

SO WHAT IS CAESAR CIPHER?

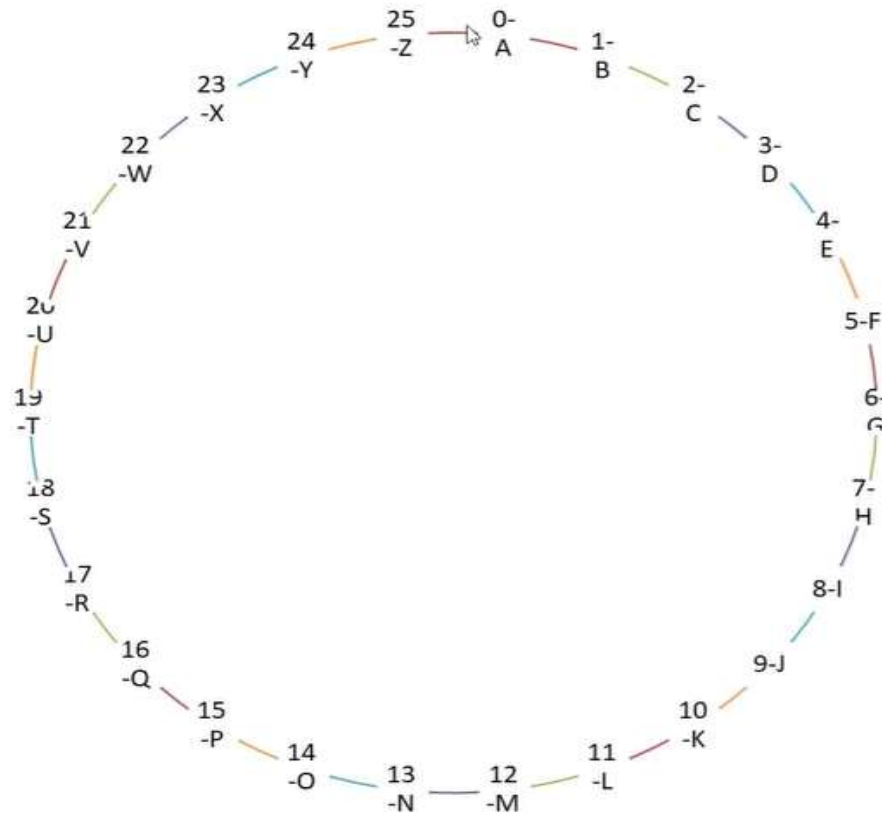
- Caesar Cipher is an encryption technique in which we replace each letter with a shift of a fixed number of letters, traditionally 3.
- For example A will be written as D, B will be written as E, C will be written as F and so on.

WHAT ARE THE REQUIREMENTS?

- To perform Caesar Cipher encryption you need 2 things.
- First thing is a reference table of alphabets and their numerical equivalent.
- Second the value of the shift Key.

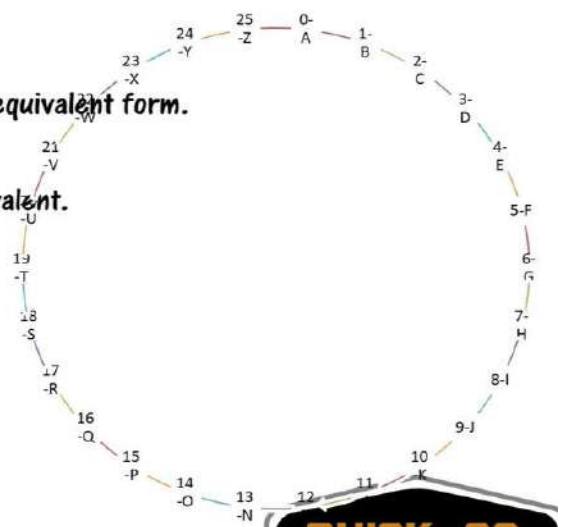
0=A	1=B	2=C	3=D	4=E
5=F	6=H	7=H	8=I	9=J
10=K	11=L	12=M	13=N	14=O
15=P	16=Q	17=R	18=S	19=T
20=U	21=V	22=W	23=X	24=Y
		25=Z		

CYCLE OF LETTERS

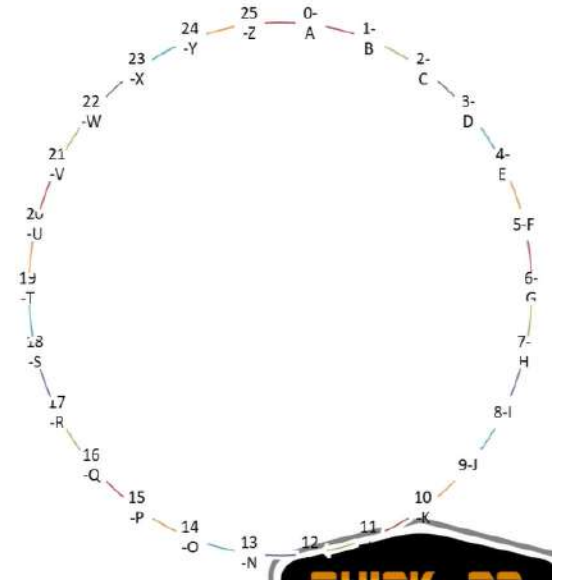


ENCRYPTION

- Now lets encrypt the word ZEBRA.
- To encrypt, first convert the Plaintext into its numerical equivalent form.
- So Z=25, E=4, B=1, R=17, A=0.
- Now add the key value=3 to each letter's numerical equivalent.
- So Z=(25+3)=28.
- 28=C
- Hence letter Z after Encryption will become C
- Similarly, E=(4+3)=7
- 7=H
- Hence letter E after encryption will become H.

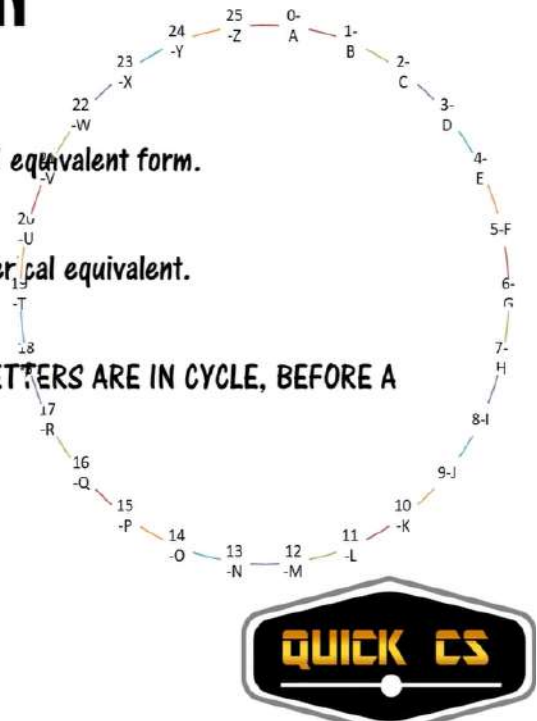


- Similarly $B = (1+3) = 4$
- $4 = E$,
- Therefore $P(B) = C(E)$
- $R = (17+3) = 20$
- $20 = U$
- Therefore $R = U$
- Finally $A = (0+3) = 3 = D$
- Hence $A = D$
- SO the encrypted text of ZEBRA = CHEUD



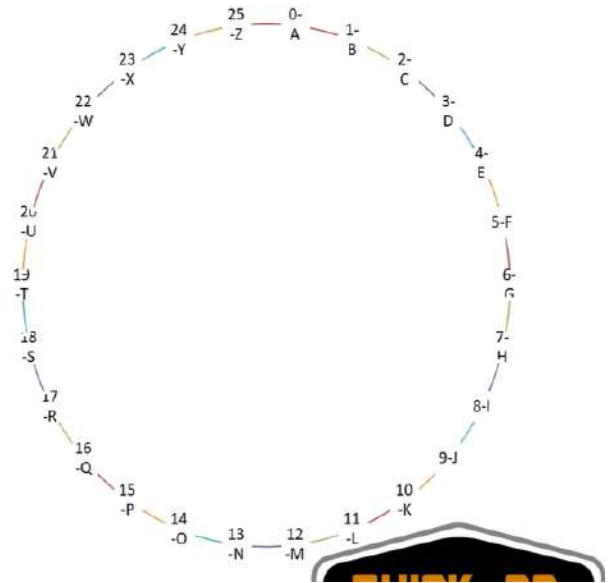
DECRYPTION

- Now let's decrypt the word CHEUD.
- To Decrypt, first convert the Cipher-text into its numerical equivalent form.
- So $C=2$, $H=7$, $E=4$, $U=20$, $D=3$.
- Now SUBTRACT the key value=3 FROM each letter's numerical equivalent.
- So $C = (2-3) = -1$.
- -1 MEANS GO 1 PLACE BEHIND FROM ZERO, SINCE ALL LETTERS ARE IN CYCLE, BEFORE A COMES Z
- $-1 = Z$
- Hence letter C after Decryption will become Z
- Similarly, $H = (7-3) = 4$
- $4 = E$
- Hence letter H after Decryption will become E.



DECRYPTION

- Similarly $E = (4 - 3) = 1$
- $I = B$,
- Therefore $C(E) = P(B)$
- $U = (20 - 3) = 17$
- $17 = R$
- Therefore $U = R$
- Finally $D = (3 - 3) = 0 = A$
- Hence $D = A$
- SO the Decrypted text of CHEUD = ZEBRA



Playfair Cipher

- It is aka Playfair square or Wheatstone-Playfair cipher.
- It is manual symmetric encryption technique.
- It is the first literal diagram or diagraph substitution cipher.
- Invented in 1847 by Charles Wheatstone.
- It is polyalphabetic cipher.
- It needs 5*5 matrix constructed using **a keyword** (Ex: MONARCHY).
- If the keyword has repeating letter, we should not fill more than once.
- We have to always ignore repeating letters.

Playfair Cipher -Algorithm Step 1

- **Generate the key Square (5x5):** The key square is a 5x5 grid of alphabets that acts as the key for encrypting the plaintext.
- Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets).
- The letter I and J will be consider as one letter.so If I is already placed then no need to place J in rest of matrix. (Or write I/J)
- The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.

For example,

Keyword: MONARCHY

M	O	N	A	R
C	H	Y		



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

Playfair Cipher -Algorithm Step 2

- **Algorithm to encrypt the plain text:** The plaintext is split into pairs of two letters also known as digraphs.
- If there is an odd number of letters, a Z is added to the last letter.

For example:

• **Plain Text:** “instruments” **After Split:** ‘in’ ‘st’ ‘ru’ ‘me’ ‘nt’ ‘sz’

- Pair cannot be made with same letter. Break the letter in single and add a bogus letter to the previous letter.

• **Plain Text:** "hello" **After Split:** ‘he’ ‘lx’ ‘lo’

- Here 'x' is the bogus letter.

Playfair cipher- algorithms rules for encryption

1. **If the both letters are in the same column:** take the letter below each one (going back to the top if at the bottom).

For example:

Diagraph: “ME”

Encrypted text = “CL”

Encryption: M → C and E → L

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

2. **If the both letters are in the same row:**

take the letter to the right of each one (going back to the leftmost if at the rightmost position).

For example:

Diagraph: “ST”

Encrypted text = “TL”

Encryption: $S \rightarrow T$ and $T \rightarrow L$

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

3. If neither of the above rules is true (If the two letters are in different rows and different columns): Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

For example:

Diagraph: “NT”

Encrypted text = “RQ”

Encryption: $N \rightarrow R$ and $T \rightarrow Q$

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

Solve example

Plain Text: “instruments”

In = GA

St=TL

Ru=ZM

Me=CL

Nt=RQ

Sz= TX

in:

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

st:

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

ru:

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

me:

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

nt:

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

sz:

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

Cipher Text or Encrypted Text= GATLZMCLRQTX

Playfair cipher- algorithms rules for Decryption

- If the both letters are in the same column:** take the letter above each one (going back to the bottom if at the top).

For example:

Diagraph: “CL”

Decrypted text = “ME”

Encryption: C → M and L → E

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

- If the both letters are in the same row:** take the letter to the left of each one (going back to the rightmost if at the leftmost position).

For example:

Diagraph: "TL"

Decrypted text = "ST"

Encryption: $T \rightarrow S$ and $L \rightarrow T$

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

3. If neither of the above rules is true (If the two letters are in different rows and different columns): Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

Example:

Diagraph: "RQ"

Decrypted text = "NT"

Encryption: $R \rightarrow N$ and $Q \rightarrow T$

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

Solve example

Cipher or encrypted Text: “GATLZMCLRQTX”

in:

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

st:

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

ru:

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

me:

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

nt:

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

sz:

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

GA= In

TL= St

ZM= Ru

CL= Me

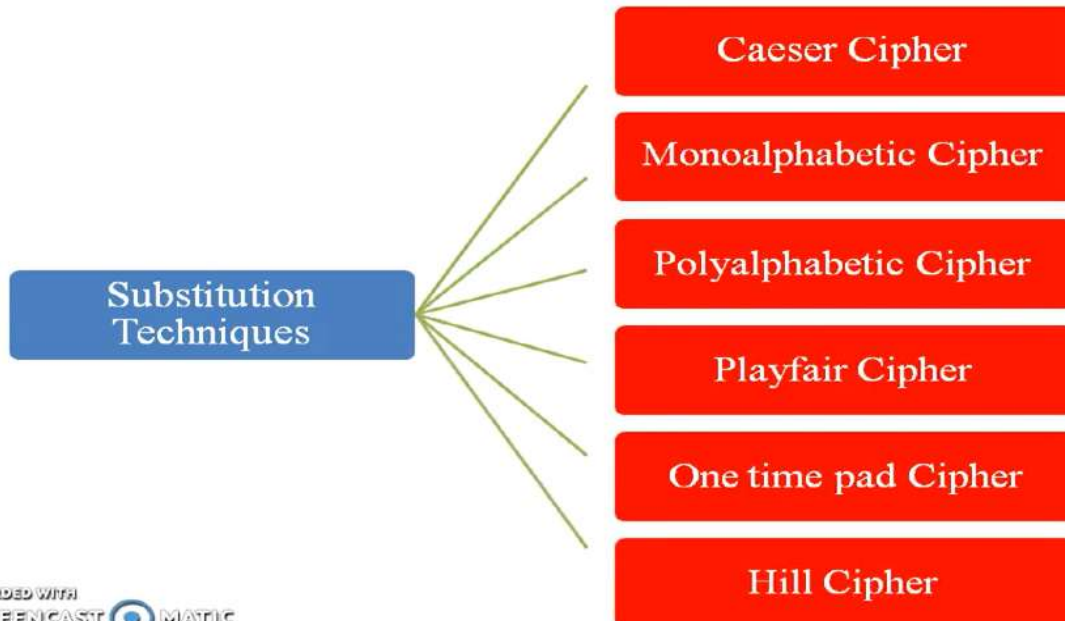
RQ= Nt

TX= Sz

So, Plain Text= instruments

Substitution Cipher

❑ Types of substitution techniques:

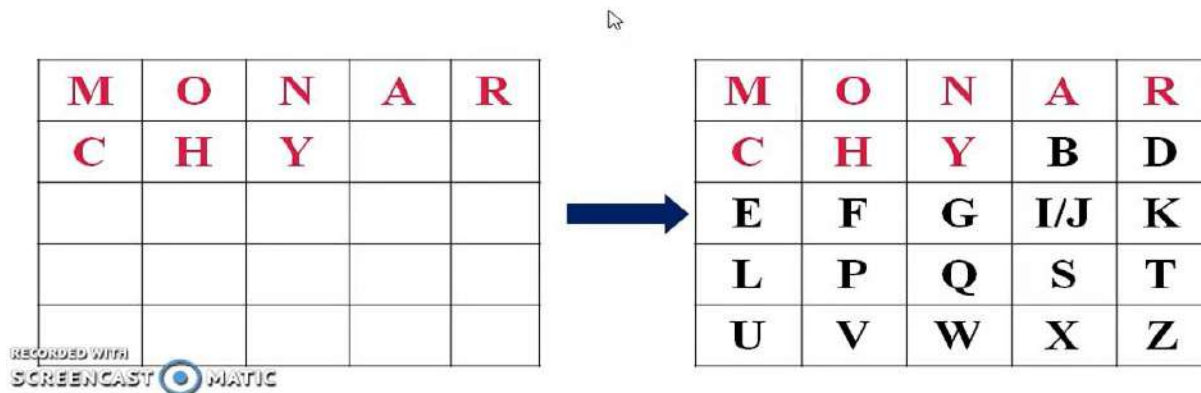


Playfair Cipher

- In this cipher techniques, more than one character are used during encryption/decryption.
- So, this cipher technique is called multiple substitution cipher technique.
- Input of this technique are keyword and plain text.
- **Keyword** is one type of string.
- Rules of encryption is as follows:

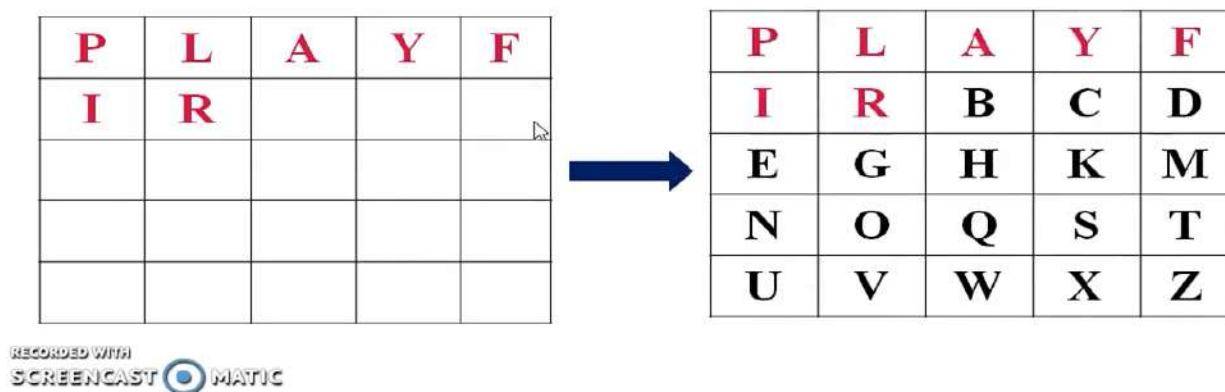
Playfair Cipher

- Construct 5x5 matrix of given keyword(key). The letter I and J will be consider as one letter. So if I is already placed then no need to place J in rest of matrix. (or write I/J).
- For example,
Keyword: MONARCHY



Playfair Cipher

- If any characters are to be repeated in given keyword, used it only once during matrix filling.
- For example,
Keyword: PLAYFAIR



Playfair Cipher

- Plain text message that we want to encrypt broken down into groups of two alphabets. (Make pair of two alphabets of given plain text)

- For example,

Plaintext: MY NAME IS MAHI

MY

NA

ME

IS

MA

HI

RECORDED WITH

Playfair Cipher

- If both alphabets are the same (or only one is left) in plain text, add an X after the first alphabet. Encrypt the new pair and continue.

- **For example – 1**

Plaintext: GREEN

GR

EX

EN

- **For example – 2**

Plaintext: NETWORK

NE

TW

OR

KX

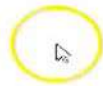
RECORDED WITH

Playfair Cipher

- If the alphabets are not in the same row or column, replace them with the alphabets in the same row respectively, but at the other pair of corners of the rectangle defined by the original pair.

- For example,
Keyword: PLAYFAIR
Plaintext: OC

O C
 ↓ ↓
 S R



Cipher Text: SR

RECORDED WITH

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

Playfair Cipher

- If both the alphabets in the pair appear in the same row of matrix, replace them with alphabets to their immediate **right** respectively.
- ✓ If the original pair is on the right side of the row, then **wrapping** around to the **left** side of the row happens.

- For example,
Keyword: PLAYFAIR
Plaintext: GM

G M
 ↓ ↓
 H E

Cipher Text: HE

RECORDED WITH
 SCREENCASTOMATIC

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

Playfair Cipher

Monoalphabetic Cipher (Net

- If both alphabets in the pair appear in the same column of matrix, replace them with alphabets immediate below from respectively.
- ✓ If the original pair is on the bottom side of the column, then wrapping around to the top side of the column happens.

- For example,
Keyword: PLAYFAIR
Plaintext: IU

I U
 ↓ ↓
 E P

Cipher Text: EP

RECORDED WITH

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

Playfair Cipher

Monoalphabetic Cipher (Net

- If both alphabets in the pair appear in the same column of matrix, replace them with alphabets immediate below from respectively.
- ✓ If the original pair is on the bottom side of the column, then wrapping around to the top side of the column happens.

- For example,
Keyword: PLAYFAIR
Plaintext: IU

I U
 ↓ ↓
 E P

Cipher Text: EP

RECORDED WITH

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

Example

❖ Find out cipher text of below plain text using “Playfair Cipher”.

