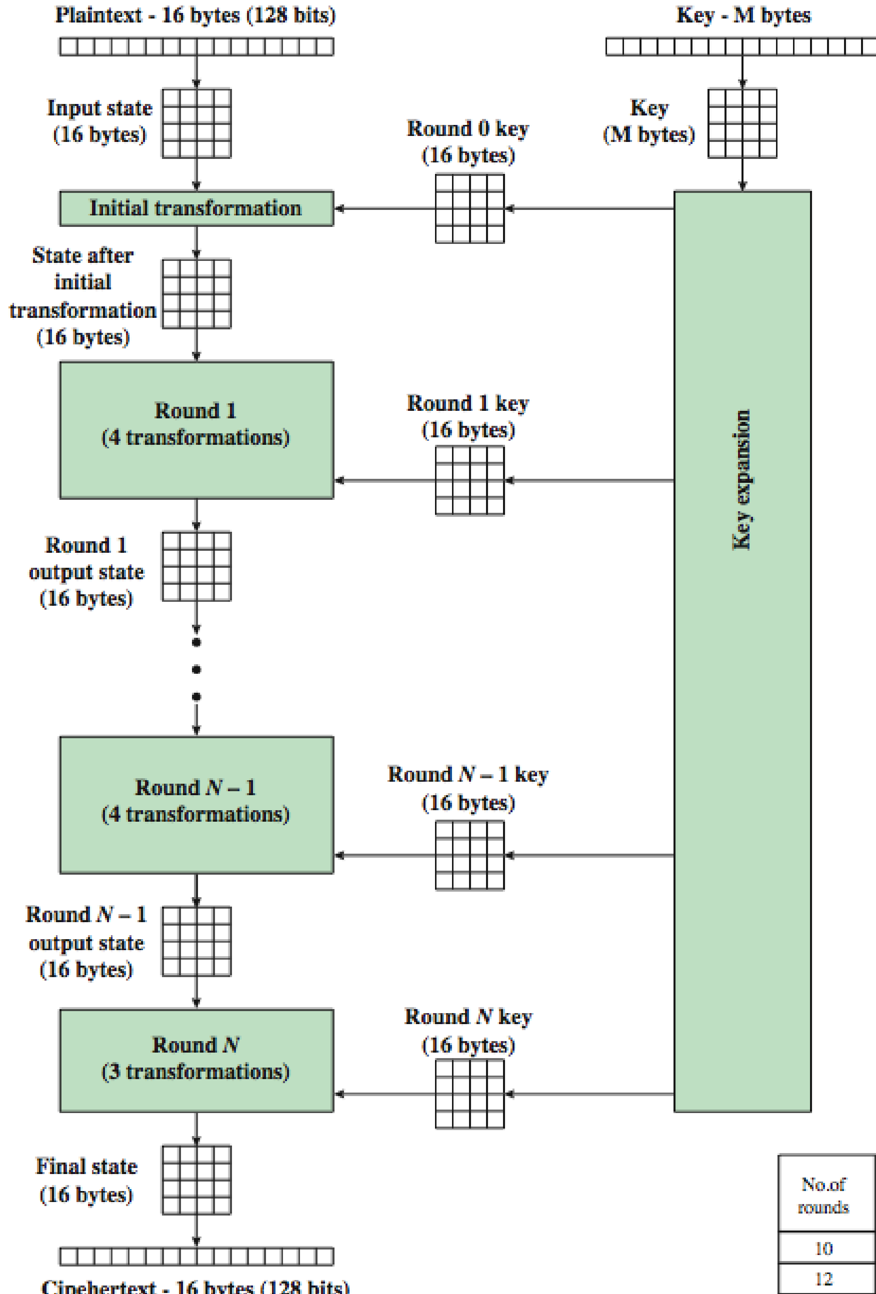
Corso di Sicurezza dei Sistemi
Prof. Salvatore D’Antonio

