Advanced Encryption Standard (AES)

(CS-452)

Background

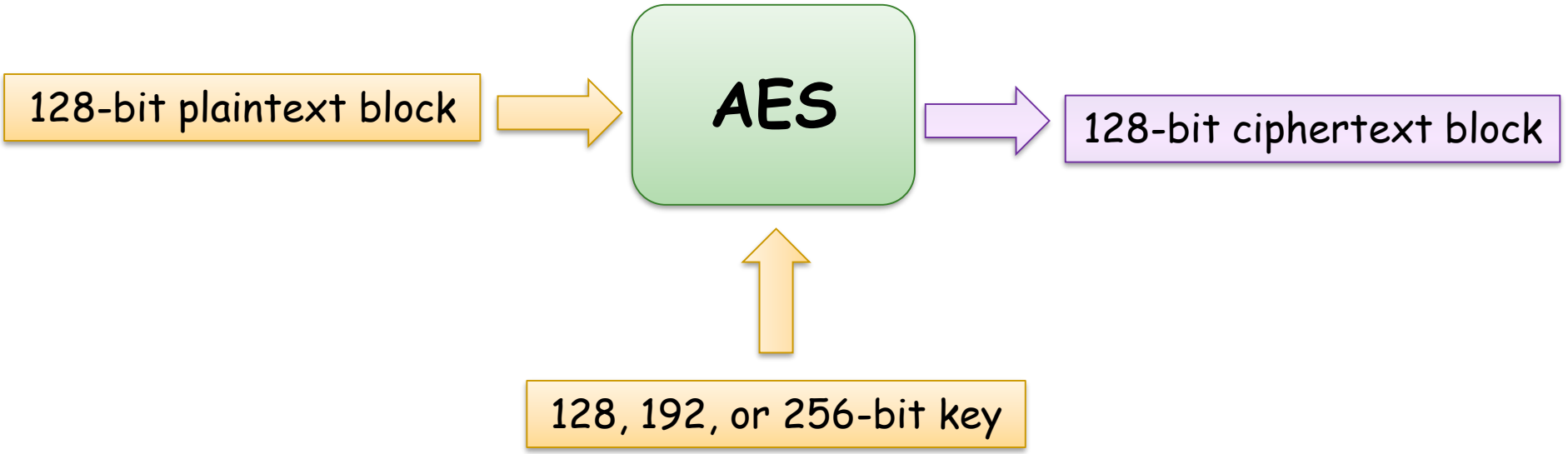
- DES is slow in software and 1-DES algorithm is vulnerable to bruteforce attacks (hence we needed 3-DES).
- In 1995 NIST started looking for an algorithm to replace DES, which was:
  - Unclassified
  - Publicly Disclosed
  - Available royalty-free for use worldwide
  - Symmetric block cipher algorithm operating on 128-bit blocks
  - Usable with key sizes of 128, 192, and 256 bits.
- From 15 contenders, the Rijndael algorithm of Dutch researchers Vincent Rijmen and Joan Daemen was chosen as the Advanced Encryption Standard (AES; FIPS 197).

AES vs DES

Characteristic	Data Encryption Standard (DES)	Advanced Encryption Standard (AES)
Security	DES is vulnerable to bruteforce attacks (hence need 3DES, which to date remains unbroken)	To date, AES has not been broken
Key Size	64-bit key	Supports key sizes of 128, 192, and 256 bit keys
Block Size	64-bit blocks	128-bit blocks
Structure	Fiestal Cipher Network	Substitution-Permutation Network
Number of Rounds	16 Rounds	10 rounds if 128-bit key is used 12 rounds if 192-bit key is used 14 rounds if 256-bit key is used

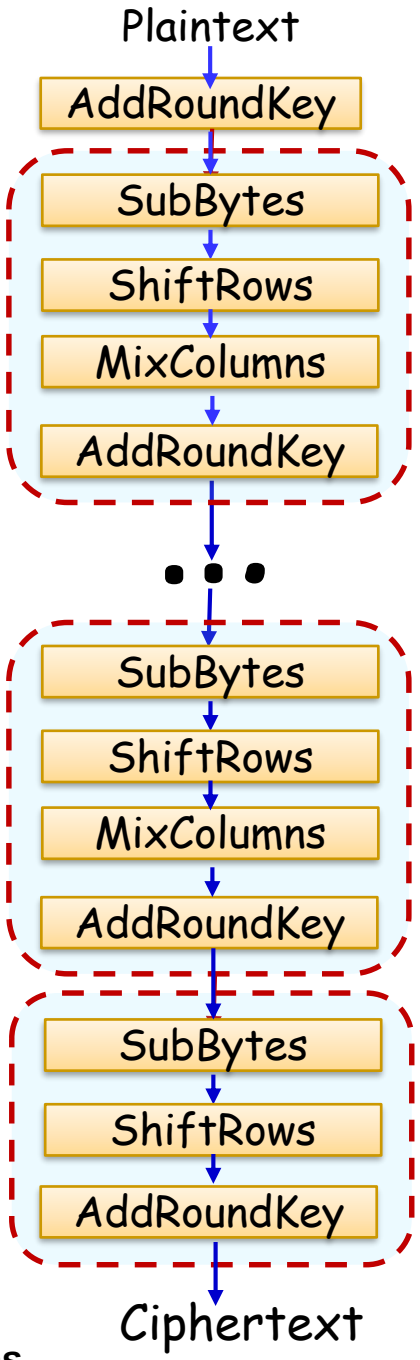
AES Overview

- AES accepts as 128-bit (i.e. 16 byte) plaintext block, a 128, 192, or 256-bit key, and produces a 128-bit ciphertext block.



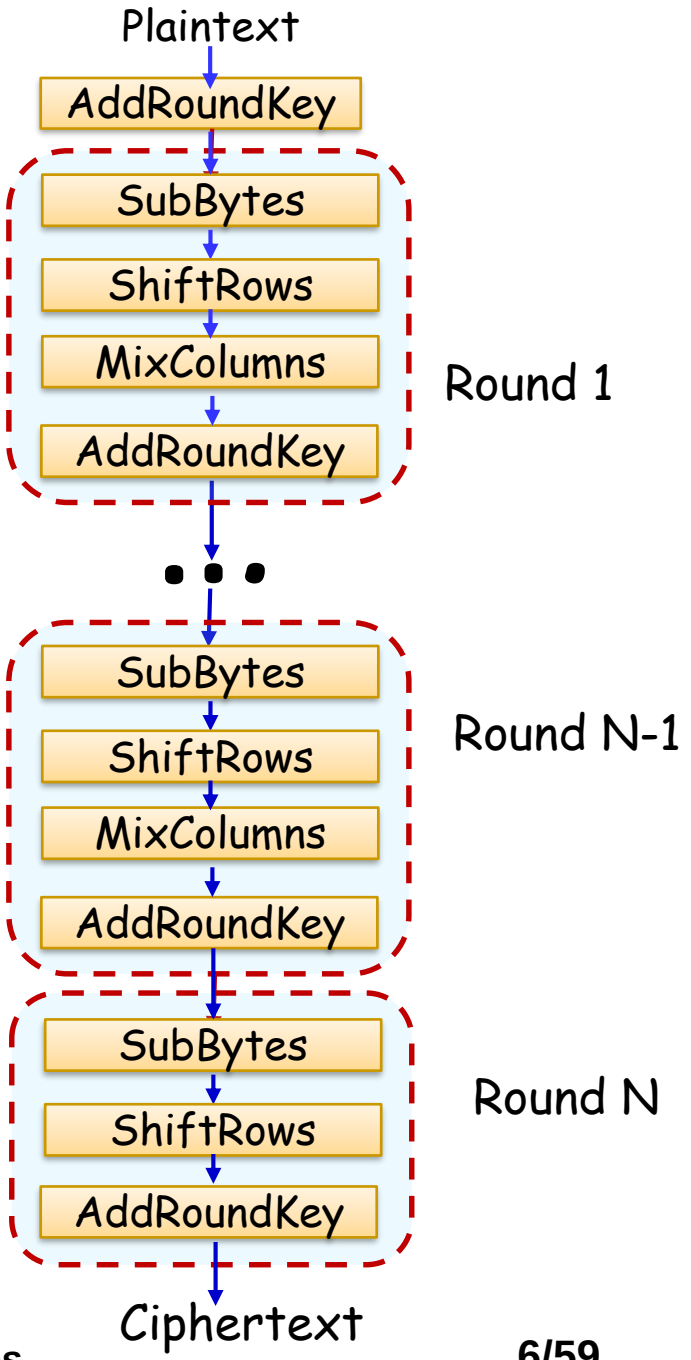
AES Structure

- 128-bit plaintext block is arranged as a 4 × 4 matrix of bytes called the "state".
- The "state" is initially XORed with the key (i.e. the first AddRoundKey operation)
- The "state" then undergoes N rounds of transformation where N is:
  - 10 when 128-bit key is used.
  - 12 when 192-bit key is used.
  - 14 when the 256-bit key is used.

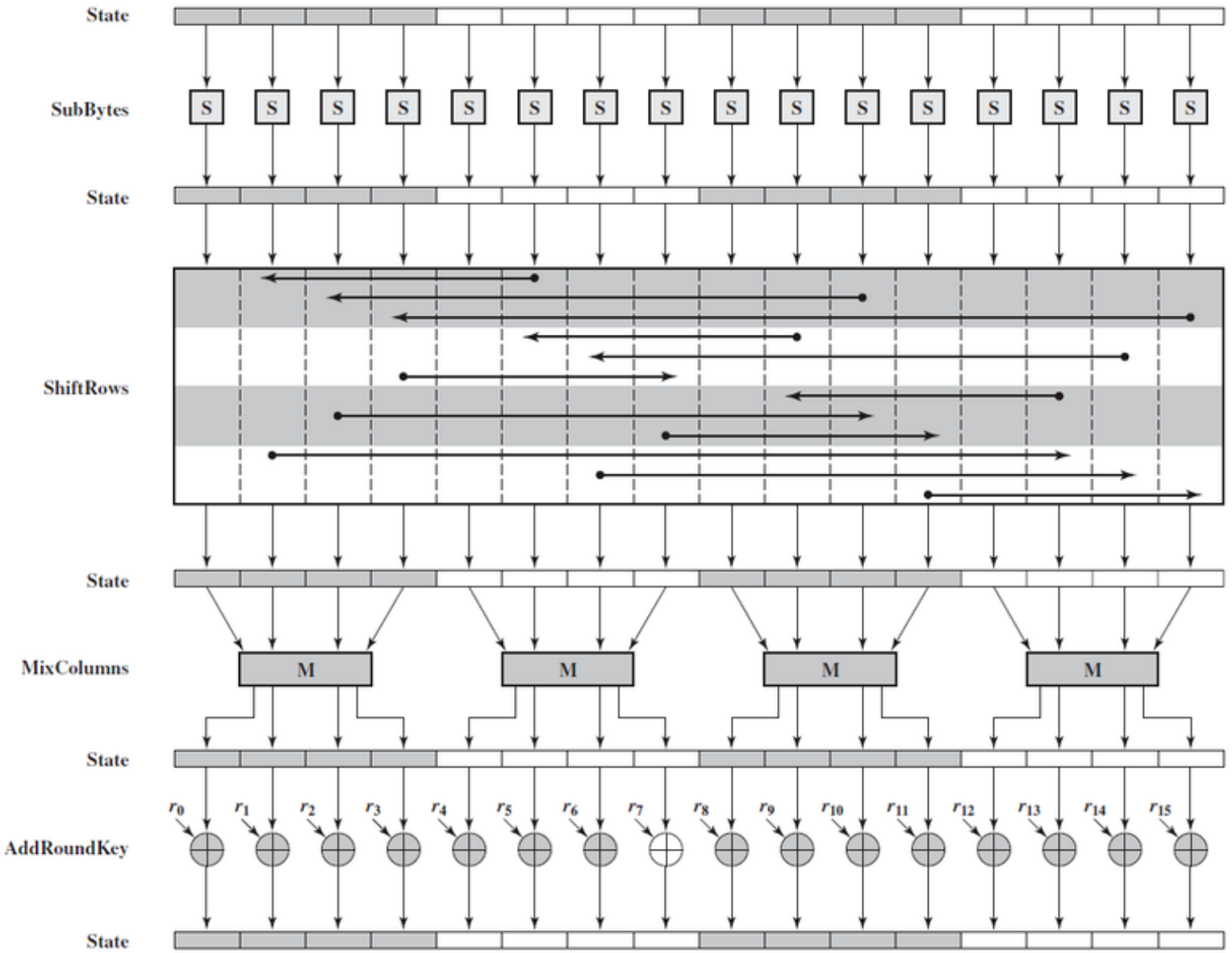


AES Structure (2)

- The first N - 1 rounds comprise 5 transformations on the plaintext:
  - SubBytes: substitute
  - ShiftRows
  - MixColumns
  - AddRoundKey
- The last round is similar to previous rounds, but omits the MixColumns transformation.



