Ceng 471 Cryptography

Symmetrical Cryptosystems
DES

Asst. Prof. Dr. Serap ŞAHİN

Izmir Institute of Technology

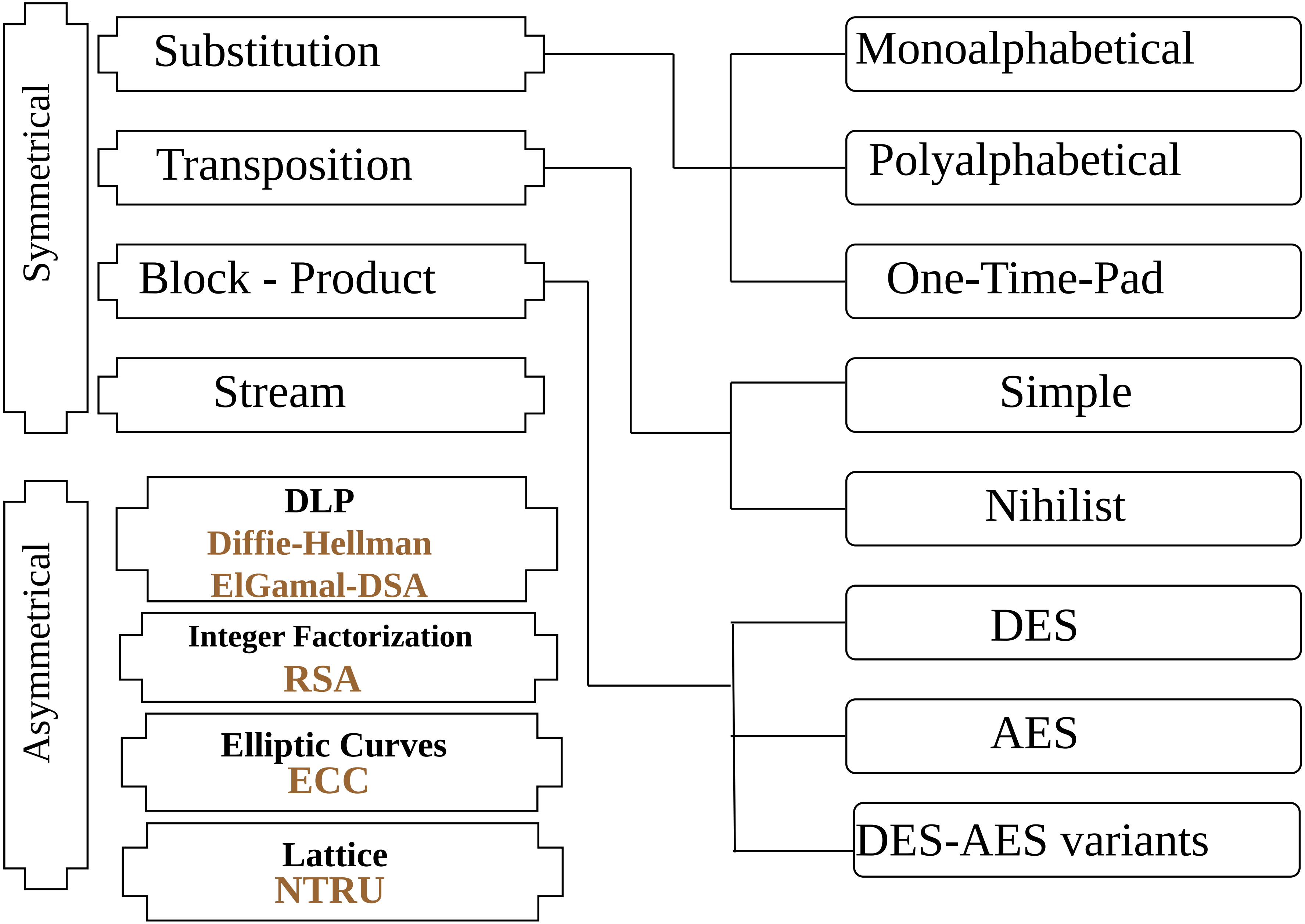
Reference Book: Introduction to Cryptography with Coding Theory 2nd Ed., W.Trappe, L. Washington

Reference Presentations:

Lecture slides of Assoc. Prof. Dr. Ahmet Koltuksuz for “Symmetric Cryptography”.

Lecture slides by Lawrie Brown for “ Cryptography and Network Security”, 5/e, by William Stallings, Chapter 3.

