



IN410
Cryptography and Secure Communications

Ahmad Fadlallah

Credits



- The following slides are the adaptation of
  - Lecture slides of Lawrie Brown based on William Stallings book “Cryptography and Network Security: Principles and Practice”
  - Lecture Slides of Dan Boneh – Stanford University
  - Lecture slides of Pawel Wocjan – University of Central California
  - Lecture Slides of Ahmed Serhrouchni – Telecom ParisTech
  - Others (References in the note section)

Course Syllabus



- Context
- History/ Classical ciphers
- **Symmetric Encryption ←**
- Asymmetric Encryption
- Cryptographic hash functions
- Key Distribution
- Security Protocols



Symmetric Encryption

- **Block Ciphers ←**
- Operation Modes
- Stream Ciphers



Advanced Encryption Standard (AES)

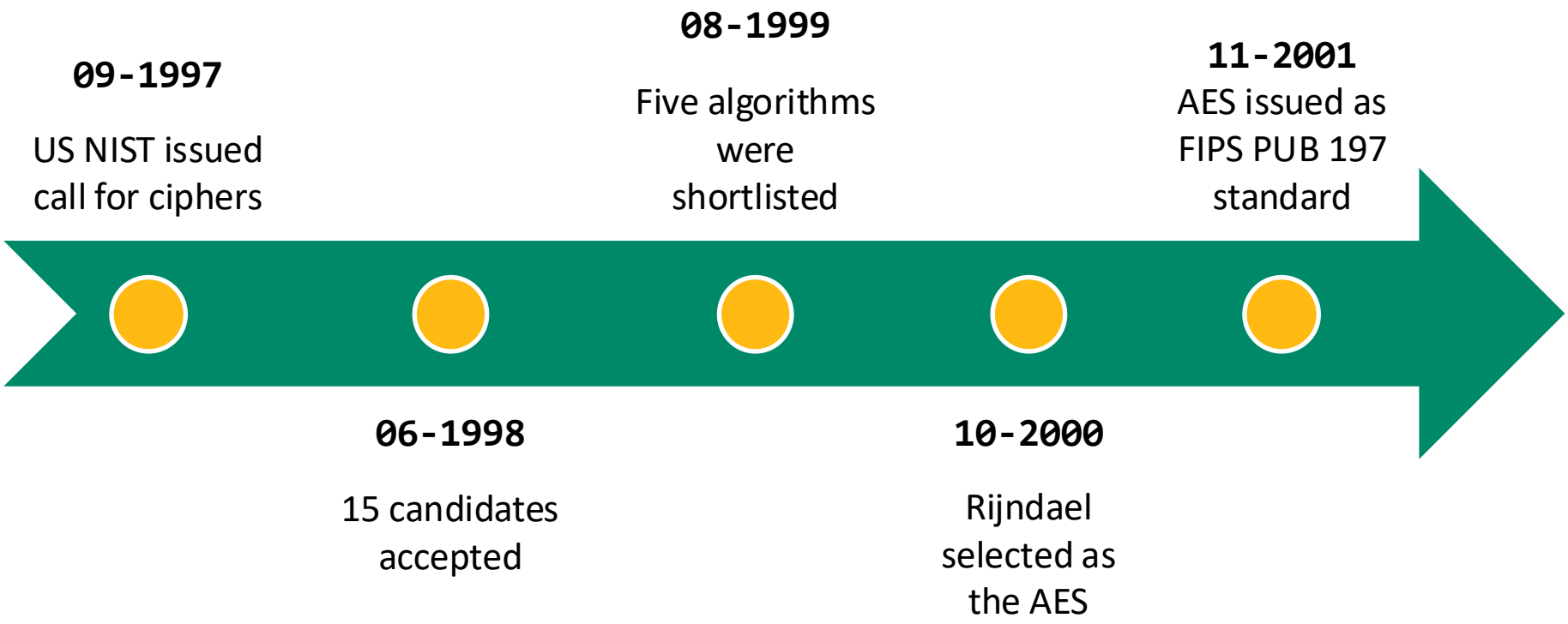
Origins



- Data Encryption Standard (DES)
  - has theoretical attacks that can break it
  - has demonstrated exhaustive key search attacks
- Triple-DES can be used, But
  - Slow and has small blocks