AES Structure: Structure of the Single Round



AES Encryption: Preparing the Plaintext

- Consider a 128-bit (i.e. 16 byte) plaintext block.
  - Each byte of the plaintext block is labeled as follows:

$(a_{0,0}, a_{1,0}, a_{2,0}, a_{3,0}, a_{0,1}, a_{1,1}, a_{2,1}, a_{3,1}, a_{0,2}, a_{1,2}, a_{2,2}, a_{3,2}, a_{0,3}, a_{1,3}, a_{2,3}, a_{3,3})$

- The bytes are then arranged in a 4 x 4 matrix i.e., the "state", as follows:

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a1_{,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a2_{,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$

Such arrangement is called a column major form

AES Encryption: Preparing the Plaintext

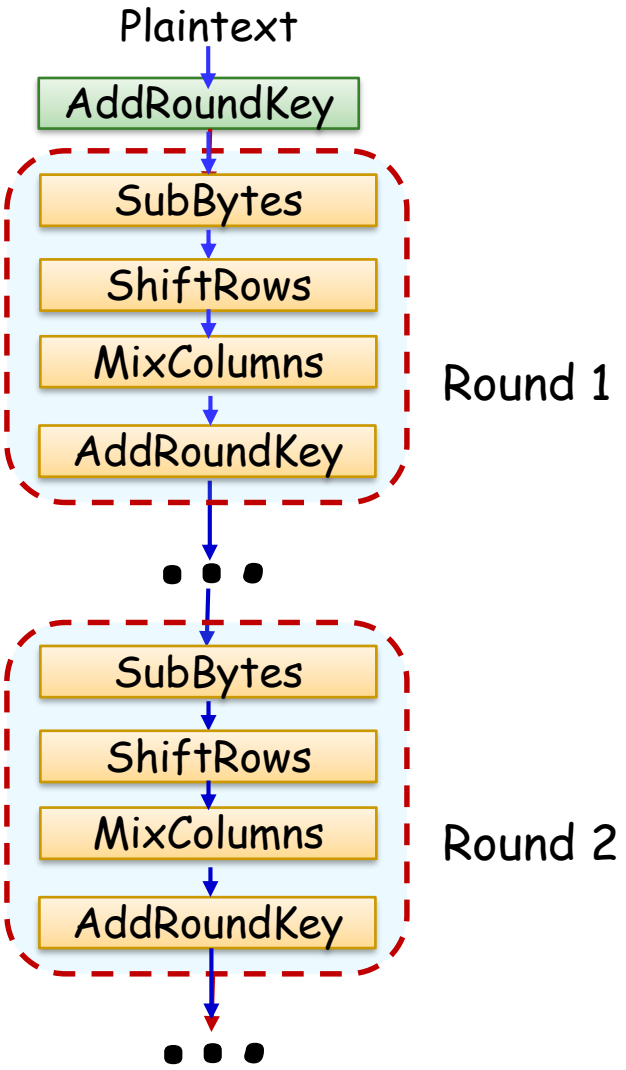
- Example: A 128-bit plaintext block:

EA 83 5C F0 04 45 33 2D 65 5D 98 AD 85 96 B0 C5 is converted into the following matrix.

$$\begin{bmatrix} EA & 04 & 65 & 85 \\ 83 & 45 & 5D & 96 \\ 5C & 33 & 98 & B0 \\ F0 & 2D & AD & C5 \end{bmatrix}$$

AES Encryption: The Initial AddRoundKey Transformation

- Next, the initial AddRoundKey operation is performed.
  - A simple bitwise XOR between the bytes of the key and bytes of the state.



AES Encryption: The Initial AddRoundKey Transformation

- Example:
  - Key: 47 37 94 ED 40 D4 E4 A5 A3 70 3A A6 4C 9F 42 BC
  - Plaintext: AC 77 66 F3 19 FA DC 21 28 D1 29 41 57 5C 00 6A
  - Both written in a 4x4 column-major matrices:
    - Key:

$$\begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix}$$

    - Plaintext:

$$\begin{bmatrix} AC & 19 & 28 & 57 \\ 77 & FA & D1 & 5C \\ 66 & DC & 29 & 00 \\ F3 & 21 & 41 & 6A \end{bmatrix}$$

AES Encryption (6): The Initial AddRoundKey Transformation

- Example:
  - Key: 47 37 94 ED 40 D4 E4 A5 A3 70 3A A6 4C 9F 42 BC
  - Plaintext: AC 77 66 F3 19 FA DC 21 28 D1 29 41 57 5C 00 6A
  - The matrices are then XORed together:

$$\begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix} \oplus \begin{bmatrix} AC & 19 & 28 & 57 \\ 77 & FA & D1 & 5C \\ 66 & DC & 29 & 00 \\ F3 & 21 & 41 & 6A \end{bmatrix} = \begin{bmatrix} EB & 59 & 8B & 1B \\ 40 & 2E & A1 & C3 \\ F2 & 38 & 13 & 42 \\ 1E & 84 & E7 & D6 \end{bmatrix}$$

E.g. $47 \oplus AC = EB$

$40 \oplus 19 = 59$

• • •

$BC \oplus 6A = D6$

AES Encryption (7): The Initial AddRoundKey Transformation

- Example:
  - Key: 47 37 94 ED 40 D4 E4 A5 A3 70 3A A6 4C 9F 42 BC
  - Plaintext: AC 77 66 F3 19 FA DC 21 28 D1 29 41 57 5C 00 6A
  - The matrices are then XORed together:

$$\begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix} \oplus \begin{bmatrix} AC & 19 & 28 & 57 \\ 77 & FA & D1 & 5C \\ 66 & DC & 29 & 00 \\ F3 & 21 & 41 & 6A \end{bmatrix} = \begin{bmatrix} EB & 59 & 8B & 1B \\ 40 & 2E & A1 & C3 \\ F2 & 38 & 13 & 42 \\ 1E & 84 & E7 & D6 \end{bmatrix}$$

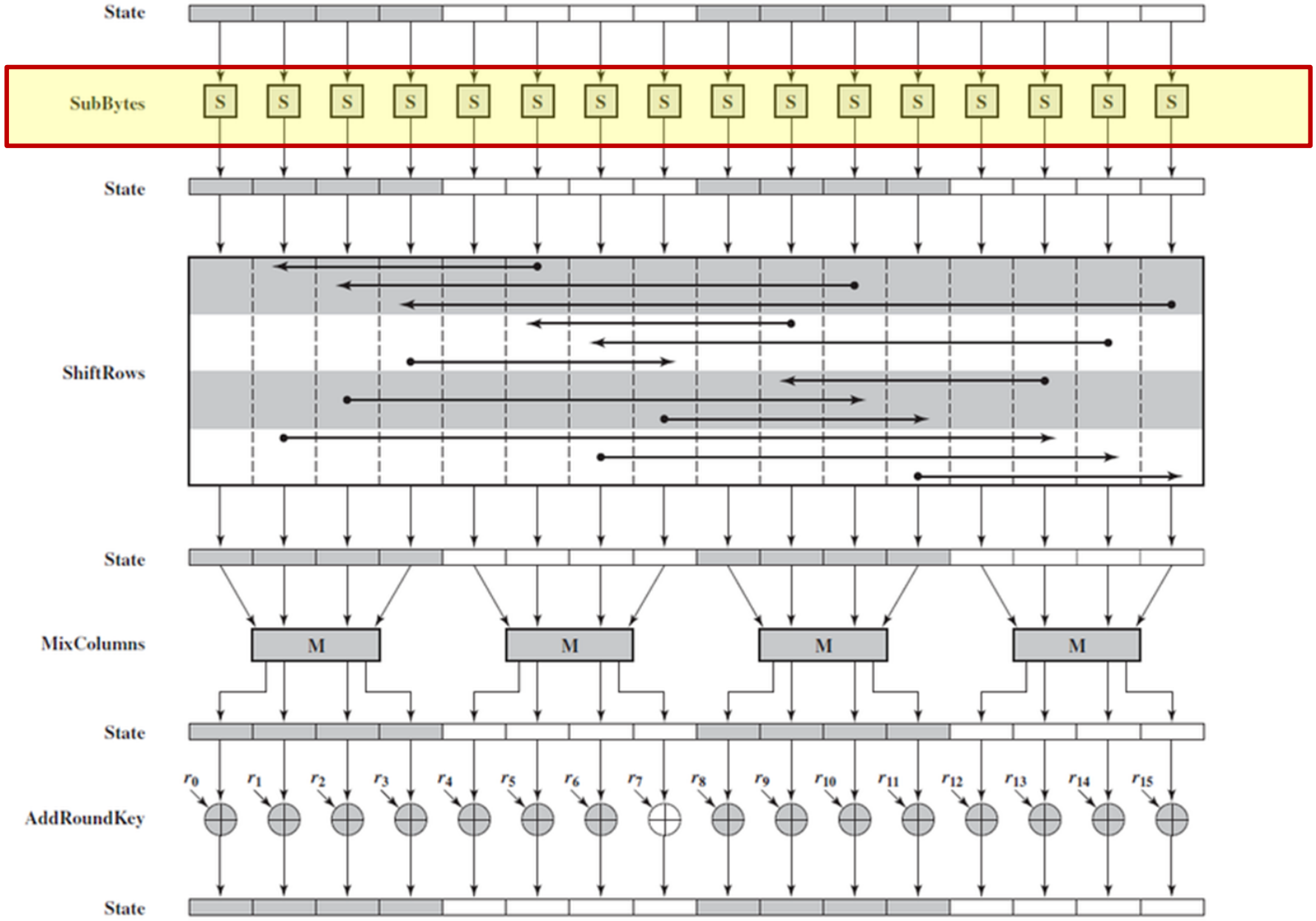
E.g. $47 \oplus AC = EB$

$40 \oplus 19 = 59$

...

