Advanced Encryption Standard (AES)

-> AES is a block cipher.

-> The key size can be 128/192/256 bits.

-> Encrypts data in blocks of 128 bits each.

-> AES performs operations on bytes of data rather than in bits.

-> The number of rounds depends on the key length.

⇒ 128-bit key - 10 rounds.
⇒ 192-bit key - 12 rounds.
⇒ 256-bit key - 14 rounds.

Creation of Round keys

A key schedule Algorithm calculates all the round keys from the key. So the initial key is used to create many different rounds keys which will be used in the corresponding round of the encryption.

Encryption :- AES Considers each block as a 16-byte (4 byte × 4 byte = 128) grid in a column-major arrayed

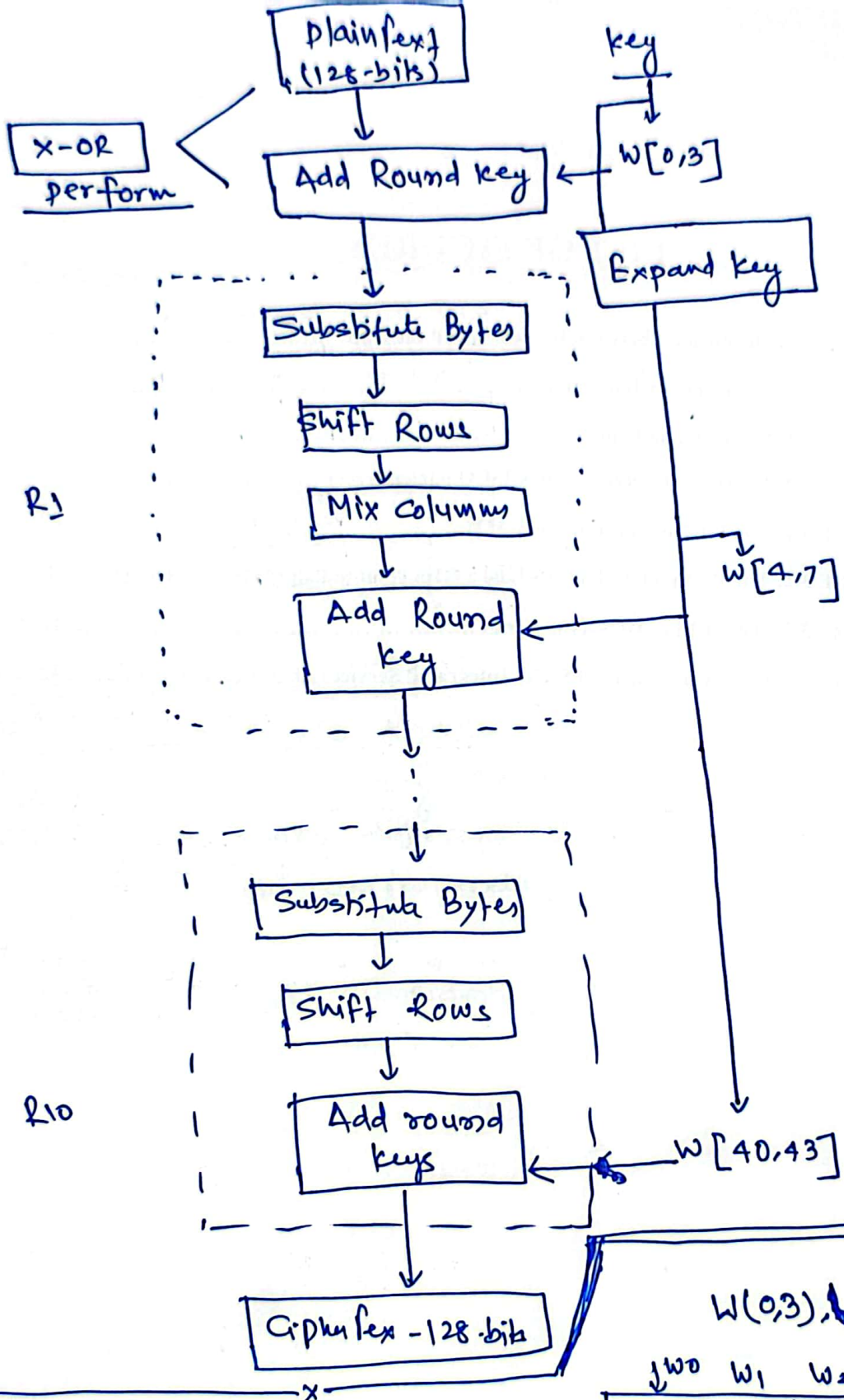
[b0 | b4 | b8 | b12
| b1 | b5 | b9 | b13
| b2 | b6 | b10 | b14
| b3 | b7 | b11 | b15]

Each round Comprises of 4 steps.

