Contemporary Cryptography：Principles and Practice

5. the Advanced Encryption Standard (AES)

Fuyou Miao，Wenchao Huang

Web：http://staff.ustc.edu.cn/~huangwc/crypto.html

Email: mfy@ustc.edu.cn，huangwc@ustc.edu.cn

huangwc@ustc.edu.cn

Key Points

- **AES** is a **block cipher** intended to **replace DES** for commercial applications. It uses a **128-bit** block size and a key size of **128, 192, or 256 bits**.

- AES does **not** use a Feistel structure. Instead, each full round consists of **four separate functions**: byte substitution, permutation, arithmetic operations over a finite field, and XOR with a key.

huangwc@ustc.edu.cn

Contents

huangwc@ustc.edu.cn

1. Introduction to AES
Motivation

- Security of DES
  - Brute-force attacks based on 56-bits keys
- A Solution: **3DES**, with 168-bit keys
  - Pros: robust against brute-force attacks
  - Cons: Low computational efficiency
- Another solution?
  - **AES**

huangwc@ustc.edu.cn

1. Introduction to AES History

- AES is a subset of the Rijndael block cipher developed by two Belgian cryptographers
  - Vincent Rijmen and Joan Daemen, who submitted a proposal to NIST during the AES selection process.
- AES has been adopted by the U.S. government.
- The Advanced Encryption Standard (AES) is defined in each of:
  - FIPS PUB 197: Advanced Encryption Standard (AES)
  - ISO/IEC 18033-3: Block ciphers

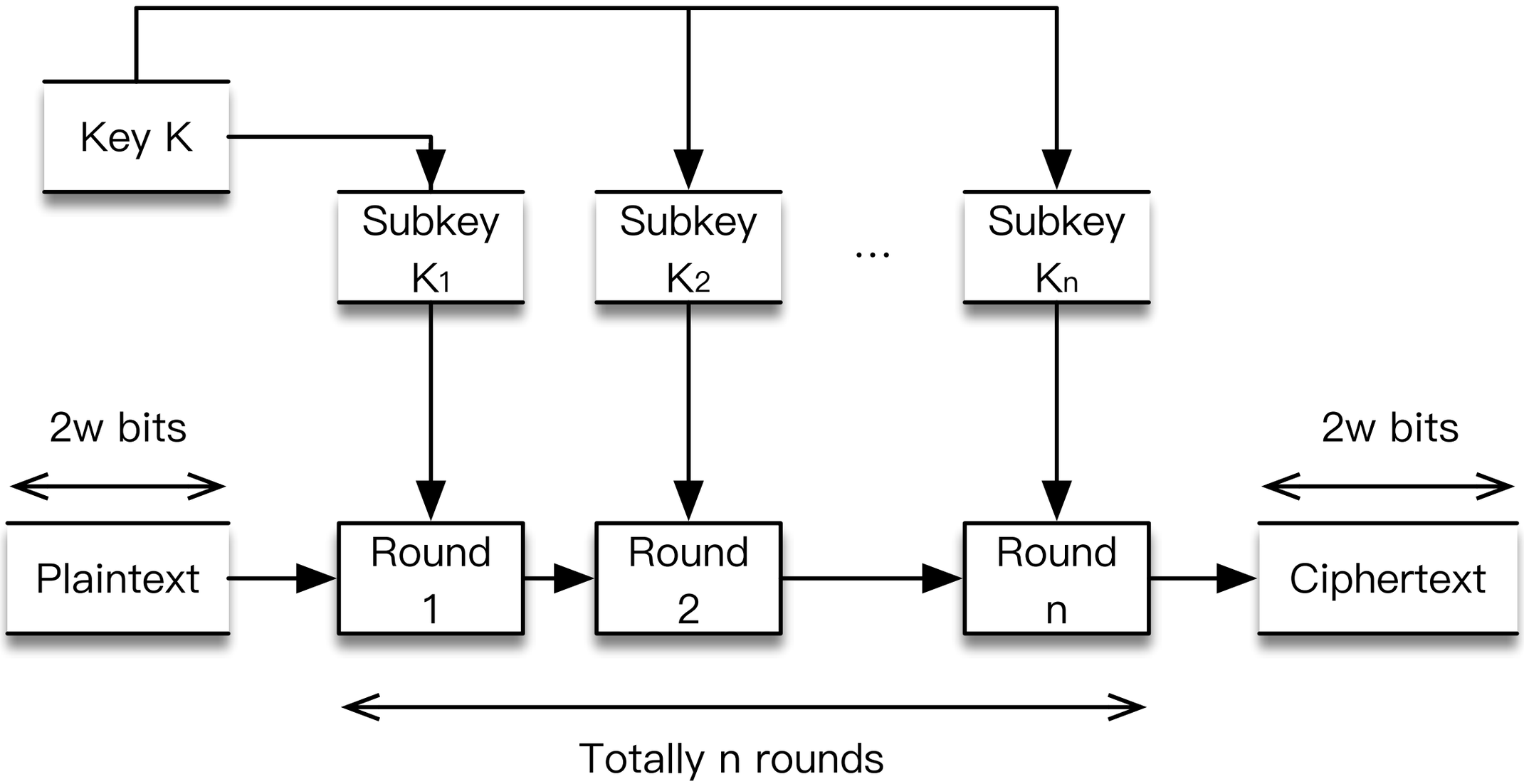
huangwc@ustc.edu.cn

2 The AES Structure Recall

- **Product cipher** (**乘积密码**):
  - *Definition*: the execution of two or more simple ciphers in sequence
  - *Feature*: the final result or product is cryptographically stronger than any of the component ciphers
- **Substitution:** Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements
- **Permutations:** A sequence of plaintext elements is replaced by a permutation of that sequence.
  - That is, no elements are added or deleted or replaced in the sequence