Advanced Encryption Standard



No.of rounds	Key Length (bytes)
10	16
12	24
14	32

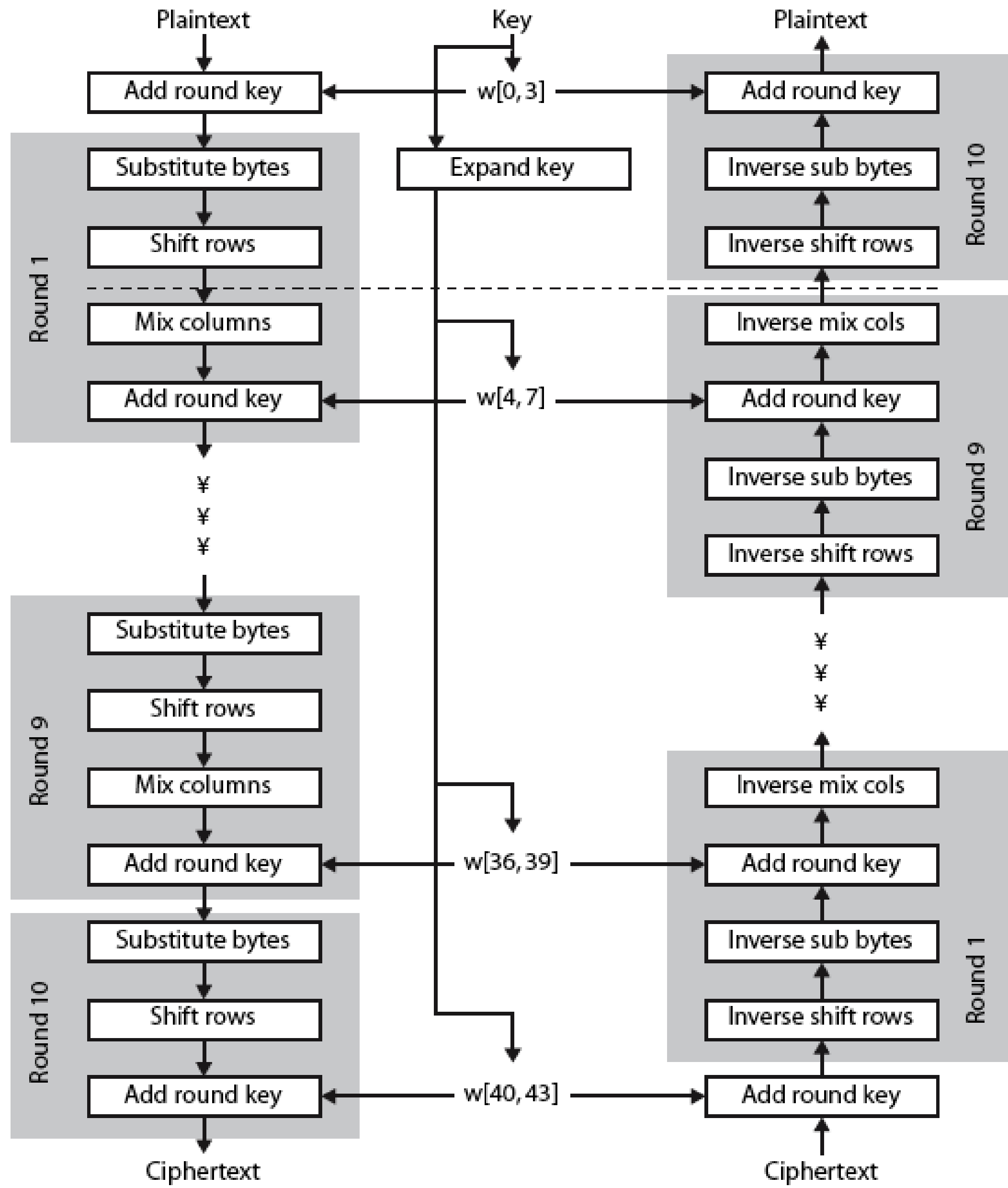
AES structure

- Data block of 4 columns of 4 bytes is state
- Key is expanded to array of words
- Has 9/11/13 rounds in which state undergoes:
  - byte substitution (1 S-box used on every byte)
  - shift rows (permute bytes between columns)
  - mix columns (subs using matrix multiply of groups)
  - add round key (XOR state with key material)

AES cipher

- Block length is limited to 128 bit (16 bytes)
- The key size can be independently specified to 128, 192 or 256 bits

Key size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Number of rounds	9	11	13
Expanded key size (words/byte)	44/176	52/208	60/240

