Modern Symmetric Key Ciphers — AES

Part II: AES Key Expansion, AES operations in $\mathbb{F}_{2^8}$, Overcoming the Limitations of DES

THE UNIVERSITY OF UTAH

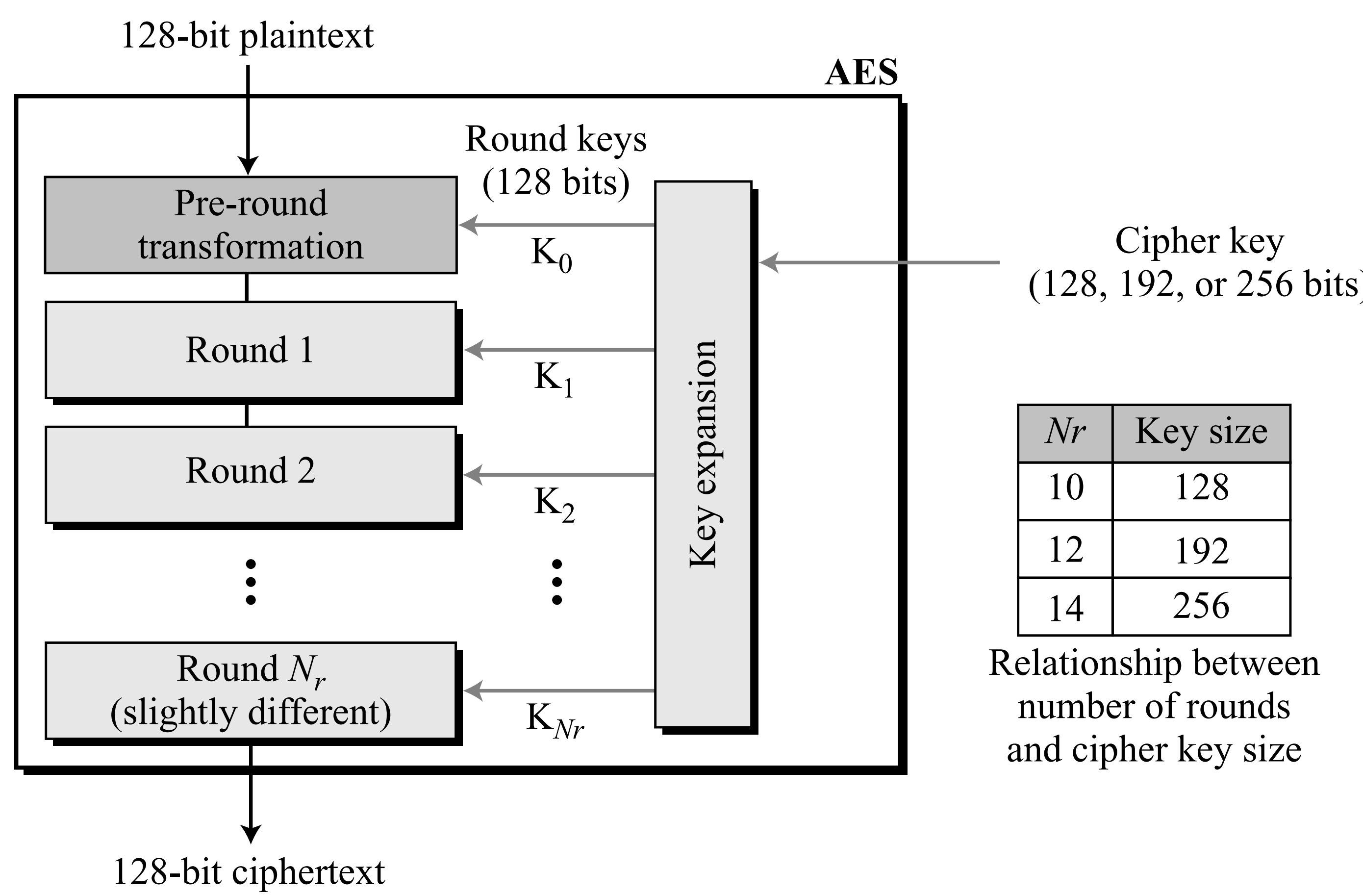
Priyank Kalla

Professor

Electrical & Computer Engineering

AES

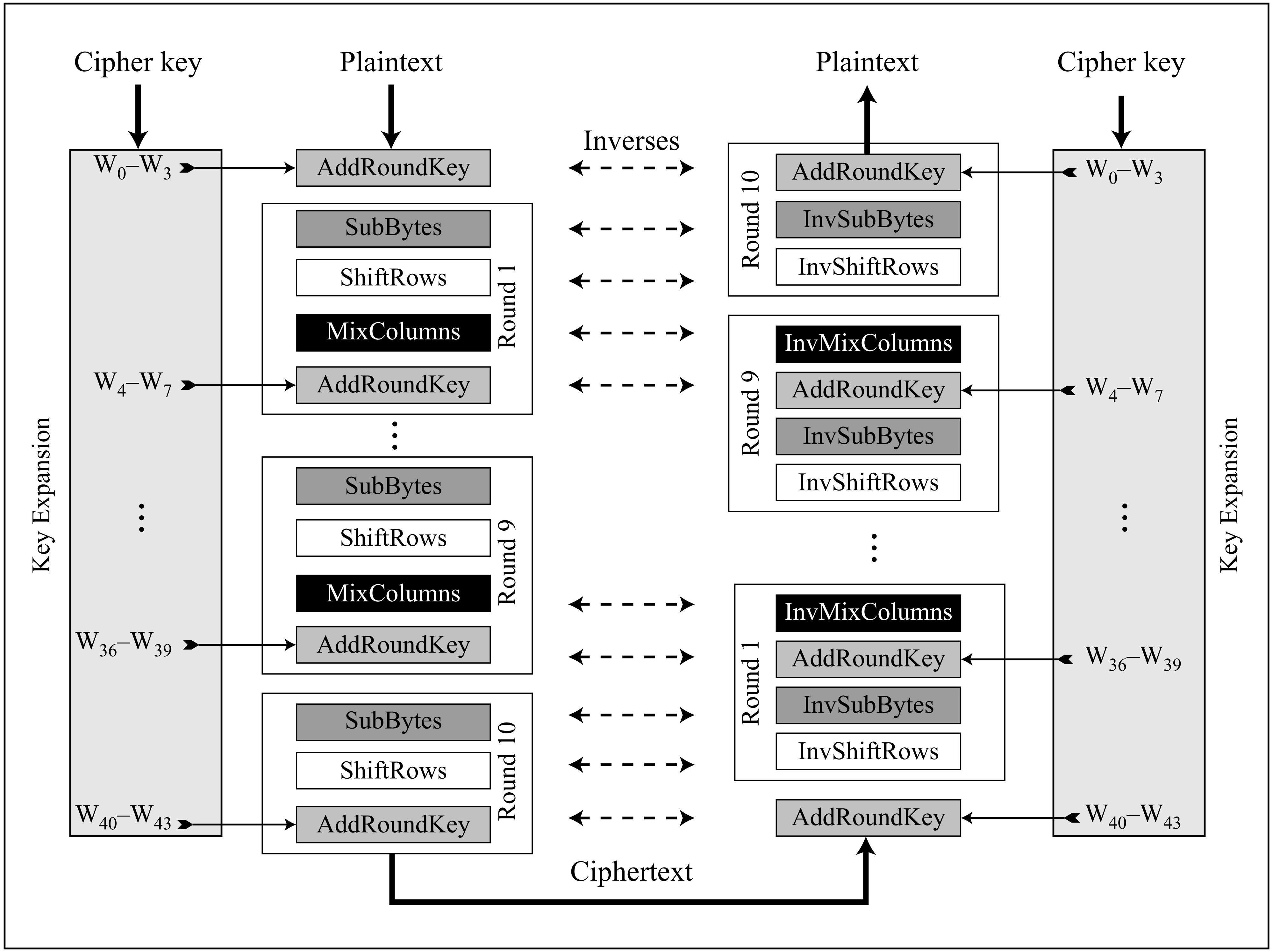
- NIST Standard ~2001
- Based on Rijndael Proposal: J. Daemen & V. Rijment
- 128-bit Block Cipher, and 128-bit internal round keys
- Number of round keys: $N_r + 1$
- Every operation is invertible: non-Feistel cipher