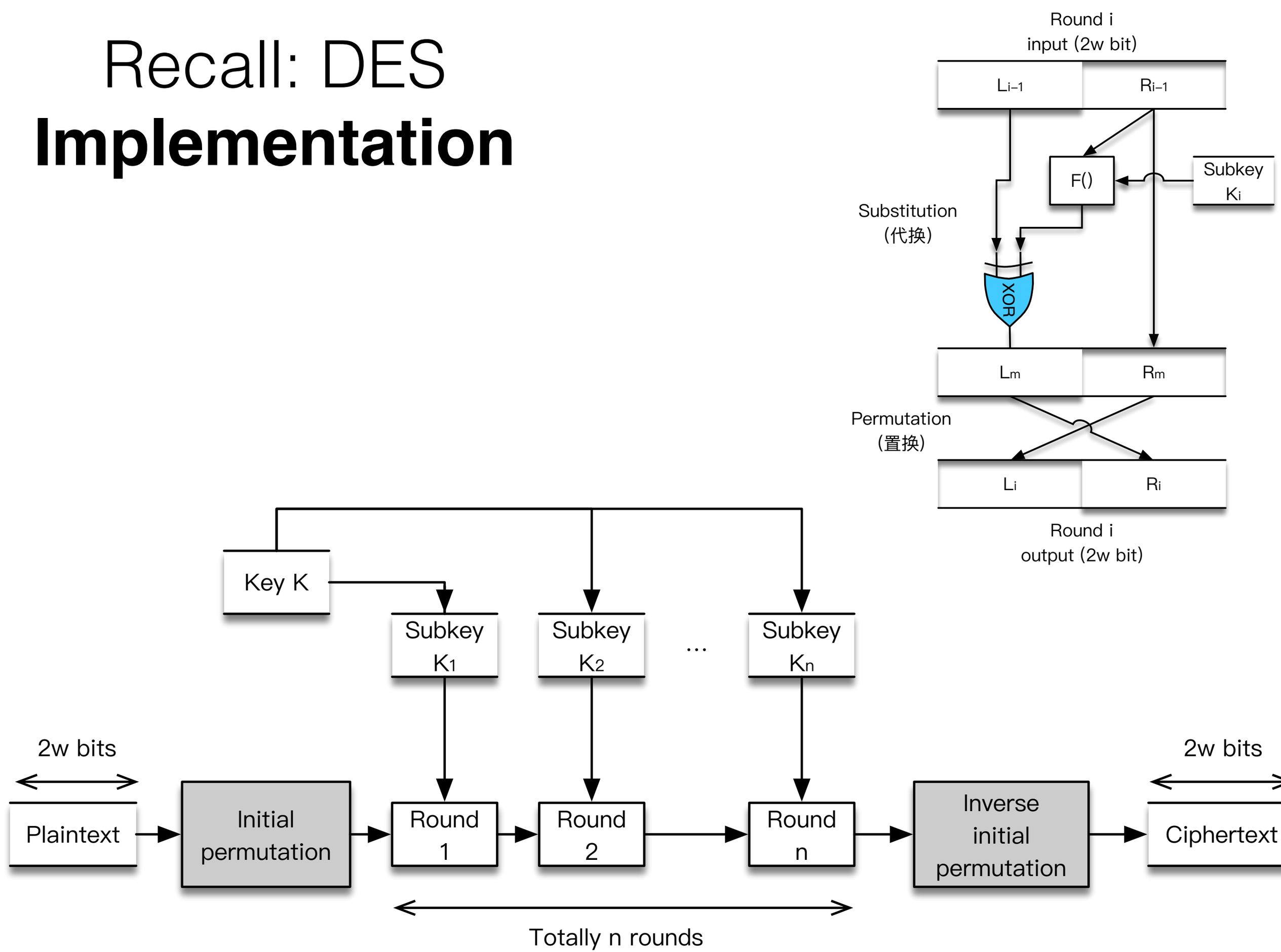
huangwc@ustc.edu.cn

2 The AES Structure Recall

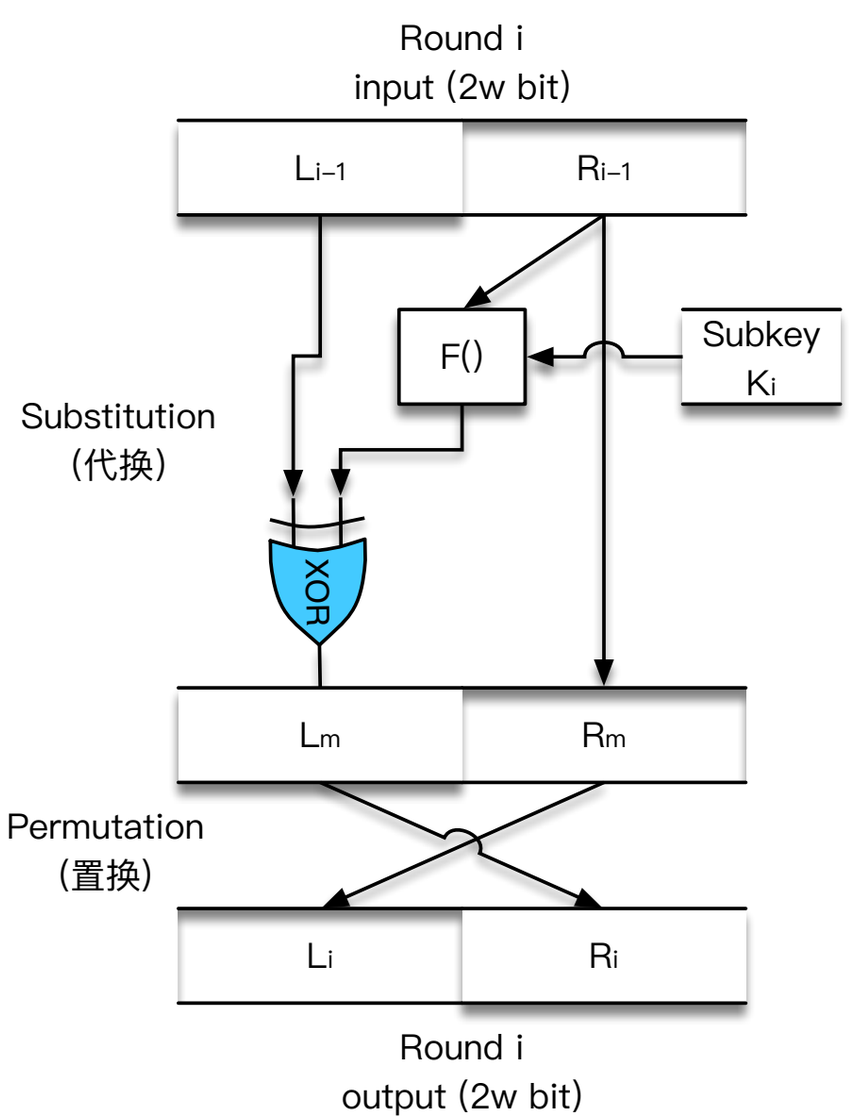


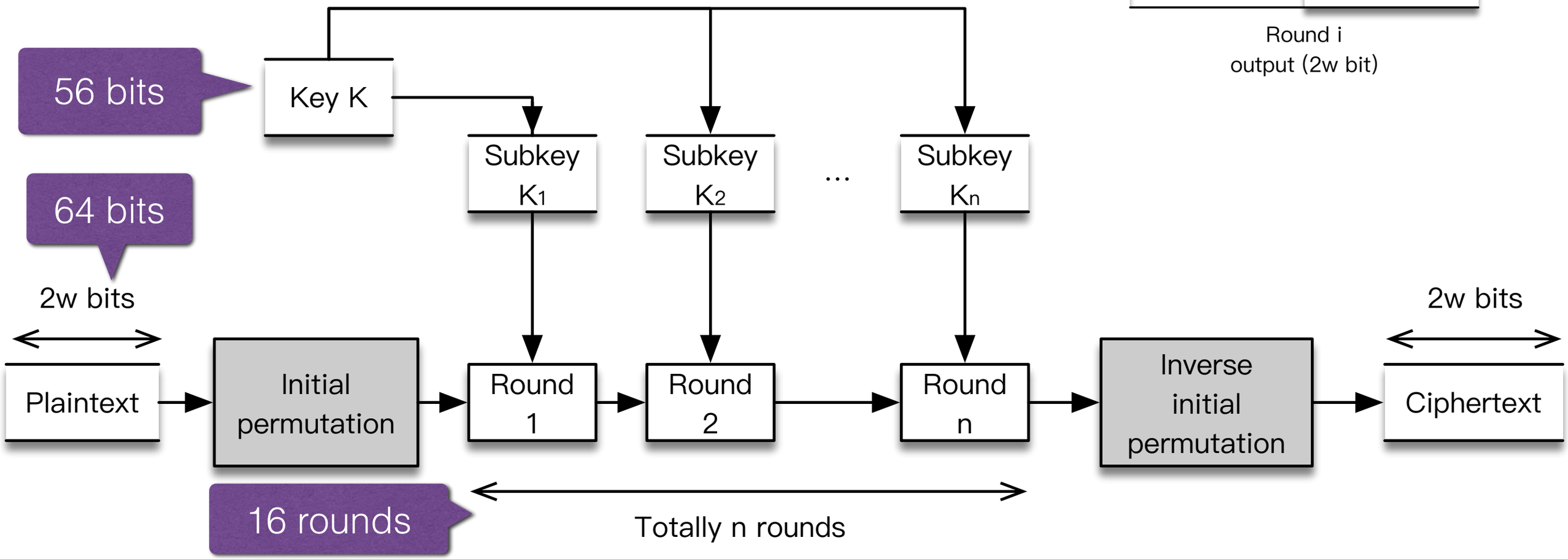
huangwc@ustc.edu.cn

Recall: DES
Implementation



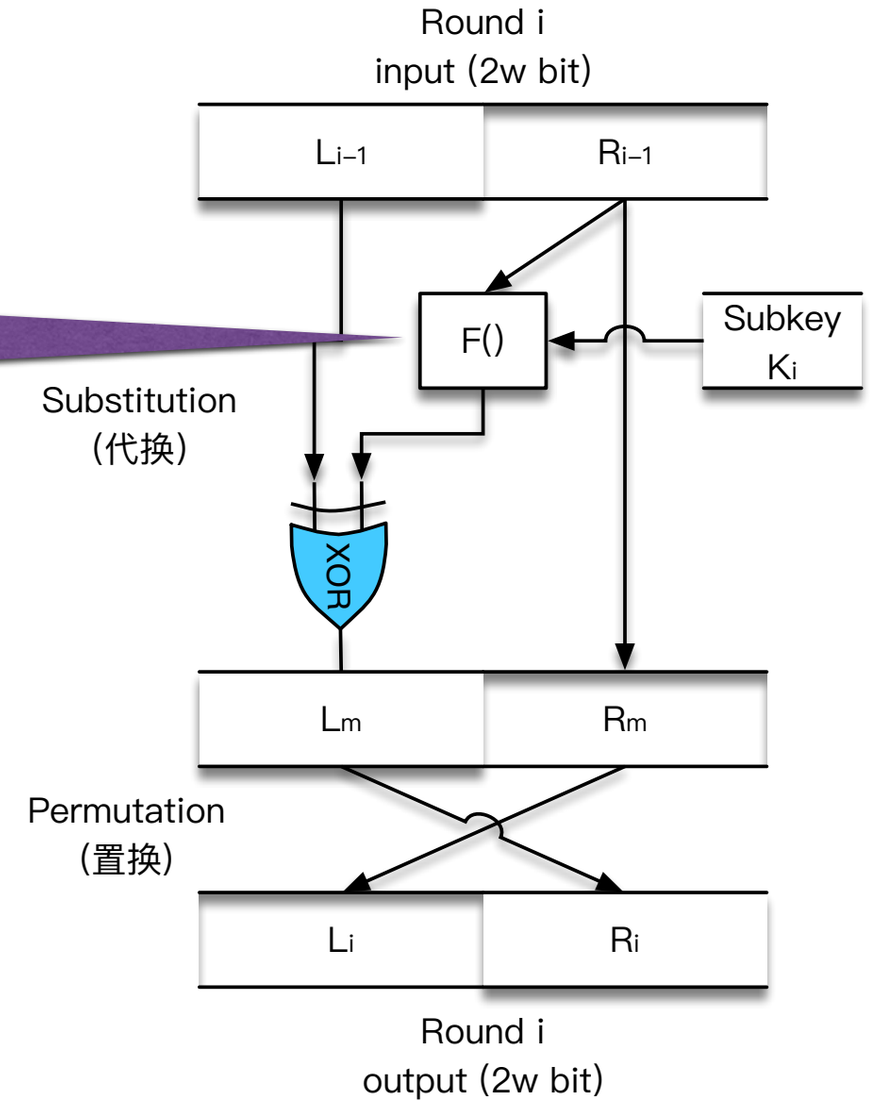
Recall: DES
Implementation





Recall: DES
Implementation

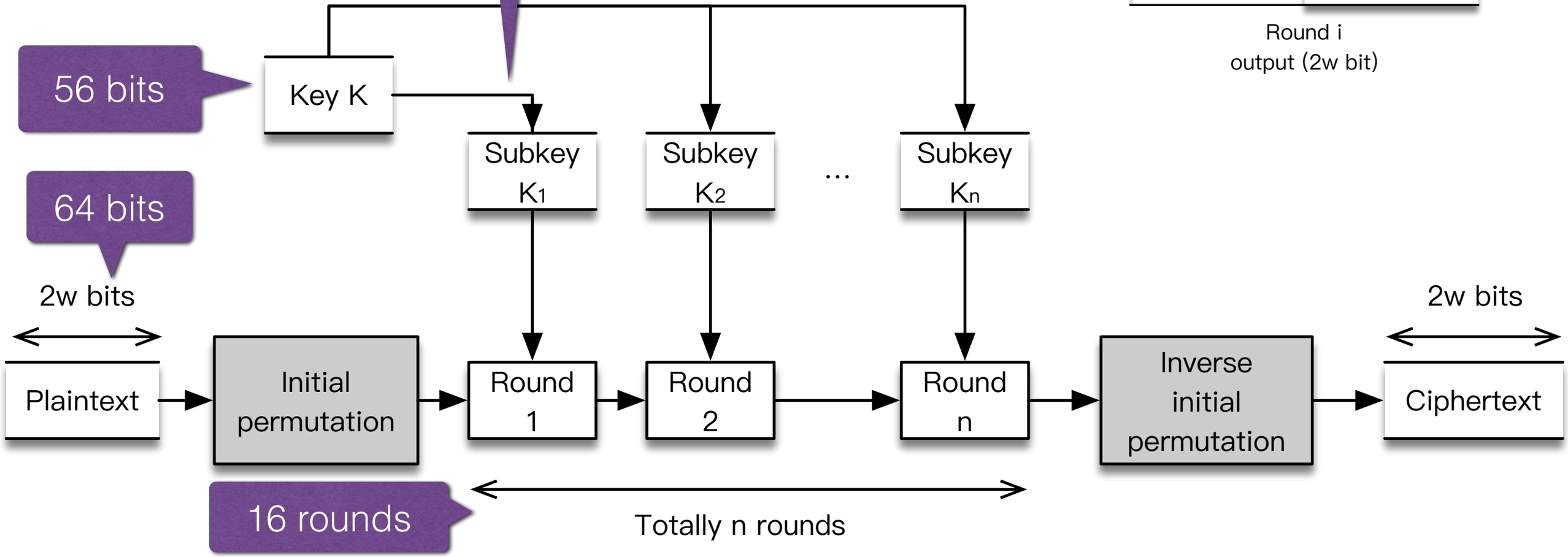
Round Function F
(Next Page)



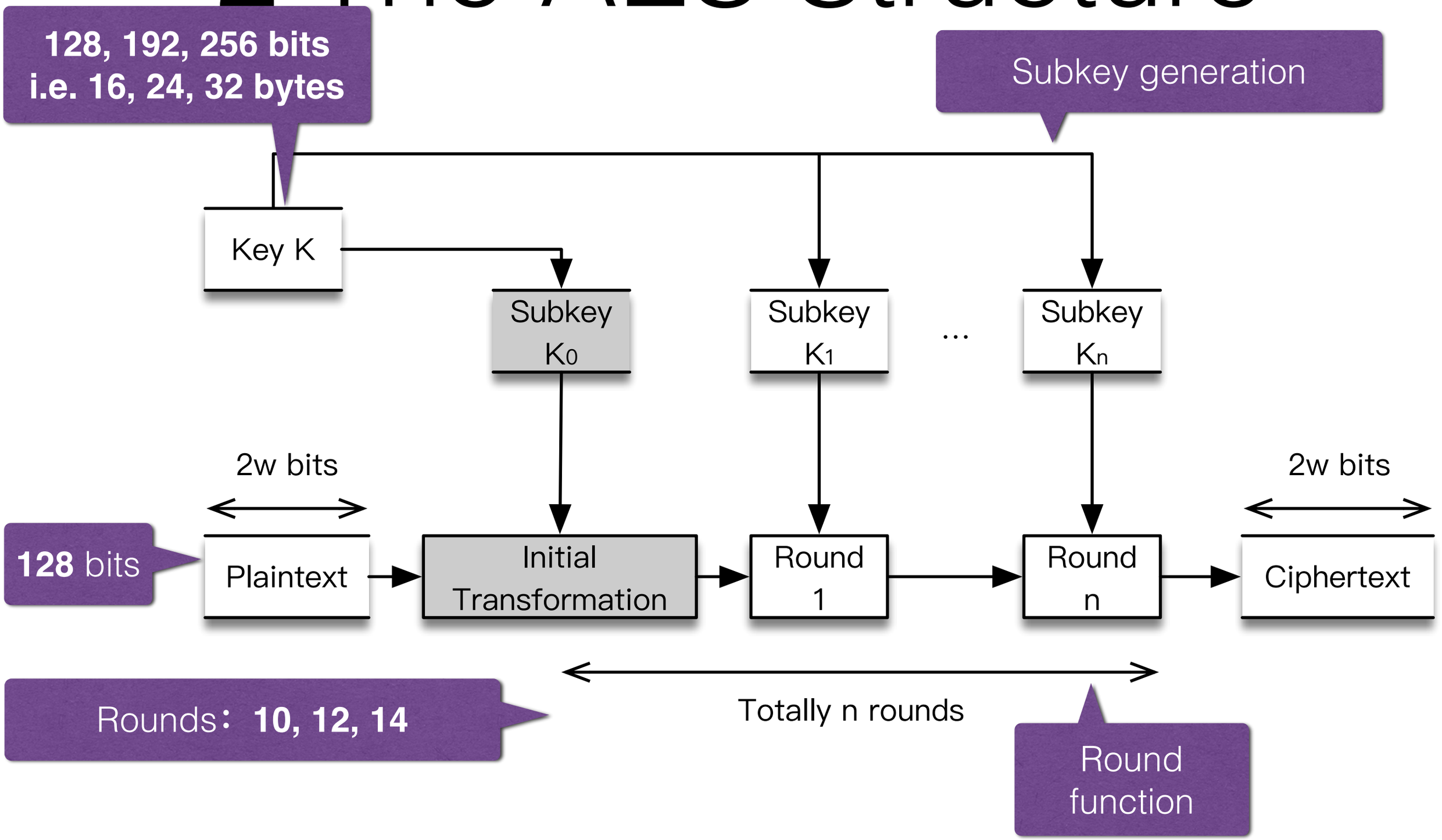
Subkey generation algorithm
(Next Page)