Introduction



Symmetrical

Substitution

Transposition

Block - Product

Stream

Monoalphabetical

Polyalphabetical

One-Time-Pad

Simple

Nihilist

DES

AES

DES-AES variants

Asymmetrical

DLP
Diffie-Hellman
ElGamal-DSA

Integer Factorization
RSA

Elliptic Curves
ECC

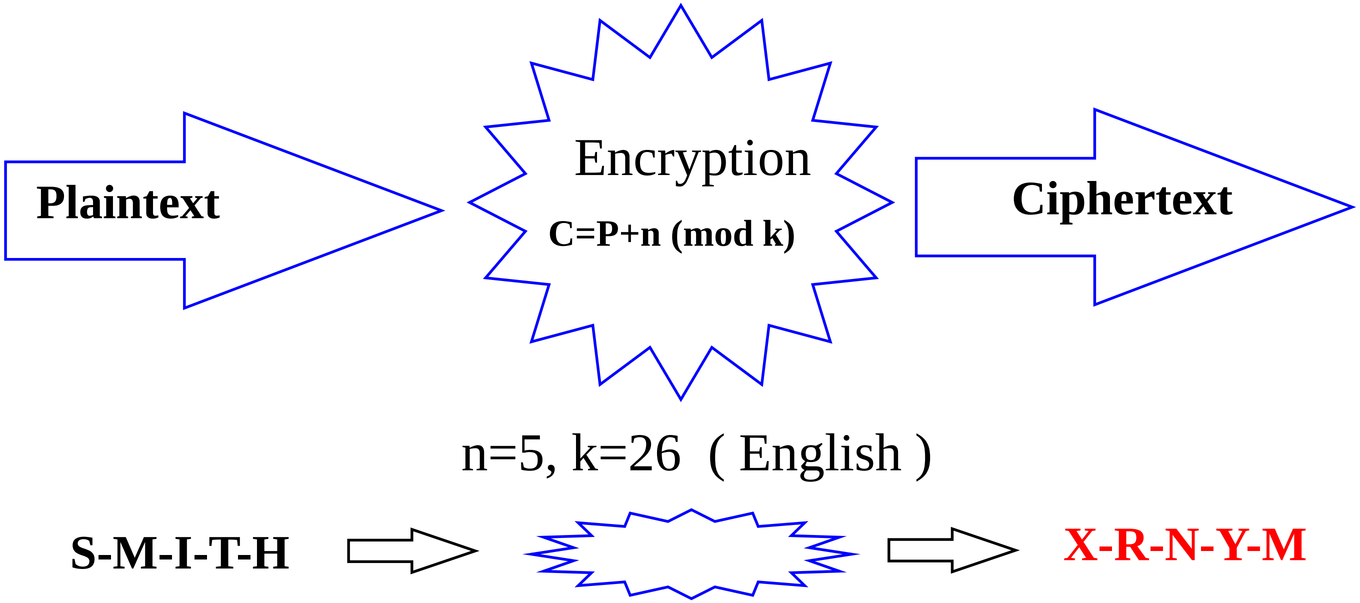
Lattice
NTRU

Symmetric Cryptography

Symmetrical

Substitution

Ceasar Cipher, 60 B.C.



Plaintext

Encryption

C=P+n (mod k)

Ciphertext

n=5, k=26 (English)

S-M-I-T-H

X-R-N-Y-M

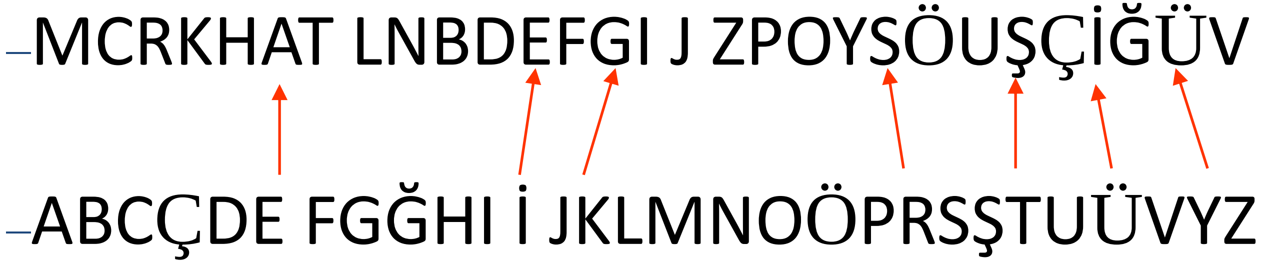
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Symmetric Cryptography

Symmetrical

Substitution

Monoalphabetical Substitution



- MCRKHAT LNBDEFGI J ZPOYSÖUŞÇİĞÜV
- ABCÇDE FGĞHI İ JKLMNOÖPRSŞTUÜVYZ

- Plaintext : TÜRKİYE
- Ciphertext : ŞİSGEÜA

Symmetric Cryptography

Symmetrical

Substitution

Polylphabetical Substitution: Vigenere Table

	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	
A	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	**00**
B	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	**01**
C	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	**02**
Ç	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	**03**
D	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	**04**
E	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	**05**
F	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	**06**
G	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	**07**
Ğ	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	**08**
H	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	**09**
I	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	**10**
İ	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	**11**
J	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	**12**
K	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	**13**
L	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	**14**
M	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	**15**
N	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	**16**
O	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	**17**
Ö	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	**18**
P	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	**19**
R	R	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	**20**
S	S	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	I	J	K	L	M	N	O	Ö	P	R	**21**
Ş	Ş	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	**22**
T	T	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	**23**
U	U	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	**24**
Ü	Ü	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	**25**
V	V	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	**26**
Y	Y	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	**27**
Z	Z	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	**28**

Plaintext:TÜRKİYE

T=00

Ü=01

R=02

K=03

İ=04

Y=05

E=06

Ciphertext:TVŞNMÇİ

Symmetric Cryptography

Symmetrical

Substitution

One Time Pad, Vernam Cipher, 1926

Step #1: Digitize the Alphabet

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
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

Step #2: Encryption

Plaintext	:	**H**	**E**	**L**	**L**	**O**
(1) Digitalization	:	**8**	**5**	**12**	**12**	**15**
(2) Random Numbers	:	**12**	**48**	**28**	**32**	**80**
Summation of 1 & 2	:	**20**	**53**	**40**	**44**	**95**
Modulus 26	:	**20**	**1**	**14**	**18**	**17**
Ciphertext	:	**T**	**A**	**N**	**R**	**Q**

Symmetric Cryptography

Symmetrical

Substitution

One Time Pad, Vernam Cipher, 1926

- The only mathematically proven unbreakable cipher. Proven by **Shannon**.
- With the conditions of :The length of the key should be equal to that of the length of the message.
- Each key should only be used once.
- Thus very popular with the intelligence agencies.

