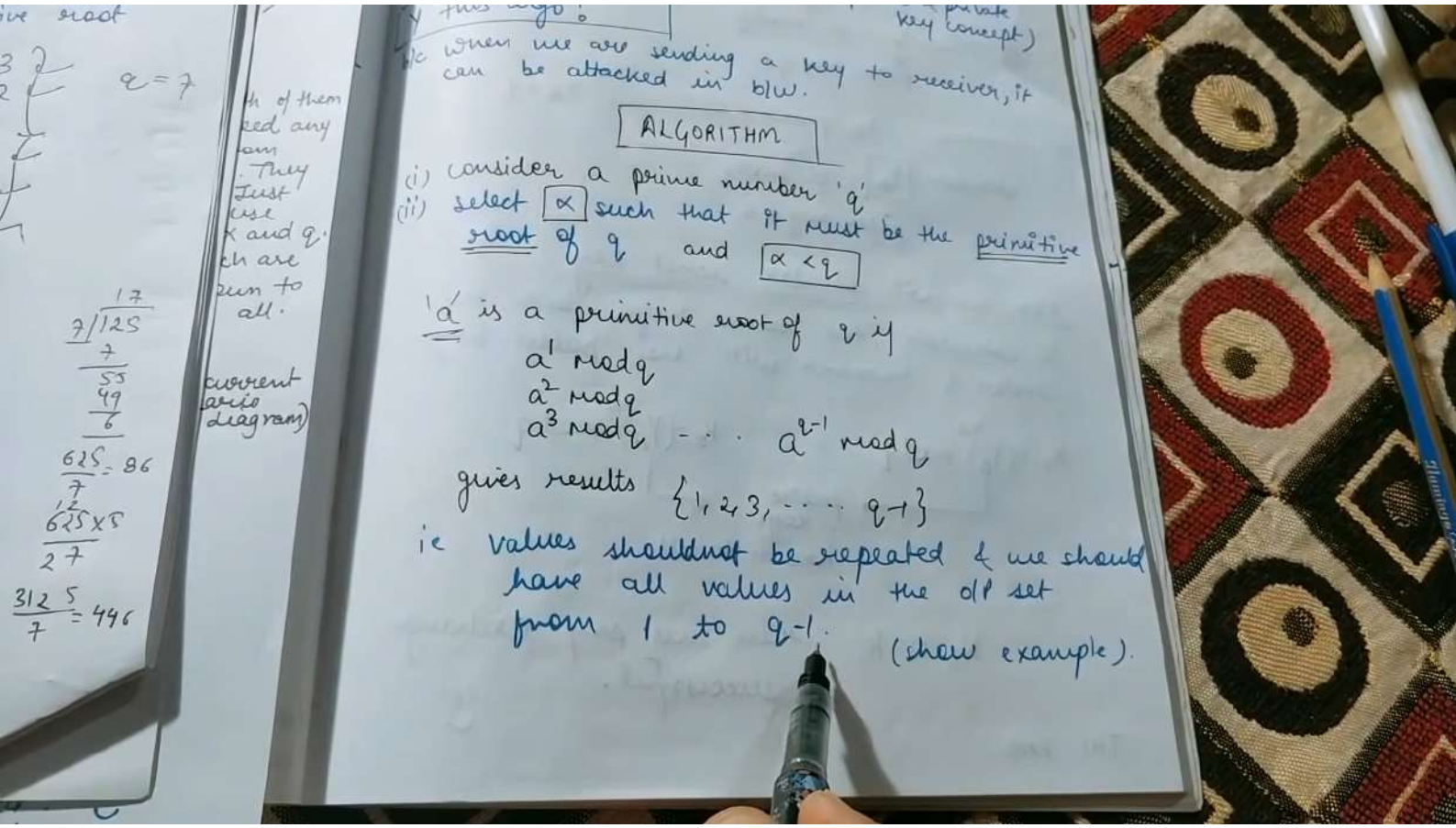
A possible weak key is a key that creates only 4 distinct round keys, in other words, the 16 round keys are divided into 4 groups and each group is made of 4 equal keys.

(v) Key clustering

means 2 or more dif keys can create the same ciphertext from the plain text.

Weakness in cipher Design

(i) Two specifically chosen i/p's to S-box array can create the same o/p.


root
q = 7
key concept)
can be attacked in blw.
ALGORITHM
(i) consider a prime number 'q'
(ii) select α such that it must be the primitive root of q and α < q
'a' is a primitive root of q if
a^1 mod q
a^2 mod q
a^3 mod q . . . a^(q-1) mod q
gives results {1, 2, 3, ... q-1}
ie values shouldnot be repeated & we should have all values in the o/p set from 1 to q-1.
(show example).

exchange algorithm

itive root

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

q = 7

Let q = 7 (prime)

∴ α < q i.e it is a primitive root

let α = 5 we can take any of the two primitive root 3 or 5.