$BC \oplus 6A = D6$

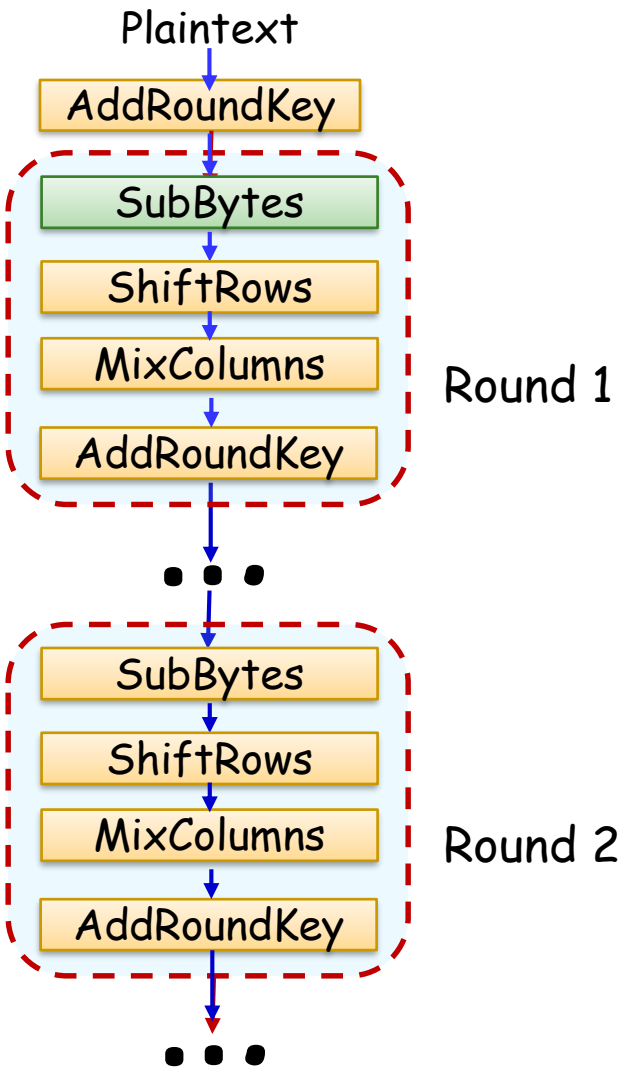
AES Encryption: The SubBytes Transformation (1)



AES Encryption: The SubBytes Transformation (2)

- Every byte of the state is replaced by a byte from a 16x16 S-Box, the Rijnadel S-Box.
- The S-Box contains a permutation of all possible 8-bit values.

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16



Process explained on the next slide.

AES Encryption: The SubBytes Transformation (3)

- Rationale: reduce correlation between the input bits and the output bits at the byte level.
  - Take a byte **a** from the state.
  - The first four bits of **a** represent the value of x in the row of the S-Box.
  - The second four bits of a represent the value of y in the column of the S-Box.
  - Replace **a** with byte **b** which lies at the intersection of the row and the column.
- Example on the next slide

AES Encryption: The SubBytes Transformation (4)

- Example: Consider state:

$$\begin{bmatrix} EA & 04 & 65 & 85 \\ 83 & 45 & 5D & 96 \\ 5C & 33 & 98 & B0 \\ F0 & 2D & AD & C5 \end{bmatrix}$$

		y														
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

AES Encryption: The SubBytes Transformation (5)

- Example: Consider state:
  - Take the first byte, **EA**

$$\begin{bmatrix} EA & 04 & 65 & 85 \\ 83 & 45 & 5D & 96 \\ 5C & 33 & 98 & B0 \\ F0 & 2D & AD & C5 \end{bmatrix}$$

		y														
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

AES Encryption : The SubBytes Transformation (6)

- Example: Consider state:

$$\begin{bmatrix} EA & 04 & 65 & 85 \\ 83 & 45 & 5D & 96 \\ 5C & 33 & 98 & B0 \\ F0 & 2D & AD & C5 \end{bmatrix}$$

  - Take the first byte, **EA**
    - The first four bits of EA are **E**.
    - The second four bits of EA are **A**.

		y														
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

AES Encryption: The SubBytes Transformation (7)

- Example: Consider state:

$$\begin{bmatrix} EA & 04 & 65 & 85 \\ 83 & 45 & 5D & 96 \\ 5C & 33 & 98 & B0 \\ F0 & 2D & AD & C5 \end{bmatrix}$$

  - Take the first byte, **EA**
    - The first four bits of EA are **E** ⇒ **the value of x**

- Example: Consider state:
  - Take the first byte, **EA**
    - The first four bits of EA are **E** ⇒ **the value of x**

AES Encryption: The SubBytes Transformation (8)

- Example: Consider state:

$$\begin{bmatrix} EA & 04 & 65 & 85 \\ 83 & 45 & 5D & 96 \\ 5C & 33 & 98 & B0 \\ F0 & 2D & AD & C5 \end{bmatrix}$$

  - Take the first byte, **EA**
    - The first four bits of EA are **E** ⇒ **the value of x**
    - The second four bits of EA are ***A*** ⇒ the value of y

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	**87**	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Replace EA with 87

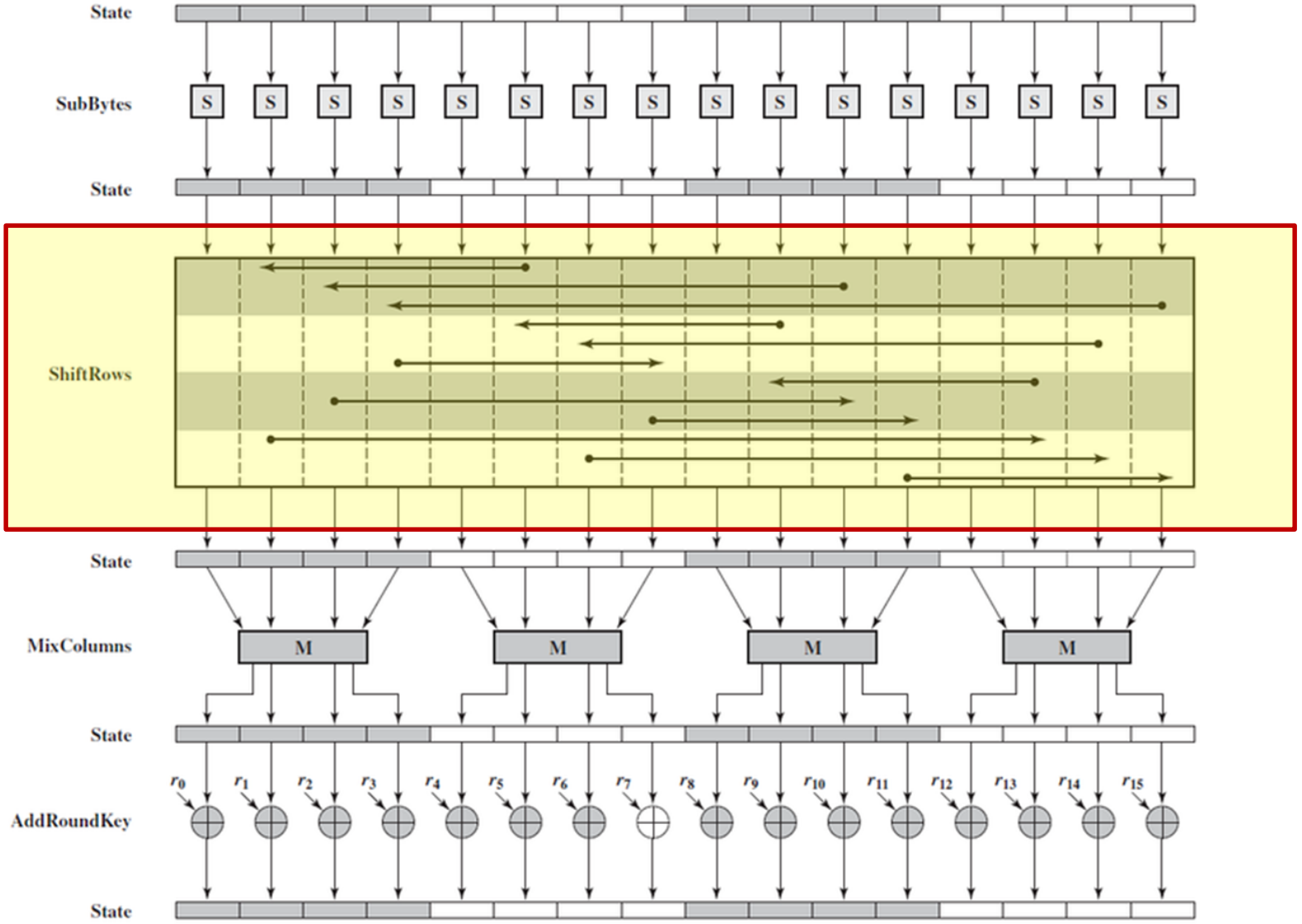
AES Encryption: The SubBytes Transformation (9)

- Example: Consider state: $\begin{bmatrix} EA & 04 & 65 & 85 \\ 83 & 45 & 5D & 96 \\ 5C & 33 & 98 & B0 \\ F0 & 2D & AD & C5 \end{bmatrix}$
  - Repeat the same process

  On all bytes of the state:

$$\begin{bmatrix} EA & 04 & 65 & 85 \\ 83 & 45 & 5D & 96 \\ 5C & 33 & 98 & B0 \\ F0 & 2D & AD & C5 \end{bmatrix} \Rightarrow \begin{bmatrix} 87 & F2 & 4D & 97 \\ EC & 6E & 4C & 90 \\ 4A & C3 & 46 & E7 \\ 8C & D8 & 95 & A6 \end{bmatrix}$$

AES Encryption: The ShiftRows Transformation (1)

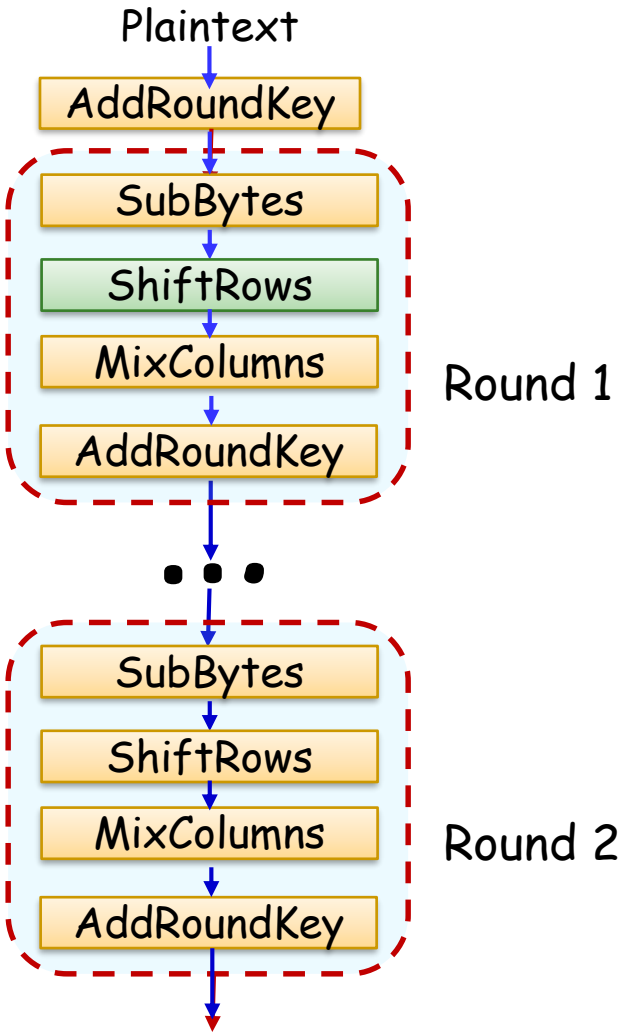


AES Encryption: The ShiftRows Transformation (2)