Plaintext: TREE IS GREEN, Keyword: ENVIRONMENT

Solution:

Plaintext: TREE IS GREEN

TR EX EI SG RE EN

TR → BV
EX → VU
EI → NR
SG → QH
RE → EN
EN → NV

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

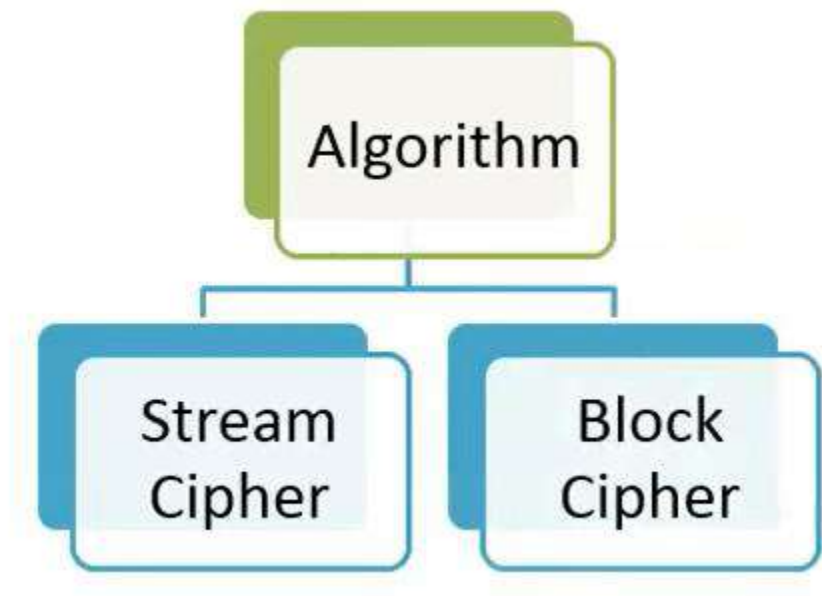
→

RECORDED WITH
ANALOGY

Cipher text: BVVUNRQHNV

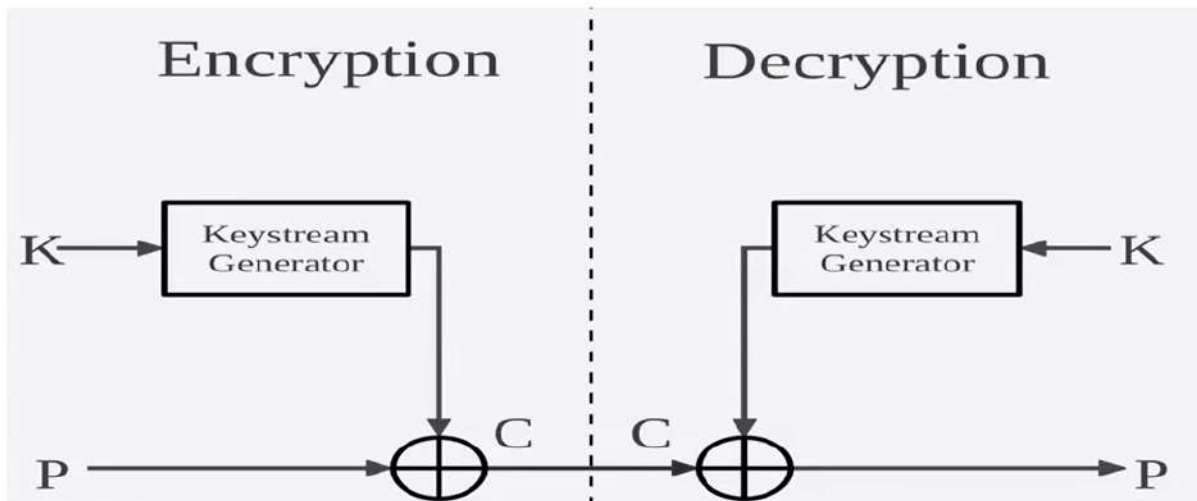
Cryptographic Algorithm

- Basically, a cryptographic algorithm is used for the transformation of plaintext into ciphertext.
- There are basically two methods on which cryptographic algorithms work:



Stream Cipher

- In stream cipher, keys and algorithms are applied to each binary digit in a data stream, one bit at a time, rather than encrypting a block of data.
- A stream cipher operates on plaintext accepting only one bit at a time.
- A stream cipher in which each bit of plain text message XORs with each bit of key to obtain a cipher text message.



- There is a key stream generator which outputs a stream of bits: $k_1, k_2, k_3, \dots, k_i$ and XORED with a stream of plaintext bits $p_1, p_2, \dots, p_i$ to produce the stream of cipher text bits.

$$C_i = P_i \oplus K_i$$

- During decryption, the cipher text bits are XORED with the same key stream to recover the plain text bits.

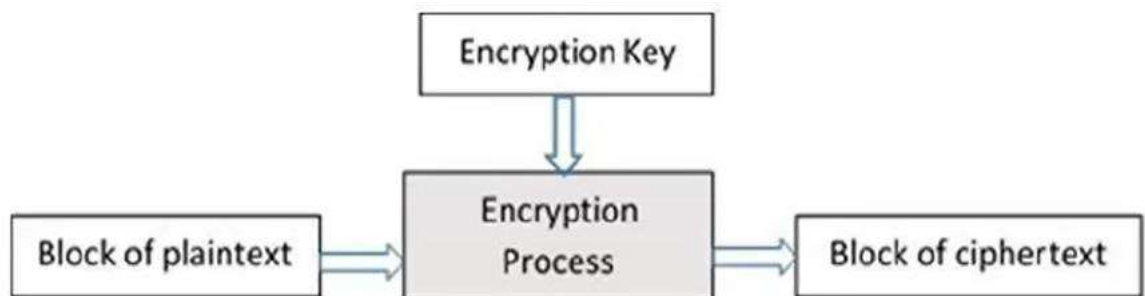
$$P_i = C_i \oplus K_i$$

❖ list of common encryption algorithms categorized under stream ciphers

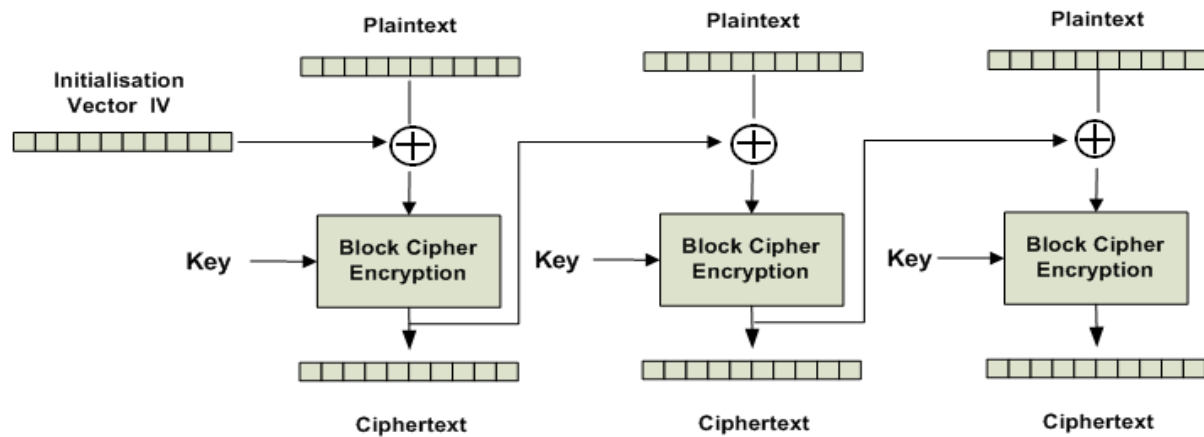
- RC4 (Rivest Cipher 4)
- Salsa20
- ChaCha20
- A5/1 and A5/2
- Trivium
- Grain

Block Cipher

- In Block cipher, plain text message is divided into fixed-size blocks and each block is encrypted with some fixed-size key.
- Divide each plain text message into blocks of 64, 128, 256 bits and apply key sizes 40, 56, 64, 80, 128, 168, 192 and 256 bits which generates cipher text blocks same as the size of plain text blocks.
- **Data Encryption Standard (DES)** is the best example of block cipher in which each block of 64-bit gets encrypted using 56-key bit and cipher text of 64-bit gets generated.



- A receiver-side decrypts messages with the same key to generate plaintext.
- Block cipher also uses the concept of a key generator.
- Block ciphers use chaining mode; this is because for repeating text patterns, the same cipher block will be generated which can give a clue to cryptanalysts regarding what the original plaintext is.
- As a chaining mode, the previous block is mixed with the current block to avoid repeats in patterns. It is more secure.

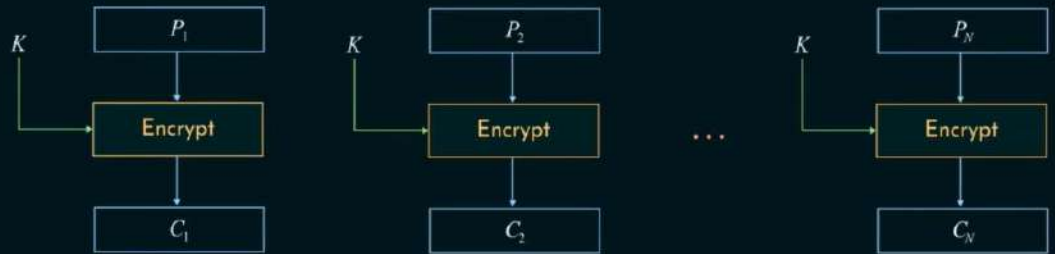


❖ list of common encryption algorithms categorized under block ciphers

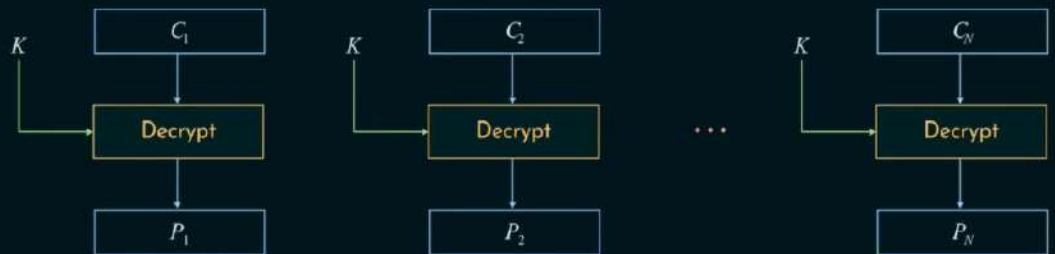
- AES (Advanced Encryption Standard)
- DES (Data Encryption Standard):
- 3DES (Triple DES)
- Blowfish
- Twofish
- Serpent
- IDEA (International Data Encryption Algorithm)

❖ Assignment of ECB and CBC

Electronic Codebook (ECB)

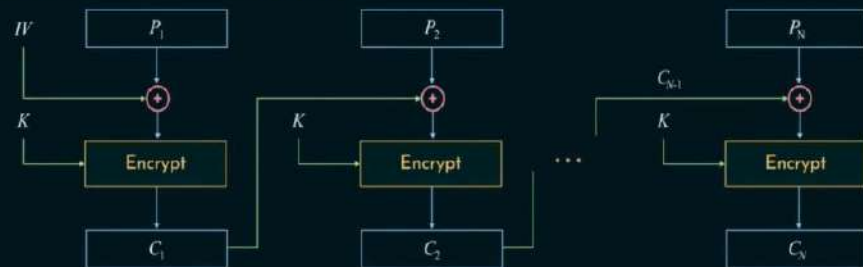


ECB - Encryption

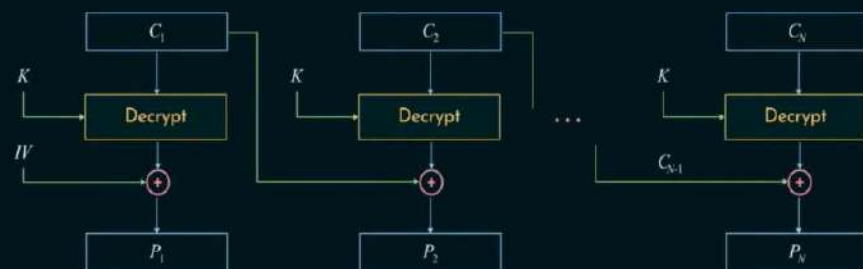


ECB - Decryption

Cipher Block Chaining (CBC)



CBC - Encryption



CBC - Decryption

Difference between block and stream cipher

No.	Stream Cipher	Block Cipher
1	Key and algorithm applied on each binary digit.	Key and algorithm applied on block of data.
2	Less time consuming in compare to block cipher.	More time consuming in compare to stream cipher.
3	Only one bit encrypting at a time, it is faster than block cipher.	Block of data is encrypting at a time block cipher is slower than stream cipher.
4	It doesn't used chaining mode.	It uses chaining mode.
5	Hardware implementation is easy using stream cipher.	Software implementation is easy using block cipher.
6	One time pad is the best example of stream cipher.	DES is the best example of block cipher.
7	Less Secure in compare to Block cipher.	More secure in compare to stream cipher.

FEISTEL STRUCTURE

What is the Design principle of Block cipher?

- Most of the block cipher technique will follow a common structure i.e., FEISTEL STRUCTURE
- **The Feistel structure**, also known as the Feistel network, is a symmetric structure used in the design of many block ciphers.
- Feistel designed a scheme for a block cipher that **alternates** between **permutation** (rearranging bits) and **substitution** (replacing bits).
- **Feistel cipher structure** encrypts plain text in several rounds, where it applies **substitution** and **permutation** to the data. Each round uses a different key for encryption, and that same key is used for the decryption process.

Encryption Inputs:

- **Inputs include:**
 - ✓ A plaintext block of length **$2w$ bits**.
 - ✓ A key **K_i** .
- **The plaintext block is split into two halves:**
 - ✓ **LE_0** (Left half)
 - ✓ **RE_0** (Right half)
- It was named after Horst Feistel, a cryptographer at IBM. Instead of encrypting the entire data block in one go, the Feistel structure divides the

block into two halves and processes them through multiple rounds. In each round, a round function is applied to one half, and the result is then XORed with the other half.

- The two halves are then swapped and the process repeats for a fixed number of rounds.

Here's a breakdown of the Feistel structure:

Encryption Process:

1. Divide: The plaintext block is divided into two equal halves, a left half (L_0) and a right half (R_0).

2. Rounds: For each round i from 1 to n (where n is the number of rounds):

- A round function F takes the right half of the previous round (R_{i-1}) and a subkey K_i as input.
- The output of the round function is XORed with the left half of the previous round (L_{i-1}). This becomes the new right half (R_i).
- The right half of the previous round (R_{i-1}) becomes the new left half (L_i). Mathematically:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

3. Combine: After the final round, the left half (L_n) and the right half (R_n) are concatenated to form the ciphertext.

Note that in some implementations, the halves are swapped after the last round.

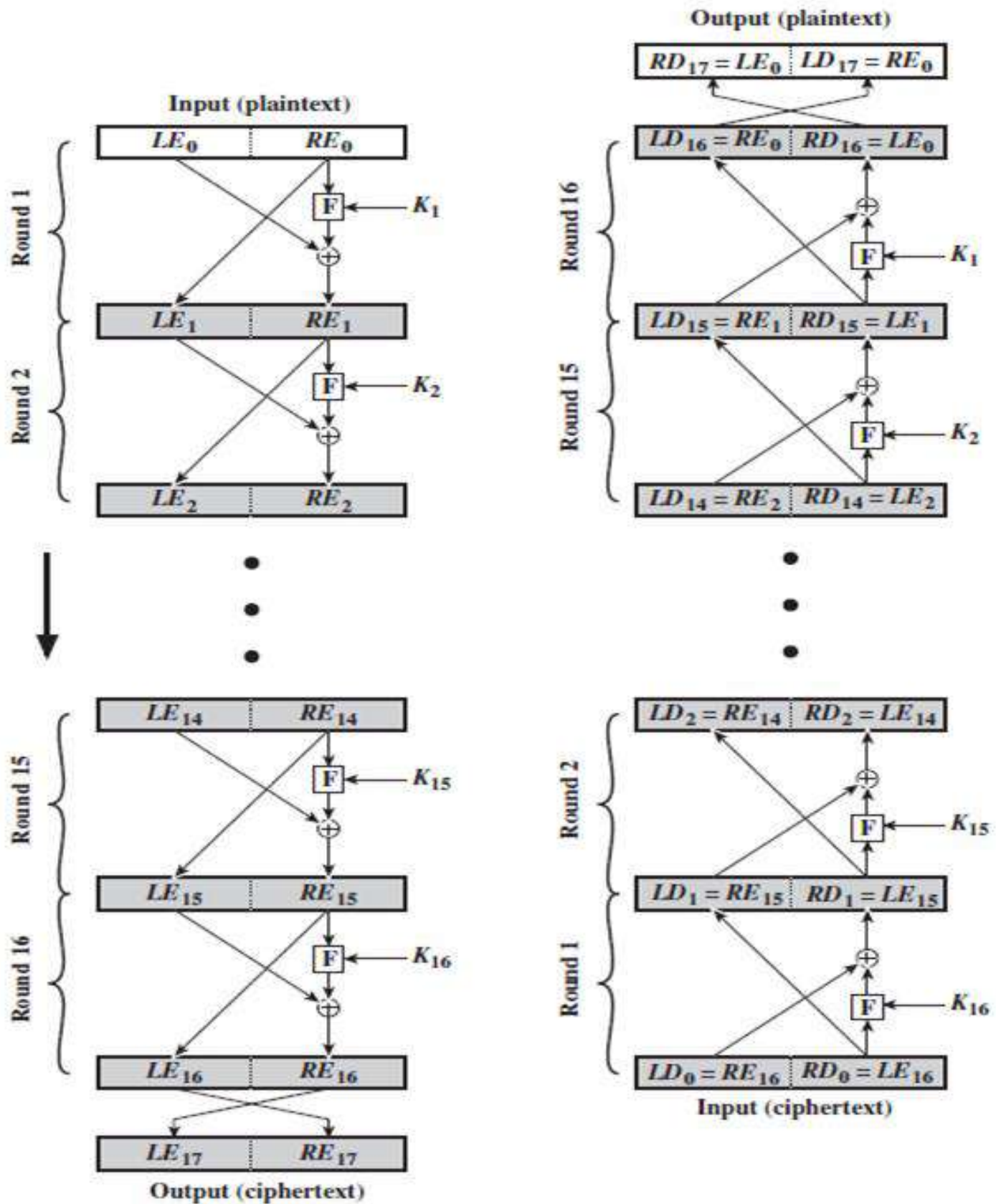
Working of Feistel Cipher Structure

- A **substitution** is performed on the left half of the data. This is done by applying a round function F to the right half of the data and then taking the exclusive-OR of the output of that function and the left half of the data.
- The round function has the same general structure for each round but is parameterized by the round subkey K_i .
- **Permutation** is performed that consists of the interchange of the two halves of the data. This structure is a particular form of the *substitution-permutation network (SPN) proposed by Shannon*.

Decryption Process:

The decryption process in a Feistel cipher is remarkably similar to the encryption process. The same round function F is used, but the subkeys are applied in reverse order ($K_n, K_{n-1}, \dots, K_1$).

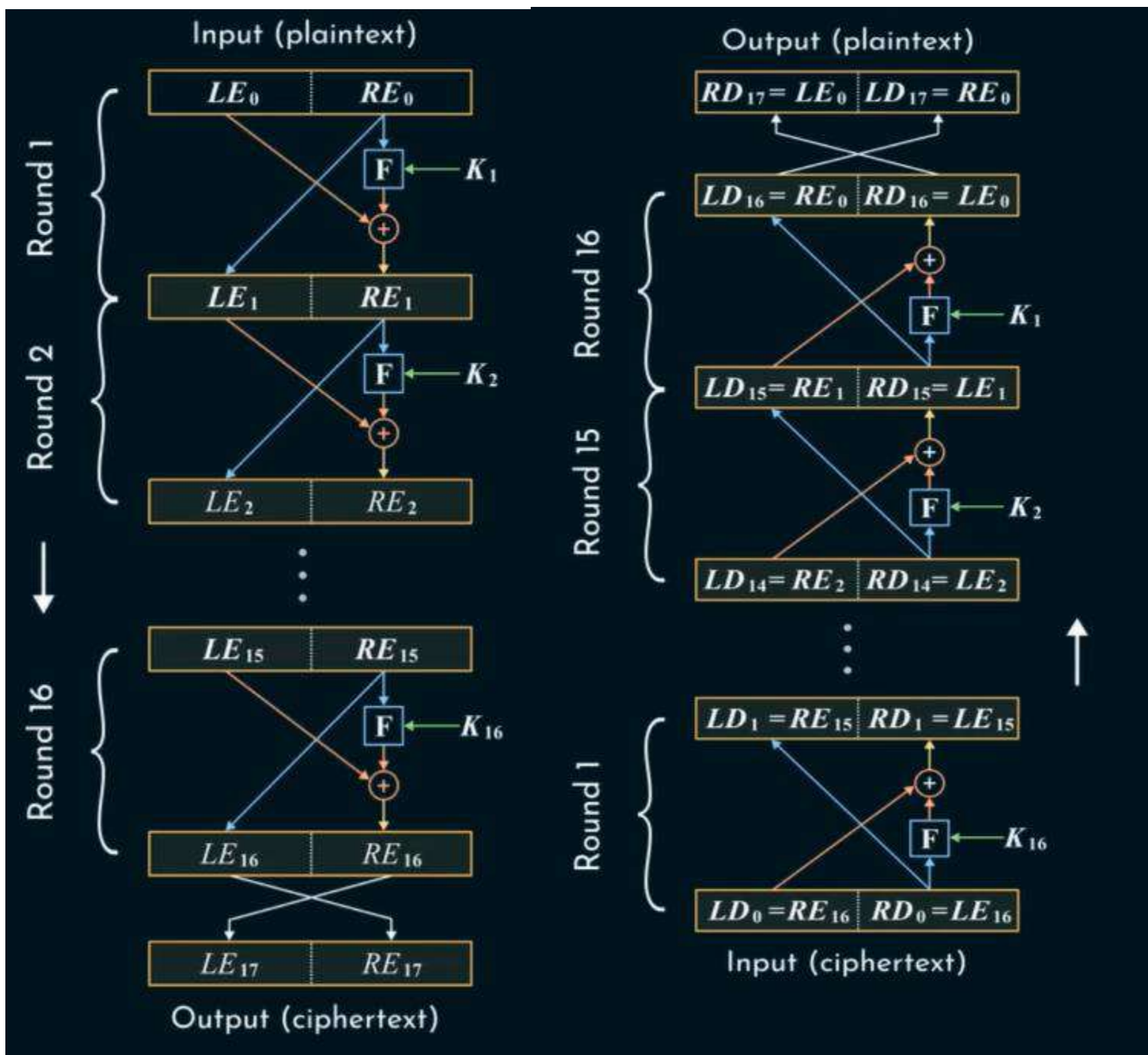
1. The ciphertext block is divided into two halves.
2. For each round i from 1 to n : $R_{i-1} = L_i$, $L_{i-1} = R_i \oplus F(L_i, K_{n-i+1})$
3. After the final round, the original plaintext block is obtained (again, considering potential swapping in the last round of encryption).



Examples of Ciphers Using Feistel Structure:

Many well-known and widely used block ciphers are based on the Feistel structure, including:

- **DES (Data Encryption Standard):** One of the earliest and most influential block ciphers.
- **Triple DES (3DES):** An enhancement of DES that applies the DES algorithm three times.
- **Blowfish:** A popular and efficient block cipher designed by Bruce Schneier.
- **Twofish:** Another strong block cipher designed as a candidate for AES.
- **FEAL (Fast Data Encipherment Algorithm):** An earlier cipher that used a Feistel structure.



