Key=128 bits => 10 Rounds

Cipher= 128 bits

1. Key Expansion

|  |  |
| --- | --- |
| **Key Size (words/bytes/bits)** | 4/16/128 |
| **Plaintext Block Size (words/bytes/bits)** | 4/16/128 |
| **Number of Rounds** | 10 |
| **Round Key Size (words/bytes/bits)** | 4/16/128 |
| **Expanded Key Size (words/bytes)** | 44/176 |

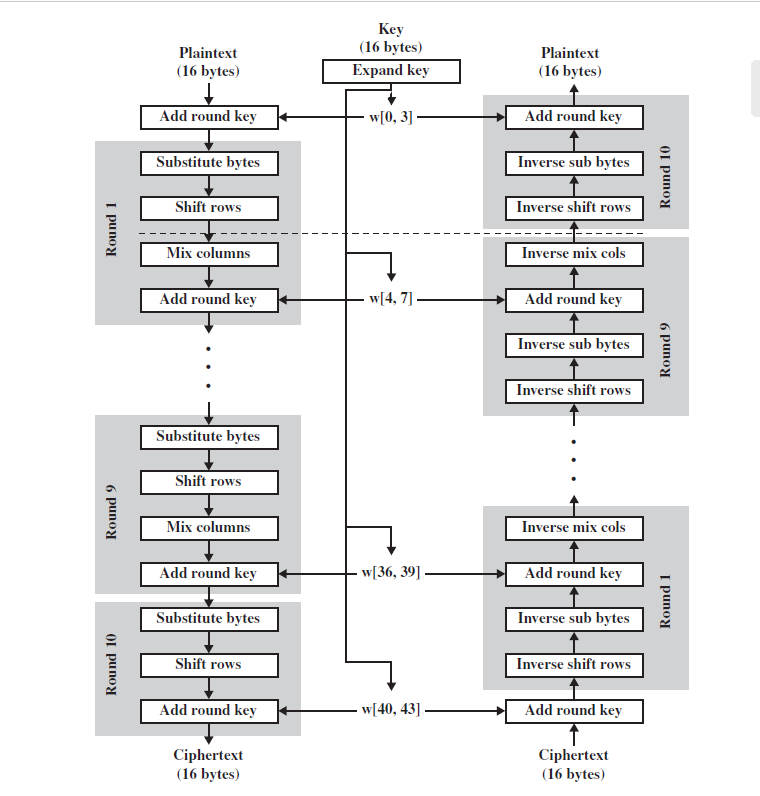
**3.** Four different stages are used, one of permutation and three of substitution:

• **Substitute bytes:** Uses an S-box to perform a byte-by-byte substitution of the block

• **ShiftRows:** A simple permutation

• **MixColumns:** A substitution that makes use of arithmetic over

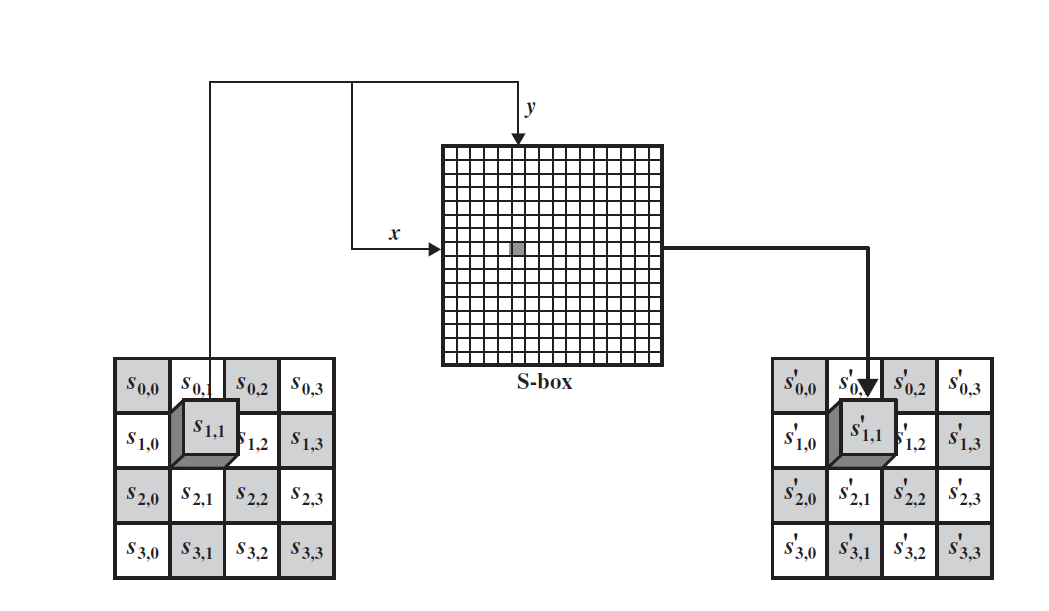
• **AddRoundKey:** A simple bitwise XOR of the current block with a portion of the expanded key

