Modern Symmetric Key Ciphers — AES

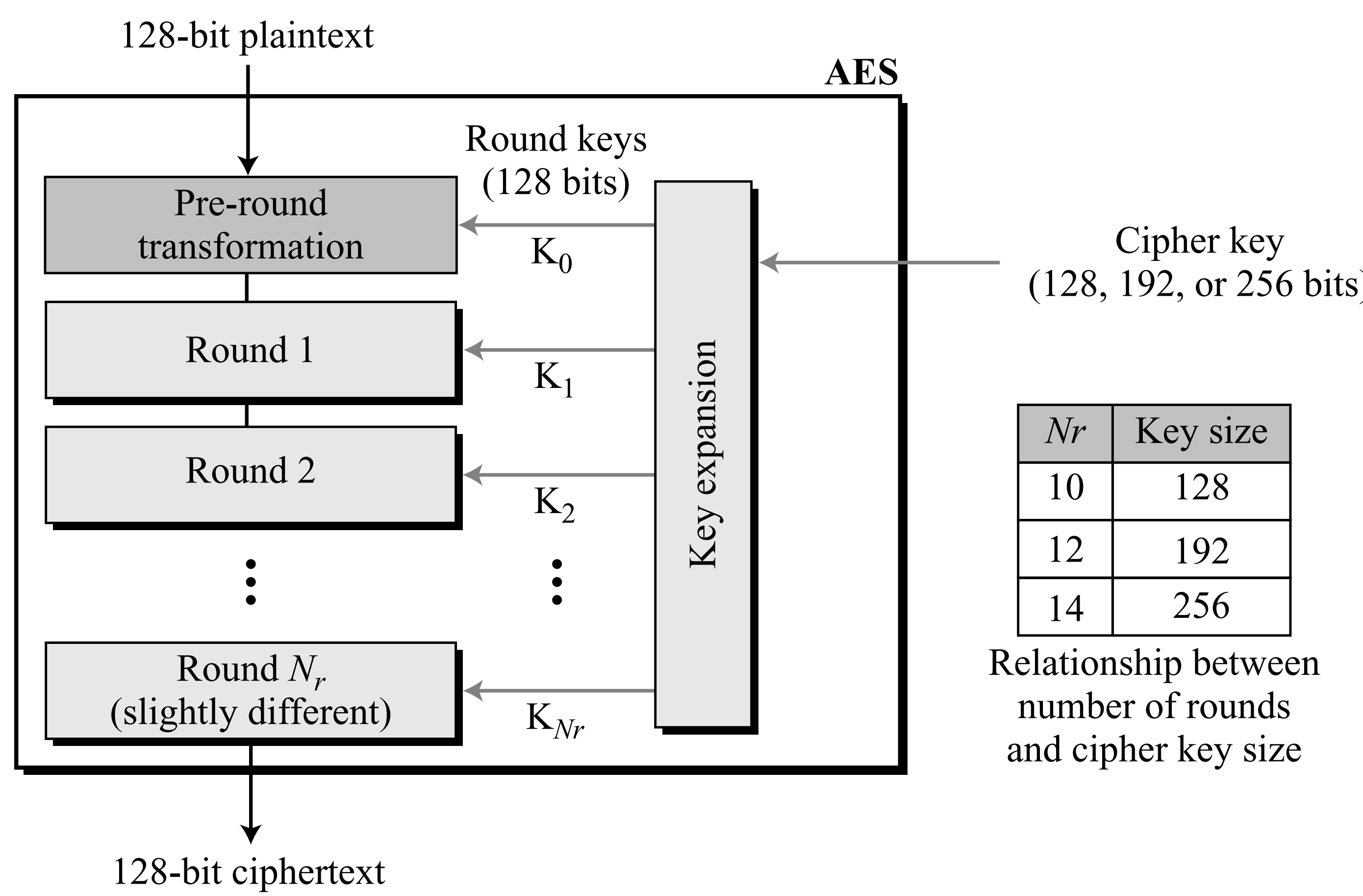
Part I: non-Fiestel AES

THE UNIVERSITY OF UTAH

Priyank Kalla
Professor
Electrical & Computer Engineering

AES

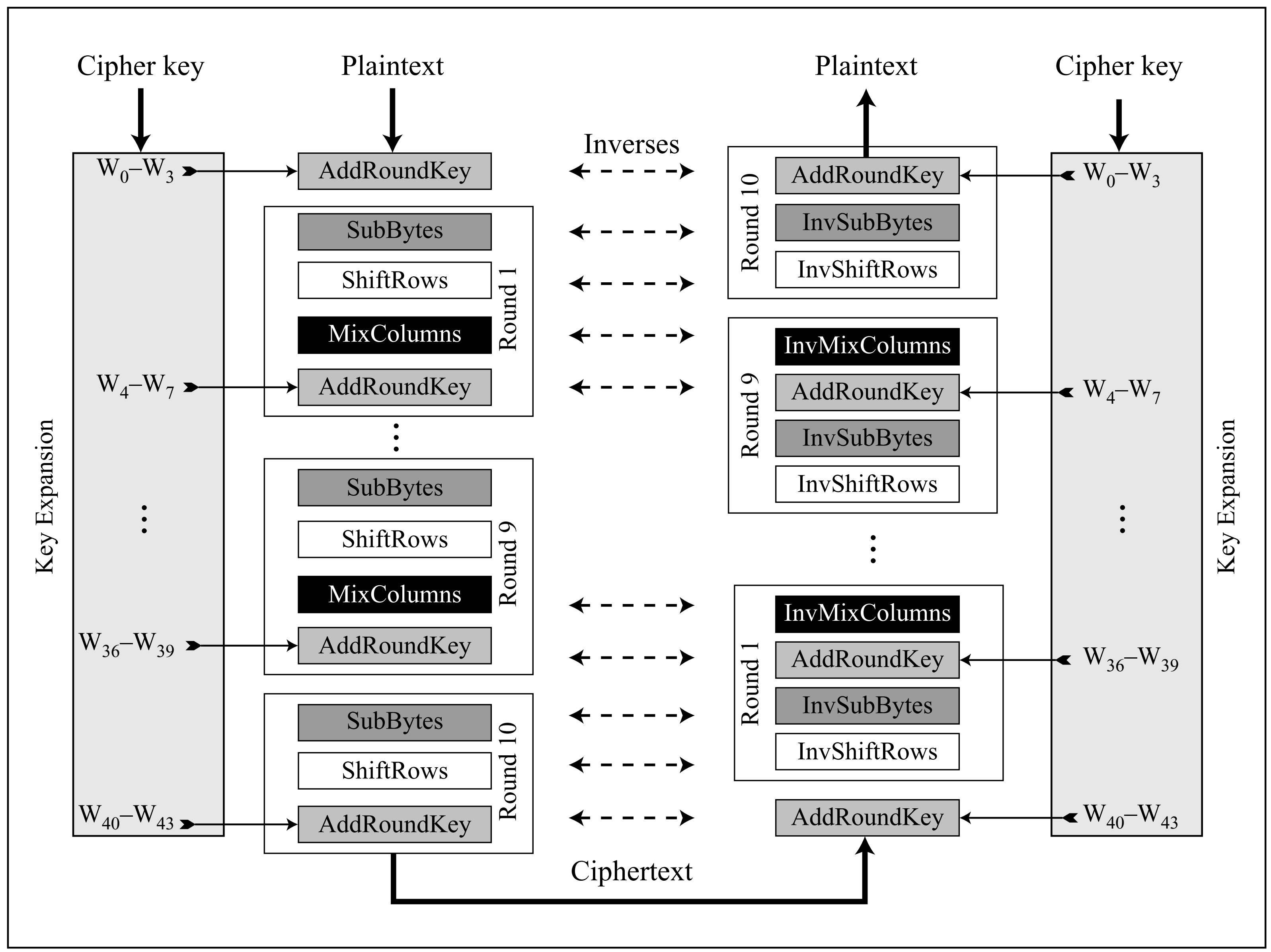
- NIST Standard ~2001
- Based on Rijndael Proposal: J. Daemen & V. Rijment
- 128-bit Block Cipher, and 128-bit internal round keys
- Number of round keys: $N_r + 1$
- Every operation is invertible: non-Feistel cipher



