3 Block Ciphers and the Data Encryption Standard

Fuyou Miao，Wenchao Huang

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

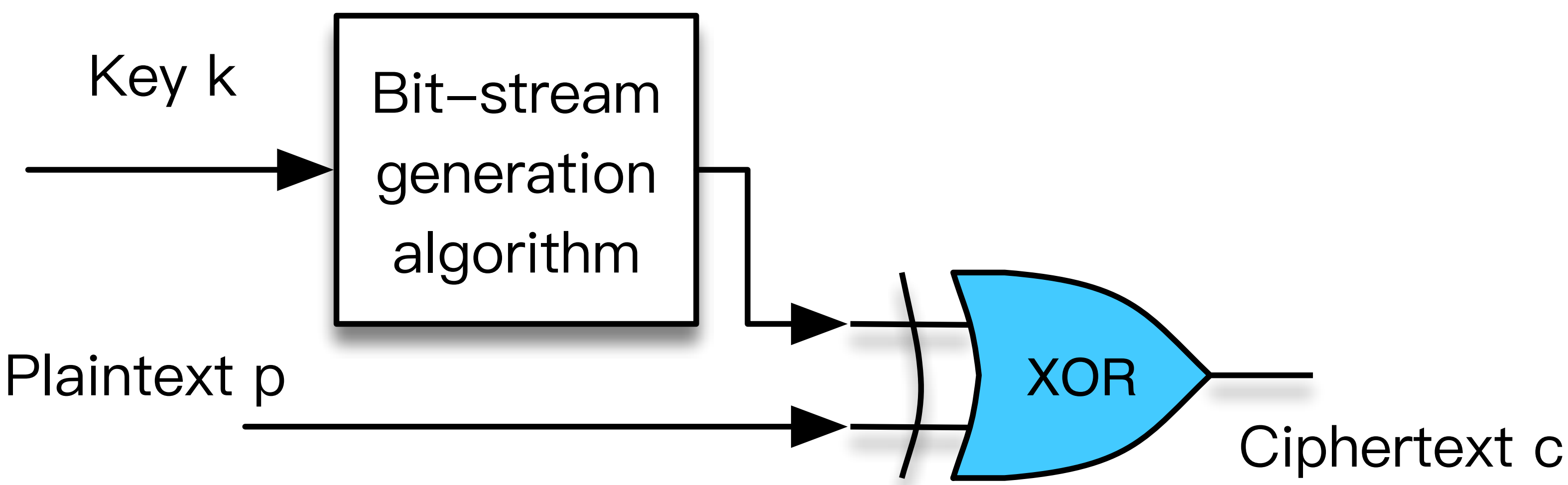
Email: mfy@ustc.edu.cn，huangwc@ustc.edu.cn

Contents

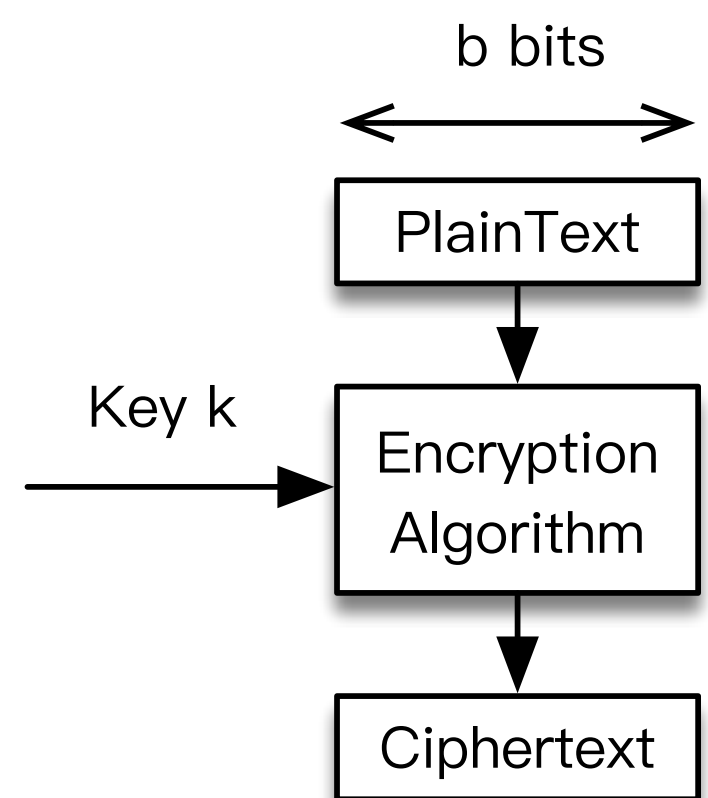
huangwc@ustc.edu.cn

1. Block Cipher Principles
Block Cipher v.s. Stream Cipher

- A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time
- A **block cipher** is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length



Stream Cipher



Block Cipher

2. The Feistel Cipher
Motivation: (1) Reversible Transformation

Reversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	00
11	01

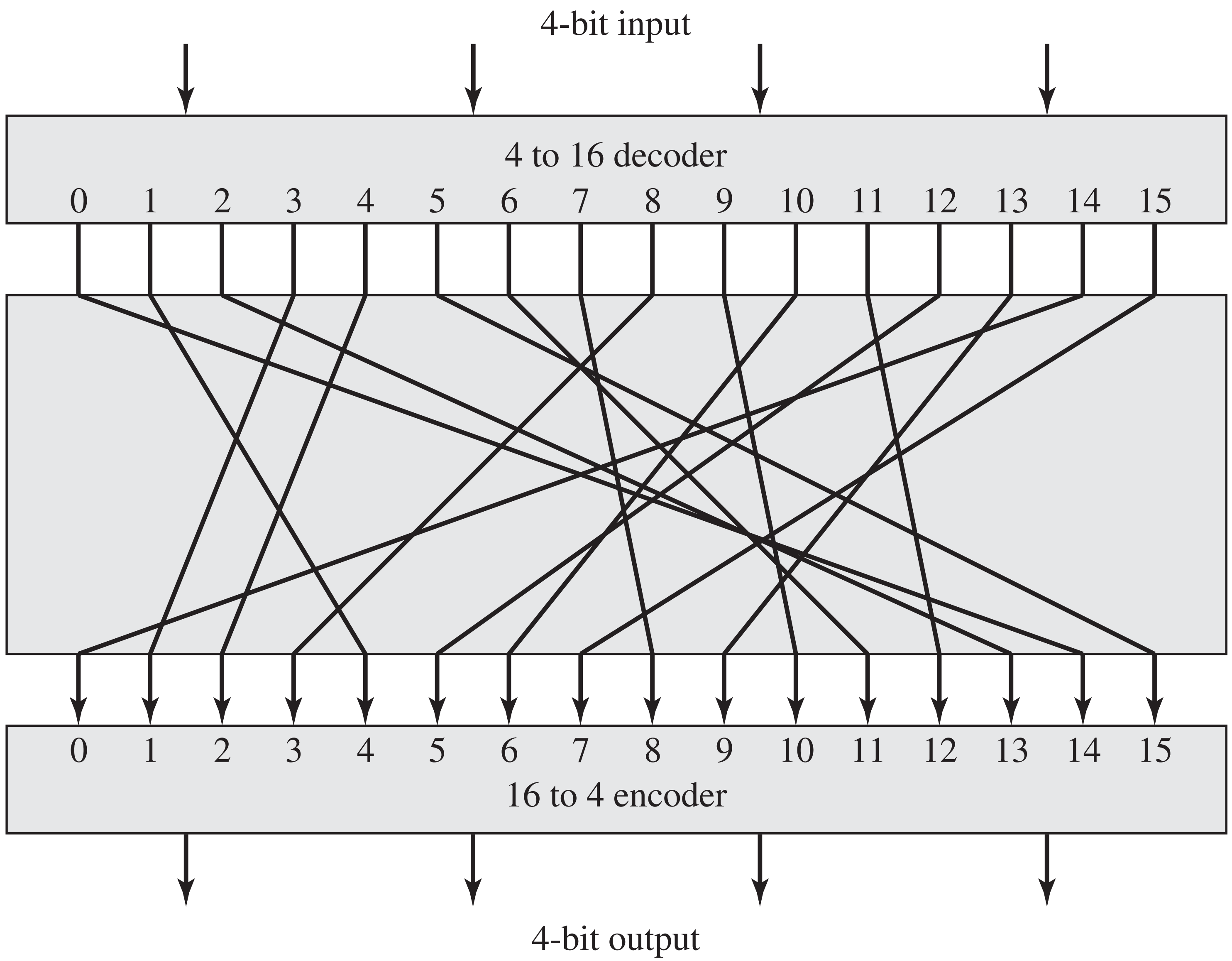
Irreversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	**01**
11	**01**

For a plaintext block of n, if we limit ourselves to reversible mappings, the number of different transformations is $2^n!$

i.e., the number of possible encryption mappings is $2^n!$

2. The Feistel Cipher Motivation: (2) **Ideal** block Cipher



Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
…	…
1111	0101

Pros: It allows for the maximum number of possible encryption mappings from the plaintext block

2. The Feistel Cipher
Motivation: (2) **Ideal** block Cipher

- If n is small
  - vulnerable to a statistical analysis of the plaintext
- If n is sufficiently large
  - not practical
  - the length of the key defined in this fashion is $n \times 2^n$ bits

n bits

2^n bits

Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
…	…
1111	0101

2. The Feistel Cipher
(3) Solution: Approximation

- Feistel proposed [FEIS73] that **(1)** we can approximate the ideal block cipher by utilizing the concept of a **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

2. The Feistel Cipher
(3) Solution: Approximation

- Feistel proposed [FEIS73] **(2)** the use of a cipher that alternates **substitutions (代换)** and **permutations (置换)** in the product

  - **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.

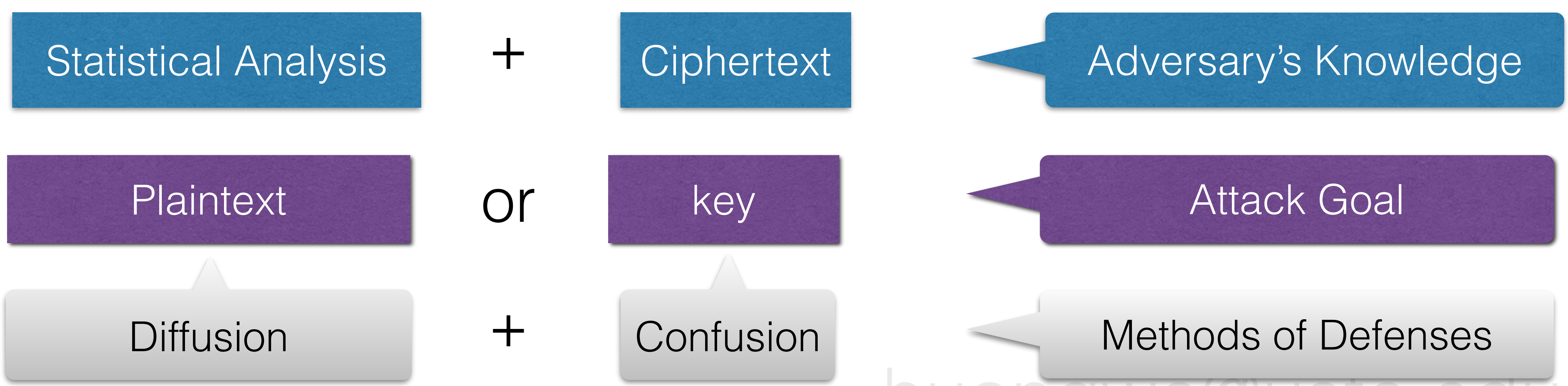
    - That is, no elements are added or deleted or replaced in the sequence

2. The Feistel Cipher
(3) Solution: Approximation

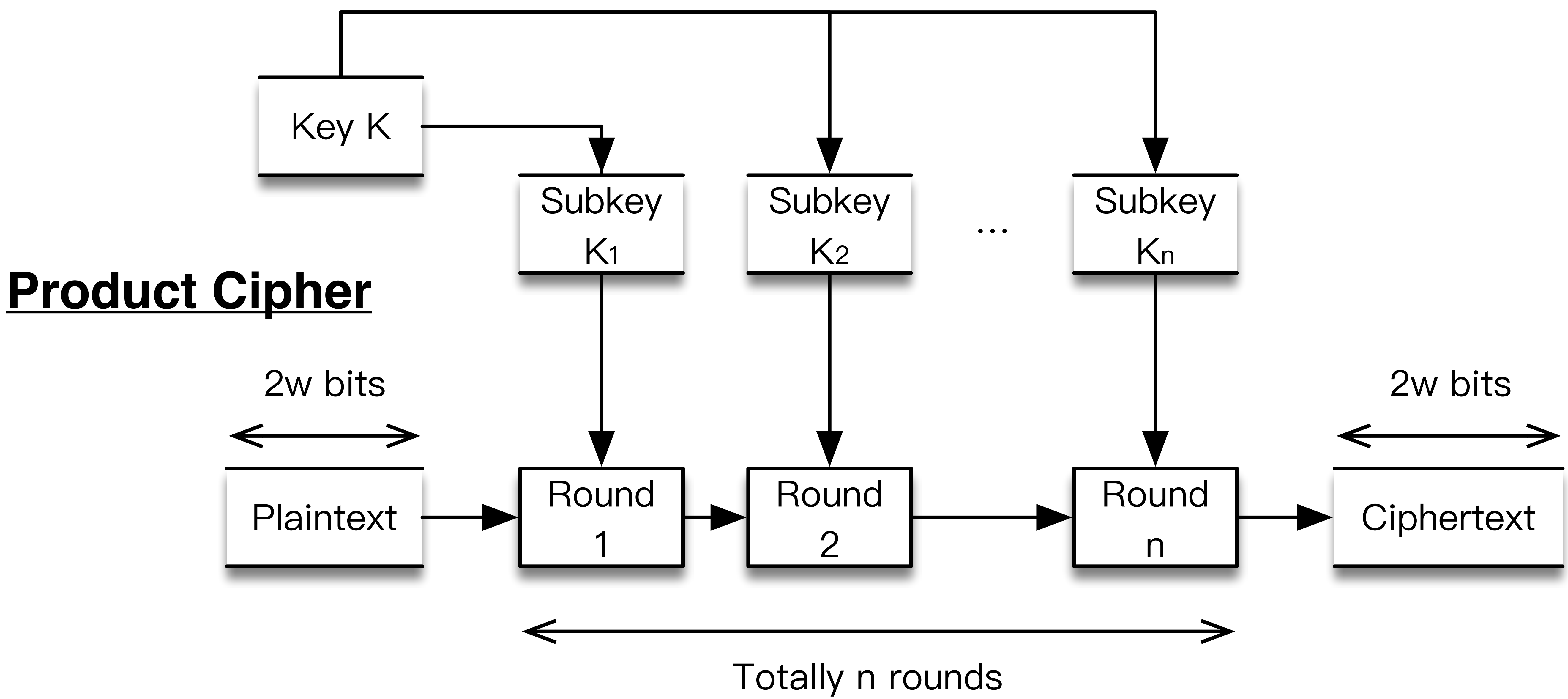
- Feistel proposed [FEIS73] **(2)** the use of a cipher that alternates **substitutions (代换)** and **permutations (置换)** in the product cipher
  - *Origin*: Claude Shannon [SHAN49] develops a product cipher that *alternates* **confusion (混淆)** and **diffusion (扩散)** functions
    - the structure used by many significant symmetric block ciphers currently in use

