

Cybersecurity Concepts

Prepared by Dr. Diaa Salama

FNT625
Advanced Encryption Standard (AES)

Lecture 8





مجموعة جامعة
الجميع الذكية
للتعليم القابضة

MIDOCEAN
UNIVERSITY

Cybersecurity Concepts

Lecture 8

FNT625

Advanced Encryption Standard (AES)

Prepared by
Dr. Diao Salama

www.midocean.ae

PLEASE
BE QUIET



10



Lecture

08

Lets Start the Eighth Lecture



Advanced Encryption Standard (AES)



Advanced Encryption Standard

Simplified Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES)

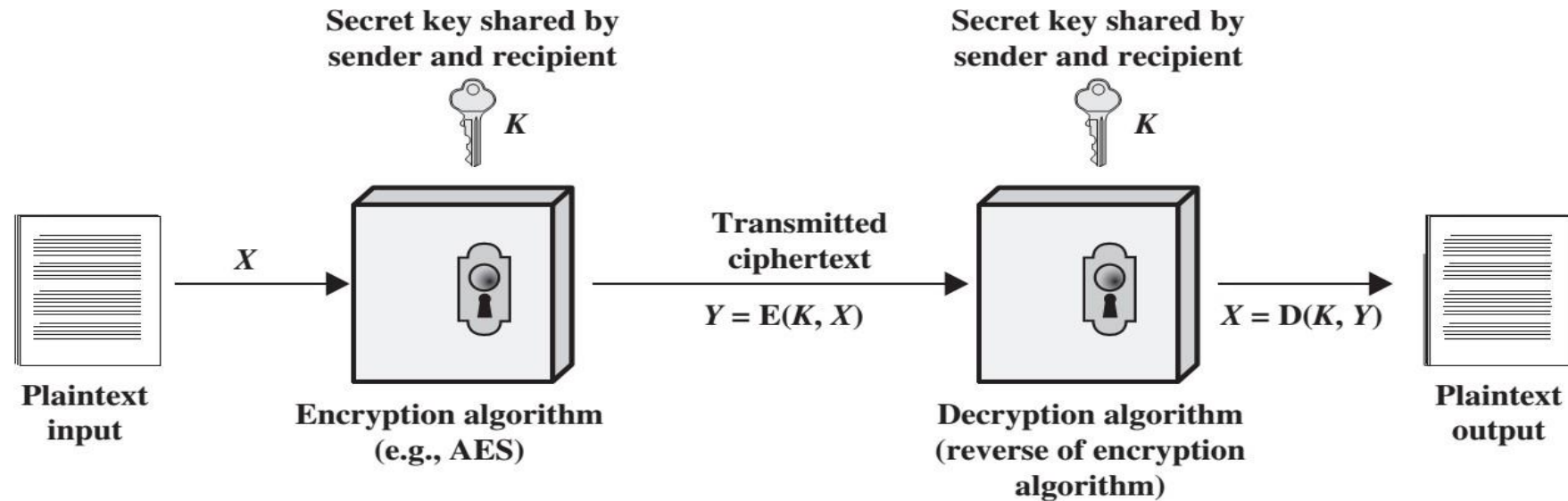
Simplified Advanced Encryption Standard

Cryptography and Network Security

Lecture

08

Advanced Encryption Standard (AES)



- 1999: NIST indicates that DES should be replaced as a standard
 - have theoretical attacks that can break it
 - have demonstrated exhaustive key search attacks
- Triple-DES should be used before a new standard is approved
 - 3DES has 168-bit key: brute-force attack is infeasible
 - Same engine as for DES is running – software/hardware can be reused
 - Confident that no effective attack exists: DES has been around for a long time
 - As far as security is concerned, 3DES is ideal to replace DES
- Drawbacks of 3DES
 - Relatively slow software implementations – DES was designed for 1970's hardware
 - DES and 3DES use 64-bit blocks: for efficiency and security, larger blocks should be used

- US NIST issued call for ciphers in 1997
 - ❑ 15 candidates accepted in Jun 98
 - ❑ 5 were shortlisted in Aug-99
 - ❑ Rijndael was selected as the AES in Oct-2000
 - ❑ issued as FIPS PUB 197 standard in Nov-2001

Requirements for the new standard

- Private key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than 3DES
- Active life of 20-30 years (+ archival use)
- Provide full specification and design details
- Both C and Java implementations
- NIST have released all submissions & unclassified analyses

■ Initial criteria:

- ❑ security – effort to practically cryptanalyze
- ❑ cost – computational efficiency, so as to be used in high-speed applications, such as broadband links
- ❑ algorithm and implementation characteristics: should be suitable for a variety of soft/hard implementations, simple enough to make analysis straightforward

■ Final criteria:

- ❑ general security: this was conducted by the public (academic) cryptographic community: people published various attacks and weaknesses of the candidates
- ❑ software and hardware implementation ease: execution speed, performance on various platforms, variation of speed with key size
- ❑ Attacks on implementation: timing attacks and power analysis
- ❑ Multiplication consumes more power and takes more time than addition
- ❑ Writing 1s consumes more power and takes more time than writing 0s
- ❑ Flexibility (in encryption/decryption, key change, other factors)

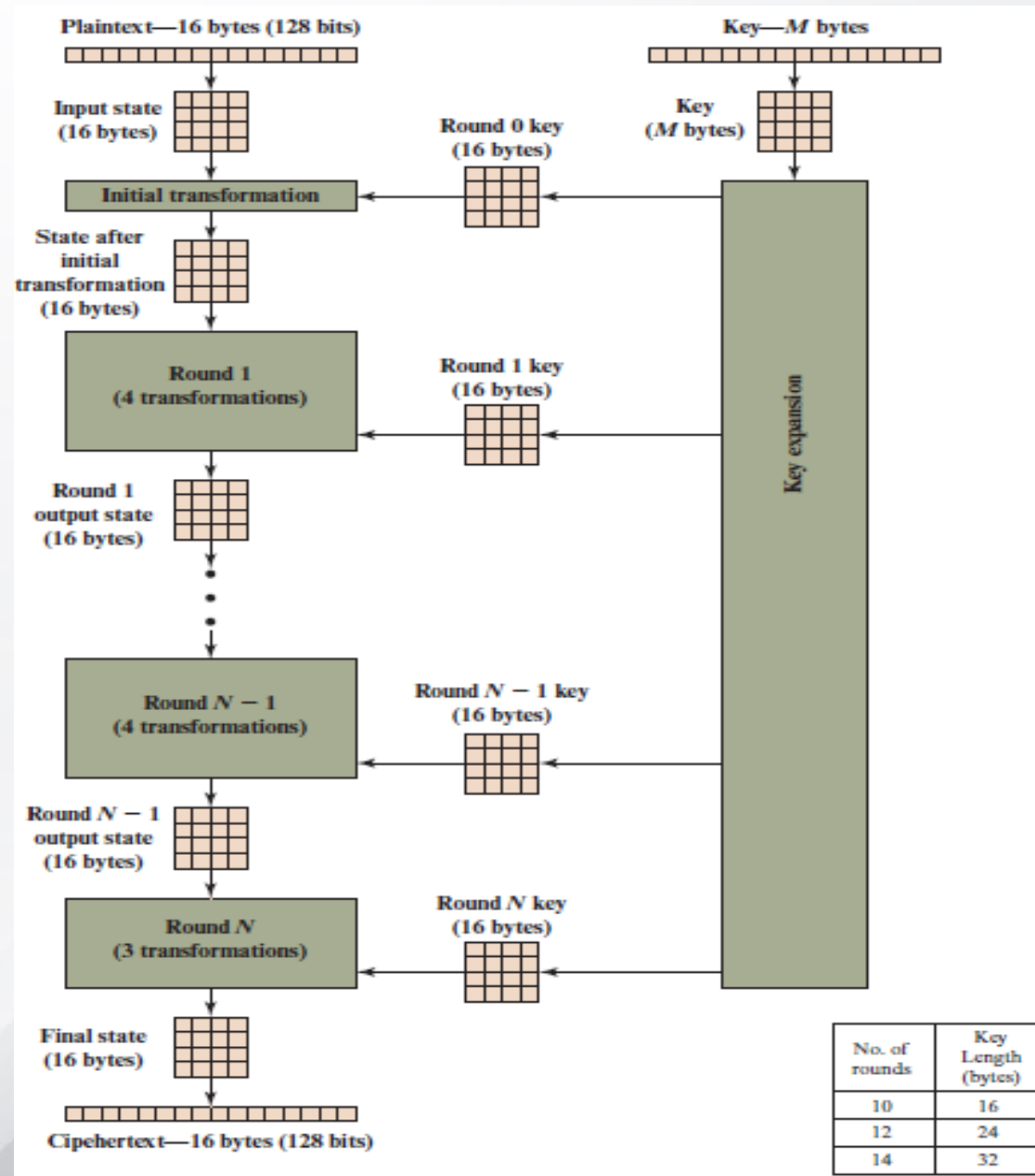
- After testing and evaluation, shortlist in Aug-99:
 - MARS (IBM) - complex, fast, high security margin
 - RC6 (USA) - v. simple, v. fast, low security margin
 - Rijndael (Belgium) - clean, fast, good security margin
 - Serpent (Euro) - slow, clean, v. high security margin
 - Twofish (USA) - complex, v. fast, high security margin
- Then subject to further analysis & comment
- Analysed contrast between algorithms with
 - few complex rounds vs. many simple rounds
 - which refined existing ciphers vs. new proposals