Nr	Key size
10	128
12	192
14	256

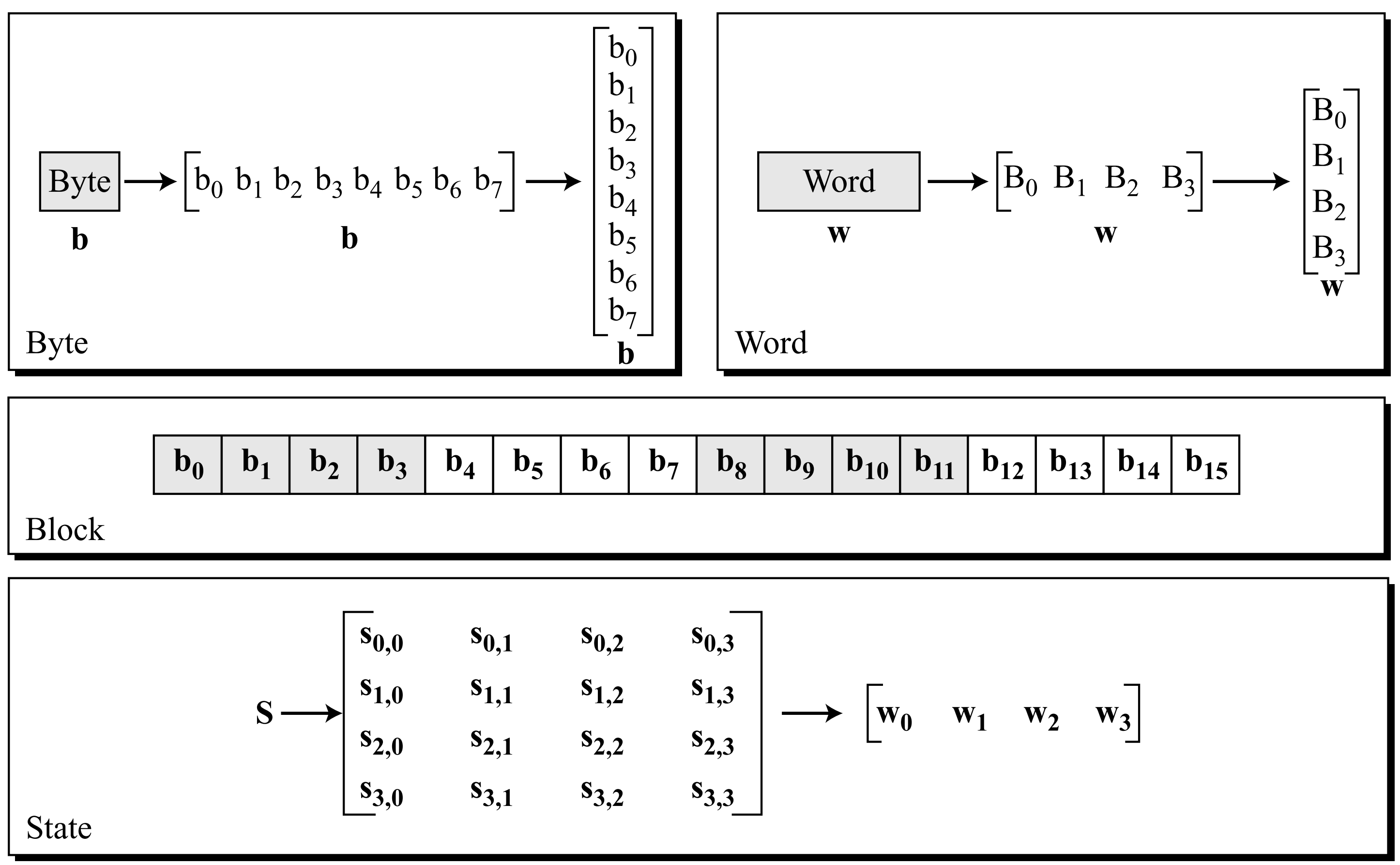
Relationship between number of rounds and cipher key size