2. The Feistel Cipher
(3) Solution: Approximation

- **Confusion (混淆)** and **Diffusion (扩散)**
    - **Diffusion**： the statistical structure of the plaintext is dissipated into long-range statistics of the ciphertext
        - by having each plaintext digit affect the value of many ciphertext digits
    - **Confusion**： make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible



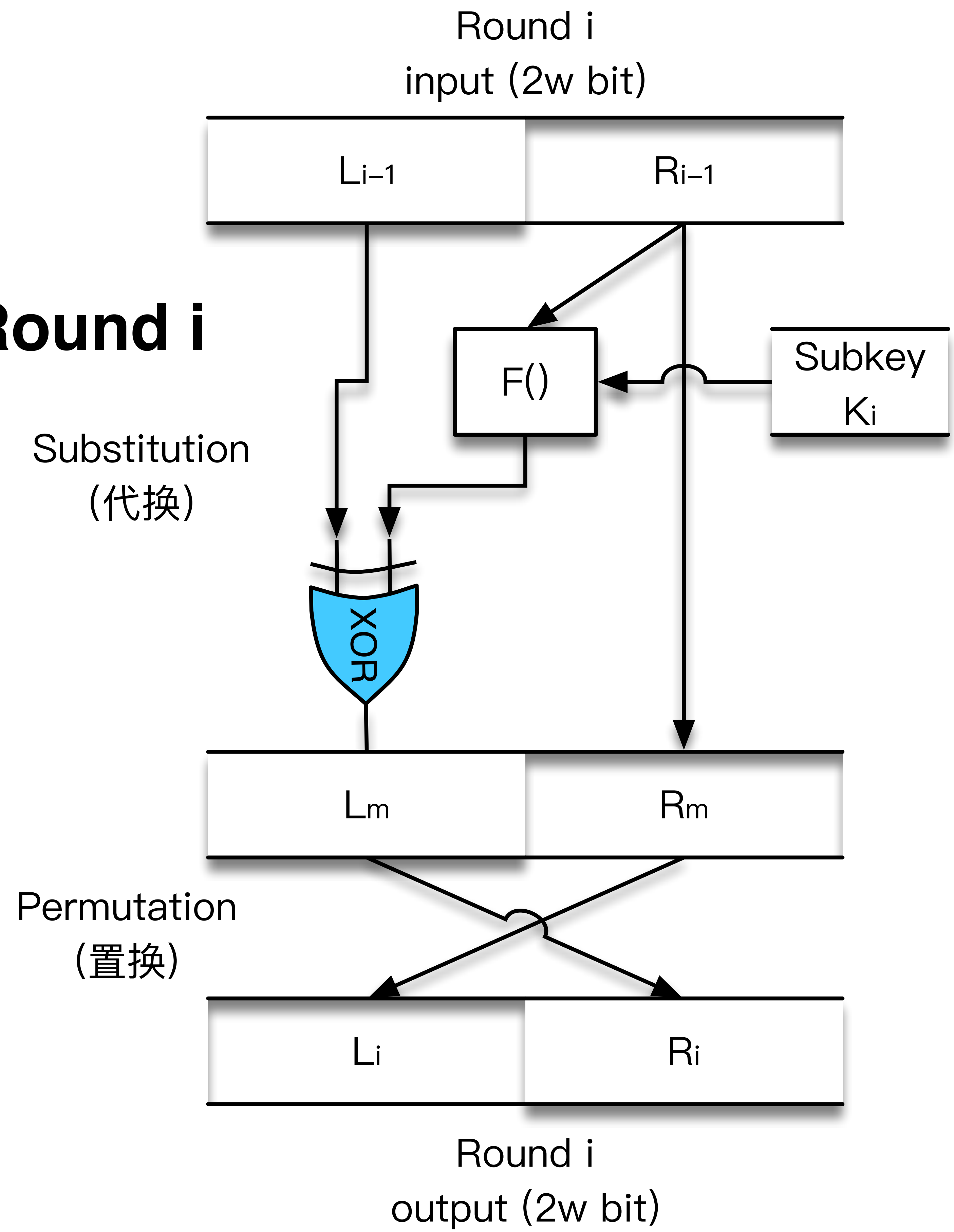
2. The Feistel Cipher
(3) Solution: Approximation



2. The Feistel Cipher (3) Solution: Approximation

- Substitution
  - **Round function**
    - F(Data,Key)
- Permutation

Round i



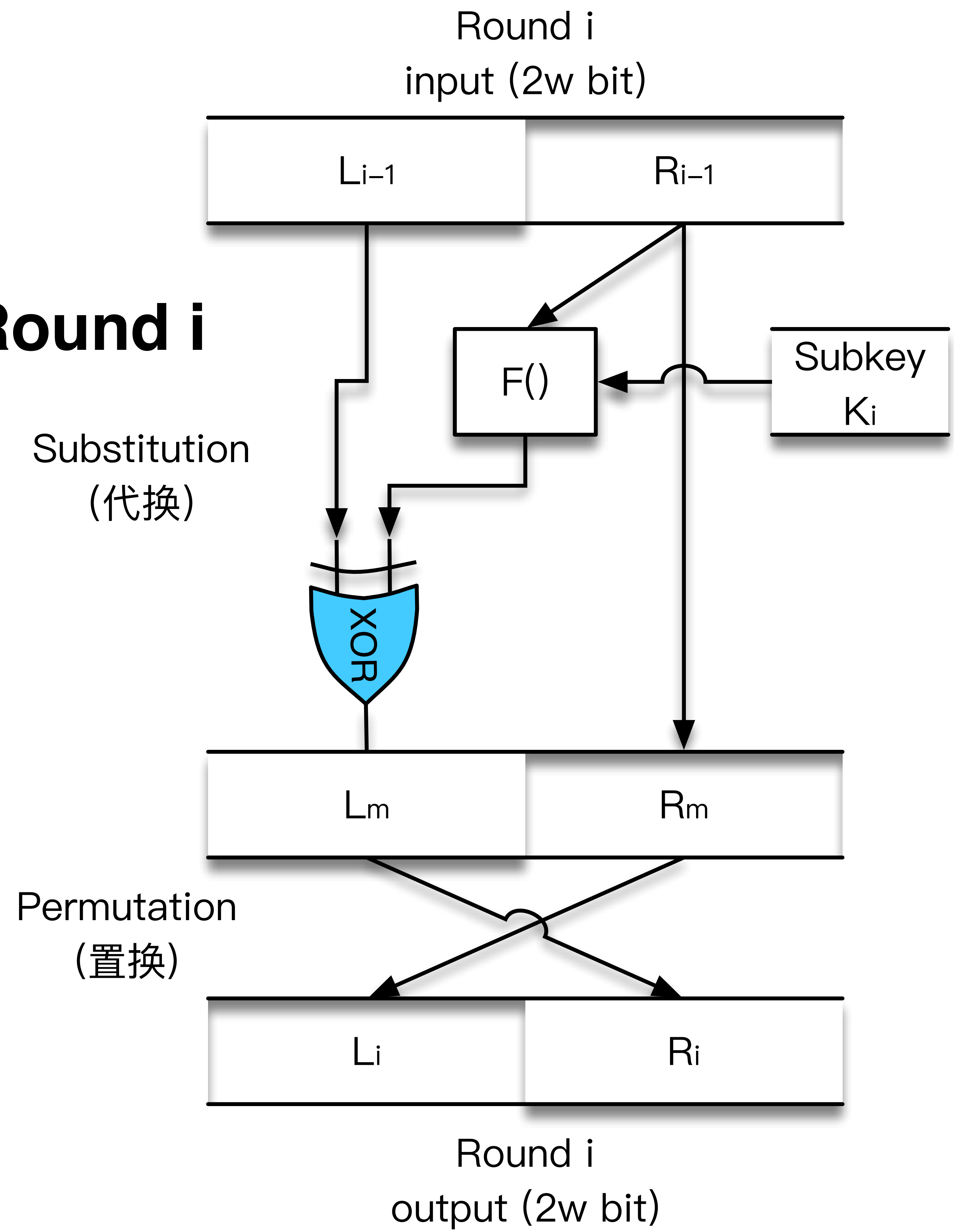
2. The Feistel Cipher (3) Solution: Approximation

- **Encryption** Algorithm **Round i**

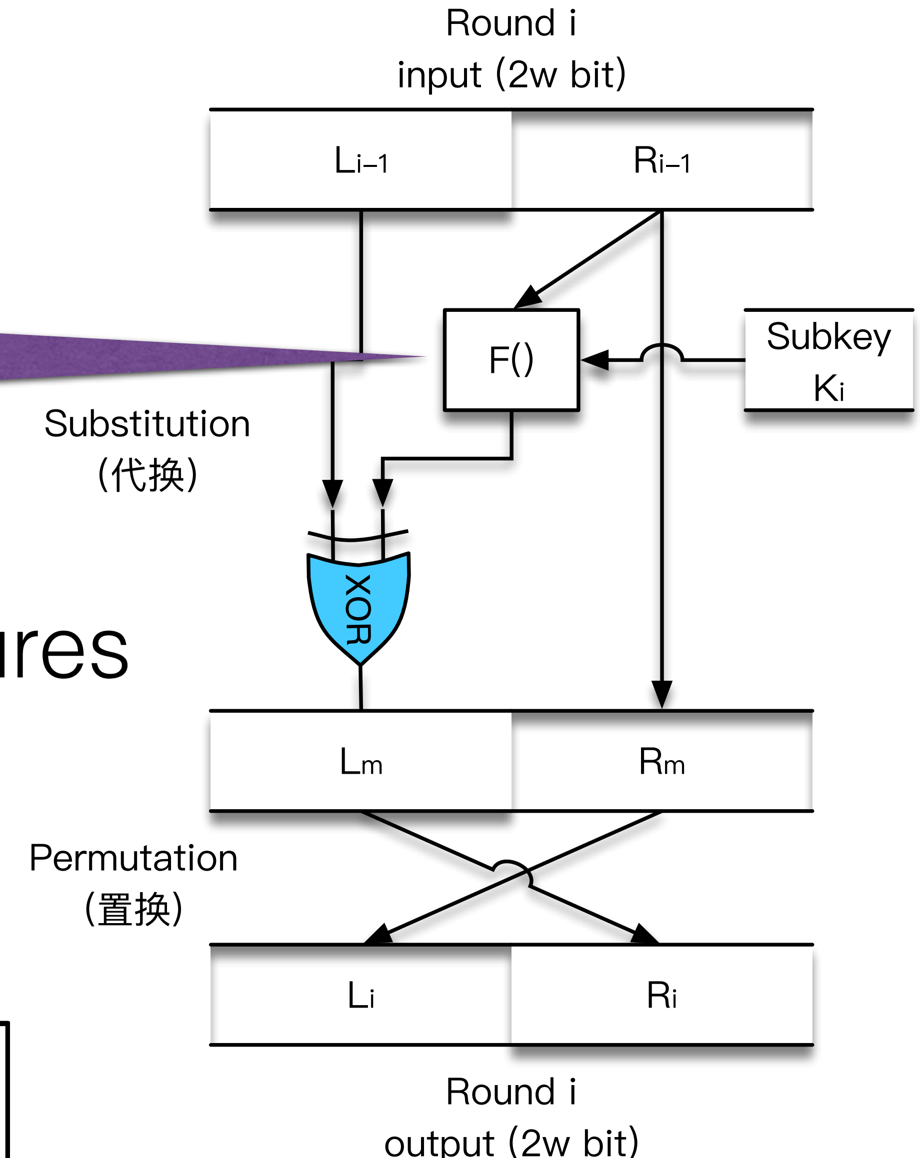
$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