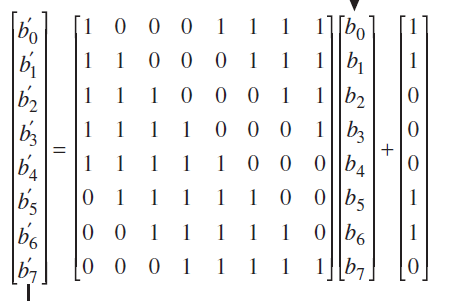


CIPHER MATRIX:4x4 , with each data of 8 bits == 16 x 8=128 bits

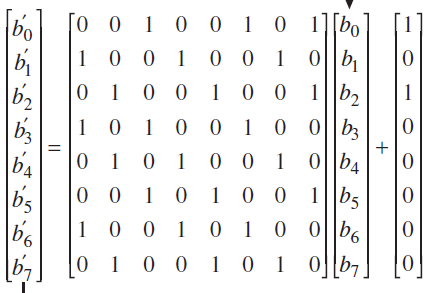
1. S-BOX Module(Cobinational , Input 8 bit and Outptut 8 bit)

Encryption

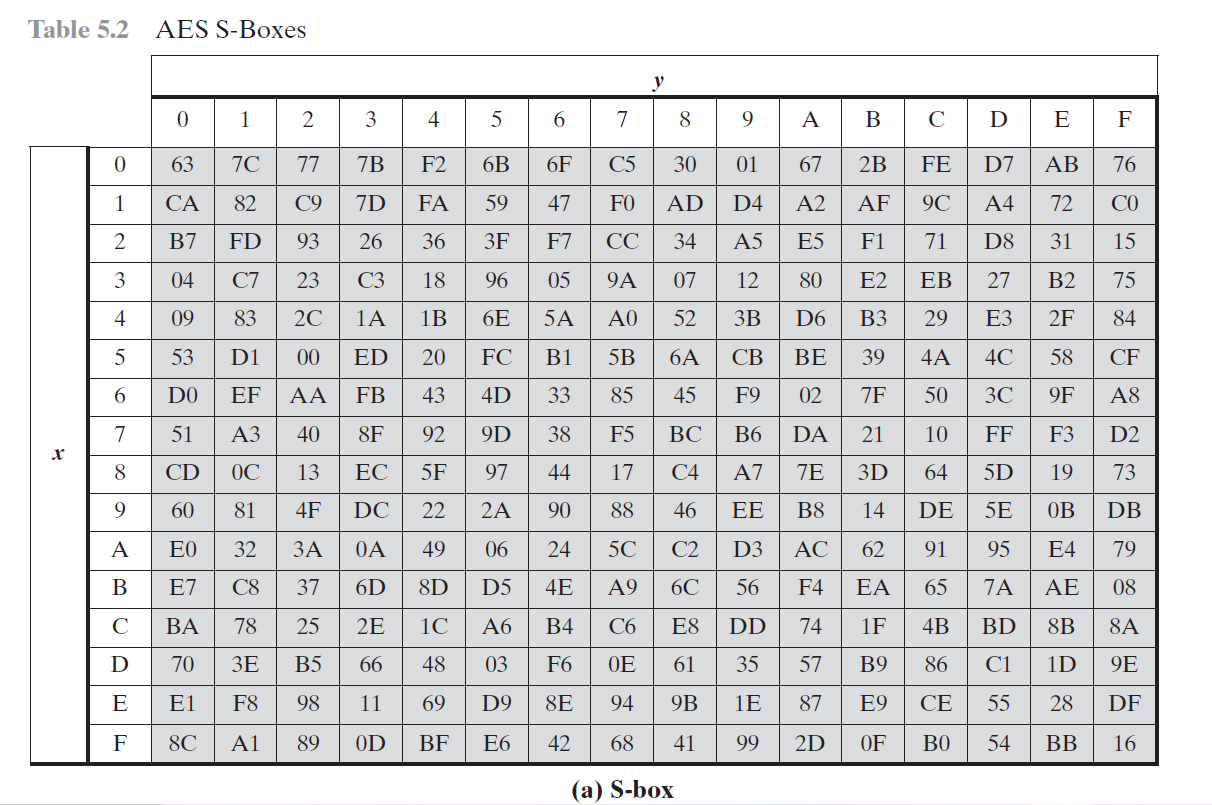


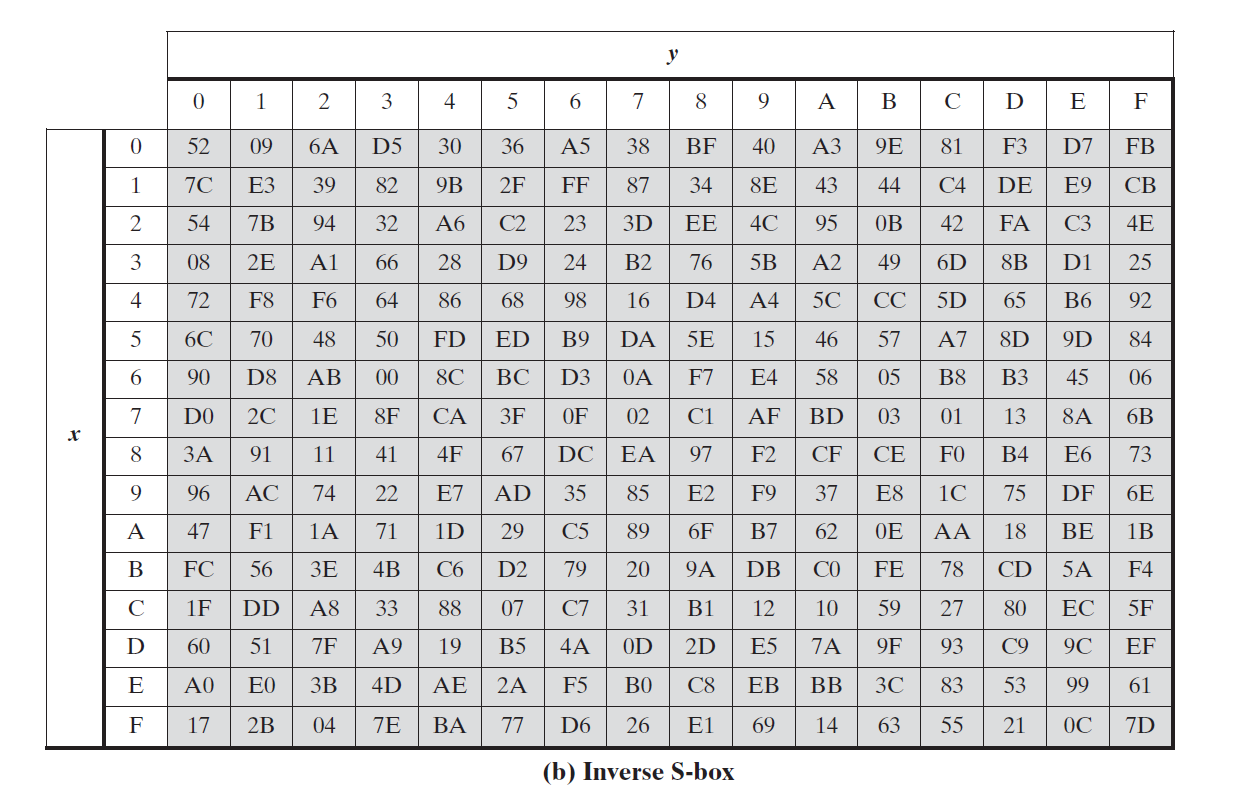


Decryption



TEST CASE

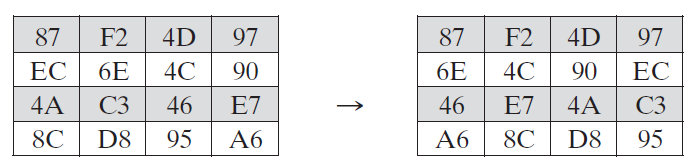