A Basic High-Level View



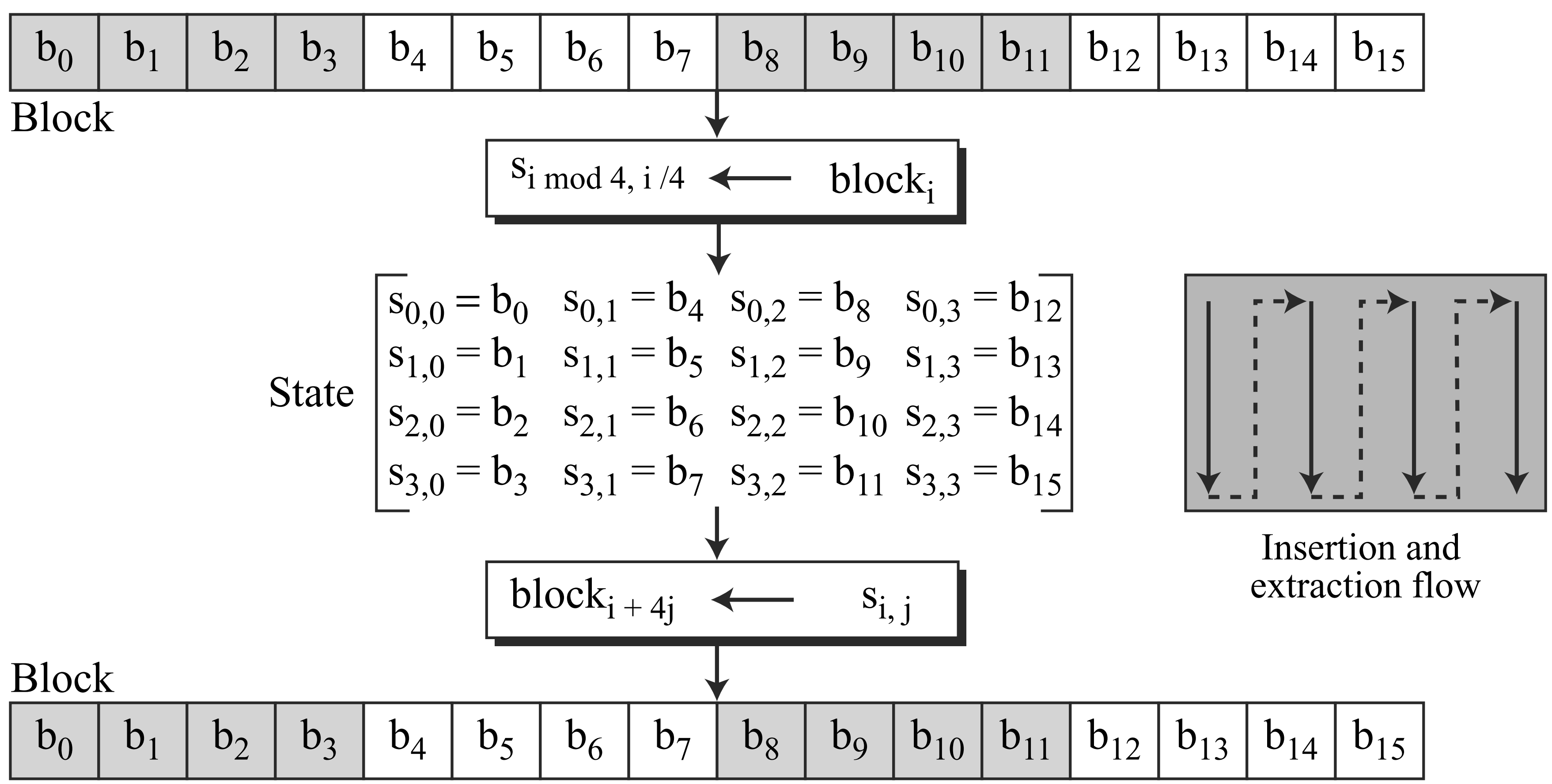
Data Units in AES

- Bits, Byte, Word (32 bits), Block (128 bit), and State
- State = Block, but inside the stages of AES, it is called a State; operations are performed as matrices