The AES Cipher – Rijndael

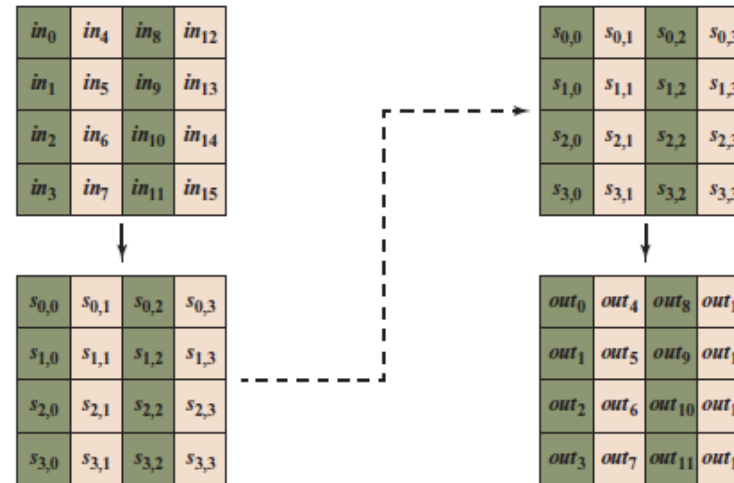
- Designed by Rijmen-Daemen in Belgium
- 128/192/256-bit keys, 128 bit data
- Does not have the structure of a classical *feistel* cipher
 - treats data in 4 groups of 4 bytes
 - operates an entire block in every round
- Designed to be:
 - resistant against known attacks
 - speed and code compactness on many platforms
 - design simplicity
- Decryption algorithm different than the encryption

- Processes data as 4 groups of 4 bytes – 128-bit block
- Input block copied into **State** array, modified at each stage of encryption or decryption and copied to the output matrix after the final round
- has 9/11/13 rounds (depending on which variant is used) in which **State** undergoes:
 - byte substitution (one S-box used on every byte)
 - shift rows: a simple permutation
 - mix columns: substitution using arithmetic in $GF(2^8)$
 - add round key (XOR **State** with the round key)
- initial XOR of the plaintext with a round key
- There is an incomplete last round (the 10th/12th/14th)
- Note: all operations can be combined into XOR and table lookups - hence very fast and efficient

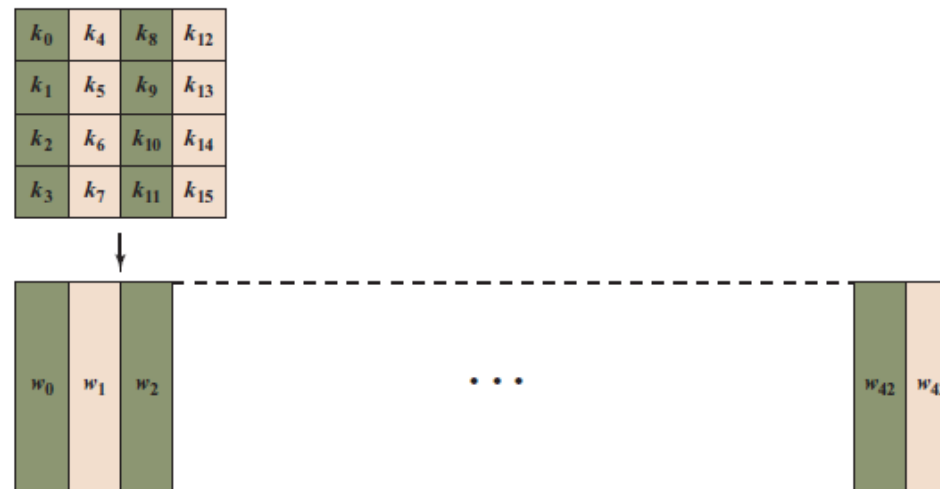
Figure 6.1 AES Encryption Process



Rijndael data structures



(a) Input, state array, and output

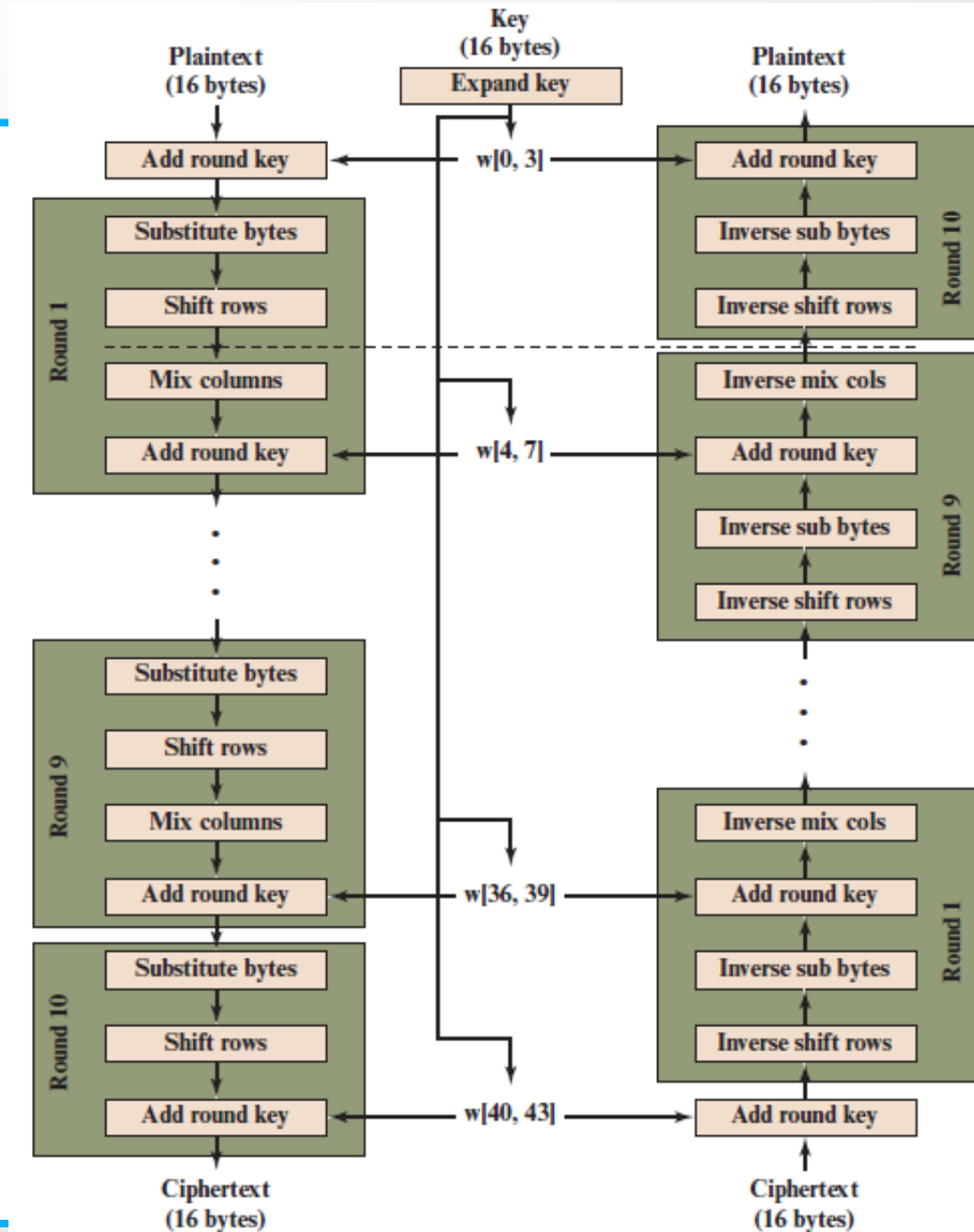


(b) Key and expanded key

Table 6.1 AES Parameters

Key Size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Plaintext Block Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Number of Rounds	10	12	14
Round Key Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Expanded Key Size (words/bytes)	44/176	52/208	60/240