⇒ SubBytes
⇒ ShiftRows
⇒ MixColumns
⇒ AddRound key

⇒ The last round doesn't have the MixColumn round.

②



W(0,3), W(4,7)

W0	W1	W2	W3
k_0	k_4	k_8	k_{12}
k_1	k_5	k_9	k_{13}
k_2	k_6	k_{10}	k_{14}
k_3	k_7	k_{11}	k_{15}

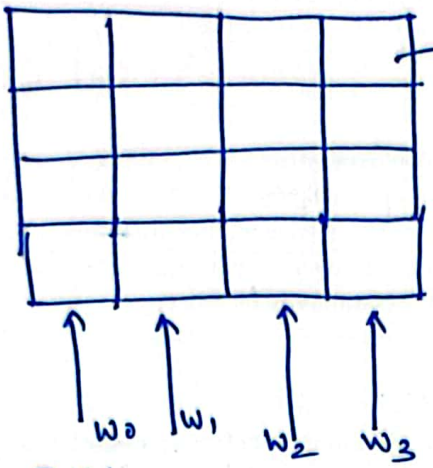
4 byte

* 1 Word = 32 bits
16 byte = 4 words

Each column = 1 word

4 words expands to 44 words

③ Input Array (4×4) = 16 byte = 4 word



Total = 16 byte = 4 word

1 word = 32 bits

8 × 4
= 32 bits
= 1 word = w_0

Key = 128-bits

k_0	k_4	k_8	k_{12}
k_1	k_5	k_9	k_{13}
k_2	k_6	k_{10}	k_{14}
k_3	k_7	k_{11}	k_{15}

w_0 w_1 w_2 w_3

⇒ 1 byte = 8 bits
⇒ 16 byte = 4 word
word = 32-bits

↓

Expansion upto 44 words

↓

k_0	w_1	w_2	. . . .	w_{43}

(4)

Add Round key

key = W[0,3], W[4,7] . . . . W[40,43]