A replacement for DES was needed

AES Timeline



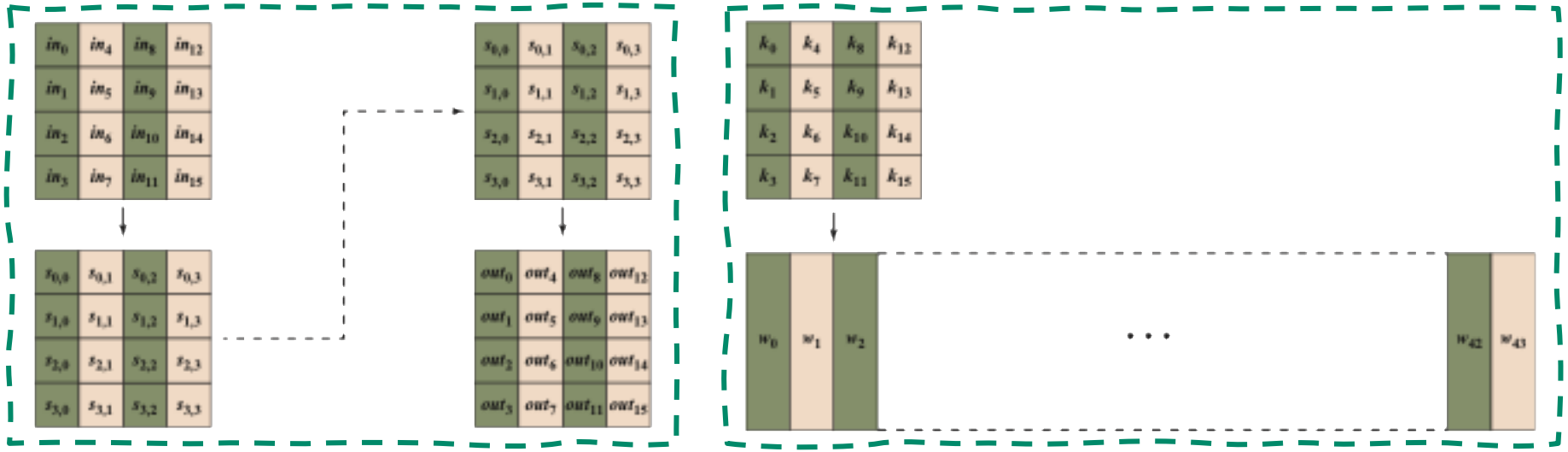
The AES Cipher - Rijndael



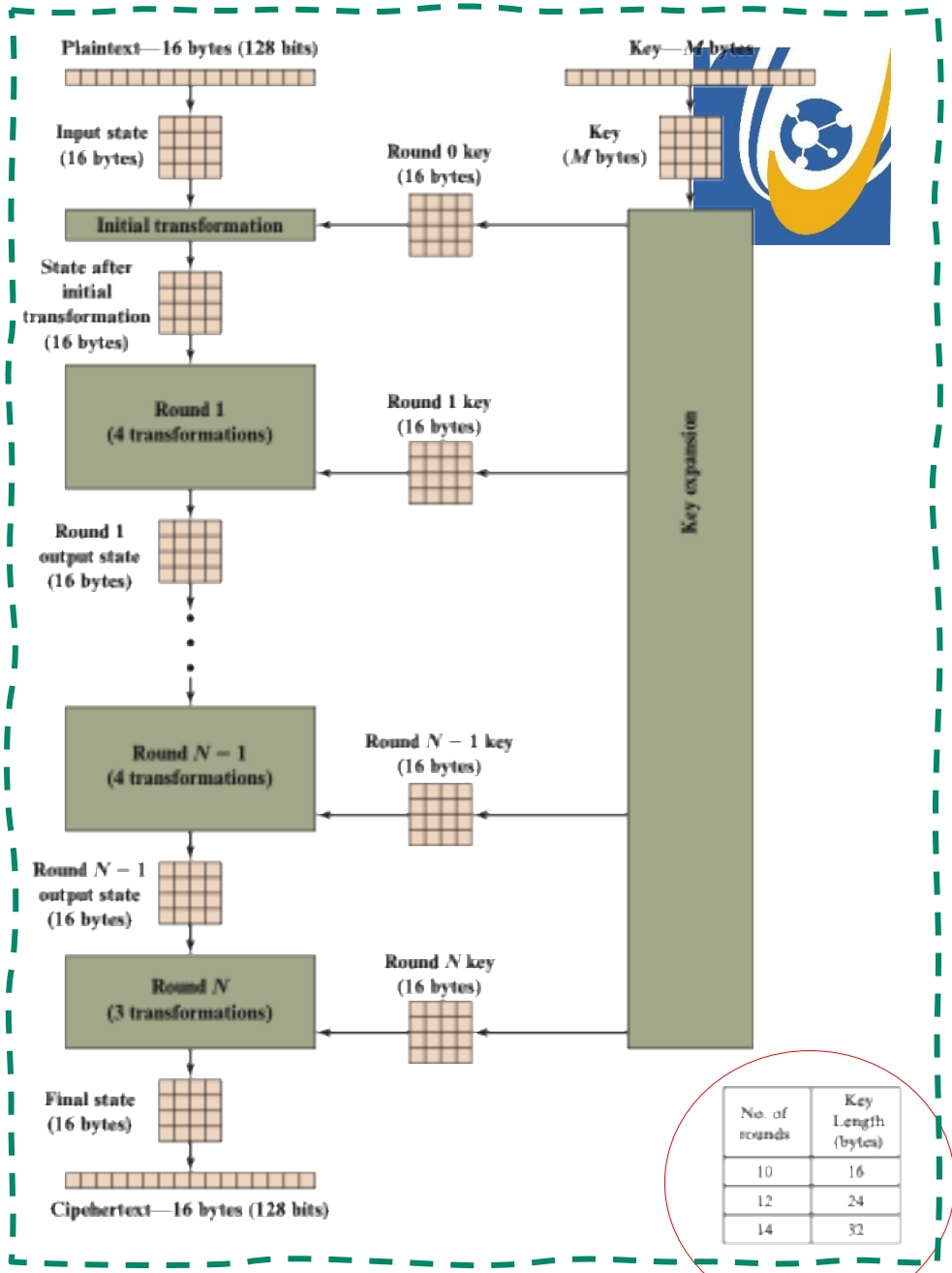
- Designed by two Belgian cryptographers Joan Daemen and Vincent Rijmen
- **Key lengths**: 128/192/256 bits
  - AES-128, AES-192 and AES-256
- **Block size**: 128 bits
- **Iterative** rather than Feistel cipher
  - Processes data as block 4x4 matrix of bytes
  - Operates on entire data block in every round

AES Structure

- **Input block size** is of 16 bytes, **arranged as a 4*4 square matrix of bytes.**
- This block is copied into the **State** array, which is modified at each stage of encryption.
- The **State** array is copied to an output matrix after the final stage.
- Keys generated at each stage are also arranged into a 4*4 square matrices