- Left circular shift the rows of the state:

Row Number	LCS Shift Amount
1	0
2	1
3	2
4	3

- Example on the next slide



AES Encryption: The ShiftRows Transformation (3)

- Example: consider state:

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ EC & 6E & 4C & 90 \\ 4A & C3 & 46 & E7 \\ 8C & D8 & 95 & A6 \end{bmatrix}$$

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ EC & 6E & 4C & 90 \\ 4A & C3 & 46 & E7 \\ 8C & D8 & 95 & A6 \end{bmatrix}$$

Row 1: Left circular shift by 0 bytes (i.e. do not shift)

Row Number	LCS Shift Amount
1	0
2	1
3	2
4	3

Result:

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

AES Encryption: The ShiftRows Transformation (4)

- Example: consider state:

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ EC & 6E & 4C & 90 \\ 4A & C3 & 46 & E7 \\ 8C & D8 & 95 & A6 \end{bmatrix}$$

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ EC & 6E & 4C & 90 \\ 4A & C3 & 46 & E7 \\ 8C & D8 & 95 & A6 \end{bmatrix}$$

Row 2: Left circular shift by 1 byte.

Row Number	LCS Shift Amount
1	0
2	1
3	2
4	3

Result:

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ & & & \\ & & & \end{bmatrix}$$

AES Encryption: The ShiftRows Transformation (5)

- Example: consider state:

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ EC & 6E & 4C & 90 \\ 4A & C3 & 46 & E7 \\ 8C & D8 & 95 & A6 \end{bmatrix}$$

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ EC & 6E & 4C & 90 \\ 4A & C3 & 46 & E7 \\ 8C & D8 & 95 & A6 \end{bmatrix}$$

Row 3: Left circular shift by 2 bytes).

Row Number	LCS Shift Amount
1	0
2	1
3	2
4	3

Result:

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ & & & \end{bmatrix}$$

AES Encryption: The ShiftRows Transformation (6)

- Example: consider state:

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ EC & 6E & 4C & 90 \\ 4A & C3 & 46 & E7 \\ 8C & D8 & 95 & A6 \end{bmatrix}$$

Row Number	LCS Shift Amount
1	0
2	1
3	2
4	3

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ EC & 6E & 4C & 90 \\ 4A & C3 & 46 & E7 \\ 8C & D8 & 95 & A6 \end{bmatrix}$$

Row 4: Left circular shift by 3 bytes.

Result:

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix}$$

AES Encryption: The ShiftRows Transformation (7)

- Example: consider state:

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ EC & 6E & 4C & 90 \\ 4A & C3 & 46 & E7 \\ 8C & D8 & 95 & A6 \end{bmatrix}$$

Row Number	LCS Shift Amount
1	0
2	1
3	2
4	3

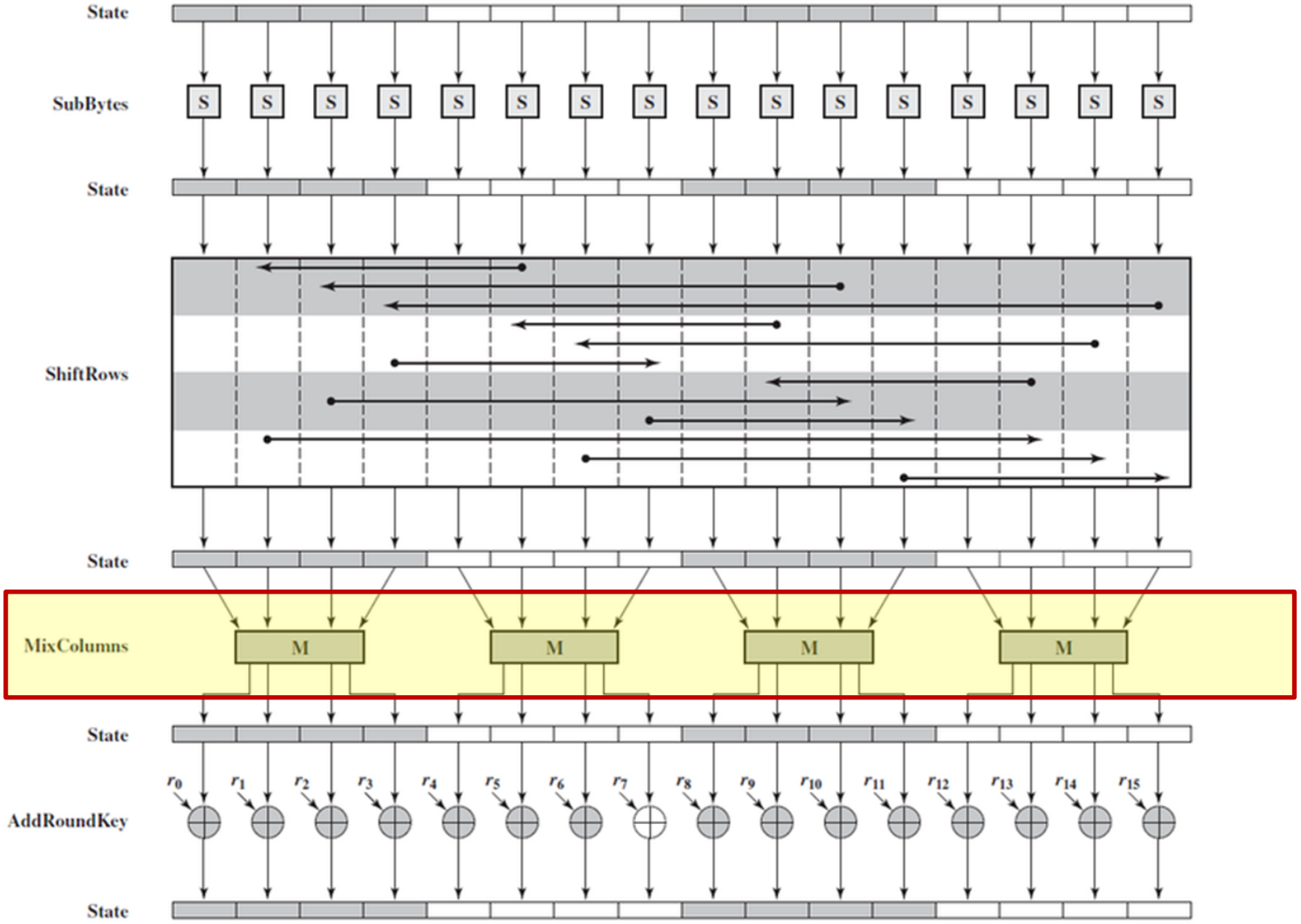
$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ EC & 6E & 4C & 90 \\ 4A & C3 & 46 & E7 \\ 8C & D8 & 95 & A6 \end{bmatrix}$$

Row 4: Left circular shift by 3 bytes.

Result:

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix}$$

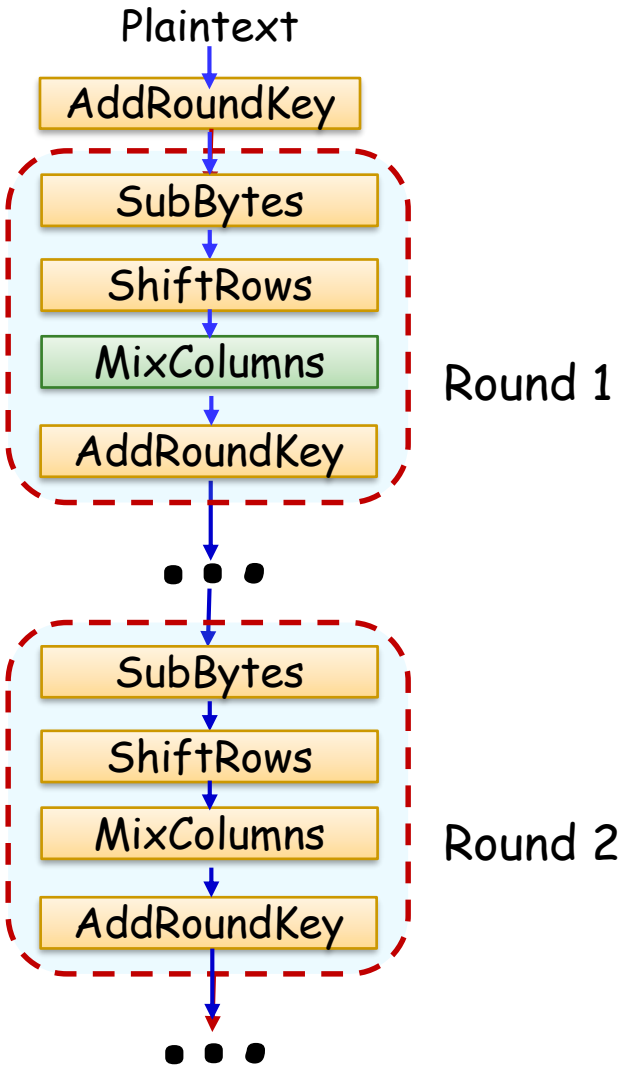
AES Encryption: The MixColumns Transformation (1)



AES Encryption: The MixColumns Transformation (2)

- Mixes the state columns.
- Based on Arithmetic in the finite field GF(2^8).
- Rationale: when performed after the ShiftRows operation ensures that each bit of the output depends on each bit of the input.
- The matrix $\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$ is multiplied by the state

All intermediate additions and multiplications of numbers is done using GF(2^8) arithmetic.



AES Encryption: The MixColumns Transformation (3)

- Review of matrix multiplication:
- Example: multiply the following matrices

1. $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 1 \times 7 + 2 \times 9 + 3 \times 11 & \end{bmatrix}$

2. $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 1 \times 7 + 2 \times 9 + 3 \times 11 & 1 \times 8 + 2 \times 10 + 3 \times 12 \end{bmatrix}$

3. $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 1 \times 7 + 2 \times 9 + 3 \times 11 & 1 \times 8 + 2 \times 10 + 3 \times 12 \\ 4 \times 7 + 5 \times 9 + 6 \times 11 & \end{bmatrix}$

4. $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 1 \times 7 + 2 \times 9 + 3 \times 11 & 1 \times 8 + 2 \times 10 + 3 \times 12 \\ 4 \times 7 + 5 \times 9 + 6 \times 11 & 4 \times 8 + 5 \times 10 + 6 \times 12 \end{bmatrix}$

AES Encryption: The MixColumns Transformation (4)

- How to multiply two matrices using finite field GF(2^8) arithmetic?
  - Same algorithm conventional matrix multiplication, BUT all intermediate additions and multiplications are done as follows:
    - Addition is replaced with XOR operation.
    - Multiplication is done by cross-referencing two tables:
      - E table and
      - L table
      - Cross-referencing is done to avoid directly dealing with complexities of field theory mathematics and to speed up implementation.
- Example on the next slide.