Nr	Key size
10	128
12	192
14	256

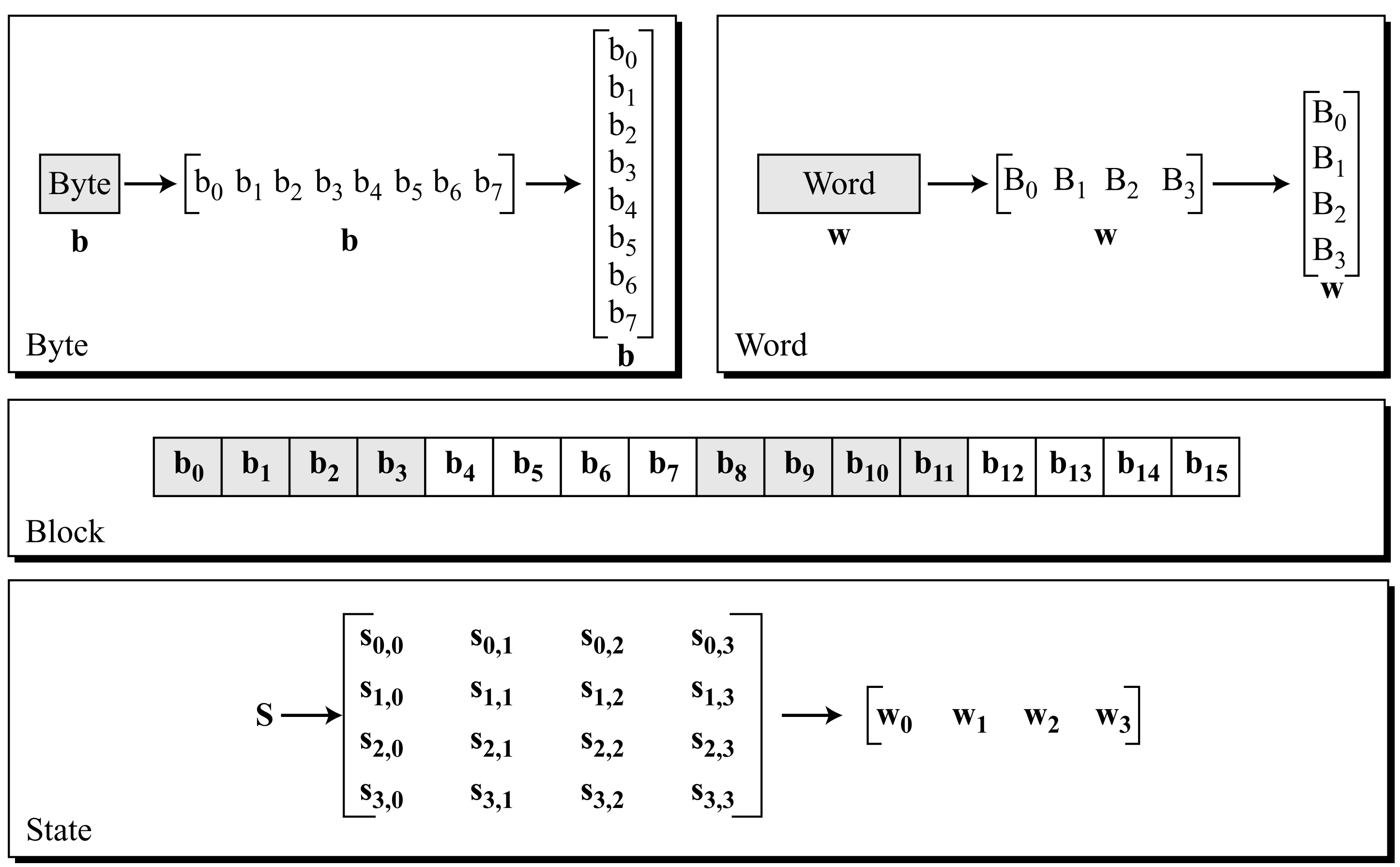
Relationship between number of rounds and cipher key size