AES Structure

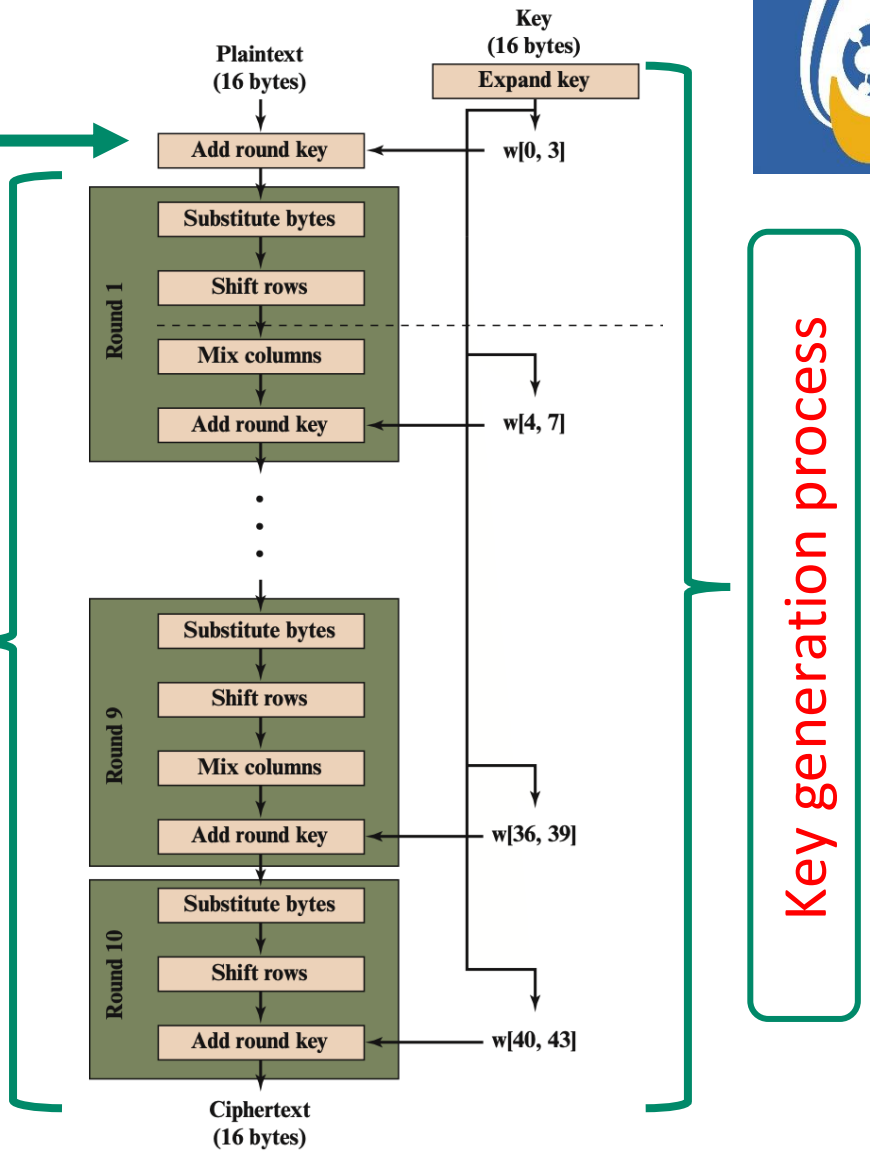


No. of rounds	Key Length (bytes)
10	16
12	24
14	32

AES Detailed Structure

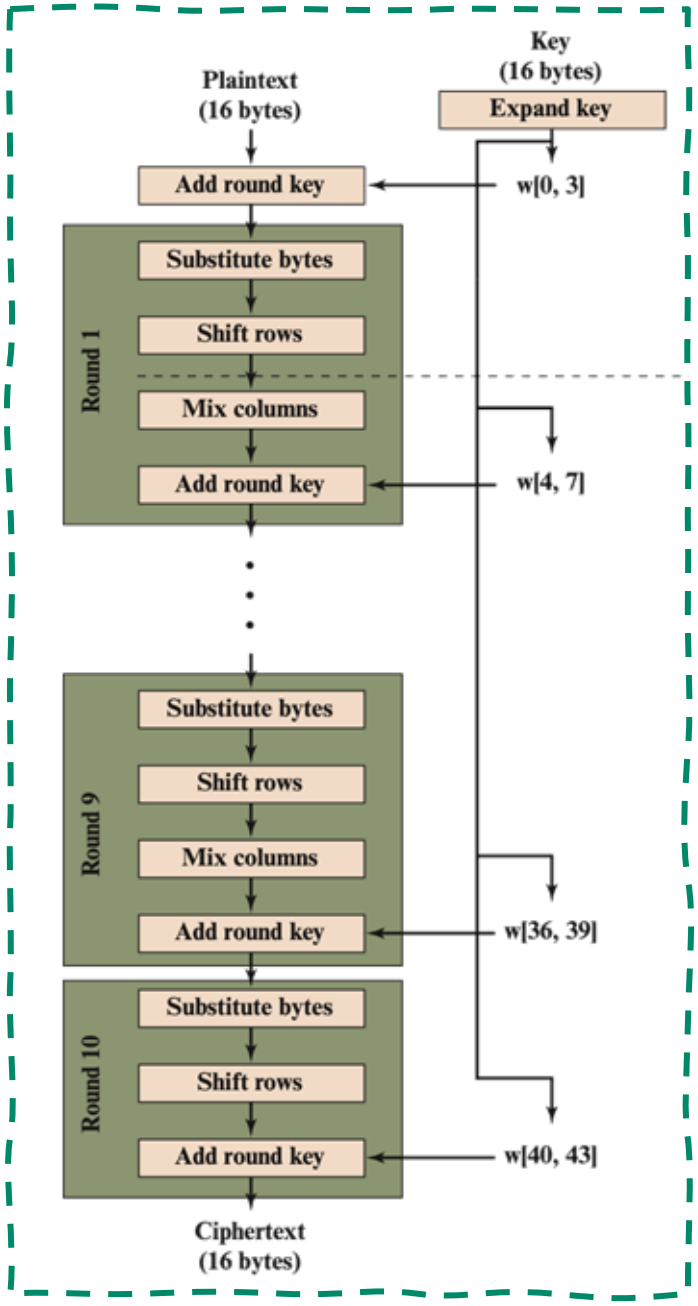
Initial Transformation: Simple *bitwise XOR* with **Round-0 key**

Rounds processing



Round Processing

- Four transformations
  - Substitute Bytes (SubBytes)
  - Shift Rows (ShiftRows)
  - Mix columns (MixColumns)
  - Add Round Key (AddRoundKey)
- *First (N-1) Rounds* include the four transformations
- *Last round (N)* includes only three transformations
- **ShiftRows is the only permutation**, the rest are substitutions