Blocks and States in Matrices

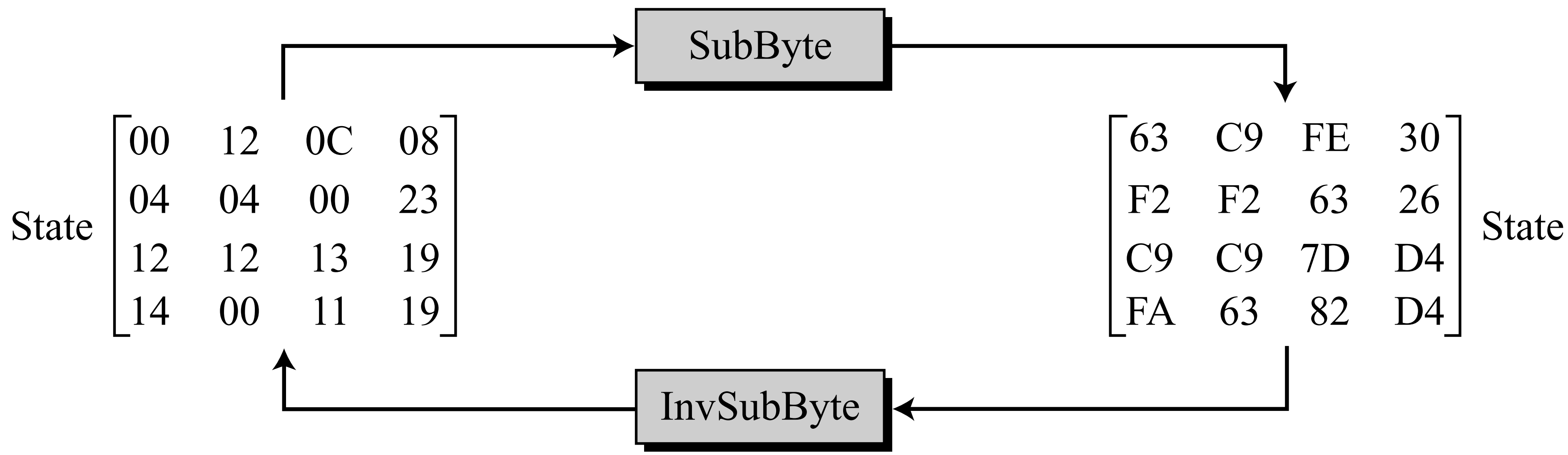
Block-to-state and state-to-block transformation



AES Transformations

- Substitutions, Permutations, Mixing, Key-Additions
- Substitution is performed on each “byte”
- One function to substitute each byte
- Substitution uses $\mathbb{F}_{2^8}$: $P(x) = x^8 + x^4 + x^3 + x + 1$ is used as the irreducible polynomial
- Computations done $\pmod{P(x)}$