A Basic High-Level View



Data Units in AES

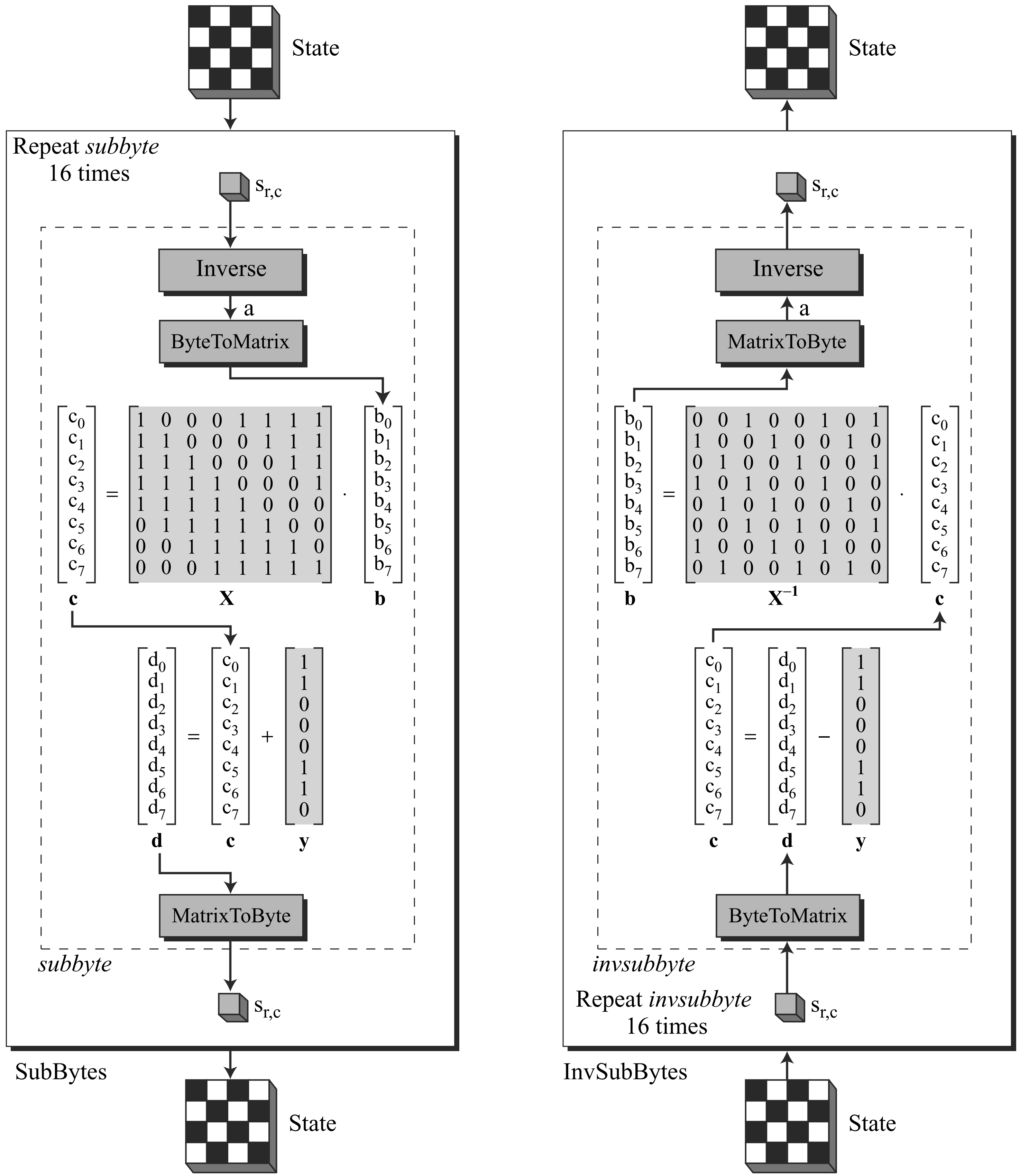
- Bits, Byte, Word (32 bits), Block (128 bit), and State
- State = Block, but inside the stages of AES, it is called a State; operations are performed as matrices



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 $\pmod{P(x)}$, $P(\alpha) = 0$

SubByte: Affine Transform



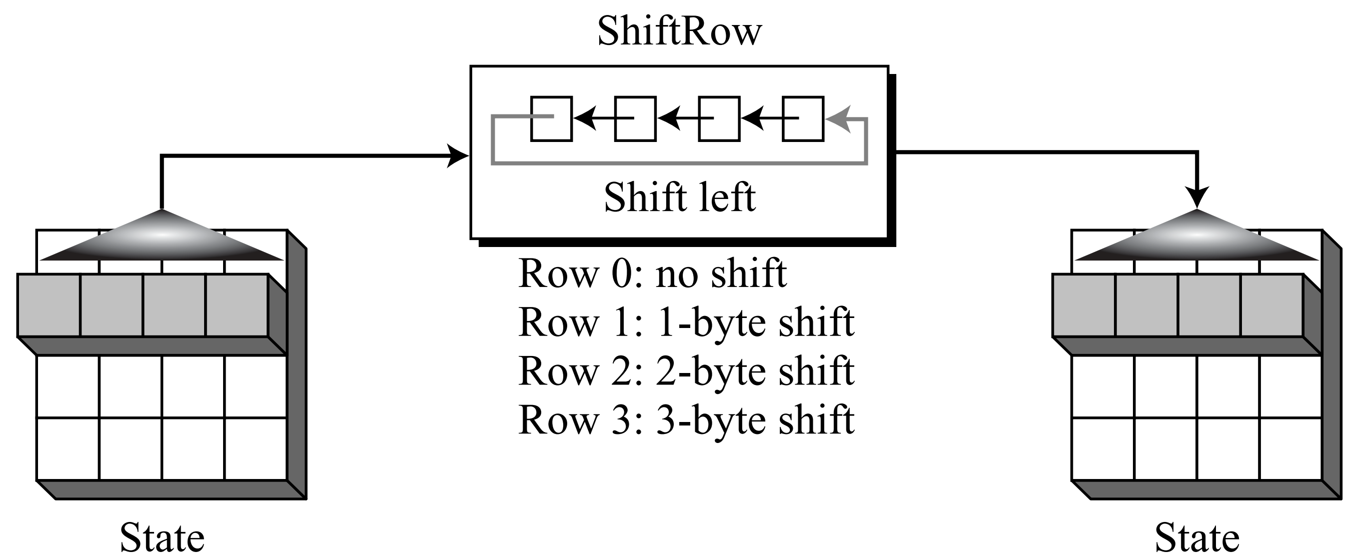
- Matrix X = given
- Matrix X^{-1} exists
- Vector y = given
- $d = Xb + y$