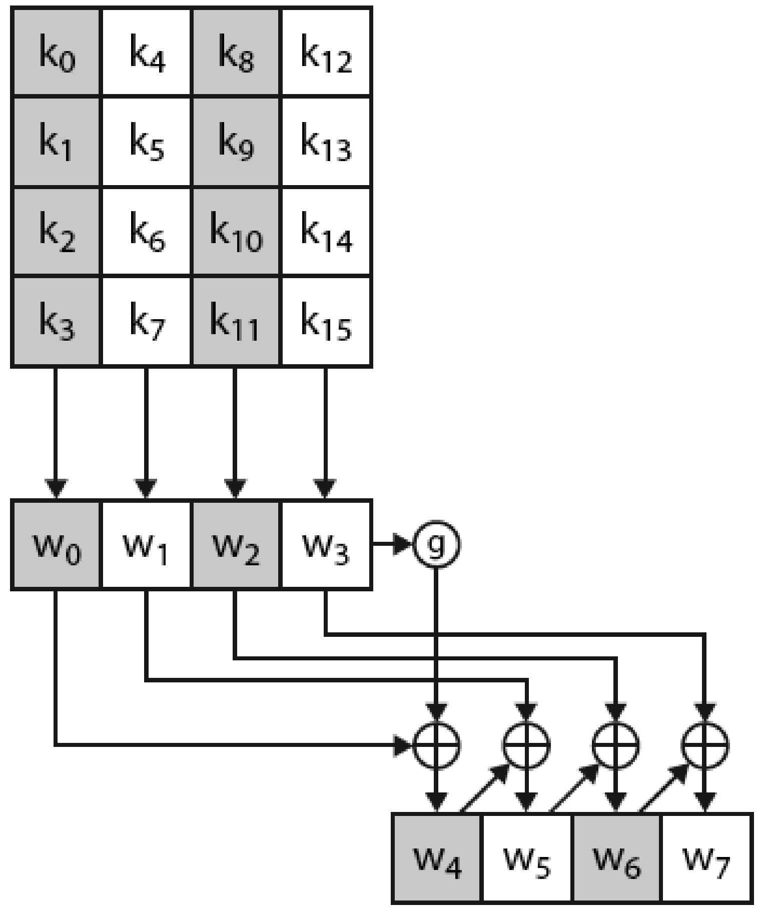


(a) Encryption (b) Decryption

AES key expansion

- takes 128-bit (16-byte) key and expands into array of 44/52/60 32-bit words
- start by copying key into first 4 words
- then loop creating words that depend on values in previous 4 places back
- in 3 of 4 cases just XOR these together
- 1st word in 4 has rotate + S-box + XOR round constant on previous, before XOR 4th back

AES key expansion



More details on AES

- has a simple structure
- only AddRoundKey uses key
- key expanded into array of 32-bit words
  - four words form round key in each round
- each stage is easily reversible
- decryption uses keys in reverse order
- decryption does recover plaintext
- final round has only 3 stages