Figure 6.3 AES Encryption and Decryption

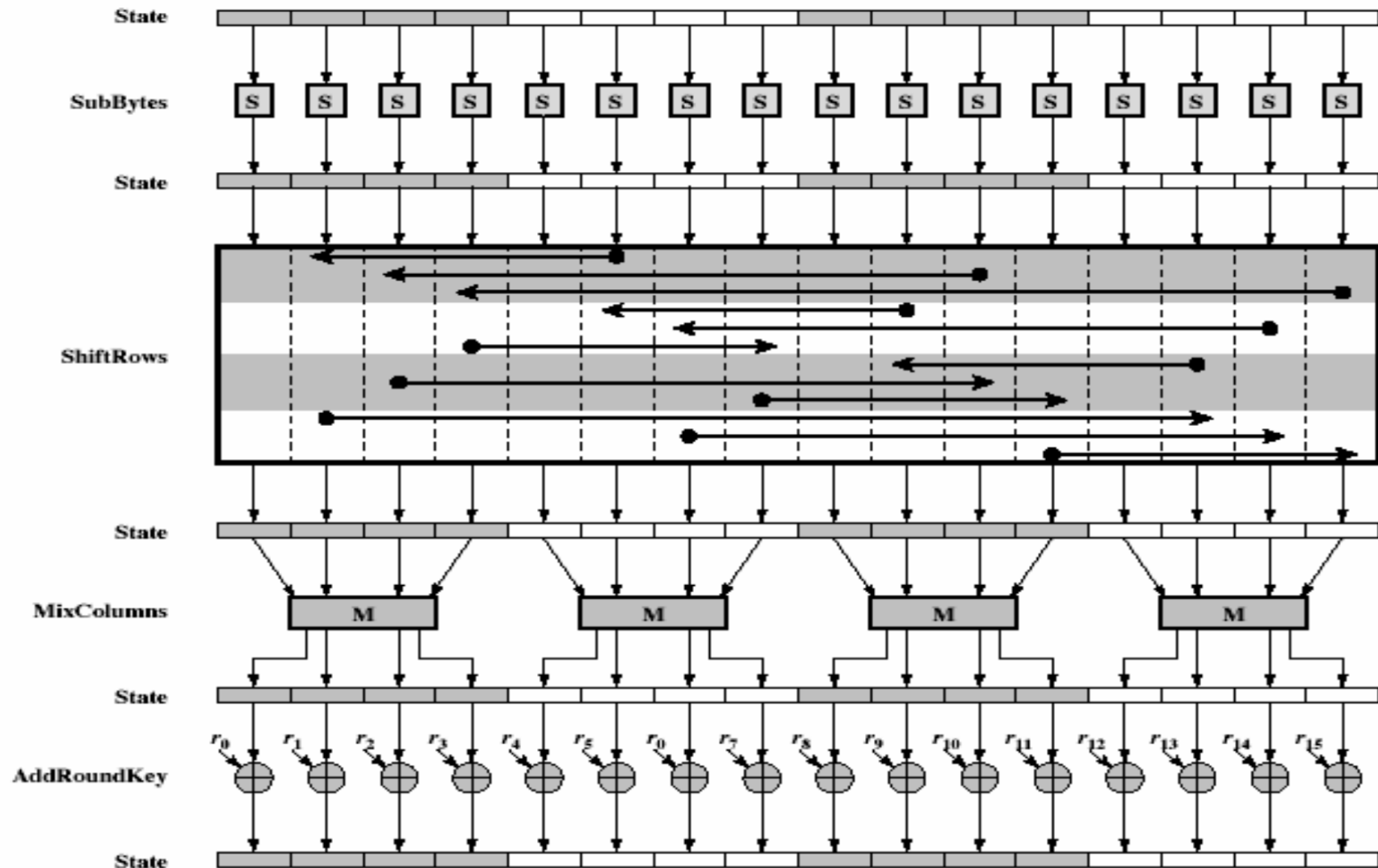


Detailed Structure (1 of 2)

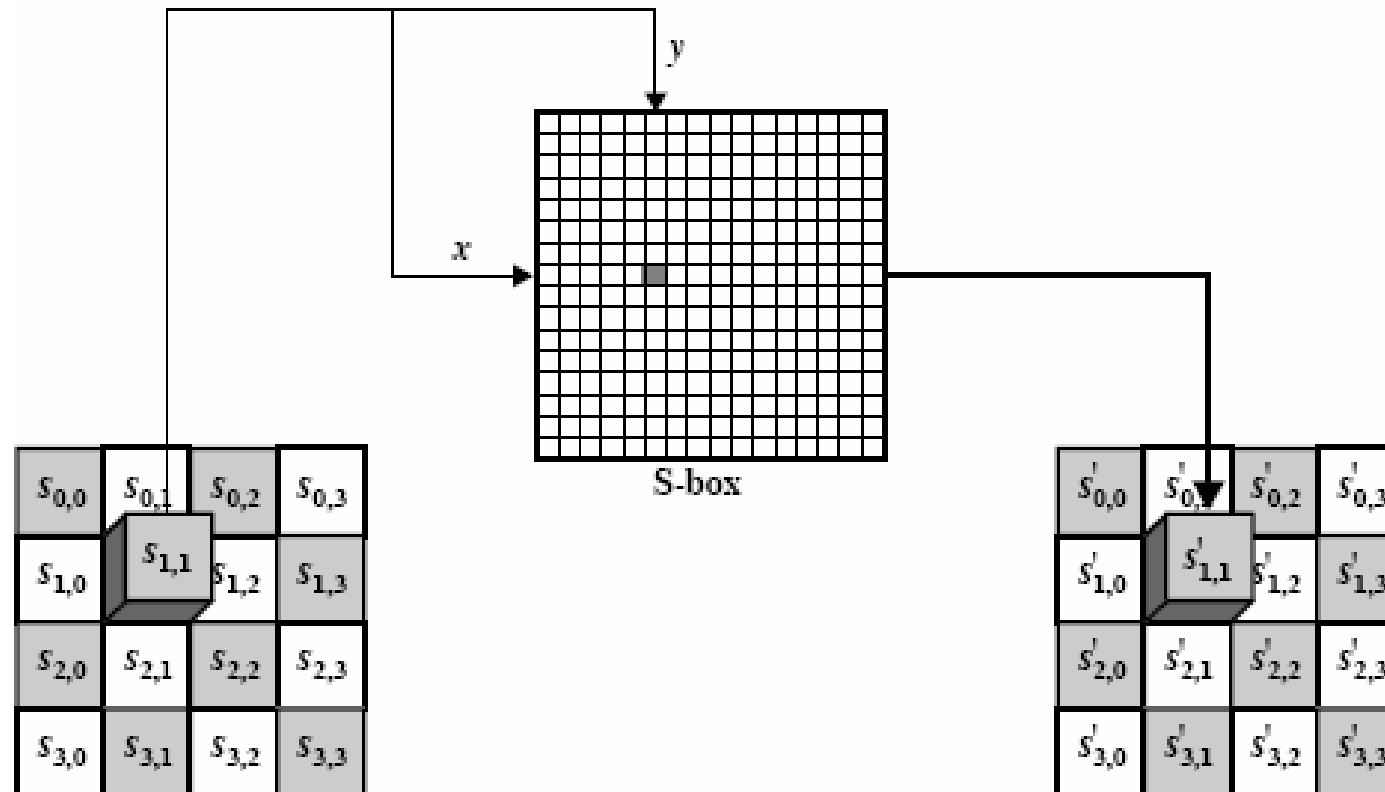
- Processes the entire data block as a single matrix during each round using substitutions and permutation
- The key that is provided as input is expanded into an array of forty-four 32-bit words, $w[i]$
- Four different stages are used:
 - Substitute bytes – uses an S-box to perform a byte-by-byte substitution of the block
 - ShiftRows – a simple permutation
 - MixColumns – a substitution that makes use of arithmetic over $GF(2^8)$
 - AddRoundKey – a simple bitwise XOR of the current block with a portion of the expanded key

- Discuss each of the four stages in each round
- For each stage, describe the encryption and decryption algorithms
- Describe the rationale of each stage
- Describe key expansion
- Each stage operates on the 128-bit **State**, i.e., 16 bytes – these will be treated as a 4x4 matrix of bytes
 - The first four bytes are on the first column, the following 4 on the second column, etc.
- In this description we assume key length of 128
 - Details are only slightly different for longer keys
 - The 128-bit key is also shown as a matrix of 4x4 bytes (each byte is a number from 0 to 255): first 4 bytes on the first column, following 4 on the 2nd columns, etc.
 - The key is expanded in an array of 44 *words* – each word has 4 bytes; these key words are used throughout the rounds: in each XOR are involved 4 words

Implementation: one AES Encryption Round



Substitute bytes transformation using The S-box



(a) Substitute byte transformation

AES S-boxes

		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

(a) S-box

AES S-boxes

		<i>y</i>															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
<i>x</i>	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

(b) Inverse S-box

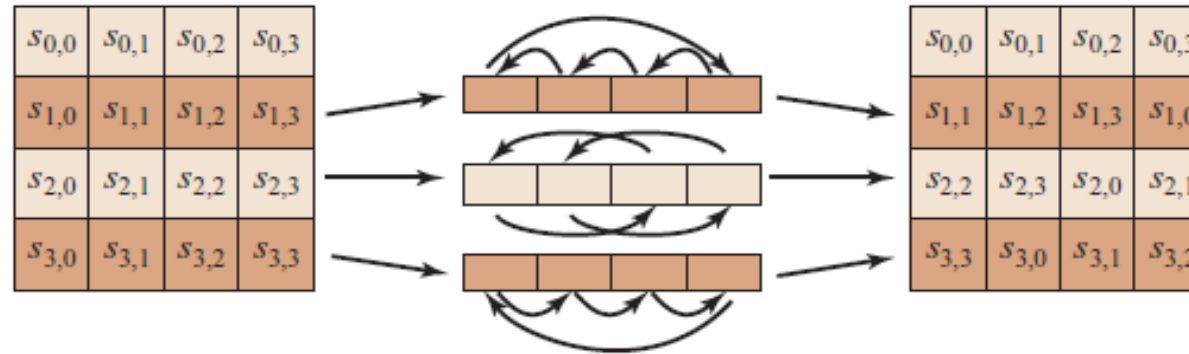
Substitute bytes transformation

- **Note:** we will denote hexadecimal numbers in parenthesis: {45}, {A6}, {CF}
- SubBytes is a simple table lookup (see next slide)
 - table of 16x16 bytes containing a permutation of all 256 8-bit values
- Each byte of **state** is replaced by a byte in the table
 - The row is given by the leftmost 4-bits of the byte
 - The column is given by the rightmost 4-bits of the byte
 - Example: byte {95} (in hexa) is replaced by the byte on row 9, column 5: {2A} (see next slide)
- For **decryption**, use the inverse S-box