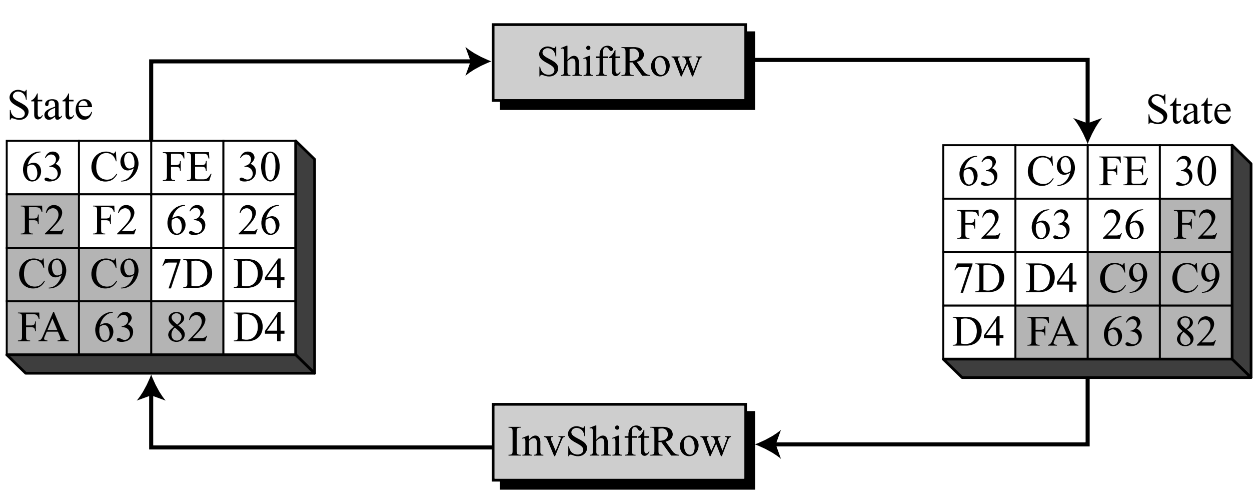
SubByte: Affine Transform

- 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

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

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

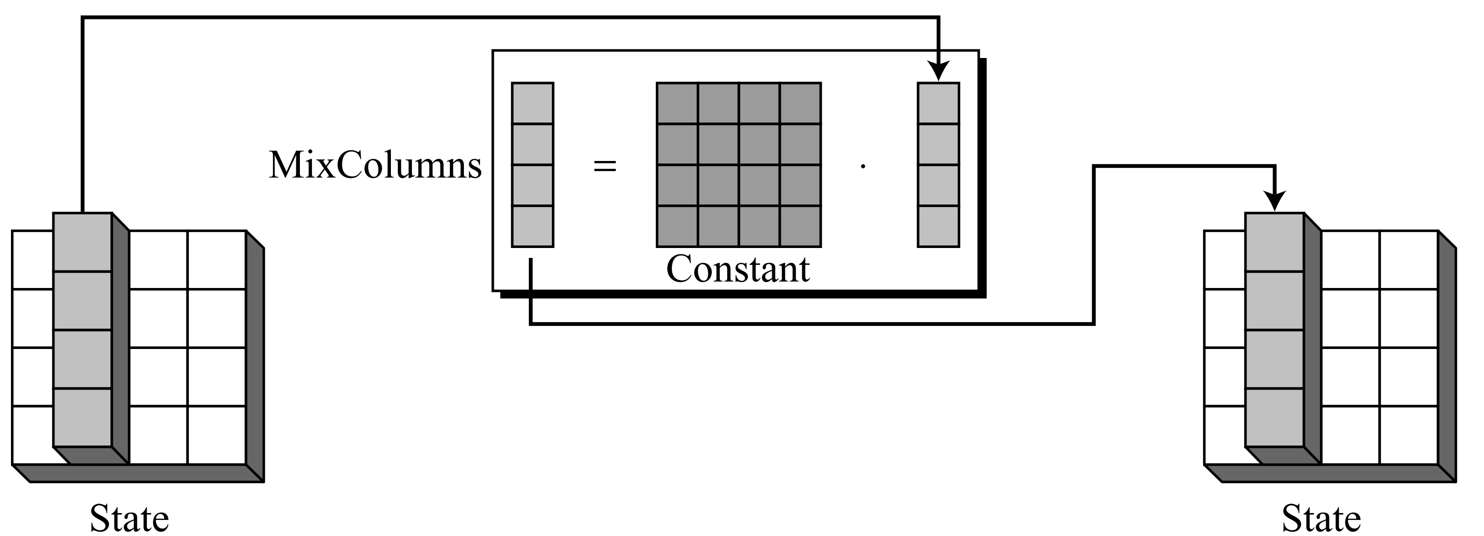
Constant matrices used by MixColumns and InvMixColumns

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

See Corresponding Singular File

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 **MixColumn** Transformations



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

Constant matrices used by MixColumns and InvMixColumns

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

Mixing Transform and Inv Mixing in $\mathbb{F}_{2^8}$

Constant matrices used by MixColumns and InvMixColumns

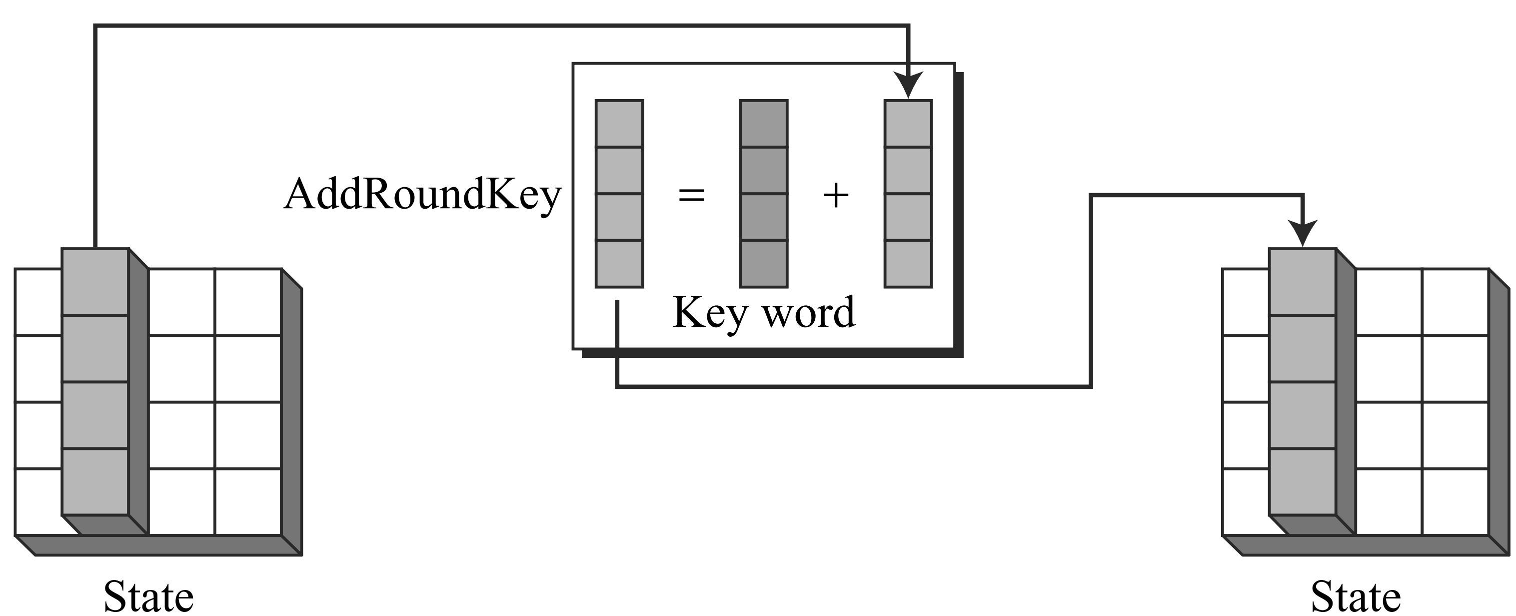
$$\underset{C}{\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}} \xleftrightarrow{\text{Inverse}} \underset{C^{-1}}{\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}}$$

