[5 + 5 = 10] 6

Q4(a)

Solution (a) The AES key expansion algorithm is explained in Algorithm 1. It has the following components:

- The AES key expansion algorithm takes an input of a 4-word (16-byte) key, and produces a linear array of 44 words (176 bytes).
- The key is copied into the first four words of the expansion key.
- If the word position (i) is not multiple of 4, then $w[i] = w[i-4] \oplus w[i-1]$.
- If the word position (i) is multiple of 4, then a complex function g is used to calculate the word:
  1. RotWord performs a one-byte circular left shift on a word. This means that an input word $[b_0, b_1, b_2, b_3]$ is transformed into $[b_1, b_2, b_3, b_0]$.
  2. SubWord performs a byte transformation on each byte of its input word, using the S-box.
  3. The result of Steps 1 and 2 is XORed with a round constant, $RCon[j]$. The "round constant" is different for each round, and is defined as $RCon[j] = (RC[j], 0, 0, 0)$, with $RC[1] = 1$, $\boxed{RC[2] = 2 * RC[j-1]}$ and with multiplication defined over the finite field $GF(2^8)$. The values of $RC[j]$ in hexadecimal are given in Table 1:

(-1)

Table 1: Values of $RC[j]$

j	1	2	3	4	5	6	7	8	9	10
$RC[j]$	01	02	04	08	10	20	40	80	1B	36

Scanned with OKEN Scanner

Algorithm 1 AESKeyExpansion (byte key[16], word W[44])

Require: 128-bit key, *key*.

Ensure: $[W[0], W[1], \ldots, W[43]]$: 44 words

word temp;
for $i = 0; i < 4; i++$ **do**
    $W[i] = (key[4*i], key[4*i+1], key[4*i+2], key[4*i+3]);$ — 1
end for
for $i = 4; i < 44; i++$ **do**
    temp $= W[i-1];$
    if $i \pmod 4 = 0$ **then**
        temp= SubWord(RotWord(temp)) $\ominus$ Rcon$[i/4]$; — 2
    end if
    $W[i] = W[i-4] \ominus$ temp; — 1
end for
return $(W[0], W[1], \ldots, W[43])$

(5)

Roundconstant values

Rc[j]

either write RC[1] = 1 & RC[j] = 2 * RC[j-1]

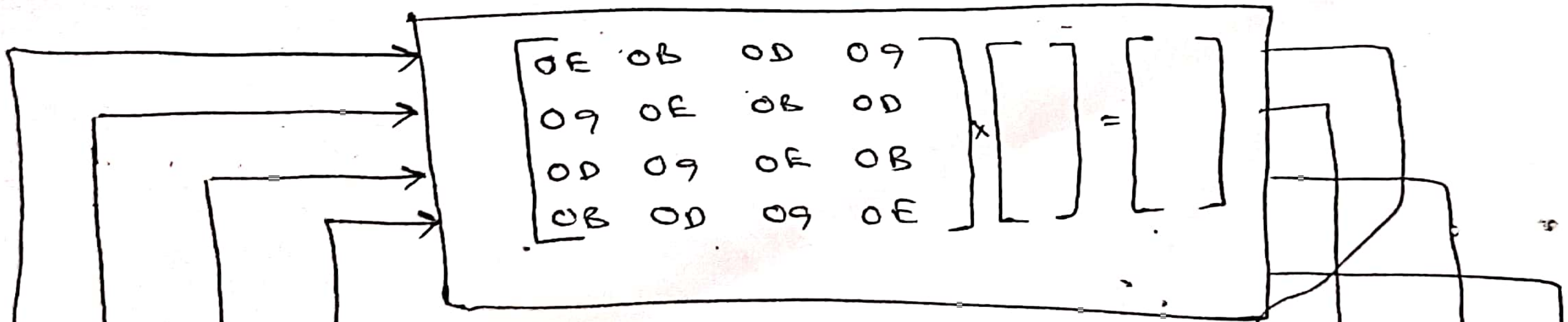
or mention ~~out~~ all values — 1 mark

Scanned with OKEN Scanner

... Transformation:

The "inverse mix column transformation", called $\boxed{\text{InvMixColumns}}$ is defined by the following matrix multiplication:

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \times \underbrace{\begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix}}_{S} = \underbrace{\begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix}}_{S'}$$



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

$$S'_{0,j} = \{0E\} * S_{0,j} \oplus \{0B\} * S_{1,j} \oplus \{0D\} * S_{2,j} \oplus \{09\} * S_{3,j}$$

$S'_{1,j} =$

$S'_{2,j} =$

$S'_{3,j} =$

$\boxed{j = 0, 1, 2, 3}$

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

Fig: Inverse Mix Column Transformation.

Scanned with OKEN Scanner