Plain text OK

Ascii key for O =79 = hexadecimal 4F= binary = 0100 1111

Ascii key for K= 75= 4B= 0100 1011

From Key generator we derived sub keys k1 and k2

K0 = 1010 0110

K1= 1011 0111

Function (F) can be any complex operation,

Here suppose we use function (F) = AND operation

Input		Output
A	B	Y=A.B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

For first round

F0= AND of R0 and K0

= 0100 1011 AND 1010 0110

= 00000010

R1=L0 XOR F0

=0100 1111 X-OR 00000010

= 01001101

L1= R0=0100 1011

For Second round

F1= AND of R1 and K1

= 01001101 AND 1011 0111= 00000101

$R2 = L1 \text{ XOR } F1$

$= 0100\ 1011 \text{ XOR } 00000101$

$= 01001110$

$L2 = R1 = 01001101$

Now swap L2 and R2

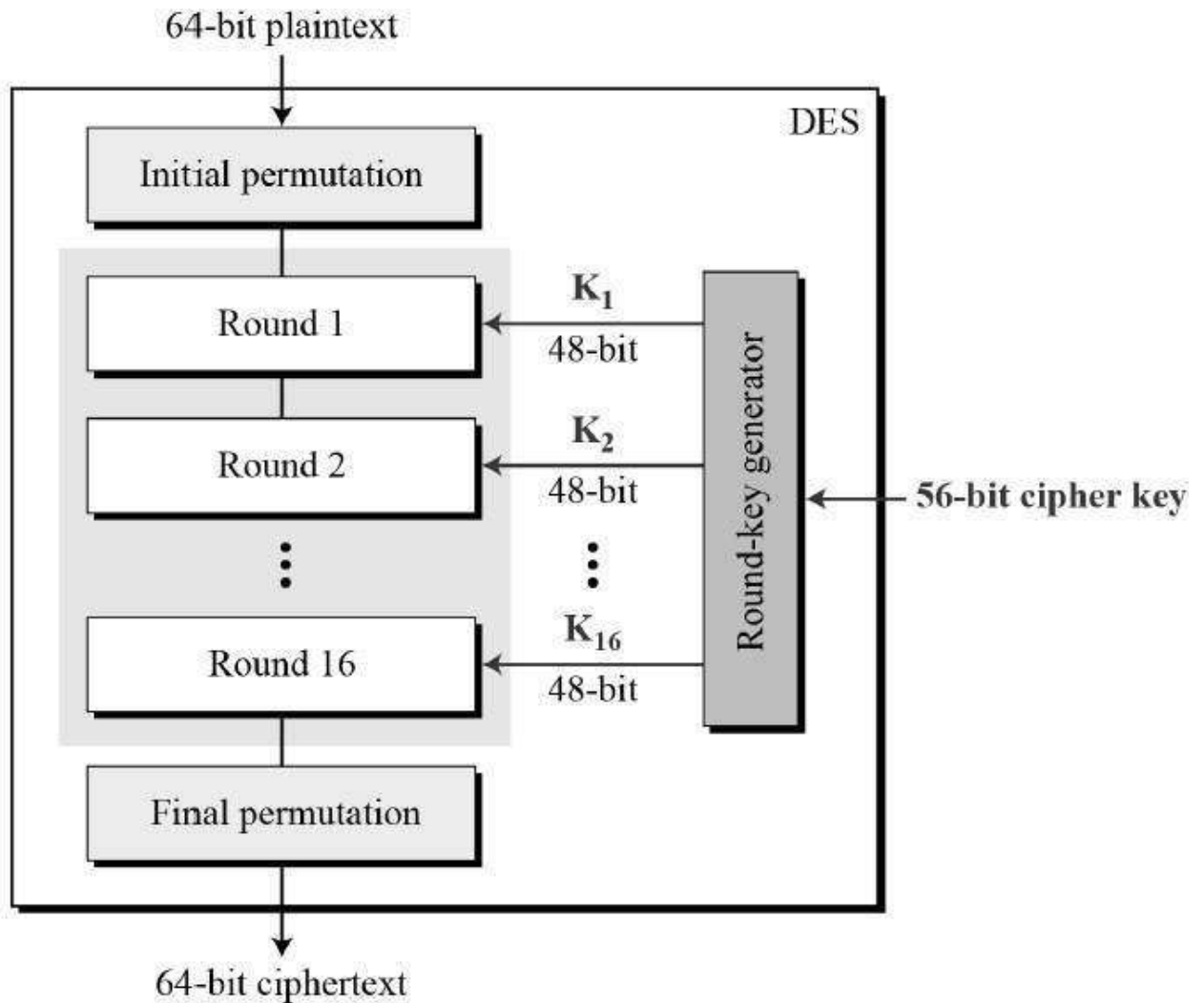
$Lc = 01001110 = 4E = 78 = N$

$Rc = 01001101 = 4D = 77 = M$

Now decryption

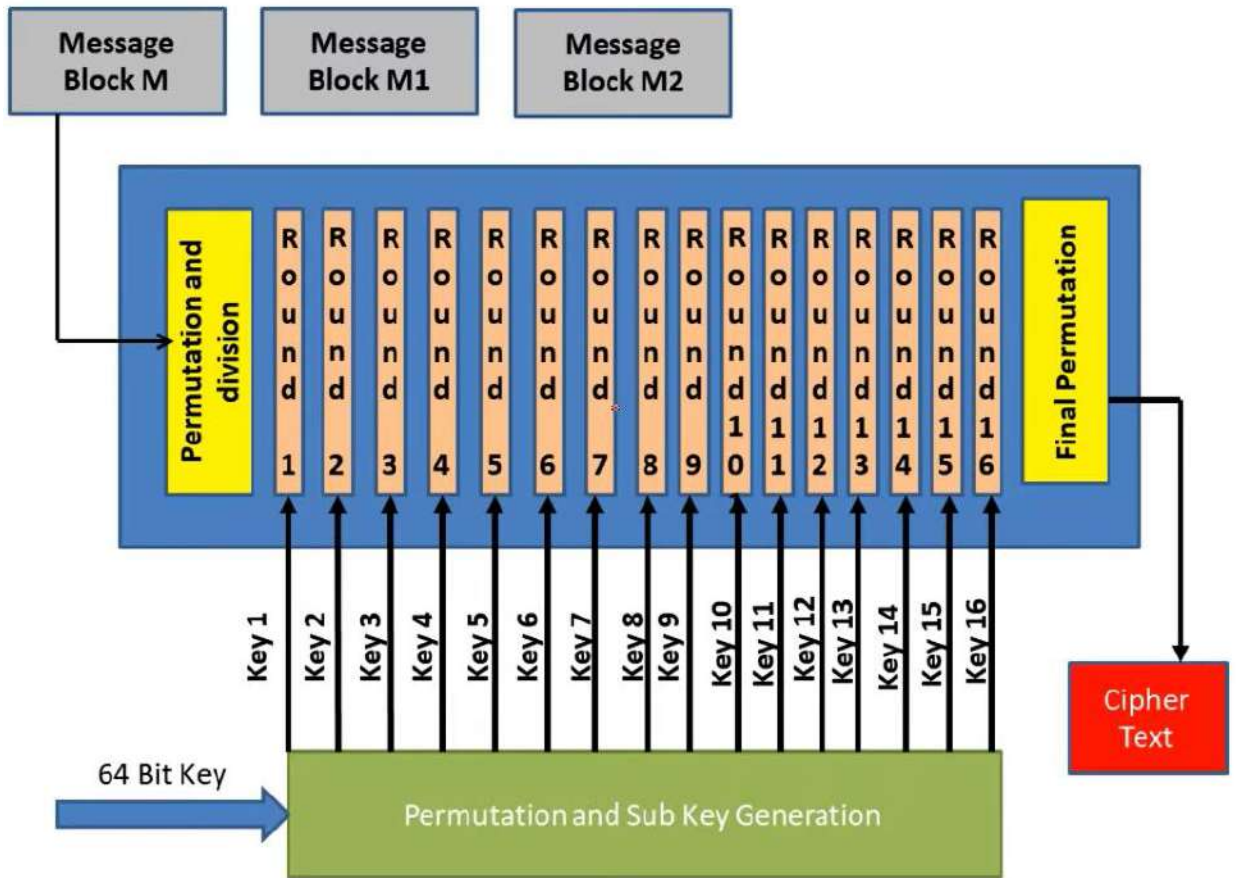
Data Encryption Standard (DES)

- Data Encryption Standard (DES) refers to a **symmetric key encryption algorithm** that was developed by the National Institute of Standards and Technology (NIST) in 1977 as an official standard for encrypting data.
- In symmetric-key algorithms, the same keys are used for both encryption and decryption thus, secure communication heavily relies on proper key management.
- DES is an implementation of a Feistel Cipher. It uses 16 rounds where substitution and permutation steps are applied to convert plaintext into ciphertext during each round of processing done by the algorithm. The plain text block size is 64-bit.
- Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only).
- Although AES has replaced it with stronger methods such as triple DES (3DES), some old systems still use this technology because it remains significant within cryptographic history.
- General Structure of DES is depicted in the following illustration



Since DES is based on the Feistel Cipher, all that is required to specify DES is –

- Round function
- Key schedule
- Any additional processing – Initial and final permutation



Phase 1

Generating 16 Sub keys

Key in Hexadecimal = 133457799BBCDFF1

K= 00010011 00110100 01010111 01111001 10011011 10111100 11011111
11110001

- A **64-bit key** is taken as input.
- This key is **permuted using the PC-1 table** (Permutation Choice 1).
- Only **56 bits** are selected; the remaining 8 bits (typically parity bits) are discarded.

PC-1 Table:

This table specifies the **bit positions** from the original 64-bit key that will be used (note: 8 parity bits are discarded).

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

PC-1

we get the 56-bit permutation

$K^+ = 1111000\ 0110011\ 0010101\ 0101111\ 0101010\ 1011001\ 1001111\ 0001111$

Next, split this key into left and right halves, C_0 and D_0 , where each half has 28 bits.

From the permuted key K^+ , we get

$C_0 = 1111000\ 0110011\ 0010101\ 0101111$

$D_0 = 0101010\ 1011001\ 1001111\ 0001111$

Iteration Number	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

C0 = 1111000 0110011 0010101 0101111
D0 = 0101010 1011001 1001111 0001111

Here we should apply circular left shift on C0 and D0 according to given schedules of left shift table.

C1 = 1110000110011001010101011111
D1 = 1010101011001 100111100011110

C2 = 1100001100110010101010111111
D2 = 010101011001 1001111000111101

C3 = 0000110011001010101011111111
D3 = 0101011001 100111100011110101

C4 =
D4 =

C5 =
D5 =

Schedules of “Left shift”

C6 =
D6 =

C7 =
D7 =

C8 =
D8 =

C9 =
D9 =

C10 =

D10 =

C11 =

D11 =

C12 =

D12 =

C13 =

D13 =

C14 =

D14 =

C15 =

D15 =

C16 =

D16 =

We now form the keys K_n , for $1 \leq n \leq 16$, by applying the following permutation table to each of the concatenated pairs $C_n D_n$.

Each pair has 56 bits, but PC-2 only uses 48 of these.

So, we are going to take C1D1 and apply on PC-2

C1 = 1110000110011001010101011111

D1 = 1010101011001 100111100011110

C1D1=11100001100110010101010111111010101011001100111100011110

PC-2 Table:

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

After applying PC-2 on C1D1 we get Key K1

K1 = 000110 110000 001011 101111 111111 000111 000001 110010

C2 = 1100001100110010101010111111

D2 = 010101011001 1001111000111101

C2D2= 1100001100110010101010111111010101011001 1001111000111101

After applying PC-2 on C2D2 we get Key K2

K2=011110011010111011011001110110111100100111100101

So all 16 keys are as follows

K1=00011011000000101110111111111000111000001110010

K2=011110011010111011011001110110111100100111100101

K3=010101011111110010001010010000101100111110011001

K4=011100101010110111010110110110110011010100011101
 K5=011111001110110000000111111010110101001110101000
 K6=011000111010010100111110010100000111101100101111
 K7=111011001000010010110111111101100001100010111100
 K8=111101111000101000111010110000010011101111111011
 K9=111000001101101111101011111011011110011110000001
 K10=101100011111001101000111101110100100011001001111
 K11=001000010101111111010011110111101101001110000110
 K12=011101010111000111110101100101000110011111101001
 K13=100101111100010111010001111110101011101001000001
 K14=010111110100001110110111111100101110011100111010
 K15=101111111001000110001101001111010011111100001010
 K16=110010110011110110001011000011100001011111110101

Phase 2

Step-2: Encode each 64-bit block of data

Let's suppose Message is

M= 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

There is an initial permutation IP of the 64 bits of the message data M by applying IP table

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

we get

IP=110011000000000011001100111111111110000101010101111000010101010

"Next divide the permuted block IP into a left half L_0 of 32 bits, and a right half R_0 of 32 bits."

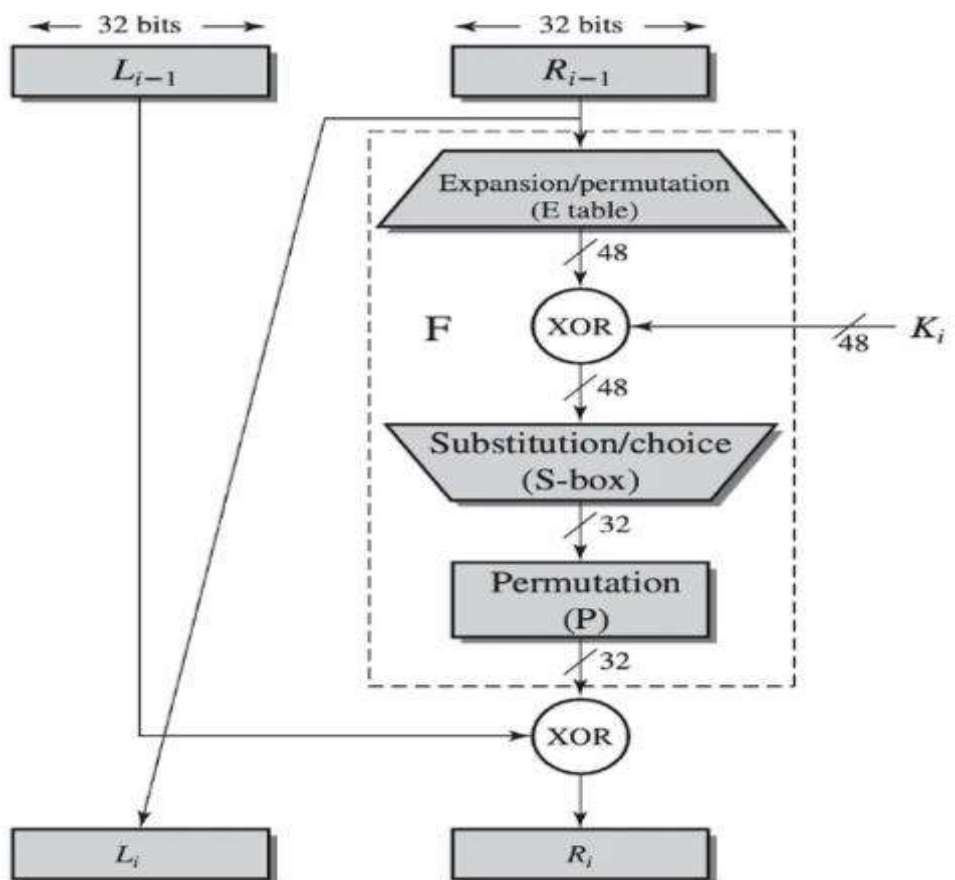
$L_0=11001100000000001100110011111111$

$R_0=11110000101010101111000010101010$

Phase 3

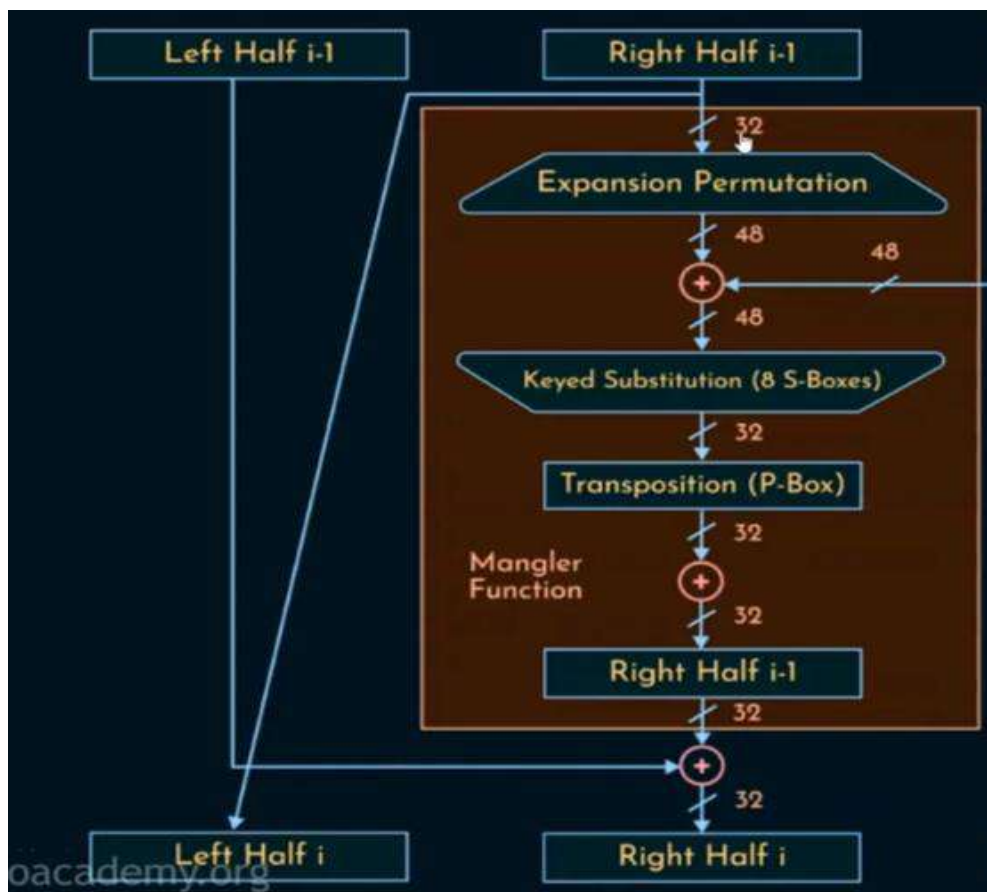
What Happens in Every Single Round?

- In a single round, the total input size is 64-bit into two halves 32–32-bit. The figure is

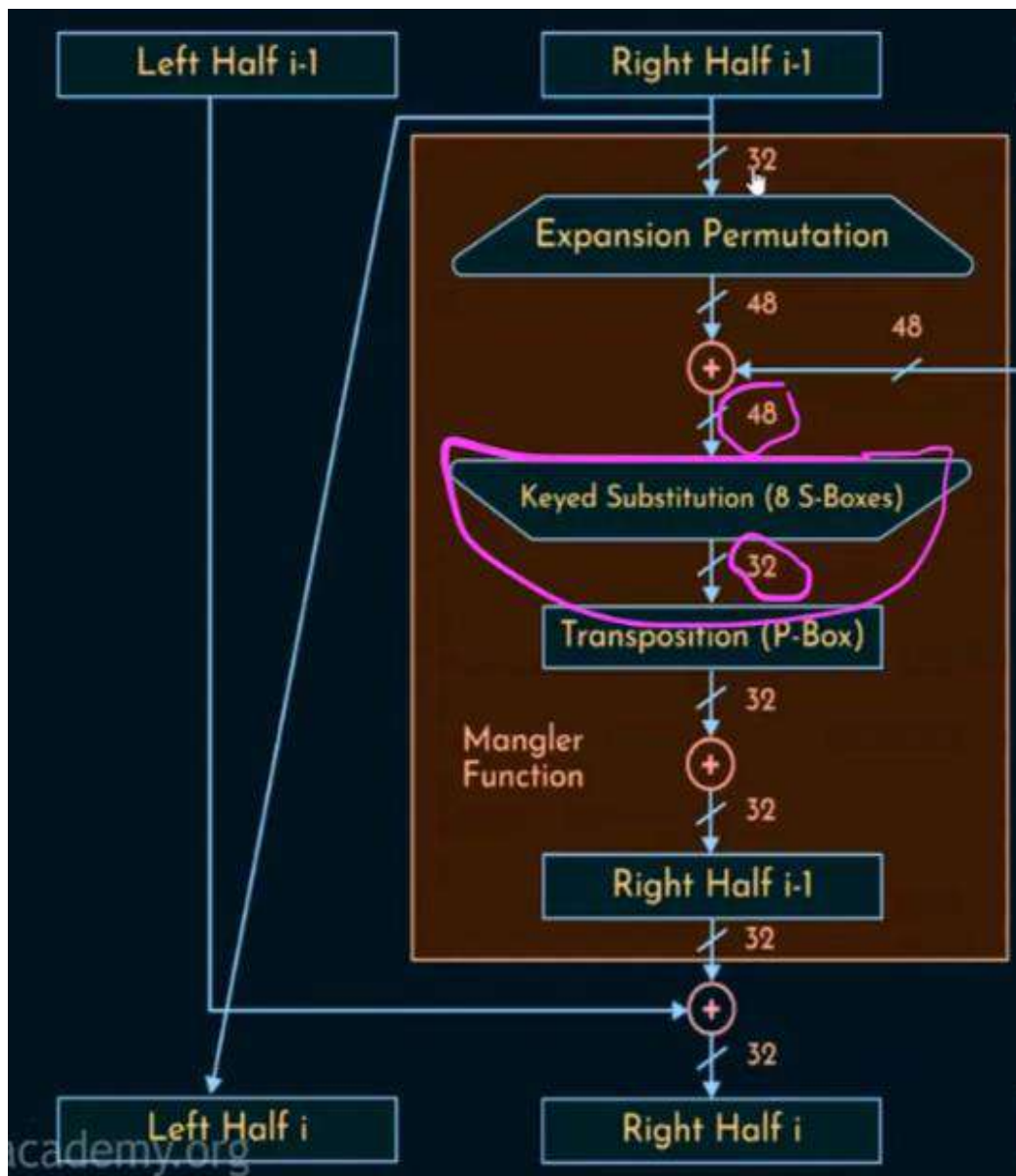


Steps of Single Round Operation:

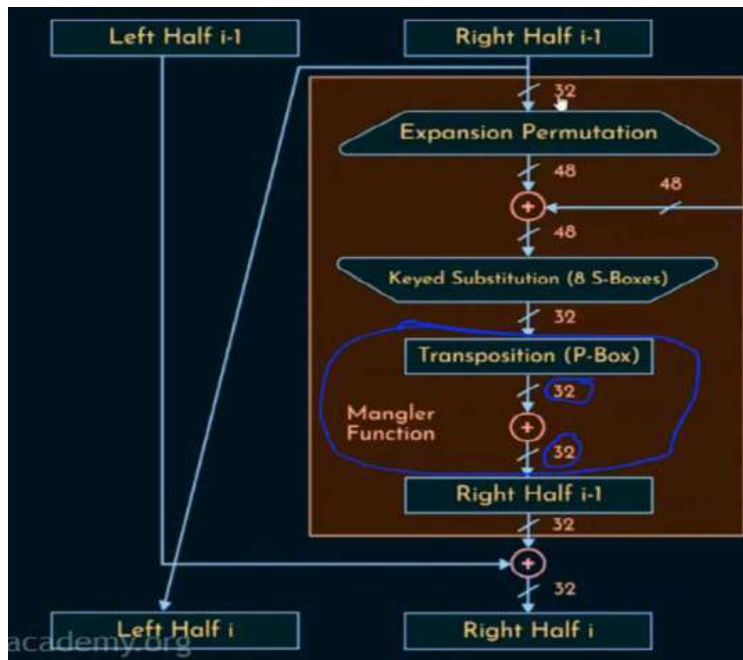
1. In the above figure, Left Half $i-1$ takes 32-bit Right Hand Side takes Half $i-1$ 32-bit, and the Right-Hand Half Side 32-bit is given to the **expansion permutation function**.
2. The expansion Permutation Function expands the 32-bit into 48-bits because we are getting the 48-bit key and our original input size was 64-bit.
3. The Right Side 32-bit has to be converted into 48-bit so that it can be XORED with the key 48-bit.