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

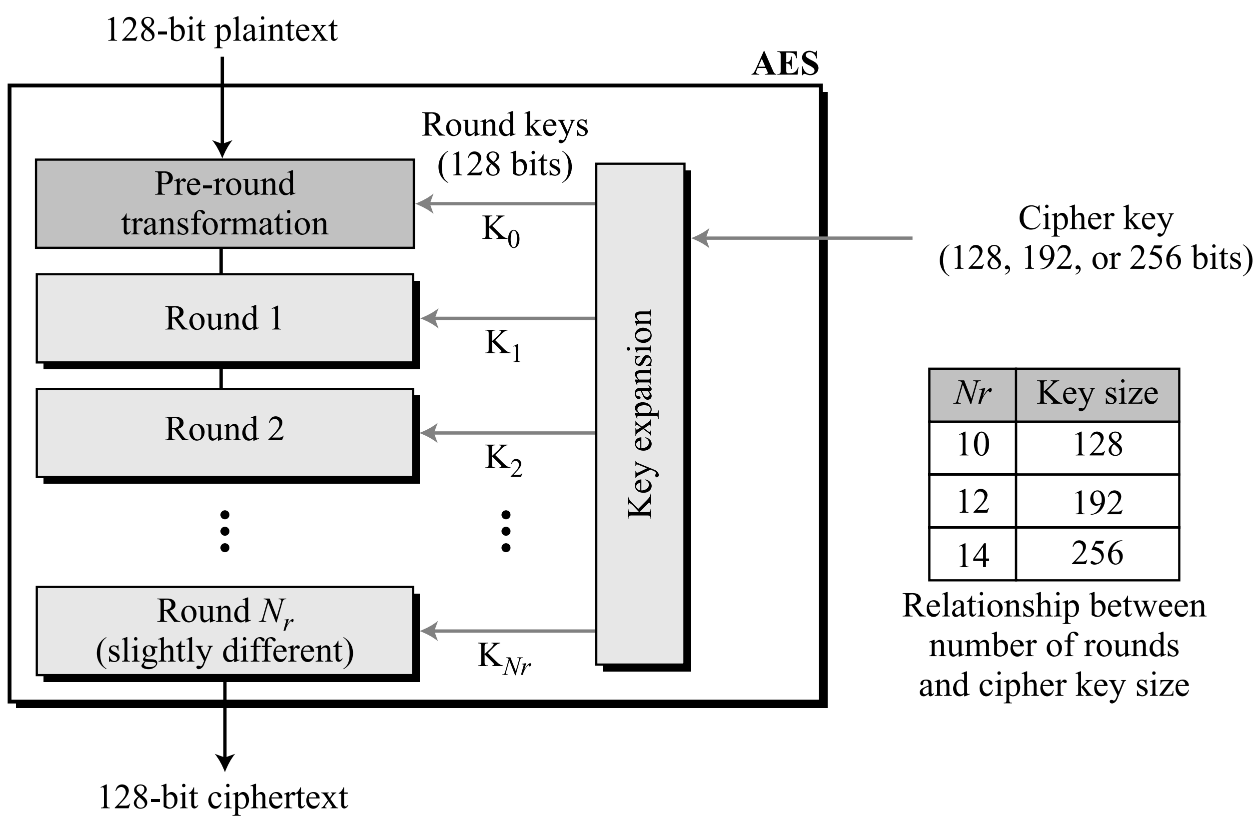
- 02 Hexadecimal = 0000 0010 = α; 03 Hex = 0000 0011 = $\alpha + 1$
- C = $\begin{bmatrix} \alpha & \alpha+1 & 1 & 1 \\ 1 & \alpha & \alpha+1 & 1 \\ 1 & 1 & \alpha & \alpha+1 \\ \alpha+1 & 1 & 1 & \alpha \end{bmatrix}$
- $C^{-1} = \begin{bmatrix} (\alpha^3+\alpha^2+\alpha) & (\alpha^3+\alpha+1) & (\alpha^3+\alpha^2+1) & (\alpha^3+1) \\ (\alpha^3+1) & (\alpha^3+\alpha^2+\alpha) & (\alpha^3+\alpha+1) & (\alpha^3+\alpha^2+1) \\ (\alpha^3+\alpha^2+1) & (\alpha^3+1) & (\alpha^3+\alpha^2+\alpha) & (\alpha^3+\alpha+1) \\ (\alpha^3+\alpha+1) & (\alpha^3+\alpha^2+1) & (\alpha^3+1) & (\alpha^3+\alpha^2+\alpha) \end{bmatrix}$
- $\alpha^3 + \alpha^2 + \alpha$ = 0000 1110 = 0E = 1st element of C^{-1}