AES Encryption: The MixColumns Transformation (5)

- Example:
  - MixColumns multiples matrix $\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$ by the state using GF(2^8) arithmetic:

Consider state: $\begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix}$

What is $\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix}$?

AES Encryption: The MixColumns Transformation (6)

- Example:
  - Following the rules of conventional matrix multiplication, but replacing additions with XOR operations:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix} =$$

$$\begin{bmatrix} 2 \times 87 \oplus 3 \times 6E \oplus 1 \times 46 \oplus 1 \times A6 & \dots & \dots \\ & \vdots & \\ & & \end{bmatrix}$$

How do we multiply in GF(2^8) finite field arithmetic?

E.g. 2×87?

Next slide…

AES Encryption: The MixColumns Transformation (7)

- GF(2^8) multiplication algorithm (using AES lookup tables):
  - Assume: we want compute $\mathbf{b_1} * \mathbf{b_2}$ where b_1 and b_2 are bytes.
  - Split b_1 into two 4-bit halves.
    - The first half specifies the row index in the L-table (next slides).
    - The second half specifies the column index into the L-table.
    - Take the value b_1' which lies at the intersection of the row and column.
  - Repeat the same process on byte b_2 to obtain byte b_2'
  - Compute $b_1' + b_2'$. If the result is > FF, subtract FF from the result. Let **c** represent final result.

AES Encryption: The MixColumns Transformation (8)

- GF(2^8) multiplication algorithm (using AES lookup tables):
  - Split **c** into two 4-bit halves:
    - The first half specifies the row index in the E-table (next slides).
    - The second half specifies the column index into the E-table.
    - The value at the intersection of the row and column is our final answer.

AES Encryption: The MixColumns Transformation (9)

- Example: what is 02×87 in GF(2^8)?

- **Step 1:** Split 02 into two 4-bit halves:
  - The first 4-bit half is 0 ⇒ the row number in the L-table.
  - The second 4-bit half is 2 ⇒ the column number in the L-table.
  - Find the value at the intersection of row 0 and column 2 in the L-table (next slide).

AES Encryption: The MixColumns Transformation (10)

- **Step 1:** Split 02 into two 4-bit halves:
  - The first 4-bit half is 0 ⇒ the row number in the L-table.
  - The second 4-bit half is 2 ⇒ the column number in the L-table.

L-Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		00	19	01	32	02	1A	C6	4B	C7	1B	68	33	EE	DF	03
1	64	04	E0	0E	34	8D	81	EF	4C	71	08	C8	F8			
2	7D	C2	1D	B5	F9	B9	27	6A	4D	E4	A6	72	9A	C9	09	78
3	65	2F	8A	05	21	0F	E1	24	12	F0	82	45	35	93	DA	8E
4	96	8F	DB	BD	36	D0	CE	94	13	5C	D2	F1	40	46	83	38
5	66	DD	FD	30	BF	06	8B	62	B3	25	E2	98	22	88	91	10
6	7E	6E	48	C3	A3	B6	1E	42	3A	6B	28	54	FA	85	3D	BA
7	2B	79	0A	15	9B	9F	5E	CA	4E	D4	AC	E5	F3	73	A7	57
8	AF	58	A8	50	F4	EA	D6	74	4F	AE	E9	D5	E7	E6	AD	E8
9	2C	D7	75	7A	EB	16	0B	F5	59	CB	5F	B0	9C	A9	51	A0
A	7F	0C	F6	6F	17	C4	49	EC	D8	43	1F	2D	A4	76	7B	B7
B	CC	BB	3E	5A	FB	60	B1	86	3B	52	A1	6C	AA	55	29	9D
C	97	B2	87	90	61	BE	DC	FC	BC	95	CF	CD	37	3F	5B	D1
D	53	39	84	3C	41	A2	6D	47	14	2A	9E	5D	56	F2	D3	AB
E	44	11	92	D9	23	20	2E	89	B4	7C	B8	26	77	99	E3	A5
F	67	4A	ED	DE	C5	31	FE	18	0D	63	8C	80	C0	F7	70	07

19 is the result

AES Encryption: The MixColumns Transformation (11)

- **Step 2:** Split 87into two 4-bit halves:
  - The first 4-bit half is 8 ⇒ the row number in the L-table.
  - The second 4-bit half is 7 ⇒ the column number in the L-table.

L-Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		00	19	01	32	02	1A	C6	4B	C7	1B	68	33	EE	DF	03
1	64	04	E0	0E	34	8D	81	EF	4C	71	08	C8	F8	69	1C	C1
2	7D	C2	1D	B5	F9	B9	27	6A	4D	E4	A6	72	9A	C9	09	78
3	65	2F	8A	05	21	0F	E1	24	12	F0	82	45	35	93	DA	8E
4	96	8F	DB	BD	36	D0	CE	94	13	5C	D2	F1	40	46	83	
5	66	DD	FD	30	BF	06	8B	62	B3	25	E2	98	22		91	10
6	7E	6E	48	C3	A3	B6	1E	42	3A	6B	28		FA	85	3D	BA
7	2B	79	0A	15	9B	9F	5E	CA	4E	D4	AC	E5	F3	73	A7	57
8	AF	58	A8	50	F4	EA	D6	74	4F	AE	E9	D5	E7	E6	AD	E8
9	2C	D7	75	7A	EB	16	0B	F5	59	CB	5F	B0	9C	A9	51	A0
A	7F	0C	F6	6F	17	C4	49	EC	D8	43	1F	2D	A4	76	7B	B7
B	CC	BB	3E	5A	FB	60	B1	86	3B	52	A1	6C	AA	55	29	9D
C	97	B2	87	90	61	BE	DC	FC	BC	95	CF	CD	37	3F	5B	D1
D	53	39	84	3C	41	A2	6D	47	14	2A	9E	5D	56	F2	D3	AB
E	44	11	92	D9	23	20	2E	89	B4	7C	B8	26	77	99	E3	A5
F	67	4A	ED	DE	C5	31	FE	18	0D	63	8C	80	C0	F7	70	07

74 is the result

AES Encryption: The MixColumns Transformation (12)

- **Step 3:** Add the values from steps 1 and 2:
  - 19 + 74 = 8D; because 8D < FF, no need to subtract FF from the result.
- **Step 4:** Split the result from the previous step, 8D, into two 4-bit halves.
  - The first 4-bit half is 8 ⇒ the row index in the E-table (next slide)
  - The second 4-bit half is D ⇒ the column index in the E-table.
  - The value at the intersection of row and column is the result of multiplication.

AES Encryption: The MixColumns Transformation (13)

- **Step 4:** Split the result from the previous step, 8D, into two 4-bit halves.
  - The first 4-bit half is 8 ⇒ the row index in the E-table (next slide)
  - The second 4-bit half is D ⇒ the column index in the E-table.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	01	03	05	0F	11	33	55	FF	1A	2E	72	96	A1	F8	13	35
1	5F	E1	38	48	D8	73	95	A4	F7	02	06	0A	1E	22	66	AA
2	E5	34	5C	E4	37	59	EB	26	6A	BE	D9	70	90	AB	E6	31
3	53	F5	04	0C	14	3C	44	CC	4F	D1	68	B8	D3	6E	B2	CD
4	4C	D4	67	A9	E0	3B	4D	D7	62	A6	F1	08	18	28	78	88
5	83	9E	B9	D0	6B	BD	DC	7F	81	98	B3	CE	49	DB	76	9A
6	B5	C4	57	F9	10	30	50	F0	0B	1D	27	69	BB	D6	61	A3
7	FE	19	2B	7D	87	92	AD	EC	2F	71	93	AE	E9	20	60	A0
8	FB	16	3A	4E	D2	6D	B7	C2	5D	E7	32	56	FA	15	3F	41
9	C3	5E	E2	3D	47	C9	40	C0	5B	ED	2C	74	9C	BF	DA	75
A	9F	BA	D5	64	AC	EF	2A	7E	82	9D	BC	DF	7A	8E	89	80
B	9B	B6	C1	58	E8	23	65	AF	EA	25	6F	B1	C8	43	C5	54
C	FC	1F	21	63	A5	F4	07	09	1B	2D	77	99	B0	CB	46	CA
D	45	CF	4A	DE	79	8B	86	91	A8	E3	3E	42	C6	51	F3	0E
E	12	36	5A	EE	29	7B	8D	8C	8F	8A	85	94	A7	F2	0D	17
F	39	4B	DD	7C	84	97	A2	FD	1C	24	6C	B4	C7	52	F6	01

15 is the answer to 02×87 in $GF(2^8)$

AES Encryption: The MixColumns Transformation (14)

- Example:
  - Following the rules of conventional matrix multiplication, but replacing additions with XOR operations:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix} =$$

$$\begin{bmatrix} 2 \times 87 \oplus 3 \times 6E \oplus 1 \times 46 \oplus 1 \times A6 & \ldots & \ldots \\ \vdots & & \\ & & \end{bmatrix}$$

(15 ⇗ B2 ⇗ 46 ⇗ A6)

Use the same process to compute the products of: $3 \times 6E$, 1×46, and $1 \times A6$