α and q → global public elements (known to everyone)

Key generation of person 1

Assume private key $X_A = 3$ ∵ ($X_A < q$, 3 < 7 yes)

calculating public key $Y_A = \alpha^{X_A} \bmod q$

$Y_A = 5^3 \bmod 7 = 125 \bmod 7$

$Y_A = 7$

Key generation of person 2

Let private key $X_B = 4$ ($X_B < q$)

Both of them dont need any info from anyone. They just use α and q. which are known to all.

calculating public key $Y_B = \alpha^{X_B} \bmod q$

$Y_B = 5^4 \bmod 7 = 2$

$Y_B = 2$

(Show current scenario diagram)

Secret key calculation by person 2

X → private key of user
Y → public key of user

key) and $X_A < q$

of B) $X_B < q$

mod q

B

secret key

secret keys, both will use public

$K_2 = (Y_A)^{X_B} \bmod$

Diffie - Hellman Key exchange

X → private key of users
Y → public key of users

(3) assume X_A (private key of A) and $X_A < q$

calculate $Y_A = \alpha^{X_A} \bmod q$

↓ public key of A

(4) assume X_B (private of B) $X_B < q$

calculate $Y_B = \alpha^{X_B} \bmod q$

↓ public key of B

will calculate secret key

the secret keys, both the ...iver will use public keys.

$K = (Y_A)^{X_B}$

$\alpha < q$ i.e. it is a primitive root

let $\alpha = 5$

we can take any of the two primitive root 3 or 5.

α and q → global public elements (known to everyone)

Key generation of person1

Assume ~~private~~ key $X_A = 3$ ∵ ($X_A < q$, 3 < 7 yes)

calculating public key $Y_A = \alpha^{X_A} \bmod q$

$Y_A = 5^3 \bmod 7 = 125 \bmod 7$

$Y_A = 7$

Both dont ne... into f... anyon...

key generation of person2

let ~~private~~ key $X_B = 4$ ($X_B < q$)

calculating ~~private~~ public key $Y_B = \alpha^{X_B} \bmod q$

$Y_B = 5^4 \bmod 7 = 2$

$Y_B = 2$

Secret key calculation by Alice/person1

$K = (Y_B)^{X_A} \bmod q$

by person...

$K = (Y_A)^{X_B}$

abhishekdit

MULTIPLE DES

1. DOUBLE DES (2DES)
2. TRIPLE DES (3DES)

Since DES attack was vulnerable to brute force attack, variations of DES called multiple DES were introduced.

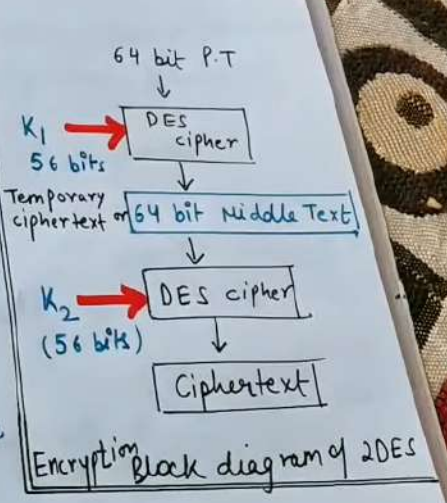
1. Double DES

→ uses 2 dif keys
(56+56) = 112 bit key

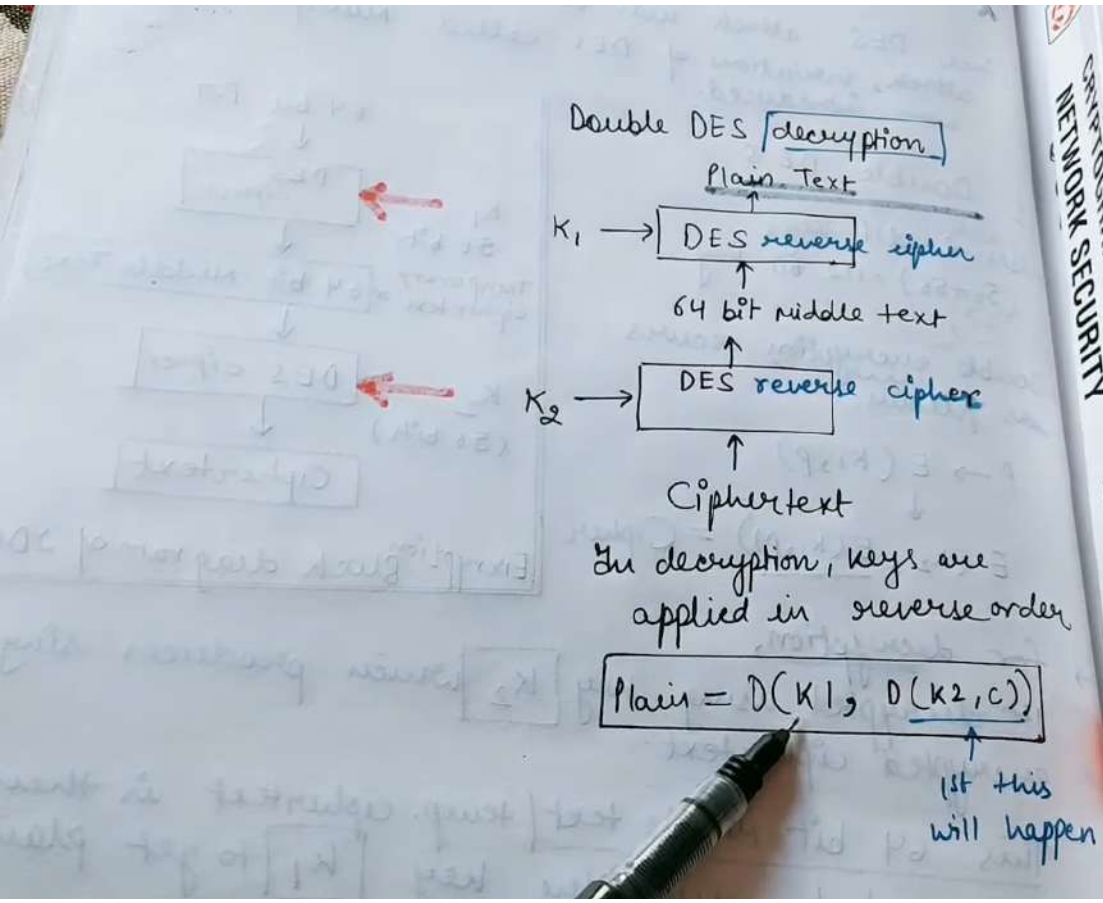
→ Double encryption occurs as follows:

$P \rightarrow E(K_1, P)$

$E(K_2, E(K_1, P)) = \text{Cipher}$



Encryption Block diagram of 2DES


Double DES decryption
Plain Text
K1
DES reverse cipher
64 bit middle text
K2
DES reverse cipher
Ciphertext
In decryption, keys are applied in reverse order
Plain = D(K1, D(K2, C))
1st this will happen


CRYPTOGRAPHY AND NETWORK SECURITY
Behrouz A Forouzan
Debdeep Mukhopadhyay
3e

Meet In the Middle Attack

... & ciphertext known

↓

decrypt pairs of all 2^{56} possible values of K_2

	Cipher	Middle	cipher
→			

in ~~table~~ table 1

Drawback of Double DES

Meet-in-the-middle attack.

This attack involves encryption from 1 end and decryption from the other end and then "matching the results in the middle" and hence the name.

Ques Explain drawback of 2DES.

Ques Explain attack on double DES.

Ques Explain Meet-in-the-middle attack. or MIM attack (at some places)

This attack requires knowing (some) plaintext/ciphertext pairs.

Let us assume plaintext = P, ciphertext = C

The attack proceeds as follows:

(i) encrypt 'P' for all 2^{56} possible values of K1 and store the results in a table and sort it.

Meet In the Middle Attack

some pairs of plaintext known & ciphertext known

↓ encrypt pairs for all 2^{56} possible values of K_1

↓ decrypt pairs of all 2^{56} possible values of K_2

no. of rows in Table = no. of possible secret keys

Plain	Cipher	Middle
ABCD	XYMP	
X$#T	MXMF	

Cipher	Middle	Cipher

SORT the results in table 1

drawback of Double DES

eet - in - the - middle attack

his attack involves encryption fr
decryption from the other end

" matching the results i
hence

Explain drawback of 2D
Explain attack on doubl
Explain Meet - in - the -

This attack requires kno
ciphertext pairs.
Let us assume plain

attack proceeds as
i) encrypt 'P' fo
and store the
sort it.

ii) Now, decrypt

decrypt pairs of all (256) possible values of K_2

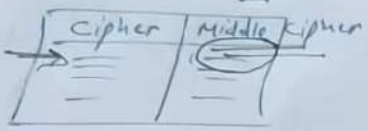


table 1

values

1st Table

Double DES

Meet-in-the-middle attack

This attack involves encryption from 1 end and decryption from the other end and then "matching the results in the middle" and hence the name.

Ques Explain drawback of 2DES.

Ques Explain attack on double DES.

Ques Explain Meet-in-the-middle attack or (at some places) MIM attack

This attack requires knowing (some) plaintext/ ciphertext pairs.

Let us assume plaintext = P, ciphertext = C

The attack proceeds as follows:

(i) encrypt (P) for all 2^{56} possible values of K1 and store the results in a table and sort it.

(ii) Now, decrypt (C) using all 2^{56} possible values of K2. As each result is produced (i.e decryption), check against the table for a match.

(iii) When there is a match, we have located a

decrypt pairs of all (256) possible values of K_2

Cipher	Middle Cipher

in ~~1st~~ table 1

ese values

1st Table

ing the results in the middle" and end and then hence the name.

Ques Explain drawback of 2DES.

Ques Explain attack on double DES.

Ques Explain Meet-in-the-middle attack or MIM attack (at some places)

This attack requires knowing (some) plaintext/ ciphertext pairs.

Let us assume plaintext = P, ciphertext = C

The attack proceeds as follows:

(i) encrypt (P) for all 2^{56} possible values of K1 and store the results in a table and sort it.

(ii) Now, decrypt (C) using all (2^{56}) possible values of K_2. (i.e. decryption) As each result is produced, check against the table for a match.

(iii) When there is a match, we have located a possibly correct pair of keys.

Note → Now, more than 1 pair of keys may result in a match, but these no. of pairs will be small We should try each possible pair of keys.

Eulers Theorem

→ also called Fermat - Euler theorem or Euler's toitient theorem

(a/c to wikipedia)

Euler's theorem states that if x and n are coprime positive integers, then

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

where $\phi(n) \rightarrow$ euler's toitient f^n.