- **Decryption** Algorithm

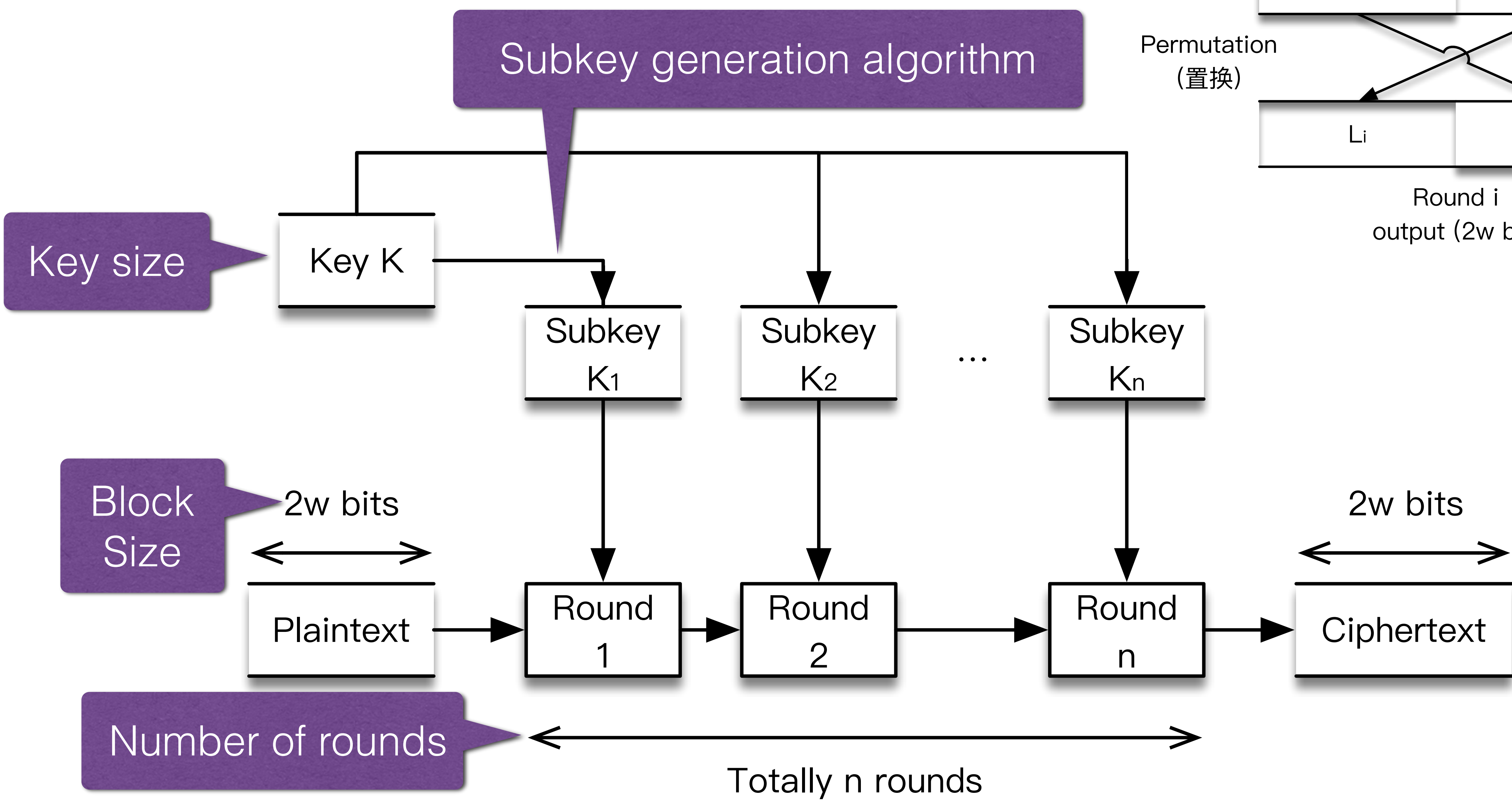
$$R_{i-1} = L_i$$
$$L_{i-1} = R_i \oplus F(R_{i-1}, K_i)$$
$$= R_i \oplus F(L_i, K_i)$$



2. The Feistel Cipher (3) Solution: Approximation



The **choice** of parameters and design features



2. The Feistel Cipher
(3) Solution: Approximation

- Other Considerations
  - **Fast** software encryption/decryption
  - **Ease** of analysis
    - *Benefits*: if the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength

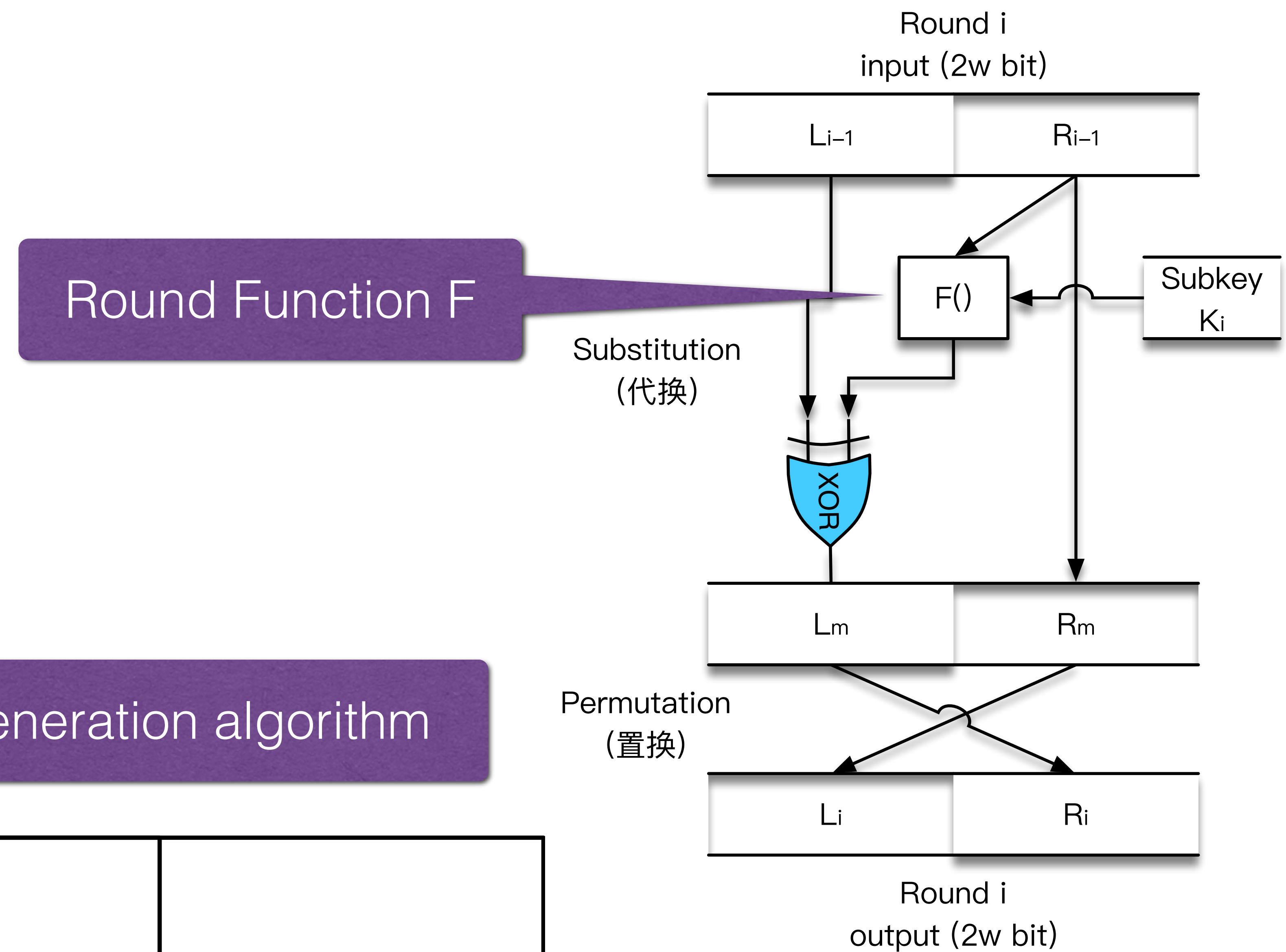
3. The Data Encryption Standard (DES)

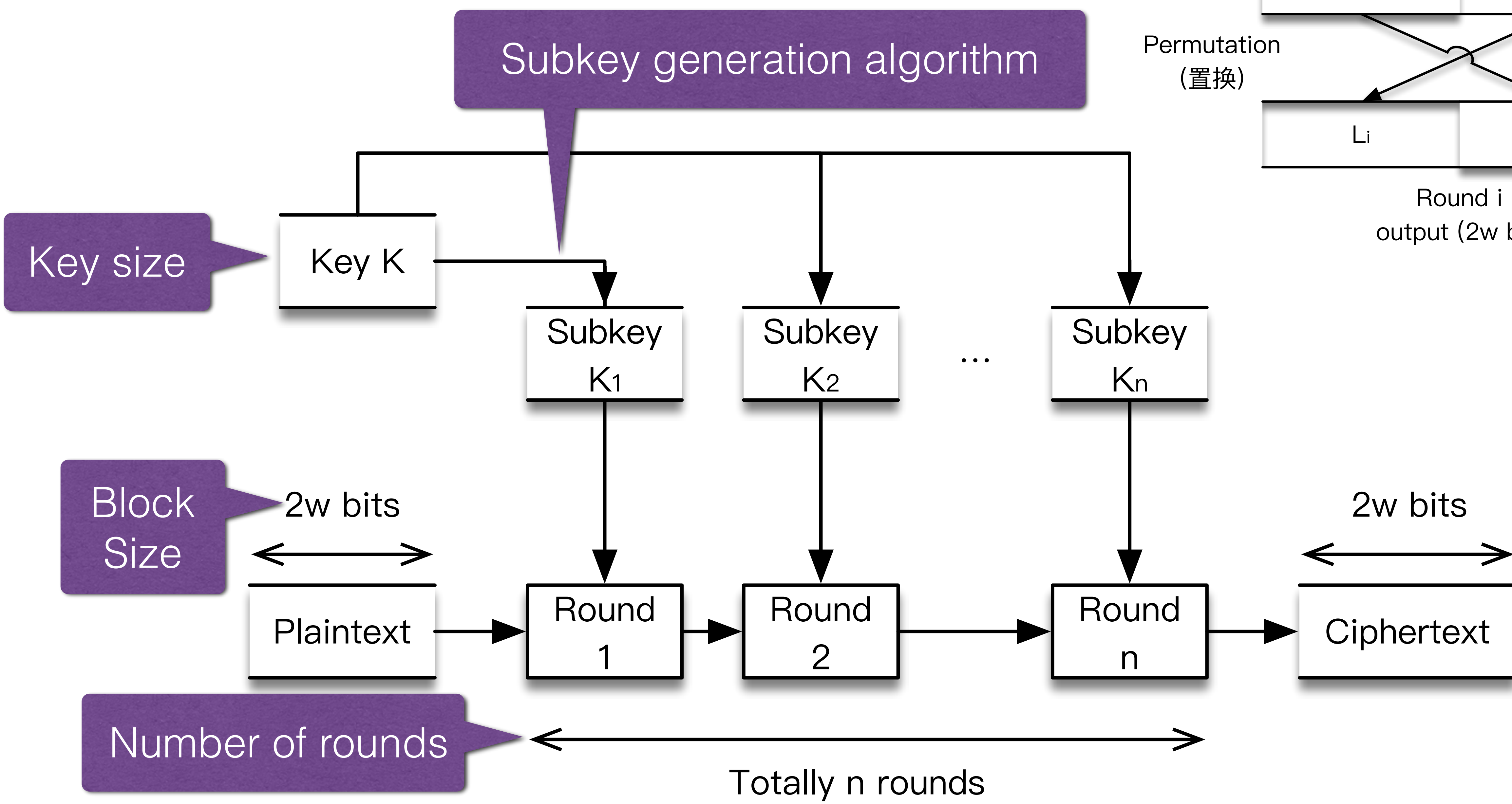
- In the late 1960s, IBM set up a research project in computer cryptography led by **Horst Feistel**.
- The project concluded in 1971 with the development of an algorithm with the designation **LUCIFER** [FEIS73]
    - sold to Lloyd's of London for use in a cash-dispensing system, also developed by IBM
- Implemented on a **single chip**
    - more resistant to cryptanalysis
    - but a reduced key size of 56 bits, in order to fit on a single chip

3. The Data Encryption Standard (DES)

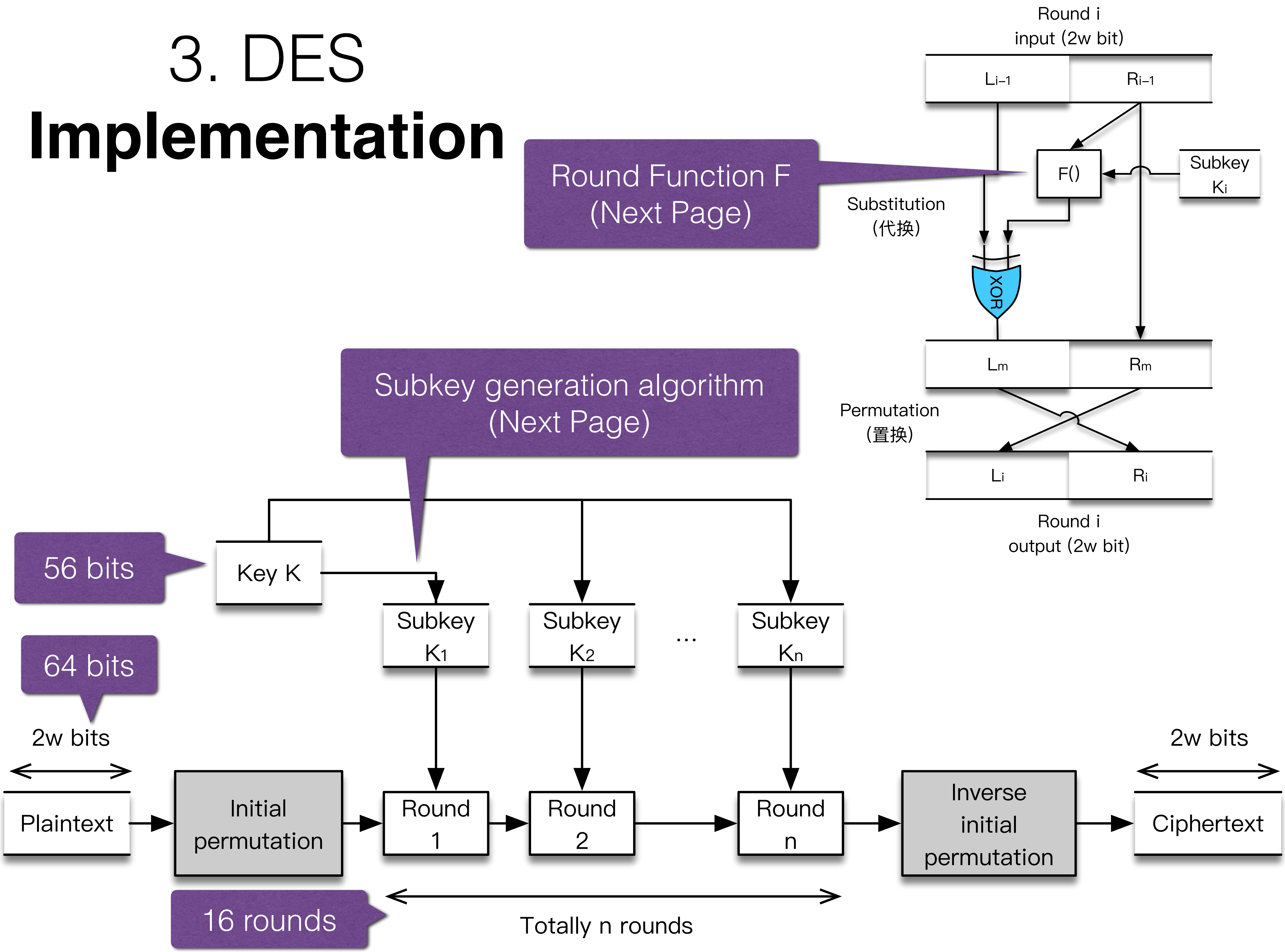
- In 1973, the National Bureau of Standards (NBS) issued a request for proposals for a **national cipher standard**
- adopted in 1977 as the Data Encryption Standard
    - criticism
        - key length of 56 bits: too short to withstand brute-force attacks
        - design criteria for the internal structure of DES, the S-boxes, were classified
- widely used, especially in financial applications
    - In 1994, NIST reaffirmed DES for federal use for another five years
    - the use of DES for applications other than the protection of classified information
    - In 1999, NIST issued a new version of its standard (FIPS PUB 46-3): **3DES**