$R_1 \rightarrow [W_0, W_3]$

$R_2 \rightarrow [W_4, W_7]$

⋮

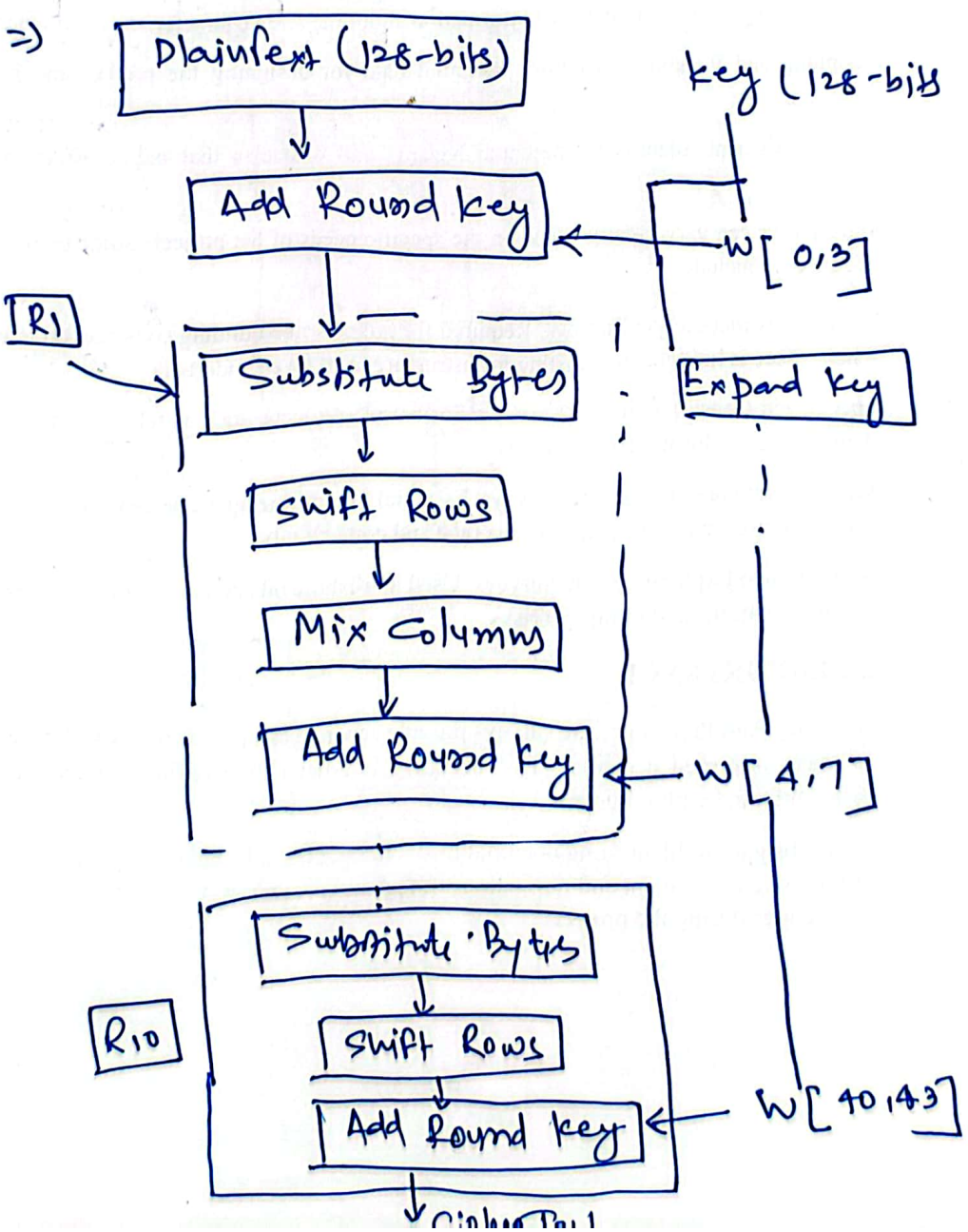
$R_{11} \rightarrow [W_{40}, W_{43}]$

$\Rightarrow R_{10} \times 4 = 40$ words

⇒ Initial 4 words for Add Round Key

i.e, total = 40 + 4
= 44 words.

⇒



(5)

State Array / State Matrix [4×4]
= 16 byte / 4 words

⇒ We use State Array to store intermediate result i.e, Result of R_1 to R_{10}.

State Matrix (4×4)