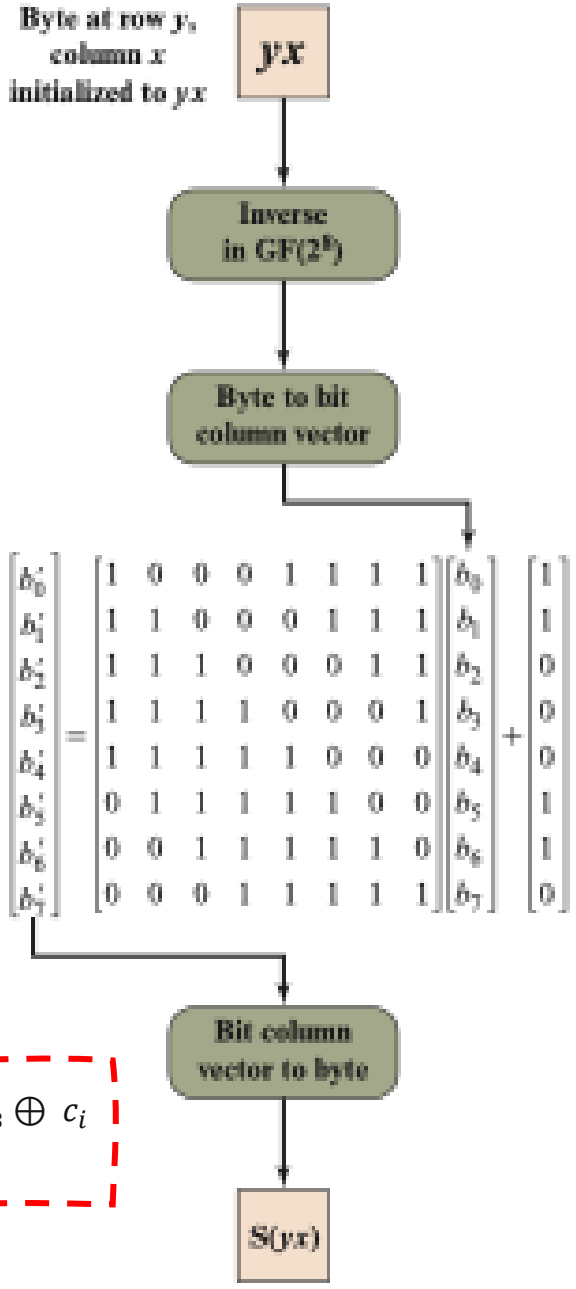


Substitute Bytes



- 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)
- **Inverse S-Box** For decryption
- S-box constructed using defined transformation of values in GF(2^8) with the prime polynomial ($x^8+x^4+x^3+x+1$)
- Designed to be resistant to all known attacks

Substitute Bytes



$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$b'_i = b_i \oplus b_{(i+4) mod\ 8} \oplus b_{(i+5) mod\ 8} \oplus b_{(i+6) mod\ 8} \oplus b_{(i+7) mod\ 8} \oplus c_i$$

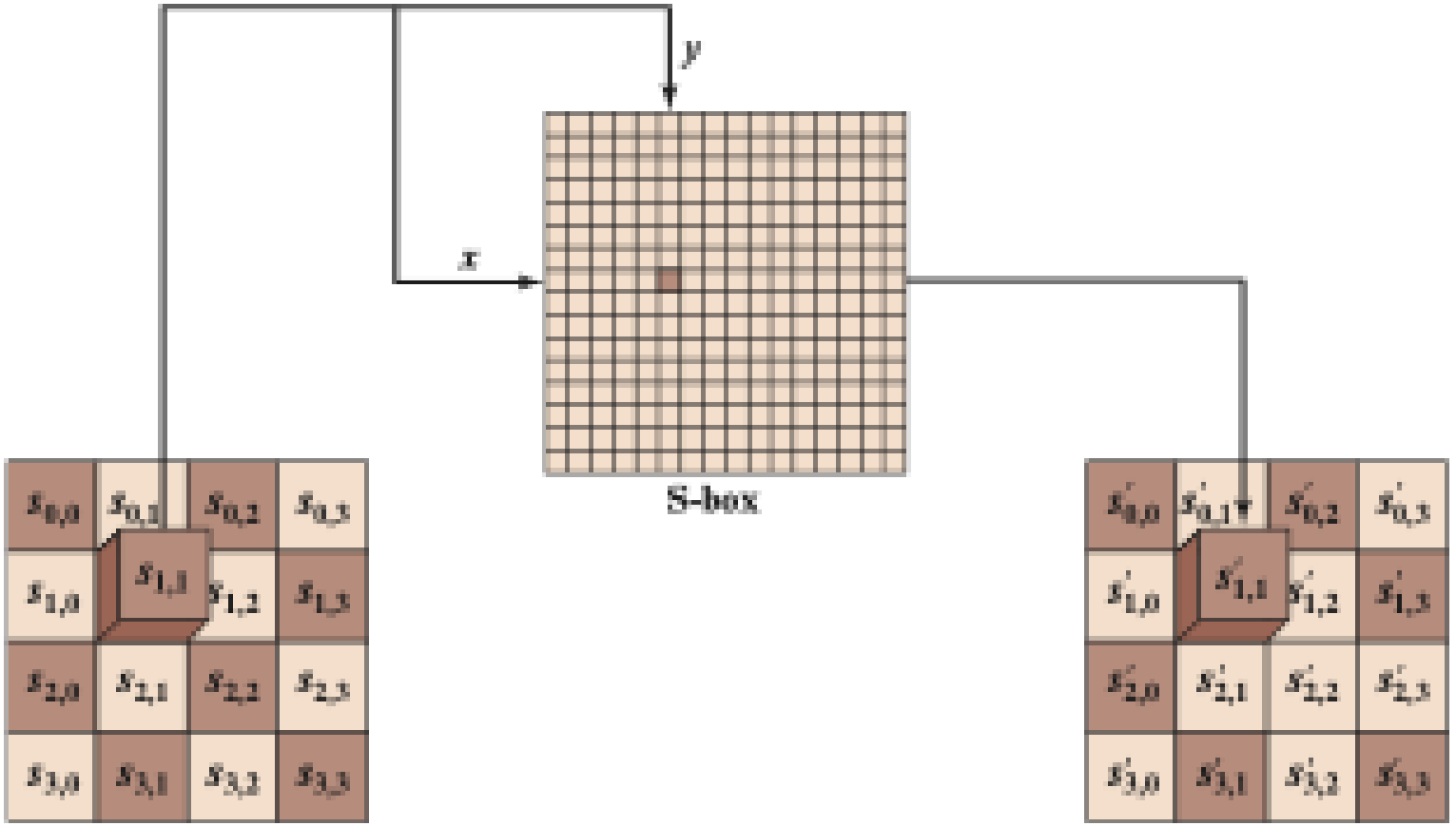
$c_i : i^{th}$ bit of byte c with the value $\{63\} = \{01100011\}$