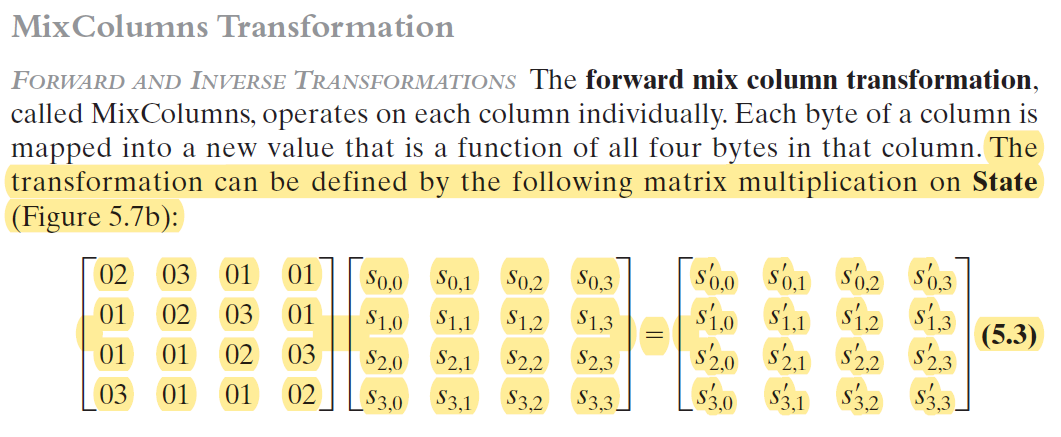
SHIFT ROWS

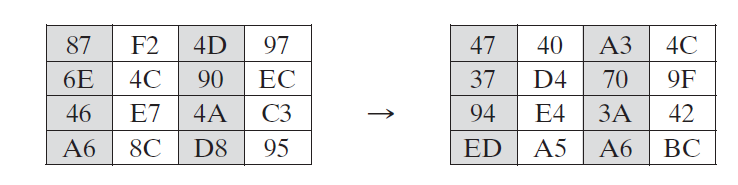
***FORWARD AND INVERSE TRANSFORMATIONS*** The **forward shift row transformation** , called ShiftRows, is depicted in Figure 5.7a.The first row of **State** is not altered. For the second row, a 1-byte circular left shift is performed. For the third row, a 2-byte circular left shift is performed. For the fourth row, a 3-byte circular left shift is performed.The following is an example of ShiftRows.

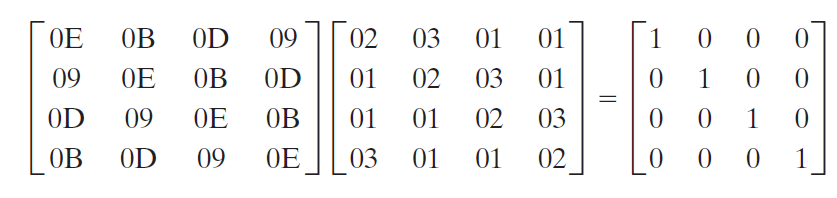
The **inverse shift row transformation**, called InvShiftRows, performs the circular shifts in the opposite direction for each of the last three rows, with a 1-byte circular right shift for the second row, and so on.