Symmetric Cryptography

Symmetrical

Transposition:
Permutation

Simple Permutation

Plaintext	: KRİPTOGRAFİ	
Index	: 1 2 3 4 5 6	
Permutation (key)	: 4 3 6 2 5 1	
Encryption	: K R İ P T O	G R A F İ -
	1 2 3 4 5 6	1 2 3 4 5 6
	4 3 6 2 5 1	4 3 6 2 5 1
Ciphertext	: P İ O R T K	F A - R İ G

Symmetric Cryptography
Modern Block Ciphers

- now look at modern block ciphers
- one of the most widely used types of cryptographic algorithms
- provide secrecy services
- focus on DES (Data Encryption Standard)

  to illustrate block cipher design principles

DES History

- IBM developed Lucifer cipher
  - by team led by Horst Feistel in late 60's
  - used 64-bit data blocks with 128-bit key
- then redeveloped as a commercial cipher with input from NSA and others
- in 1973 NBS issued request for proposals for a national cipher standard for unclassified computer data
- IBM submitted their revised Lucifer which was eventually accepted as the DES

Data Encryption Standard (DES)

- Used in most EFT and EFTPOS from banking industry
  - It was reconfirmed as a standard for 5 years twice
  - Currently 3DES is recommended
- DES became a federal standard in November 76
  - adopted in 1977 by National Bureau of Standards - NBS (now NIST – National Institutes of Standards and Technologies)
    - As Federal Information Processing Standard 46 - FIPS PUB 46
  - ANSI X3.92-1981 (hardware + software)
  - ANSI X3.106-1983 (modes of operation)
  - Australia AS2805.5-1985
- encrypts 64-bit data using 56-bit key

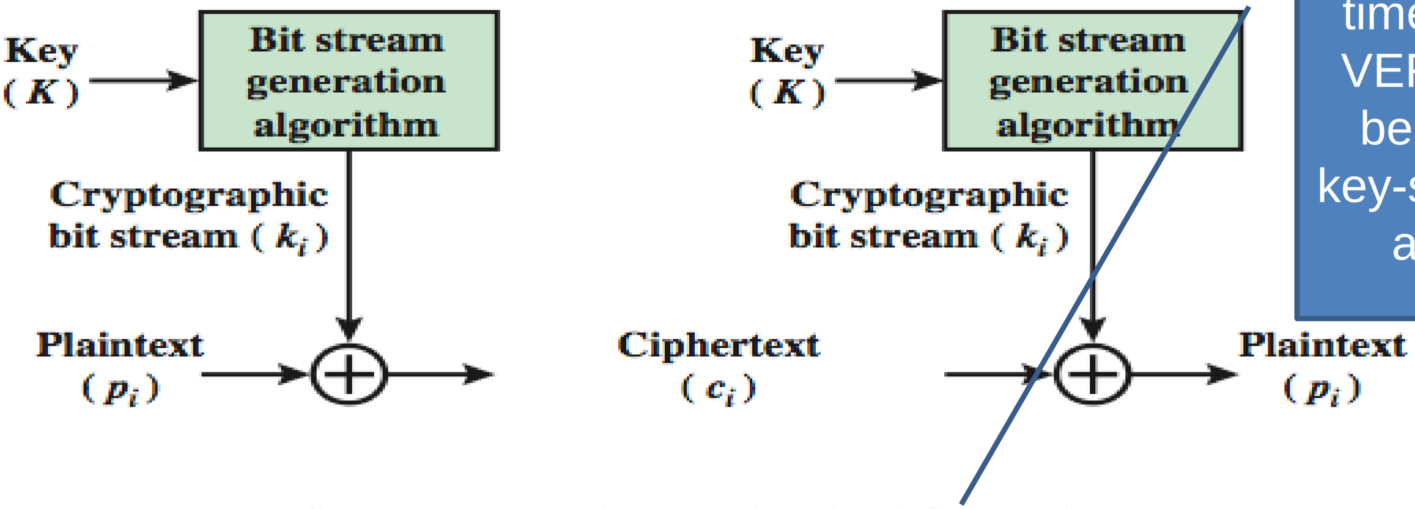
DES Design Criteria

- The standard is public, the design criteria is classified
- One of the biggest controversies is the key size (56 bits)
  - W Diffie, M Hellman "Exhaustive Cryptanalysis of the NBS Data Encryption Standard" IEEE Computer 10(6), June 1977, pp74-84
  - M Hellman "DES will be totally insecure within ten years" IEEE Spectrum 16(7), Jul 1979, pp 31-41
- Another controversy: is there a back door?

Block vs Stream Ciphers

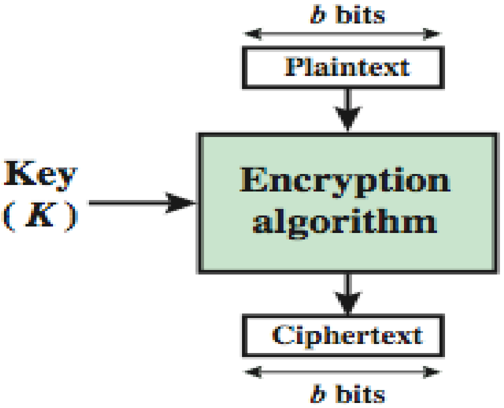
- Block ciphers process messages in blocks, each of which is then en/decrypted
- Like a substitution on very big characters
  - 64-bits or more
- Stream ciphers process messages a bit or byte at a time when en/decrypting
- Many current ciphers are block ciphers
  - better analyzed
  - broader range of applications

Block vs Stream Ciphers


Key
(K)
Bit stream generation algorithm
Cryptographic bit stream (k_i)
Plaintext
(p_i)
Key
(K)
Bit stream generation algorithm
Cryptographic bit stream (k_i)
Ciphertext
(c_i)
Plaintext
(p_i)

In the ideal case, a one-time pad version of the VERNAM cipher would be used, in which the key-stream (k) is as long as the plaintext bit stream (p).