0x95

0x8A

0x2A

Substitute Bytes



AES S-Box



Example: byte {95} is replaced by byte in row 9 column 5 which has value {2A}

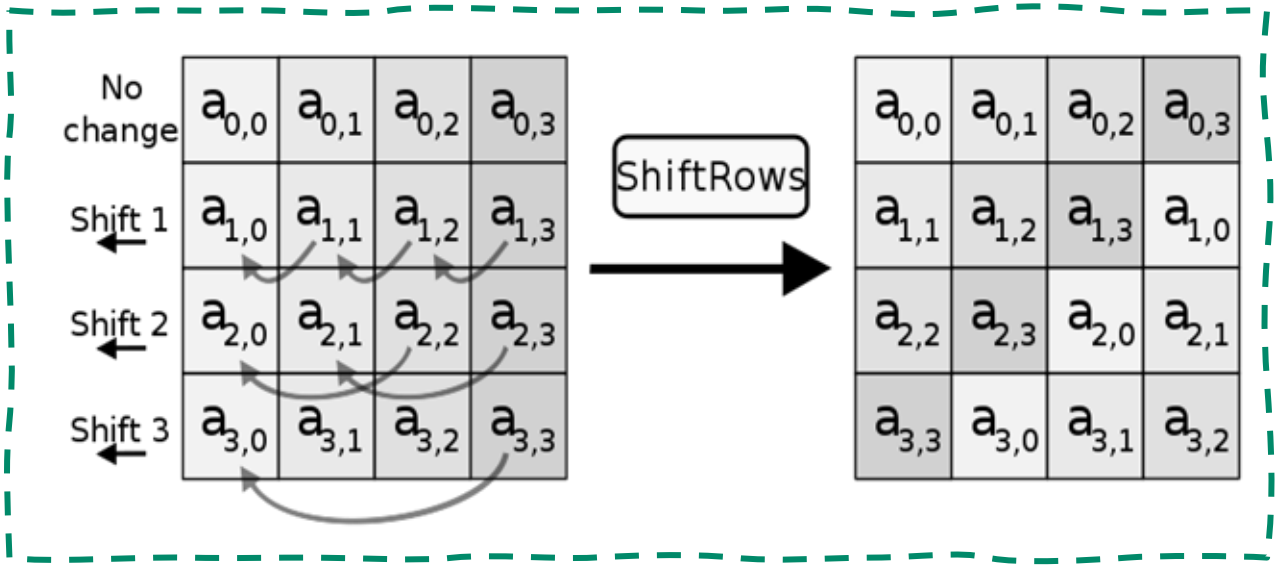
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	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

Inverse AES S-Box



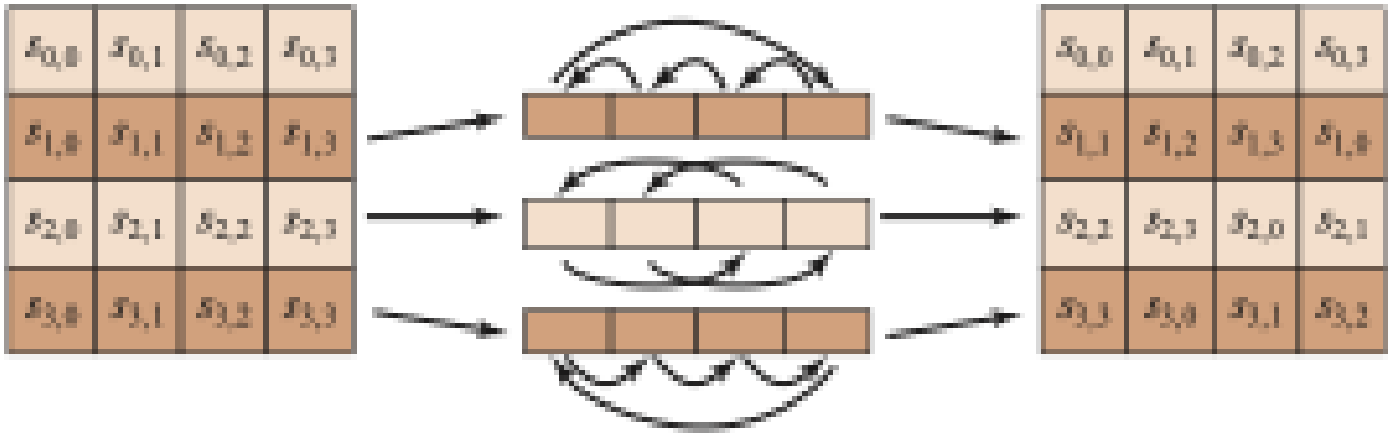
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED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

Shift Rows



- A circular byte shift in each row
  - 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

→

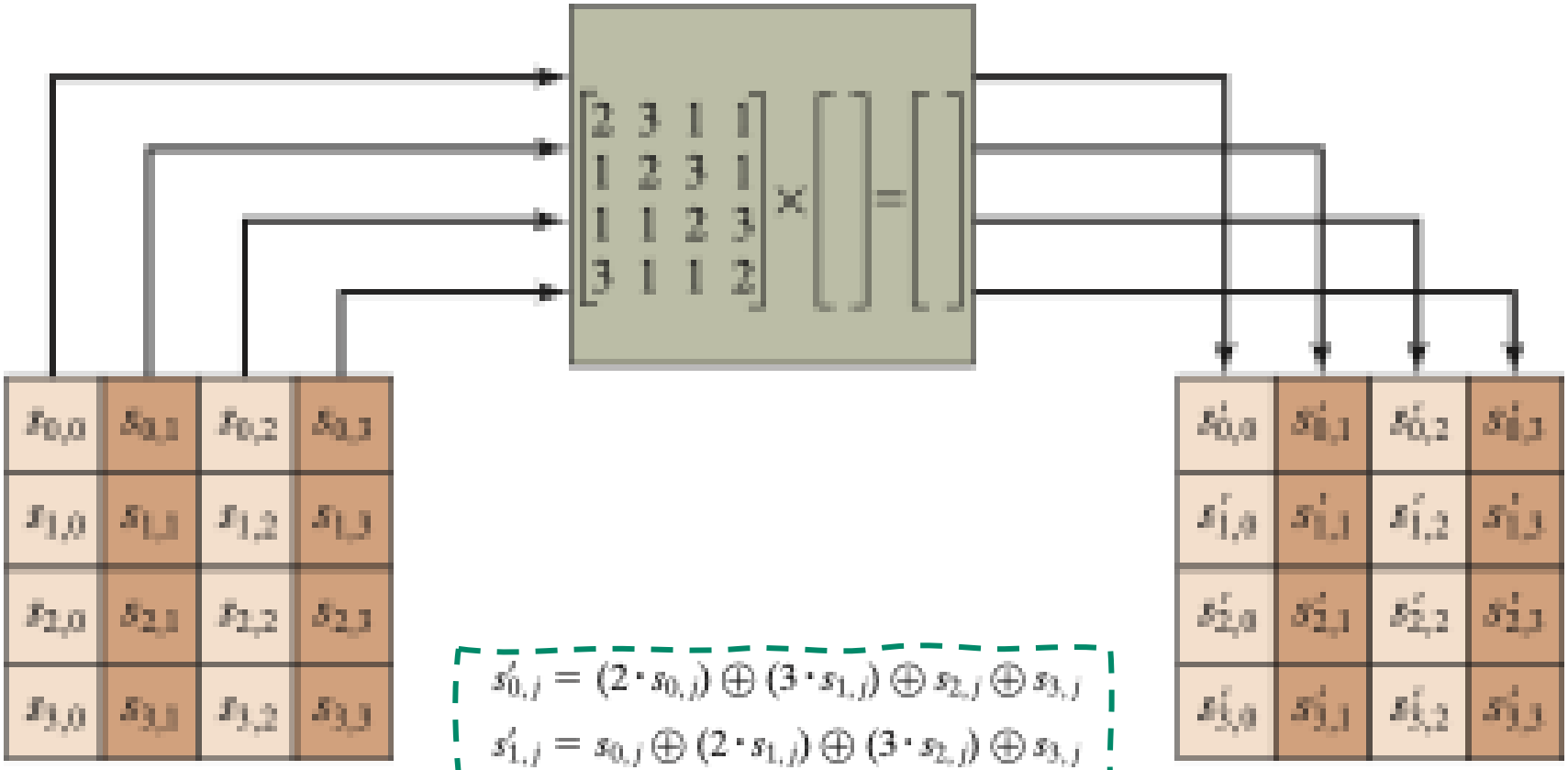
87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