56 bits

64 bits

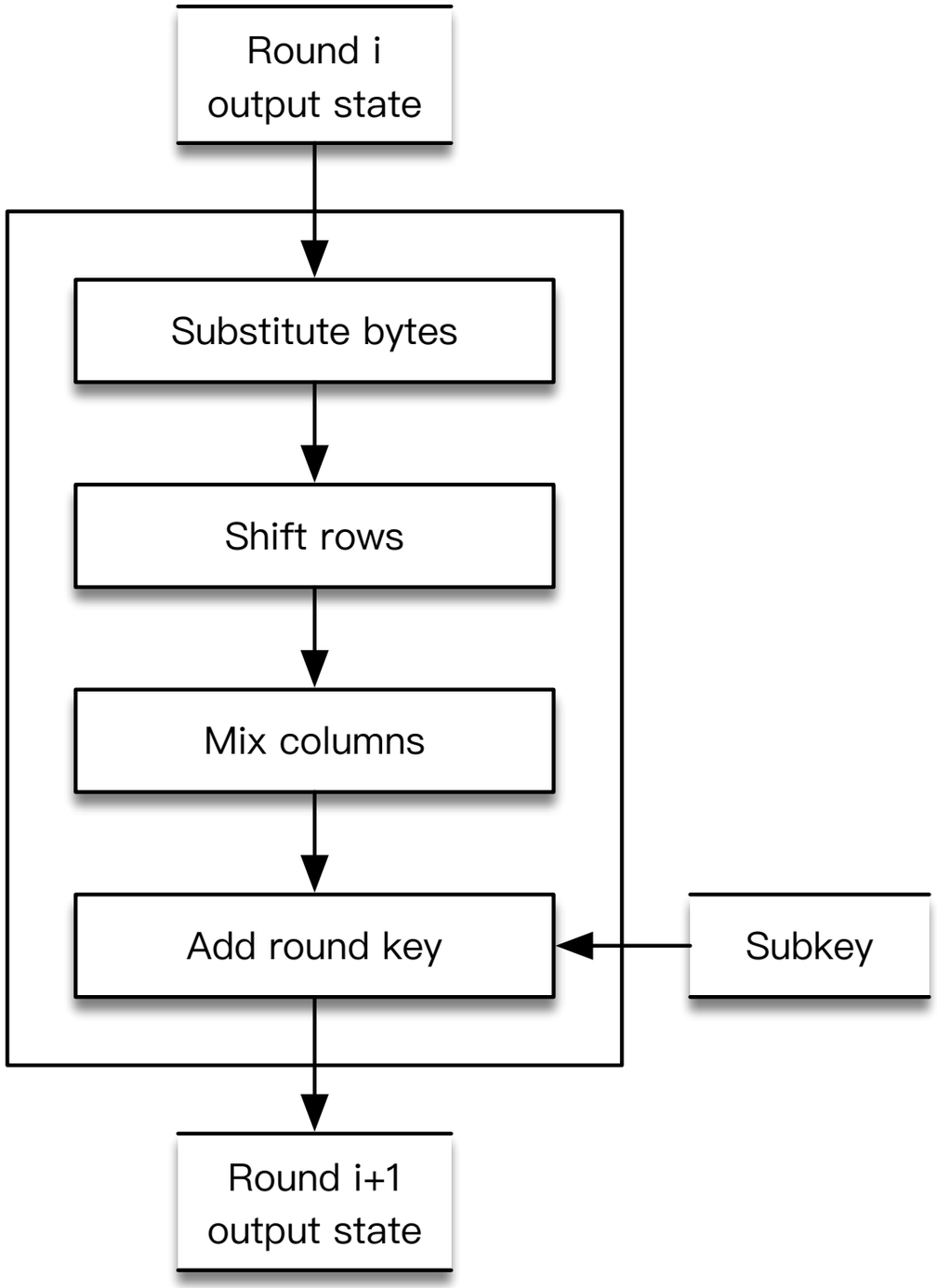


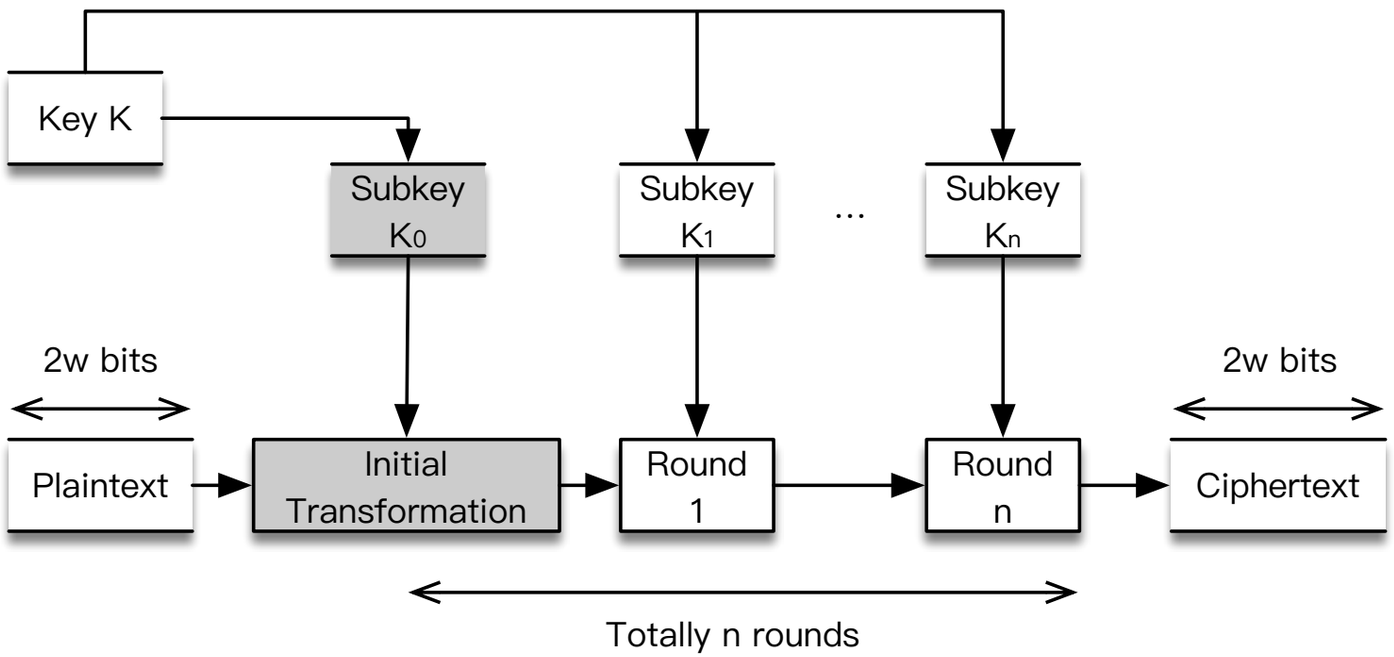
2 The AES Structure



huangwc@ustc.edu.cn

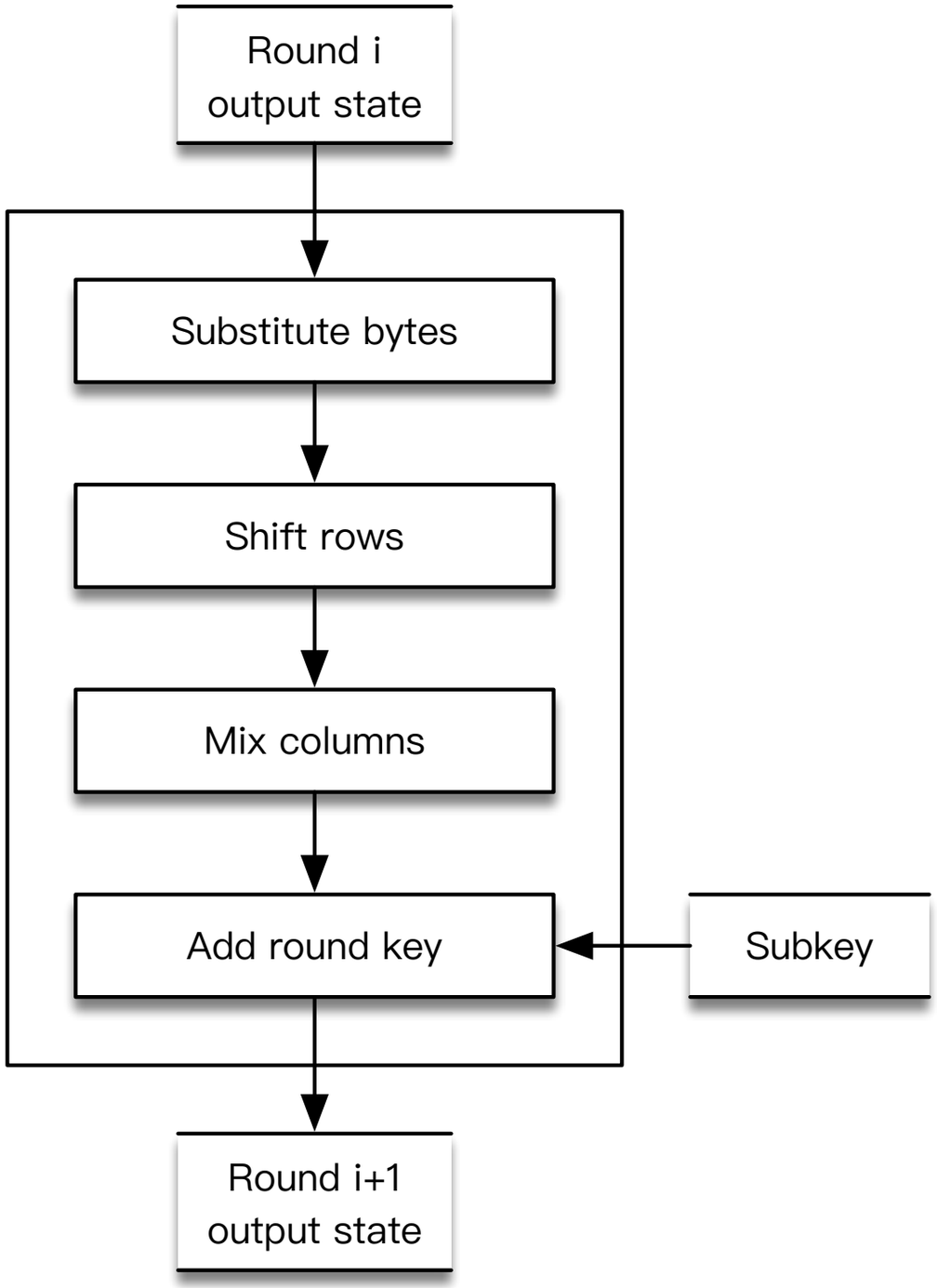
2 The AES Structure



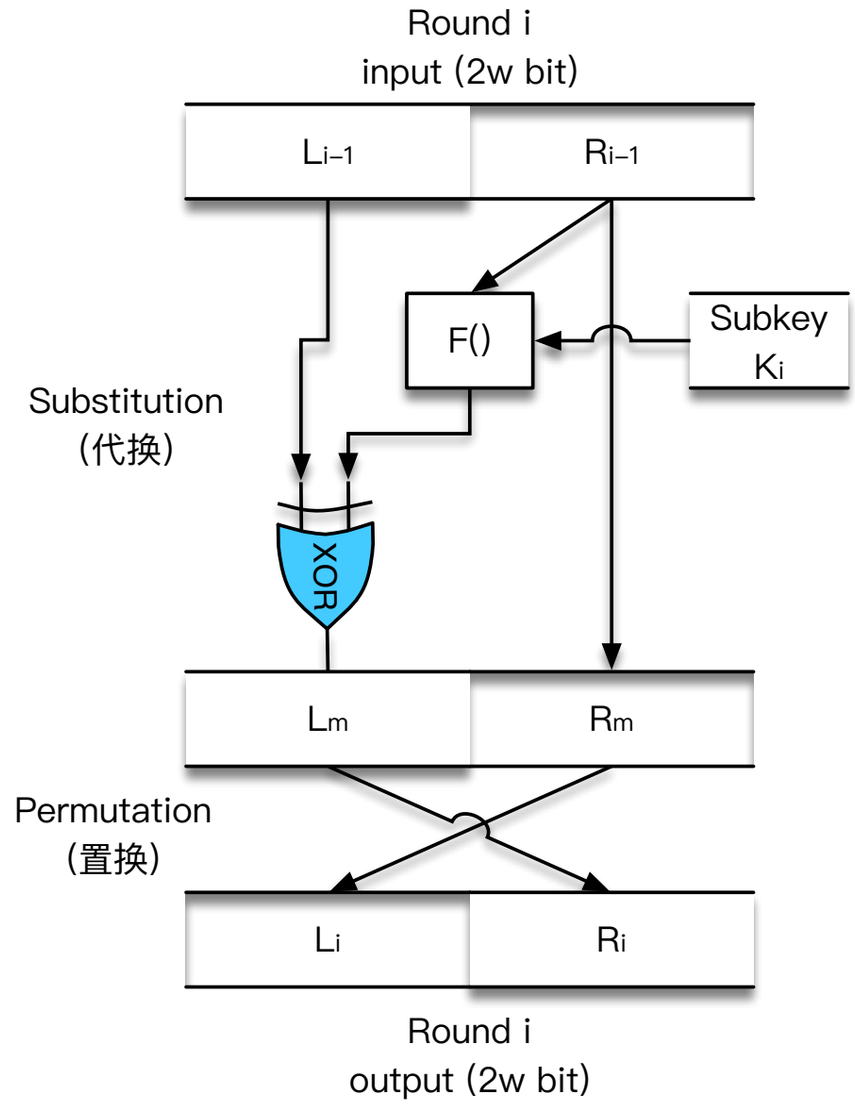


huangwc@ustc.edu.cn

3 AES Round Function

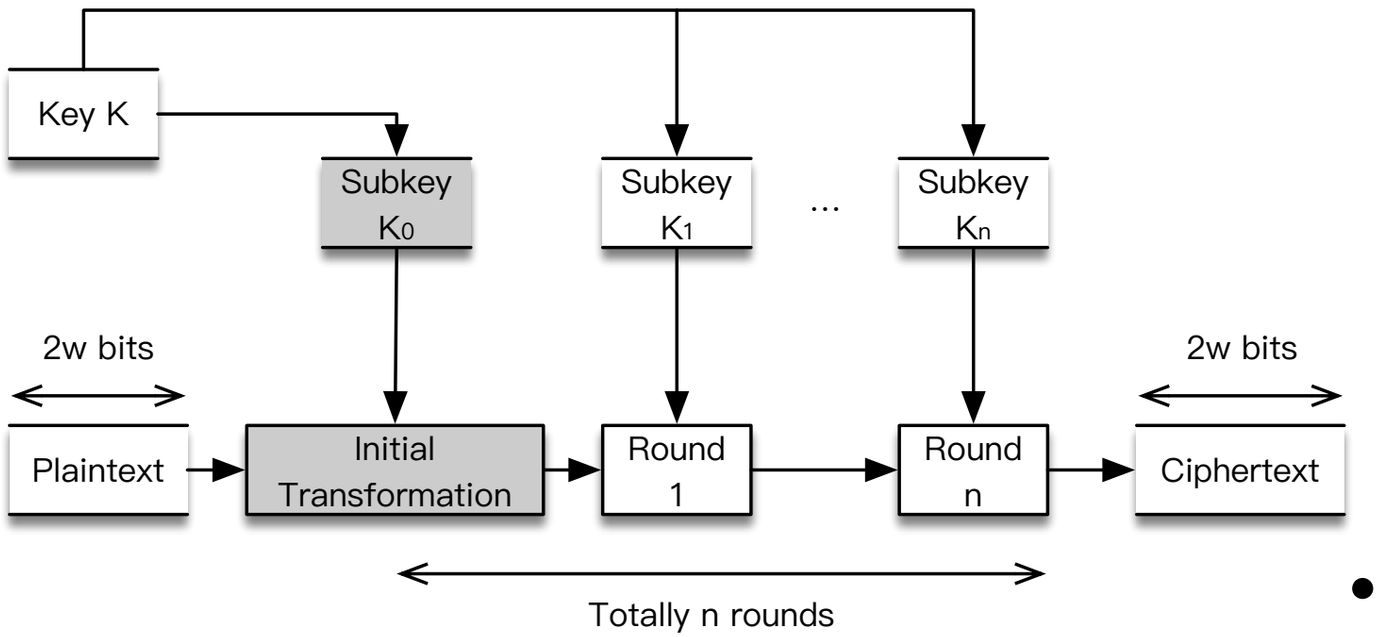


- **No** Feistel structure



huangwc@ustc.edu.cn

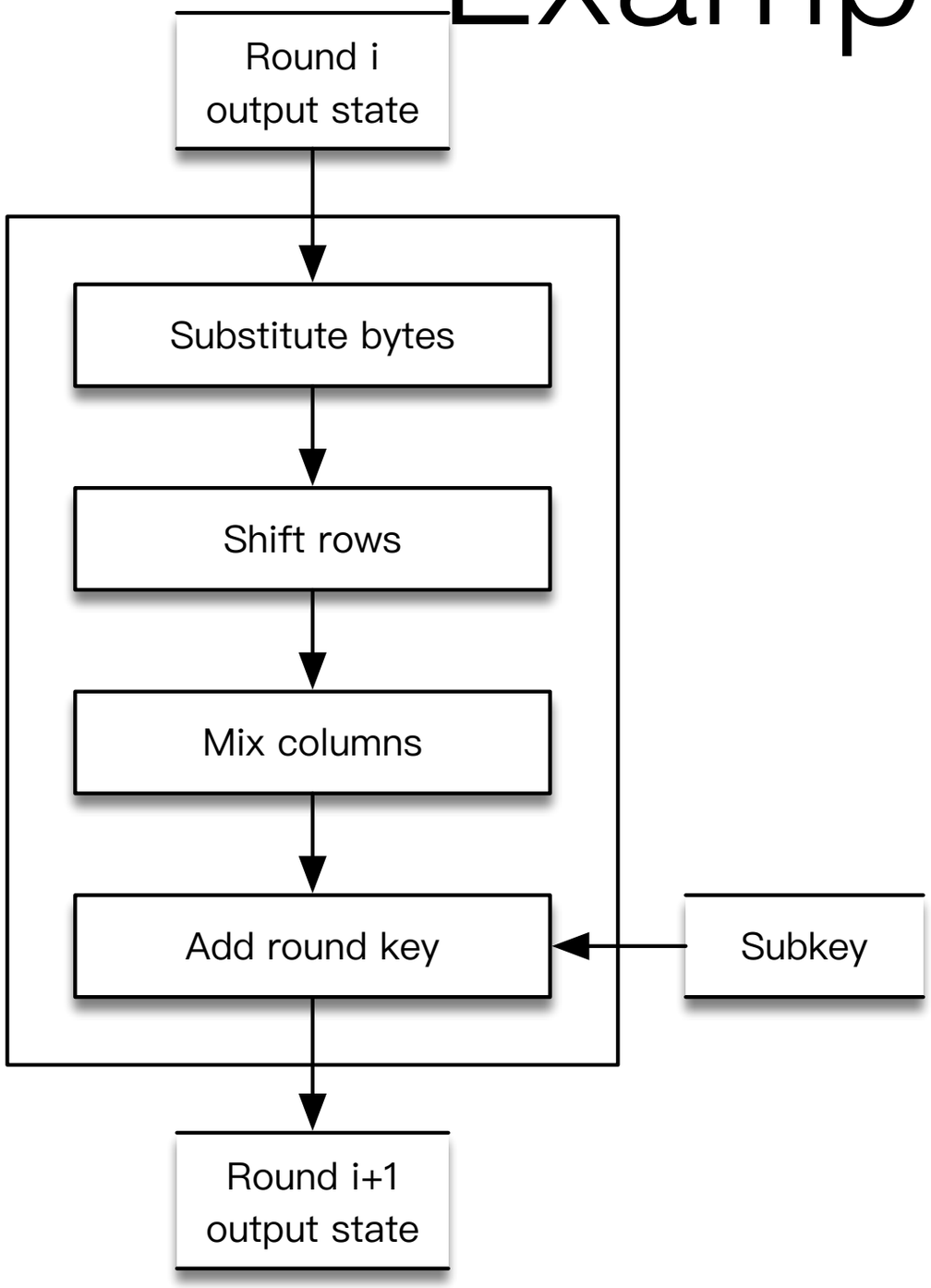
3 AES Round Function Example: AES-**128**



- The cipher consists of N rounds
  - 10 rounds for a 16-byte key
  - 12 rounds for a 24-byte key
  - 14 rounds for a 32-byte key
- When N=**10**
  - 16-byte key
  - 11 subkeys are generated

huangwc@ustc.edu.cn

3 AES Round Function Example: AES-**128**



- Round function
  - Input
    - state （16 bytes)
    - subkey （16 bytes)
- Output
  - state (16 bytes)

huangwc@ustc.edu.cn

3 AES Round Function Recall

Example

The Advanced Encryption Standard (AES) uses arithmetic in the finite field GF(2^8), with the irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$. Consider the two polynomials $f(x) = x^6 + x^4 + x^2 + x + 1$ and $g(x) = x^7 + x + 1$. Then