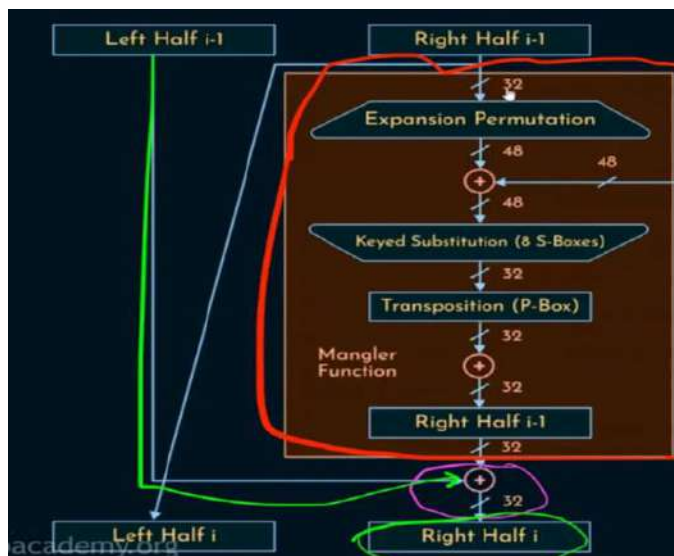
4. Again 8 S-Box reduces the input size of 48-bit to 32-bit and the output of the S-Box is given to the P-Box (Transposition).



5. P-Box changes the position of the bit and again gives to the right half $i - 1$ which performs the XORed Operation.



6. Now the right half $i-1$ 32-bit is XORed with the left-hand $i-1$ and the output is the 32-bit Whole Right Hand Half part. That means the figure shows the whole red bordered part output XORed with the left hand $i-1$ 32-bit input gives the Right Half Output i .

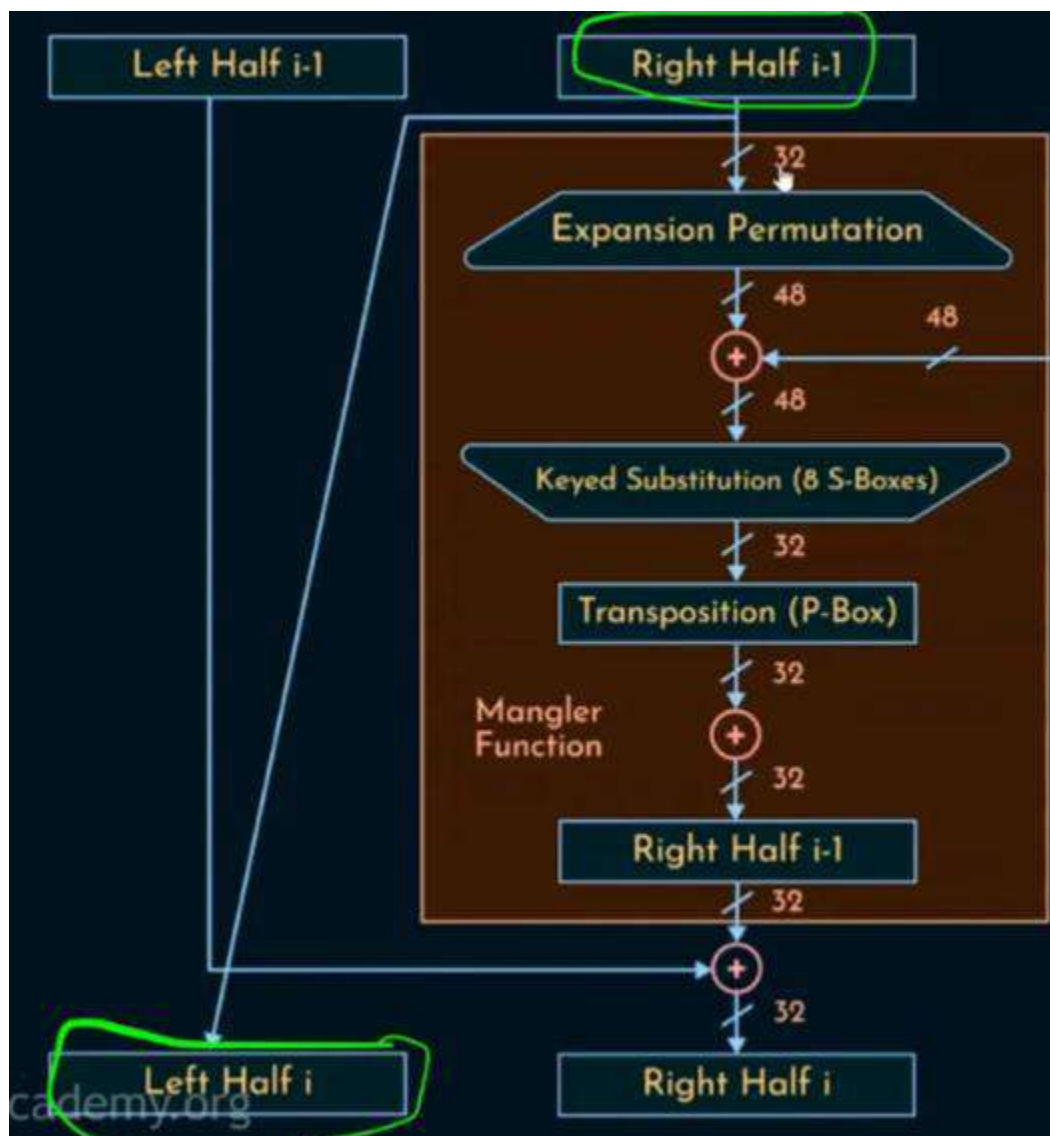


Note: Now we get the Right Half i and the Left Half i , these two 32-bit will be the input for another second-round operation up to 16-round operations. That means if we observe in the above figure before the expansion permutation, S-box substitution, transposition, and right half $i-1$ right half $i-1$ is the output for left half i which will be the input for another round operation.

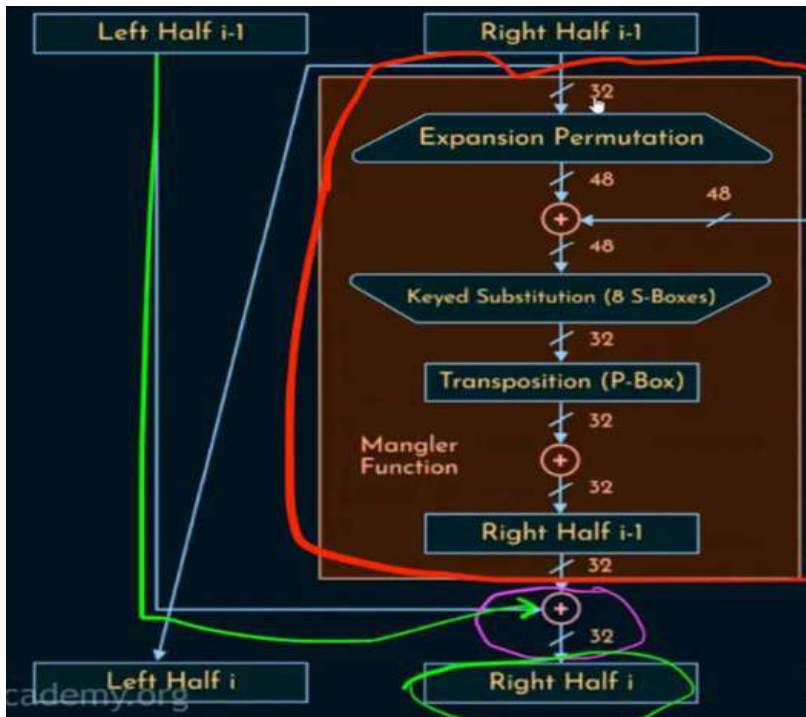
The Whole Steps of a Single Round Operation will be performed 16 times because DES uses 16 Round Operations.

If we try to outline the equation of the left and right-hand sides. See the green bordered part in the figure.

$$L_i = R_{i-1}$$

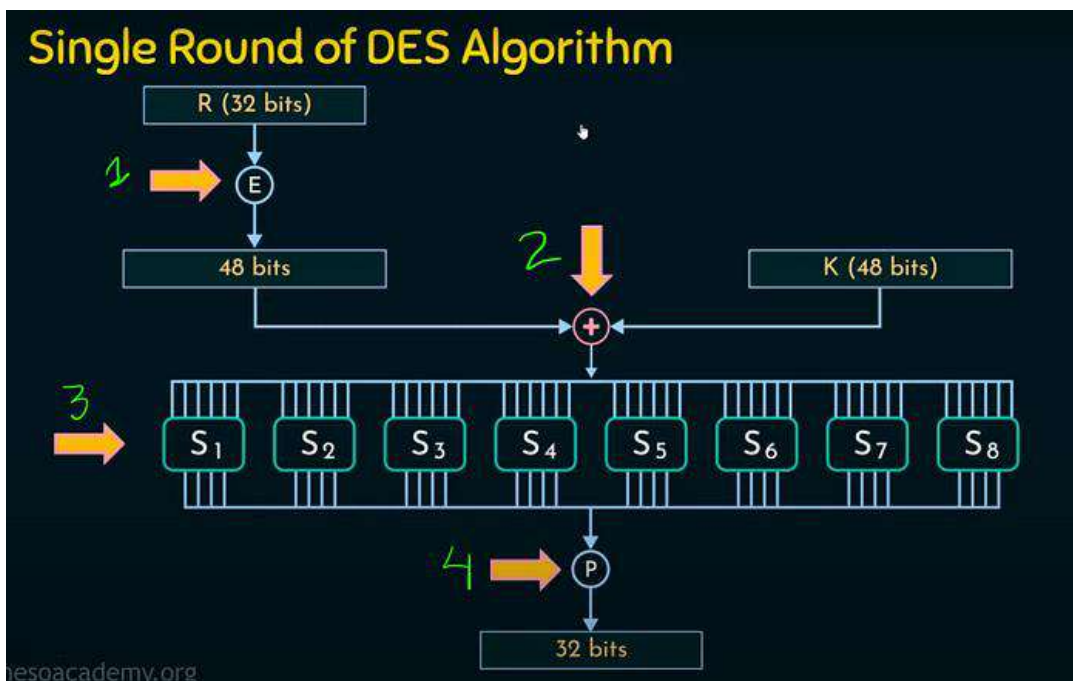


Then if we again look at the red bordered part as Mangler Function denoted by F we get the equation as $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$



Mangler Function

The Mangler Function F is the $F(R_{i-1}, K_i)$.

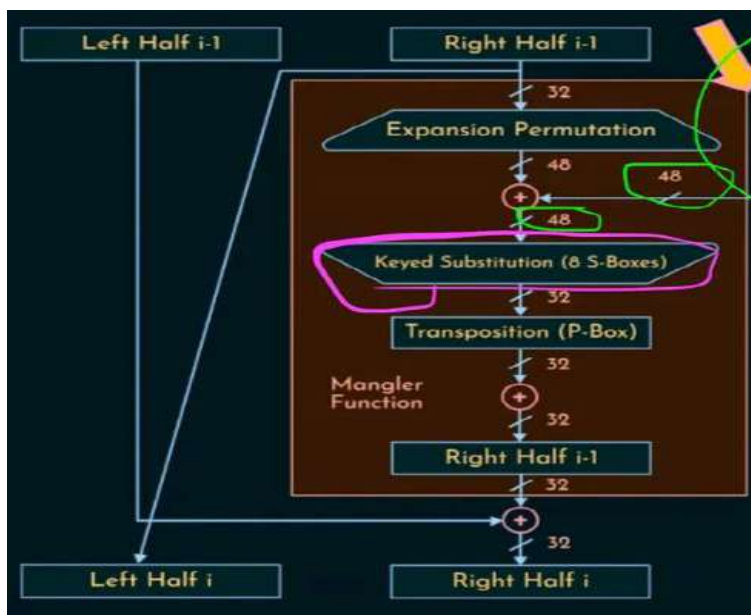


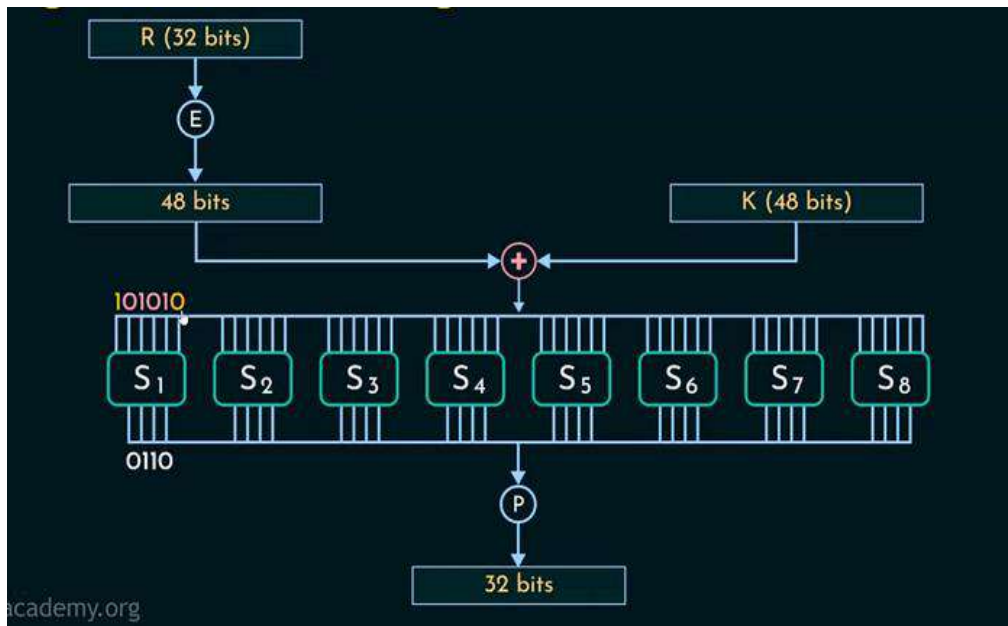
How Does Mangler Function / F Function Work?

- In the above figure green colored is step 1 is 32 Bits we need to Convert it into again 48 Bits.
- For converting 32 Bits to 48 Bits, we have to use below table which is known as the Expansion Permutation table.

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

- By arranging the right-hand sides 32 Bits According to this E-box we get 48 Bits.
- Now the R (32 Bit) operation is completed and we have the output of 48 bits from the R (32 Bits).
- We have to perform the **output of R (32 Bits)** XOR with the K (48 bits) the 48 Bits.
- After the XOR again we have to send them to the S-Box a purple-colored border part box. S-box takes 8 blocks of input 6 bits and output 8 blocks of 4 bits which makes it 32 bits.





How 6-bit Input in the S-Box Block is converted into 4-Bit Block?

Box S_1

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	6	5	3	8
10	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
11	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

For example, $S_1^*(101010) = 6 = 0110$.

- The figure shows all the 6-bit of 8 blocks. and this is for only S_1 . S_1 is taking 101010 input and giving output as 0110.
- The first and last bit from the $S_1 = 101010$ is 1 and 0 that is (10 which is a row in the table and the remaining 0101 is a column in the table).
- The value corresponding to row 10 and column 0101 is 6 (0110).
- In this way, we perform S-Box for all 8 S-blocks. The output 32-bit we will obtain from the S-box is again permuted.

Permutation Function Applied to the 32-bit Output from S-box

- The 32-bit output positions are only changed again and shuffled using the permutation function (P) which table is given below.

The 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

Phase 4

- Finally, after 16 rounds we get L_{16} (Left Half after 16 Rounds) and R_{16} (Right Half after 16 Rounds):

L_{16} (Left Half after 16 Rounds): 0100 0011 0100 0010 0011 0010 0011 0100

R_{16} (Right Half after 16 Rounds): 0000 1010 0100 1100 1101 1001 1001 0101

- In DES, before the final permutation, L_{16} and R_{16} are swapped:
- Concatenated as $R_{16}L_{16}$: 0000 1010 0100 1100 1101 1001 1001 0101 0100 0011 0100 0010 0011 0010 0011 0100
- Applying the inverse of the initial permutation (IP^{-1}) table on $R_{16}L_{16}$: we get
10000101 11101000 00010011 01010100 00001111 00001010 10110100
00000101

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

Now convert each 8-bit binary group to hexadecimal:

- 10000101 → 85
- 11101000 → E8
- 00010011 → 13
- 01010100 → 54
- 00001111 → 0F
- 00001010 → 0A
- 10110100 → B4
- 00000101 → 05

Cipher text in hexadecimal format is **85E813540F0AB405**.

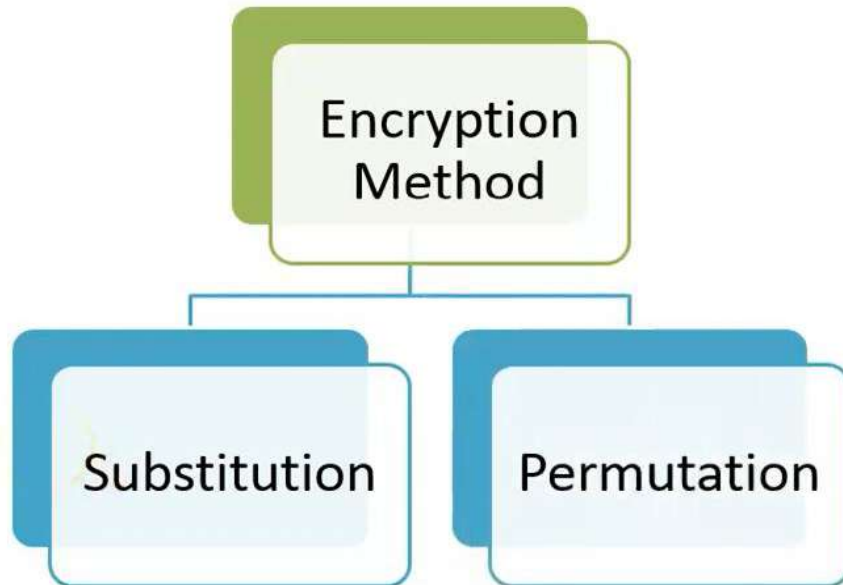
Like the we encrypt every block of message.

The truth

- DES is in secure as it uses a 56-bit key, which is now considered too short to withstand brute-force attacks.
- In **January 1999**, **distributed.net** and the **Electronic Frontier Foundation (EFF)** broke a DES key in just **22 hours and 15 minutes**, proving its vulnerability.
- DES has been **superseded by the Advanced Encryption Standard (AES)**, which offers stronger security.
- DES has been officially **withdrawn as a standard by the National Institute of Standards and Technology (NIST)**.

Encryption Methods

- There are two basic methods of encryption:



Confusion and Diffusion

- **Claude Shannon** introduced these two terms:
- **Confusion**
 - It is a technique of ensuring that a cipher text gives no clue about plain text.
 - It is used in block and stream cipher methods.
 - Achieved by Substitution technique.
 - Ex: ABC → XYZ
- **Diffusion**
 - Increases the redundancy of the plain text by spreading it across rows and columns.
 - It is used in block cipher methods.
 - Achieved by permutation known as Transposition technique.
 - Ex: ABC → CAB

No	Confusion	Diffusion
1	Cipher text cannot give the clue about plain text.	Increase the redundancy of plain text and generate cipher text.
2	Confusion technique is used in both block and stream cipher.	Diffusion technique is only used in block cipher.
3	Confusion hides the relation between the ciphertext and key.	Diffusion hides the relation between the ciphertext and the plaintext.
4	This technique is possible through substitution algorithm.	While it is possible through transportation algorithm.
5	If a single bit in the key is changed, all the bits in the ciphertext will also have to be changed.	In case a symbol in the plaintext is changed, several or all symbols in the cipher text will also have to be changed.

Why Was the AES Encryption Algorithm necessary?

- When the **Data Encryption Standard** algorithm, also known as the DES algorithm, was formed and standardized, it made sense for that generation of computers.
- Going by today's computational standards, breaking into **the DES algorithm** became easier and faster with every year, as seen in the image below.

Chronology of DES Cracking	
Broken for the first time	1997
Broken in 56 hours	1998
Broken in 22 hours and 15 minutes	1999
Capable of broken in 5 minutes	2021

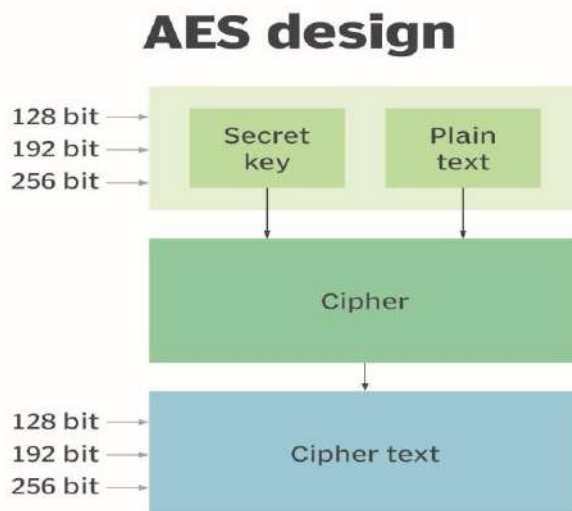
Source: Wikipedia

- A more robust algorithm was the need of the hour, with longer key sizes and stronger ciphers to break into.
- They created the **triple DES to fix this problem**, but it never became mainstream because of its relatively **slower pace**.
- **Thus, the Advanced Encryption Standard came into existence to overcome this drawback.**

❖ **AES (Advanced Encryption Standard)**

- The **AES Encryption** algorithm (also known as the **Rijndael algorithm**) is a **symmetric block cipher** algorithm with a block/chunk size of **128 bits**.
- It is developed by the **National Institute of Standards and Technology (NIST)** in 2001.

- It is widely used today as it is much stronger than **DES and triple DES** despite being harder to implement.
- It converts these individual blocks using keys of **128, 192, and 256 bits**.
- It is based on a **substitution-permutation network**, also known as an **SP network**.
- It consists of a series of linked operations, including replacing inputs with specific outputs (**substitutions**) and others involving bit shuffling (**permutations**).