Mix Columns

- Each column is processed separately
- Each byte is replaced by a value dependent on all 4 bytes in the column
- Effectively a matrix multiplication in $GF(2^8)$ using prime polynomial
- $m(x) = x^8 + x^4 + x^3 + x + 1$
- Decryption requires use of inverse matrix

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

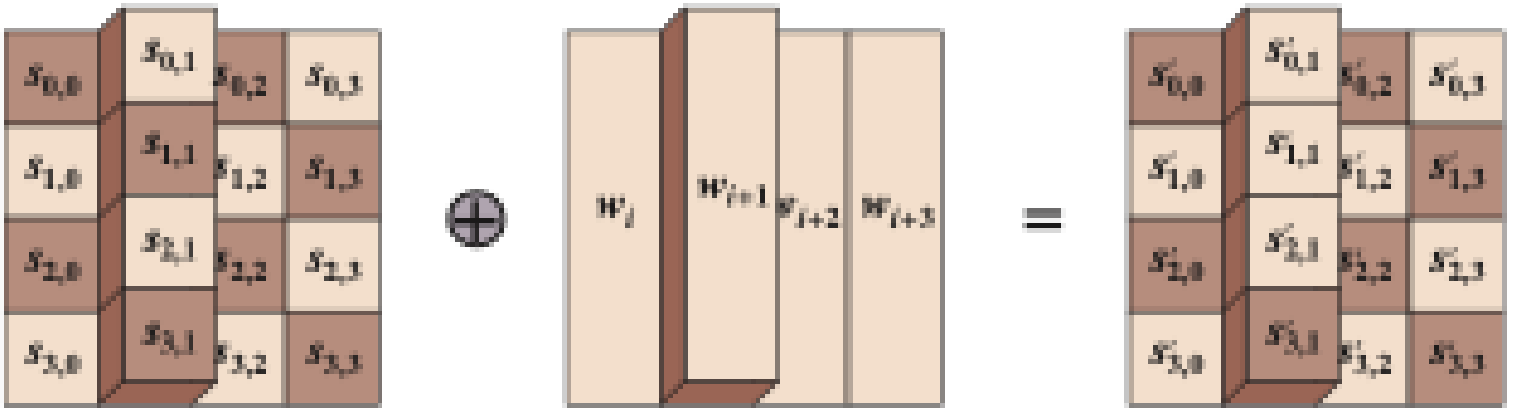
Mix Columns



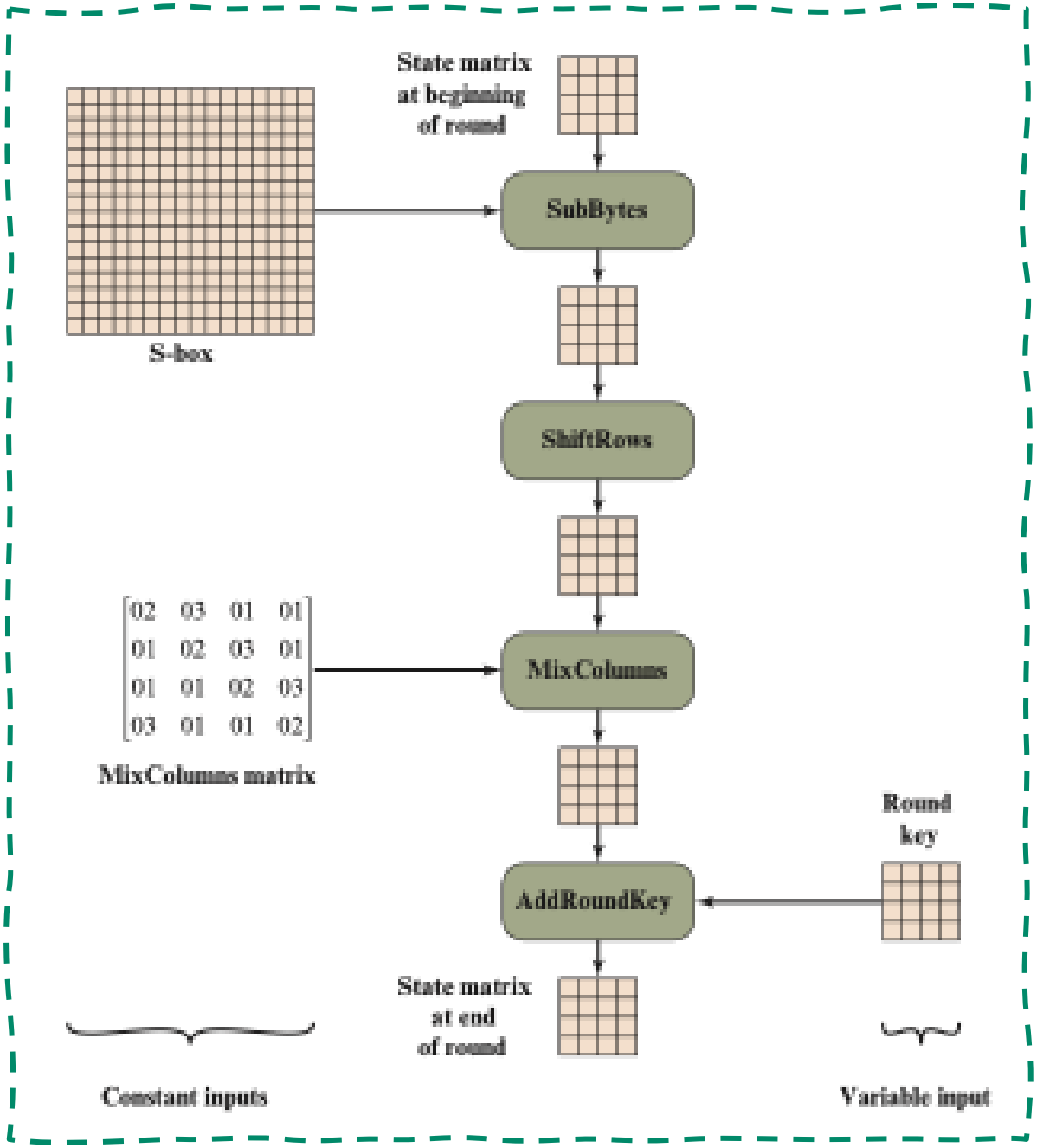
$$s'_{0,j} = (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$
$$s'_{1,j} = s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j}) \oplus s_{3,j}$$
$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j})$$
$$s'_{3,j} = (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j})$$