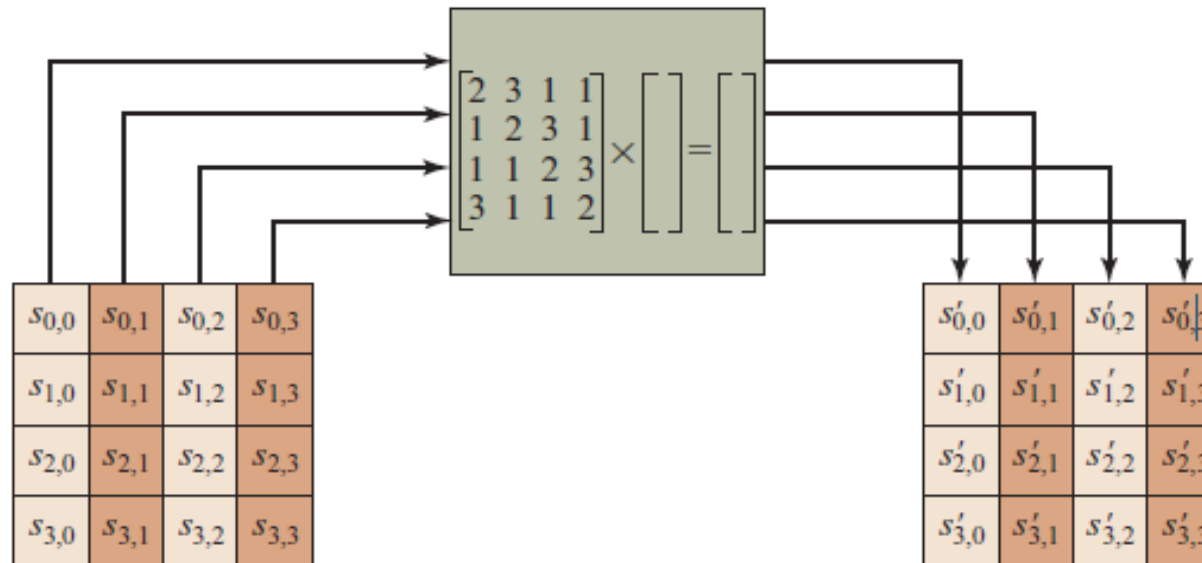
Shift Row Transformations

- For **encryption**, 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
- For **decryption**, all shifts are done to right
- **Why this particular shift?**
 - Since **state** is processed by columns, this step permutes bytes between the columns
 - The 4 bytes of one column are spread out to 4 different columns
- This step provides “permutation” of the data, whereas the other steps provide substitutions (recall Shannon’s S-P networks)