$$3 \times 6E = 15$$
$$1 \times 46 = B2$$
$$1 \times A6 = A6$$

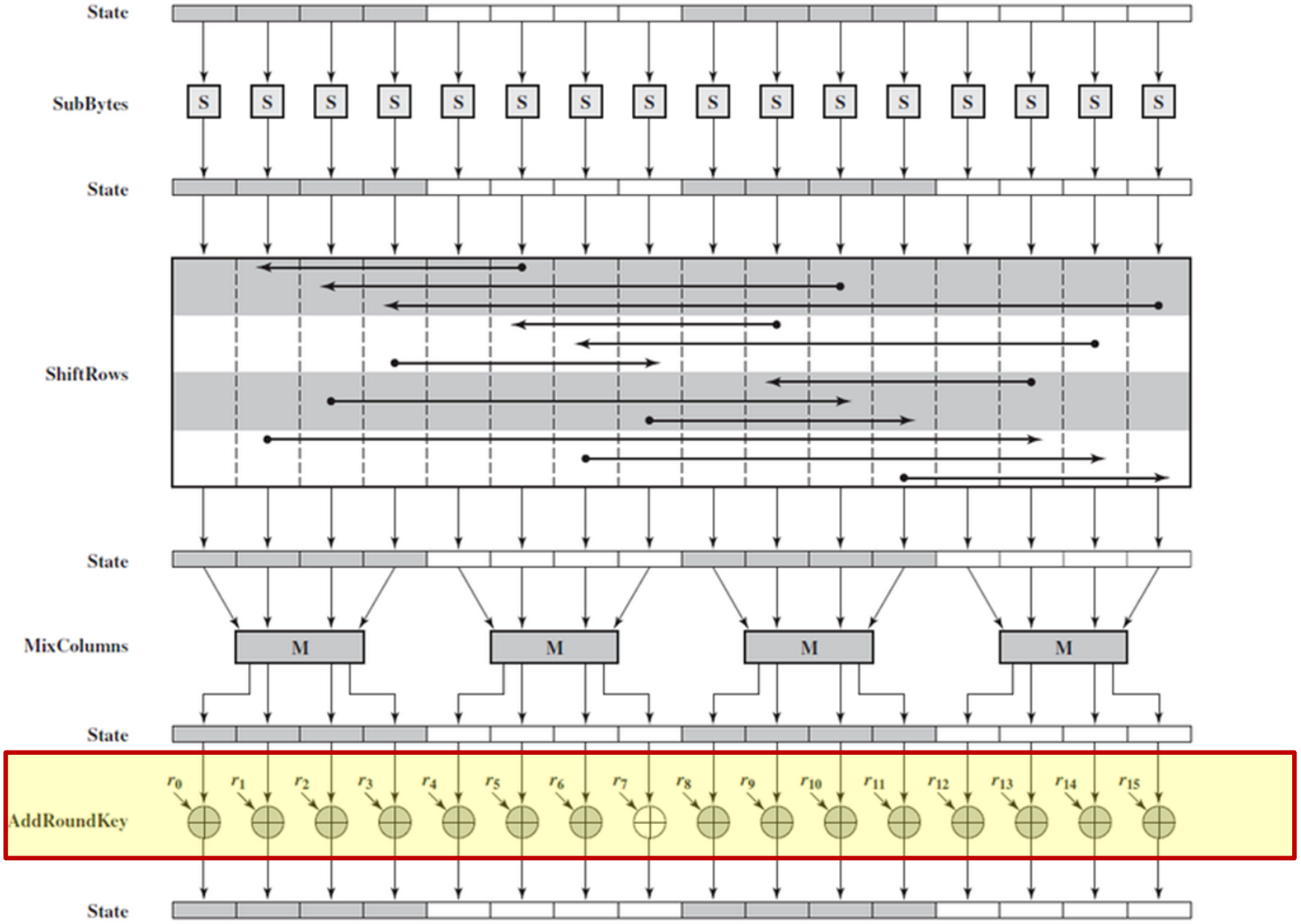
Answer: 15 ⊕ B2 ⊕ 46 ⊕ A6 = 47

AES Encryption: The MixColumns Transformation (15)

- Example:
  - Repeat the same process for the rest of the rows/columns.

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix} = \begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix}$$

AES Encryption: The AddRoundKey Transformation (1)

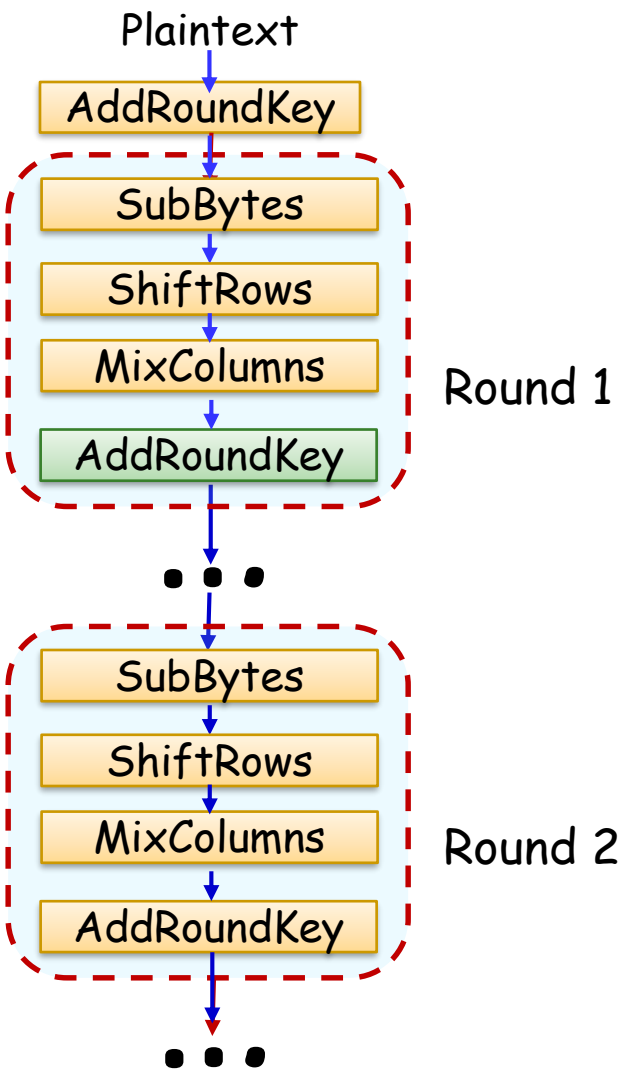


AES Encryption: The AddRoundKey Transformation (2)

- Rationale: a change in one bit of the encryption key should affect the round keys for several rounds.
- Uses the initial key to derive a round key and XORs the resulting key with the state.
- The 16-byte (i.e. 128-bit) key is converted into a vector of 4 sixteen byte elements.

$$\begin{bmatrix} K_0 & K_4 & K_8 & K_{12} \\ K_1 & K_5 & K_9 & K_{13} \\ K_2 & K_6 & K_{10} & K_{14} \\ K_3 & K_7 & K_{11} & K_{15} \end{bmatrix} \Rightarrow [W_0 \quad W_1 \quad W_2 \quad W_3]$$

Where $W_0 = K_0K_1K_2K_3$, $W_1 = K_4K_5K_6K_7$, etc..



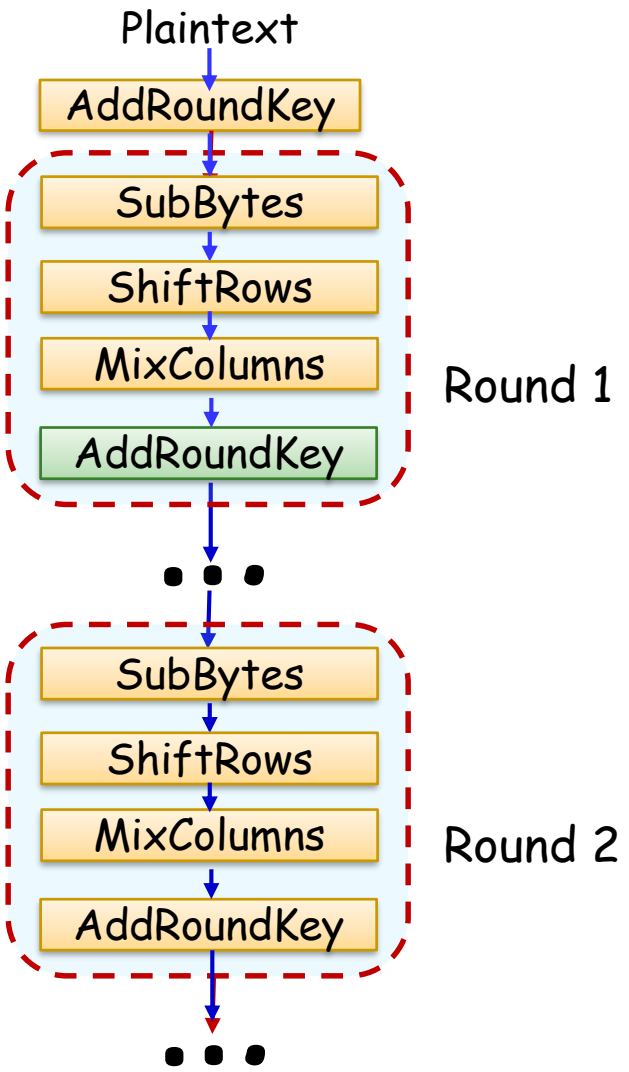
AES Encryption: The AddRoundKey Transformation (3)

- Vector $[W_0 \ W_1 \ W_2 \ W_3]$ is then expanded into a 44-byte vector

  $[W_0 \ W_1 \ W_2 \ W_3 \ \ldots \ W_{43}]$

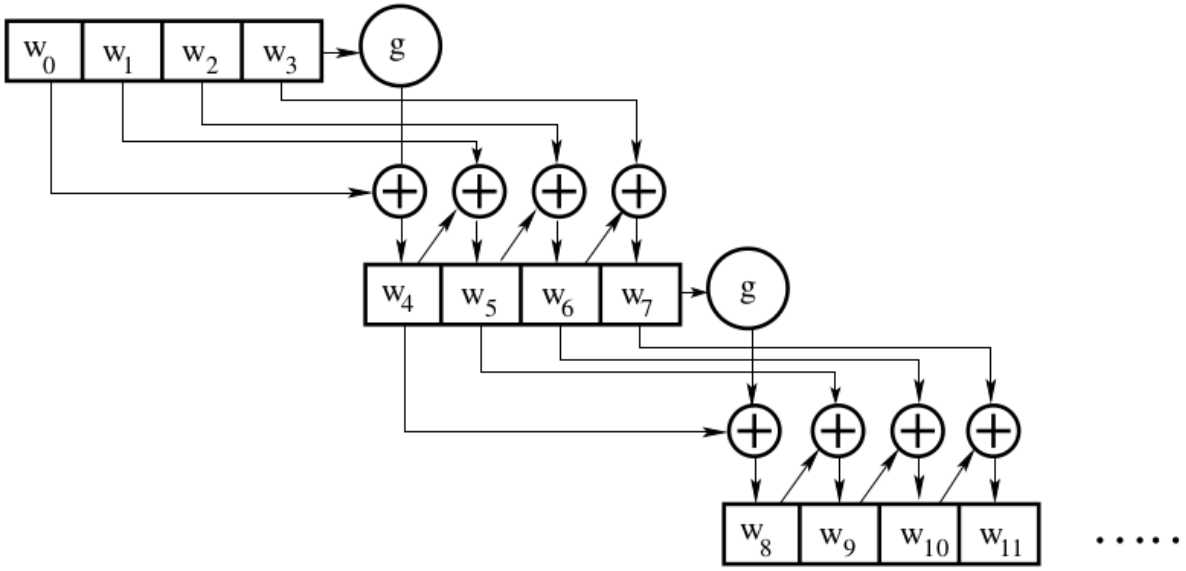
- $W_0 \ldots W_3$ represent the key for the first round.

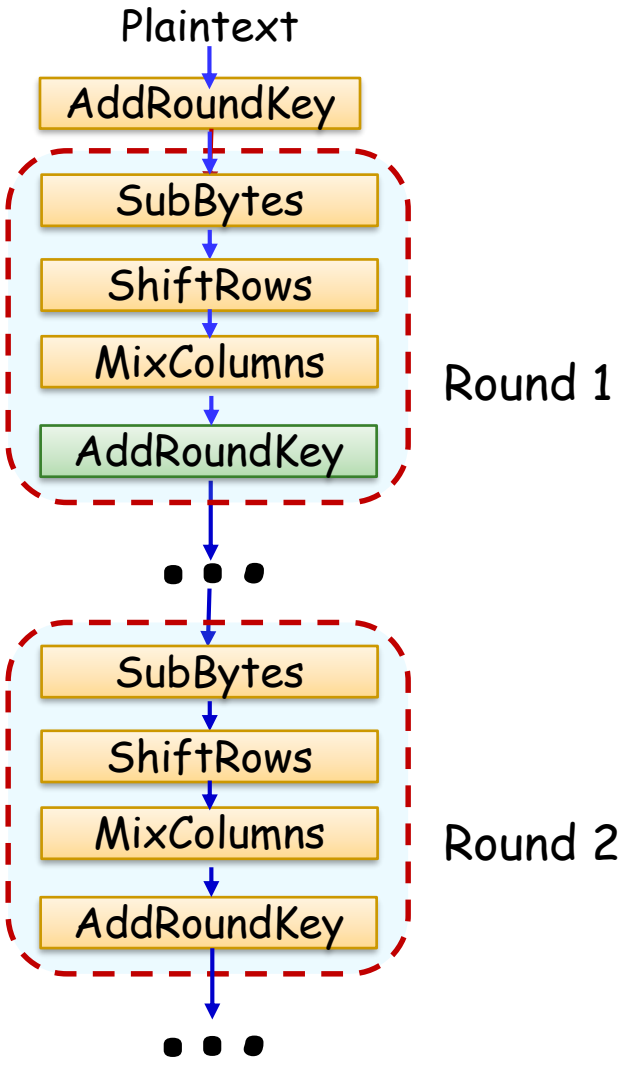
- $W_4 \ldots W_7$ represent the key for the second round.