(a) Stream Cipher Using Algorithmic Bit Stream Generator



(b) Block Cipher

Block Cipher Principles

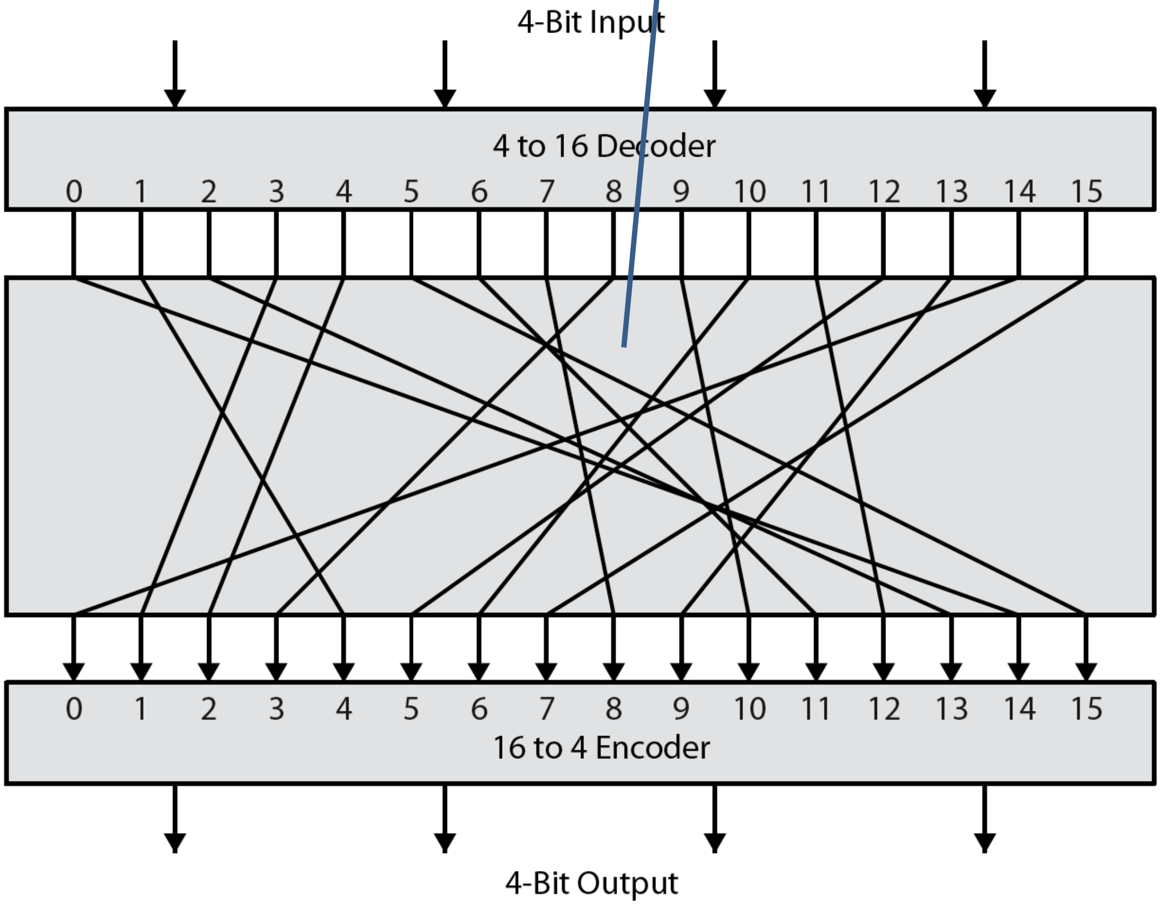
- A block cipher operates on a plaintext block of n bits to produce a ciphertext block of n bits,
- most symmetric block ciphers are based on a **Feistel Cipher Structure.**
- needed since must be able to **decrypt** ciphertext to recover messages efficiently
- block ciphers look like an extremely large substitution
- would need table of 2^{64} entries for a 64-bit block
  - In general, for an *n*-bit general substitution block cipher, the size of the key is $n \times 2^n$. For a 64-bit block, which is a desirable length to thwart statistical attacks, the key size is $64x\ 2^{64} = 2^{70} = 10^{21}$ bits.
- instead create from smaller building blocks
- using idea of **a product cipher**

Ideal Block Cipher

The encryption and decryption mappings can be defined by a tabulation

A 4-bit input produces one of 16 possible input states, which is mapped by the substitution cipher into a unique one of 16 possible output states, each of which is represented by 4 ciphertext bits.

It illustrates a tiny 4-bit substitution to show that each possible input can be arbitrarily mapped to any output - which is why its complexity grows so rapidly.



The encryption and decryption mappings can be defined by a tabulation, as shown in this Figure.

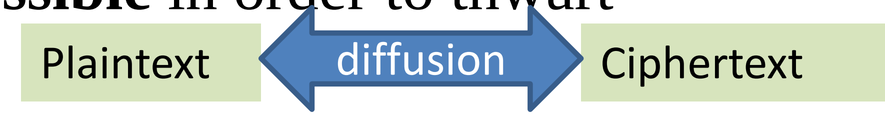
Claude Shannon and Substitution-Permutation Ciphers

- The concept of a product cipher, which is the execution of two or more simple ciphers in sequence in such a way that the final result or product is cryptographically stronger than any of the component ciphers.
- Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper.
- This form basis of modern block ciphers.
- S-P nets are based on the two primitive cryptographic operations seen before:
  - *substitution* (S-box)
  - *permutation* (P-box)

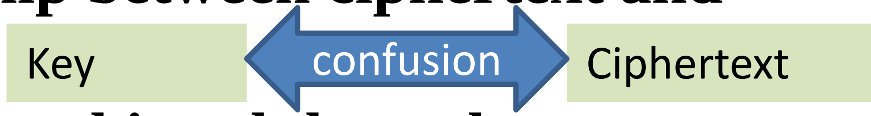
  Critically, it was the technique of layering groups of S-boxes separated by a larger P-box to form the S-P network, a complex form of a product cipher.
- provide *confusion* & *diffusion* of message & key

Confusion and Diffusion

- Cipher needs to completely obscure statistical properties of original message,
- A one-time pad does this.
- More practically Shannon suggested combining S & P elements to obtain:
  - **diffusion** – The mechanism of diffusion seeks to **make the statistical relationship between the plaintext and ciphertext as complex as possible** in order to thwart attempts to deduce the key.