Note → It is a generalized version of fermat's Theorem.

eg let $x = 11$, $n = 10$ both are coprime

∴ we can represent them as

$\phi(n)$

where $\phi(n)$ → euler's totient fn.

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

Note → It is a generalized version of fermat's Theorem.

eg let $x = 11$, $n = 10$ both are coprime

∴ we can represent them as

$11^{\phi(10)} \equiv 1 \bmod 10$

$11^4 \equiv 1 \bmod 10$

$14641 \equiv 1 \bmod 10$ which is true

$\phi(10) = \phi(2) * \phi(5)$

$= 1 * 4$

$= 4$

$\phi(a*b) = \phi(a) * (\phi(b)$

$\phi(n) = n-1$

$11^8 = 214,358,881$

Note →

$\bmod n$

$\phi(4) = 2$

$n = 4$

$\bmod 4$

$11^{\phi(10)} \equiv 1 \bmod 10$

$11^{4} \equiv 1 \bmod 10$

$14641 \equiv 1 \bmod 10$ which is true

$\phi(10) = \phi(2) * \phi(5)$

$= 1 * 4$

$= 4$

$\phi(a*b) = \phi(a) * (\phi(b)$

$\phi(4) = 3$

Note → $x^{\phi(n) \cdot a} \equiv 1 \bmod n$

$11^{8} = 214,358,881$

i.e $11^{4*2} \equiv 1 \bmod 10$

$11^{40} \equiv 1 \bmod 10$

i.e. any multiple of $\phi(n)$ will give the same result.

Euler's Toitient function

→ It is represented using phi as $\phi(n)$ and may also be called Euler's phi function.

→ Euler's toitient fn is defined as the no. of +ve integers less than n that are coprime to n.

$n \geq 1$

$\phi(5) = \{1, 2, 3, 4\}$

$\phi(6) = \{1, 5\}$

no. of elements in these sets is the toitient fn.

Note → Two integers a, b are said to be relatively prime, mutually prime or

ϕ is defined as the no. of +ve integers less than n that are coprime to n.

$n \geq 1$

$\phi(1) = 1$

$\phi(5) = \{1, 2, 3, 4\} = 4$

$\phi(6) = \{1, 5\} = 2$

① 2 3 4 5
✗ ✗ ✗ ✓

no. of elements in these sets is the totient fn.

Note → Two integers a, b are said to be relatively prime, mutually prime or coprime if the only +ve integer/factor that divides both of them is 1.

Now, when $n \rightarrow$ prime $\quad \phi(n) = n - 1$

eg $\phi(5) = 4$ // we have seen the eg. above

$\phi(23) = 23 - 1$

$= 22$

Now, when $n \rightarrow$ prime $\quad \phi(n) = n-1$

eg $\phi(5) = 4$ // we have seen the eg. above

$\phi(23) = 23-1$
$= 22$

Also, $\phi(a*b) = \phi(a) * \phi(b)$ // a and b should be coprime

eg $\phi(35) = \phi(7) * \phi(5)$
$= 6*4 = 24$

eg $\phi(12) = \phi(3) * \phi(4) = 2*2 = 4$

$\phi(15) = \phi(3) * \phi(5) = 2*4 = 8$

{1, 2, 4, [redacted], 7, 8, [redacted], 11, 13, 14}

n	$\phi(n)$	nos. coprime to n
1	1	1
2	1	1
3	2	1, 2
4	2	1, 3
5	4	1, 2, 3, 4
6	2	1, 5
7	6	1, 2, 3, 4, 5, 6
8	4	1, 3, 5, 7
9	6	1, 2, 4, 5, 7, 8
10	4	1, 3, 7, 9

Euler's
are

wher

Note →

eg

to n

Euler's Theorem

→ also called Fermat - Euler theorem or Euler's toitient theorem

(a/c to wikipedia)

Euler's theorem states that if x and n are coprime positive integers, then

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

where $\phi(n) \rightarrow$ euler's toitient f^n.

Note → It is a generalized version of fermat's Theorem.

eg let $x = 11$, $n = 10$ both are coprime

∴ we can represent them as

$$11^{\phi(10)} \equiv 1 \bmod 10$$

$x^{n-1} \equiv 1 \bmod n$, $x^n \equiv x \bmod n$

Ques verify fermats theorem for $x = 6$, $n = 7$

$x^{n-1} \equiv 1 \bmod n$

$6^{7-1} = 6^6 = 46656 \equiv 1 \bmod 7$

$x^n \equiv x \bmod n$

$6^7 = 279936$

$279936 \equiv 6 \pmod 7$

which is true.

not be modified

ld be safe

uldnot be
ryptanalysts.

(iii) # HASH FUNCTIONS

→ • Similar to MAC (msg. authentication code)
BUT it doesnot use a key.

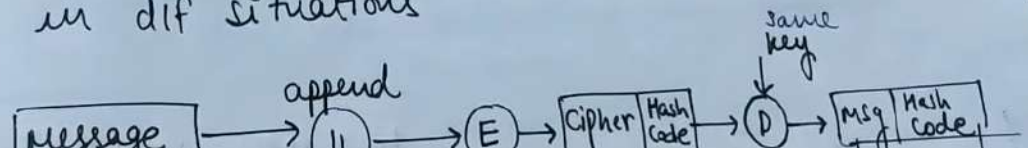
→ • Takes in variable size message and produce a fixed output.
↓
called Hash Code | Hash value | or Message Digest

→ The only i/p is the message.
→ A hash value 'h' is generated by a fn H

$H(M)$ = fixed length code 'h'
↑ variable length msg ↑ hashcode

They are also called Compression Functions.