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

W_0 W_1 W_2 W_3

$S_{0,0}$ i.e, $S_{1,0}$

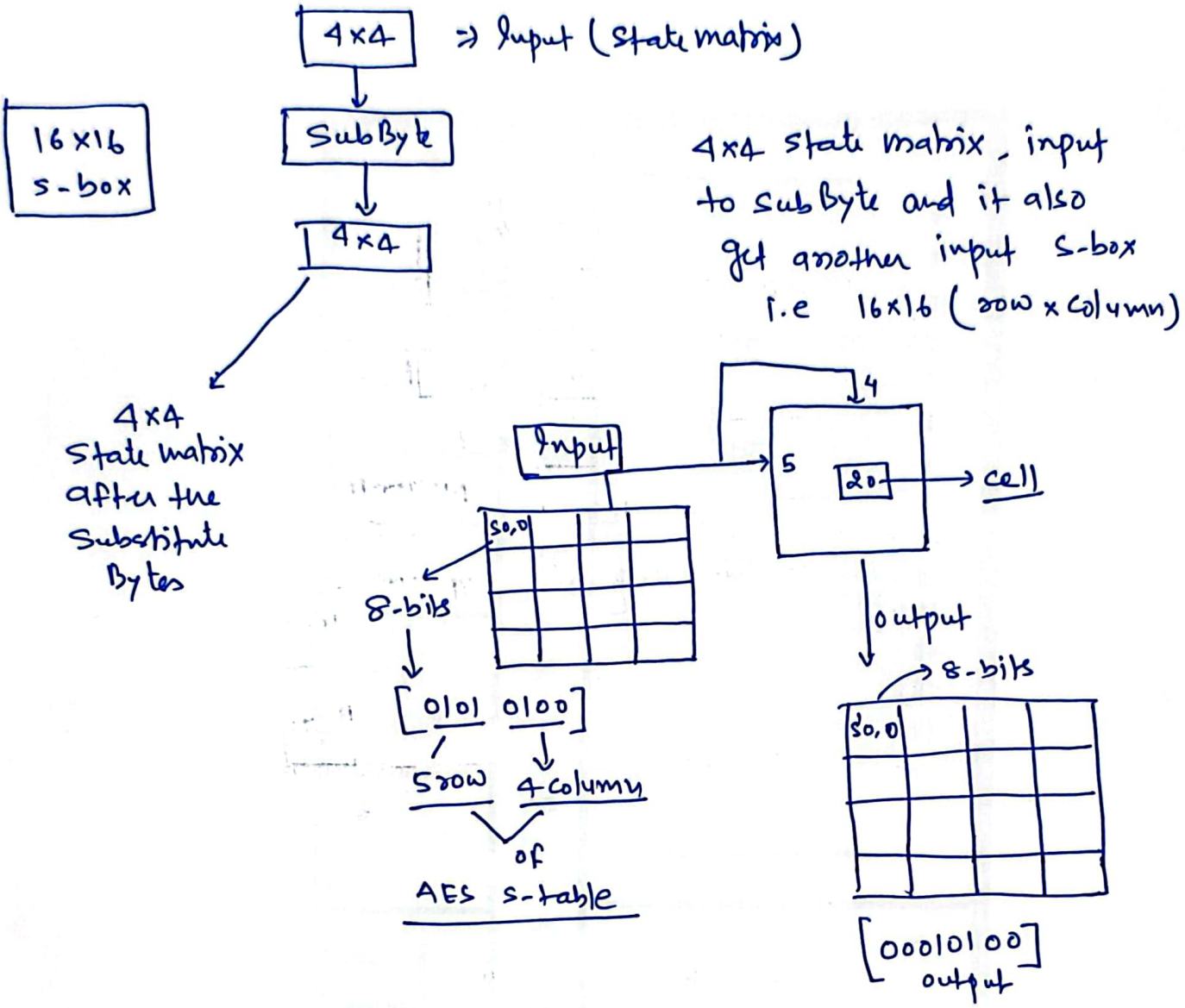
Row → Byte, Column → word.

1st byte of 0th word.

Then we pass the result of rounds into next round.

(6)

Events under each rounds



* Input

$S_{0,0}$ = 0101 0100

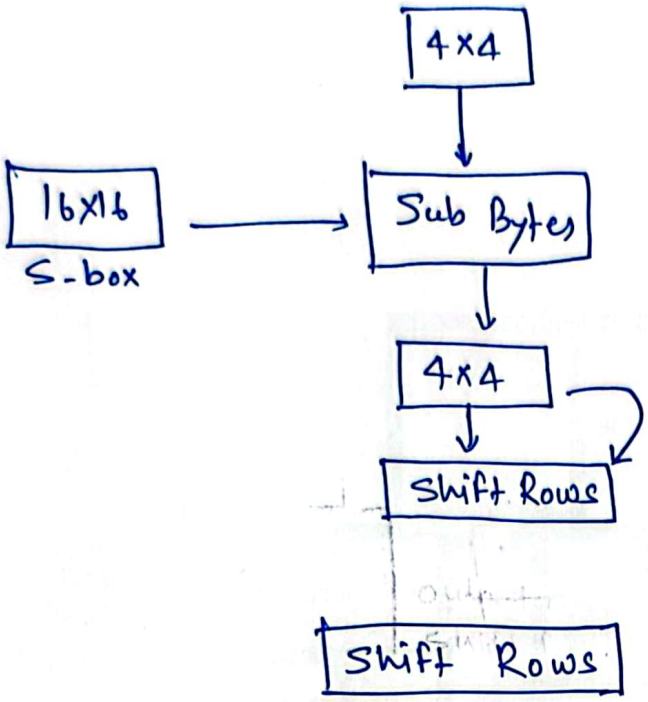
Row Column

Intersection 5×4 = 20 = 00010100

= 8 bits = output

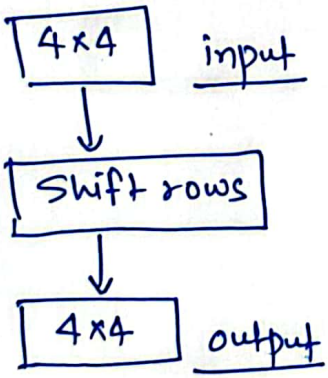
and we will repeat for each cell i.e, 16 cells.