Add Round Key

- XOR state with 128-bits of the round key
- Same for decryption but with reverse key order
- Designed to be as simple as possible



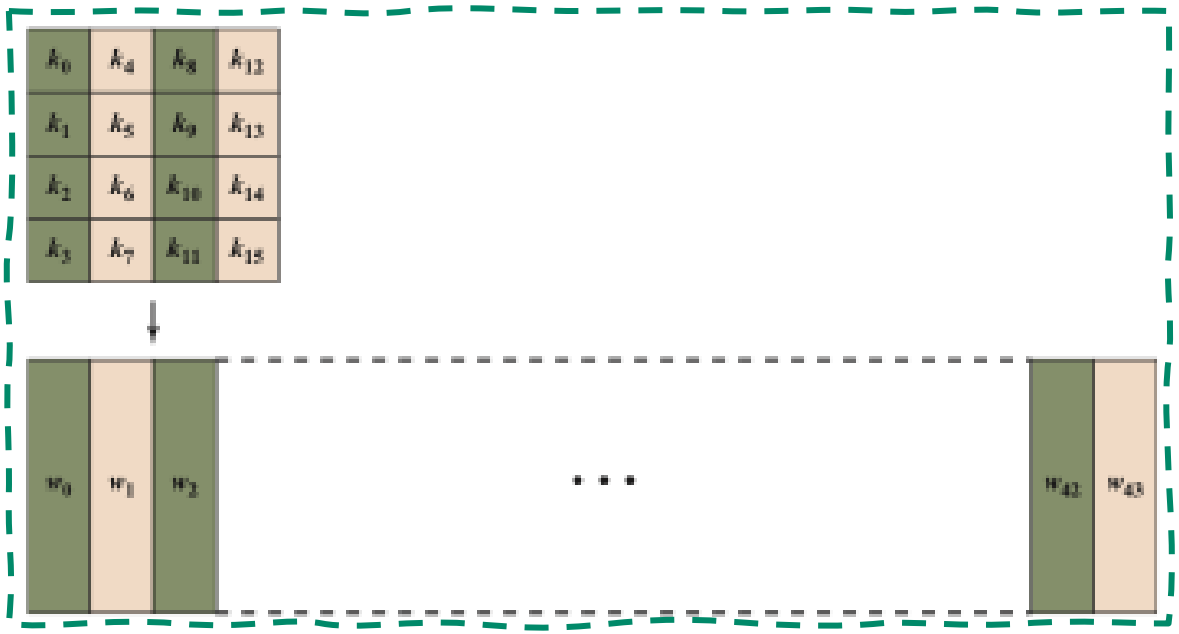
AES Round


State matrix at beginning of round
SubBytes
S-box
ShiftRows
02 03 01 01
01 02 03 01
01 01 02 03
03 01 01 02
MixColumns
MixColumns matrix
Round key
AddRoundKey
State matrix at end of round
Constant inputs
Variable input

AES Key Expansion (for AES-128)



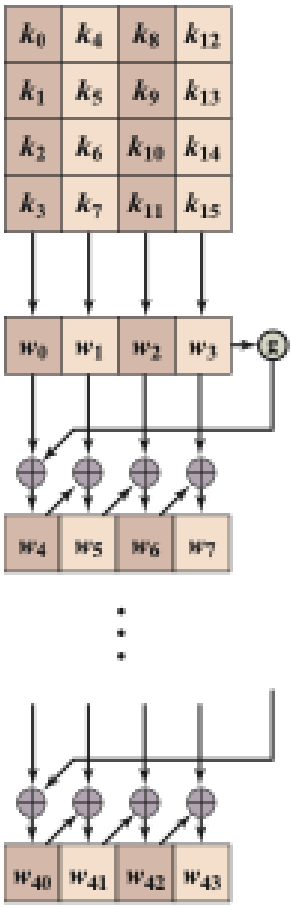
- Takes 128-bit (16-byte) key and expands into array of 44 words (word size = 32-bits)
- Start by copying key into first 4 words
- Then loop creating words that depend on values in previous & 4 places back