MIX COLUMNS





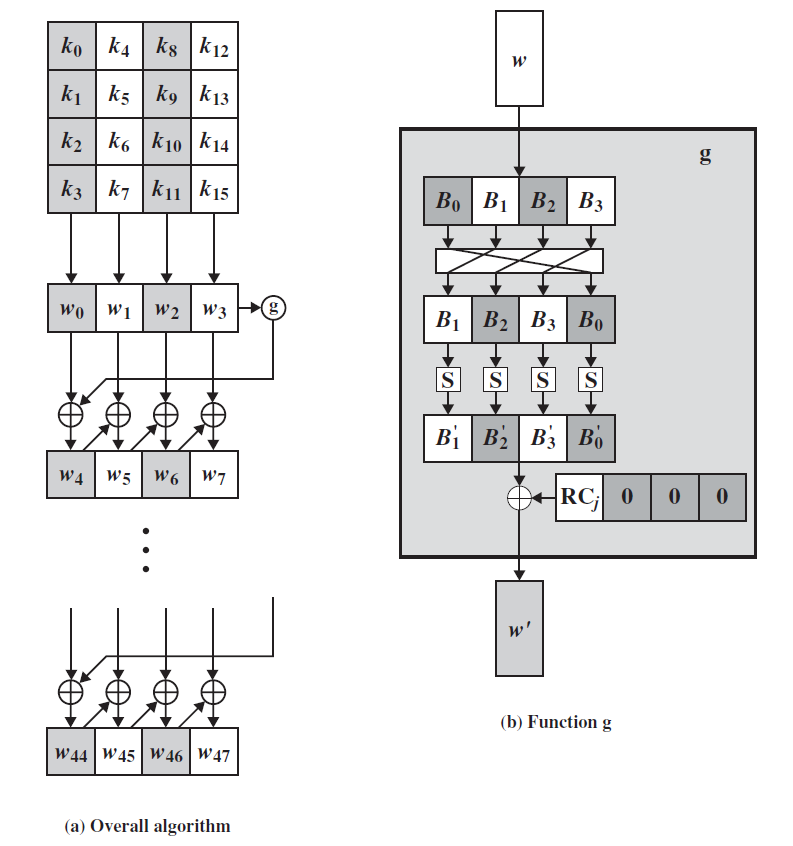


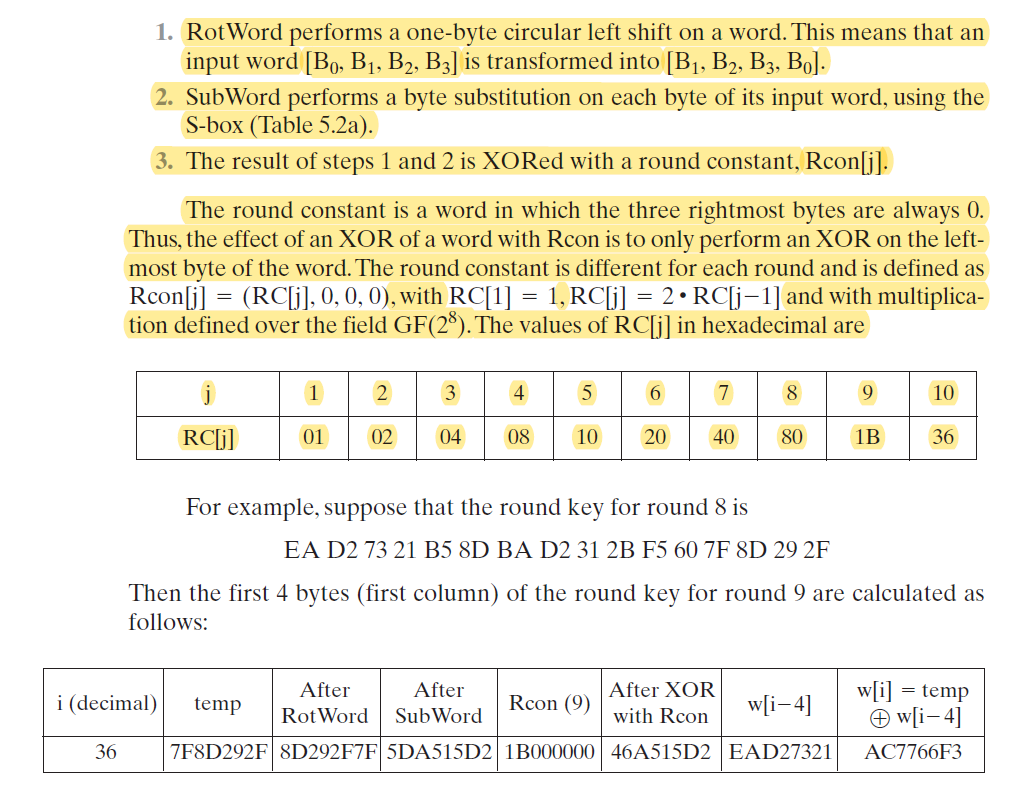
**AddRoundKey Transformation**

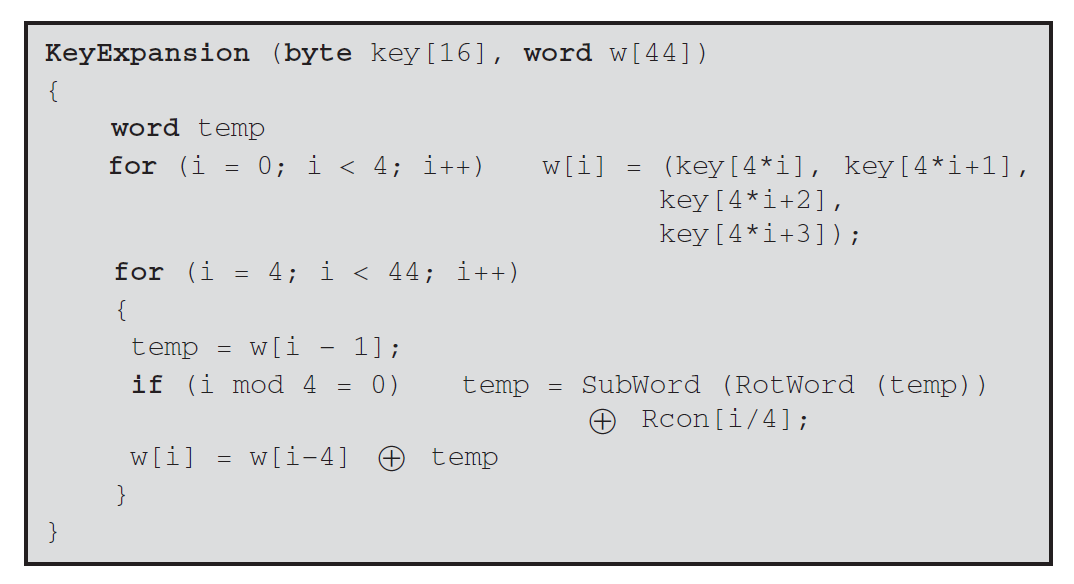
***FORWARD AND INVERSE TRANSFORMATIONS*** In the **forward add round key transformation**, called AddRoundKey, the 128 bits of **State** are bitwise XORed with