3. DES
Recall Feistel Cipher

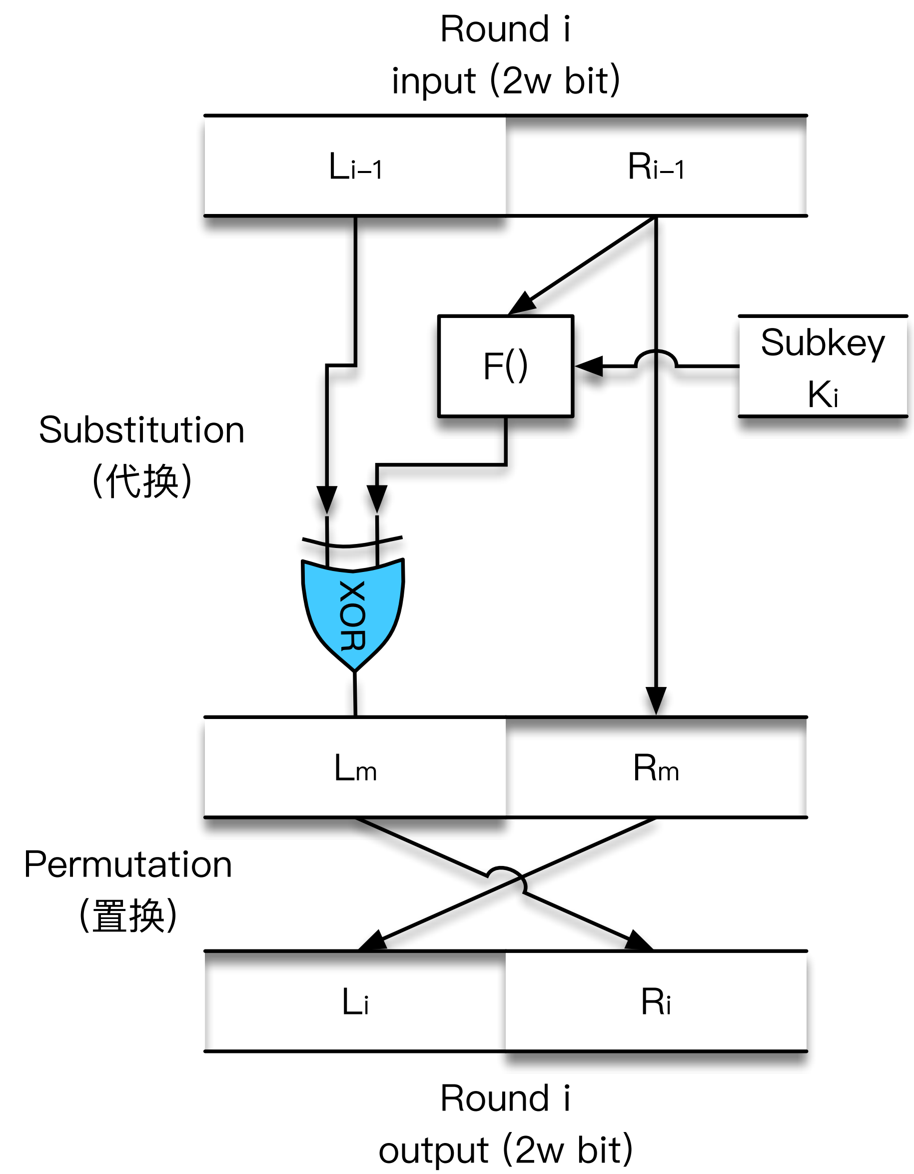


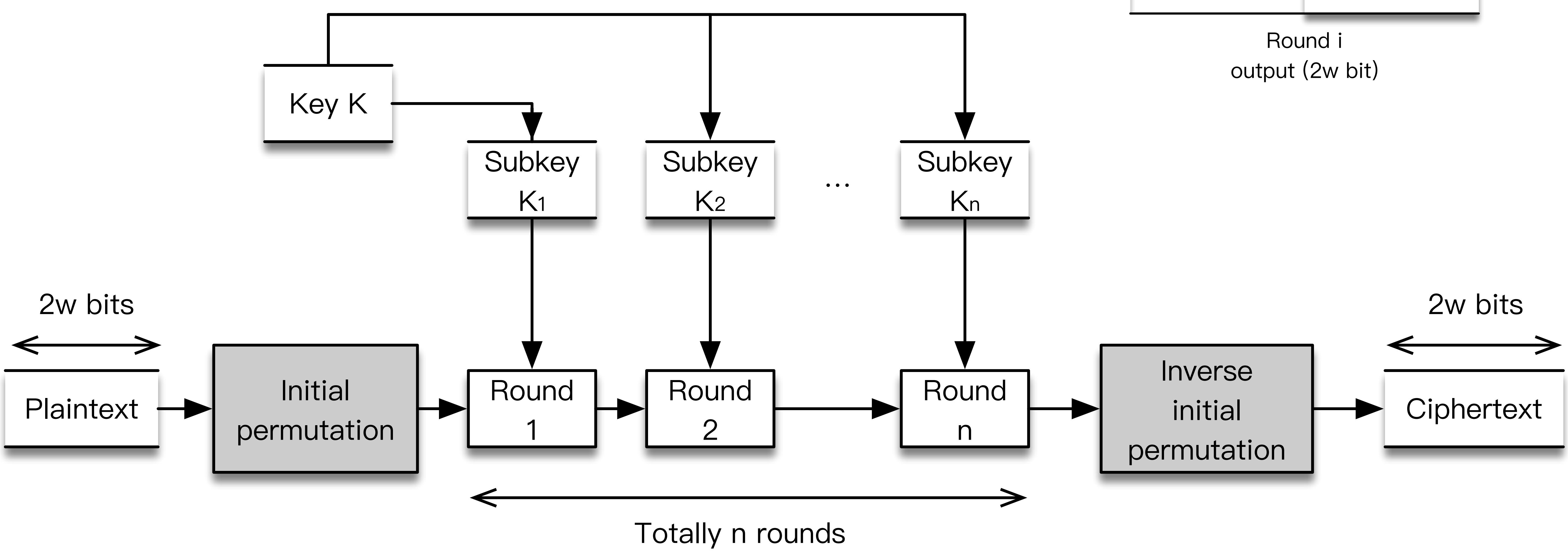


3. DES Implementation

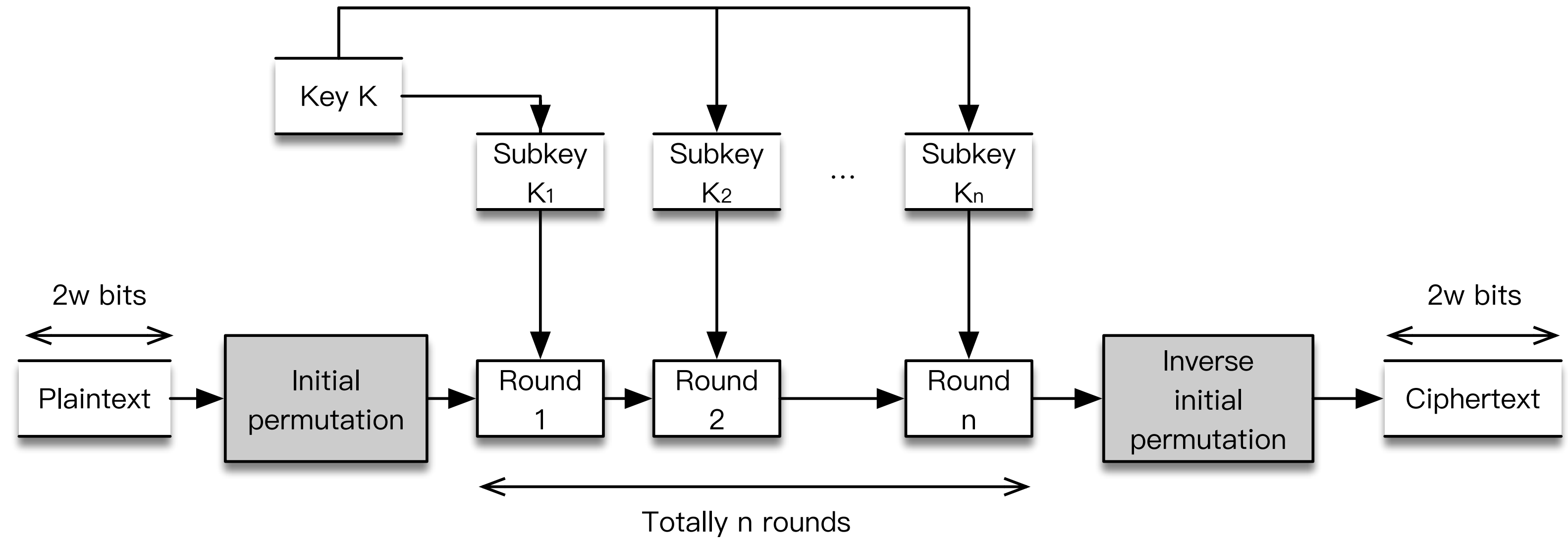


3. DES
Permutation





3. DES
Permutation



Initial Permutation

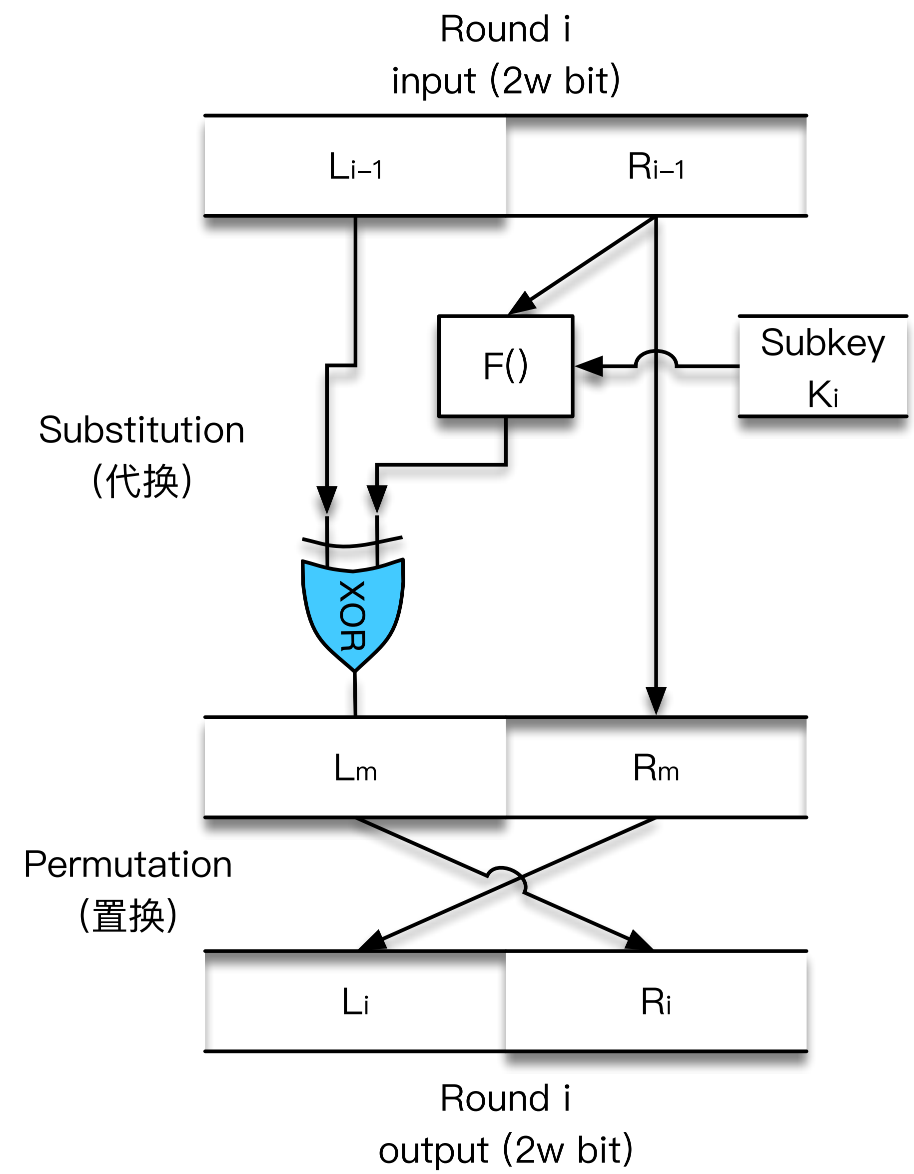
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