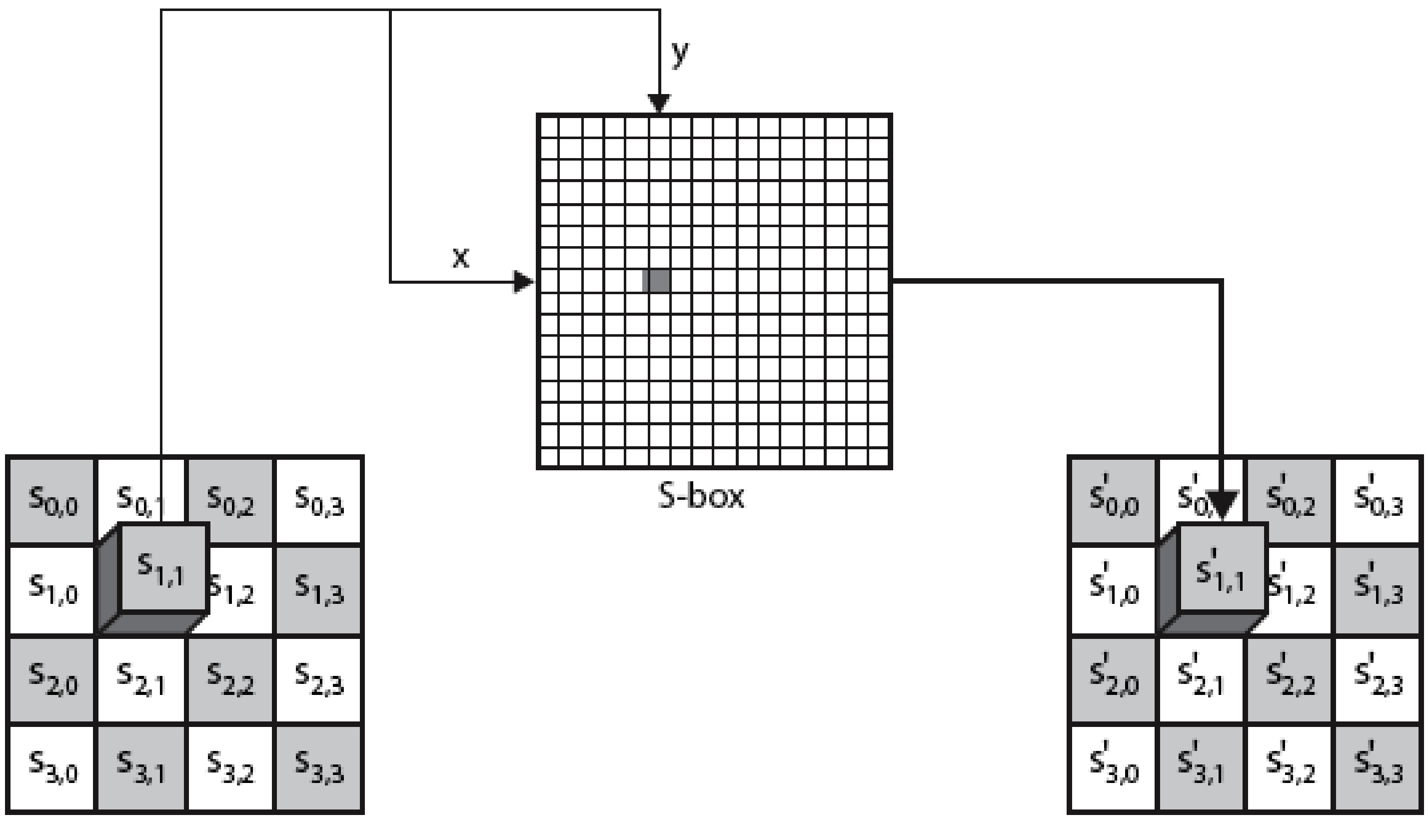
Four stages

- Byte substitution
- Shift rows
- Mix columns
- Add round key

Byte substitution

- a simple substitution of each byte
- uses one table of 16x16 bytes containing a permutation of all 256 8-bit values
- each byte of state is replaced by byte indexed by row (left 4-bits) & column (right 4-bits)
- eg. byte {95} is replaced by byte in row 9 column 5 which has value {2A}
- S-box constructed using defined transformation of values in GF(2^8)
- designed to be resistant to all known attacks