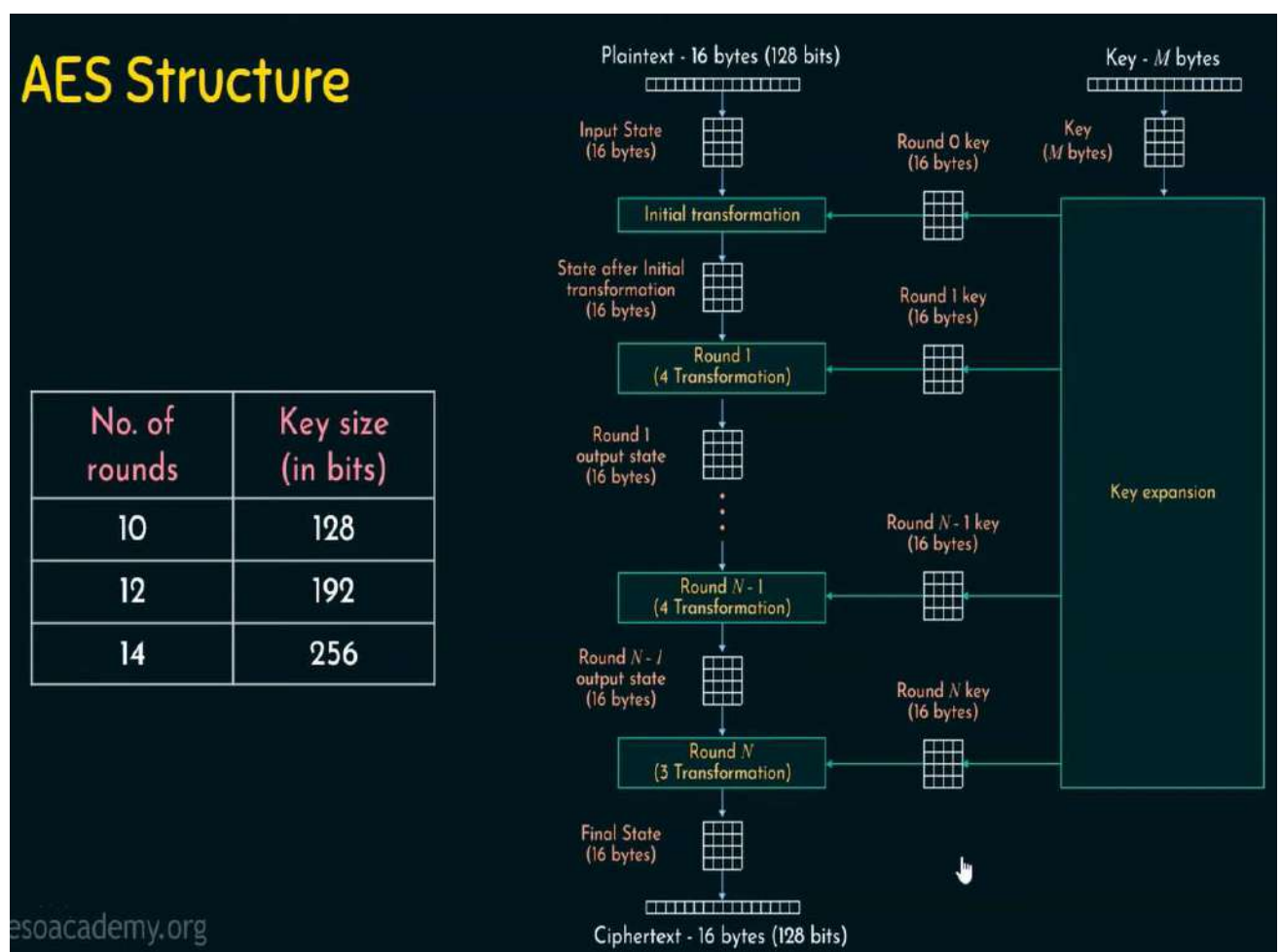
❖ What are the Features of AES?

1. Symmetric key symmetric block cipher
2. 128-bit data, 128/192/256-bit keys
3. Stronger and faster than Triple-DES

❖ Operation of AES

- AES is an iterative rather than Feistel cipher. It is based on **substitution permutation network**.
- It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (**substitutions**) and others involve shuffling bits around (**permutations**).

- Interestingly, AES performs all its computations on **bytes rather than bits**. Hence, AES treats the **128 bits of a plaintext block as 16 bytes**. These 16 bytes are arranged in four columns and four rows for processing as a matrix
- Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.
- The schematic of AES structure is given in the following illustration –



AES Parameters

	AES-128	AES-192	AES-256
Key Size	128	192	256
Plaintext Size	128	128	128
Number of rounds	10	12	14
Round Key Size	128	128	128

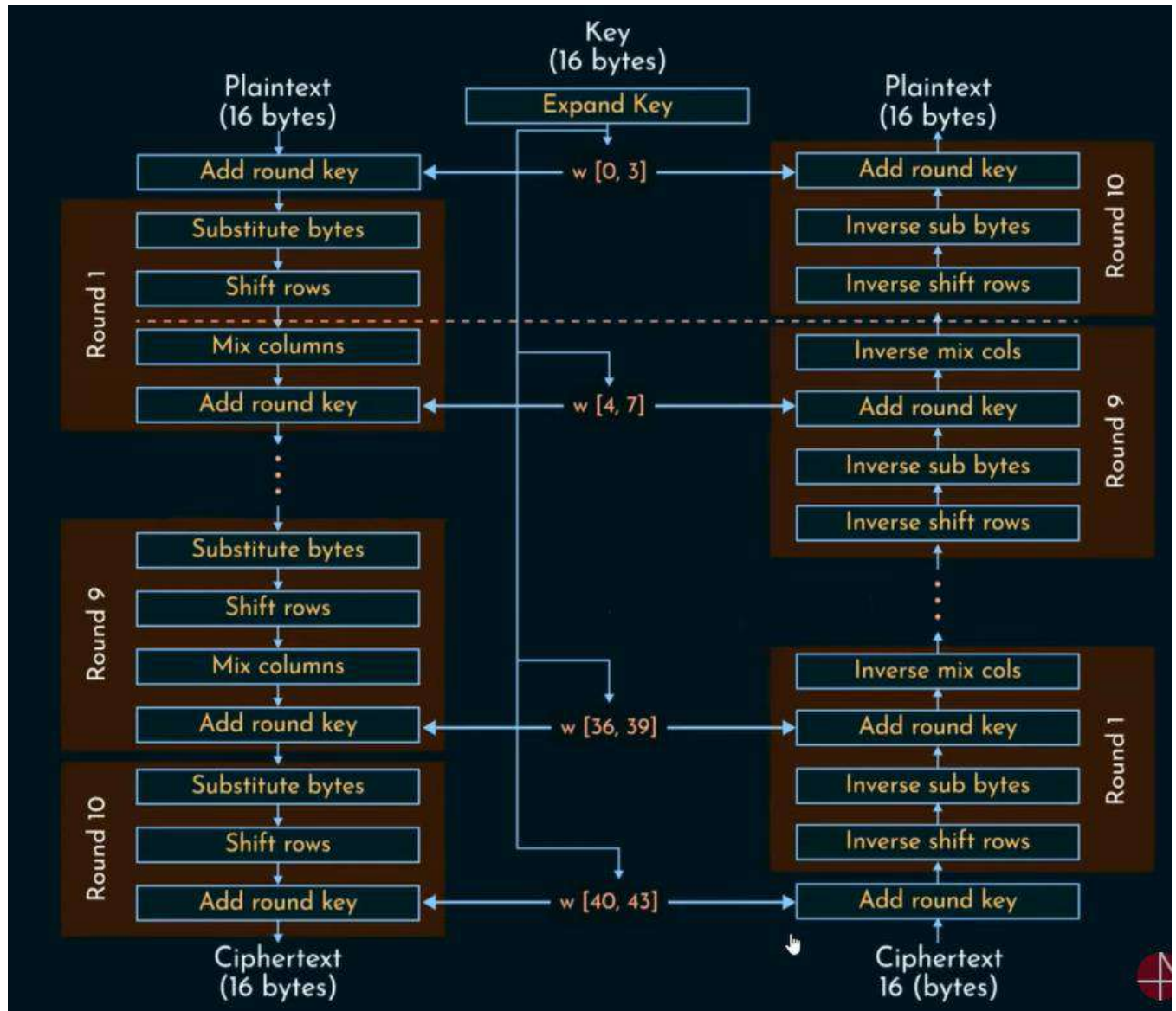
How Does AES Work?

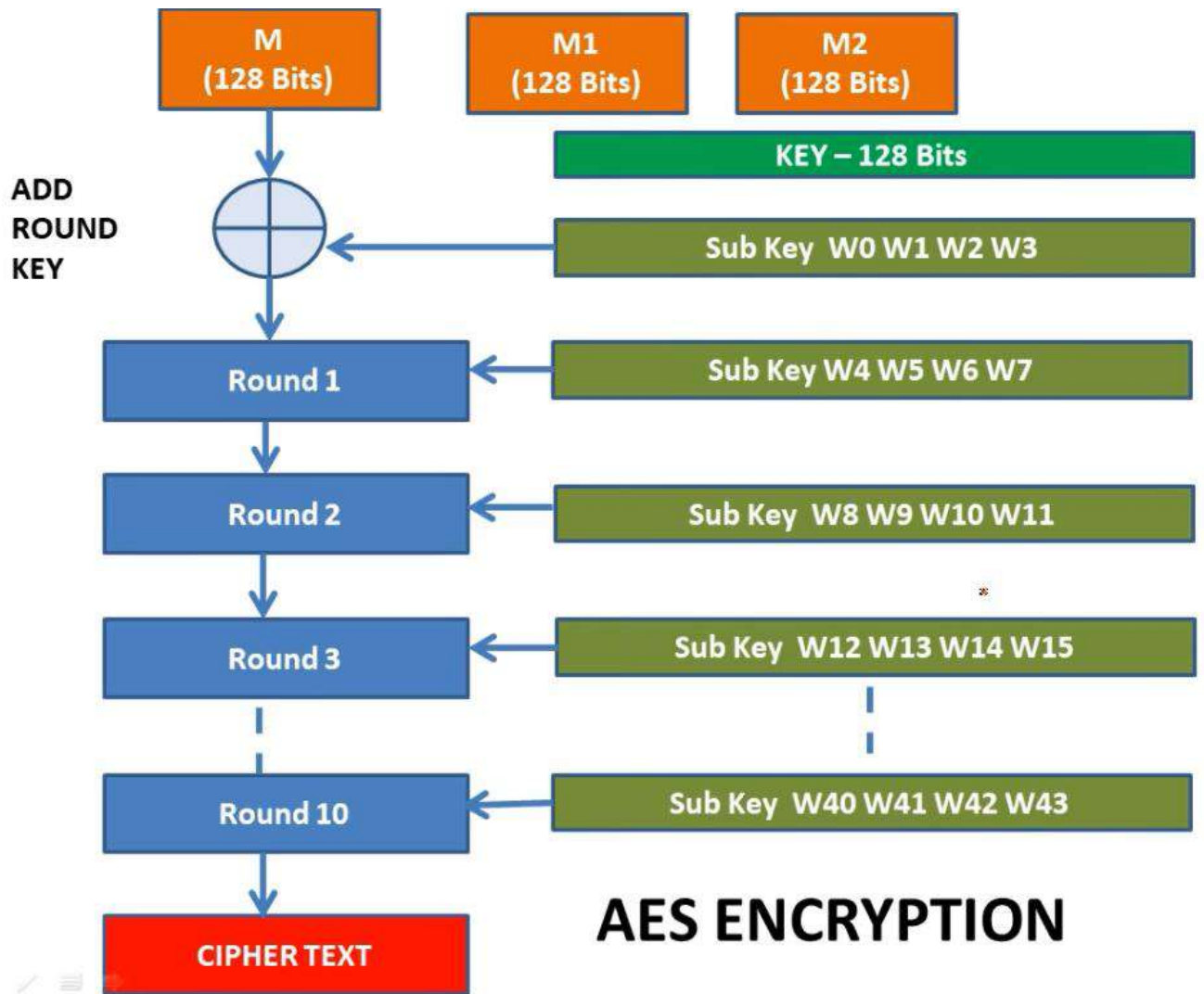
- To understand the way AES works, you first need to learn how it transmits information between multiple steps.
- Since a single block is 16 bytes, a 4x4 matrix holds the data in a single block, with each cell holding a single byte of information.

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

- The matrix shown in the image above is known as a state array. Similarly, the key being used initially is expanded into $(n+1)$ keys, with n being the number of rounds to be followed in the encryption process.
 - So, for a 128-bit key, the number of rounds is 10, with no. of keys to be generated being $10+1$, which is a total of 11 keys.
1. It takes the input of 16 bytes (128 bits) and outputs the ciphertext (128 bits).
 2. Every $4 \times 4 = 16$ bytes are element state arrays that can store 16 bytes of information
 3. Input state: it is storing input plain text of 16 bytes and gives to the initial transformation
 4. In the initial transformation, the input plain text of 16 bytes along with the transformation function and the output is given to the state after initial transformation element arrays which is again given to the Round 1
 5. Round 1 Take step 4 output and perform 4 transformations with it
 6. Similarly, Round 2 takes the previous step round operations + 4 transformations output until Round $N - 1$
 7. At the last Round N , the output of Round $N - 1$ is the input of Round N which is the last round and it has only 3 Transformation
 8. The output of step 7 is now stored in element arrays. And the 16 bytes is the actual cipher that we want from this AES Encryption Structure.

AES Encryption and Decryption





Key Expansion in AES

Key in text= sathcjisboring

"sathcjisboring" = 16 characters → 1 character = 1 byte (in ASCII)

01110011 01100001 01110100 01101001 01110011 01101000 01100011
 01101010 01101001 01110011 01100010 01101111 01110010 01101001
 01101110 01100111

Hex representation:

73 61 74 69 73 68 63 6a 69 73 62 6f 72 69 6e 67
 b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10} b_{11} b_{12} b_{13} b_{14} b_{15} b_{16}

Now representing key in 4*4 matrix we get

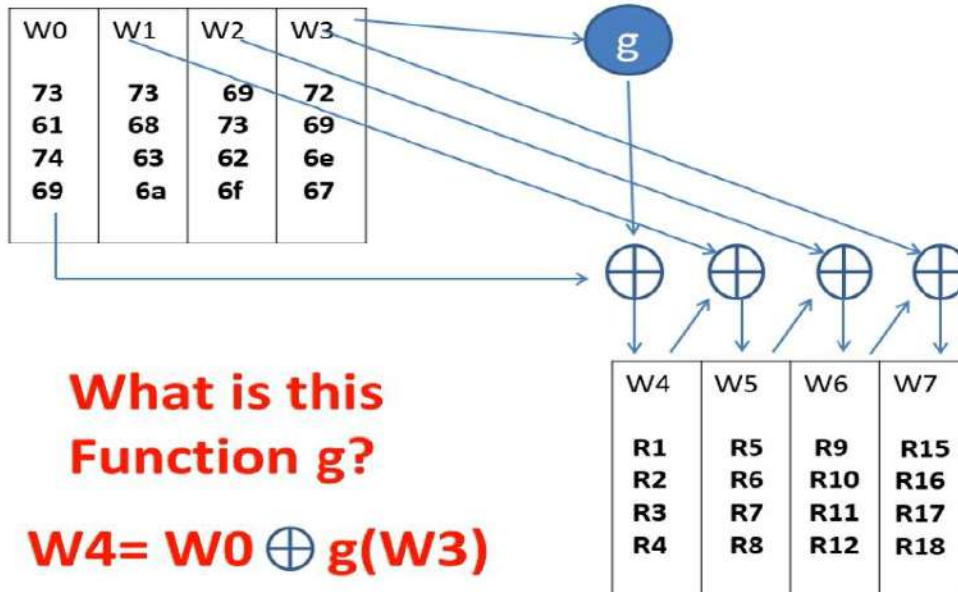
$$\begin{bmatrix} b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \\ b_4 & b_8 & b_{12} & b_{16} \end{bmatrix}$$

*W0	W1	W2	W3	W4	W5	W6	W7			<u>W43</u>
b₁	b₅	b₉	b₁₃							
b₂	b₆	b₁₀	b₁₄							
b₃	b₇	b₁₁	b₁₅							
b₄	b₈	b₁₂	b₁₆							

W0	W1	W2	W3	W4	W5	W6	W7			W43
73	73	69	72							
61	68	73	69							
74	63	62	6e							
69	6a	6f	67							

Now how to expand available 4 words in to 40 words?

Key Expansion in AES



So, the idea is to generate enough keys (44 words) *ie., 11 keys* based on the initial key (16 bytes, 4 words) to be used in 10 rounds of the encryption process.

What is this function g?

$$W4 = W0 \oplus g(W3)$$

Step 1

- Take W3 do a cyclic left shift 1 for each byte and we will be able to get RotWord (X1).
- Rot word performs a one-byte circular left shift on a word.
- This means that an input word [B0, B1, B2, B3] is transformed into [B1, B2, B3, B0]

Step 2

- From RotWord we have to find Subword (Y1)
- SubWord performs a byte substitution on each byte of its input word, using the AES S-box

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

Step 3

- we have to perform XOR operation on subword with the constant called the round constant, Rcon[i]
- Round constant is

[illegible]

W3	RotWord (X1)	SubWord (Y1)	The result Y1 is XORed with a round constant, Rcon[j].
72	69	f9	
69	6e	9f	
6e	67	85	
67	72	40	

Y1 11111001100111111000010101000000
R1 00000000100000000000000000000000
g(w3) 11111000100111111000010101000000

$g(w3) = \text{F8 9F 85 40}$

Now,

$$W4 = W0 \oplus g(W3)$$

$$\begin{array}{r}
 01110011 \ 01100001 \ 01110100 \ 01101001 \\
 \oplus \\
 11111000 \ 10011111 \ 10000101 \ 01000000 \\
 \\
 = \quad 10001011 \ 11111110 \ 11110001 \ 00101001 \\
 = \quad 8b \ fe \ f1 \ 29
 \end{array}$$

After getting W4 it will be easy, now we have to take W4 and $\oplus$ with W1, we get w5.

Likewise, we have to take W5 and $\oplus$ with W2 we get W6.

Likewise, we have to take W_6 and $\oplus$ with W_3 we get W_7 .

So ,

$$W_4 = 8b\ fe\ f1\ 29$$

$$W_5 = f8\ 96\ 92\ 43$$

$$W_6 = 91\ e5\ f0\ 2c$$

$$W_7 = e3\ 8c\ 9e\ 4b$$

So, sub key for round 1 we get = $8b\ fe\ f1\ 29\ f8\ 96\ 92\ 43\ 91\ e5\ f0\ 2c\ e3\ 8c\ 9e\ 4b$

Now for sub key 2 we have to find W_8 to W_{11}

$$W_8 = W_4 \oplus g(W_7)$$

$$W_9 = W_8 \oplus W_5$$

$$W_{10} = W_9 \oplus W_6$$

$$W_{11} = W_{10} \oplus W_7$$

So, after combining output of $W_8\ W_9\ W_{10}$ and W_{11} we get sub key 2 for round 2.

Like that we have to achieve W_0 to W_{43} .

Add round key Or Initial Transformation

- In the initial transformation, we add the round key and the plain text of 16 bytes.
- W [0,1,2,3] is a round key given to the initial transformation and added rounded key with the plain text of 128 bits and finally XORed Operation is performed. Once the XORed Operation is performed it goes into round 1

M
(128 Bits)

secretmessagenow

73 65 63 72 65 74 6d 65 73 73 61 67 65 6e 6f 77

$$\begin{bmatrix} 73 & 65 & 73 & 65 \\ 65 & 74 & 73 & 6e \\ 63 & 6d & 61 & 6f \\ 72 & 65 & 67 & 77 \end{bmatrix} \oplus \begin{bmatrix} 73 & 73 & 69 & 72 \\ 61 & 68 & 73 & 69 \\ 74 & 63 & 62 & 6e \\ 69 & 6a & 6f & 67 \end{bmatrix}$$

73 01110011

73 01110011

Result 00000000

The output matrix we get is state array.

$$\begin{bmatrix} 00 & 16 & 1a & 17 \\ 04 & 1c & 00 & 07 \\ 17 & 0e & 03 & 01 \\ 1b & 0f & 0f & 10 \end{bmatrix}$$

AES Detailed Round Operations

- This section describes what happens in each round or the transformation functions.
 - There are 4 operations in every round except (in round 10 or the last round of encryption).
1. Substitute Bytes (Sub-Bytes) - Substitution Operations
 2. Shift Rows - Permutation Operation
 3. Mix Columns - Substitution Operations
 4. Add Round Key - Substitution Operations

Byte Substitution

- Does a simple replacement of each byte of the block data using a S-box.
- Left four bits determines row and right four bits determines column.
- **input to the round one is output of Add round key Or Initial Transformation**

$$\begin{bmatrix} 00 & 16 & 1a & 17 \\ 04 & 1c & 00 & 07 \\ 17 & 0e & 03 & 01 \\ 1b & 0f & 0f & 10 \end{bmatrix}$$

AES S-Box

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

- By using this S box Substitute each Bytes of output of **Add round key Or Initial Transformation**.
- Output will be

$$\begin{bmatrix} 63 & 47 & a2 & f0 \\ f2 & 9c & 63 & c5 \\ f0 & ab & 7b & 7c \\ af & 76 & 76 & ca \end{bmatrix}$$

Shift Rows

- ❖ Output state array (i.e., 4*4 matrix) of Byte Substitution will be input of Shift rows
- ❖ Shift rows simply shifts the rows bytes.
 - First row: No change
 - Second row: one byte cyclical left shift
 - Third row: two - byte cyclical left shift
 - Fourth row: three - byte cyclical left shift

$$\begin{bmatrix} 63 & 47 & a2 & f0 \\ f2 & 9c & 63 & c5 \\ f0 & ab & 7b & 7c \\ af & 76 & 76 & ca \end{bmatrix} = \begin{bmatrix} 63 & 47 & a2 & f0 \\ 9c & 63 & c5 & f2 \\ 7b & 7c & f0 & ab \\ ca & af & 76 & 76 \end{bmatrix}$$

- So, output of Shift Rows in round function is given below

$$\begin{bmatrix} 63 & 47 & a2 & f0 \\ 9c & 63 & c5 & f2 \\ 7b & 7c & f0 & ab \\ ca & af & 76 & 76 \end{bmatrix}$$

- And this state array 4*4 matrix will be input for mix column.

Mix Columns

- The third transformation function under round operation of AES is called as mix column, operates on each column individually.
- In AES mix column there is one pre-defined 4*4 matrix

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

Predefine Matrix

- So, this predefined matrix is multiplied with state array.
- State array = output of shift rows function.
- Here both predefined matrix and state array matrix is multiplied and generate new state array.

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

 $*$

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

 $=$

$S'_{0,0}$	$S'_{0,1}$	$S'_{0,2}$	$S'_{0,3}$
$S'_{1,0}$	$S'_{1,1}$	$S'_{1,2}$	$S'_{1,3}$
$S'_{2,0}$	$S'_{2,1}$	$S'_{2,2}$	$S'_{2,3}$
$S'_{3,0}$	$S'_{3,1}$	$S'_{3,2}$	$S'_{3,3}$

Predefine Matrix

State Array

New State Array

- New state array is the output of Mix Column transformation.
- Each byte of a column is mapped in to a new value that is a function of all four bytes in that column.