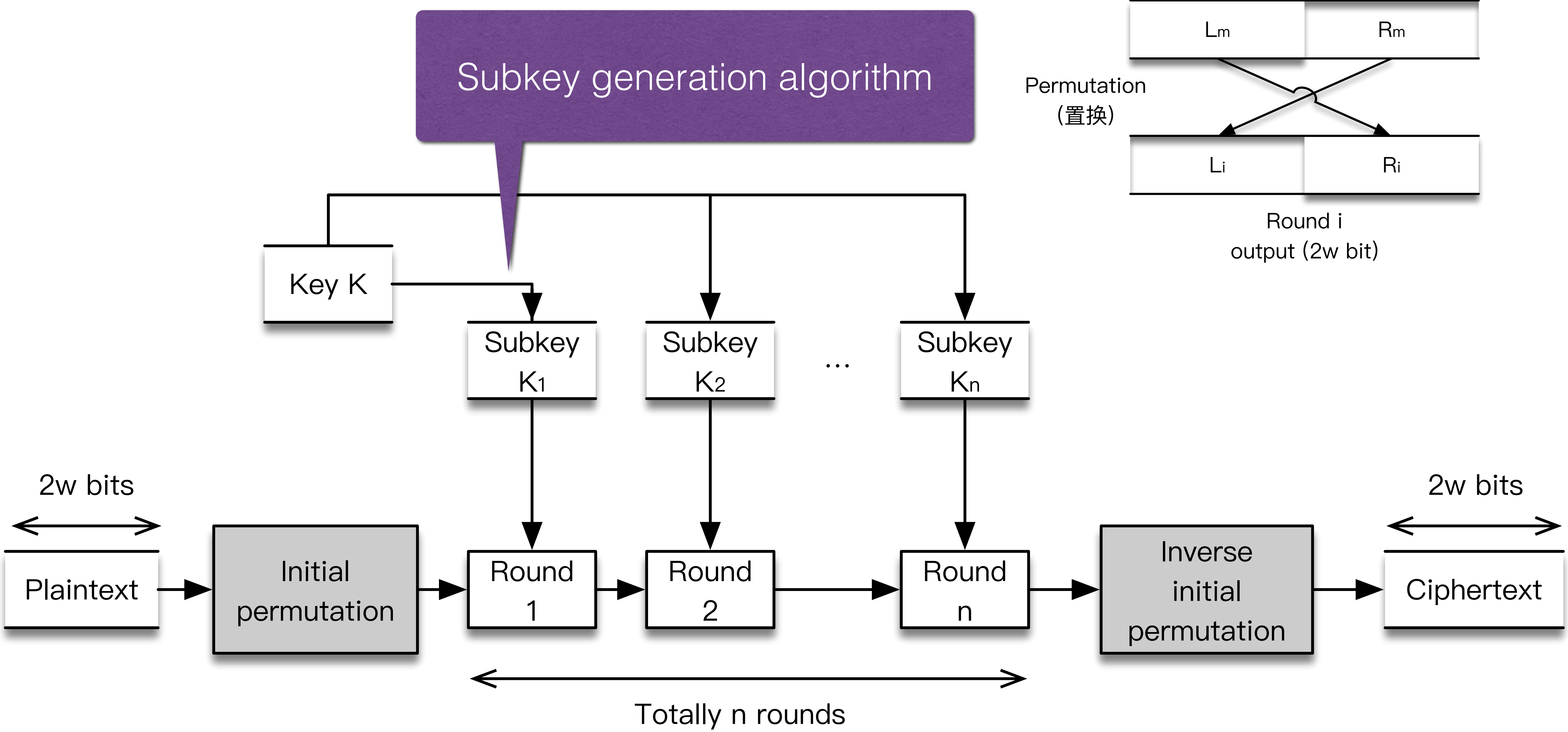
Reverse initial permutation

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	49	18	58	26
33	1	41	9	48	17	57	25