A E S Row and Column Operations

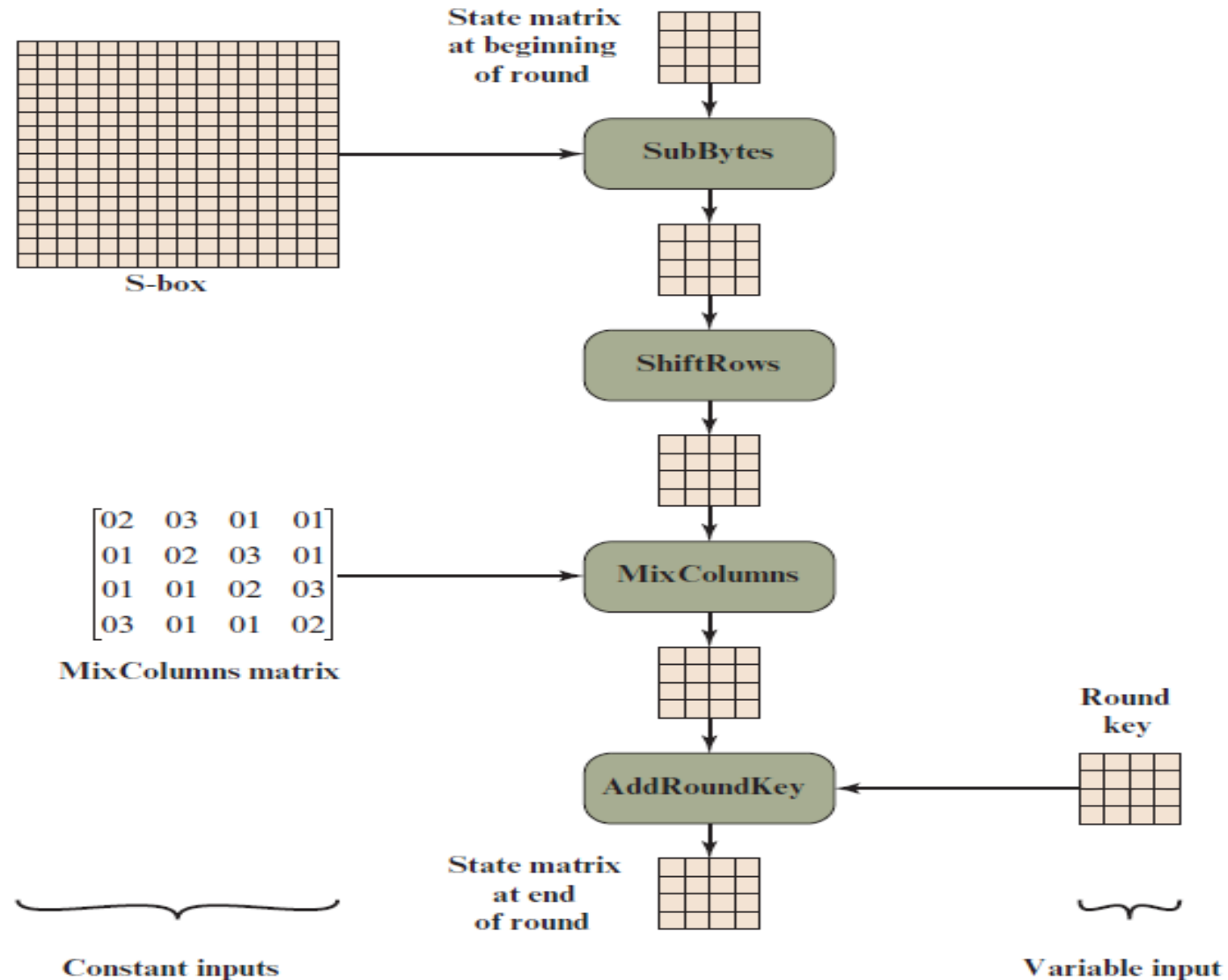


(a) Shift row transformation

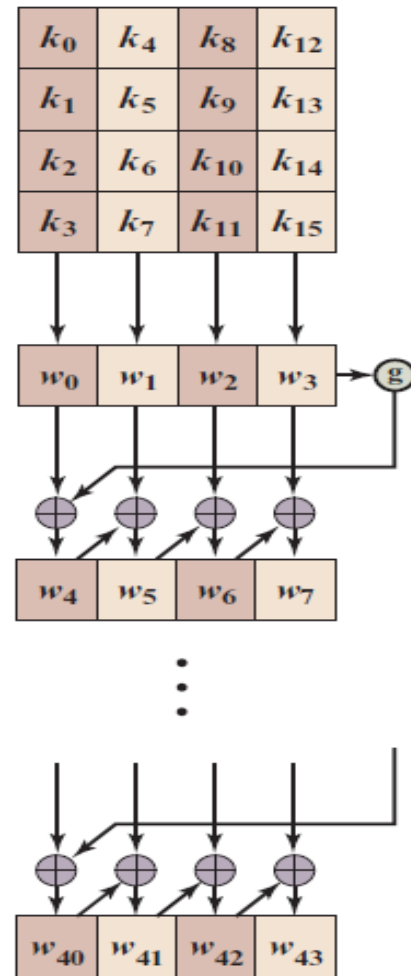


(b) Mix column transformation

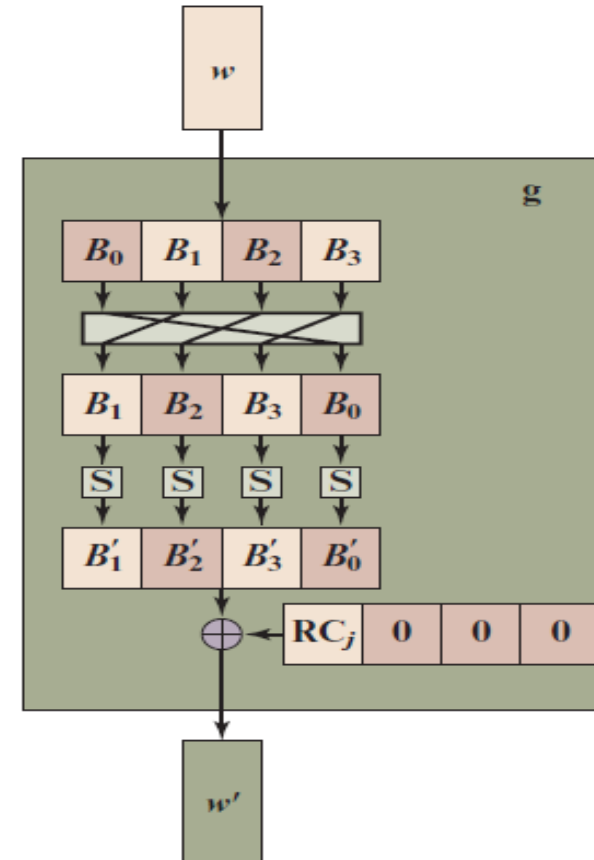
Inputs for Single A E S Round



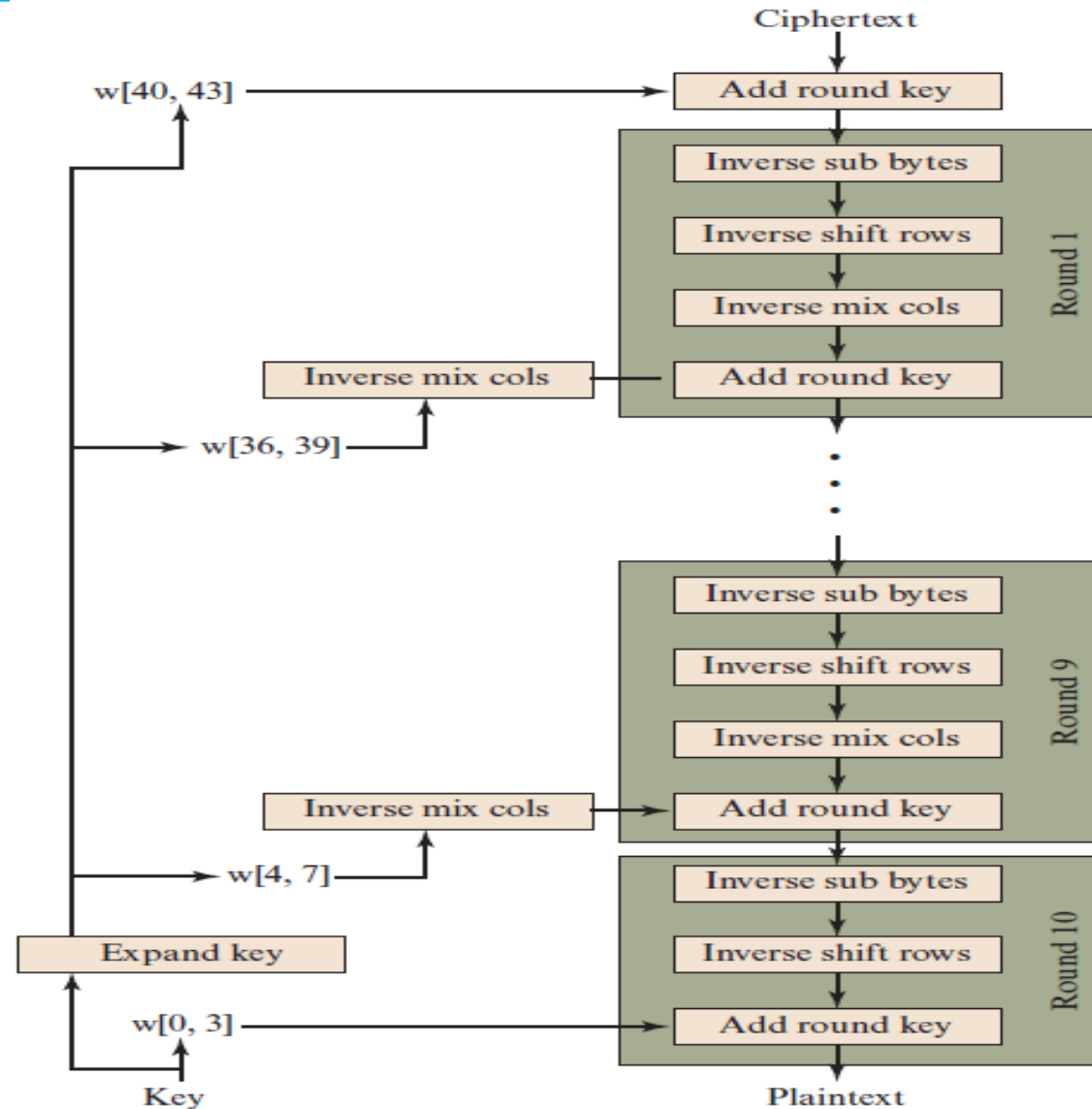
A E S Key Expansion



(a) Overall algorithm

(b) Function g

Equivalent Inverse Cipher



- XOR state with 128-bits of the round key
- Same in the decryption – use the round keys in reverse order

$$\oplus \begin{array}{|c|c|c|c|} \hline s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ \hline s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ \hline s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ \hline s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \\ \hline \end{array} \approx \begin{array}{|c|c|c|c|} \hline w_i & w_{i+1} & w_{i+2} & w_{i+3} \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ \hline s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ \hline s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ \hline s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \\ \hline \end{array}$$

(b) Add Round Key Transformation



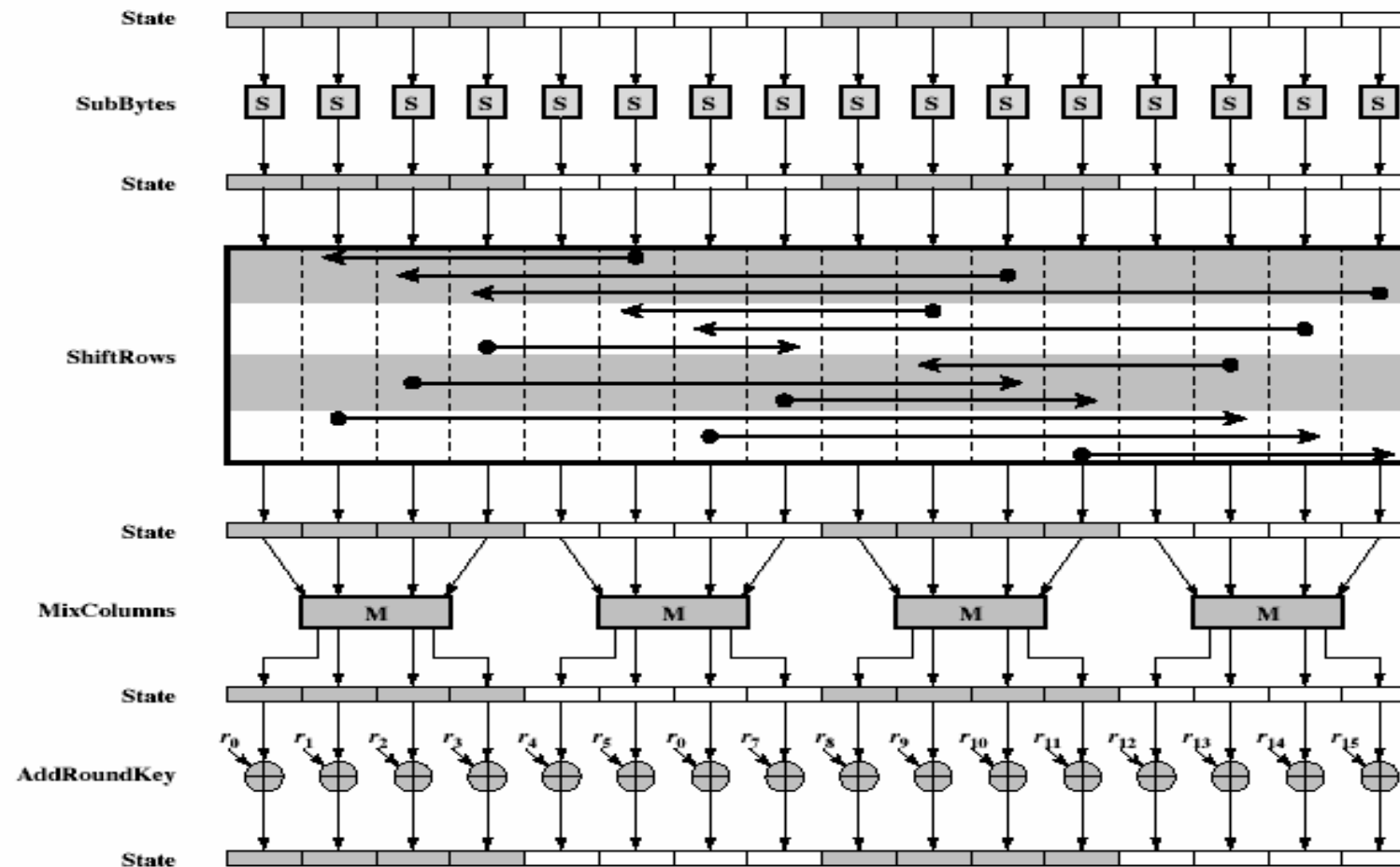
(b) Key and expanded key

AES Key Expansion

Takes 128-bit (16-byte) key and expands into an array of 44 32-bit words (each word has 4 bytes)

- AES decryption is not identical to encryption since steps must be performed in reverse
 - ❑ This is a disadvantage because different software is needed for encryption and decryption
 - ❑ Encryption is an iteration of SubBytes, ShiftRows, MixColumns, AddRoundKey
 - ❑ Decryption is an iteration of InvShiftRows, InvSubBytes, AddRoundKey, InvMixColumns
- Decryption can be modified to use the encryption software (switch the order of the first two operations and the order of the last two operations in each round), provided we operate a small change in the key expansion
 - ❑ Before we apply AddRoundKey in each round, we apply InvMixColumns to the matrix formed by the 4 words participating in the current round

- Implementation aspects: AES Round with State seen as an array with 16 bytes rather than as a 4x4 matrix



- Can efficiently implement on 8-bit CPU
 - byte substitution works on bytes using a table of 256 entries
 - shift rows is simple byte shifting
 - add round key works on byte XORs
 - mix columns requires matrix multiply in $GF(2^8)$ which works on byte values, can be simplified to use a table lookup

- Can efficiently implement on 32-bit CPU
 - redefine steps to use 32-bit words
 - can precompute 4 tables of 256-words
 - then each column in each round can be computed using 4 table lookups
 - + 4 XORs
 - at a cost of 16Kb to store tables
- Designers believe this very efficient implementation was a key factor in its selection as the AES cipher

Advanced Encryption Standard

Simplified Advanced Encryption Standard (AES)

Simplified Advanced Encryption Standard

S-AES Encryption and Decryption

S-AES Key Generation

S-AES Encryption

S-AES Decryption

Simplified Advanced Encryption Standard

S-AES Encryption and Decryption

S-AES Key Generation

S-AES Encryption

S-AES Decryption

- ❑ Simplified AES (S-AES) was developed by Professor Edward Schaefer of Santa Clara University in 2003
- ❑ its purpose is educational, since its key and block size are very small 16bits
- ❑ it is possible for students to encrypt or decrypt a block doing all operations by hand
- ❑ it easier for students to understand the structure AES