- For example:

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

?	?	?	?
?	?	?	?
?	?	?	?
?	?	?	?

Predefine Matrix
State Array
New State Array

- ❖ Here values in both matrix is in hexadecimal format. so, we cannot solve it as simple multiplication as in decimal. For this matrix multiplication we have to use binary number system and polynomial theorem based on the finite field arithmetic's.

AES Mix Column

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

?			

$$\{02\} * \{87\} \oplus \{03\} * \{6E\} \oplus \{01\} * \{46\} \oplus \{01\} * \{A6\}$$

$$X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X^1 + 1$$

$02 = 0000\ 0010 = X$
 $87 = 1000\ 0111 = X^7 + X^2 + X + 1$

$$X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X^1 + 1$$

$$\begin{aligned} \{02\} * \{87\} &= X * (X^7 + X^2 + X + 1) \\ &= X^8 + X^3 + X^2 + X \\ &= X^4 + X^3 + 1 + X^3 + X^2 + X \\ &= X^4 + X^2 + 1 \\ &= 0001\ 0101 \end{aligned}$$

Use irreducible Polynomial Theorem, GF (2⁸)
 $X^8 = X^4 + X^3 + X + 1$

$$X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X^1 + 1$$



- Here binary value is mapped with A Galois field, also known as a finite field $GF(2^8)$ polynomial theorem. i.e., $(X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X + 1)$
- If power is greater than 7 i.e., (X power >7) we use irreducible polynomial Theorem, $GF(2^8) = (X^8 + X^4 + X^3 + X + 1)$ -- remember this always
 $= (X^8 + X^3 + X^2 + X) \bmod (X^8 + X^4 + X^3 + X + 1)$, **note: here reduce same element and write it once.**

$$= X^4 + X^2 + 1$$

$$= X^4 + X^2 + X^0 = 00010101$$

AES Mix Column

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

?			

$$\{02\} * \{87\} \oplus \{03\} * \{6E\} \oplus \{01\} * \{46\} \oplus \{01\} * \{A6\}$$

$$02 = 0000\ 0010 = X$$

$$87 = 1000\ 0111 = X^7 + X^2 + X + 1$$

$$\begin{aligned} 02 * 87 &= X * (X^7 + X^2 + X + 1) \\ &= X^8 + X^3 + X^2 + X \\ &= X^4 + \cancel{X^3} + \cancel{X} + 1 + \cancel{X^3} + X^2 + \cancel{X} \\ &= X^4 + X^2 + 1 \\ &= 0001\ 0101 \end{aligned}$$

$$03 = 0000\ 0011 = X + 1$$

$$6E = 0110\ 1110 = X^6 + X^5 + X^3 + X^2 + X$$

$$\begin{aligned} 03 * 6E &= (X+1) * (X^6 + X^5 + X^3 + X^2 + X) \\ &= X^7 + \cancel{X^6} + X^4 + \cancel{X^3} + \cancel{X^2} + \cancel{X} + X^6 + X^5 + \cancel{X^3} + \cancel{X^2} + X \\ &= X^7 + X^5 + X^4 + X \\ &= 1011\ 0010 \end{aligned}$$

$$01 * 46 = 46 = 0100\ 0110$$

$$01 * A6 = A6 = 1010\ 0110$$

$$01 = 0000\ 0001 = X^0 = 1$$

$$46 = 0100\ 0110 = X^6 + X^2 + X^1$$

$$= 01000110$$

$$= (01) * (46) = 1 * (X^6 + X^2 + X^1) = X^6 + X^2 + X^1 = 01000110$$



AES Mix Column

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47			

$$\{02\} * \{87\} \oplus \{03\} * \{6E\} \oplus \{01\} * \{46\} \oplus \{01\} * \{A6\} = \{47\}$$

$$02 * 87 = 00010101$$

$$03 * 6E = 10110010$$

$$01 * 46 = 01000110$$

$$01 * A6 = 10100110$$

$$\begin{array}{r} 00010101 \\ 10110010 \\ 01000110 \\ 10100110 \\ \hline 01000111 \end{array} = \{47\}$$

4 7

In Ex-or

Odd time 1's = 1

Even time 1's = 0

AES Mix Column

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47			
?			

$$\{01\} * \{87\} \oplus \{02\} * \{6E\} \oplus \{03\} * \{46\} \oplus \{01\} * \{A6\}$$

$$01 * 87 = 87 = 10000111$$

$$02 = 00000010 = X$$

$$6E = 01101110 = X^6 + X^5 + X^3 + X^2 + X$$

$$\begin{aligned} 02 * 6E &= X * (X^6 + X^5 + X^3 + X^2 + X) \\ &= X^7 + X^6 + X^4 + X^3 + X^2 \\ &= 11011100 \end{aligned}$$

$$03 = 00000110 = X + 1$$

$$46 = 01000110 = X^6 + X^2 + X$$

$$\begin{aligned} 03 * 46 &= (X + 1) * (X^6 + X^2 + X) \\ &= X^7 + X^3 + \cancel{X^2} + X^6 + \cancel{X^2} + X \\ &= X^7 + X^6 + X^3 + X \\ &= 11001010 \end{aligned}$$

$$01 * A6 = A6 = 10100110$$

AES Mix Column

2	3	1	1
<u>1</u>	<u>2</u>	<u>3</u>	<u>1</u>
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47			
?			

$$\{01\} * \{87\} \oplus \{02\} * \{6E\} \oplus \{03\} * \{46\} \oplus \{01\} * \{A6\}$$

$$\left. \begin{aligned}
 01 * 87 &= 1\ 0\ 0\ 0\ 0\ 1\ 1\ 1 \\
 02 * 6E &= 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0 \\
 03 * 46 &= 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0 \\
 01 * A6 &= 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0
 \end{aligned} \right\} \text{Perform Ex-or}$$

AES Mix Column

2	3	1	1
<u>1</u>	<u>2</u>	<u>3</u>	<u>1</u>
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47			
37			

$$\{01\} * \{87\} \oplus \{02\} * \{6E\} \oplus \{03\} * \{46\} \oplus \{01\} * \{A6\} = \{37\}$$

$$\begin{aligned}
 01 * 87 &= 1\ 0\ 0\ 0\ 0\ 1\ 1\ 1 \\
 02 * 6E &= 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0 \\
 03 * 46 &= 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0 \\
 01 * A6 &= 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0 \\
 \hline
 &= 0\ 0\ 1\ 1\ 0\ 1\ 1\ 1 = \{37\} \\
 &\quad \quad \quad \underline{\quad 3 \quad} \quad \underline{\quad 7 \quad}
 \end{aligned}$$

AES Mix Column

2	3	1	1
1	2	3	1
<u>1</u>	<u>1</u>	<u>2</u>	<u>3</u>
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47			
37			
?			

$$\{01\} * \{87\} \oplus \{01\} * \{6E\} \oplus \{02\} * \{46\} \oplus \{03\} * \{A6\}$$

$$01 * 87 = 87 = 1000\ 0111$$

$$01 * 6E = 6E = 0110\ 1110$$

$$02 = 0000\ 0010 = X$$

$$46 = 0100\ 0110 = X^6 + X^2 + X$$

$$\begin{aligned} 02 * 46 &= X * (X^6 + X^2 + X) \\ &= X^7 + X^3 + X^2 \\ &= 1000\ 1100 \end{aligned}$$

$$03 = 0000\ 0110 = X + 1$$

$$A6 = 1010\ 0110 = X^7 + X^5 + X^2 + X$$

$$\begin{aligned} 03 * A6 &= (X + 1) * (X^7 + X^5 + X^2 + X) \\ &= \cancel{X^8} + X^6 + X^3 + \cancel{X^2} + X^7 + X^5 + \cancel{X^2} + X \\ &= X^4 + \cancel{X^3} + \cancel{X} + 1 + X^6 + \cancel{X^3} + X^7 + X^5 + \cancel{X} \\ &= X^7 + X^6 + X^5 + X^4 + 1 \\ &= 1111\ 0001 \end{aligned}$$

Use irreducible Polynomial Theorem, GF (2⁸)

$$X^8 = X^4 + X^3 + X + 1$$



AES Mix Column

2	3	1	1
1	2	3	1
<u>1</u>	<u>1</u>	<u>2</u>	<u>3</u>
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47			
37			
?			

$$\{01\} * \{87\} \oplus \{01\} * \{6E\} \oplus \{02\} * \{46\} \oplus \{03\} * \{A6\}$$

$$01 * 87 = 1\ 0\ 0\ 0\ 0\ 1\ 1\ 1$$

$$01 * 6E = 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0$$

$$02 * 46 = 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0$$

$$03 * A6 = 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1$$

Perform Ex-or

AES Mix Column

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47			
37			
94			

$$\{01\} * \{87\} \oplus \{01\} * \{6E\} \oplus \{02\} * \{46\} \oplus \{03\} * \{A6\} = \{94\}$$

$$01 * 87 = 1000\ 0111$$

$$01 * 6E = 0110\ 1110$$

$$02 * 46 = 1000\ 1100$$

$$03 * A6 = 1111\ 0001$$

$$\begin{array}{r}
 1000\ 0111 \\
 0110\ 1110 \\
 1000\ 1100 \\
 1111\ 0001 \\
 \hline
 1001\ 0100 \\
 \hline
 \end{array}
 = \{94\}$$

9
4

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47			
37			
94			
?			

$$\{03\} * \{87\} \oplus \{01\} * \{6E\} \oplus \{01\} * \{46\} \oplus \{02\} * \{A6\}$$

$$03 = 0000\ 0011 = X + 1$$

$$87 = 1000\ 0111 = X^7 + X^2 + X + 1$$

$$\begin{aligned}
 03 * 87 &= (X + 1) * (X^7 + X^2 + X + 1) \\
 &= \boxed{X^8} + X^3 + \cancel{X^2} + \cancel{X} + X^7 + \cancel{X^2} + \cancel{X} + 1 \\
 &= X^4 + \cancel{X^3} + X + \cancel{1} + X^3 + X^7 + \cancel{1} \\
 &= X^7 + X^4 + X \\
 &= 1001\ 0010
 \end{aligned}$$

$$01 * 6E = 6E = 0110\ 1110$$

$$01 * 46 = 46 = 0100\ 0110$$

$$02 = 0000\ 0010 = X$$

$$A6 = 1010\ 0110 = X^7 + X^5 + X^2 + X$$

$$\begin{aligned}
 02 * A6 &= X * (X^7 + X^5 + X^2 + X) \\
 &= \boxed{X^8} + X^6 + X^3 + X^2 \\
 &= X^4 + \cancel{X^3} + X + 1 + X^6 + \cancel{X^3} + X^2 \\
 &= X^6 + X^4 + X^2 + X + 1 \\
 &= 0101\ 0111
 \end{aligned}$$

Use irreducible Polynomial Theorem, GF (2⁸)

$$X^8 = X^4 + X^3 + X + 1$$



2	3	1	1		87	F2	4D	97		47			
1	2	3	1	*	6E	4C	90	EC		37			
1	1	2	3		46	E7	4A	C3	=	94			
<u>3</u>	<u>1</u>	<u>1</u>	<u>2</u>		A6	8C	D8	95		?			

$$\{03\} * \{87\} \oplus \{01\} * \{6E\} \oplus \{01\} * \{46\} \oplus \{02\} * \{A6\}$$

$$\left. \begin{array}{l} 03 * 87 = 10010010 \\ 01 * 6E = 01101110 \\ 01 * 46 = 01000110 \\ 02 * A6 = 01010111 \end{array} \right\} \text{Perform Ex-or}$$

2	3	1	1		87	F2	4D	97		47			
1	2	3	1	*	6E	4C	90	EC		37			
1	1	2	3		46	E7	4A	C3	=	94			
<u>3</u>	<u>1</u>	<u>1</u>	<u>2</u>		A6	8C	D8	95		ED			

$$\{03\} * \{87\} \oplus \{01\} * \{6E\} \oplus \{01\} * \{46\} \oplus \{02\} * \{A6\} = \{ED\}$$

$$\begin{array}{l} 03 * 87 = 10010010 \\ 01 * 6E = 01101110 \\ 01 * 46 = 01000110 \\ 02 * A6 = 01010111 \\ \hline = \begin{array}{cc} 1110 & 1101 \\ \text{E} & \text{D} \end{array} = \{ED\} \end{array}$$

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47	?		
37			
94			
ED			

$$\{02\} * \{F2\} \oplus \{03\} * \{4C\} \oplus \{01\} * \{E7\} \oplus \{01\} * \{8C\}$$

$$02 = 0000\ 0010 = X$$

$$F2 = 1111\ 0010 = X^7 + X^6 + X^5 + X^4 + X$$

$$\begin{aligned} 02 * F2 &= X * (X^7 + X^6 + X^5 + X^4 + X) \\ &= X^8 + X^7 + X^6 + X^5 + X^2 \\ &= X^4 + X^3 + X + 1 + X^7 + X^6 + X^5 + X^2 \\ &= X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X + 1 \\ &= 1111\ 1111 \end{aligned}$$

$$03 = 0000\ 0011 = X + 1$$

$$4C = 0100\ 1100 = X^6 + X^3 + X^2$$

$$\begin{aligned} 03 * 4C &= (X+1) * (X^6 + X^3 + X^2) \\ &= X^7 + X^4 + \cancel{X^3} + X^6 + \cancel{X^3} + X^2 \\ &= X^7 + X^6 + X^4 + X^2 \\ &= 1101\ 0100 \end{aligned}$$

$$01 * E7 = E7 = 1110\ 0111$$

$$01 * 8C = 8C = 1000\ 1100$$

Use irreducible Polynomial Theorem, GF (2³)

$$X^8 = X^4 + X^3 + X + 1$$

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47	?		
37			
94			
ED			

$$\{02\} * \{F2\} \oplus \{03\} * \{4C\} \oplus \{01\} * \{E7\} \oplus \{01\} * \{8C\}$$

$$\left. \begin{aligned} 02 * F2 &= 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ 03 * 4C &= 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0 \\ 01 * E7 &= 1\ 1\ 1\ 0\ 0\ 1\ 1\ 1 \\ 01 * 8C &= 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0 \end{aligned} \right\} \text{Perform Ex-or}$$

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47	40		
37			
94			
ED			

$$\{02\} * \{F2\} \oplus \{03\} * \{4C\} \oplus \{01\} * \{E7\} \oplus \{01\} * \{8C\} = \{40\}$$

$$\begin{aligned}
 02 * F2 &= 11111111 \\
 03 * 4C &= 11010100 \\
 01 * E7 &= 11100111 \\
 01 * 8C &= 10001100 \\
 \hline
 &= 01000000 = \{40\}
 \end{aligned}$$

4 0

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47	40		
37	?		
94			
ED			

$$\{01\} * \{F2\} \oplus \{02\} * \{4C\} \oplus \{03\} * \{E7\} \oplus \{01\} * \{8C\}$$

❖ Like this we convert all byte of 4 * 4 matrix and final byte will be as...

AES Mix Column

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	?

$$\{03\} * \{97\} \oplus \{01\} * \{EC\} \oplus \{01\} * \{C3\} \oplus \{02\} * \{95\}$$

$$03 = 00000011 = X + 1$$

$$97 = 10010111 = X^7 + X^4 + X^2 + X + 1$$

$$\begin{aligned}
 03 * 97 &= (X + 1) * (X^7 + X^4 + X^2 + X + 1) \\
 &= X^8 + X^5 + X^3 + X^2 + X + X^7 + X^4 + X^2 + X + 1 \\
 &= X^4 + X^3 + X + X^7 + X^5 + X^4 + X^3 + 1 \\
 &= X^7 + X^5 + X \\
 &= 10100010
 \end{aligned}$$

$$01 * EC = EC = 11101100$$

$$01 * C3 = C3 = 11000011$$

$$02 = 00000010 = X$$

$$95 = 10010101 = X^7 + X^4 + X^2 + 1$$

$$\begin{aligned}
 02 * 95 &= X * (X^7 + X^4 + X^2 + 1) \\
 &= X^8 + X^5 + X^3 + X \\
 &= X^4 + X^3 + X + 1 + X^5 + X^3 + X \\
 &= X^5 + X^4 + 1 \\
 &= 00110001
 \end{aligned}$$

Use irreducible Polynomial Theorem, GF(2⁸)

$$X^8 = X^4 + X^3 + X + 1$$



AES Mix Column

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

=

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$$\{03\} * \{97\} \oplus \{01\} * \{EC\} \oplus \{01\} * \{C3\} \oplus \{02\} * \{95\} = \{BC\}$$

$$03 * 97 = 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0$$

$$01 * EC = 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0$$

$$01 * C3 = 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1$$

$$02 * 95 = 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1$$

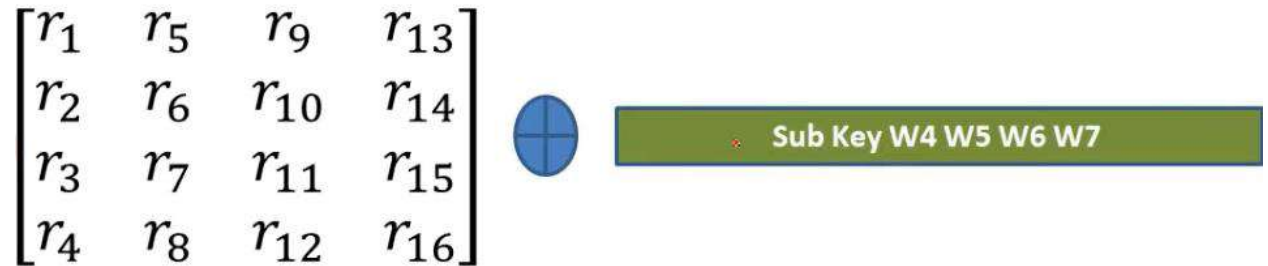
$$= \begin{array}{c} \hline 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0 \\ \hline \end{array} = \{BC\}$$

B
C

- ❖ So, output of Mix column will become input to 4th transformation of round function i.e., **Add Round key**

Add Round key

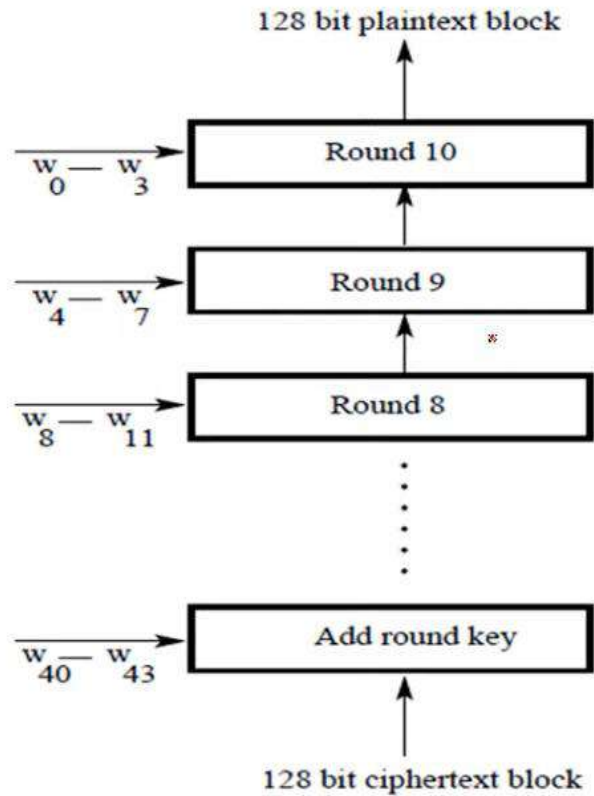
- In the forward add round key transformation, called AddRoundKey, the 128 bits of State are bitwise XORed with the 128 bits of the round key.
- For round 1 it is W4, W5, W6, W7 (sub key) is used.



- And output of round 1 i.e., 4*4 matrix will be input of round 2.
- Like this we have to perform up to N-1 Round.
- At the last Round N, the output of Round N - 1 is the input of Round N which is the last round and it has only 3 Transformation i.e., Substitute Bytes (Sub-Bytes), Shift Rows and Add Round Key. We do not perform Mix columns in last round.
- And output of last round is Cipher text.

AES Decryption

Decryption



Decryption -Round

- Substitute Bytes (Sub-Bytes)
- Shift Rows
- Mix Columns – (Not applicable for last round)
- Add Round Key

Substitute Bytes (Sub-Bytes)

- **Substitute Bytes**- an inverse S box is used for Bytes Substitution

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

- **Shift Rows** – Rows are shifted right in decryption
 - First row: No change
 - Second row: one-byte cyclical right shift
 - Third row: two - byte cyclical right shift
 - Fourth row: three - byte cyclical right shift

$$\begin{bmatrix} 63 & 47 & a2 & f0 \\ 9c & 63 & c5 & f2 \\ 7b & 7c & f0 & ab \\ ca & af & 76 & 76 \end{bmatrix}$$

$$\begin{bmatrix} 63 & 47 & a2 & f0 \\ f2 & 9c & 63 & c5 \\ f0 & ab & 7b & 7c \\ af & 76 & 76 & ca \end{bmatrix}$$

- Mix Columns

We use pre-defined 4*4 matrix in encryption

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} * \begin{bmatrix} 63 & 47 & a2 & f0 \\ 9c & 63 & c5 & f2 \\ 7b & 7c & f0 & ab \\ ca & af & 76 & 76 \end{bmatrix}$$

Here we use another pre-defined 4*4 matrix in decryption and multiply with output of shift rows state array matrix.

0E	0B	0D	09
09	0E	0B	0D
0D	09	0E	0B
0B	0D	09	0E

$$* \begin{bmatrix} r_1 & r_5 & r_9 & r_{13} \\ r_2 & r_6 & r_{10} & r_{14} \\ r_3 & r_7 & r_{11} & r_{15} \\ r_4 & r_8 & r_{12} & r_{16} \end{bmatrix}$$

- Add Round Key

- ✓ Here output of mix columns will be xored with subkey of round 1 i.e., W36, W37, W38 and W39.
- ✓ And after performing all round we get plaintext.

Number theory

- Number theory forms a crucial foundation for modern cryptography and plays a vital role in cybersecurity.
- It provides the mathematical tools and principles necessary for secure communication and data protection.
- Most of the encryption is based heavily on number theory and abstract algebra concept.
- Here's how key concepts from number theory are applied:
 1. Prime Numbers
 - ✓ **Public key cryptography**, like RSA, relies heavily on large prime numbers.
 2. Modular Arithmetic
 - ✓ Operations like $a \bmod n$ are central in encryption algorithms.
 - ✓ **Diffie-Hellman key exchange**, RSA, and elliptic curve cryptography all use modular arithmetic.
 3. Greatest Common Divisor (GCD)
 - ✓ Used to ensure keys are **co-prime** in RSA, meaning they share no common factors with the modulus.
 - ✓ **Euclidean algorithm** helps find GCD efficiently.
 4. Euler's Totient Function ($\phi(n)$)
 - ✓ **Critical in RSA for computing the private key.**

Prime Number

- A **prime number** is a natural number greater than 1 that has **exactly two distinct factors**: 1 and itself.
- A **factor** of a number is another number that divides it **exactly**—that is, with **no remainder**.