Add Round Key

- Addition modulo 2 = XOR
- Inverse of each other



Key Expansion



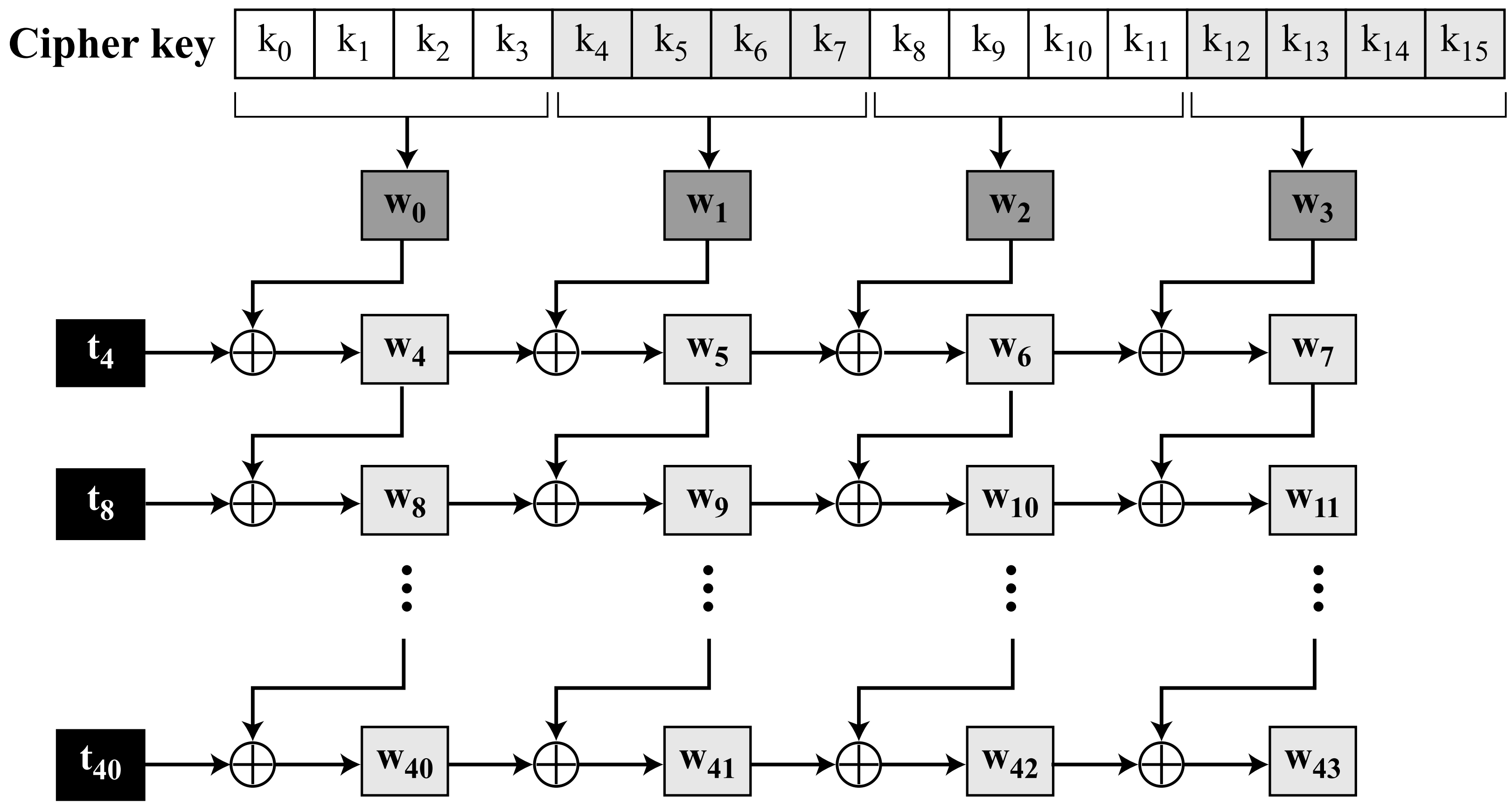
Nr	Key size
10	128
12	192
14	256

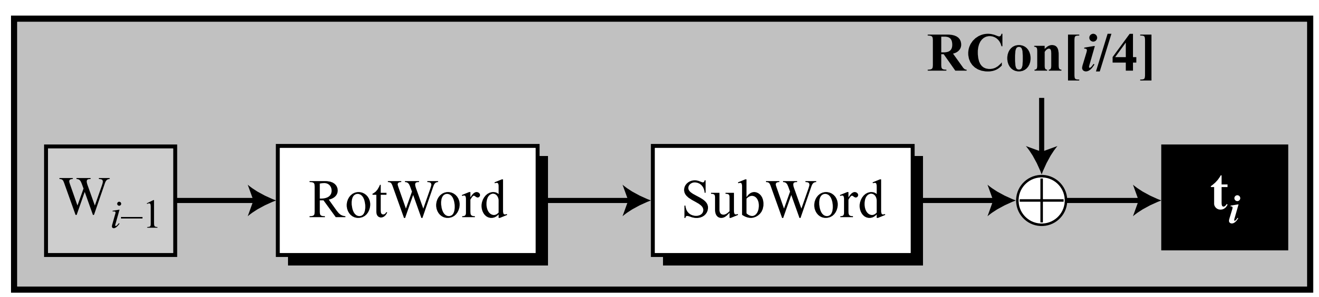
Relationship between number of rounds and cipher key size

Round	*Words*			
Pre-round	$\mathbf{w}_0$	$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$
1	$\mathbf{w}_4$	$\mathbf{w}_5$	$\mathbf{w}_6$	$\mathbf{w}_7$
2	$\mathbf{w}_8$	$\mathbf{w}_9$	$\mathbf{w}_{10}$	$\mathbf{w}_{11}$
. . .	. . .			
N_r	$\mathbf{w}_{4N_r}$	$\mathbf{w}_{4N_r+1}$	$\mathbf{w}_{4N_r+2}$	$\mathbf{w}_{4N_r+3}$

- N_r rounds = $N_r + 1$ Keys
- Each round Key = 128 bits = 4 words, 1 word = 4 bytes
- Words: $w_0, w_1, \ldots, w_{4(N_r+1)-1}$
- AES-128: $w_0, \ldots, w_{43}$

Key Expansion

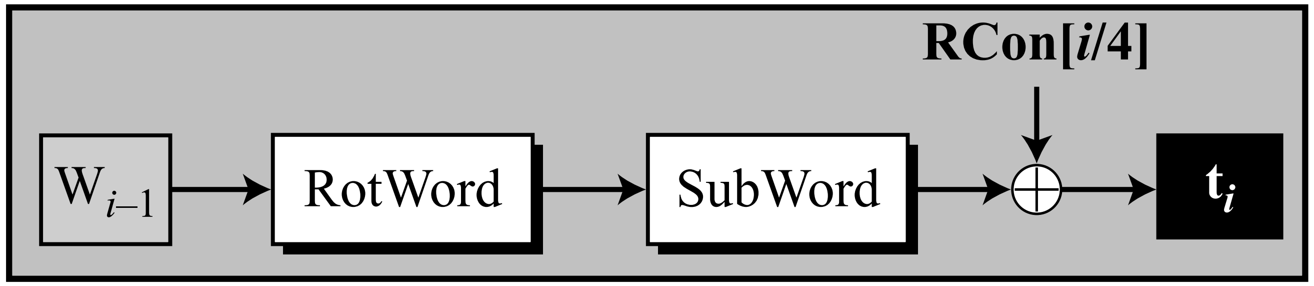




Making of t_i (temporary) words $i = 4\ N_r$

- t4 = SubWord(RotWord(w3)) + RCon[1] /*corrected*/

Key Expansion



Making of t_i (temporary) words $i = 4\ N_r$.

- t4 = SubWord(RotWord(w3)) + RCon[1]
- t8 = SubWord(RotWord(w7)) + RCon[2]
- t40 = SubWord(RotWord(w39)) + RCon[10]
- RotWord = Rotate = Cyclic Shift 1 bit
- SubWord = Substitution, substitution for each byte in the word
- RCon = Round Constant = which are given = $\alpha^{i-1} \pmod{P(\alpha)}$, i= the round

Round Constants

Round	*Constant (RCon)*	*Round*	*Constant (RCon)*
1	$(\underline{\mathbf{01}}\ 00\ 00\ 00)_{16}$	6	$(\underline{\mathbf{20}}\ 00\ 00\ 00)_{16}$
2	$(\underline{\mathbf{02}}\ 00\ 00\ 00)_{16}$	7	$(\underline{\mathbf{40}}\ 00\ 00\ 00)_{16}$
3	$(\underline{\mathbf{04}}\ 00\ 00\ 00)_{16}$	8	$(\underline{\mathbf{80}}\ 00\ 00\ 00)_{16}$
4	$(\underline{\mathbf{08}}\ 00\ 00\ 00)_{16}$	9	$(\underline{\mathbf{1B}}\ 00\ 00\ 00)_{16}$
5	$(\underline{\mathbf{10}}\ 00\ 00\ 00)_{16}$	10	$(\underline{\mathbf{36}}\ 00\ 00\ 00)_{16}$

$$P(\alpha) = \alpha^8 + \alpha^4 + \alpha^3 + \alpha + 1$$

$$
\begin{array}{llllll}
RC_1 & \to x^{1-1} = x^0 & \text{mod } \textit{prime} & = 1 & \to 00000001 & \to 01_{16} \\
RC_2 & \to x^{2-1} = x^1 & \text{mod } \textit{prime} & = x & \to 00000010 & \to 02_{16} \\
RC_3 & \to x^{3-1} = x^2 & \text{mod } \textit{prime} & = x^2 & \to 00000100 & \to 04_{16} \\
RC_4 & \to x^{4-1} = x^3 & \text{mod } \textit{prime} & = x^3 & \to 00001000 & \to 08_{16} \\
RC_5 & \to x^{5-1} = x^4 & \text{mod } \textit{prime} & = x^4 & \to 00010000 & \to 10_{16} \\
RC_6 & \to x^{6-1} = x^5 & \text{mod } \textit{prime} & = x^5 & \to 00100000 & \to 20_{16} \\
RC_7 & \to x^{7-1} = x^6 & \text{mod } \textit{prime} & = x^6 & \to 01000000 & \to 40_{16} \\
RC_8 & \to x^{8-1} = x^7 & \text{mod } \textit{prime} & = x^7 & \to 10000000 & \to 80_{16} \\
RC_9 & \to x^{9-1} = x^8 & \text{mod } \textit{prime} & = x^4 + x^3 + x + 1 & \to 00011011 & \to 1B_{16} \\
RC_{10} & \to x^{10-1} = x^9 & \text{mod } \textit{prime} & = x^5 + x^4 + x^2 + x & \to 00110110 & \to 36_{16}
\end{array}
$$