Simplified Advanced Encryption Standard

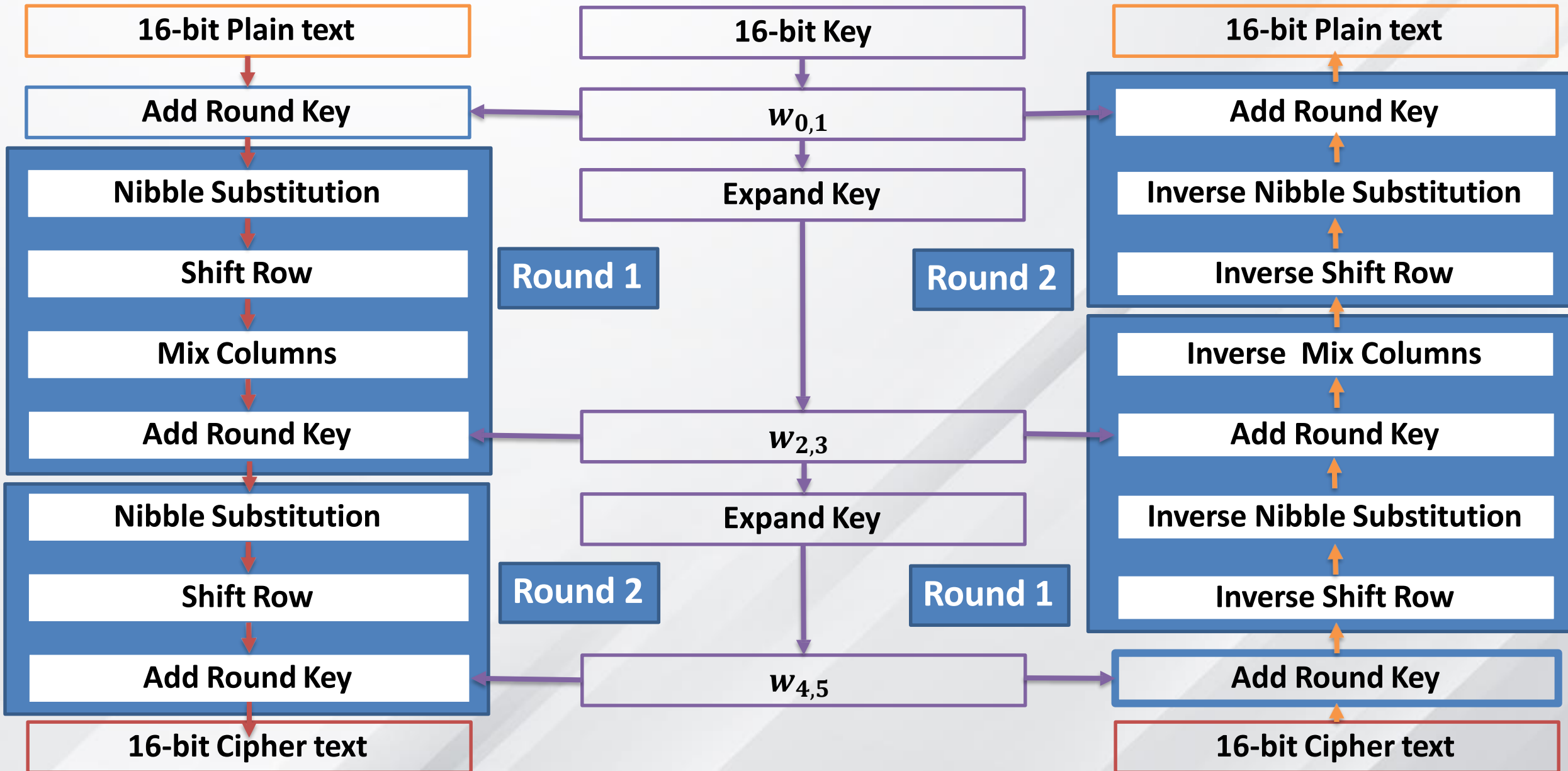
S-AES Encryption and Decryption

S-AES Key Generation

S-AES Encryption

S-AES Decryption

S-AES Encryption and Decryption



S-AES Encryption Example

□ 16-bit Plaintext, P= D7 28

=1101 0111 0010 1000

□ 16-bit Key, K= 4A F5

=0100 1010 1111 0101

Table of Contents

Simplified Advanced Encryption Standard

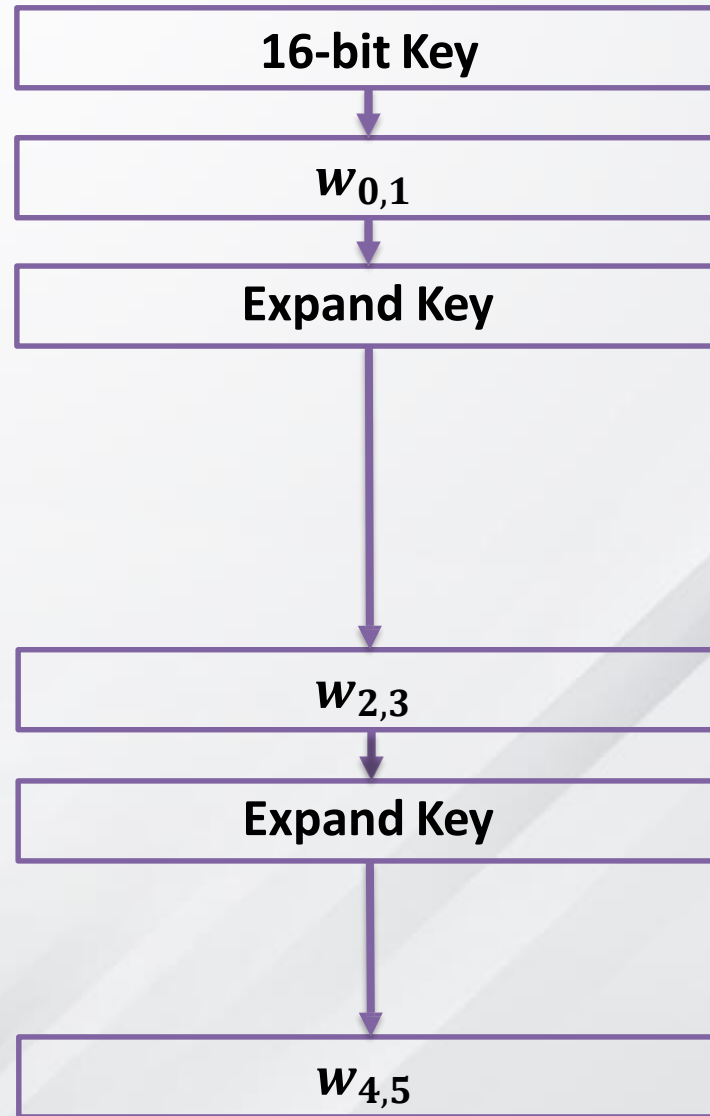
S-AES Encryption and Decryption

S-AES Key Generation

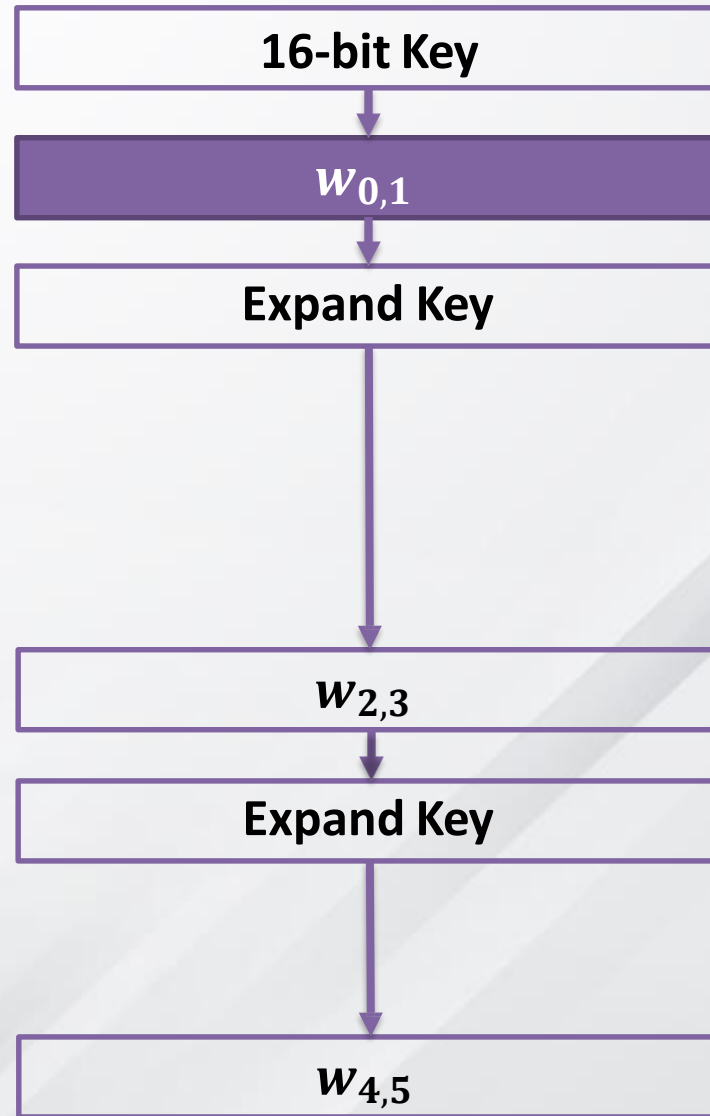
S-AES Encryption

S-AES Decryption

-AES Key Generation



-AES Key Generation



S-AES Key Generation

□ $K = 4A\ F5$

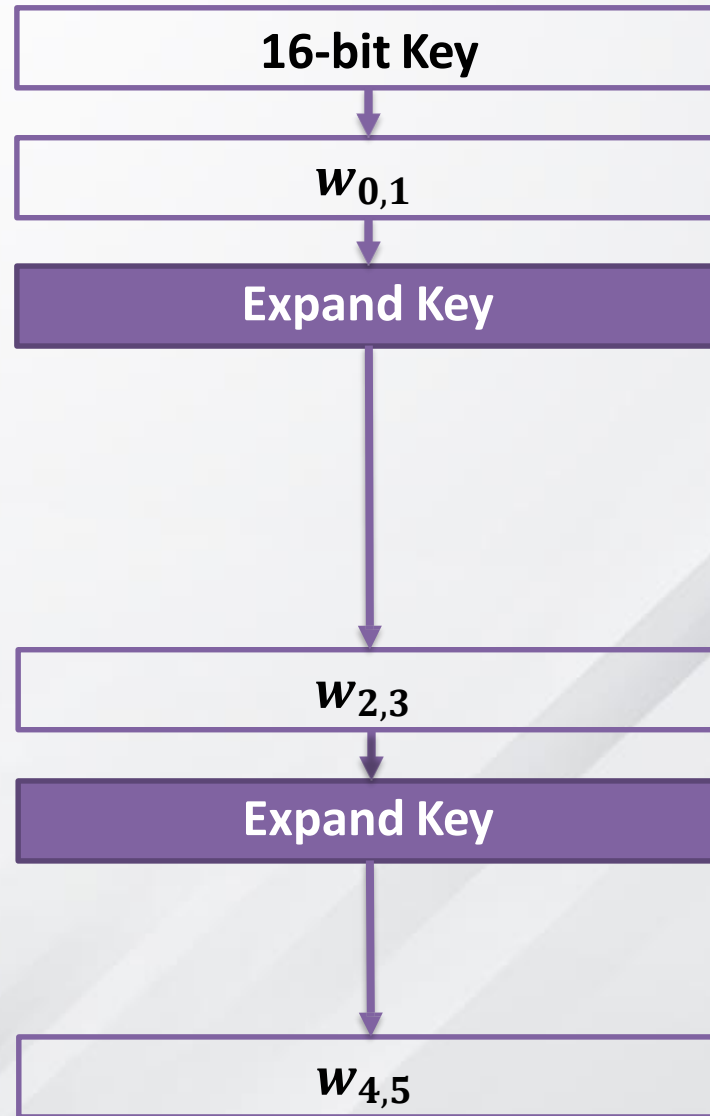
$= 0100\ 1010\ 1111\ 0101$

□ The input key, K , is split into 2 words, w_0 and w_1 :