There are dif methods to provide authentication in dif situations

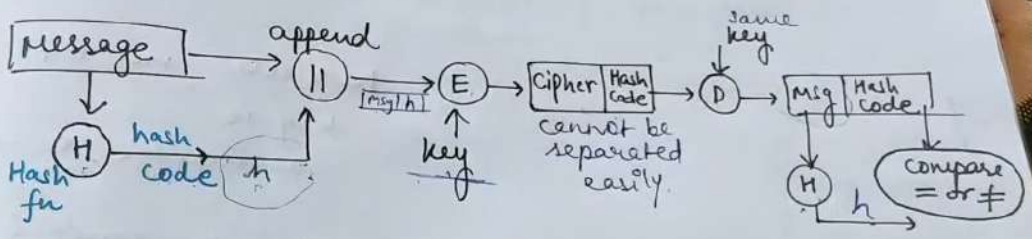
(i) 

They are also called Compression Functions.

Variable length msg → hash code

There are dif methods to provide authentication in dif situations

(i)



authentication + confidentiality

if both hashcodes equal in the end

Maintained b/c msg was encrypted before sending

b/c only A & B share the secret key, the msg must have come from A & has not been altered.

ION

5 principles of security.

thenticity of the msg is v imp.

must be there to authenticate

authentication

n

tion Code (MAC)

MAC (message authentication code)

→ We will use a secret key to generate a small fixed size (block) of data called MAC or cryptographic checksum.

→ It is then appended with the message.

→ The communicating parties will share a secret common key.
which will be used to create the MAC

Let A → sender
B → receiver

When A sends a msg to B, it calculates the MAC as a fn of the message and the key.

$$MAC = C(K, M)$$

where
M = input message

AUTHENTICATION

→ It is one of the 5 principles of security.
→ verifying the authenticity of the msg is v.imp.
→ An authenticator must be there to authenticate the message.

Methods to produce authentication

1. Message encryption
2. Message Authentication Code (MAC)
3. Hash functions

MAC (Message au

→ We will use a sec
blo
small fixed size

→ It is then appe
→ The communicati
secret common
wi

Let A → send
B → rec

When A send
MAC as a

be there to authenticate

thentication

ipher text

Code (MAC)

value

small fixed size of data called MAC or cryptographic checksum.

→ It is then appended with the message.

→ The communicating parties will share a secret common key.

2MB → 2Kb which will be used to create the MAC

Let A → sender
B → receiver

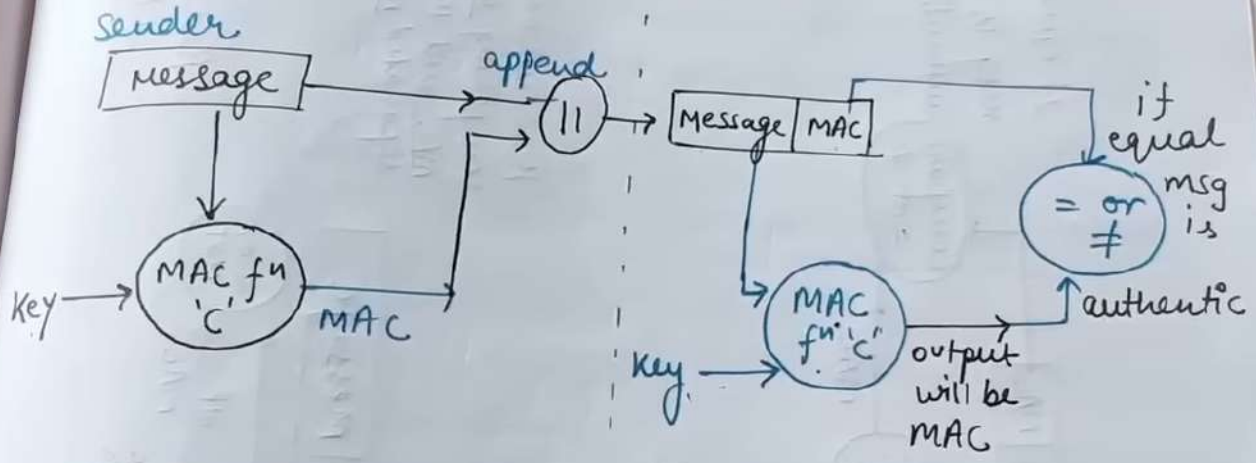
When A sends a msg to B, it calculates the MAC as a fn of the message and the key.