The factors of 6 are: 1, 2, 3, 6

Because:

$$1 \times 6 = 6$$
$$2 \times 3 = 6$$
- Prime number plays very important role in cryptography.

- Examples of prime number: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, ...
- **Note:** 1 is **not** a prime number because it has only one factor.
- In **RSA encryption**, two large prime numbers are multiplied to create a modulus. The difficulty of factoring this large number (product of primes) ensures security.
- **Facts about prime number**
 - ✓ Only even prime: 2
 - ✓ Smallest prime number: 2
 - ✓ Except for 2 and 5, all prime number ends in the digit 1,3,7 or 9

Why prime numbers in cryptography?

- ✓ Many encryption algorithms are based on prime numbers.
- ✓ Very fast to multiply two large prime numbers.
- ✓ Extremely computer-intensive to do the reverse.
- ✓ Factoring very large prime numbers is very hard i.e., take computers a long time.

Composite Number

- ✓ Numbers that have more than 2 factors are called as Composite numbers
- ✓ Positive Integers that have more than 2 factors
- ✓ Numbers that are divisible by more than two numbers.
- ✓ 4, 6, 8, 9, 10, 12, 14, 15, 16, ...

Greatest Common Divisor (GCD)

- ✓ GCD of two or more integers which are not zero is the greatest positive integer that divides each of the integers.
- ✓ GCD of two numbers is the greatest number that divides both the number
- ✓ **GCD of 12 and 8 is 4**

Here divisor of 12 and 8 is 2 and 4.

We can divide 12 by 2, 4, and 6.

We can divide 8 by 2 and 4. Here **Greatest Common Divisor is 4.**

How to find GCD

- ✓ GCD by Division Method
- ✓ First take two of the given numbers, divide the greater by the smaller number and then divide the divisor by the remainder.
- ✓ The divisor which does not leave a remainder is the GCD of the two numbers.
 1. Find the **GCD** of 30 and 45.
 2. Find the **GCD** of 442 and 546.
 3. Find the **GCD** of 442, 546 and 424.

Euclidean Algorithm

- ❖ Euclidean Algorithm or Euclid's Algorithm.
- ❖ For computing the Greatest Common Divisor (GCD).
- ❖ aka Highest Common Factor (HCF).

Understanding GCD – Example 1

	12	33
Divisors	1, 2, 3, 4, 6, 12	1, 3, 11, 33
Common Divisors	1, 3	
Greatest Common Divisor (GCD)	3	

$$\therefore \text{GCD}(12, 33) = 3$$

Understanding GCD – Example 2

	25	150
Divisors	1, 5, 25	1, 2, 3, 5, 6, 10, 15, 25, 30, 50, 75, 150
Common Divisors	1, 5, 25	
Greatest Common Divisor (GCD)	25	

$$\therefore \text{GCD}(25, 150) = 25$$

Understanding GCD – Example 3

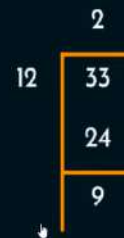
	13	31
Divisors	1, 13	1, 31
Common Divisors	1	
Greatest Common Divisor (GCD)	1	

$$\therefore \text{GCD}(13, 31) = 1$$

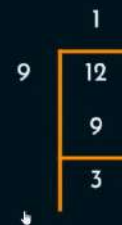
Euclid's Algorithm for finding GCD

Find the GCD(12, 33).

Q	A	B	R
2	33	12	9



Q	A	B	R
2	33	12	9
	12	9	



Q	A	B	R
2	33	12	9
1	12	9	3

Q	A	B	R
2	33	12	9
1	12	9	3
3	9	3	0

	3
3	9
	9
	0



Euclid's Algorithm for finding GCD

$$\text{GCD}(12, 33) = 3.$$

Q	A	B	R
2	33	12	9
1	12	9	3
3	9	3	0
X	3	0	X

0	3



Euclid's Algorithm for finding GCD

$$\text{GCD}(750, 900) = 150.$$

Q	A	B	R
1	900	750	150
5	750	150	0
X	150	0	X



Euclid's Algorithm for finding GCD

$$\text{GCD}(252, 105) = 21.$$

Q	A	B	R
2	252	105	42
2	105	42	21
2	42	21	0
X	21	0	X



Find the GCD(1005, 105).

Euclid's Algorithm for finding GCD

Prerequisite: $a > b$

Euclid_GCD (a, b):

if $b = 0$ then

return a ;

else

return Euclid_GCD ($b, a \bmod b$);

Euclid's Algorithm – Example 1

Example 1: Find the GCD (50, 12).

Solution:

Here $a=50, b=12$

$$\text{GCD}(a, b) = \text{GCD}(b, a \bmod b)$$

$$\text{GCD}(50, 12) = \text{GCD}(12, 50 \bmod 12) = \text{GCD}(12, 2)$$

$$\text{GCD}(12, 2) = \text{GCD}(2, 12 \bmod 2) = \text{GCD}(2, 0) = 2$$

$$\text{GCD}(50, 12) = 2$$

Euclid's Algorithm – Example 2

Example 2: Find the GCD (83, 19).

Solution:

Here $a=83, b=19$

$$\text{GCD}(a, b) = \text{GCD}(b, a \bmod b)$$

$$\text{GCD}(83, 19) = \text{GCD}(19, 83 \bmod 19) = \text{GCD}(19, 7)$$

$$\text{GCD}(19, 7) = \text{GCD}(7, 19 \bmod 7) = \text{GCD}(7, 5)$$

$$\text{GCD}(7, 5) = \text{GCD}(5, 7 \bmod 5) = \text{GCD}(5, 2)$$

$$\text{GCD}(5, 2) = \text{GCD}(2, 5 \bmod 2) = \text{GCD}(2, 1)$$

$$\text{GCD}(2, 1) = \text{GCD}(1, 2 \bmod 1) = \text{GCD}(1, 0) = 1$$

$$\text{GCD}(83, 19) = 1$$

Find the GCD (529, 123).

Co-Prime Number or relatively prime

- Set of numbers that have GCD as 1 are called as Co-Prime Numbers.
- For instance, 7 and 8 are co-prime numbers.

Here factor of 7 is 1 and 7

And factor of 8 is 1, 2, 4 and 8 and in both there is only one common factor that is 1 so these two numbers 7 and 8 are co-prime number.

Two numbers are said to be relatively prime, if they have no prime factors in common, and their only common factor is 1.

- ❖ If $\text{GCD}(a, b) = 1$ then 'a' and 'b' are relatively prime numbers.
- ❖ Co-prime.

Question 1: Are 4 and 13 relatively prime?

Solution:

	4	13
Divisors	1, 2, 4	1, 13
Common Divisors	1	
Greatest Common Divisor (GCD)	1	

$$\text{GCD}(4, 13) = 1$$

Yes, 4 and 13 are relatively prime numbers.

Question 2: Are 15 and 21 relatively prime?

Solution:

	15	21
Divisors	1, 3, 5, 15	1, 3, 7, 21
Common Divisors	1, 3	
Greatest Common Divisor (GCD)	3	

$$\text{GCD}(15, 21) = 3$$

No, 15 and 21 are not relatively prime numbers.

Relatively Prime Numbers

a	b	GCD(a, b)	Relatively Prime?	Remarks
11	17	1	Yes	'a' and 'b' are prime
11	21	1	Yes	'a' is prime and 'b' is composite
12	77	1	Yes	'a' and 'b' are composite

Find the GCD(790, 121) using Euclid's algorithm and determine whether they are relatively prime or not.

Properties of Co-Prime Numbers

- 1 is coprime with every other number
- Prime numbers are co-prime to each other .example 3,5
- Any two successive numbers are always co-prime. example 18,19
- The sum of any two co-prime numbers is always co-prime with their product. example 2 and 3.

$$2+3=5$$

$$2*3=6, 5 \text{ and } 6 \text{ are co-prime.}$$

Euler's Totient Function (Phi Function)

Euler's Totient Function

- ❖ Denoted as $\Phi(n)$.
- ❖ $\Phi(n)$ = Number of positive integers less than 'n' that are relatively prime to n.

Solution:

Here $n=5$.

Numbers less than 5 are 1, 2, 3 and 4.

GCD	Relatively Prime?
GCD (1, 5) = 1	✓
GCD (2, 5) = 1	✓
GCD (3, 5) = 1	✓
GCD (4, 5) = 1	✓

$\therefore \Phi(5) = 4$.

Example 2: Find $\Phi(11)$.

Solution:

Here $n=11$.

Numbers less than 11 are 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10.

GCD	Relatively Prime?
GCD (1, 11) = 1	✓
GCD (2, 11) = 1	✓
GCD (3, 11) = 1	✓
GCD (4, 11) = 1	✓
GCD (5, 11) = 1	✓

GCD	Relatively Prime?
GCD (6, 11) = 1	✓
GCD (7, 11) = 1	✓
GCD (8, 11) = 1	✓
GCD (9, 11) = 1	✓
GCD (10, 11) = 1	✓

$\therefore \Phi(11) = 10$.

Example 3: Find $\Phi(8)$.

Solution:

Here $n=8$.

Numbers less than 8 are 1, 2, 3, 4, 5, 6, and 7.

GCD	Relatively Prime?
GCD (1, 8) = 1	✓
GCD (2, 8) = 2	✗
GCD (3, 8) = 1	✓
GCD (4, 8) = 4	✗

GCD	Relatively Prime?
GCD (5, 8) = 1	✓
GCD (6, 8) = 2	✗
GCD (7, 8) = 1	✓

$\therefore \Phi(8) = 4$.

Euler's Totient Function

$\Phi(n)$	Criteria of 'n'	Formula
	'n' is prime.	$\Phi(n) = (n-1)$
	$n = p \times q$. 'p' and 'q' are primes.	$\Phi(n) = (p-1) \times (q-1)$
	$n = a \times b$. Either 'a' or 'b' is composite. Both 'a' and 'b' are composite.	$\Phi(n) = n \times \left(1 - \frac{1}{p_1}\right) \left(1 - \frac{1}{p_2}\right) \dots$ where $p_1, p_2, \dots$ are distinct primes.

Euler's Totient Function

Example 1: Find $\Phi(5)$.

Solution:

Here $n=5$.

'n' is a prime number.

$$\Phi(n) = (n-1)$$

$$\Phi(5) = (5-1)$$

$$\Phi(5) = 4$$

So, there are 4 numbers that are lesser than 5 and relatively prime to 5.

Euler's Totient Function

Example 2: Find $\Phi(31)$.

Solution:

Here $n=31$.

'n' is a prime number.

$$\Phi(n) = (n-1)$$

$$\Phi(31) = (31-1)$$

$$\Phi(31) = 30$$

So, there are 30 numbers that are lesser than 31 and relatively prime to 31.

Euler's Totient Function

Example 3: Find $\Phi(35)$.

Solution:

Here $n=35$.

'n' is a product of two prime numbers 5 and 7.

Let us assign $p=5$ and $q=7$.

$$\Phi(n) = (p-1) \times (q-1)$$

$$\Phi(35) = (5-1) \times (7-1)$$

$$\Phi(35) = 4 \times 6$$

$$\Phi(35) = 24$$

So, there are 24 numbers that are lesser than 35 and relatively prime to 35.

Euler's Totient Function

GCD	RP?
GCD(1,35)	✓
GCD(2,35)	✓
GCD(3,35)	✓
GCD(4,35)	✓
GCD(5,35)	✗
GCD(6,35)	✓
GCD(7,35)	✗
GCD(8,35)	✓
GCD(9,35)	✓

GCD	RP?
GCD(10,35)	✗
GCD(11,35)	✓
GCD(12,35)	✓
GCD(13,35)	✓
GCD(14,35)	✗
GCD(15,35)	✗
GCD(16,35)	✓
GCD(17,35)	✓
GCD(18,35)	✓

GCD	RP?
GCD(19,35)	✓
GCD(20,35)	✗
GCD(21,35)	✗
GCD(22,35)	✓
GCD(23,35)	✓
GCD(24,35)	✓
GCD(25,35)	✗
GCD(26,35)	✓
GCD(27,35)	✓

GCD	RP?
GCD(28,35)	✗
GCD(29,35)	✓
GCD(30,35)	✗
GCD(31,35)	✓
GCD(32,35)	✓
GCD(33,35)	✓
GCD(34,35)	✓

24

Euler's Totient Function

Example 4: Find $\Phi(1000)$.

Solution:

Here $n = 1000 = 2^3 \times 5^3$.

Distinct prime factors are 2 and 5.

$$\Phi(n) = n \times \left(1 - \frac{1}{p_1}\right) \left(1 - \frac{1}{p_2}\right) \dots$$

$$\Phi(1000) = 1000 \times \left(1 - \frac{1}{2}\right) \left(1 - \frac{1}{5}\right)$$

$$\Phi(1000) = 1000 \times \left(\frac{1}{2}\right) \left(\frac{4}{5}\right)$$

$$\Phi(1000) = 400$$

Euler's Totient Function

Example 5: Find $\Phi(7000)$.

Solution:

Here $n = 7000 = 2^3 \times 5^3 \times 7^1$

Distinct prime factors are 2, 5 and 7.

$$\Phi(n) = n \times \left(1 - \frac{1}{p_1}\right) \left(1 - \frac{1}{p_2}\right) \left(1 - \frac{1}{p_3}\right) \dots$$

$$\Phi(7000) = 7000 \times \left(1 - \frac{1}{2}\right) \left(1 - \frac{1}{5}\right) \left(1 - \frac{1}{7}\right)$$

$$\Phi(7000) = 7000 \times \left(\frac{1}{2}\right) \left(\frac{4}{5}\right) \left(\frac{6}{7}\right)$$

$$\Phi(7000) = 2400$$

1. Find $\Phi(369)$.

2. Find $\Phi(372)$.

Euler's Theorem

- **Euler's Theorem** is a key concept in number theory, named after the Swiss mathematician **Leonhard Euler**.
- It states that if **a** is any integer that is **coprime** with a positive integer **n**, then:

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

- This means that raising **a** to the power of $\phi(n)$ (Euler's Totient Function) will always leave a remainder of 1 when divided by **n**.
- This is generalized version of **Fermat's Little Theorem**.
- Interestingly, **Fermat's Little Theorem** is just a special case of Euler's Theorem. When **n** is a **prime number p**, $\phi(p) = p - 1$, so Euler's Theorem becomes Fermat's Theorem.

Example:

Let's say $a = 2$ and $n = 5$.

1. Calculate $\phi(5)$:

Euler's totient function, $\phi(n)$, counts the number of positive integers less than or equal to n that are relatively prime to **n**. Since 5 is a prime number, all integers from 1 to 4 are relatively prime to it. Therefore, $\phi(5) = 4$.

2. Apply Euler's Theorem:

According to Euler's theorem, $2^{\phi(5)} \equiv 1 \pmod{5}$. Substituting $\phi(5) = 4$, we get $2^4 \equiv 1 \pmod{5}$.

3. Verify the result:

$2^4 = 16$. When 16 is divided by 5, the remainder is 1. So, $16 \equiv 1 \pmod{5}$ is true, which confirms Euler's theorem.

Fermat's Theorem

- It is the specific version of Euler's Theorem, i.e., if one number among the 2 coprime numbers is prime, then this Theorem is applied.
- Fermat's theorem is also called a Fermat's little theorem defines that if P is prime and ' a ' is a positive integer not divisible by P then –

- $a^{P-1} \equiv 1 \pmod{P}$

- Another form is $a^P \equiv a \pmod{P}$.

Example $a=3$, $p=5$

Example 1: Does Fermat's theorem hold true for $p=5$ and $a=2$?

Solution:

Given: $p=5$ and $a=2$.

$$a^{p-1} \equiv 1 \pmod{p}$$

$$2^{5-1} \equiv 1 \pmod{5}$$

$$2^4 \equiv 1 \pmod{5}$$

$$16 \equiv 1 \pmod{5}$$

Therefore, Fermat's theorem holds true for $p=5$ and $a=2$.

Example 3: Prove Fermat's theorem does not hold for $p=6$ and $a=2$.

Solution:

$$a^{p-1} \equiv 1 \pmod{p}$$

$$2^{6-1} \equiv 1 \pmod{6}$$

$$2^5 \equiv 1 \pmod{6}$$

$$32 \equiv 1 \pmod{6}$$

$$32 \equiv 2 \pmod{6}$$

Therefore, Fermat's theorem does not hold true for $p=6$ and $a=2$.

Primality Testing

- Primality testing is the problem of determining whether a given positive integer is a prime number (divisible only by 1 and itself) or a composite number (having more than two divisors). This field is crucial in modern cryptography, particularly for generating keys in systems like RSA, which rely on large prime numbers.
- **Basic Methods of Primality Testing**
 1. Trial Division (Basic and Deterministic)
 2. Fermat Primality Test
 3. Miller-Rabin Primality Test
 4. AKS Primality Test (Agrawal–Kayal–Saxena)

Fermat Primality Test

- The Fermat Primality Test is a probabilistic method for determining whether a given number is prime.
- It's based on Fermat's Little Theorem, a fundamental result in number theory.
- **As we know Fermat's Little Theorem**

- ✓ Fermat's Little Theorem states that if p is a prime number, then for any integer a not divisible by p (i.e., $\gcd(a, p) = 1$), the following congruence holds:

- $a^{p-1} \equiv 1 \pmod{p}$

- ✓ This means that if you raise a to the power of $p-1$ and then divide by p , the remainder will be 1.

How the Fermat Primality Test Works

- **Choose a number to test:** Let n be the odd integer you want to test for primality.
- **Choose a random base:** Select a random integer a such that $1 < a < n-1$.

Is ' p ' prime?

Test:

$a^p - a \rightarrow$ ' p ' is prime if this is a multiple of ' p ' for all $1 \leq a < p$.

Example

Question 1: Is 5 prime?

Solution:

$a^p - a \rightarrow$ ' p ' is prime if this is a multiple of ' p ' for all $1 \leq a < p$.

$$1^5 - 1 = 1 - 1 = 0$$

$$2^5 - 2 = 32 - 2 = 30$$

$$3^5 - 3 = 243 - 3 = 240$$

$$4^5 - 4 = 1024 - 4 = 1020$$

$\therefore 5$ is prime

Example

Question 2: Is 3753 prime?

Solution:

$a^p - a \rightarrow 'p' \text{ is prime if this is a multiple of 'p' for all } 1 \leq a < p$

$$1^{3753} - 1$$

$$2^{3753} - 2$$

$$3^{3753} - 3$$

$$4^{3753} - 4$$

...

$$3752^{3753} - 3752$$



Miller-Rabin Primality Test

Miller-Rabin Primality Test



Michael Oser Rabin



Gary Lee Miller

- ❖ Miller-Rabin primality test or Rabin-Miller primality test.
- ❖ Probabilistic primality test.
- ❖ Similar to Fermat primality test and the Solovay-Strassen primality test.
- ❖ Checks whether a specific property, which is known to hold for prime values, holds for the number under testing.

Algorithm

Step 1: Find $n-1 = 2^k \times m$

Step 2: Choose 'a' such that $1 < a < n-1$

Step 3: Compute $b_0 = a^m \pmod{n}$, ... , $b_i = b_{i-1}^2 \pmod{n}$

+1 $\rightarrow$ Composite

-1 $\rightarrow$ Probably Prime

Example

Question: Is 561 prime?

Solution:

Given $n = 561$.

Step 1:

$$n-1 = 2^k \times m$$

$$560 = 2^4 \times 35$$

So $k = 4$, and $m = 35$

$$\frac{560}{2^1} = 280 \quad \left| \quad \frac{560}{2^2} = 140 \quad \left| \quad \frac{560}{2^3} = 70 \quad \left| \quad \frac{560}{2^4} = 35 \quad \left| \quad \frac{560}{2^5} = 17.5 \right.$$

Example

Question: Is 561 prime?

Solution:

Given $n = 561$.

Step 2:

Choosing $a = 2$; $1 < 2 < 560$

Example

Question: Is 561 prime?

Solution:

Given $n = 561$.

Step 3:

Compute $b_0 = a^m \pmod{n}$

$$b_0 = a^m \pmod{n}$$

$$b_0 = 2^{35} \pmod{561} = 263$$

Is $b_0 = \pm 1 \pmod{561}$? **NO**

So calculate b_1

$$b_1 = b_0^2 \pmod{n}$$

$$b_1 = 263^2 \pmod{561}$$

$$b_1 = 166$$

Is $b_1 = \pm 1 \pmod{561}$? **NO**

$$b_2 = b_1^2 \pmod{n}$$

$$b_2 = 166^2 \pmod{561}$$

$$b_2 = 67$$

Is $b_2 = \pm 1 \pmod{561}$? **NO**

$$b_3 = b_2^2 \pmod{n}$$

$$b_3 = 67^2 \pmod{561}$$

$$b_3 = 1 \rightarrow \text{Composite}$$

$\therefore 561$ is composite.

❖ Assignment Modular Exponentiation

What is the Extended Euclidean Algorithm (EEA)?

- It's an **extension** of the standard **Euclidean Algorithm** - which is used to find the **greatest common divisor (gcd)** of two integers.
- The **EEA** goes a step further:
It also finds **integers “s”** and **“t”** such that:
- Using the E.E.A. with input a and b, we can calculate:
 - ✓ The gcd of **a** and **b**
 - ✓ **s** and **t** such that
 $s * a + t * b = \text{gcd}(a, b)$
- This is called **Bézout's Identity**, and **s** and **t** are known as the **Bézout coefficients**.
- Find **s** and **t**
 - ✓ $s = s1 - qs2$
 - ✓ $t = t1 - qt2$

Example: Find the gcd of (161,28) and the value of “s” and “t”.

GCD(161, 28) = ?

q	r1	r2	r	s1	s2	s	t1	t2	t
5	161	28	21	1	0	1	0	1	-5
1	28	21	7	0	1	-1	1	-5	6
3	21	7	0	1	-1	4	-5	6	-23
X	7	0	X	-1	4	X	6	-23	X

GCD(161, 28) = 7

$$\begin{array}{l}
 a \cdot s1 + b \cdot t1 = d \\
 s = s1 - s2 \cdot q \\
 t = t1 - t2 \cdot q \\
 \hline
 s1 = 1; s2 = 0 \\
 t1 = 0; t2 = 1 \\
 \hline
 s = 1 - 0 \times 5 = 1 \\
 t = 0 - 1 \times 5 = -5 \\
 \hline
 s = 0 - 1 \times 1 = -1 \\
 t = 1 - (-5 \times 1) = 6
 \end{array}$$

Formulas

Initialization for first step

- Answer can be tested as $s * a + t * b = \gcd(a, b)$
 $= [(-1) * 161] + [6 * 28]$
 $= 7$

Assignment 1. Use the Extended Euclidean Algorithm to compute: $\gcd(23, 101)$ Also, find integers s and t such that:

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

Assignment 2. Use the Extended Euclidean Algorithm to compute: $\gcd(252, 198)$ Also, find integers s and t such that:

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

Primitive Root

A number ' α ' is a primitive root modulo n if every number coprime to n is congruent to a power of ' α ' modulo n .