□ $w_0 = 0100\ 1010$

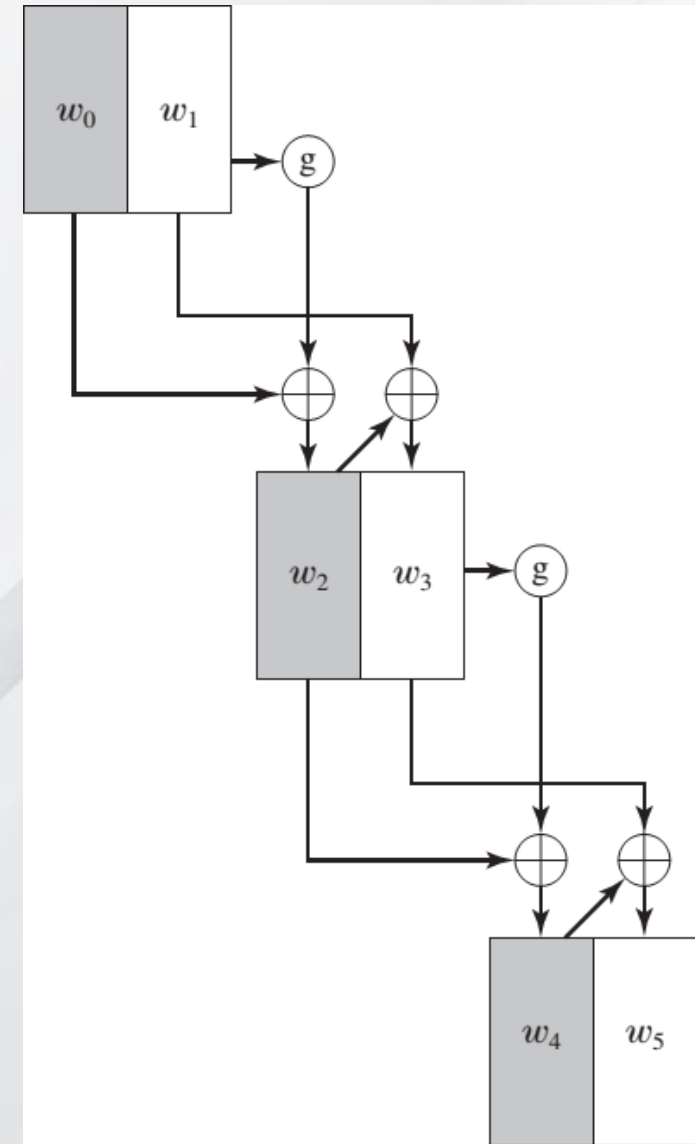
□ $w_1 = 1111\ 0101$

-AES Key Generation



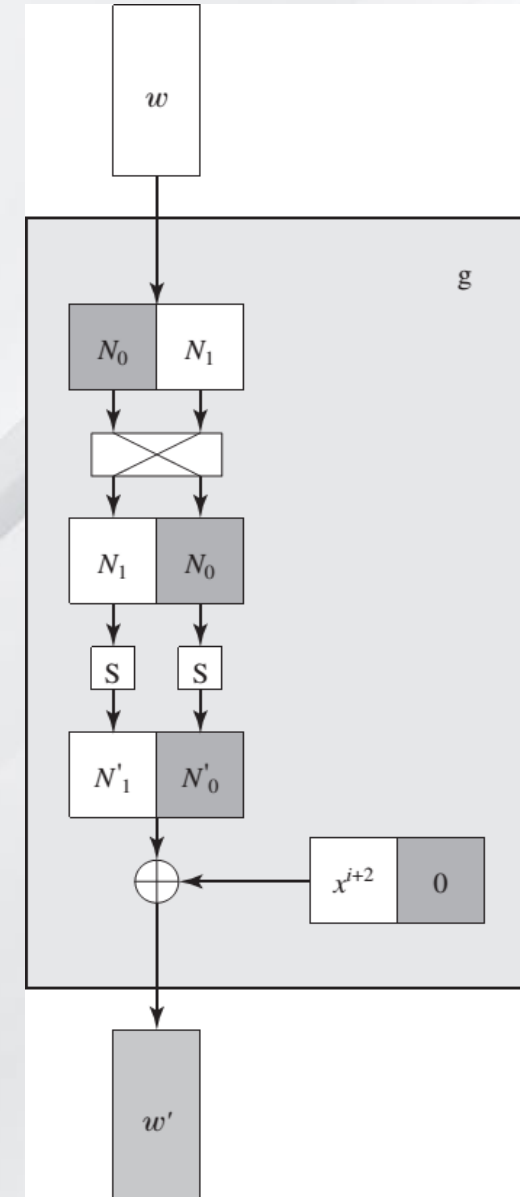
S-AES Key Generation

□ S-AES Key Expansion

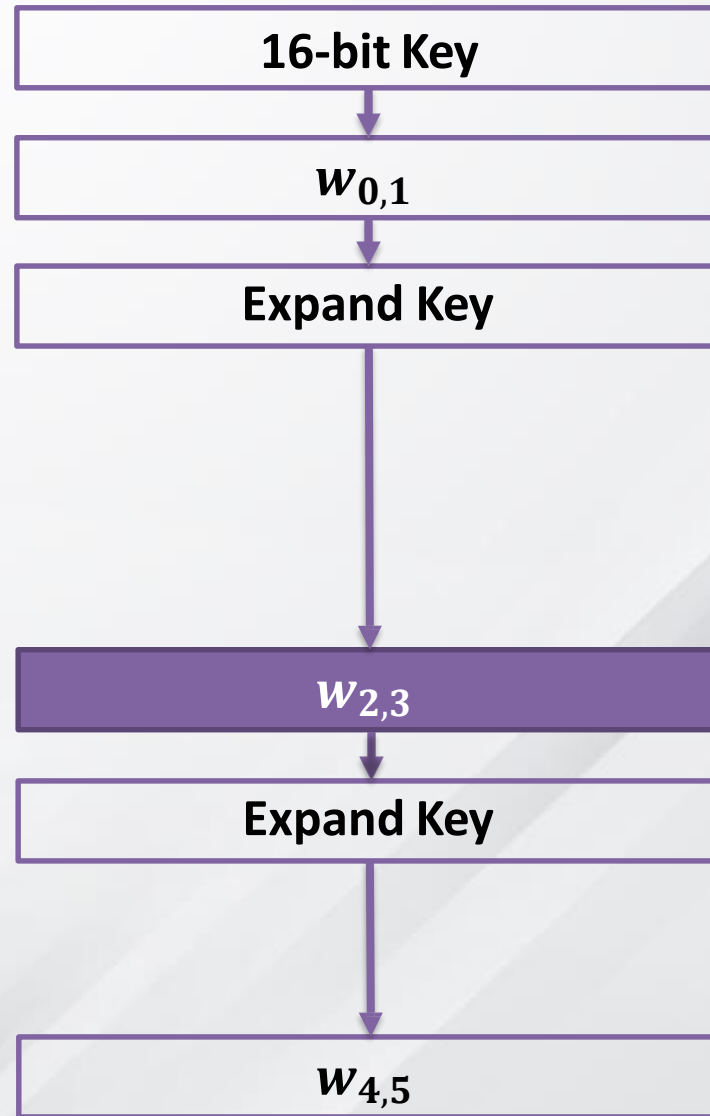


S-AES Key Generation

❑ Function g



-AES Key Generation



-AES Key Generation

- ❑ $w_0 = 0100\ 1010$, $w_1 = 1111\ 0101$
- ❑ $w_2 = w_0 \oplus \text{Rcon}(1) \oplus \text{SubNib}(\text{RotNib}(w_1))$
- ❑ **RotNib()** is “rotate the nibbles”, which is equivalent to swapping the nibbles , **Rcon** is a round constant
- ❑ **SubNib()** is “apply S-Box substitution on nibbles using encryption S-Box”
- ❑ $\text{RotNib}(w_1) = 0101\ 1111$
- ❑ $\text{SubNib}(0101\ 1111) = 0001\ 0111$
- ❑ $\text{Rcon}(1) = 10000000$