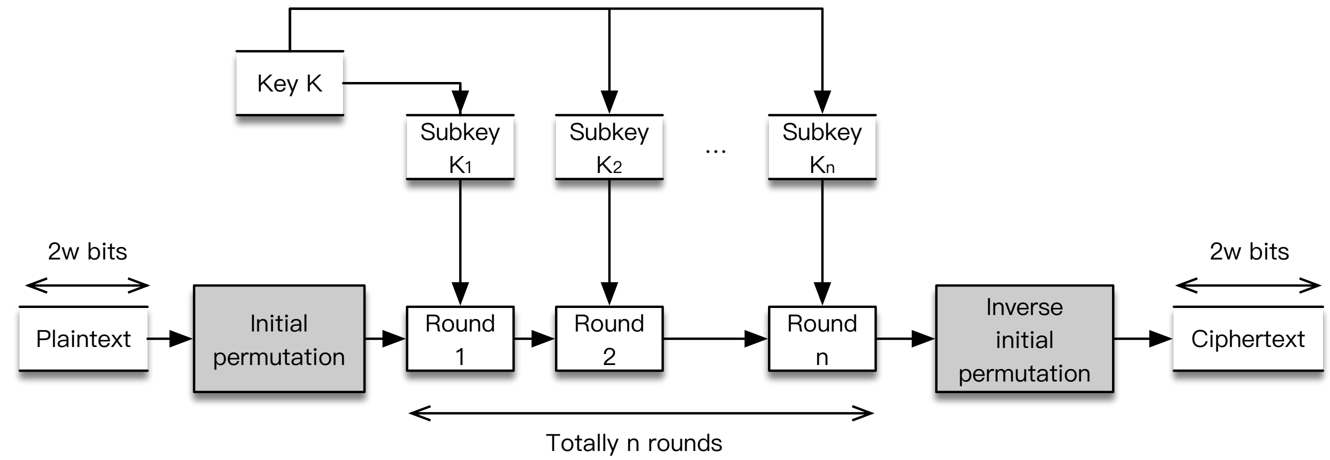
3. DES

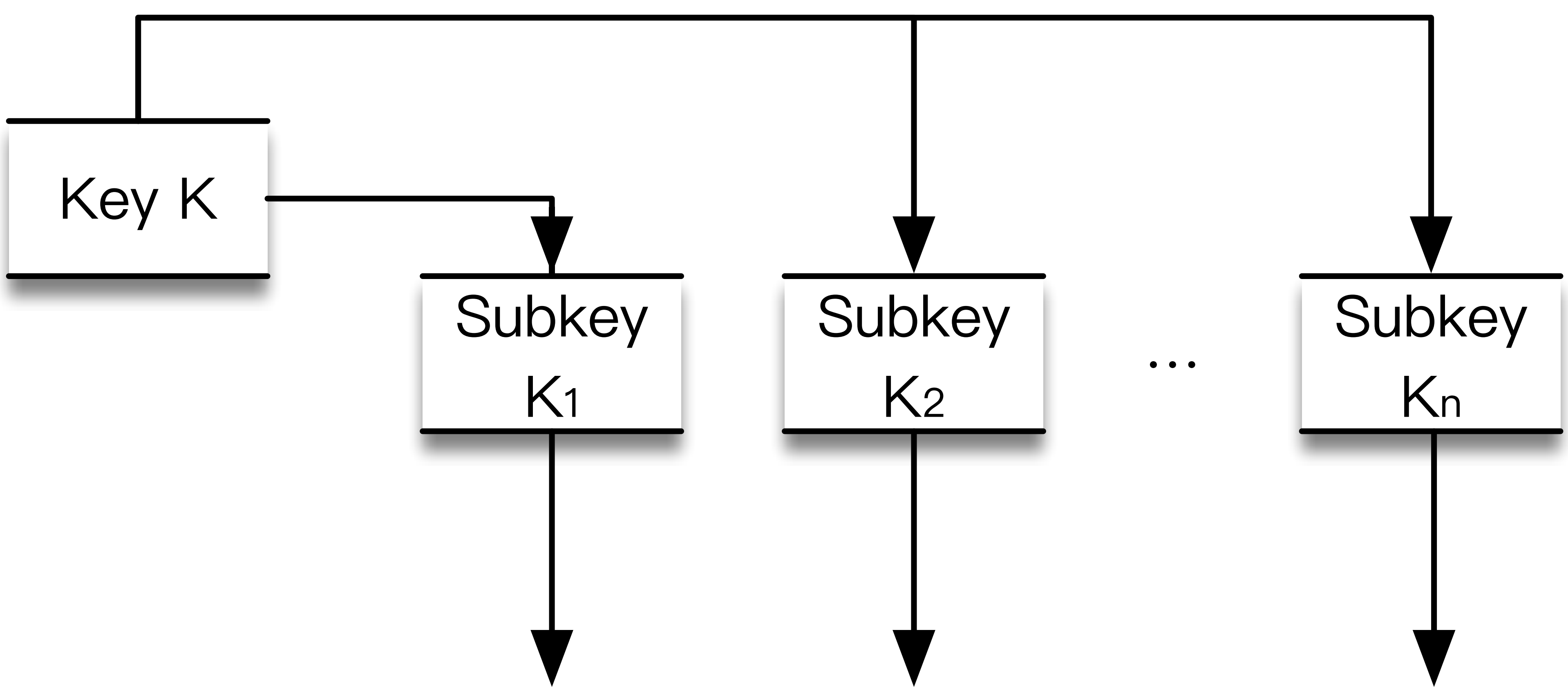
Subkey generation



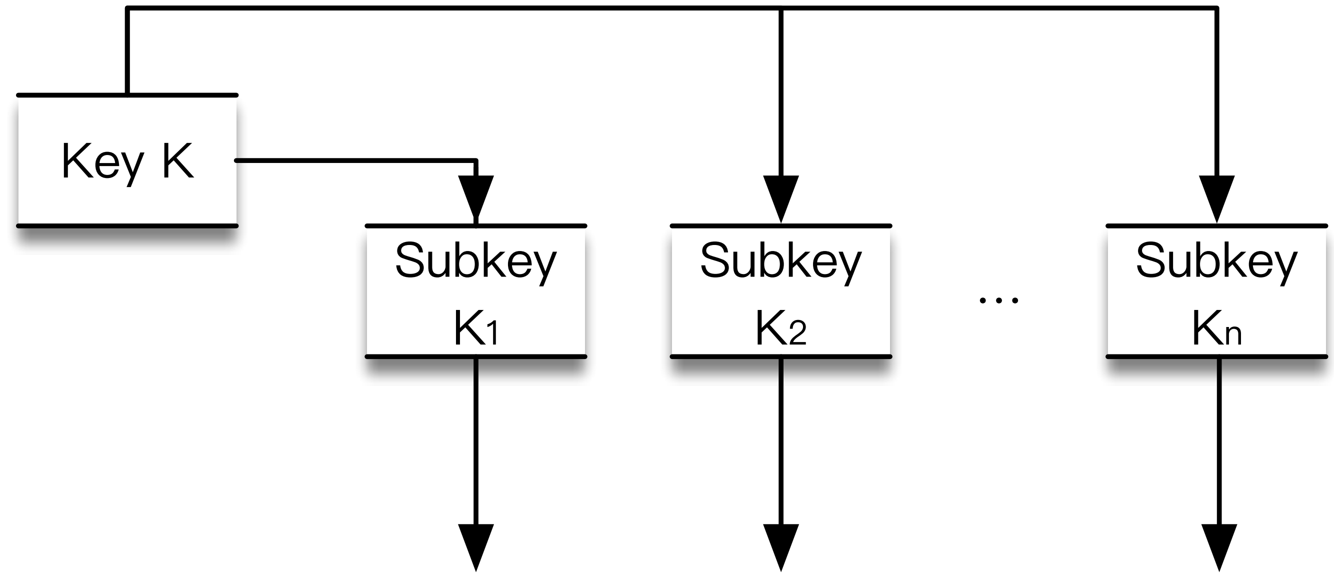


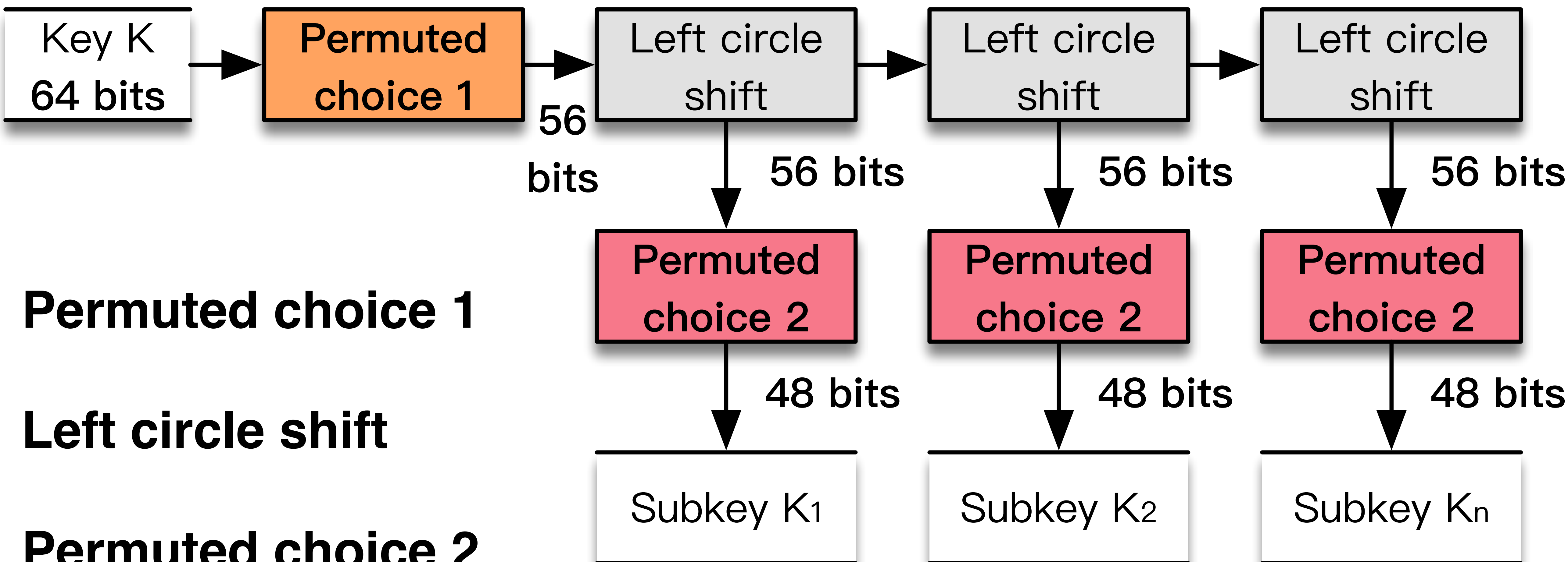
3. DES
Subkey generation





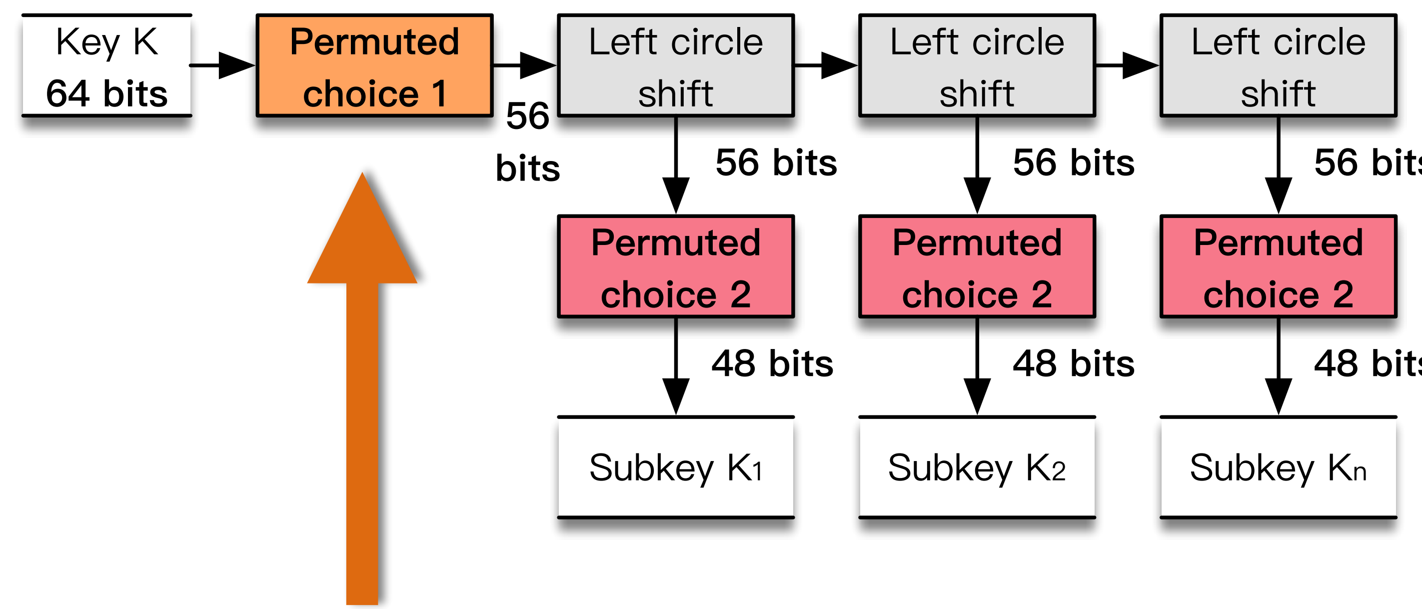
3. DES
Subkey generation





- **Permuted choice 1**
- **Left circle shift**
- **Permuted choice 2**

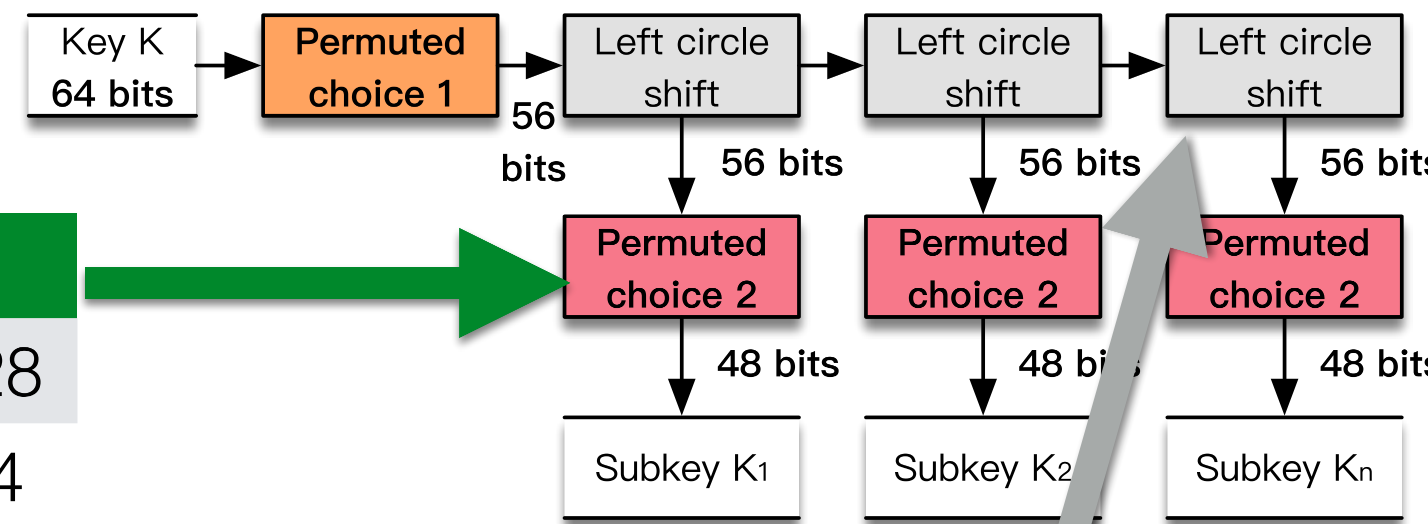
3. DES
Subkey generation



Key K							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Permuted choice 1						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

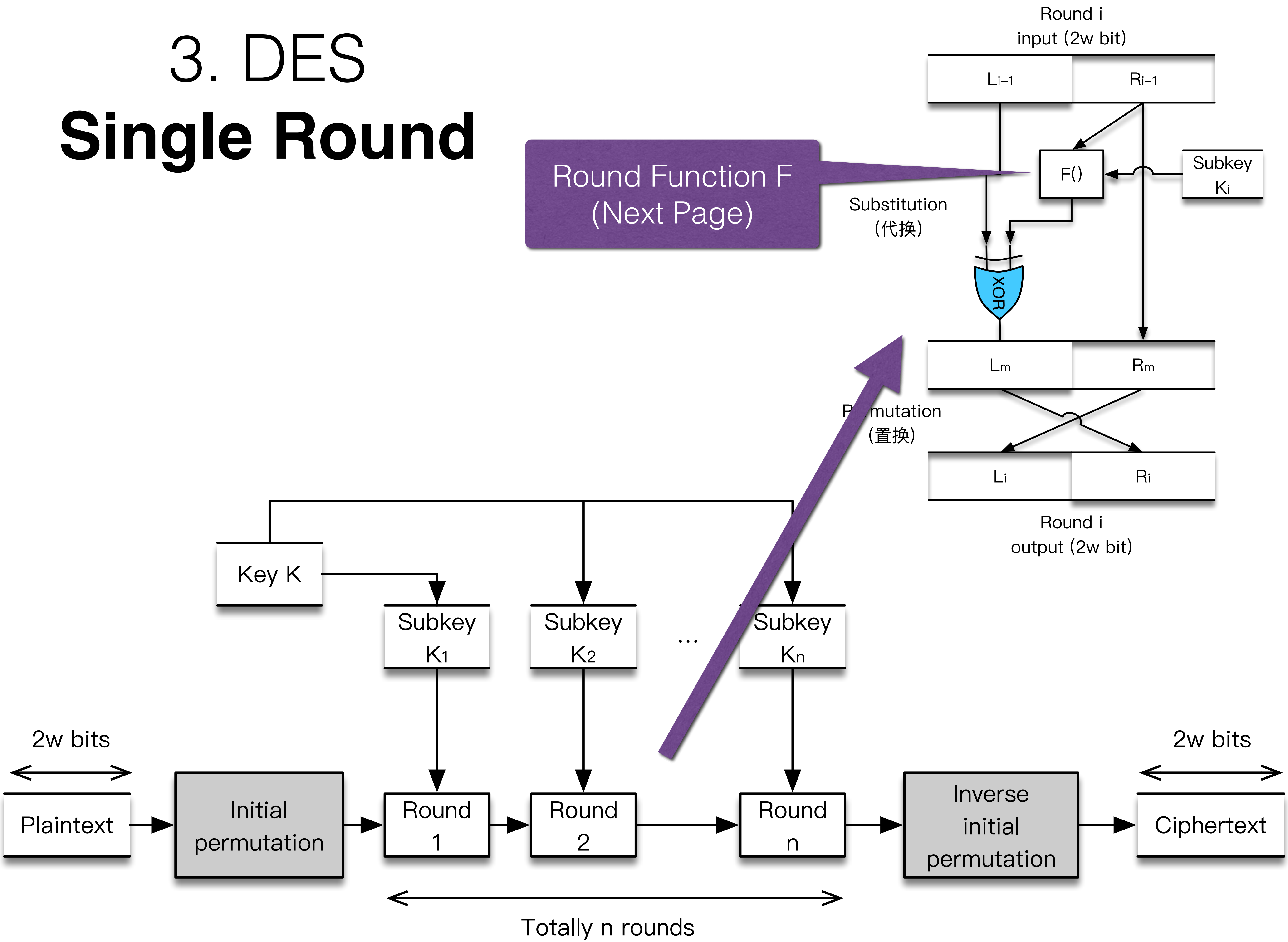
3. DES
Subkey generation



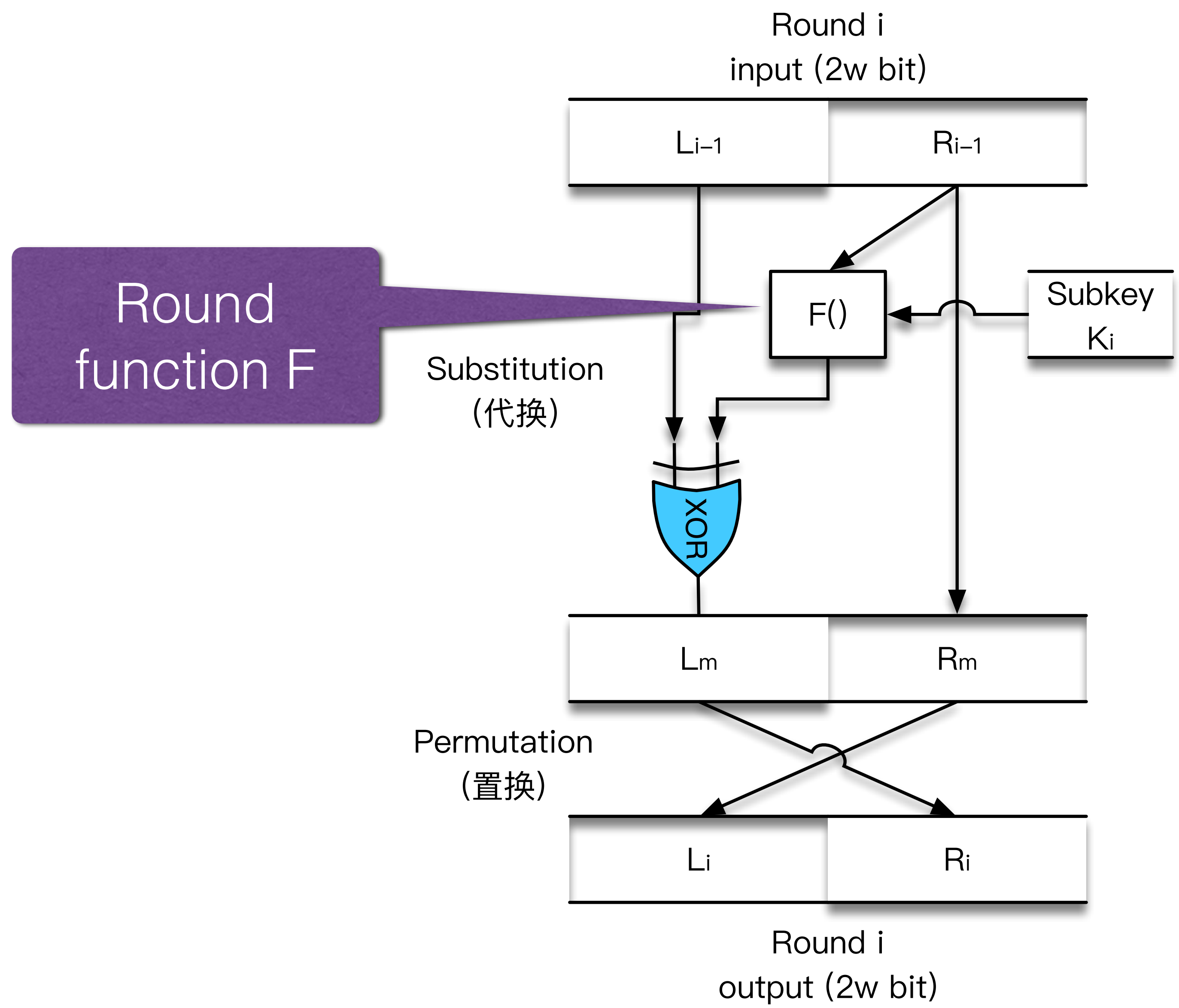
Permuted choice 2							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Left circle shift																
Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Shift count	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