$$MAC = C(K, M)$$

where

M = input message
C = MAC function
K = shared secret key

(i) MAC - for authentication



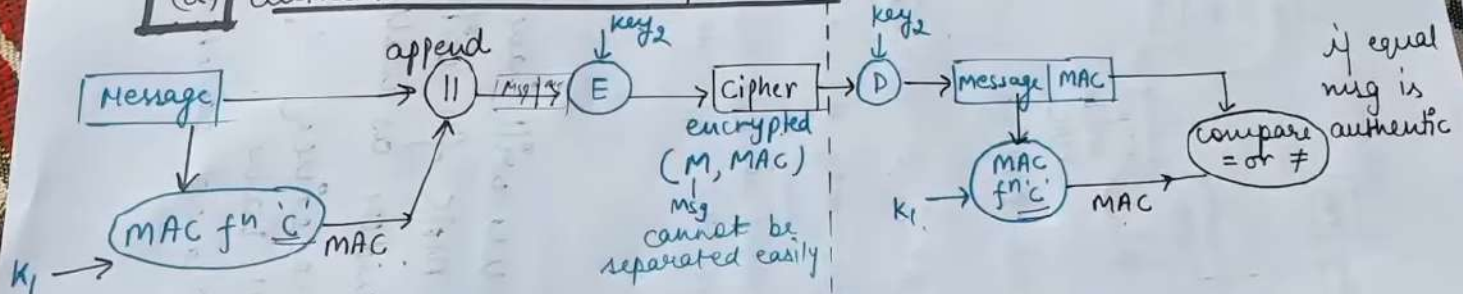
we separate the msg and the MAC

only authentication is achieved

NO

2) MAC - for authentication & confidentiality

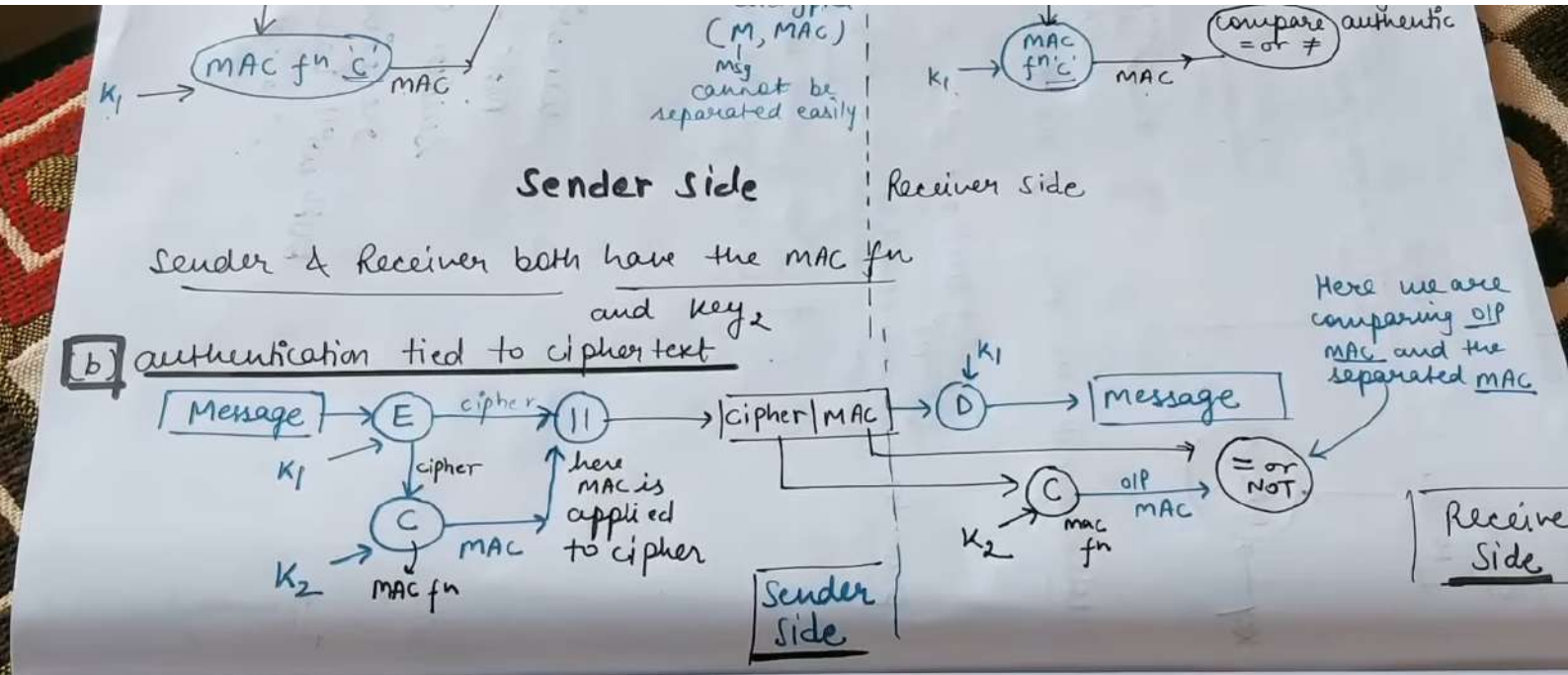
(a) authentication tied to plain text



Sender Side | Receiver side

Sender & Receiver both have the MAC fn and key2

Here we are comparing o/p MAC and the



How is confidentiality achieved?

If hacker attacks, he will get the

(ii) authen

has be

Algorithm

Rivest - Shamir - Adleman developed in 1978

→ It is an Asymmetric cryptographic algo. (2 keys) ie public and private key concept is used here.