Substitute bytes



Substitute byte example

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

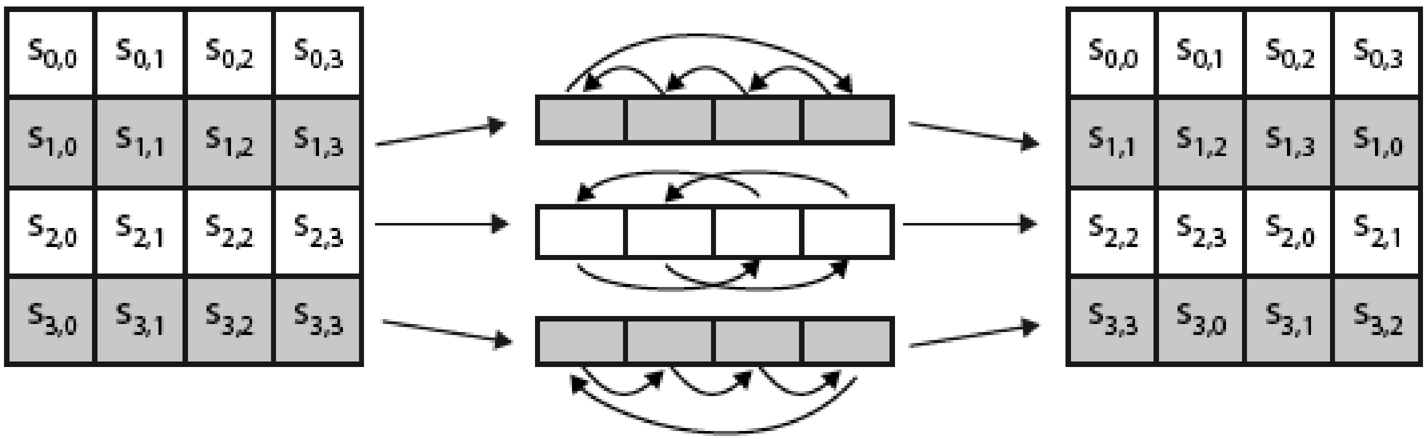
→

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

Shift rows

- a circular byte shift
  - 1st row is unchanged
  - 2nd row does 1 byte circular shift to left
  - 3rd row does 2 byte circular shift to left
  - 4th row does 3 byte circular shift to left
- decrypt inverts using shifts to right
- since state is processed by columns, this step permutes bytes between the columns

Shift rows



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

→

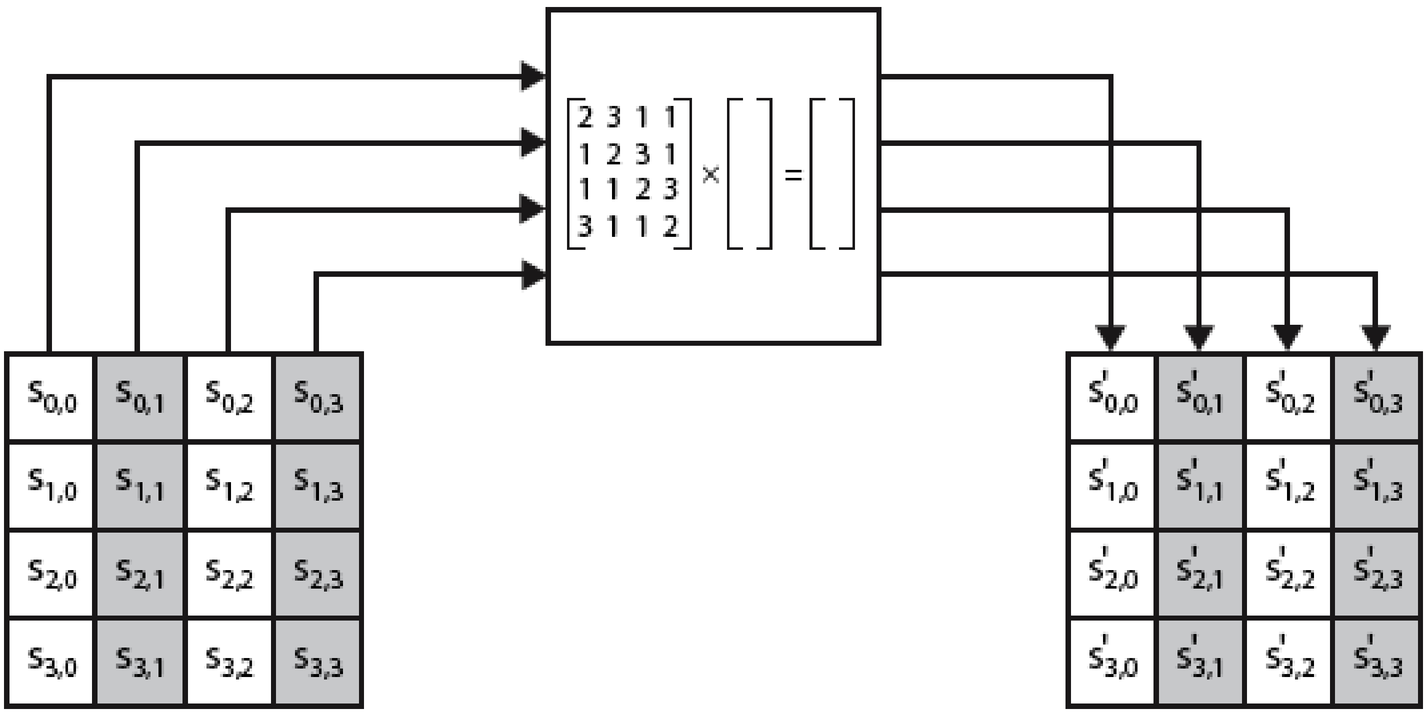
87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