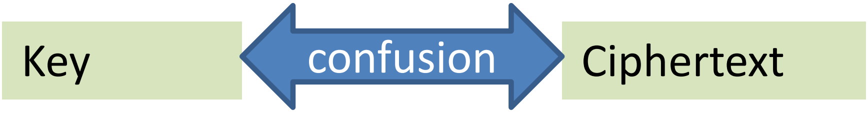
  - **confusion** – makes **relationship between ciphertext and key as complex as possible**



- At the simplest level, **diffusion is achieved through numerous permutations** and **confusions is achieved through the XOR operation**.

Confusion and Diffusion

- Good confusion can only be achieved



  - when each character of the ciphertext depends on several parts of the key, and
  - this dependence appears to be random to the observer.

Ciphers that do not offer much confusion (such as Vigen`ere cipher) are vulnerable to frequency analysis.

Confusion and Diffusion

Plaintext

diffusion

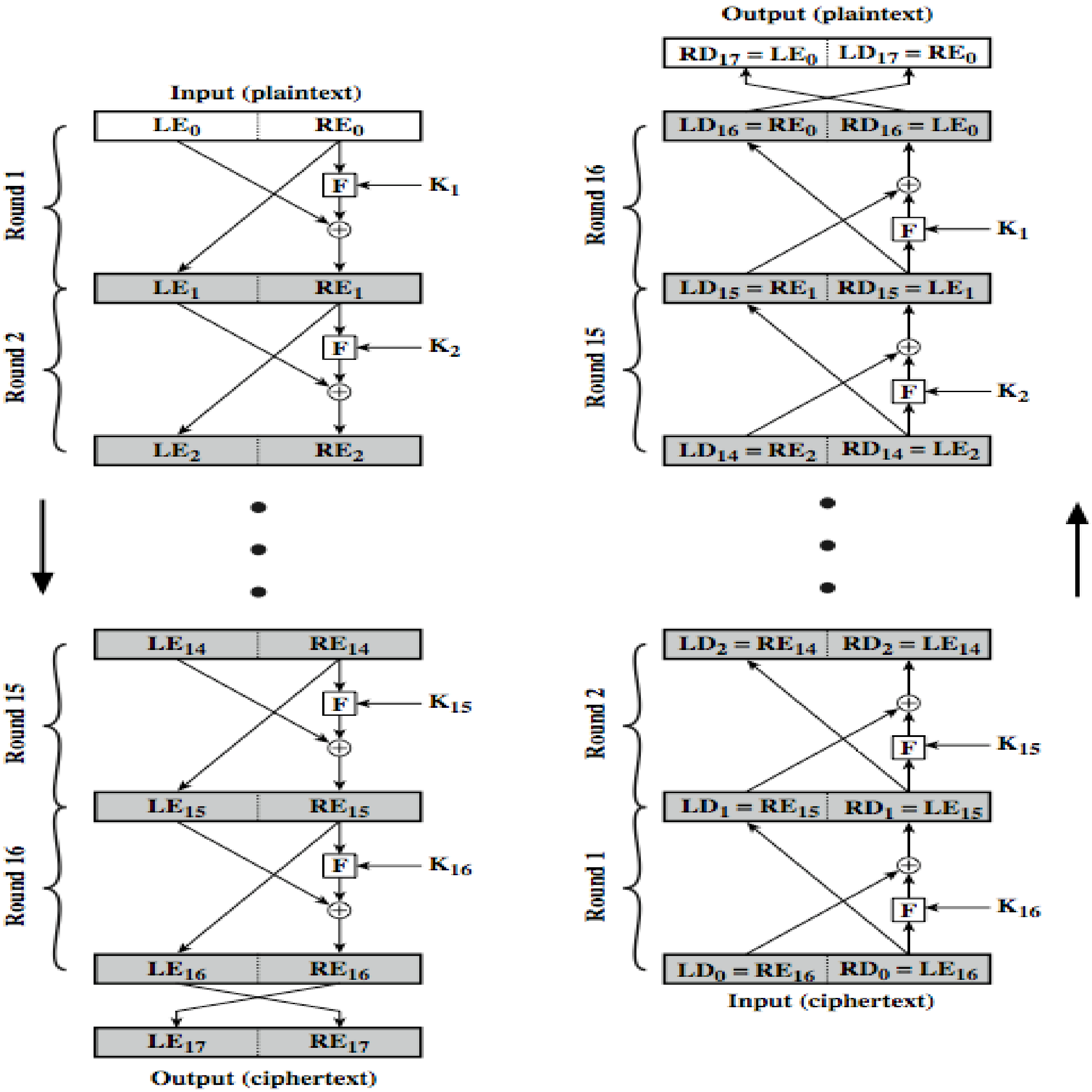
Ciphertext

- Good diffusion spreads the influence of a single plaintext letter over many ciphertext letters.
  - In terms of the frequency statistics of letters, diagrams, etc. in the plaintext, diffusion randomly spreads them across several characters in the ciphertext.
  - This means that much more ciphertexts are needed to do a meaningful statistical attack on the cipher.

Feistel Cipher Structure

- Horst Feistel; working at IBM Thomas J Watson Research Labs devised the **feistel cipher** (early 70`s)
  - **His main contributions** was the invention of a suitable structure which adapted Shannon's S-P network in an easily inverted structure.
- partitions input block into two halves
  - process through multiple rounds which
  - perform a substitution on left data half
  - based on round function of right half & subkey
  - then have permutation swapping halves
- Essentially the same h/w or s/w is used for both encryption and decryption, with just a slight change in how the keys are used. One layer of S-boxes and the following P-box are used to form the round function.