$$\begin{aligned} f(x) \times g(x) &= x^{13} + x^{11} + x^9 + x^8 + x^7 \\ &\quad + x^7 + x^5 + x^3 + x^2 + x \\ &\quad + x^6 + x^4 + x^2 + x + 1 \\ &= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \\ &= x^7 + x^6 + 1 \end{aligned}$$

huangwc@ustc.edu.cn

3 AES Round Function GF(2^8)

- GF(2^8)
- Irreducible Function $m(x) = x^8 + x^4 + x^3 + x + 1$
- Example: $A = (a_7 a_6 \dots a_1 a_0) \quad B = (b_7 b_6 \dots b_1 b_0)$

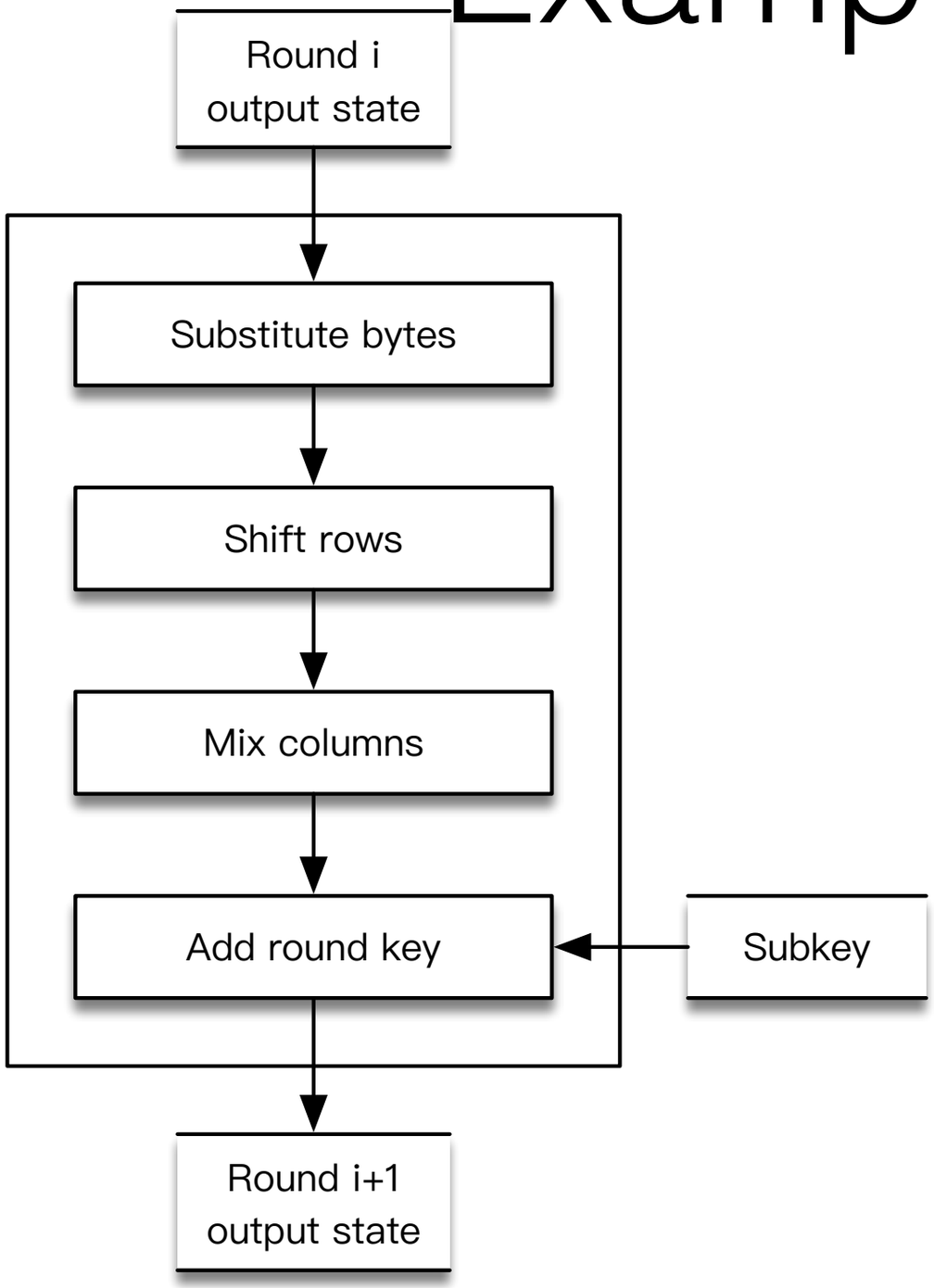
$$A + B = (c_7 c_6 \dots c_1 c_0) \qquad c_i = a_i \oplus b_i$$

$$\{02\} \cdot A = (a_6 \dots a_1 a_0 0), \text{if } a_7 = 0$$

$$\{02\} \cdot A = (a_6 \dots a_1 a_0 0) \oplus (00011011), \text{if } a_7 = 1$$

huangwc@ustc.edu.cn

3 AES Round Function Example: AES-**128**



- Observation
  - The structure is quite simple
  - The subkey is only used in 4th stage
  - Each stage is easily reversible
  - The decryption algorithm is not identical to the encryption algorithm
  - The final round of both encryption and decryption consists of only three stages