AES-192 & AES-256: Key Expansion

- Block-size is still 128-bits = Four 32-bit words
- AES-192: Words are generated in groups of 6, but each round key is still 128-bits wide (4 32-bit words)
  - $w_0, \ldots, w_5$: created directly by Cipher key, for Round 0
  - If $i \pmod 6 \neq 0 : w_i \leftarrow w_{i-1} + w_{i-6}$
  - Else, $w_i \leftarrow t_i + w_{i-6}$
  - $t_i = SubWord(RotWord(w_{i-1})) + RCon[i/6], \forall i \geq 6$
- AES-256: Keys are generated as a group of 8 words

AES Keys & Security

- Large key size thwarts brute-force attacks
- No differential or linear cryptanalysis attacks yet known (I’m not aware of any Quantum attack yet!)
- No statistical attacks: SubByte + ShiftRow + MixColum removes frequency pattern
- Similar cipher keys create vastly different round keys after round 2
- Round Constants RCons[ ] removes any symmetry created by other transforms
- Each bit of cipher-key is diffused into several round-keys. Change of a bit in cipher key causes changes in several round-keys
- AES has lived on for ~25 years!

Key Generation: Security

- Consider two keys K1 and K2 that differ in just 1 bit
- They generate completely different round keys

Cipher Key 1: 12 45 A2 A1 23 31 A4 A3 B2 CC A**A** 34 C2 BB 77 23
Cipher Key 2: 12 45 A2 A1 23 31 A4 A3 B2 CC A**B** 34 C2 BB 77 23

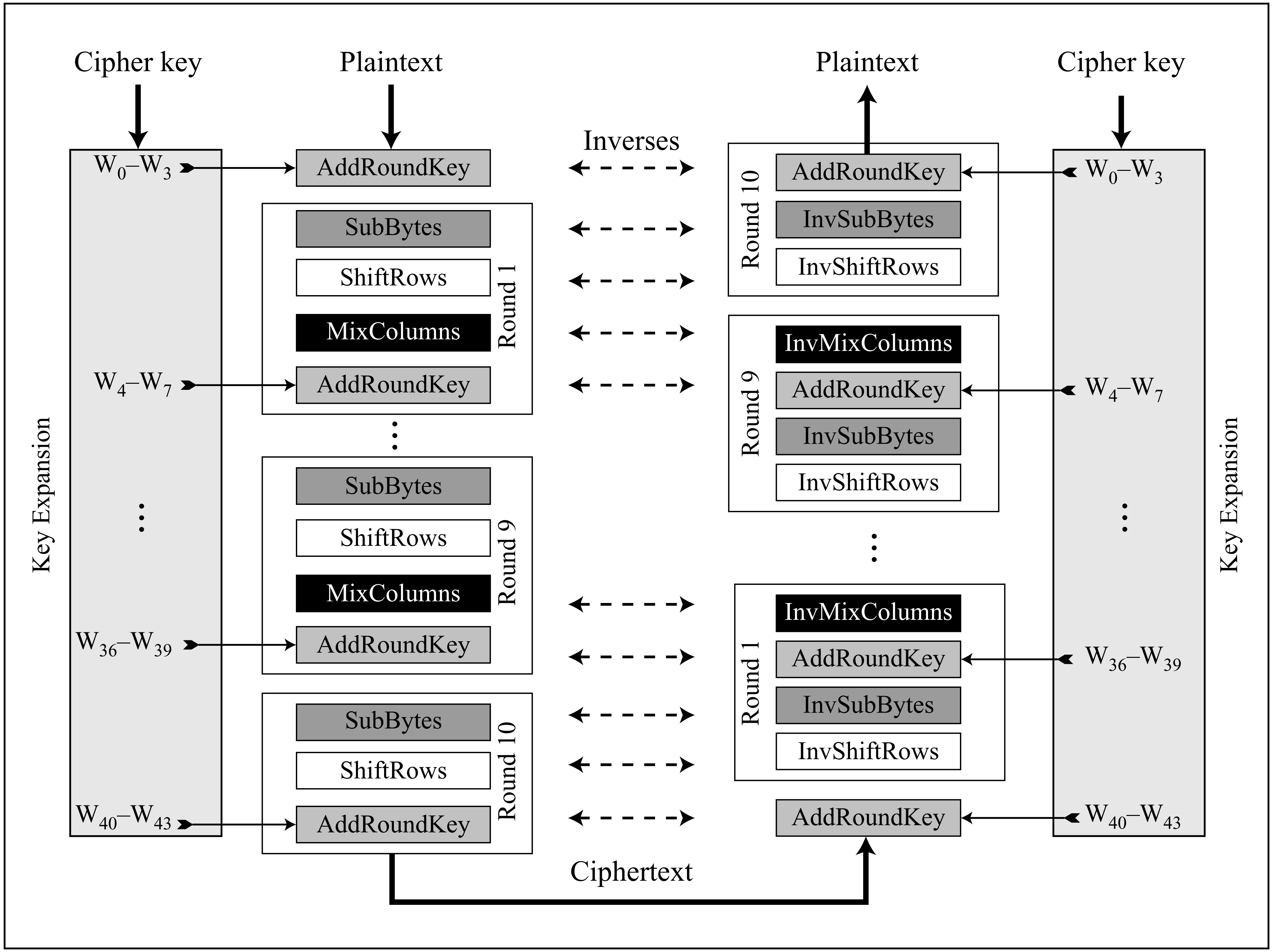
R.	*Round keys for set 1*	*Round keys for set 2*	*B. D.*
—	1245A2A1 2331A4A3 B2CCAA34 C2BB7723	1245A2A1 2331A4A3 B2CCAB34 C2BB7723	01
1	F9B08484 DA812027 684D8A13 AAF6FD30	F9B08484 DA812027 684D8B13 AAF6FC30	02
2	B9E48028 6365A00F 0B282A1C A1DED72C	B9008028 6381A00F 0BCC2B1C A13AD72C	17
3	A0EAF11A C38F5115 C8A77B09 6979AC25	3D0EF11A 5E8F5115 55437A09 F479AD25	30
4	1E7BCEE3 DDF49FF6 1553E4FF 7C2A48DA	839BCEA5 DD149FB0 8857E5B9 7C2E489C	31
5	EB2999F3 36DD0605 238EE2FA 5FA4AA20	A2C910B5 7FDD8F05 F78A6ABC 8BA42220	34
6	82852E3C B4582839 97D6CAC3 C87260E3	CB5AA788 B487288D 430D4231 C8A96011	56
7	82553FD4 360D17ED A1DBDD2E 69A9BDCD	588A2560 EC0D0DED AF004FDC 67A92FCD	50
8	D12F822D E72295C0 46F948EE 2F50F523	0B9F98E5 E7929508 4892DAD4 2F3BF519	44
9	99C9A438 7EEB31F8 38127916 17428C35	F2794CF0 15EBD9F8 5D79032C 7242F635	51
10	83AD32C8 FD460330 C5547A26 D216F613	E83BDAB0 FDD00348 A0A90064 D2EBF651	52