AES Encryption: The AddRoundKey Transformation (4)

- Expanding vector $[W_0 \ W_1 \ W_2 \ W_3]$ to then expanded into a 44-byte vector $[W_0 \ W_1 \ W_2 \ W_3 \ \ldots \ W_{43}]$:

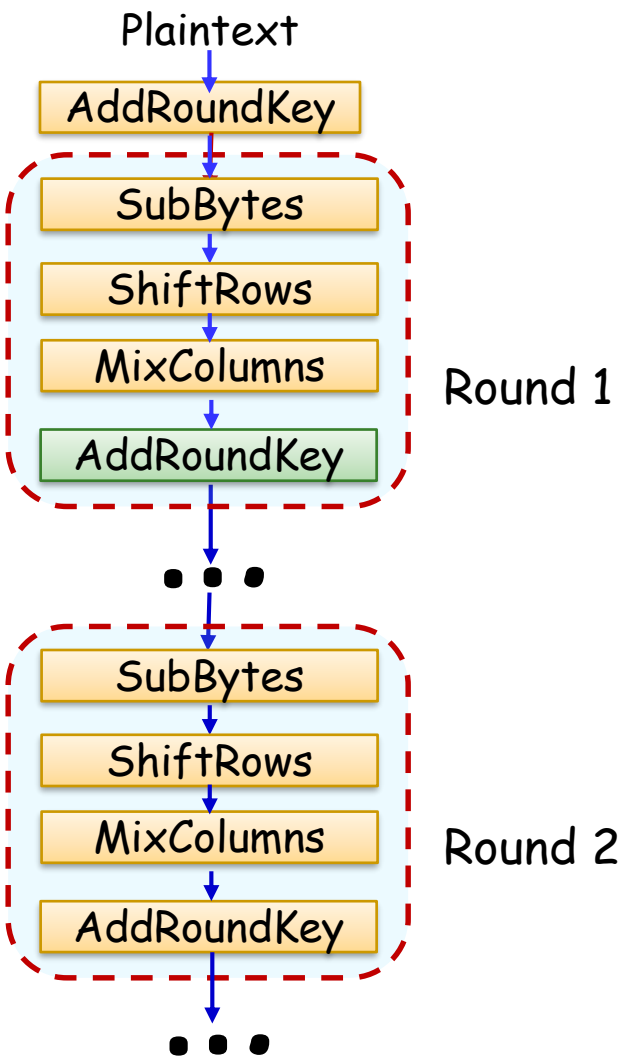




- Function **g** is defined in the next slide.

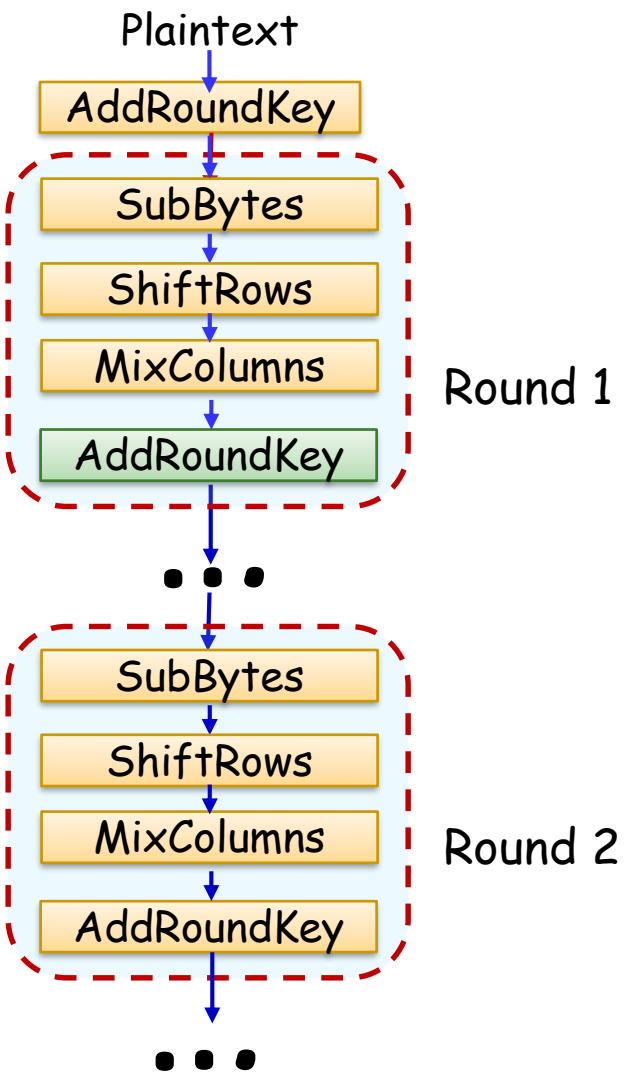
AES Encryption: The AddRoundKey Transformation (5)

- Expanding vector $[W_0 \ W_1 \ W_2 \ W_3]$ to then expanded into a 44-byte vector $[W_0 \ W_1 \ W_2 \ W_3 \ \ldots \ W_{43}]$:
- Function **g** is defined in the next slide g(W) where W is a 4-byte word, works as follows:
  - Step 1: Perform a one-byte left circular rotation on W.
  - Step 2:
    - Perform the SubBytes step on the matrix.



AES Encryption: The AddRoundKey Transformation (6)

- Expanding vector $[W_0 \ W_1 \ W_2 \ W_3]$ to then expanded into a 44-byte vector $[W_0 \ W_1 \ W_2 \ W_3 \ \ldots \ W_{43}]$:
- Function **g** is defined in the next slide g(W) where W is a 4-byte word, works as follows:
  - Step 3: XOR the vector from step 2 with the round constant – a vector of four 4-byte words; different for each round.
  - Next slide gives the table for each round.

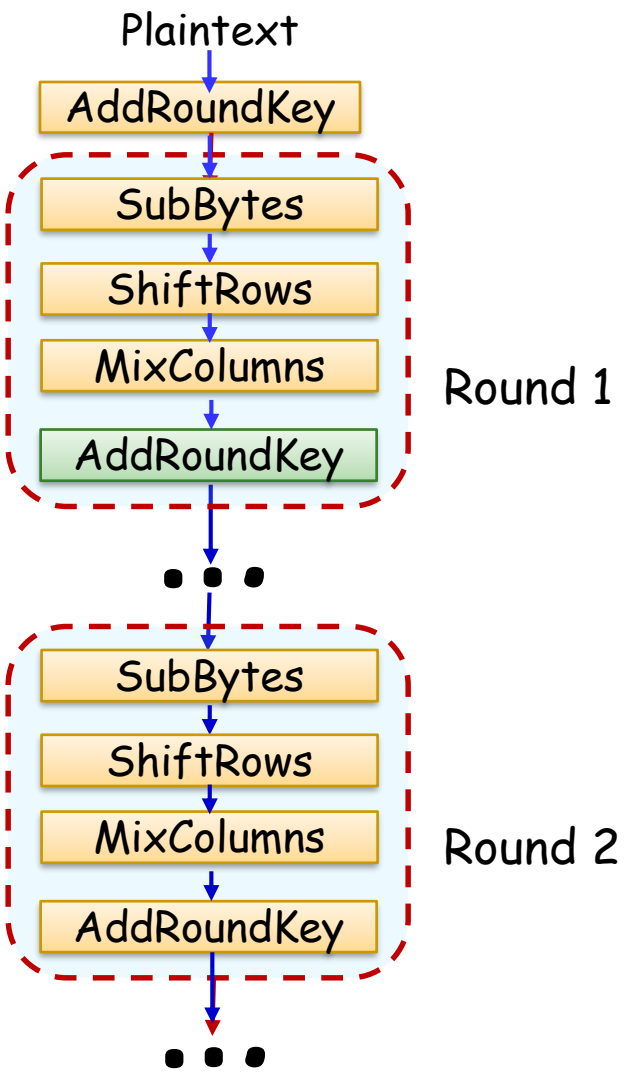


AES Encryption: The AddRoundKey Transformation (7)

Round	Round Constant
1	[01, 00, 00, 00]
2	[02, 00, 00, 00]
3	[04, 00, 00, 00]
4	[08, 00, 00, 00]
5	[10, 00, 00, 00]
6	[20, 00, 00, 00]
7	[40, 00, 00, 00]
8	[80, 00, 00, 00]
9	[1B, 00, 00, 00]
10	[36, 00, 00, 00]

AES Encryption: The AddRoundKey Transformation (8)

- Expanding vector $[W_0 \ W_1 \ W_2 \ W_3]$ to then expanded into a 44-byte vector $[W_0 \ W_1 \ W_2 \ W_3 \ \ldots \ W_{43}]$:
- Function **g** is defined in the next slide g(W) where W is a 4-byte word, works as follows:
  - Step 4: XOR the result from step 2 with the round key.