huangwc@ustc.edu.cn

State

SubBytes

State

ShiftRows

State

MixColumns

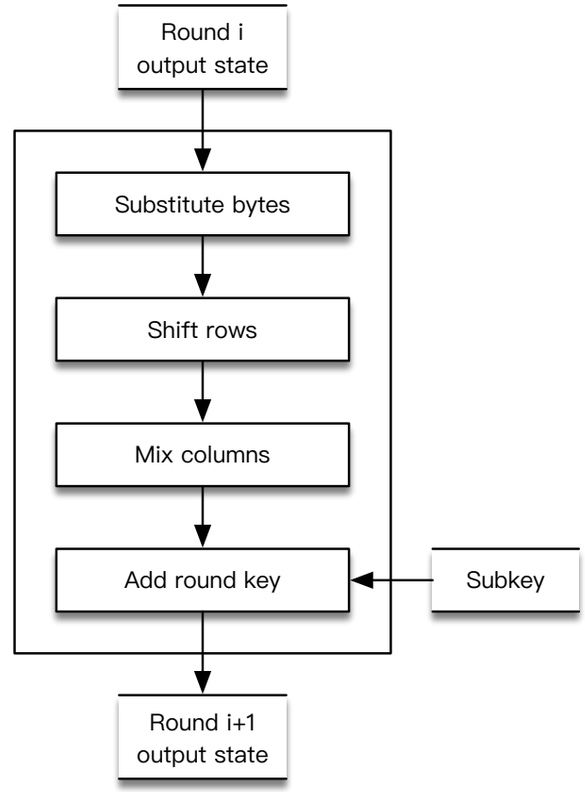
State

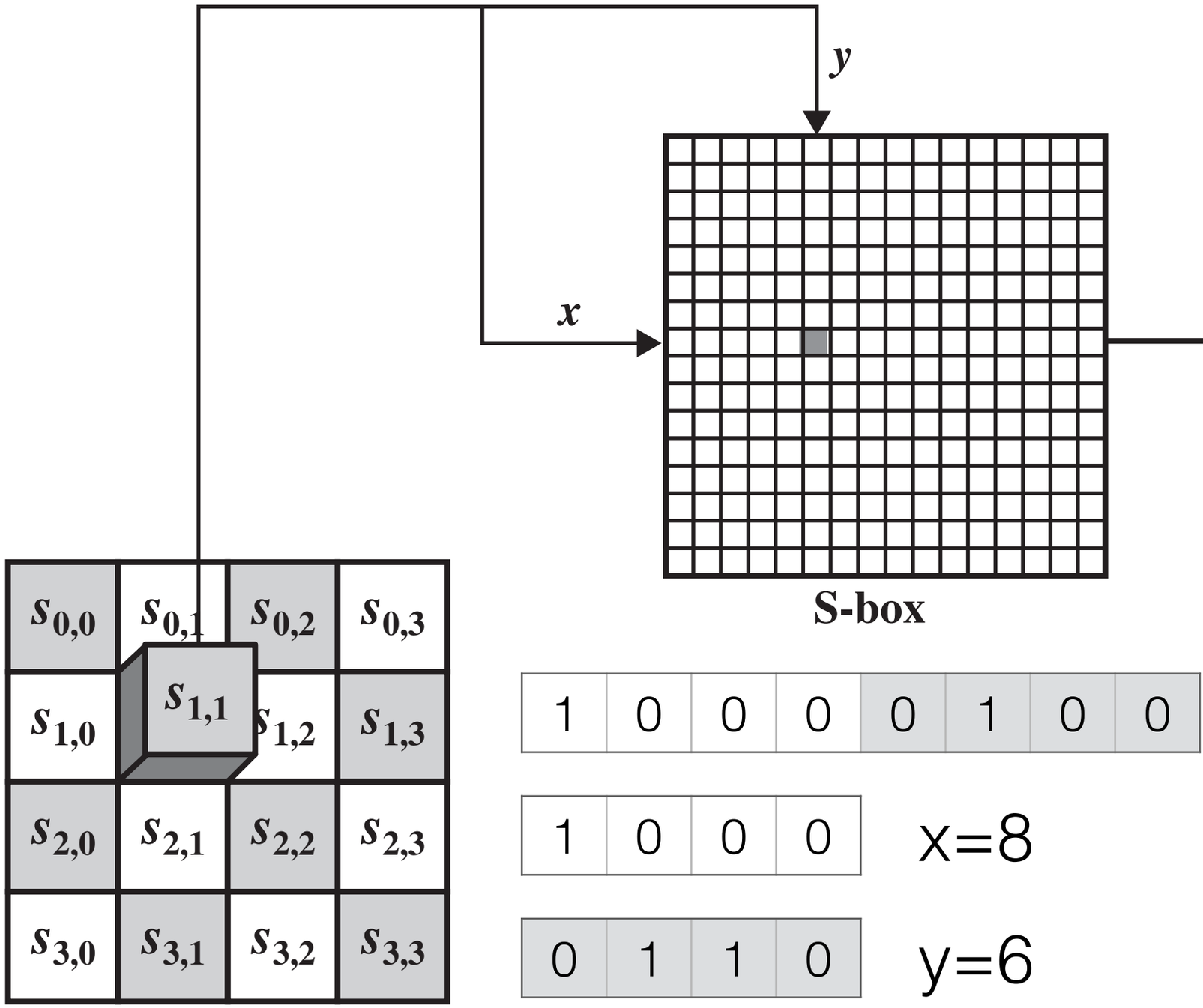
AddRoundKey

State

nuangwc@ustc.edu.cn

3 AES Round Function
Stage 1: Substitute bytes





1	0	0	0	0	1	0	0

1	0	0	0

x=8

0	1	1	0

y=6

1	0	0	0

x=8

0	1	1	0

y=6

0	1	0	0	0	1	0	0

s=44

S-box

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

0	1	0	0	0	1	0	0

s=44

1	0	0	0

x=8

0	1	1	0

y=6

Inverse S-box

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

3 AES Round Function Stage 1: Substitute bytes

- How to build S-box and Inverse S-box?

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

→

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

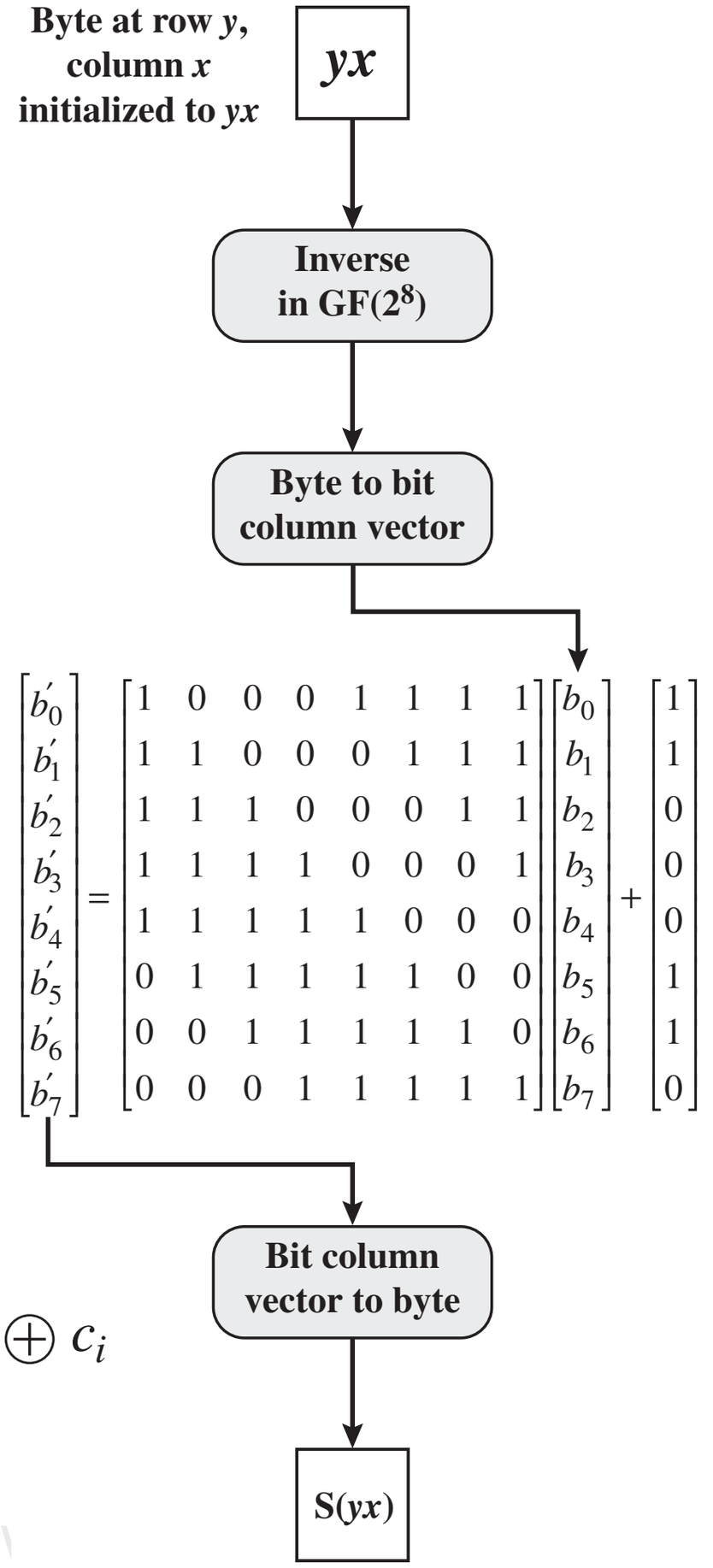
huangwc@ustc.edu.cn

3 AES Round Function
Stage 1: Substitute bytes

1. Initialize the S-box with the **byte values** row by row
   - The first row contains {00}, {01}, {02}, … , {0F}
   - The second row contains {10}, {11}, etc.
   - the value of the byte at row *y*, column *x* is {*yx*}
2. Map each byte in the S-box to its **multiplicative inverse** in the finite field GF(2^8)
   - Specially, the value {00} is mapped to itself.
3. Transformation: Consider that each byte in the S-box consists of 8 bits labeled $(b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0)$, compute b_i'

