ASSIGNMENT 2
AES AND DES

Answer 1: A.

Simple DES algorithm is implemented in which 64-bit Plain text and key are used. The input from user is taken at run time and the output is also provided at the run time. Modified algorithms also uses the initial provided plain text and key such that avalanche effect can be measured properly.

Modifications in current DES algorithm:

1. Modifications regarding **Substitution operation**: In this operation, the S-box table is changed to below-defined values, keeping the others same to obtain the effect of changing the substitution.

```
S_box[ 0 ]
[0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8]
[4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0]
[15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13]
[14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7]
S_box[ 1 ]
[3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5]
[15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10]
[0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15]
[13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9]
S_box[ 2 ]
[13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1]
[13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7]
[10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8]
[1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12]
S_box[ 3 ]
[7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15]
[10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4]
[13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9]
[3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14]
S_box[ 4 ]
[14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6]
[4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14]
[2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9]
[11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3]
S_box[ 5 ]
[9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6]
[12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11]
[10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8]
[4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13]
S_box[ 6 ]
[4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1]
[13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6]
[1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2]
[6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12]
S_box[ 7 ]
[13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7]
[1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2]
[7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8]
[2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11]
```

2. Modifications Regarding **Permutation Operation**; In this operation the Permuation tables are modified as shown below, keeping others same to brain the effect of chaing premutation.

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

3. For modifying the **XOR** operation it is changed to **X-NOR** operation.
4. For modifying the **Block size**. The user is prompted to enter the modified block size and then the respective block size is used for algorithm.
5. For modifying the **key generation** the value of key generation tables are changed as defined below keeping the values of other tables same..

```
# parity bit drop table
  keyp = [
      7, 62, 54, 46, 38, 30, 22,
      57, 49, 41, 33, 25, 17, 9,
      63, 55, 47, 39, 31, 23, 15,
      10, 2, 59, 51, 43, 35, 27,
      19, 11, 3, 60, 52, 44, 36,
      1, 58, 50, 42, 34, 26, 18,
      14, 6, 61, 53, 45, 37, 29,
      21, 13, 5, 28, 20, 12, 4 ]
  # Compression of key from 56 bits to 48 bits
  key_comp = [
        41, 52, 31, 37, 47, 55,
        30, 40, 51, 45, 33, 48,
        46, 42, 50, 36, 29, 32,
        23, 19, 12, 4, 26, 8,
        16, 7, 27, 20, 13, 2,
        44, 49, 39, 56, 34, 53,
        14, 17, 11, 24, 1, 5,
        3, 28, 15, 6, 21, 10]
```

For obtaining the avalanche effect on the different input and different modifications refer the table below in which two different inputs are tested for different modifications using the same key "SECURITY" and effects are recorded.

For Substitution:

Plain Text	Cipher Text without Modifications	Cipher Text with modifications	Number of different characters	Avalanche effect(%)
IITJODHP	4CE85A0C4243F9A6	3A663BAAF729E688	16	100
JayeshBu	C21BC980583FBC90	BC87F586DCA22977	15	93.75

For Permutation:

Plain Text	Cipher Text without Modifications	Cipher Text with modifications	Number of different characters	Avalanche effect(%)
IITJODHP	4CE85A0C4243F9A6	AE565E64E3999E0F	15	93.75
JayeshBu	C21BC980583FBC90	408A111B58BFAF96	12	75

For XOR:

Plain Text	Cipher Text without Modifications	Cipher Text with modifications	Number of different characters	Avalanche effect(%)
IITJODHP	4CE85A0C4243F9A6	82E1D011B4B3DE67	14	87.5
JayeshBu	C21BC980583FBC90	864BD714EBF56F19	15	93.75

For Block size: 32 (IITJ) and 48(Jayesh) block size is used in first and second respectively.

Plain Text	Cipher Text without Modifications	Cipher Text with modifications	Number of different characters	Avalanche effect(%)
IITJODHP	4CE85A0C4243F9A6	F10C82DE93CB92F9	16	100
JayeshBu	C21BC980583FBC90	7FB59361362F8C7F	14	87.5

For Key Generation:

Plain Text	Cipher Text without Modifications	Cipher Text with modifications	Number of different characters	Avalanche effect(%)
IITJODHP	4CE85A0C4243F9A6	3A7F25895AEC3D4F	16	100
JayeshBu	C21BC980583FBC90	905D2E2680052FF4	16	100

Answer1: B.

Des encryption was broken means someone was to get the plain text from the ciphertext using the key this may happen because the key exchange was weak and hence key got leaked and then the attack was performed. As we know that the key size is 56 bit which would be around $2^{10} = 7.2 * 10^{16}$ which when tried using brute force can be broken within 13 months if the power of the machine is 1000 keys per microsecond or about 1 or 2 days when using specific machines having high parallel processing power. And it would also become easier in case half of the bits are 0's and half 1's like when 0000000011111111 is used as a key.