Mix columns

- each column is processed separately
- each byte is replaced by a value dependent on all 4 bytes in the column
- a matrix multiplication in GF(2^8) using prime poly $m(x) = x^8+x^4+x^3+x+1$

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Mix columns



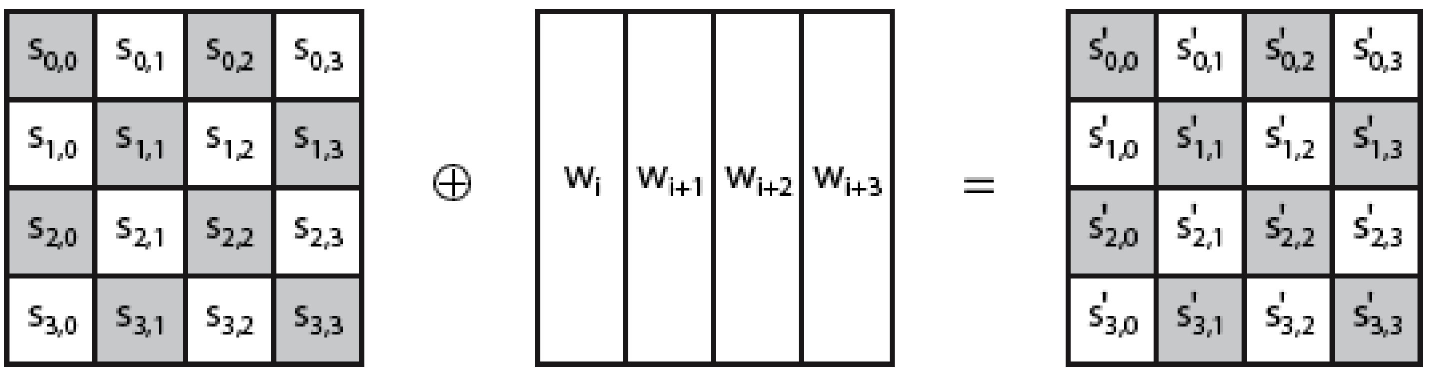
2 3 1 1
1 2 3 1
1 1 2 3
3 1 1 2
×
=
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

AES arithmetic

- uses arithmetic in the finite field $GF(2^8)$
- with irreducible polynomial
- $m(x) = x^8 + x^4 + x^3 + x + 1$
- which is (100011011) or {11B}
- e.g.
- $\{02\} \bullet \{87\}$ mod {11b} = (1 0000 1110) mod {11B}
- = (1 0000 1110) xor (1 0001 1011) = (0001 0101)

Add Round Key

- XOR state with 128-bits of the round key



Lab – AES bit flipping

- Changing the ciphertext to maliciously change the plaintext (attack to data integrity)
- AES Ciphertext:
- B60086CD1E68CEF25BC1BEC429D8F3C01D45F0196331DA5012B99067A25463A493CCBF690FD88F850BD5273C5A7D72B6
- Ciphertext 96-char, 48-byte long
- 3 AES 16-byte blocks
- Let's assume that
  - First block=username
  - Second block=email address
  - Third block=from *date* to *date* (validity)

AES bit flipping - 1

- Username=B60086CD1E68CEF25BC1BEC429D8F3C0
- Email=1D45F0196331DA5012B99067A25463A4
- From_to=93CCBF690FD88F850BD5273C5A7D72B6
  - FROM=93CCBF690FD88F85
  - TO=0BD5273C5A7D72B6
- The expected date format is YYYYMMDD
- In the cases of Counter Mode (CTR), Cipher Feedback Mode (CFB) and Output Feedback Mode (OFB), a bit-flip in the ciphertext will lead to a bit-flip in the plaintext at the same position
  - 0**B**D5273C5A7D72B6

AES bit flipping - 2

- In the case of CBC, a bit-flip modifies a bit in the plaintext of the following block at the same position
- Email=1D45F0196331DA5012B99067A25463A4
- Focus is on 12B99067A25463A4
- Format is YYYYMMDD
- Bit-flipping applies to 1**2**