No AES Weak Keys

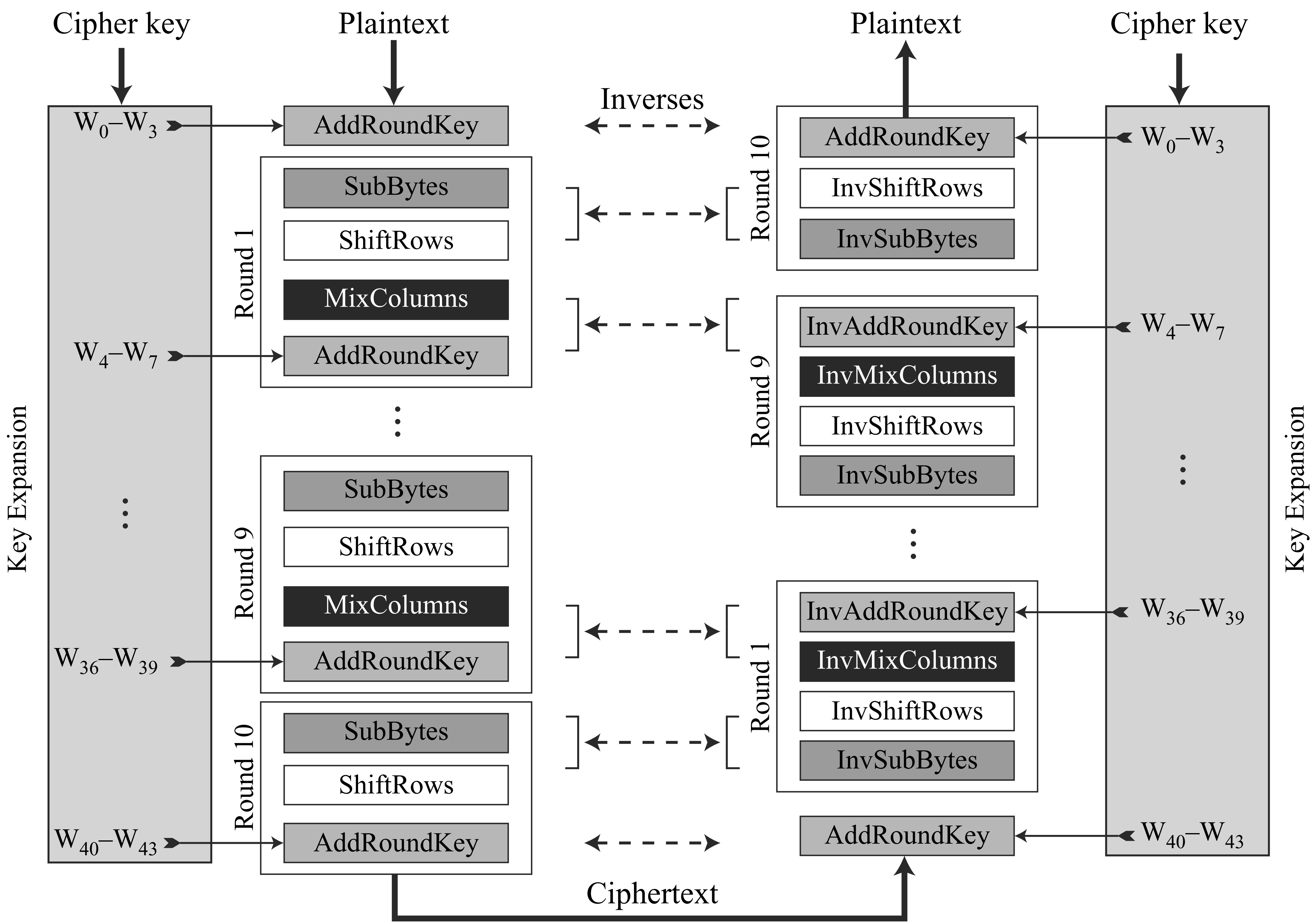
- When Cipher Key = ALL 0s:
- After Round 2, all keys are different

Pre-round:	00000000	00000000	00000000	00000000
Round 01:	62636363	62636363	62636363	62636363
Round 02:	9B9898C9	F9FBFBAA	9B9898C9	F9FBFBAA
Round 03:	90973450	696CCFFA	F2F45733	0B0FAC99
...	...	...	...	...
Round 10:	B4EF5BCB	3E92E211	23E951CF	6F8F188E

Original Cipher Design

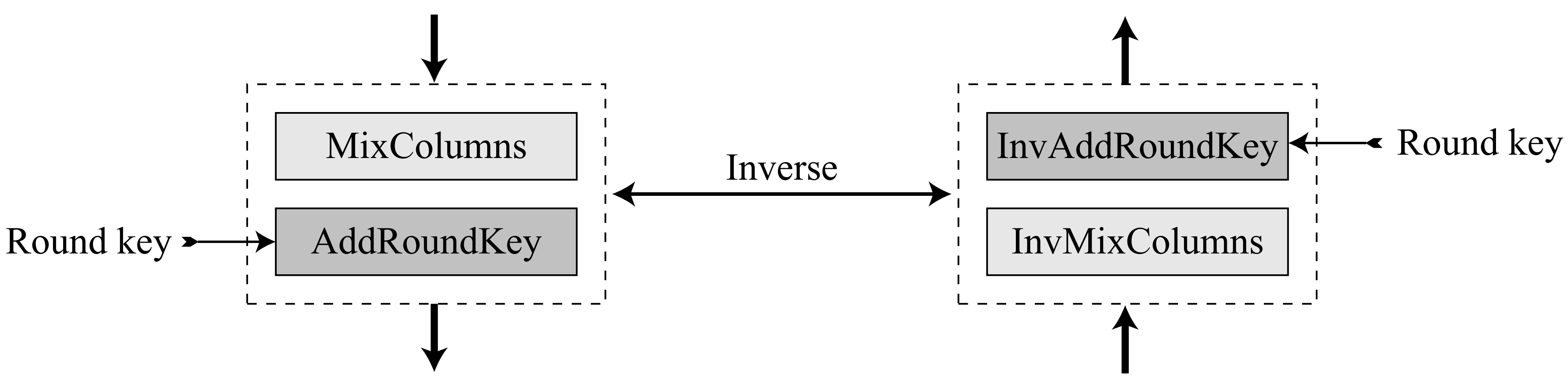


Alternate Design: Symmetrical Inverse



Alternate Design: Symmetrical Inverse

- Original AES: In decryption, "AddRoundKey" is used as is
- In the symmetrical Cipher, "InvAddRoundKey" is used
  - Multiply Key matrix with C^{-1}, where C is used in MixColumns



Cipher: $T = CS \oplus K$
Inverse Cipher: $C^{-1}T \oplus C^{-1}K = C^{-1}(CS \oplus K) \oplus C^{-1}K = C^{-1}CS \oplus C^{-1}K \oplus C^{-1}K = S$