Feistel Cipher Structure

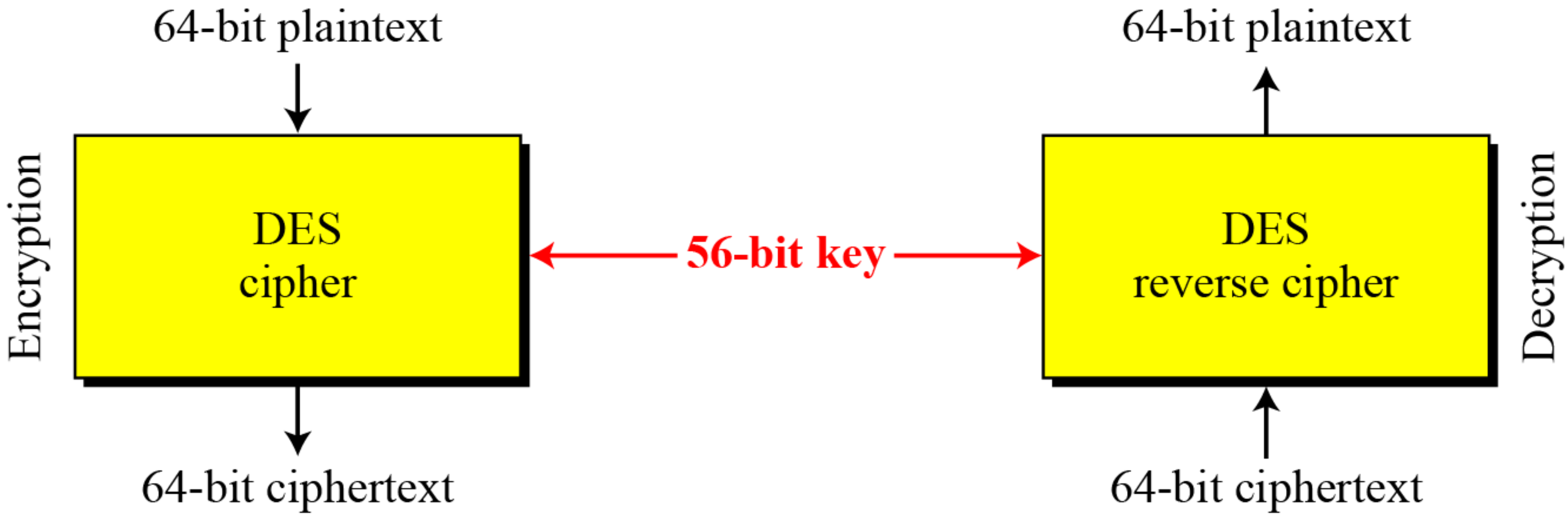


Feistel Cipher Design Elements

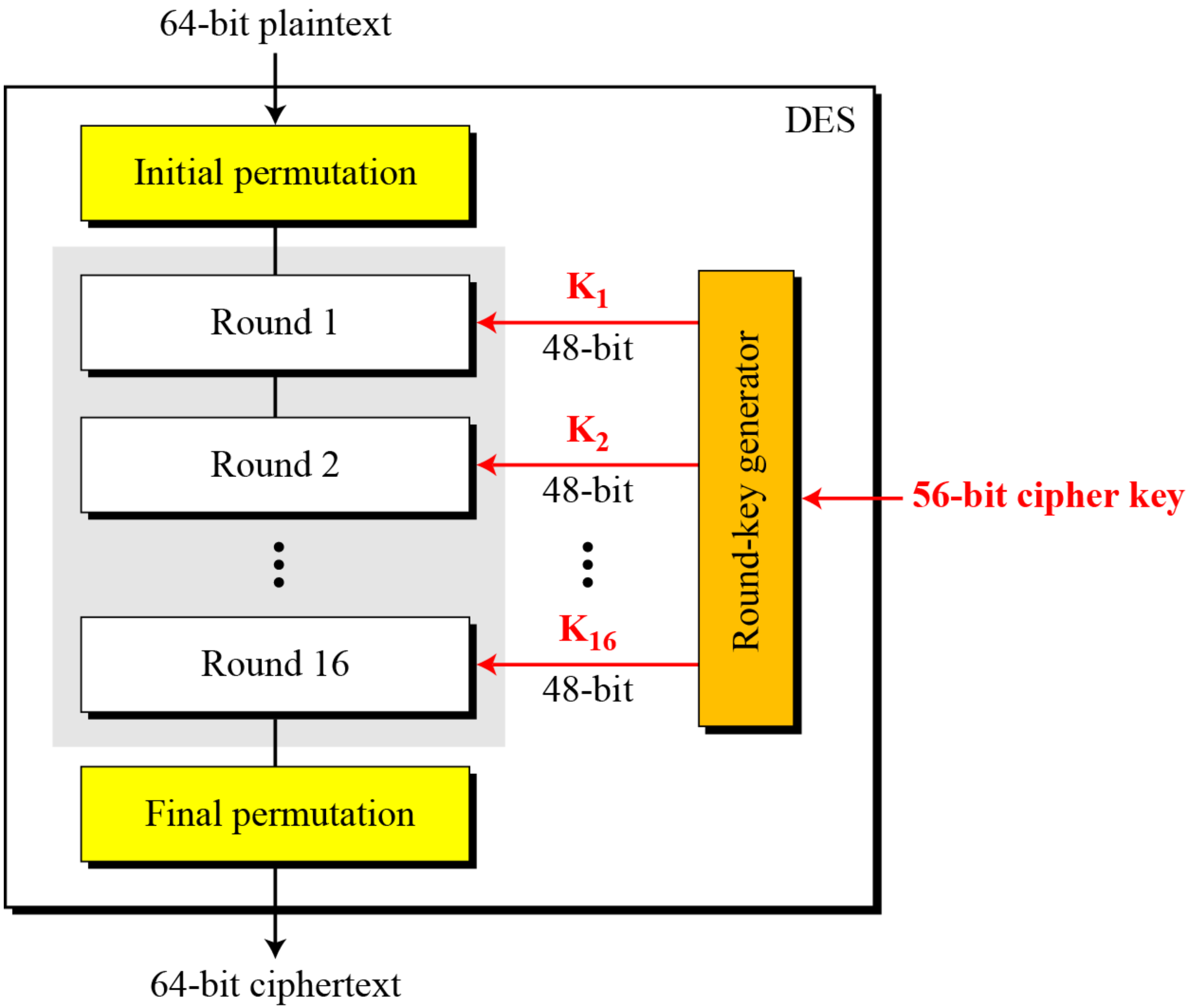
- **block size**; - increasing size improves security, but slows cipher
- **key size;** increasing size improves security, makes exhaustive key searching harder, but may slow cipher
- **number of rounds;** increasing number improves security, but slows cipher
- **subkey generation algorithm;** greater complexity can make analysis harder, but slows cipher
- **round function;** greater complexity can make analysis harder, but slows cipher
- **fast software en/decryption;** more recent concern for practical use
- **ease of analysis;** for easier validation & testing of strength