→ The acronym RSA is made from the initial letters of the surnames of Ron Rivest, Adi Shamir, and Leonard Adleman.

→ Public key → known to all users in N/w
Private key → kept secret, not shareable to all.

If public key of user A is used for encryption, we have to use the private key of same user for decryption.

The RSA scheme is a block cipher in which the plain text and ciphertext are integers blw 0 and n-1 for some value n.

1. Key Generation

(i) select 2 large prime nos 'p' and 'q' → for higher security.

(ii) calculate $n = p * q$

(iii) calculate $\phi(n) = (p-1) * (q-1)$ // eulers Toitient fn

(iv) choose value of e

$1 < e < \phi(n)$ and $\gcd(\phi(n), e) = 1$

(v) calculate

$d = e^{-1} \bmod \phi(n)$

ie $ed = 1 \bmod \phi(n)$

(vi) public key = $\{e, n\}$

(vii) private key = $\{d, n\}$

2. Encryption

Plaintext = $M < n$ IMP

// c → ciphertext

Let $p = 3, q = 11$

$n = p * q = 3 * 11 = 33$

$\phi(n) = 2 * 10 = 20$ $\quad \because \phi(n) = (p-1)(q-1)$

So, let $\boxed{e = 7}$ as $1 < 7 < 20$ and $\gcd(7, 20) = 1$

Now, $d \equiv e^{-1} \bmod \phi(n)$

$de \equiv 1 \bmod \phi(n) \rightarrow de \bmod \phi(n) = 1$

$7 * d \equiv 1 \bmod \phi(n)$

$(7*d) \bmod 20 = 1$ $\quad (\because d = 3)$

↓ multiplicative inverse of 7

ind multiples of $\phi(n)$ ie here 20, and just no. satisfying a value greater this ie $(7*d)$ should be 21.

using extended euclidean

↓ in next video (I will use this method).

Key Generation

for higher security.

select 2 large prime nos 'p' and 'q'

calculate $n = p * q$

calculate $\phi(n) = (p-1) * (q-1)$ // Eulers Toitient fn

choose value of e

$1 < e < \phi(n)$ and $\gcd(\phi(n), e) = 1$

calculate

$d \equiv e^{-1} \bmod \phi(n)$

ie $ed \equiv 1 \bmod \phi(n) \rightarrow ed \bmod \phi(n) = 1$

public key = $\{e, n\}$

private key = $\{d, n\}$

Encryption

Plaintext = M < n (M)

$C = M^e \bmod n$ // C → ciphertext

Decryption

$M = C^d \bmod n$

used in ency

$de \equiv 1 \bmod \phi(n) \rightarrow de \bmod \phi(n) = 1$

$7 * d \equiv 1 \bmod \phi(n)$

20 → 21
40 41
60 61

$(7*d) \bmod 20 = 1 \quad (\because d = 3)$

↓ multiplicative inverse of 7

// find multiples of $\phi(n)$ ie here 20, and just find a no. satisfying a value greater than this ie $(7*d)$ should be 21.

We can solve it using extended euclidean algorithm also

↓ in next video (I will use this method).

since $e = 7, d = 3$

public key = $\{e, n\} = \{7, 33\}$

private key = $\{d, n\} = \{3, 33\}$

ENCRYPTION

$C = M^e \bmod n$

Let $M = 31$

$C = 31^7 \bmod 33 = 4 \rightarrow \boxed{C = 4}$

DECRYPTION

$M = C^d \bmod n = 4^3 \bmod 33 = 31$

$\boxed{M = 31}$

select 2 large prime nos p and q (for higher security)

calculate $n = p * q$

calculate $\phi(n) = (p-1) * (q-1)$ // Euler Toitient fn

choose value of e

$1 < e < \phi(n)$ and $\gcd(\phi(n), e) = 1$

calculate

$d \equiv e^{-1} \bmod \phi(n)$

ie $ed \equiv 1 \bmod \phi(n) \rightarrow ed \bmod \phi(n) = 1$

public key = $\{e, n\}$

private key = $\{d, n\}$

2. Encryption

Plaintext = M ($< n$) IMP

$C = M^e \bmod n$

// C → ciphertext

3. Decryption

$M = C^d \bmod n$

Note → (e, n) is public key used in encryp

(d, n) → private key used for decrypti

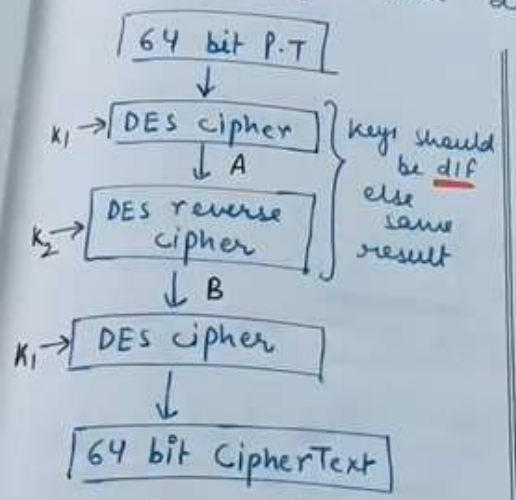
g to break
force.

security.