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

4. Bit column vector to byte into S(y,x)



$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

huang

3 AES Round Function Stage 1: Substitute bytes

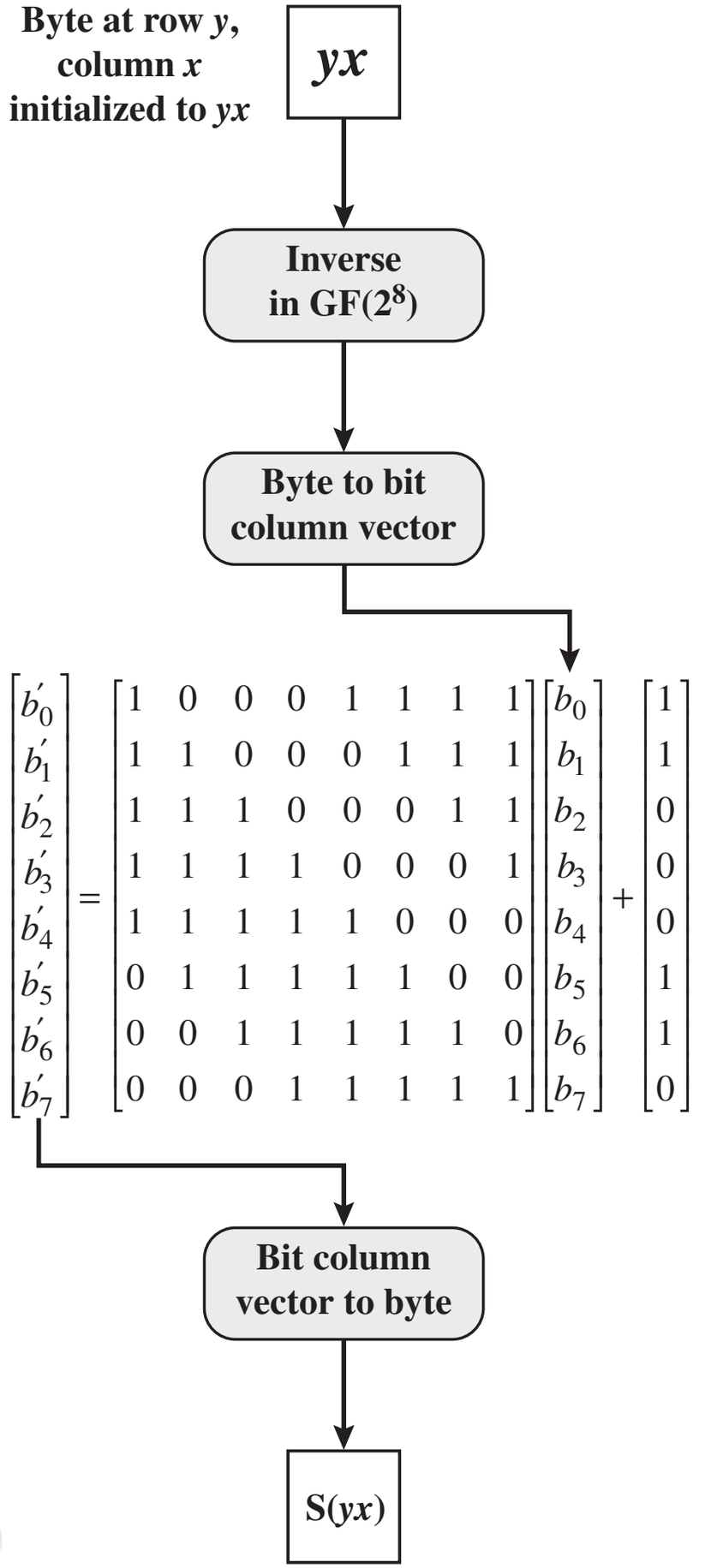
Example: input Row 9, Column 5

1. The value of byte: {95}

2. The inverse of {95} is {8A}
   ={10001010}

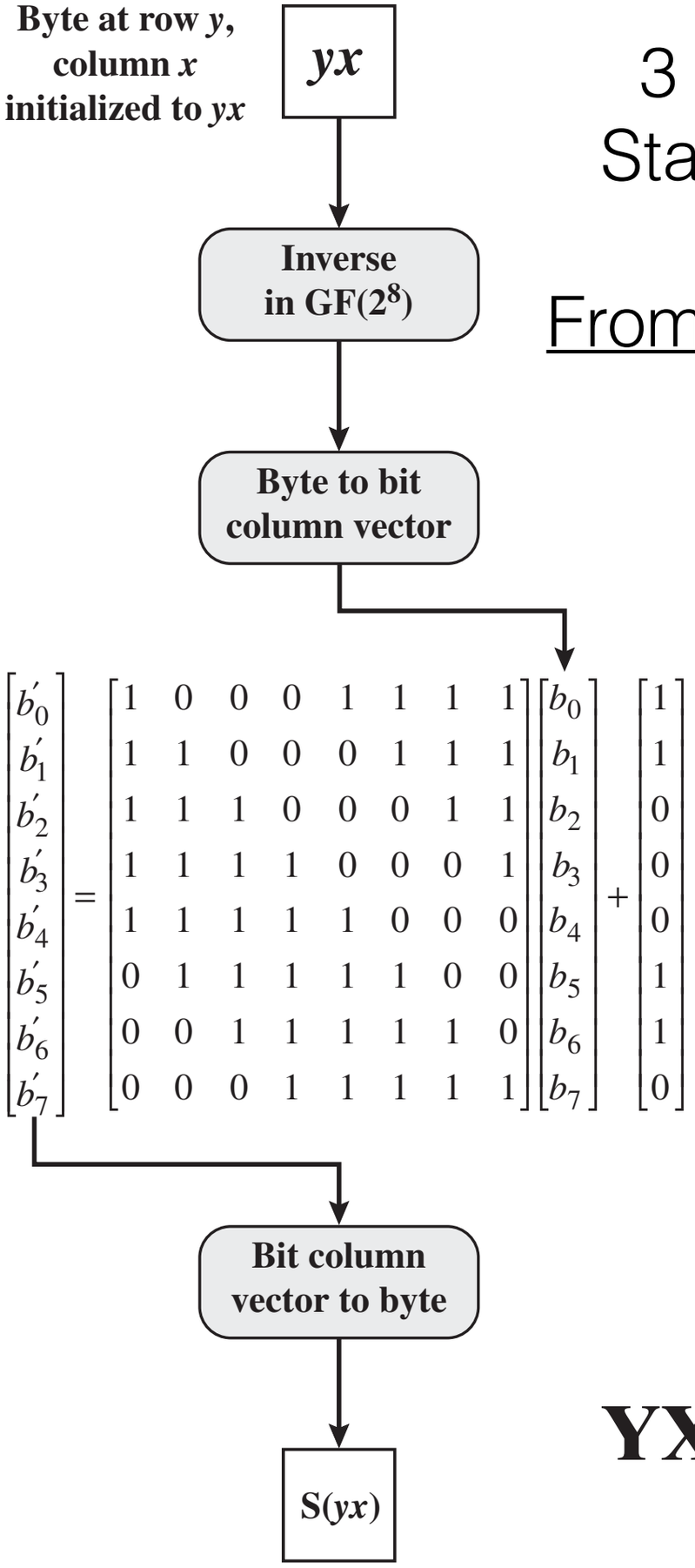
3. M*[0 1 0 1 0 0 0 1]'+[1 1 0 0 0 1 1 0]'

4. S(95)={2A}



3 AES Round Function
Stage 1: Substitute bytes

From S-box to Inverse S-box



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

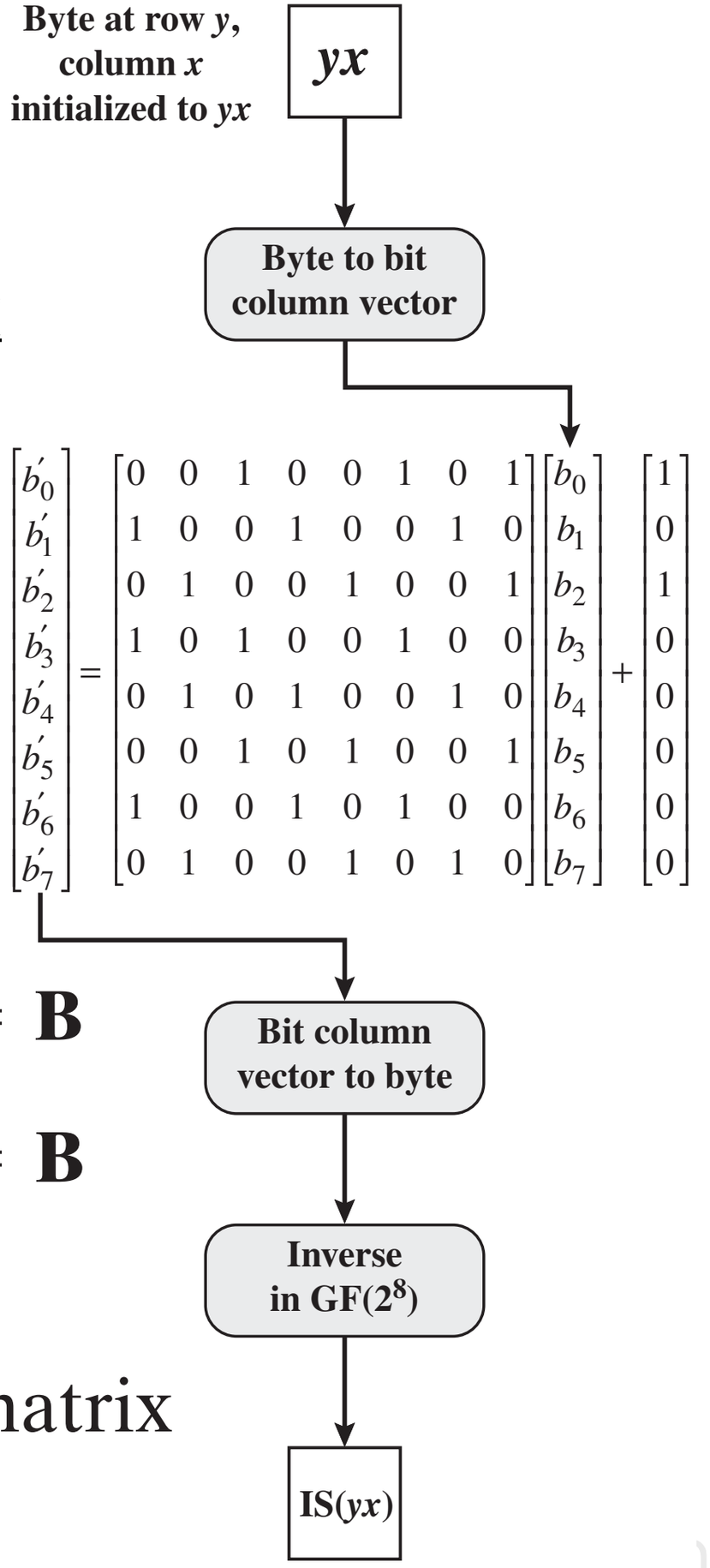
$$\mathbf{B}' = \mathbf{XB} \oplus \mathbf{C}$$

$$\mathbf{Y}(\mathbf{XB} \oplus \mathbf{C}) \oplus \mathbf{D} = \mathbf{B}$$

$$\mathbf{YXB} \oplus \mathbf{YC} \oplus \mathbf{D} = \mathbf{B}$$

$$\mathbf{YC} = \mathbf{D}$$

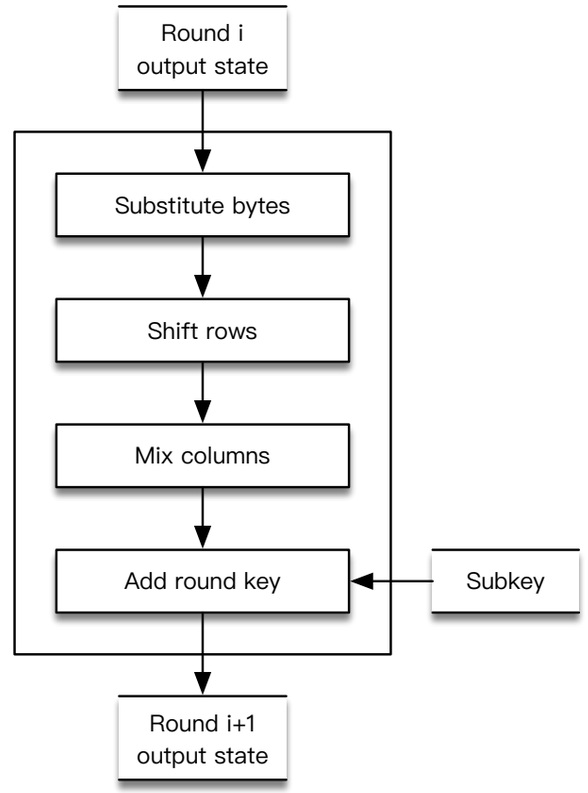
YX equals the identity matrix



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

huang...@uste.edu.cn

3 AES Round Function
Stage 1: Substitute bytes



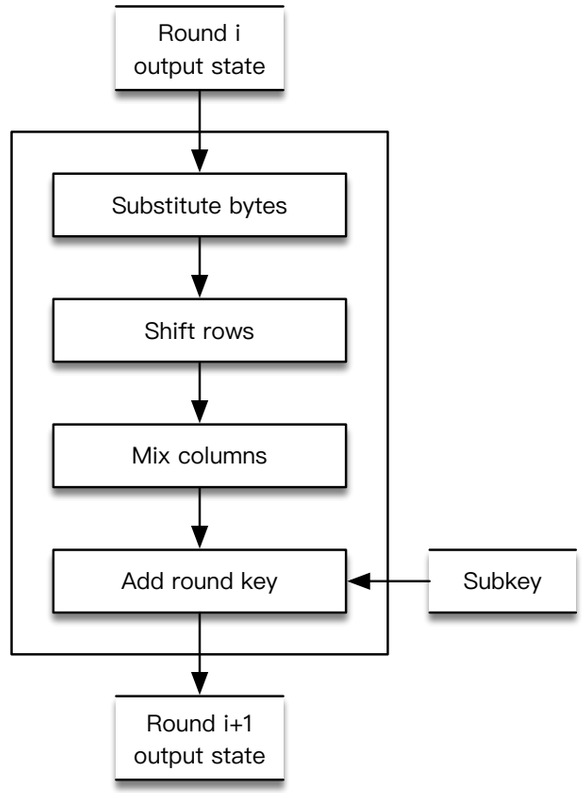
- The Rationale of Stage 1
  - Resistant to known cryptanalytic attacks.
  - low correlation between input bits and output bits

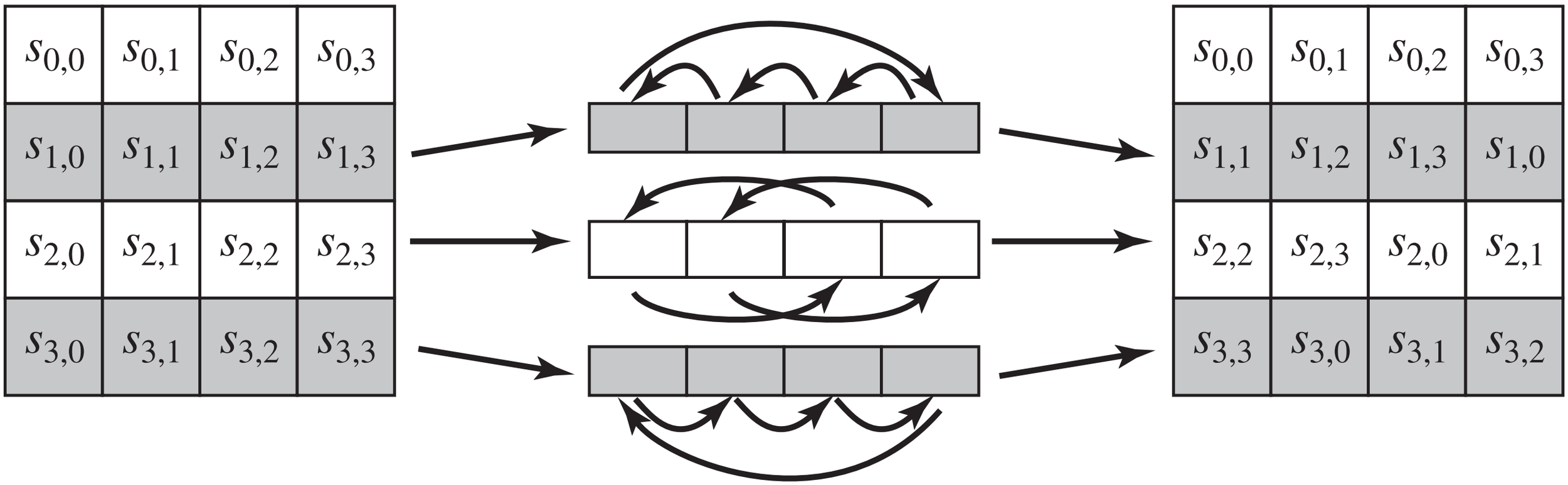
$$b_i' = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$$