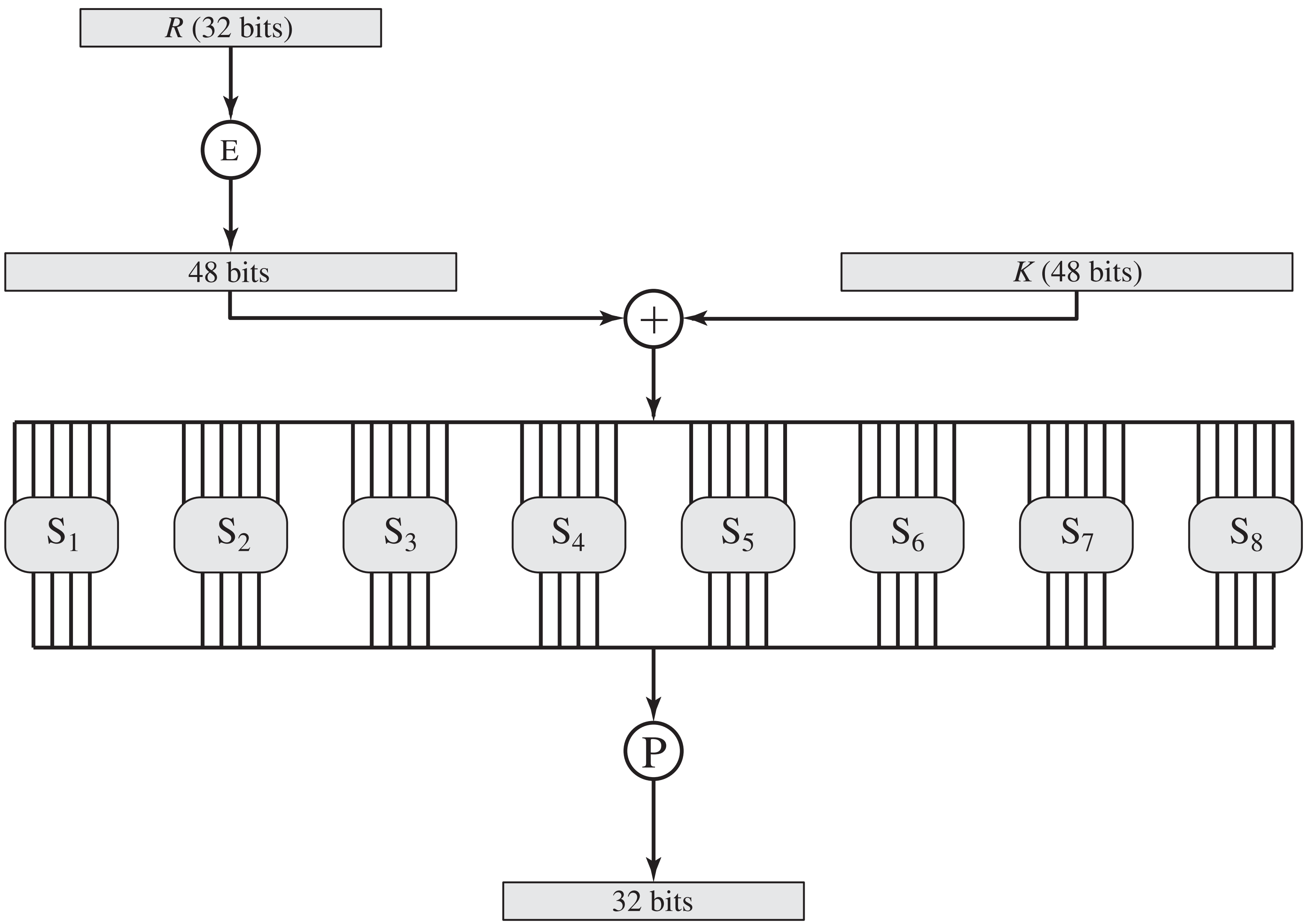
3. DES
Single Round

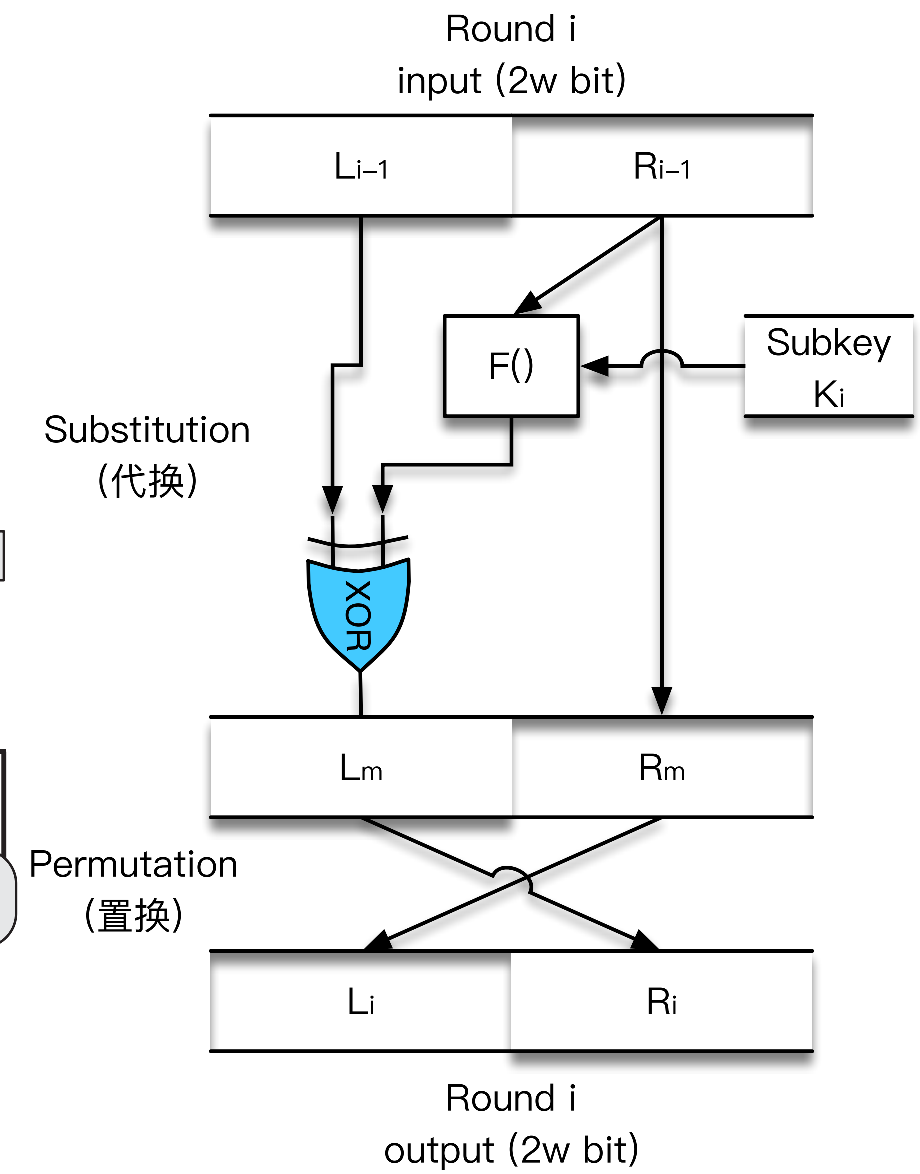


3. DES
Single Round

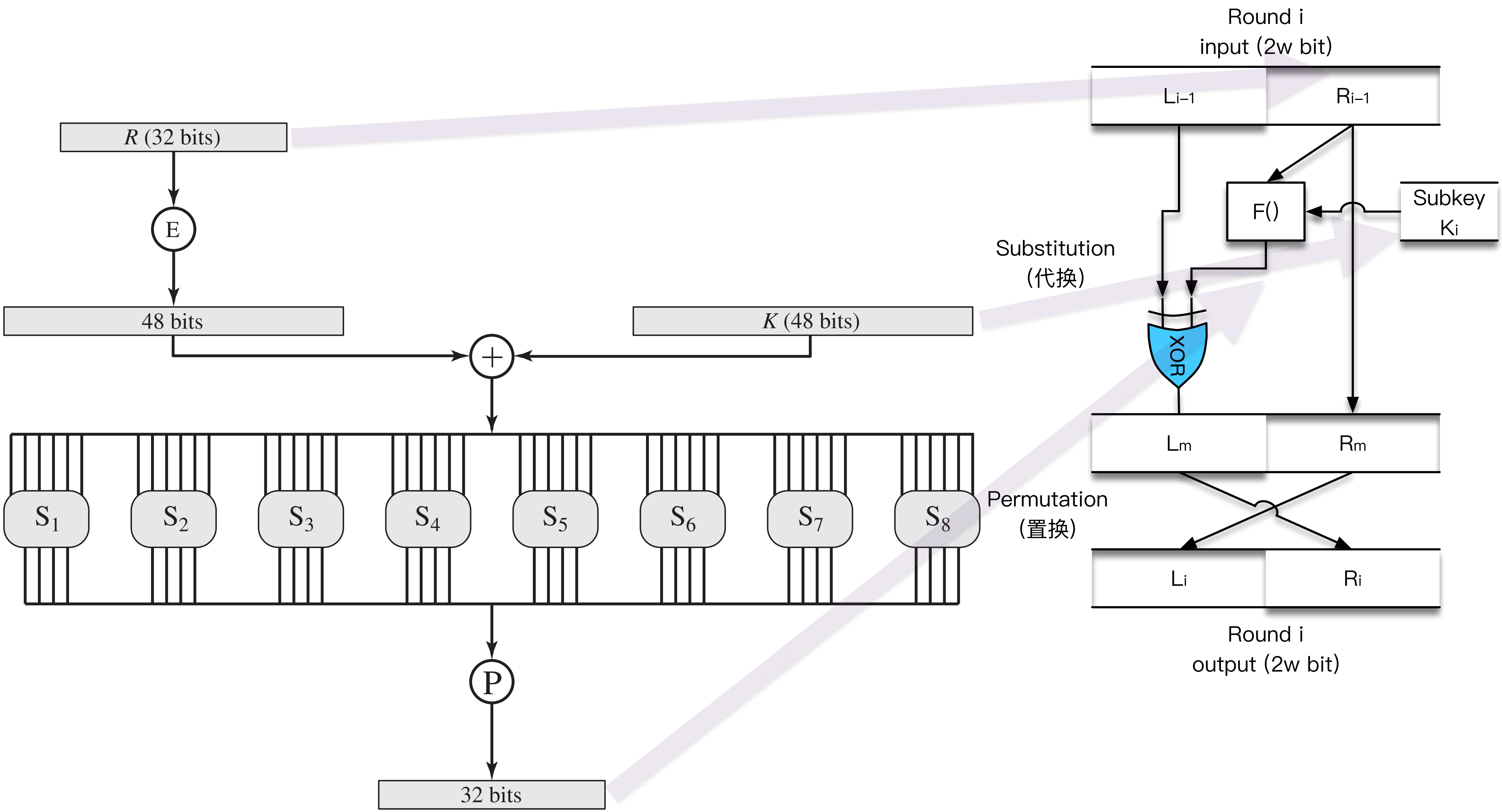


3. DES
Single Round

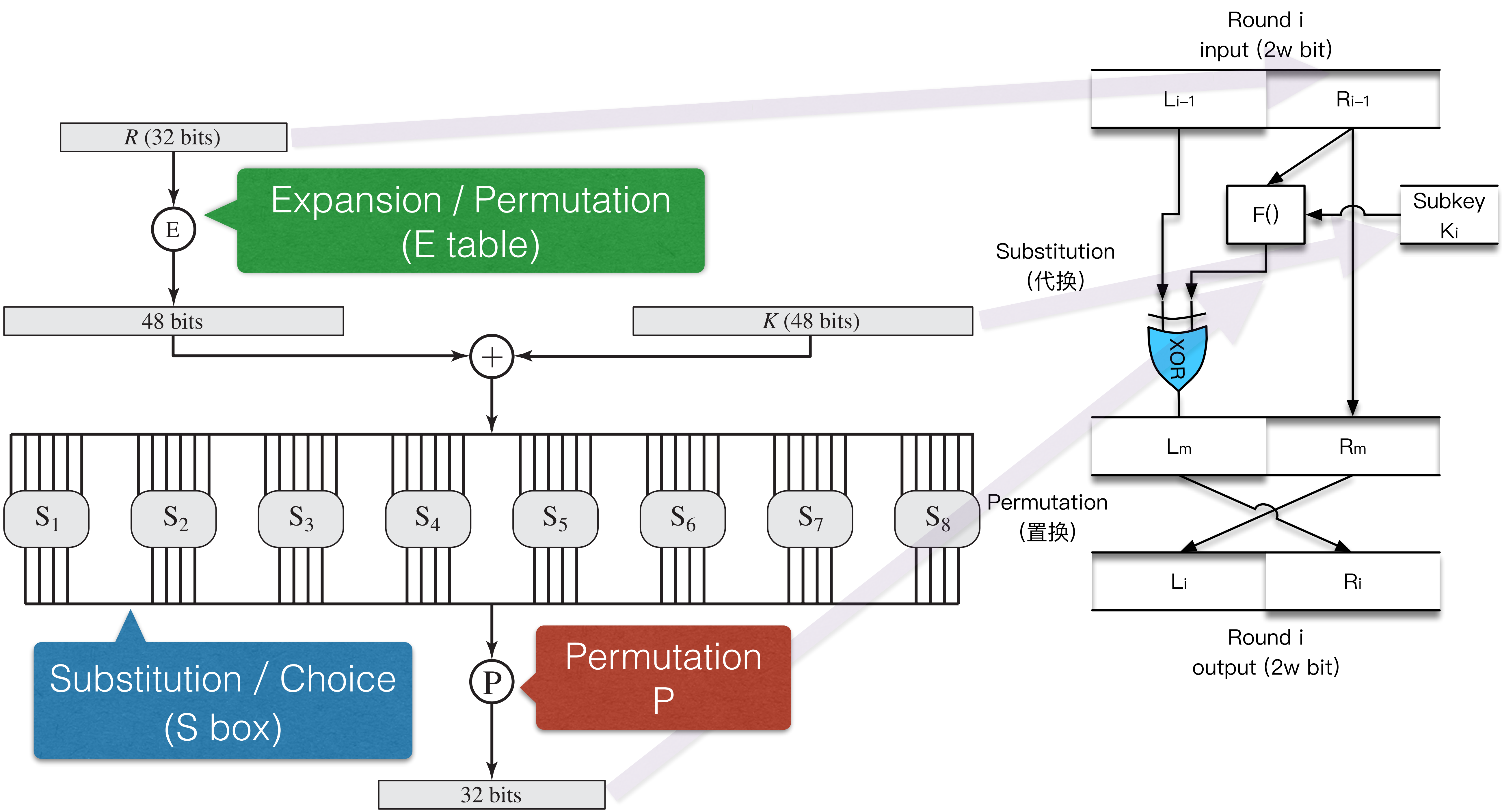




3. DES
Single Round

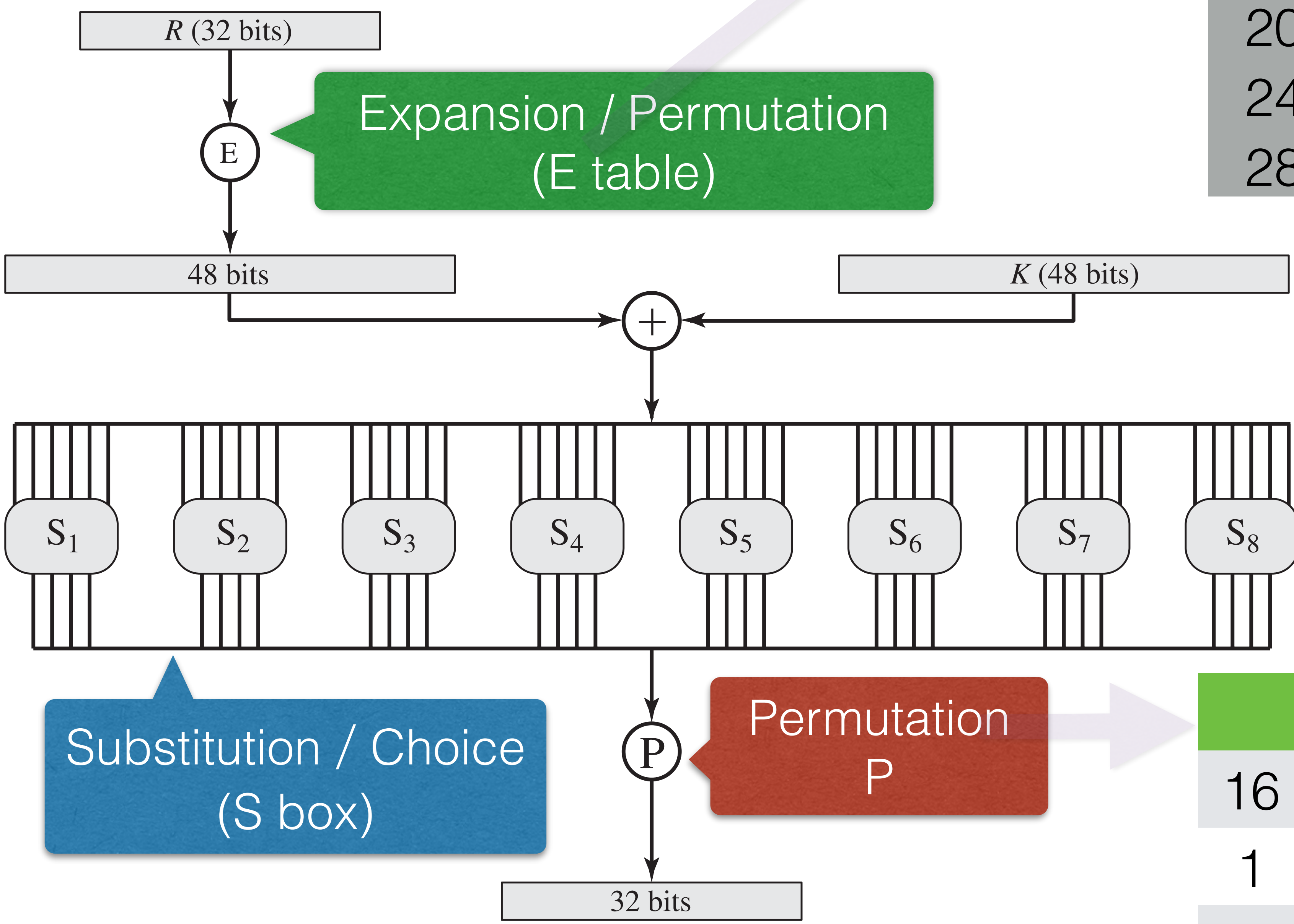


3. DES
Single Round



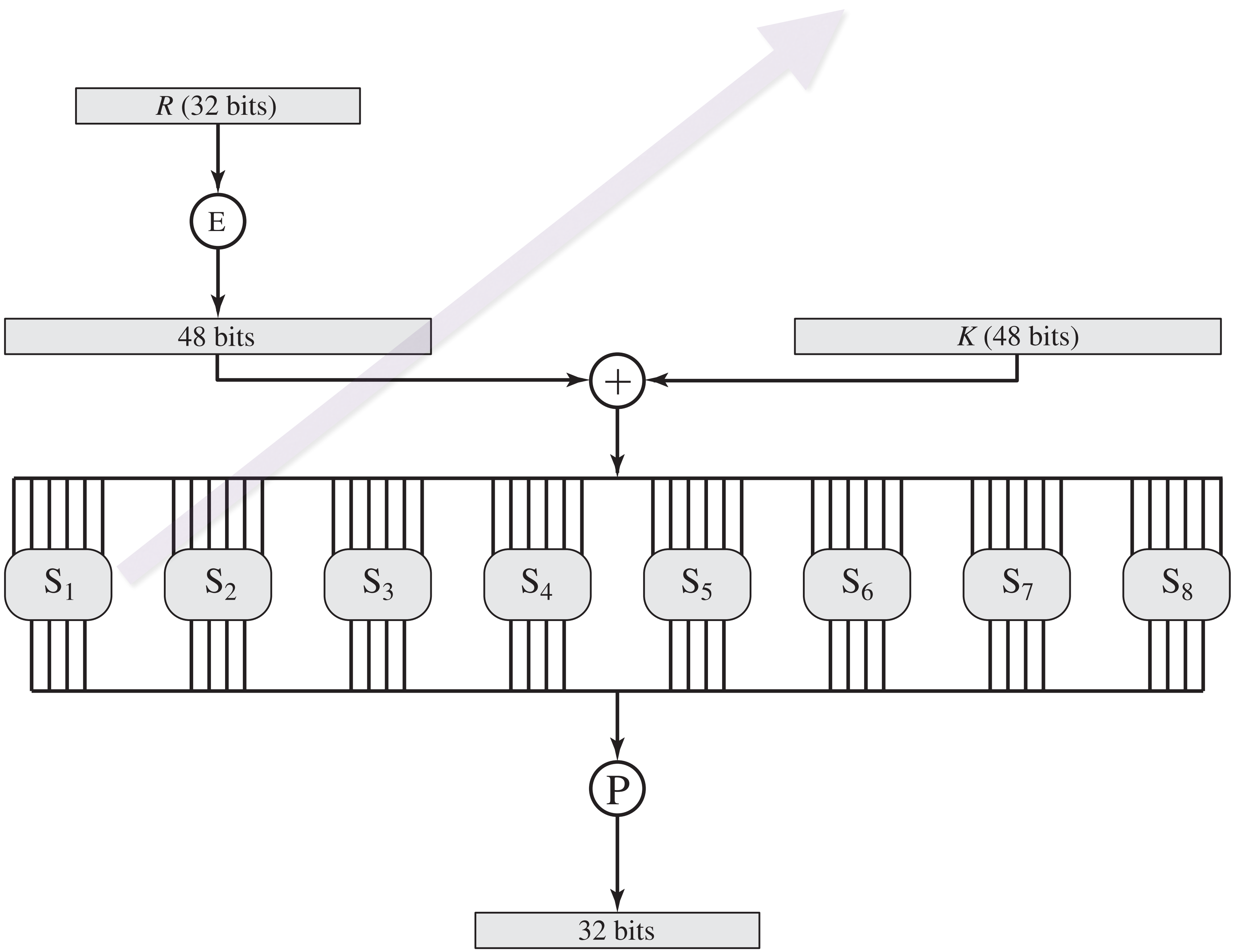
3. DES
Single Round

Expansion / Permutation (E table)					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1



Permutation P							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13



Example： input

0	1	1	0	0	1

0	1

=Row 1

1	1	0	0

=Col 12

Value: 9 Output:

1	0	0	1

4. The Strength of DES The Use of **56-Bit** Keys

- Brute-force attacks on 2^{56} possible keys $\simeq 7.2 \times 10^{16}$ keys
  - 1977, Diffie and Hellman, postulated that the technology existed to build a parallel machine with 1 million encryption devices
    - each of which could perform one encryption per microsecond [DIFF77]
    - average search time $\simeq$10 hours
    - $20 million in 1977 dollars
  - 1998, a special-purpose “DES cracker” machine that was built for less than $250,000
    - The attack took less than three days
- Resolution: AES and triple DES

4. The Strength of DES
The **Avalanche Effect** (**雪崩效应**)

- *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
- *Example*:
  - When a bit of the plaintext is changed
    - After just three rounds, 18 bits differ between the two blocks.
    - On completion, the two ciphertexts differ in *32 bit* positions.
  - When a bit of the key is changed
    - After just three rounds, 25 bits differ between the two blocks.
    - On completion, the two ciphertexts differ in *30 bit* positions.

4. The Strength of DES
Other issues

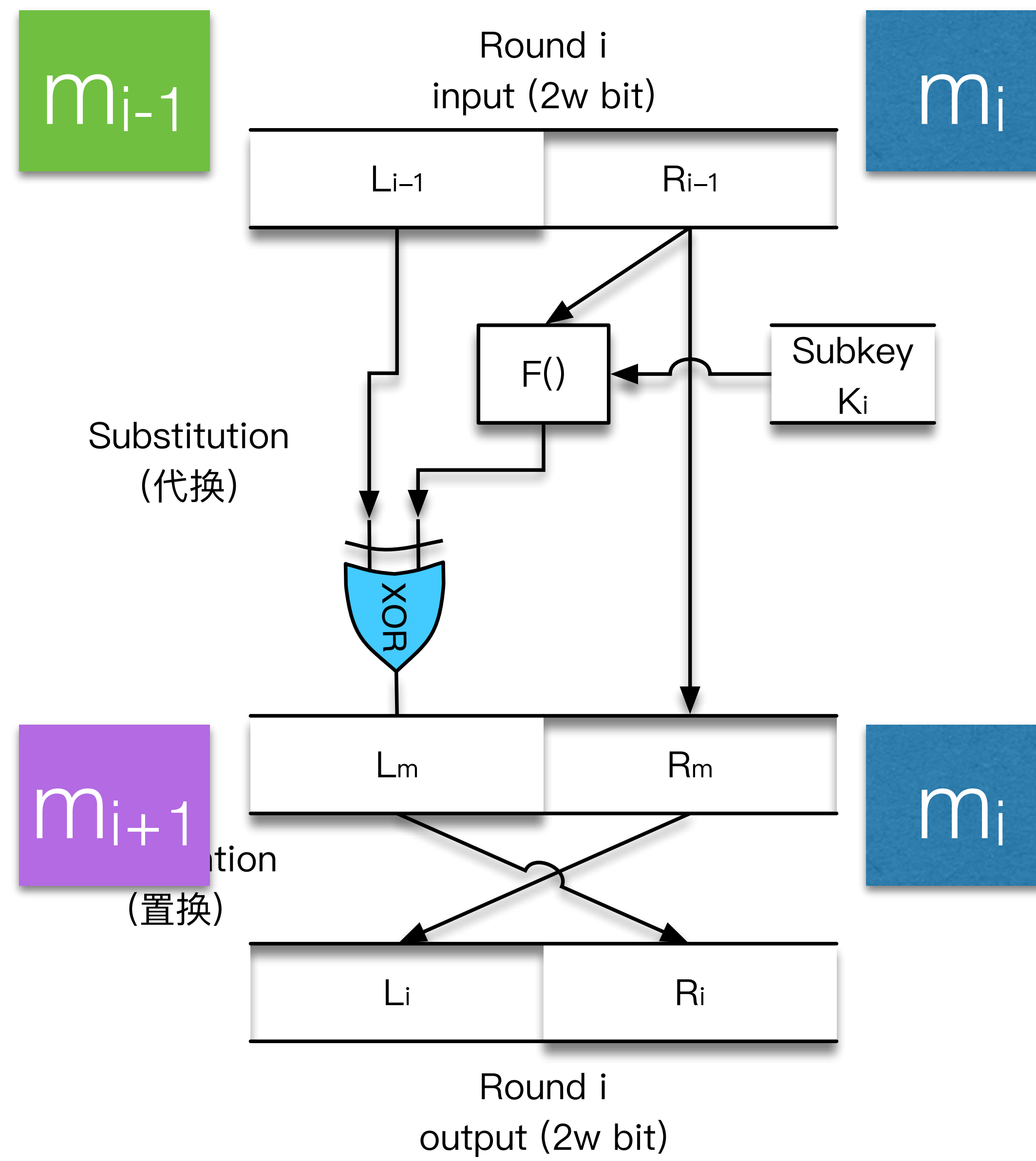
- The **Nature** of the DES Algorithm
  - Weakness of S-boxes?
    - Not discovered yet
- **Timing** attacks
  - *Definition*: information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertexts
  - Feature: This is a long way from knowing the actual key, but it is an intriguing first step.