DES is a block cipher

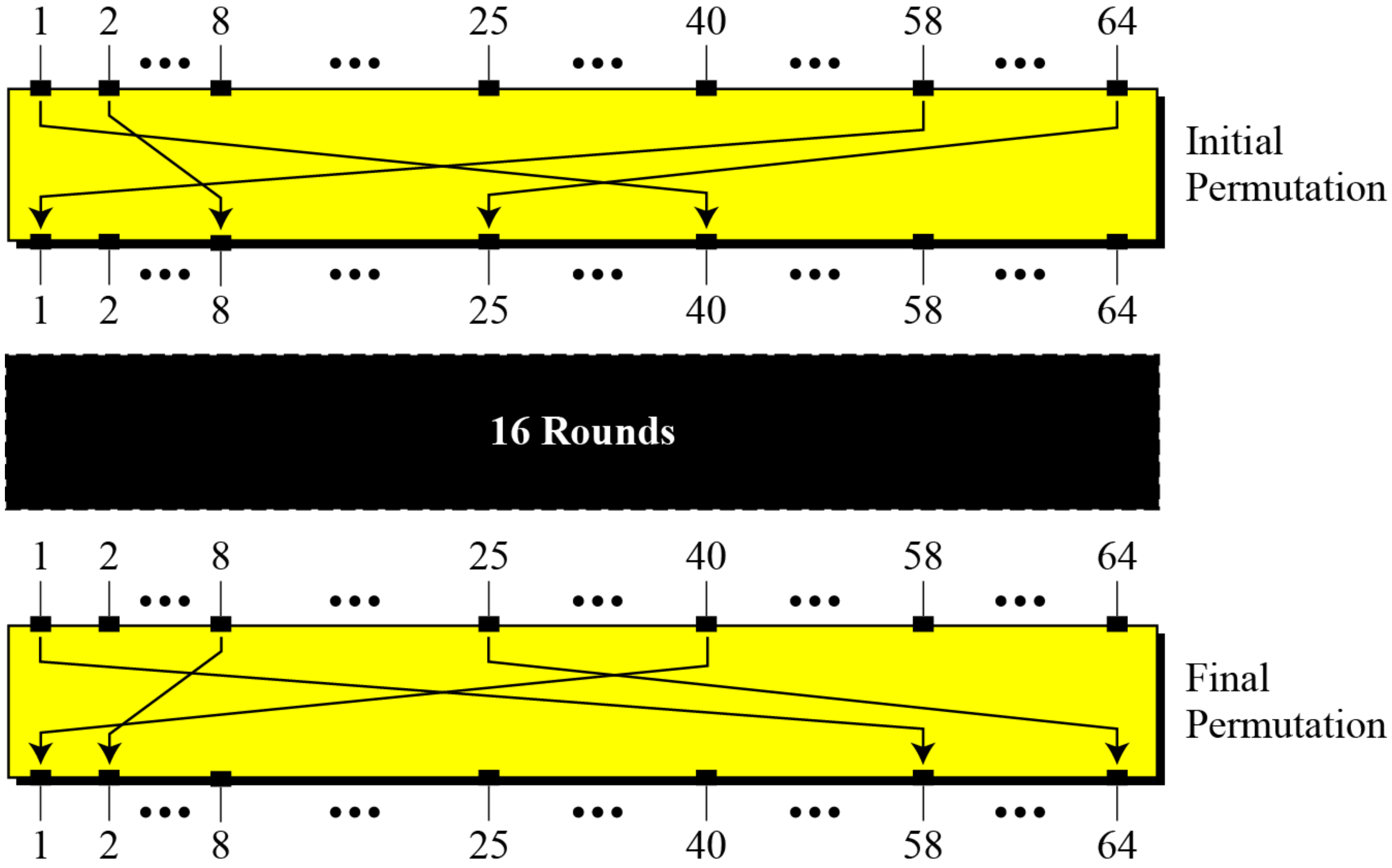
- *Encryption and decryption with DES*



General structure of DES



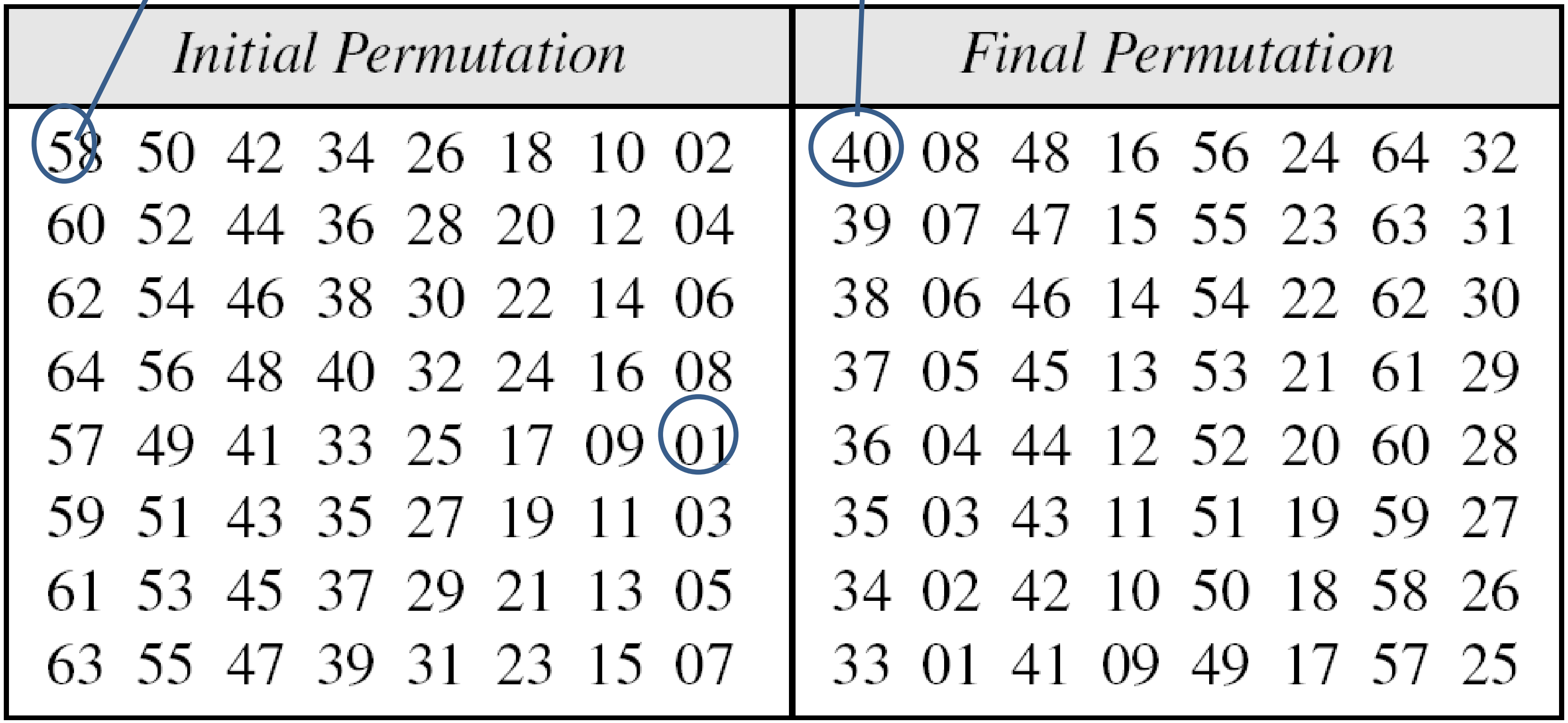
Initial and final permutation steps in DES



Initial and final permutation tables

1st bit is permuted to 40th bit position.
58th bit is permuted to 1st bit position

40th bit is permuted to 1th bit position.



Initial Permutation								*Final Permutation*							
58	50	42	34	26	18	10	02	40	08	48	16	56	24	64	32
60	52	44	36	28	20	12	04	39	07	47	15	55	23	63	31
62	54	46	38	30	22	14	06	38	06	46	14	54	22	62	30
64	56	48	40	32	24	16	08	37	05	45	13	53	21	61	29
57	49	41	33	25	17	09	01	36	04	44	12	52	20	60	28
59	51	43	35	27	19	11	03	35	03	43	11	51	19	59	27
61	53	45	37	29	21	13	05	34	02	42	10	50	18	58	26
63	55	47	39	31	23	15	07	33	01	41	09	49	17	57	25

Example 1/a

Find the output of the initial permutation box when the input is given in hexadecimal as:

0x0000 0080 0000 0002

Solution