The "SubByte" Transform



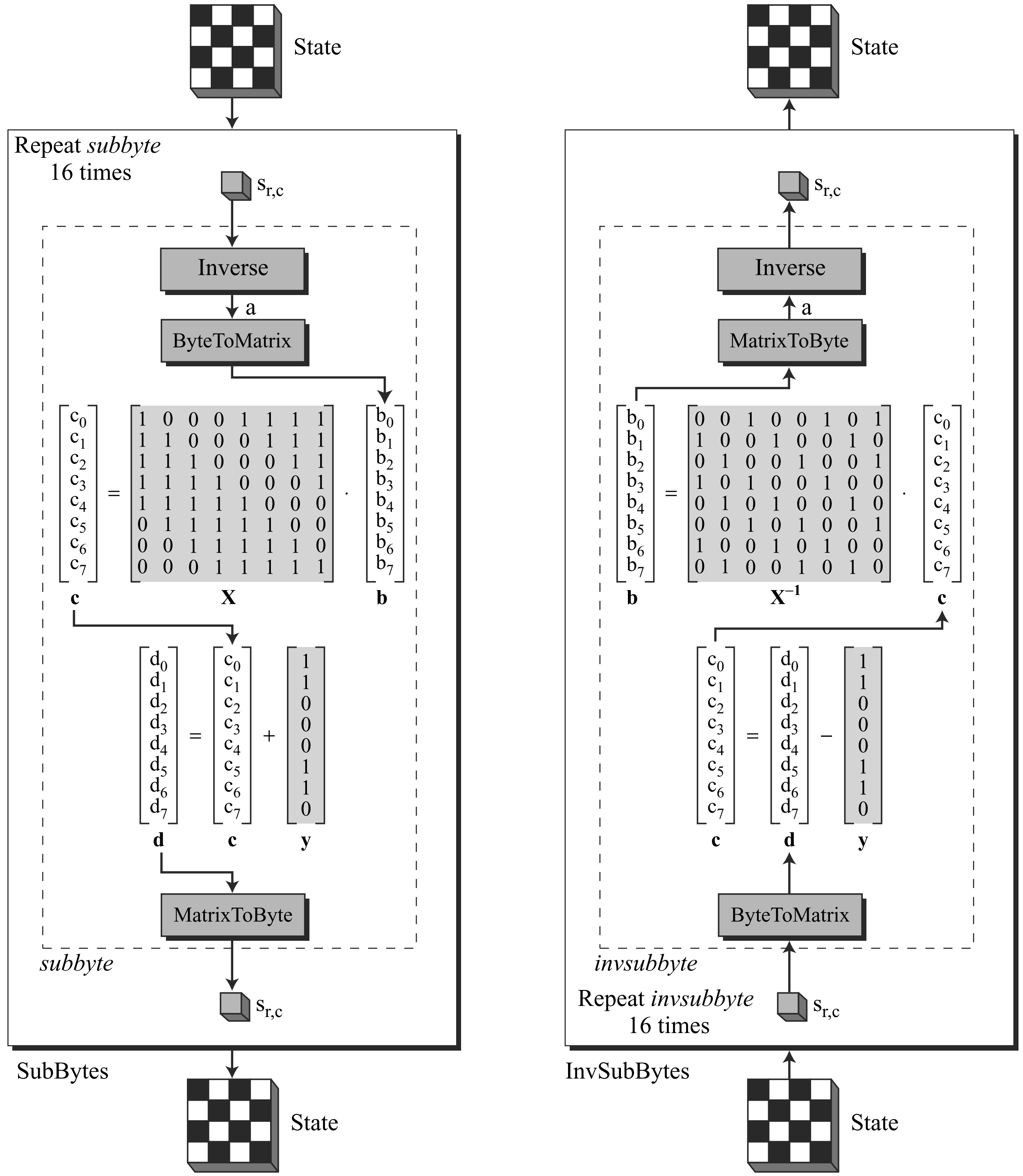
- Each element of the matrix = 2 hexadecimal words = byte

subbyte: $\rightarrow$ $\mathbf{d} = \mathbf{X}\,(s_{r,c})^{-1} \oplus \mathbf{y}$

invsubbyte: $\rightarrow$ $[\mathbf{X}^{-1}(\mathbf{d} \oplus \mathbf{y})]^{-1} = [\mathbf{X}^{-1}(\mathbf{X}\,(s_{r,c})^{-1} \oplus \mathbf{y} \oplus \mathbf{y})]^{-1} = [(s_{r,c})^{-1}]^{-1} = s_{r,c}$