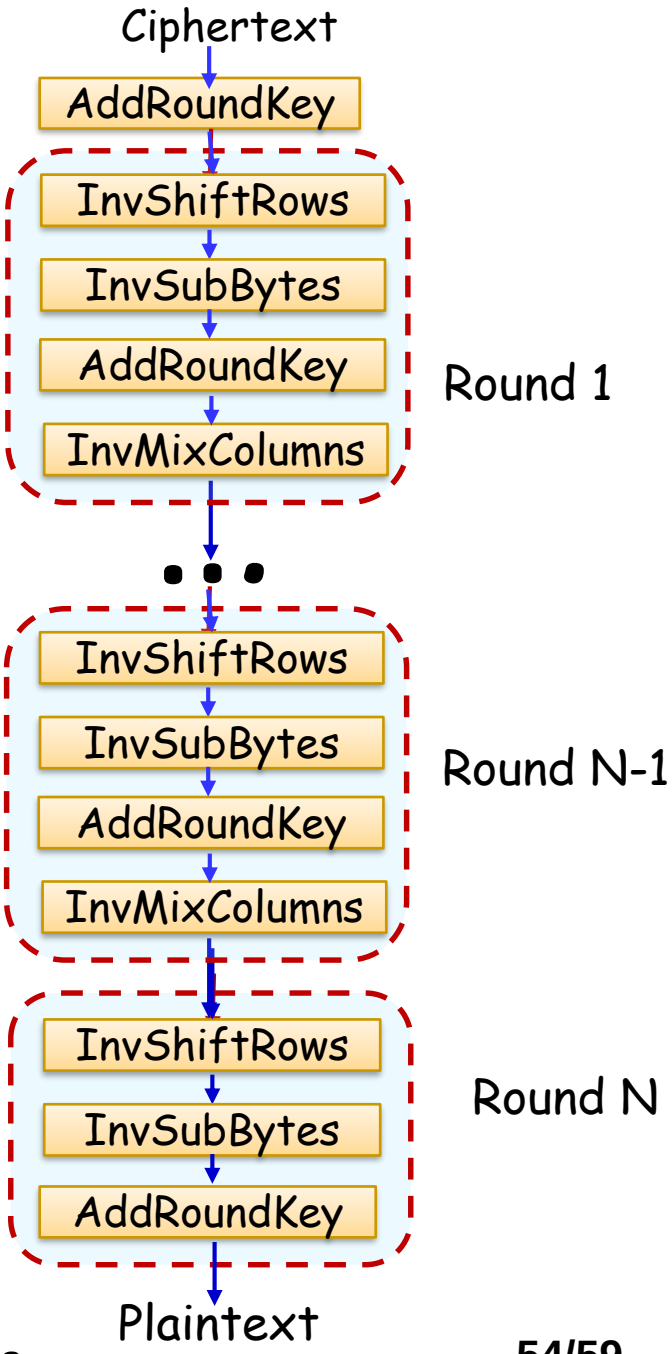
AES Encryption: The AddRoundKey Transformation (8)

- Example: Expansion of AES key 0f 15 71 c9 47 d9 e8 59 0c b7 ad d6 af 7f 67 98 for the first 4 rounds
  - RotWord(w): Left circular shift w by one byte
  - SubWord(x): apply SubBytes transformation to x
  - Rcon(i): the round constant for round i

Key Words	Auxiliary Function
w0 = 0f 15 71 c9 w1 = 47 d9 e8 59 w2 = 0c b7 ad w3 = af 7f 67 98	RotWord(w3) = 7f 67 98 af = x1 SubWord(x1) = d2 85 46 79 = y1 Rcon(1) = 01 00 00 00 y1 ⊕ Rcon(1) = d3 85 46 79 = z1
w4 = w0 ⊕ z1 = dc 90 37 b0 w5 = w4 ⊕ w1 = 9b 49 df e9 w6 = w5 ⊕ w2 = 97 fe 72 3f w7 = w6 ⊕ w3 = 38 81 15 a7	RotWord(w7) = 81 15 a7 38 = x2 SubWord(x4) = 0c 59 5c 07 = y2 Rcon(2) = 02 00 00 00 y2 ⊕ Rcon(2) = 0e 59 5c 07 = z2
w8 = w4 ⊕ z2 = d2 c9 6b b7 w9 = w8 ⊕ w5 = 49 80 b4 5e w10 = w9 ⊕ w6 = de 7e c6 61 w11 = w10 ⊕ w7 = e6 ff d3 c6	RotWord(w11) = ff d3 c6 e6 = x3 SubWord(x2) = 16 66 b4 83 = y3 Rcon(3) = 04 00 00 00 y3 ⊕ Rcon(3) = 12 66 b4 8e = z3
w12 = w8 ⊕ z3 = c0 af df 39 w13 = w12 ⊕ w9 = 89 2f 6b 67 w14 = w13 ⊕ w10 = 57 51 ad 06 w15 = w14 ⊕ w11 = b1 ae 7e c0	RotWord(w15) = ae 7e c0 b1 = x4 SubWord(x3) = e4 f3 ba c8 = y4 Rcon(4) = 08 00 00 00 y4 ⊕ Rcon(4) = ec f3 ba c8 = 4

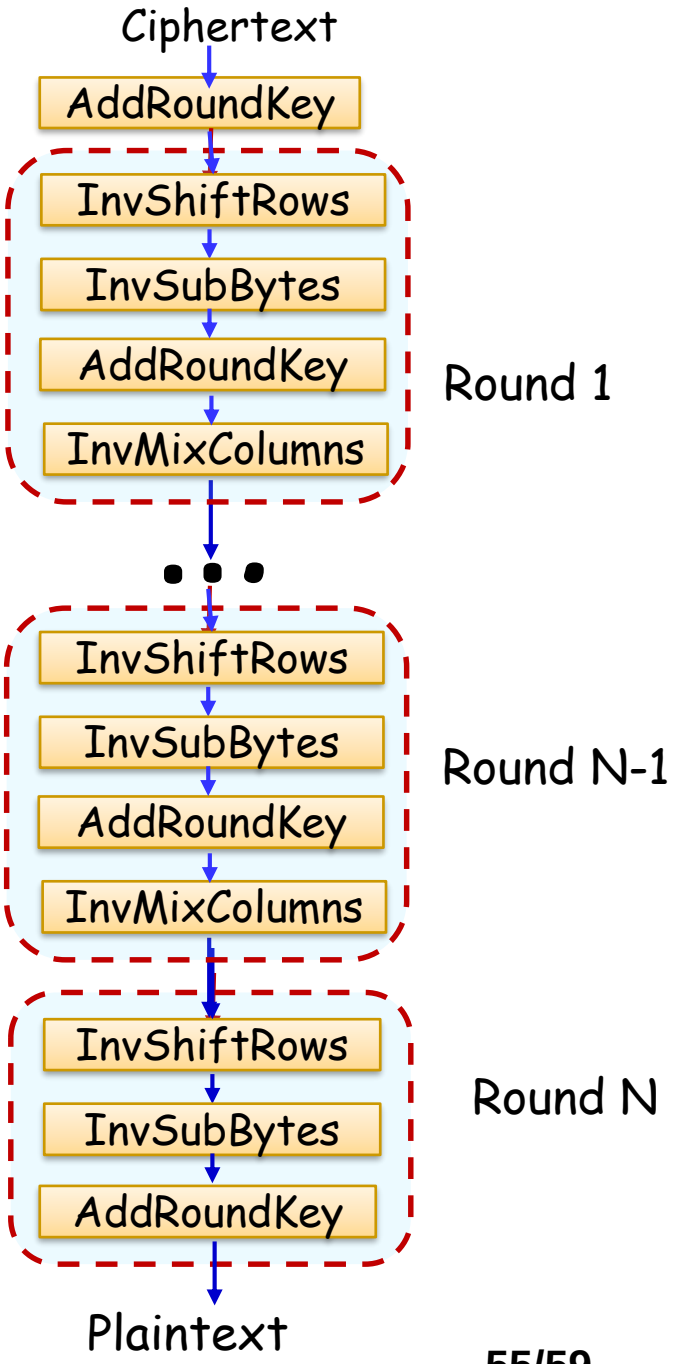
AES Decryption (1)

- Unlike DES decryption, AES decryption differs significantly from encryption.
  - The order of steps is different:
    - Encryption:
      - SubBytes
      - ShiftRows
      - MixColumns
      - AddRoundKey
    - Decryption (each operation except AddRoundKey is an inverse of its encryption counterpart):
      - InvShiftRows
      - InvSubBytes
      - AddRoundKey
      - InvMixColumns



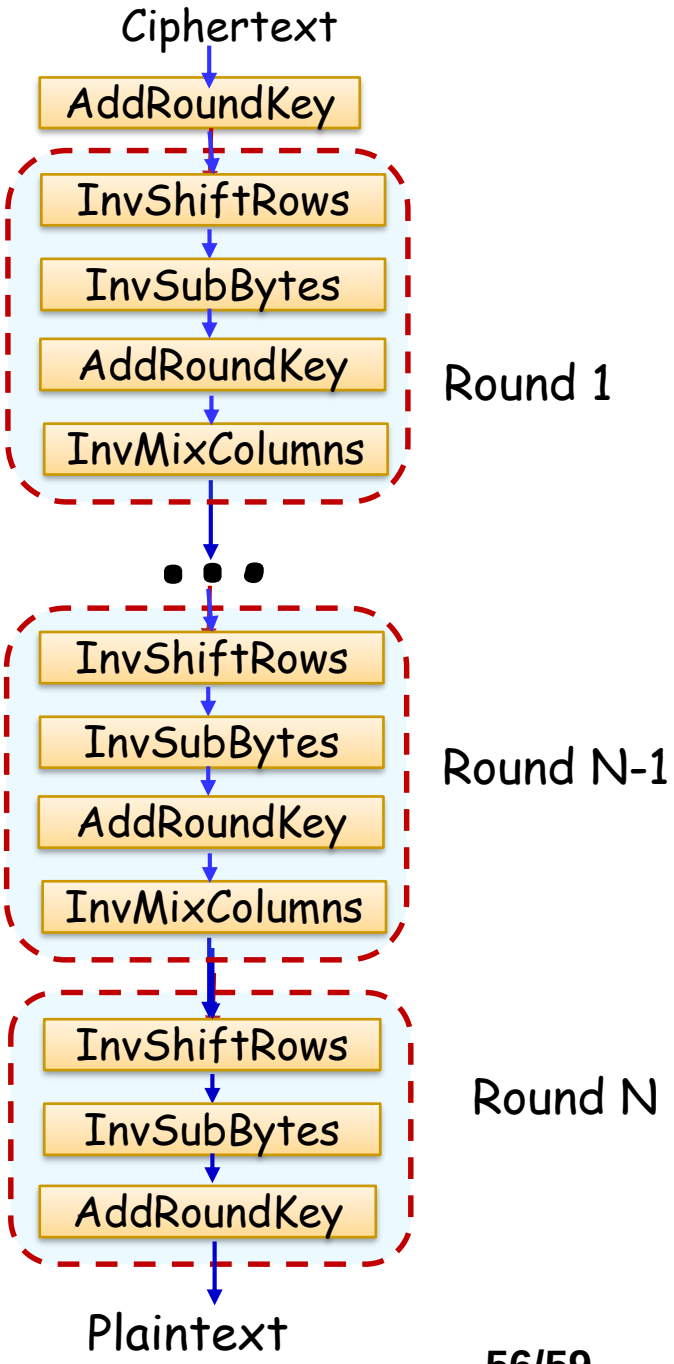
AES Decryption (2)

- Unlike DES decryption, AES decryption differs significantly from encryption
- This is because unlike DES, AES is not based on the Feistal Cipher architecture, but on the Substitution Permutation Network (SPN)



AES Decryption (3)

- SPNs require that all substitutions and permutations are invertible
  - I.e., for each substitution/permutation operation there is an operation that can reverse the effects of that substitution or permutation
- Feistal cipher does not have this requirement, and instead reverses the order of keys



Attacks on AES

- No successful, practical attacks to date.
- Approaches were developed for executing timing attacks against vulnerable AES implementations on some systems.
- Documents revealed by Edward Snowden showed that NSA is investigating whether tau statistic can be used to successfully break AES.

AES in Real World Applications

- AES is currently used in:
  - WPA-2 wireless security protocol.
  - HTTPS and SSL security protocols.
  - BitLocker full disk encryption utility.
  - BitCoin system (used for encrypting the wallet.dat file)
  - MANY others…

References

- *Advanced Encryption Standard by Example.* www.adamberent.com/documents/aesbyexample.pdf

- *AES: The Advanced Encryption Standard Lecture Notes on "Computer and Network Security".* https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture8.pdf

- William Stallings. *Cryptography and Network Security*, 6th Edition.

- Michael Goodrich and Roberto Tamssia. *Introduction to Computer Security*, 1st Edition.