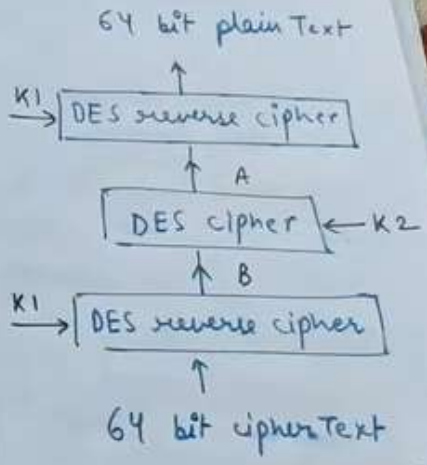
TRIPLE DES

i.e. 3DES

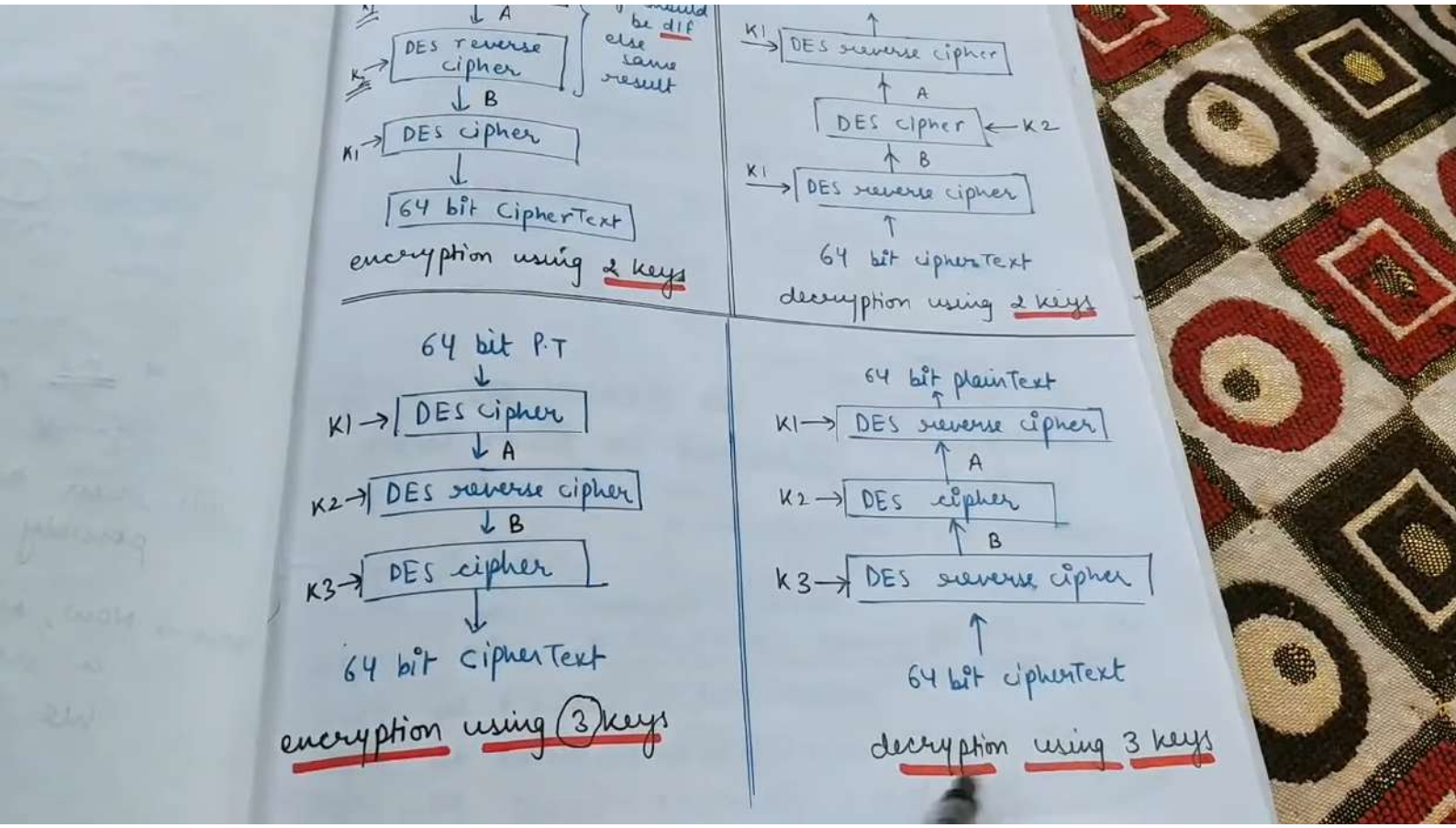
→ 2 or 3 keys are used.
→ much stronger than double DES.



encryption using 2 keys



decryption using 2 keys


A
DES reverse cipher
K2
B
K1
DES cipher
64 bit CipherText
encryption using 2 keys
be dif
else same result
K1
DES reverse cipher
A
DES cipher
K2
B
K1
DES reverse cipher
64 bit cipherText
decryption using 2 keys
64 bit P.T
K1
DES cipher
A
K2
DES reverse cipher
B
K3
DES cipher
64 bit CipherText
encryption using (3) keys
64 bit plainText
K1
DES reverse cipher
A
K2
DES cipher
B
K3
DES reverse cipher
64 bit cipherText
decryption using 3 keys