Definition made easy:

' α ' is said to be a primitive root of prime number ' p ', if $\alpha^1 \bmod p$, $\alpha^2 \bmod p$, $\alpha^3 \bmod p$, $\dots$, $\alpha^{p-1} \bmod p$ are distinct.

Primitive Root

Example 1: Is 2 a primitive root of prime number 5?

Solution:

$2^1 \bmod 5$	$2 \bmod 5$	2	✓
$2^2 \bmod 5$	$4 \bmod 5$	4	✓
$2^3 \bmod 5$	$8 \bmod 5$	3	✓
$2^4 \bmod 5$	$16 \bmod 5$	1	✓

Yes, 2 is a primitive root of prime number 5.

Example 2: Is 3 a primitive root of prime number 7?

Solution:

$3^1 \bmod 7$	$3 \bmod 7$	3	✓
$3^2 \bmod 7$	$9 \bmod 7$	2	✓
$3^3 \bmod 7$	$6 \bmod 7$	6	✓
$3^4 \bmod 7$	$18 \bmod 7$	4	✓
$3^5 \bmod 7$	$12 \bmod 7$	5	✓
$3^6 \bmod 7$	$15 \bmod 7$	1	✓

Yes, 3 is a primitive root of 7.

Example 3: Is 2 a primitive root of prime number 7?

Solution:

$2^1 \bmod 7$	$2 \bmod 7$	2	✓
$2^2 \bmod 7$	$4 \bmod 7$	4	✓
$2^3 \bmod 7$	$8 \bmod 7$	1	✓
$2^4 \bmod 7$	$16 \bmod 7$	2	✗
$2^5 \bmod 7$	$4 \bmod 7$	4	✗
$2^6 \bmod 7$	$8 \bmod 7$	1	✗

No, 2 is not a primitive root of 7.

Question 1: Is 2 a primitive root of 11?

Question 2: What are the primitive roots of number 5?

Hint:

$1^1 \bmod 5$	$2^1 \bmod 5$	$3^1 \bmod 5$	$4^1 \bmod 5$
$1^2 \bmod 5$	$2^2 \bmod 5$	$3^2 \bmod 5$	$4^2 \bmod 5$
$1^3 \bmod 5$	$2^3 \bmod 5$	$3^3 \bmod 5$	$4^3 \bmod 5$
$1^4 \bmod 5$	$2^4 \bmod 5$	$3^4 \bmod 5$	$4^4 \bmod 5$

Diffie Hellman Key Exchange Algorithm

- The Diffie Hellman algorithm widely known as Key exchange algorithm or key agreement algorithm developed by Whitefield Diffie and Martin Hellman in 1976.
- The purpose of the algorithm is to enable two users to **securely exchange a key** that can be used for subsequent encryption of messages.
- It is a methods to exchange keys if we use symmetric key approach.
- We will use **asymmetric** (public key and private key concept) encryption to exchange the secret key between user.

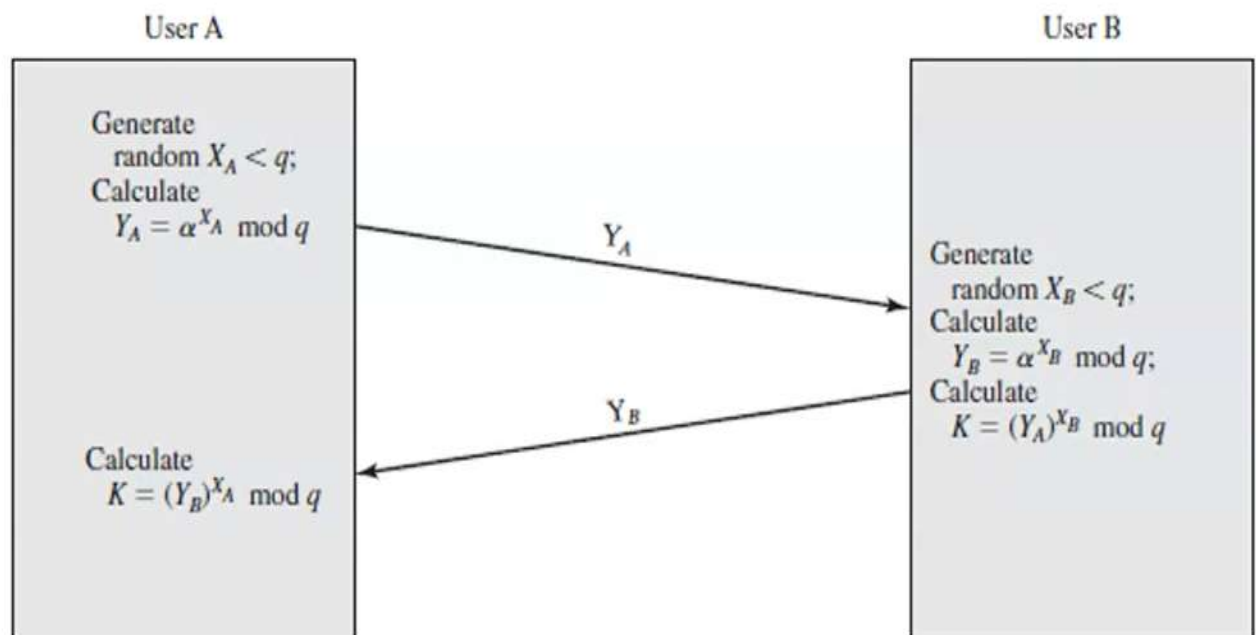


Figure: Diffie-Hellman Key exchange algorithm

It is not an encryption algorithm.

Diffie-Hellman Key Exchange Algorithm

- The algorithm works as follow:

1. Select q (prime number) and α (α is primitive root of q)
2. **User A Key Generation:**
select $X_A, X_A < q$
Calculate public Y_A ,
 $Y_A = \alpha^{X_A} \bmod q$
 Y_A Shared with user B
3. **User B Key Generation:**
select $X_B, X_B < q$
Calculate public Y_B ,
 $Y_B = \alpha^{X_B} \bmod q$,
 Y_B shared with user A
4. Calculation of secret key by **user A** :
 $K = (Y_B)^{X_A} \bmod q$
5. Calculation of secret key by **user B** :
 $K = (Y_A)^{X_B} \bmod q$

Diffie-Hellman Key Exchange Algorithm step by step explanation

➤ **Step – 1:** Select q (prime number) and α (α is primitive root of q)

➤ **Step – 2: User A Key Generation:**

select $X_A, X_A < q$
Calculate public Y_A ,
 $Y_A = \alpha^{X_A} \bmod q$
 Y_A Shared with **user B**

❖ Explanation with example:

1. Here $q = 7, \alpha = 17$.

2. *User A Key Generation*

Here $X_A = 6$, where $X_A < q$

$$Y_A = \alpha^{X_A} \bmod q \rightarrow 17^6 \bmod 7 \rightarrow 1$$

$$Y_A = 1$$

➤ **Step – 3: User B Key Generation:**

select $X_B, X_B < q$

Calculate public Y_B ,

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

Y_B Shared with **user A**

❖ **Explanation with example:**

3. User B Key Generation

Here $X_B = 4$, where $X_B < q$

$$Y_B = \alpha^{X_B} \bmod q \rightarrow 17^4 \bmod 7 \rightarrow 4$$

$$Y_B = 4$$

➤ **Step – 4: Calculation of secret key by **user A** :**

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

➤ **Step – 5: Calculation of secret key by **user B** :**

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

❖ **Explanation with example:**

4. Calculation of secret key by **user A** :

$$K = (Y_B)^{X_A} \bmod q \rightarrow 4^6 \bmod 7 \rightarrow 1$$

$$K = 1$$

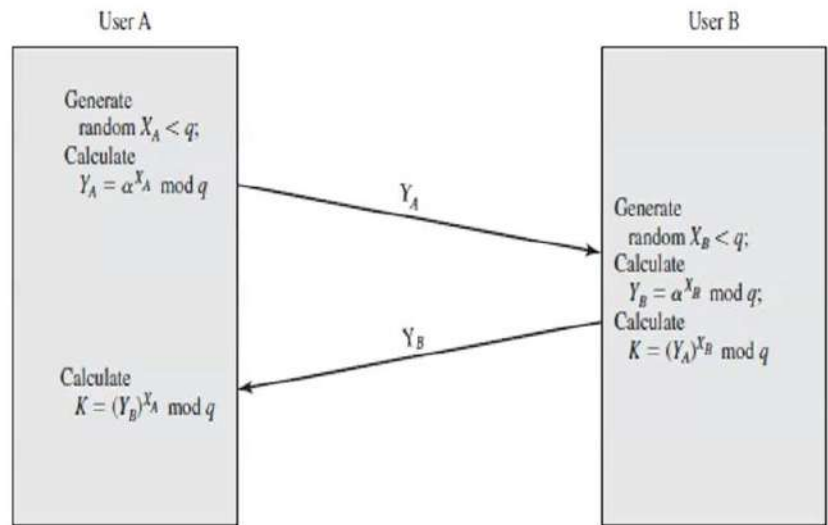
5. Calculation of secret key by **user B** :

$$K = (Y_A)^{X_B} \bmod q \rightarrow 1^4 \bmod 7 \rightarrow 1$$

$$K = 1$$

• Same Secret key generate both sides

$$\begin{aligned}
 K &= (Y_B)^{X_A} \bmod q \\
 &= (a^{X_B} \bmod q)^{X_A} \bmod q \\
 &= (a^{X_B})^{X_A} \bmod q \\
 &= (a^{X_A})^{X_B} \bmod q \\
 &= (a^{X_A} \bmod q)^{X_B} \bmod q \\
 &= (Y_A)^{X_B} \bmod q \\
 &= K
 \end{aligned}$$



Exercise - 1

Exercise – 1: $q = 353$, $a = 3$, $X_A = 97$, $X_B = 233$.

Determine **Public Key** and **Shared key** for both users using Diffie-Hellman key exchange algorithm.

Solution:

- Here $q = 353$, $a = 3$
- Calculate **Public key for user A**. $X_A = 97$,
 $Y_A = a^{X_A} \bmod q = 3^{97} \bmod 353 = 40$
- Calculate **Public key for user B**. $X_B = 233$,
 $Y_B = a^{X_B} \bmod q = 3^{233} \bmod 353 = 248$
- Calculation of **secret key by user A**,
 $K = (Y_B)^{X_A} \bmod q = 248^{97} \bmod 353 = 160$
- Calculation of **secret key by user B**,
 $K = (Y_A)^{X_B} \bmod q = 40^{233} \bmod 353 = 160$

$$Y_A = 40$$

$$Y_B = 248$$

$$K = 160$$

$$K = 160$$

Exercise - 2

Exercise – 2: $q = 23, \alpha = 5, X_A = 6, X_B = 15$.

Determine **Public Key** and **Shared key** for both users using Diffie-Hellman key exchange algorithm.

Solution:

1. Here $q = 23, \alpha = 5$

2. Calculate **Public key for user A**. $X_A = 6$,

$$Y_A = \alpha^{X_A} \bmod q = 5^6 \bmod 23 = 8$$

$$Y_A = 8$$

3. Calculate **Public key for user B**. $X_B = 15$,

$$Y_B = \alpha^{X_B} \bmod q = 5^{15} \bmod 23 = 19$$

$$Y_B = 19$$

4. Calculation of **secret key by user A**,

$$K = (Y_B)^{X_A} \bmod q = 19^6 \bmod 23 = 2$$

$$K = 2$$

5. Calculation of **secret key by user B**,

$$K = (Y_A)^{X_B} \bmod q = 8^{15} \bmod 23 = 2$$

$$K = 2$$

Exercise - 3

Exercise – 3: $q = 11, \alpha = 17, X_A = 8, X_B = 12$.

Determine **Public Key** and **Shared key** for both users using Diffie-Hellman key exchange algorithm.

Solution:

1. Here $q = 11, \alpha = 17$

2. Calculate **Public key for user A**. $X_A = 8$,

$$Y_A = \alpha^{X_A} \bmod q = 17^8 \bmod 11 = 4$$

$$Y_A = 4$$

3. Calculate **Public key for user B**. $X_B = 12$,

$$Y_B = \alpha^{X_B} \bmod q = 17^{12} \bmod 11 = 3$$

$$Y_B = 3$$

4. Calculation of **secret key by user A**,

$$K = (Y_B)^{X_A} \bmod q = 3^8 \bmod 11 = 5$$

$$K = 5$$

5. Calculation of **secret key by user B**,

$$K = (Y_A)^{X_B} \bmod q = 4^{12} \bmod 11 = 5$$

$$K = 5$$

RSA Algorithm

- Ron Rivest, Adi Shamir and Len Adleman have developed this algorithm (Rivest-Shamir-Adleman). It is a block cipher which converts plain text into cipher text and vice versa at receiver side.
- **The algorithm works as follow:**
 1. Select two prime numbers p and q where $p \neq q$.
 2. Calculate $n = p * q$.
 3. Calculate $\Phi(n) = (p-1) * (q-1)$.
 4. Select e such that, e is relatively prime to $\Phi(n)$
i.e. $(e, \Phi(n)) = 1$ and $1 < e < \Phi(n)$
 5. Calculate $d = e^{-1} \bmod \Phi(n)$ or $ed = 1 \bmod \Phi(n)$.
 6. Public key = $\{e, n\}$, private key = $\{d, n\}$.
 7. Find out cipher text using the formula,
 $C = P^e \bmod n$ where, $P < n$ and
 $C = \text{Cipher text}, P = \text{Plain text}, e = \text{Encryption key and } n = \text{block size}.$
 8. $P = C^d \bmod n$. Plain text P can be obtain using the given formula.
where, $d = \text{decryption key}.$

- It is a public key (Asymmetric) encryption technique.
- RSA key Can be typically 2024 or 2048 bits long.

RSA Algorithm step by step explanation

- **Step – 1:** Select two prime numbers p and q where $p \neq q$.
- **Step – 2:** Calculate $n = p * q$.
- **Step – 3:** Calculate $\Phi(n) = (p-1) * (q-1)$.
- **Step – 4:** Select e such that, e is relatively prime to $\Phi(n)$
i.e. $(e, \Phi(n)) = 1$ and $1 < e < \Phi(n)$

❖ Explanation with example:

1. Two prime numbers $p = 13, q = 11$.
2. $n = p * q = 13 * 11 = 143$.
3. $\Phi(n) = (13 - 1) * (12 - 1) = 12 * 10 = 120$.
4. Select $e = 13, \gcd(13, 120) = 1$.

➤ **Step – 5:** Calculate $d = e^{-1} \bmod \Phi(n)$ or $ed = 1 \bmod \Phi(n)$.

❖ Explanation with example:

5. Finding d:

$$\rightarrow e * d \bmod \Phi(n) = 1$$

$$\rightarrow 13 * d \bmod 120 = 1$$

(How to find: $d * e = 1 \bmod \Phi(n) \rightarrow d = ((\Phi(n) * i) + 1) / e$

$$d = (120 + 1) / 13 = 9.30 (\because i = 1)$$

$$d = (240 + 1) / 13 = 18.53 (\because i = 2)$$

$$d = (360 + 1) / 13 = 27.76 (\because i = 3)$$

$$d = (480 + 1) / 13 = 37 (\because i = 4)$$

How to find D

$$d = (\Phi(n) * i + 1) / e$$

Put value of $i = 1, 2, 3, 4, \dots$

Until, integer value of d is found.

$$\begin{aligned} d &= [(20 * 1) + 1] / 7 \\ &= 21/7 = 3 \end{aligned}$$

For exp; if $e=11$

$$\begin{aligned} d &= [(20 * 1) + 1] / 11 \\ &= 21/11 = 1.90 \end{aligned}$$

$$\begin{aligned} d &= [(20 * 2) + 1] / 11 \\ &= 41/11 = 3.7 \end{aligned}$$

➤ **Step – 6:** Public key = {e, n}, private key = {d, n}.

➤ **Step – 7:** Find out cipher text using the formula,

$$C = P^e \bmod n \text{ where, } P < n$$

C = Cipher text, P = Plain text, e = Encryption key and n = block size.

➤ **Step – 8:** $P = C^d \bmod n$. Plain text P can be obtain using the given formula.

where, d = decryption key.

❖ Explanation with example:

6. Public key = {13, 143} and private key = {37, 143}.

7. **Encryption :** Plain text $P = 13$. (where, $P < n$)

$$C = P^e \bmod n = 13^{13} \bmod 143 = 52. \quad \boxed{C = 52}$$

8. **Decryption:**

$$P = C^d \bmod n = 52^{37} \bmod 143 = 13. \quad \boxed{P = 13}$$

Exercise - 1

Question: P and Q are two prime numbers. $P=7$, and $Q=17$. Take public key $E=5$. If plain text value is 6, then what will be cipher text value according to RSA algorithm? Again calculate plain text value from cipher text.

Solution:

1. Two prime numbers $P=7$, $Q=17$

2. $n = P * Q = 17 * 7 = 119$ $\boxed{n = 119}$

3. $\Phi(n) = (P-1) * (Q-1) = (17-1) * (7-1) = 16 * 6 = 96$ $\boxed{\Phi(n) = 96}$

4. Public key $E = 5$. $\boxed{E = 5}$

5. Calculate $d = 77$. $d = ((\Phi(n) * i) + 1) / e$ $\boxed{d = 77}$

$$d = ((96 * 1) + 1) / 5 = 19.4$$

$$d = ((96 * 2) + 1) / 5 = 38.6$$

$$d = ((96 * 3) + 1) / 5 = 57.8$$

$$d = ((96 * 4) + 1) / 5 = 77 \text{ (Stop finding } d \text{ because getting integer value)}$$

6. Public key = {e, n} = {5, 119}, private key = {d, n} = {77, 119}.

7. Plain text $PT = 6$, $CT = PT^E \bmod n = 6^5 \bmod 119 = 41$. $\boxed{\text{Cipher Text} = 41}$

8. Cipher text $CT = 41$, $PT = CT^d \bmod n = 41^{77} \bmod 119 = 6$. $\boxed{\text{Plain Text} = 6}$

Exercise - 2

Question: In a public key cryptosystem using RSA algorithm, user uses two prime numbers 5 and 7. He chooses 11 as Encryption key, find out decryption key. What will be the cipher text, if the plaintext is 2? Decrypt the cipher text, what will be the value of plain text?

Solution:

1. Two prime numbers $p = 5, q = 7$
2. $n = p * q = 5 * 7 = 35$ **$n = 35$**
3. $\Phi(n) = (p-1) * (q-1) = (5-1) * (7-1) = 4 * 6 = 24$ **$\Phi(n) = 24$**
4. Public key $e = 11$. **$e = 11$**
5. Calculate $d = 11$. $d = ((\Phi(n) * i) + 1) / e$ **$d = 11$**
6. Public key = $\{e, n\} = \{11, 35\}$, private key = $\{d, n\} = \{11, 35\}$.
7. Plain text $P = 2, C = P^e \bmod n = 2^{11} \bmod 35 = 18$. **Cipher Text = 18**
8. Cipher text $C = 18, P = C^d \bmod n = 18^{11} \bmod 35 = 2$. **Plain Text = 2**

Exercise - 3

Question: P and Q are two prime numbers. $P=17$, and $Q=11$. Take public key $E=7$. If plain text value is 5, then what will be cipher text value & private key value according to RSA algorithm? Again calculate plain text value from cipher text.

Solution:

1. Two prime numbers $p = 17, q = 11$
2. $n = p * q = 17 * 11 = 187$ **$n = 187$**
3. $\Phi(n) = (p-1) * (q-1) = (17-1) * (11-1) = 16 * 10 = 160$ **$\Phi(n) = 160$**
4. Public key $e = 7$. **$e = 7$**
5. Calculate $d = 23$. $d = ((\Phi(n) * i) + 1) / e$ **$d = 23$**
6. Public key = $\{e, n\} = \{7, 187\}$, private key = $\{d, n\} = \{23, 187\}$.
7. Plain text $P = 5, C = P^e \bmod n = 5^7 \bmod 187 = 146$. **Cipher Text = 146**
8. Cipher text $C = 146, P = C^d \bmod n = 146^{23} \bmod 187 = 5$. **Plain Text = 5**

Exercise - 4

Question: P and Q are two prime numbers. P=3, and Q=11. Take public key E=3. If original message is 00111011, then what will be cipher text value & private key value according to RSA algorithm? Again calculate plain text value from cipher text.

Solution:

1. Two prime numbers $p = 3, q = 11$

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

3. $\Phi(n) = (p-1) * (q-1) = (3-1) * (11-1) = 2 * 10 = 20$ **$\Phi(n) = 20$**

4. Public key $e = 3$. **$e = 3$**

5. Calculate $d = 7$. $d = ((\Phi(n) * i) + 1) / e$ **$d = 7$**

6. Public key = $\{e, n\} = \{3, 33\}$, private key = $\{d, n\} = \{7, 33\}$.

7. Plain text $P = (00111011)_2 = (59)_{10}$, $C = P^e \bmod n = 59^3 \bmod 33 = 20$. **Cipher Text = 20**

8. Cipher text $C = 20$, $P = C^d \bmod n = 20^7 \bmod 33 = 26$. **Plain Text = 26**

Exercise - 5

Question: P and Q are two prime numbers. P=13, and Q=17. Take public key E=19. If original message is 12, then what will be cipher text value & private key value according to RSA algorithm? Again calculate plain text value from cipher text.

Solution:

1. Two prime numbers $p = 13, q = 17$

2. $n = p * q = 13 * 17 = 221$ **$n = 221$**

3. $\Phi(n) = (p-1) * (q-1) = (13-1) * (17-1) = 12 * 16 = 192$ **$\Phi(n) = 192$**

4. Public key $e = 19$. **$e = 19$**

5. Calculate $d = 91$. $d = ((\Phi(n) * i) + 1) / e$ **$d = 91$**

6. Public key = $\{e, n\} = \{19, 221\}$, private key = $\{d, n\} = \{91, 221\}$.

7. Plain text $P = 12$, $C = P^e \bmod n = 12^{19} \bmod 221 = 181$. **Cipher Text = 181**

8. Cipher text $C = 181$, $P = C^d \bmod n = 181^{91} \bmod 221 = 12$. **Plain Text = 12**