But this does not mean that Des is not important there are many algorithms that are cracked but this does not mean we cannot use them in applications in the case of Des the only problem is the key size so when we increase the key size we can use Des for a variety of applications the best example of this is Doule Des which uses a 112-bit key and Triple Des which use 3 different independent keys for encryption and decryption.

Answer 2:

The Algorithm is implemented for 128-bit input that is 16 characters are used for plain text and key. The input is provided by the user in the input.txt file and output if provided in the output.txt file. The key is aside by the user while running the program as mentioned in the question. The implementation currently only works for 128 bits is 16 characters and does not work for more than 16 characters. But in the future, this can be increased to an arbitrary length.

The modifications done are:

A. For Shifting - the **first row is shifted by two places**, **second unchanged, third shifted by 1, and the fourth-row shifted by 3 places respectively**.
B. For **Xor** operation modification the xor operation is changed to **Xnor** operation by using the subtraction operation that is for two 4 bit numbers's a and b Xnor is 0xf - (a ^ b).
C. For **Mix columns,** operation modifications were made in the operation based on the original code provided in https://cs.ru.nl/~joan/papers/JDA_VRI_Rijndael_2002.pdf under section 4.1.2. In which **column a[1] was replaced with a[3]**.

For Avalanche Effect based on the above modifications refer to the table below(Two different inputs were provided) is summarised below:

For Shift row operation:

Plain text	Ciphertext without modification	Ciphertext with modification	Number of different characters	Avalanche effect
jayesh budhwani	c3378702d2362d3b62b95ad7cbc77711	733d82fc227732d1f9deb5a94da69095	28	89.6%
Hello How are yu	4bc403438330786d792f3d93811790e7	b6583d53e3fe1aad401e3a036e3424e1	26	81.25%

For Xor operation:

Plain text	Ciphertext without modification	Ciphertext with modification	Number of different characters	Avalanche effect
jayesh budhwani	c3378702d2362d3b62b95ad7cbc77711	9adfee368cc8e5c90ff6c8b42286f1c0	32	100%
Hello How are yu	4bc403438330786d792f3d93811790e7	14f478255fe96e3321c02520940dee8f	31	96.87%

For Mix Column operation:

Plain text	Ciphertext without modification	Ciphertext with modification	Number of different characters	Avalanche effect
jayesh budhwani	c3378702d2362d3b62b95ad7cbc77711	dee1e62f1425d58f520d547a71260eef	30	96%
Hello How are yu	4bc403438330786d792f3d93811790e7	3334bfeaf860fd20b4c428d421df8d9c	29	90.625%

For running the Programs:

1. Use python for running the programs.
2. For question 1 Plain text and key are taken form the user while running the program.
3. Download all the files including input_q2.txt and output_q2.txt.
4. For question 2 input is provided in input_q2.txt and can be changed for different input.
   NOTE: If ‘ symbol is inserted in the file then it is treated as input and is also encrypted.
5. Output of question 2 can be found in output_q2.txt file and the key used is to be entered by the user while running the program as was asked in the question.
6. For running the programs use **python M21CS007_q1.py** and **python M21CS007_q2.py.**

Note:
In some cases in DES algorithm for certain input patterns of the key the values of the key are converted into 56 bits in which case the output is not computed.

References:

1. https://cs.ru.nl/~joan/papers/JDA_VRI_Rijndael_2002.pdf for AES Algorithm.

DES Algorithm

- Dry run of 1 round:-

Input :- plaintext :- JayeshB4

Hex :- 4A61 7965 7336 8427 5

Key :- SECURITY

Hex :- 5345 4355 5249 5459.

Now Key generation

Key.
~~Plaintext~~ in Binary.

0101 0011 0100 0101 0100 0011 0101 0101 0101 0101 0010 0100 1001 0101 0100 0101 1001 (64 bits).

Converting it to 56 bits using initial permutation table.

$$\begin{bmatrix} 57 & 49 & 41 & 33 & 25 & 17 & 9 \\ 1 & 58 & 50 & 42 & 34 & 26 & 18 \\ 10 & 2 & 59 & 51 & 43 & 35 & 27 \\ \vdots & & & & & & \end{bmatrix}_{7\times 8}$$

like $57^{th} \to 0$, $49^{th} \to 0$, $41^{th} \to 0$, $1^{st} \to 0$.

Binary: 00000000 11111111 00000000 11010001 01010 00101010100000 1001

dividing into 28 bits.

left = 0000 0000 1111 1111 0000 0000 1101

Right = 0001 0101 0100 1010 1010 0000 1001.

After shifting using the tables we shift by 1 unit to left

left = 0000 0001 1111 1110 0000 0001 1010.

Right = 0010 1010 1001 0101 0100 0001 0010.

Combining :- Left + Right

we get First Round Key as.