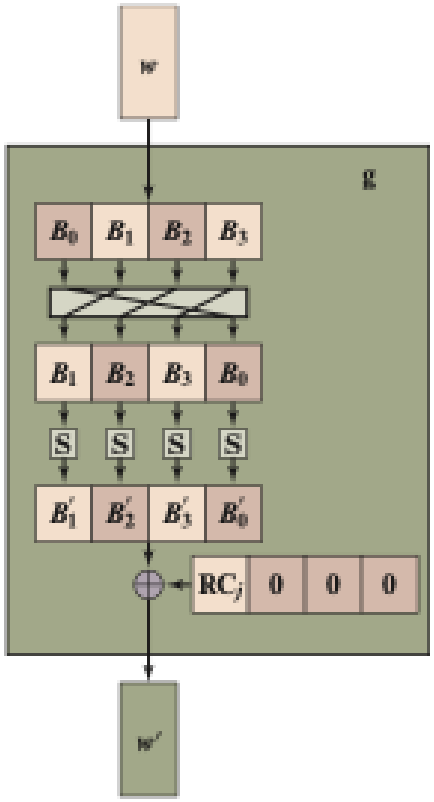
AES Key Expansion

```
KeyExpansion (byte key[16], word w[44])
{
    word temp
    for (i = 0; i < 4; i++) w[i] = (key[4*i], key[4*i+1],
                                    key[4*i+2],
                                    key[4*i+3]);
    for (i = 4; i < 44; i++)
    {
    temp = w[i - 1];
    if (i mod 4 = 0) temp = SubWord (RotWord (temp))
                            ⊕ Rcon[i/4];
    w[i] = w[i-4] ⊕ temp
    }
}
```

j	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36



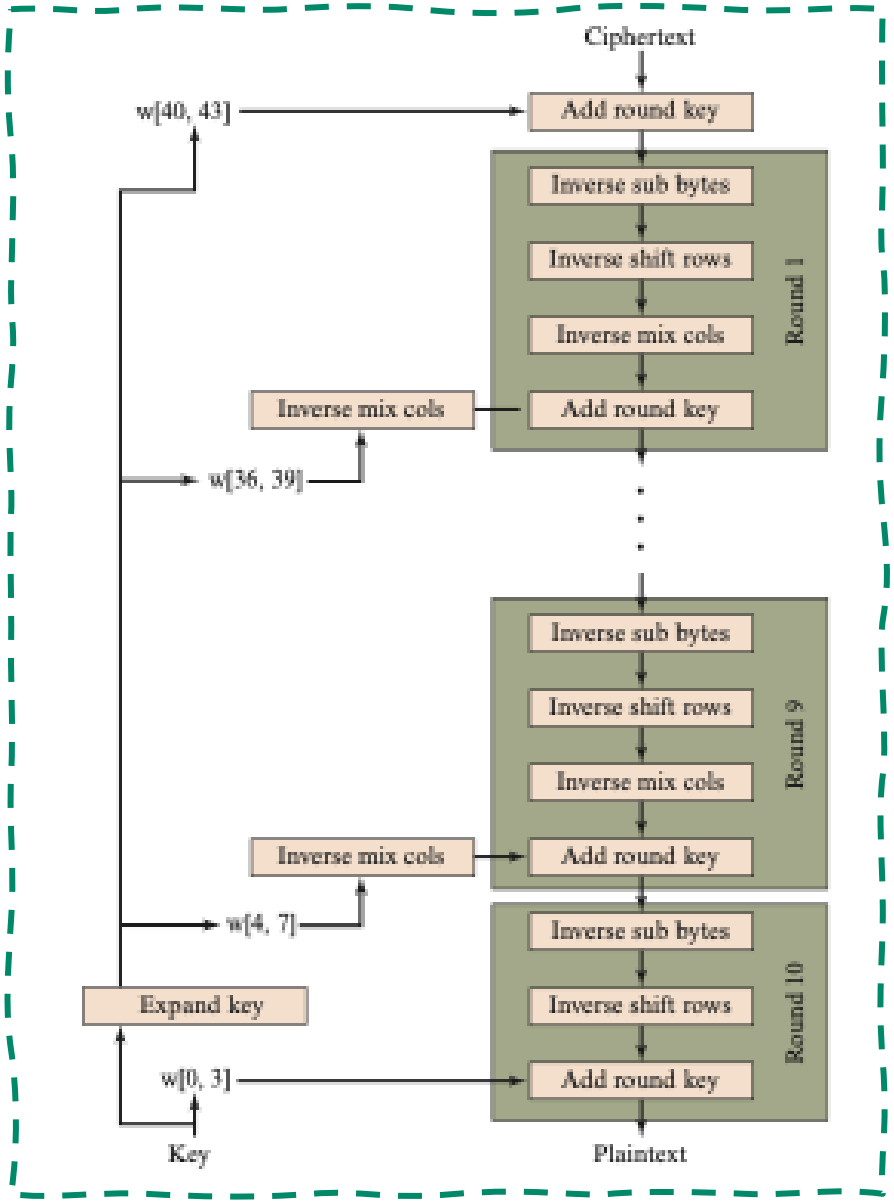
(a) Overall algorithm



(b) Function g

AES Decryption

- AES decryption is not identical to encryption since steps done in reverse
- but can define an equivalent inverse cipher with steps as for encryption
  - but using inverses of each step
  - with a different key schedule



AES Avalanche

Round		Number of Bits that Differ
	0123456789abcdeffedcba9876543210 0023456789abcdeffedcba9876543210	1
0	0e3634aece7225b6f26b174ed92b5588 0f3634aece7225b6f26b174ed92b5588	1
1	657470750fc7ff3fc0e8e8ca4dd02a9c c4a9ad090fc7ff3fc0e8e8ca4dd02a9c	20
2	5c7bb49a6b72349b05a2317ff46d1294 fe2ae569f7ee8bb8c1f5a2bb37ef53d5	58
3	7115262448dc747e5cdac7227da9bd9c ec093dfb7c45343d689017507d485e62	59
4	f867aee8b437a5210c24c1974cffeabc 43efdb697244df808e8d9364ee0ae6f5	61
5	721eb200ba06206dcbd4bce704fa654e 7b28a5d5ed643287e006c099bb375302	68
6	0ad9d85689f9f77bc1c5f71185e5fb14 3bc2d8b6798d8ac4fe36a1d891ac181a	64
7	db18a8ffa16d30d5f88b08d777ba4eaa 9fb8b5452023c70280e5c4bb9e555a4b	67
8	f91b4fbfe934c9bf8f2f85812b084989 20264e1126b219aef7feb3f9b2d6de40	65
9	cca104a13e678500ff59025f3bafaa34 b56a0341b2290ba7dfdfbddcd8578205	61
10	ff0b844a0853bf7c6934ab4364148fb9 612b89398d0600cde116227ce72433f0	58



Exercise (S-AES)