- **The SubBytes and InvSubBytes transformations are inverses of each other.**

SubByte: Affine Transform



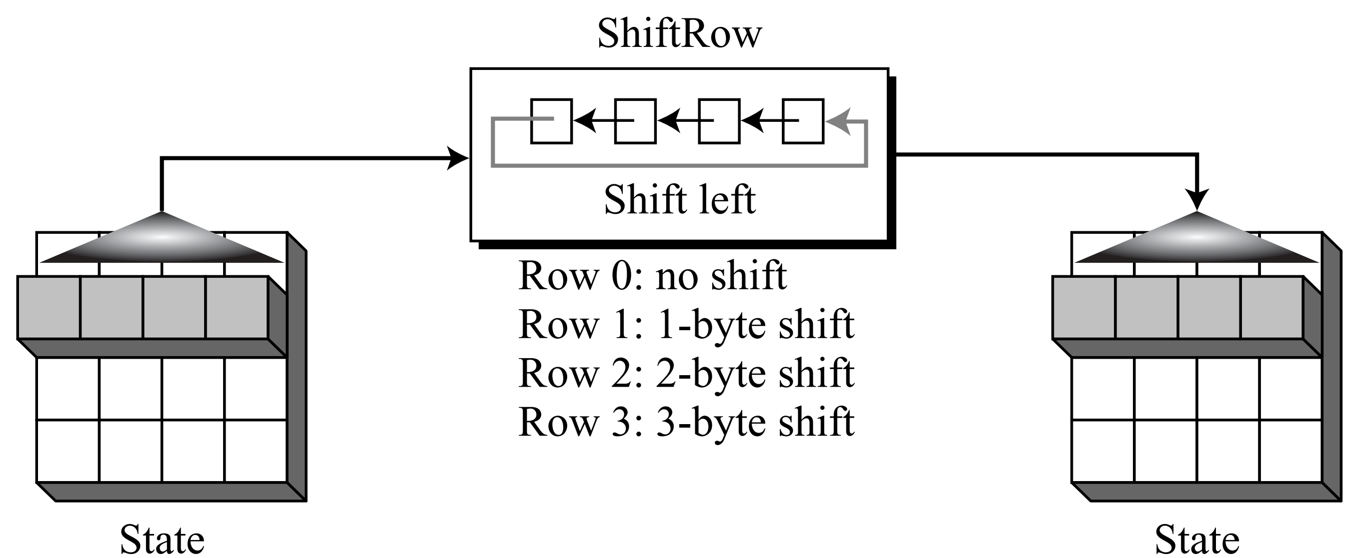
- Example: transform 0C to FE

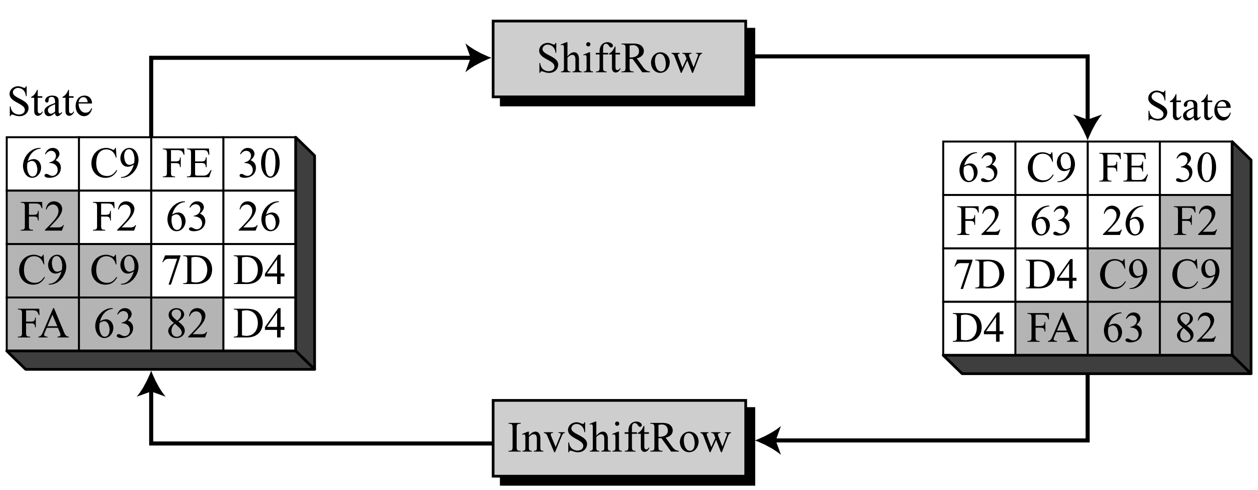
SubByte: Affine Transform

- 0C = 0000 1100 = $x^3 + x^2$
- Compute multiplicate inverse of 0C
  $(\mathrm{mod}\ x^8 + x^4 + x^3 + x + 1) = x^7 + x^5 + x^4$ = 1011 0000
- Multiply by matrix X = 10011101 = c
- XOR GF(2) addition operation to get d = 11111110 = FE
- Matrix X and vector y is fixed
- $s_{r,c} = X \cdot s_{r,c}^{-1} + y$: Affine cipher!! (Linear)
- But, the $s_{r,c}^{-1}$ operation makes it non-linear!