(7)



Result will move for next function i.e, shift Rows.

⇒ Rows are going to shift in circular fashion towards left.



Input

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

⇓

Output

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

* Rotation depends on the row number.

Ex- Suppose row no is zero,

* Zero byte will be shifted Circular to the left.

* Suppose row no is 1

1 byte is going to shifted Circular to the left.

(8)

Mix Column

⇒ Multiplication of input matrix (constant) x [(4x4) state matrix]

4x4

16x16 Sbox → Sub Byte

4x4

Shift rows

4x4

⇒ Mix Column

4x4

(Input for Second round)

output of shift rows ⇓ [4x4 state matrix]

4x4 Input → Mix Column

[4x4 predefined matrix / Constant Input]

Column 1 x 1st colm

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

x

4x4 Constant matrix

i.e, Column 1 x 1st colm

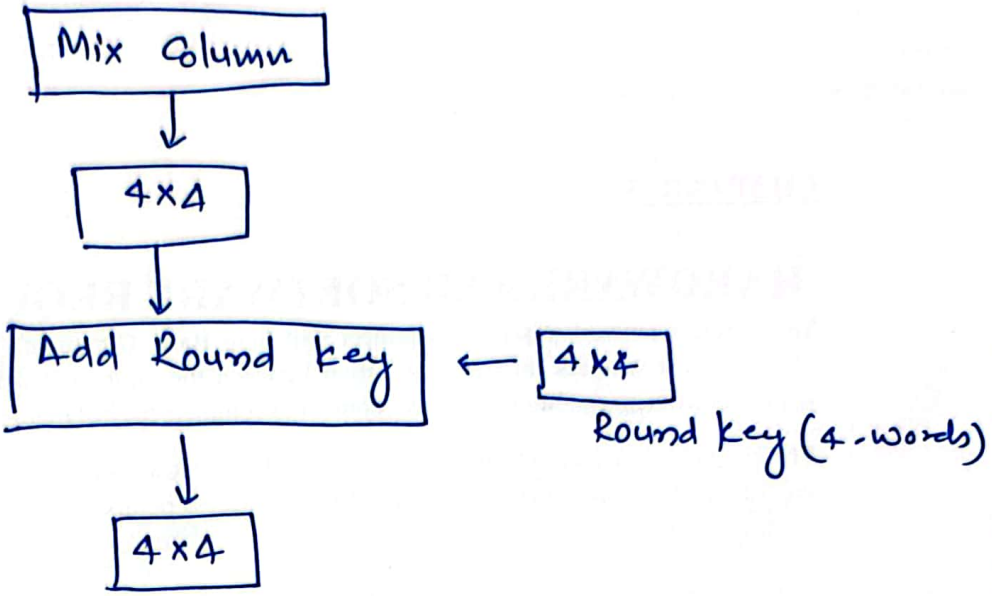
4x4 Input (Result / Output of Shift Row)

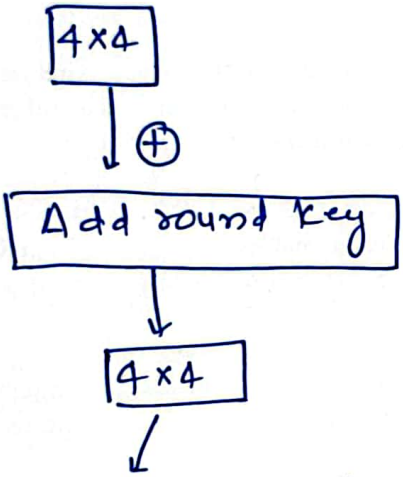
Output

4x4 matrix (output)

output of Mix column ⇓ Resultant column after multiplication

(9)





Now, it is ready to go to the next round.
i.e, this state matrix we get at the end of each round.