the 128 bits of the round key. The first matrix is **State**, and the second matrix is the round key.

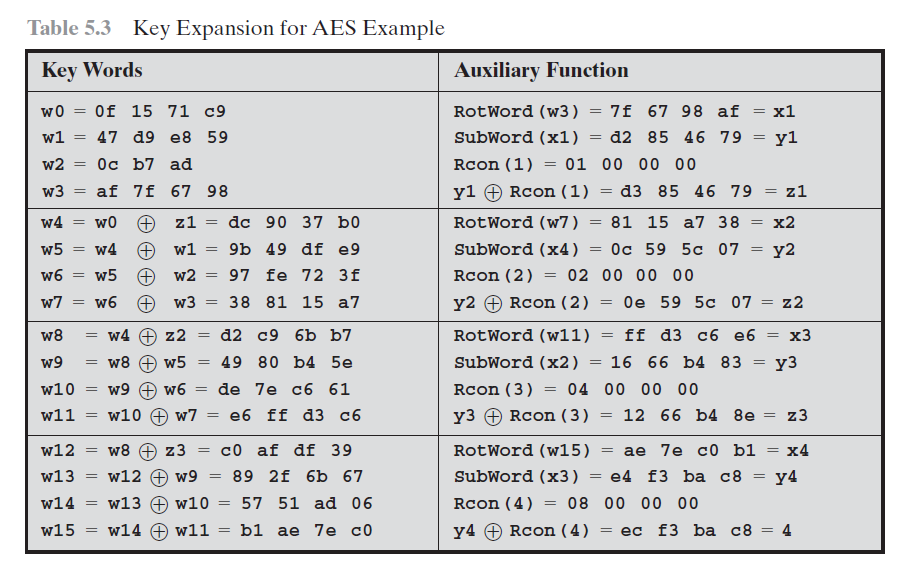
The **inverse add round key transformation** is identical to the forward add round key transformation, because the XOR operation is its own inverse.

KEY EXPANSION









OVERALL TESTCASE: 