Shift-Row Transformation

- Cyclic Left Shift = permutation
- In decryption: do a cyclic right shift





Mixing Transform

- Substitution changes the value of the byte
- Permutation exchanges the bytes in the matrix
- We need an inter byte transformation: change bits inside a byte based on bits of neighbouring bytes: AES uses Mix Column Transformations

Mixing bytes using matrix multiplication

$$\begin{matrix} a\mathbf{x}+b\mathbf{y}+c\mathbf{z}+d\mathbf{t} \\ e\mathbf{x}+f\mathbf{y}+g\mathbf{z}+h\mathbf{t} \\ i\mathbf{x}+j\mathbf{y}+k\mathbf{z}+l\mathbf{t} \\ m\mathbf{x}+n\mathbf{y}+o\mathbf{z}+p\mathbf{t} \end{matrix} \begin{matrix} \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \end{matrix} \underbrace{\begin{bmatrix} \square \\ \square \\ \square \\ \square \end{bmatrix}}_{\text{New matrix}} = \underbrace{\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix}}_{\text{Constant matrix}} \cdot \underbrace{\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \\ \mathbf{t} \end{bmatrix}}_{\text{Old matrix}}$$

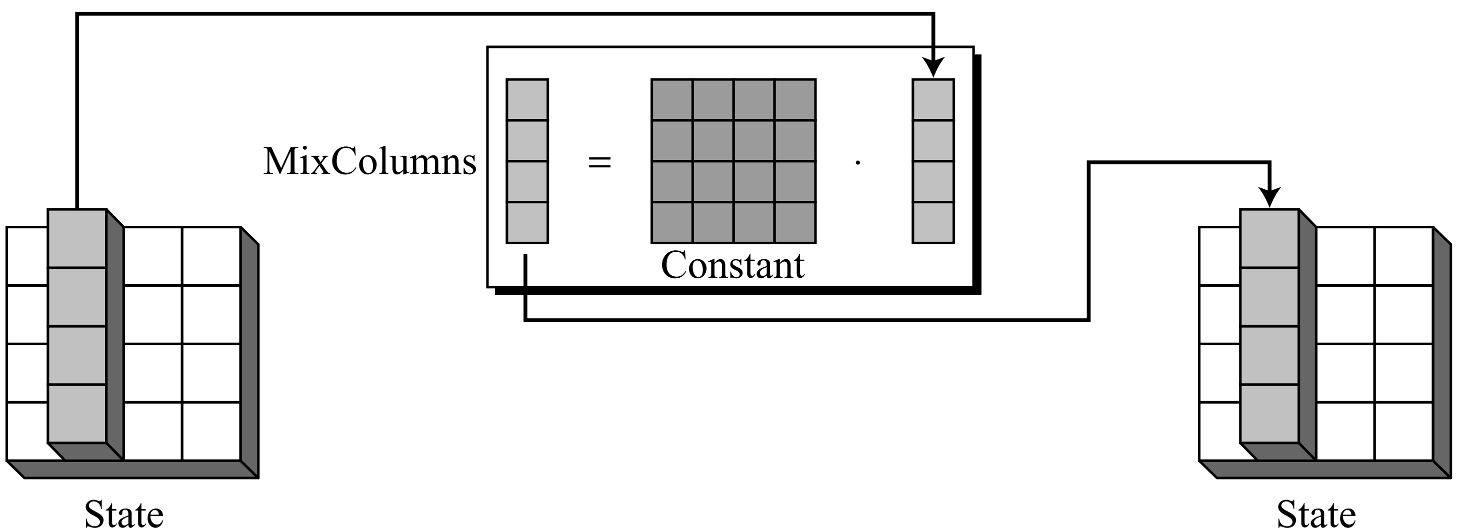
Multiplication done
$(\mathrm{mod}\ x^8 + x^4 + x^3 + x + 1)$
in $\mathbb{F}_{2^8}$

Constant matrices used by MixColumns and InvMixColumns

$$\underbrace{\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}}_{\mathrm{C}} \xleftrightarrow{\text{Inverse}} \underbrace{\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}}_{\mathrm{C}^{-1}}$$

Mixing Transform

- Substitution changes the value of the byte
- Permutation exchanges the bytes in the matrix
- We need an inter byte transformation: change bits inside a byte based on bits of neighbouring bytes: AES uses Mix Column Transformations



Multiplication done
$(\bmod\ x^8 + x^4 + x^3 + x + 1)$
in $\mathbb{F}_{2^8}$

Constant matrices used by MixColumns and InvMixColumns

$$\underset{C}{\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}} \xleftrightarrow{\text{Inverse}} \underset{C^{-1}}{\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}}$$

Add Round Key

- Addition modulo 2 = XOR
- Inverse of each other