    - The output is **not** a **linear** mathematical function of the input
      - The nonlinearity is due to the use of the **multiplicative inverse**
    - No fixed points, i.e., S-box(a)=a

huangwc@ustc.edu.cn

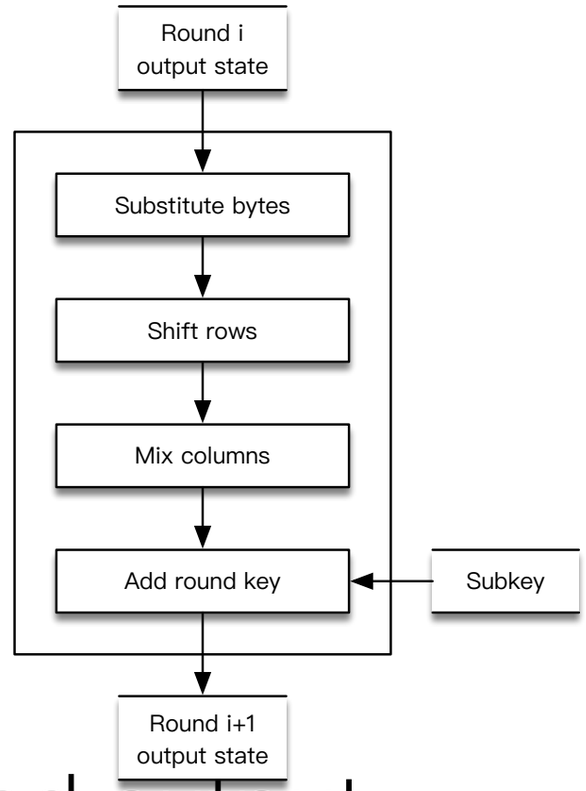
3 AES Round Function
Stage 2: Shift rows





huangwc@ustc.edu.cn

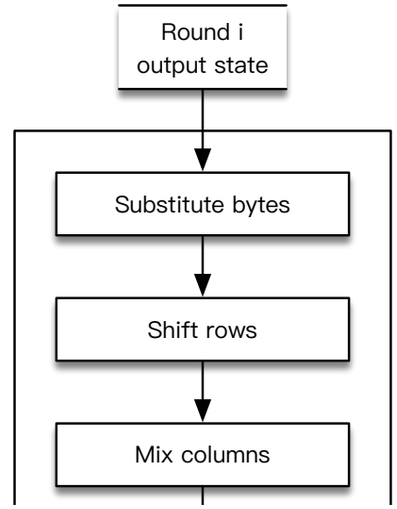
3 AES Round Function
Stage 2: Shift rows

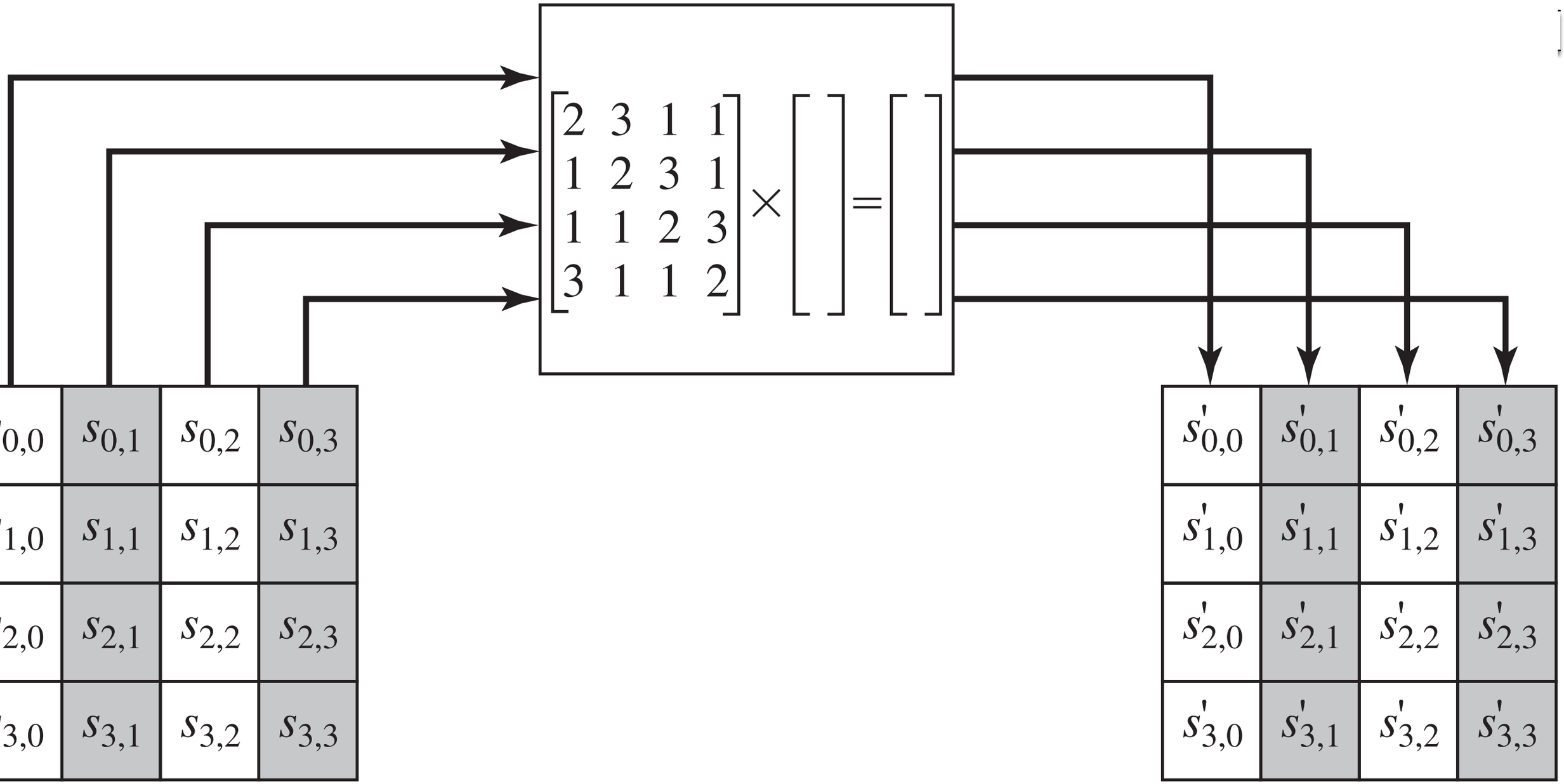


- The Rationale of Stage 2
  - The **State**, as well as the cipher input and output, is treated as an array of four **4-byte columns**.
  - A row shift moves an individual byte from one **column** to another
  - The transformation ensures that the 4 bytes of one column are **spread** out to four **different columns**

huangwc@ustc.edu.cn

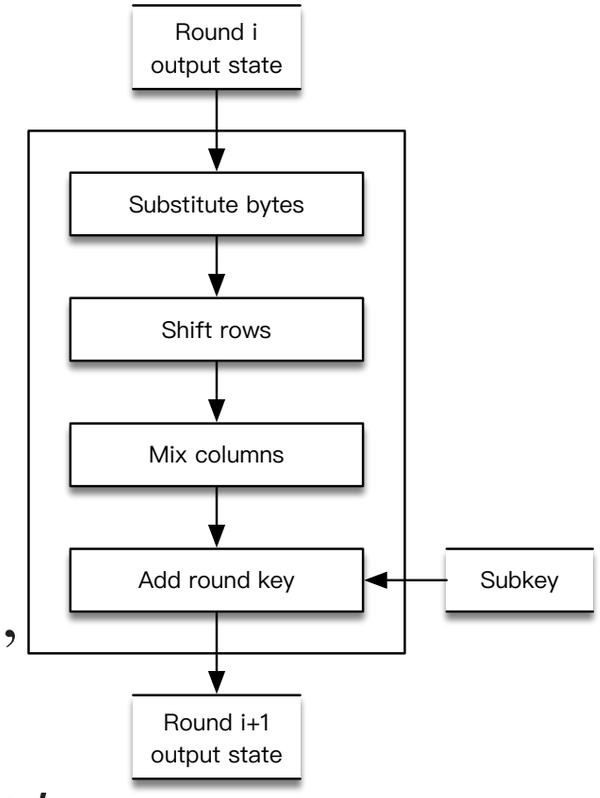
3 AES Round Function
Stage 3: Mix Columns





huangwc@ustc.edu.cn

3 AES Round Function
Stage 3: Mix Columns



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

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

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

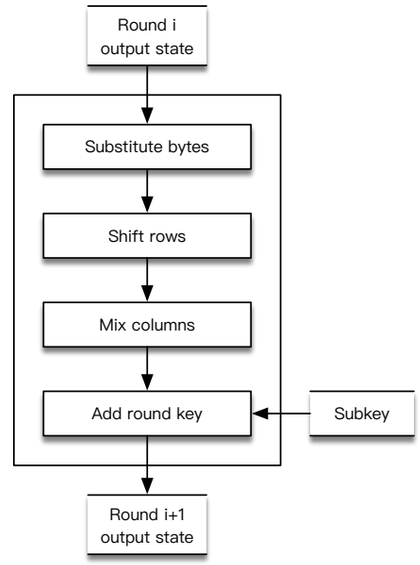
$$s'_{3,j} = (3 \bullet s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \bullet s_{3,j})$$

$$\{02\} \cdot A = (a_6 \ldots a_1 a_0 0), \text{if } a_7 = 0$$

$$\{02\} \cdot A = (a_6 \ldots a_1 a_0 0) \oplus (00011011), \text{if } a_7 = 1$$

huangwc@ustc.edu.cn

3 AES Round Function
Stage 3: Mix Columns



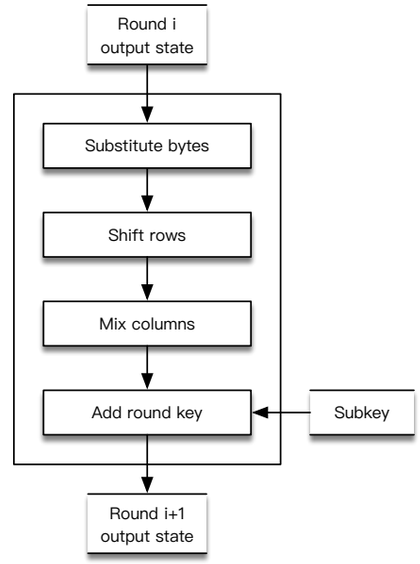
- Inverse mix column transformation

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

huangwc@ustc.edu.cn

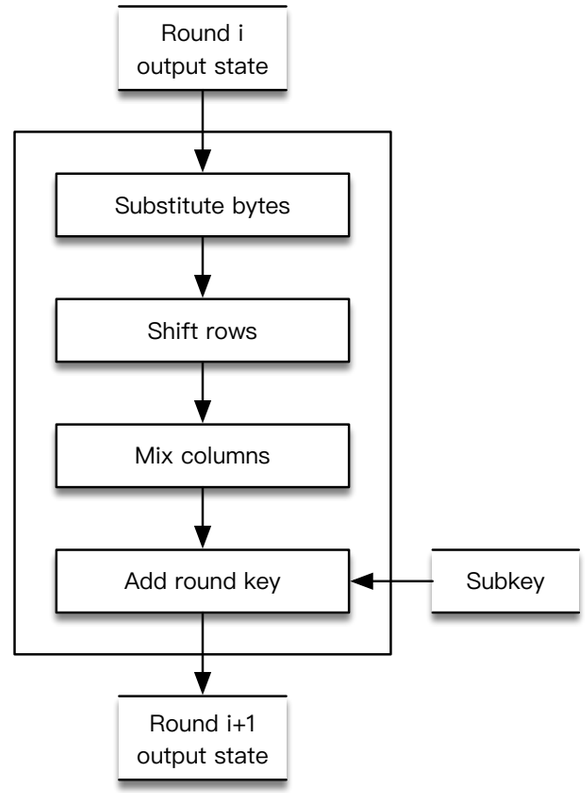
3 AES Round Function
Stage 3: Mix Columns