  - DES is robust against Timing attacks

5. Differential and Linear Cryptanalysis
Motivation: **cryptanalytic attacks** on DES

- Search space:
  - brute-force attack: 2^{56}
  - Is there any attacks satisfying: $< 2^{56}$?
- two most powerful and promising approaches:
  - Differential Cryptanalysis
  - Linear Cryptanalysis

5. Differential and Linear Cryptanalysis
(1) Differential Cryptanalysis

- [BIHA93]: Input： 2^{47} **chosen plaintext**， Output： Key
- Method：
  - observe the behavior of pairs of text blocks evolving along each round of the cipher
  - instead of observing the evolution of a single text block



5. Differential and Linear Cryptanalysis
(2) Linear Cryptanalysis

- [MATS93]Input： 2^{43} **given plaintext,** Output： Key
- Method： finding linear approximations to describe the transformations.
  - n-bits plaintext represented as P[1], ..., P[n]
  - n-bits ciphertext represented as C[1], ..., C[n]
  - m-bits key represented K[1], ..., K[m],
  - Define： $A[i, j, ..., k] = A[i] \oplus A[j] \oplus ... \oplus A[k]$
  - Goal： find the following equations as many as possible：
    - $P[\alpha_1, \alpha_2, ..., \alpha_a] \oplus C[\beta_1, \beta_2, ..., \beta_b] = K[\gamma_1, \gamma_2, ..., \gamma_c]$
    - Solve the key using the equations

Homework

3.9 Show that DES decryption is, in fact, the inverse of DES encryption.

3.11 Compare the initial permutation table (Table 3.2a) with the permuted choice one table (Table 3.4b). Are the structures similar? If so, describe the similarities. What conclusions can you draw from this analysis?

3.12 When using the DES algorithm for decryption, the 16 keys (K_1, K_2, ..., K_{16}) are used in reverse order. Therefore, the right-hand side of Figure 3.5 is not valid for decryption. Design a key-generation scheme with the appropriate shift schedule (analogous to Table 3.4d) for the decryption process.

3.14 Show that in DES the first 24 bits of each subkey come from the same subset of 28 bits of the initial key and that the second 24 bits of each subkey come from a disjoint subset of 28 bits of the initial key.

huangwc@ustc.edu.cn