S-Box		<i>j</i>			
		00	01	10	11
<i>i</i>	00	9	4	A	B
	01	D	1	8	5
	10	6	2	0	3
	11	C	E	F	7

S-AES Key Generation

$$\square w_0 = 0100\ 1010, w_1 = 1111\ 0101$$

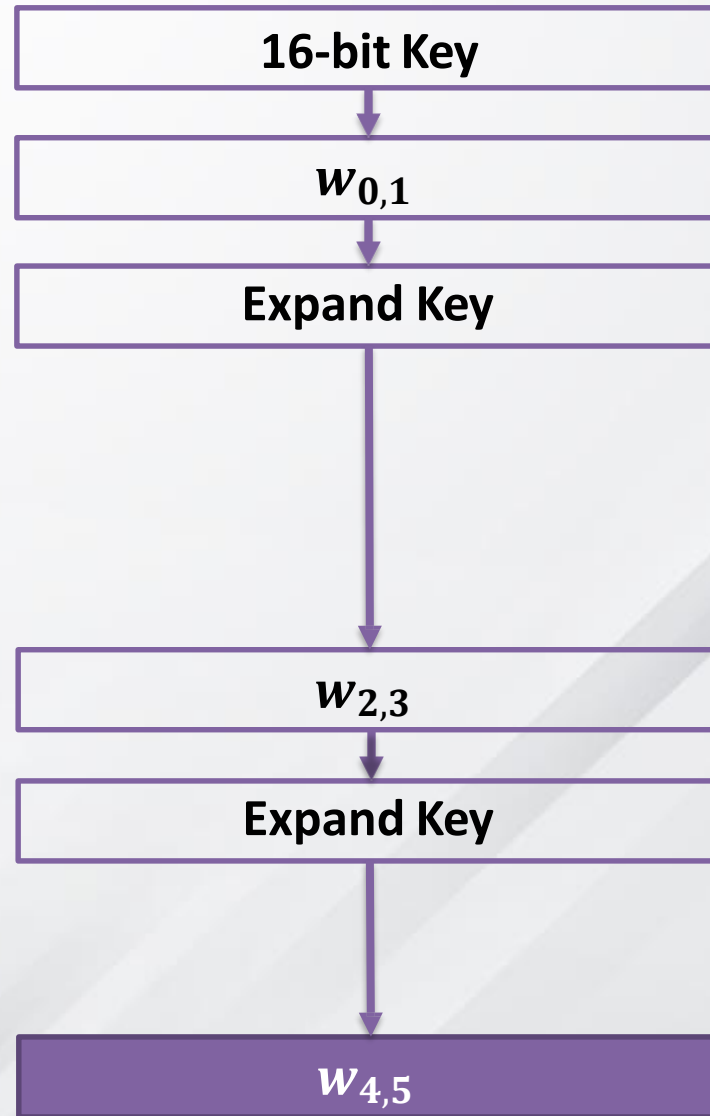
$$\begin{aligned}\square w_2 &= w_0 \oplus \text{Rcon}(1) \oplus \text{SubNib}(\text{RotNib}(w_1)) \\ &= 0100\ 1010 \oplus 1000\ 0000 \oplus 0001\ 0111 \\ &= 0100\ 1010 \oplus 1001\ 0111 = 1101\ 1101\end{aligned}$$

$$\square w_2 = 1101\ 1101$$

$$\square w_3 = w_2 \oplus w_1 = 1101\ 1101 \oplus 1111\ 0101 = 0010\ 1000$$

$$\square w_3 = 0010\ 1000$$

-AES Key Generation



-AES Key Generation

$$\square w2 = 1101\ 1101, w3 = 0010\ 1000$$

$$\square w4 = w2 \oplus \text{Rcon}(2) \oplus \text{SubNib}(\text{RotNib}(w3))$$

$$\bullet = 1101\ 1101 \oplus 0011\ 0000 \oplus \text{SubNib}(1000\ 0010)$$

$$\bullet = 1101\ 1101 \oplus 0011\ 0000 \oplus 0110\ 1010$$

$$\bullet = 1101\ 1101 \oplus 0101\ 1010$$

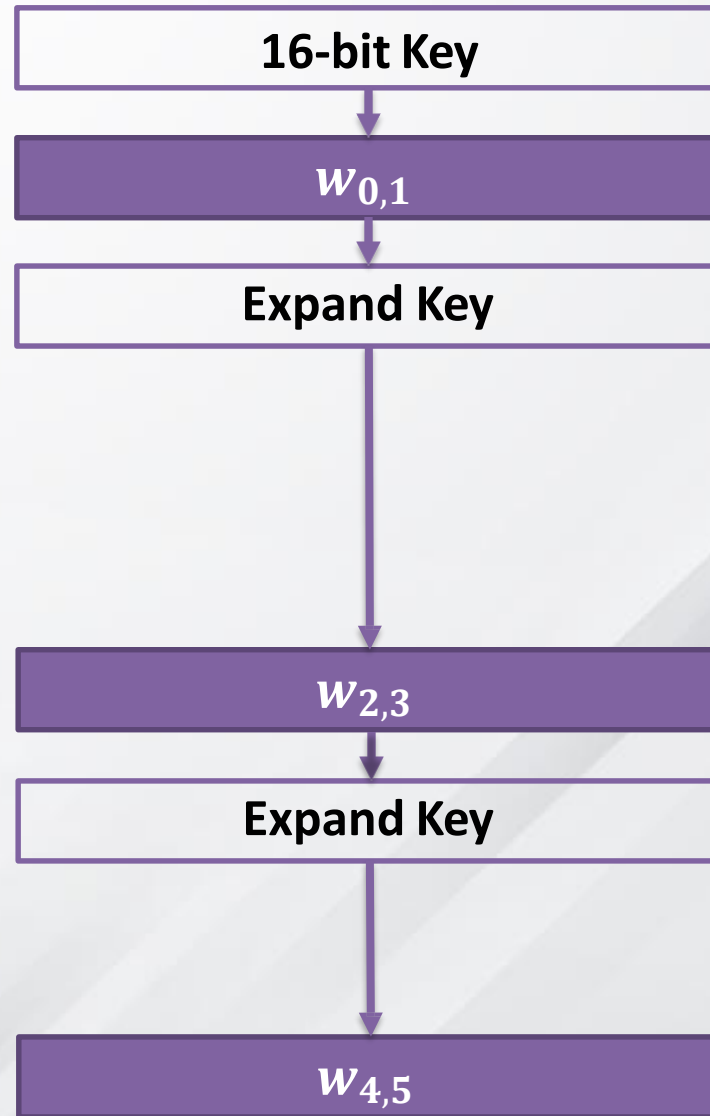
$$\square w5 = w4 \oplus w3$$

$$= 1000\ 0111 \oplus 0010\ 1000$$

$$= 1010\ 1111$$

S-Box		<i>j</i>			
		00	01	10	11
<i>i</i>	00	9	4	A	B
	01	D	1	8	5
	10	6	2	0	3
	11	C	E	F	7

-AES Key Generation



S-AES Key Generation

□ Key

□ Key0 = w_0w_1

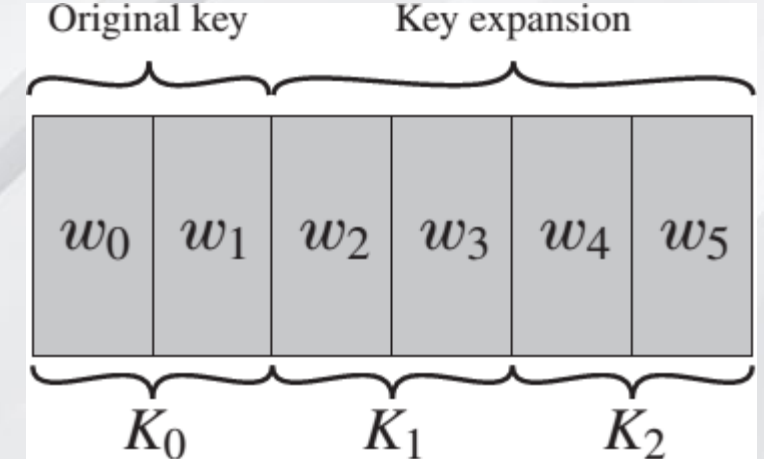
= 0100 1010 1111 0101

□ Key1 = w_2w_3

= 1101 1101 0010 1000

□ Key2 = w_4w_5

= 1000 0111 1010 1111





مجموعة جامعة
الجميع الذكية
للتعليم القابضة

MIDOCEAN
UNIVERSITY



Thank You