- The Rationale of Stage 3
  - The **coefficients** of the matrix are based on a linear code with **maximal distance** between code words
    - which ensures a good **mixing** among the **bytes of each column**
  - The **choice of coefficients** in MixColumns, which are all **{01}, { 02}, or {03}**, was influenced by implementation considerations

huangwc@ustc.edu.cn

3 AES Round Function
Stage 4: Add Round Key



- The first matrix is State
- the second matrix is the round key

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$\oplus$

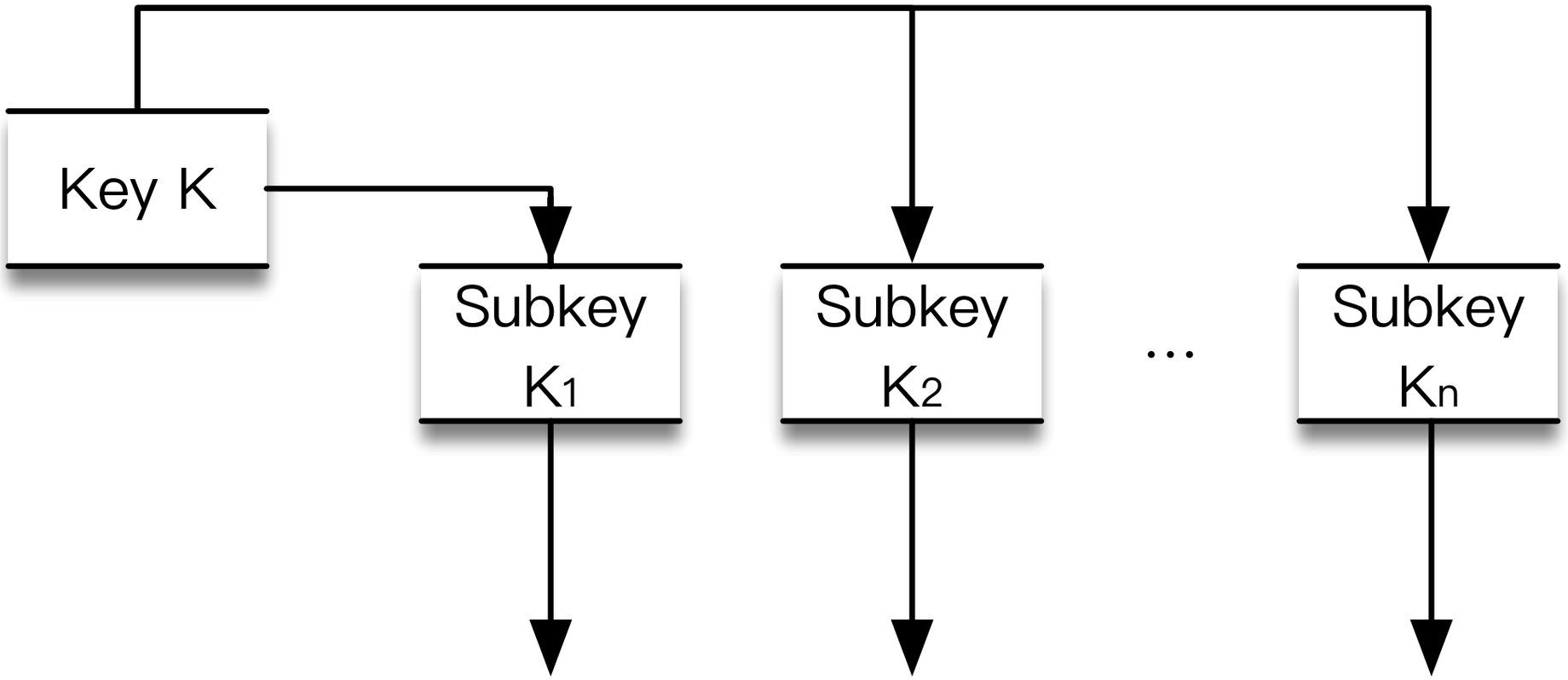
AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

$=$

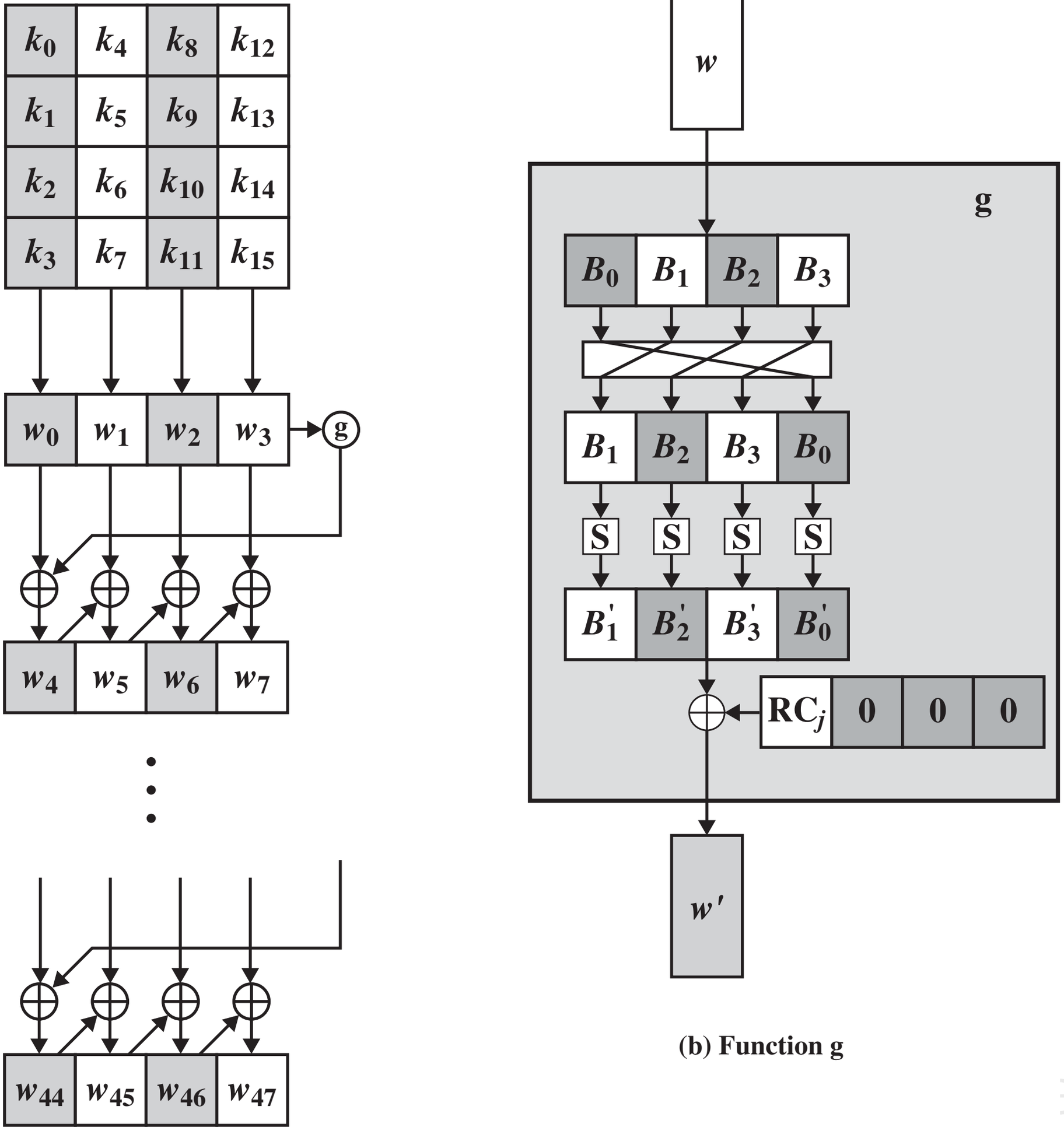
EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D6

huangwc@ustc.edu.cn

4 AES Key Expansion

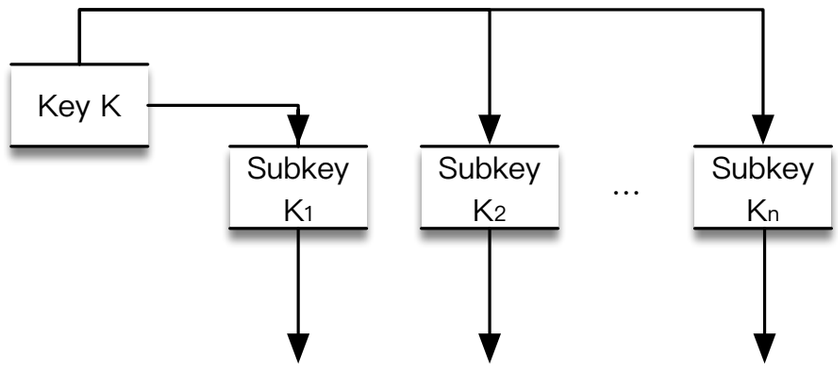


huangwc@ustc.edu.cn


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}
w_0 w_1 w_2 w_3
g
w_4 w_5 w_6 w_7
w_{44} w_{45} w_{46} w_{47}
w
g
B_0 B_1 B_2 B_3
B_1 B_2 B_3 B_0
S S S S
B'_1 B'_2 B'_3 B'_0
RC_j 0 0 0
w'

(b) Function g

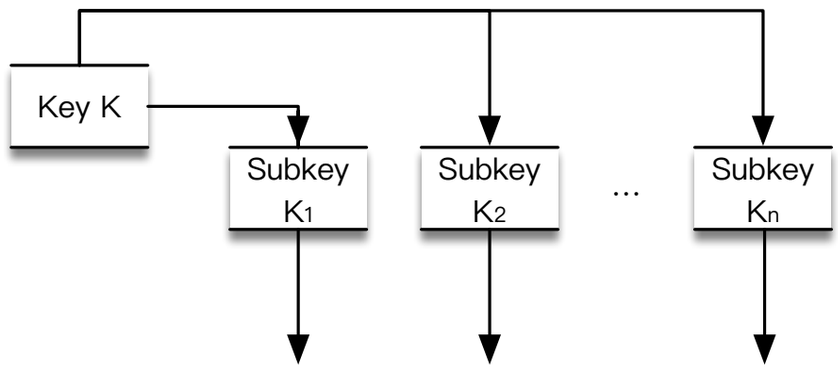
4 AES Key Expansion



- The rational of AES Key Expansion
  - Knowledge of a part of the cipher key or round key does not enable calculation of many other round-key bits.
  - An invertible transformation
    - knowledge of any N_k consecutive words of the expanded key enables regeneration the entire expanded key (N_k = key size in words)
  - Speed on a wide range of processors.

huangwc@ustc.edu.cn

4 AES Key Expansion



- The rational of AES Key Expansion
  - Usage of round constants to eliminate symmetries.
  - Diffusion (扩散) of cipher key differences into the round keys
    - each key bit affects many round key bits.
- Enough nonlinearity to prohibit the full determination of round key differences from cipher key differences only.
- Simplicity of description

huangwc@ustc.edu.cn

5 Security Analysis
The Avalanche Effect (雪崩效应)

- (Recall) *Definition*: a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the ciphertext

- If the change were small, this might provide a way to reduce the size of the plaintext or key space to be searched

huangwc@ustc.edu.cn

5 Security Analysis
Cryptanalytic attacks: XSL attack

- **XSL attack**
  - In 2002
  - A theoretical attack
    - Announced by Nicolas Courtois and Josef Pieprzyk
    - A weakness in the AES algorithm
    - partially due to the low complexity of its nonlinear components
- Since then, other papers have shown that the attack, as originally presented, is unworkable

huangwc@ustc.edu.cn

5 Security Analysis
Cryptanalytic attacks: XSL attack

- The process of XSL attack
  - The problem can be **reduced** to another problem
    - Solving multivariate quadratic equations (**多元二次方程组**)
  - However, solving multivariate quadratic equations (MQ) over a finite set of numbers is an **NP-hard problem** (in the general case) with several applications in cryptography

huangwc@ustc.edu.cn

5 Security Analysis
Cryptanalytic attacks: XSL attack

- Discussions on XSL attack

The method has some merit, and is worth investigating, but it does not break Rijndael as it stands.

The XSL attack is not an attack. It is a dream.

XSL may be a dream. It may also be a very bad dream and turn into a nightmare.

Example: input Row 9, Column 5

1. The value of byte: {95}
2. The inverse of {95} is {8A}={10001010}
3. M*[0 1 0 1 0 0 0 1]'+[1 1 0 0 0 1 1 0]'
4. S(95)={2A}

huangwc@ustc.edu.cn

5 Security Analysis
Cryptanalytic attacks: XSL attack

- Discussions on XSL attack

We have one criticism of AES: we don't quite trust the security…

What concerns us the most about AES is its simple algebraic structure…

No other block cipher we know of has such a simple algebraic representation.

Example: input Row 9, Column 5

1. The value of byte: {95}
2. The inverse of {95} is {8A}={10001010}
3. M*[0 1 0 1 0 0 0 1]'+[1 1 0 0 0 1 1 0]'
4. S(95)={2A}

huangwc@ustc.edu.cn

5 Security Analysis Cryptanalytic attacks

- 2009
  - The complexity of 2^{119}
- 2009.12
  - The complexity of $2^{99.5}$
- All the previous attacks are impractical

huangwc@ustc.edu.cn

5 Security Analysis
Side-channel Attack (旁路攻击)

- 2005.4 Cache-timing attack
  - OpenSSL's AES encryption
    - Requirements: 200 million chosen plaintext
- 2005.10 several cache-timing attacks
  - One requirements: 800 operations triggering encryptions
  - 65ms

huangwc@ustc.edu.cn

5 Security Analysis
Side-channel Attack (旁路攻击)

- 2009
  - an attack on some hardware implementations
  - Differential fault analysis
  - The complexity of 2^{32}
- 2010
  - AES-128 implementations that use compression tables, such as OpenSSL
  - A "near real time" recovery of secret keys
  - Without the need for either cipher text or plaintext
  - Requirements：the ability to run unprivileged code on the system performing the AES encryption

huangwc@ustc.edu.cn

5 Security Analysis
Side-channel Attack (旁路攻击)

- 2016
  - Attacks on128 bit AES
    - a substantial improvement over previous works that require between 100 and a million encryptions
  - Requirements：
    - recover the complete 128-bit AES key in just 6–7 blocks of plaintext/ciphertext
    - standard user privilege
    - key-retrieval algorithms run under a minute

huangwc@ustc.edu.cn

5 Security Analysis
Side-channel Attack (旁路攻击)

- Protections against timing-related side-channel attacks

- Many modern CPUs have built-in hardware instructions for AES

huangwc@ustc.edu.cn

Homework

- Null

huangwc@ustc.edu.cn