0000 0011 1111 1110 0000 0001 1010 0010 1010 1001 0101 0100 0001 0010.

After this we do 48 bit compression.

1011 0000 1001 0010 0100 1010 0111 0101 0010 1000 0011 0000

B 0 9 2 4 A 7 5 2 8 3 0

Hence First Round Key = B09 24A7 5283 0.

Now

After doing initial permutation using the defined table.
we get FF94889E00BE2851 ⇒ using table

Now Splitting this data into LPT and RPT

LPT = FF94889E

RPT = 00BE2551.

Now RPT is expanded into 48 bits from 32 bits.
Like. 8015FC10AAA2. This is done using
the Expansion Table.

Then. This Expanded is XOR with Round 1
Key found befor. as.

```
  8015FC10AAA2
⊕ B092 4A75 2830
  3087B6658292
```

Then this 48 bit data is compressed to 32 bits.
Using S.box tables 1. of like.

011000 ...

0th row 6th value ⇒ 13 ⇒ D. using Sbox
0th row 4th value ⇒ 15 ⇒ F. table

like this we get DF1C6109

After this the 32 bits are permuted.
using the table which gives 4C8CF8D2.
Then this is XORed with LPT and swapped.
which gives. LPT = 00BE2551, RPT = B31870C.

AES Algorithm.

• Dry Run :- of 1 Round.

let Plain text :- 54 776F 20 4F 6E 65 20 4E 69 6E
65 20 54 776F

Key :- 54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75

Now w[0] = [54, 68, 61, 74]
w[1] = [73, 20, 6D, 79]
w[2] = [20, 4B, 75, 6E]
w[3] = [67, 20, 46, 75]

Now we Need to find. $w[4] = w[0] \oplus g(w[3])$

$w[4] = [54, 68, 61, 74] \oplus g(w[3])$

$g(w[3])$ = Circular Shift of w[3], Bytes.
Substitution using S-box then
adding round constent.

= [B6, 5A, 9D, 85]

$w[4] = [54, 68, 61, 74] \oplus [B6, 5A, 9D, 85]$

54 0101 0100
B6 1011 0110
⊕ 1110 0010 ⇒ E2

w[4] = [E2, 32, FC, F1].

Similary $w[5] = w[4] \oplus w[1]$ and So on.

w[5] = [91, 12, 91, 88]. , w[6] = [B1, 59, E4, E6]

w[7] = [D6, 79, A2, 93].

First round key :- E2 32 FC F1 91 12 91 88
B1 59 E4 E6 D6 79 A2 93.

Now Initial state Matrix = $\begin{bmatrix} 54 & 4F & 4E & 20 \\ 77 & 6E & 69 & 54 \\ 6F & 6r & 6E & 77 \\ 20 & 20 & 6r & 6F \end{bmatrix}$

Round Key For Initial round key = $\begin{bmatrix} 54 & 73 & 20 & 67 \\ 68 & 20 & 4B & 20 \\ 61 & 6D & 7r & 4e \\ 74 & 79 & 6E & 7r \end{bmatrix}$

New State = XOR of given state Matrix and key

$= \begin{bmatrix} 54 \oplus 54 & \cdots \\ 77 \oplus 68 & \\ \vdots & \end{bmatrix} = \begin{bmatrix} 00 & 3C & 6E & 47 \\ 1F & 4E & 22 & 74 \\ 0E & 08 & 1B & 31 \\ 54 & 59 & 0B & 1A \end{bmatrix}$

Now using sbox to Substitute the State matrix values like 00 is replaced by 0th row 0th Column, etc.

New Matrix after Substitution = $\begin{bmatrix} 63 & EB & 9F & A0 \\ C0 & 2F & 93 & 92 \\ AB & 30 & AF & C7 \\ 20 & CB & 2B & A2 \end{bmatrix}$

Now row Shifting occurs as row 1 → 0 row 2 → 1, row-3 → 2, row 4 → 3.

which gives $\begin{bmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{bmatrix}$

Now Mix Column.

$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{bmatrix} = \begin{bmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{bmatrix}$

$02 \cdot 63 \oplus 03 \cdot 2F \oplus 01 \cdot AF \oplus 01 \cdot A2 = BA$

Finally we Need to do add round key. with round 1 key generated before.

$$\begin{bmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{bmatrix} \oplus \begin{bmatrix} E2 & 91 & B1 & D6 \\ 32 & 12 & 59 & 79 \\ FC & 91 & E4 & A2 \\ F1 & 88 & E6 & 93 \end{bmatrix}$$

BA	1011	1010
E2	1110	0010
⊕	0101	1000 = 58

which gives $\begin{bmatrix} 58 & 1F & 59 & CD \\ 47 & B6 & D4 & 39 \\ 08 & 1C & E2 & DF \\ 8B & BA & E8 & CE \end{bmatrix}$

which is output of round 1.

In this way aes algorithm work.