Only bit 25 and bit 63 are 1s; the other bits are 0s. In the final permutation, bit 25 becomes bit 64 and bit 63 becomes bit 15. The result is

0000	0000
0000	0000
0000	0000
1000	0000
0000	0000
0000	0000
0000	0000
0000	0010

0000	0000
0000	0010
0000	0000
0000	0000
0000	0000
0000	0000
0000	0000
0000	0001

Initial Permutation							
58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

0x0002 0000 0000 0001

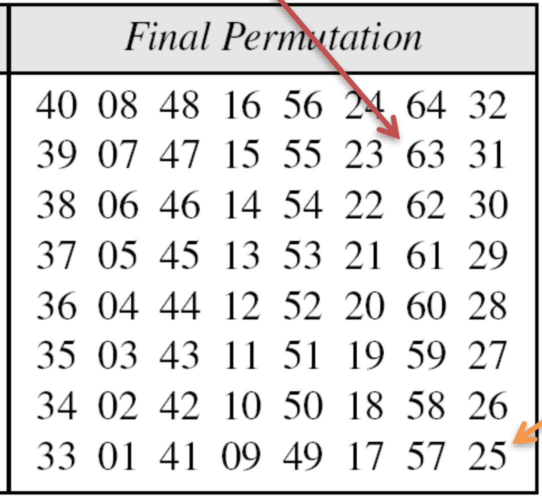
Example 1/b

Prove that the initial and final permutations are the inverse of each other by finding the output of the final permutation if the input is

0x0002 0000 0000 0001

Solution

The input has only two 1s; the output must also have only two 1s. Using tables, we can find the output related to these two bits. Bit 15 in the input becomes bit 63 in the output. Bit 64 in the input becomes bit 25 in the output. So the output has only two 1s, bit 25 and bit 63. The result in hexadecimal is



Final Permutation

40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25

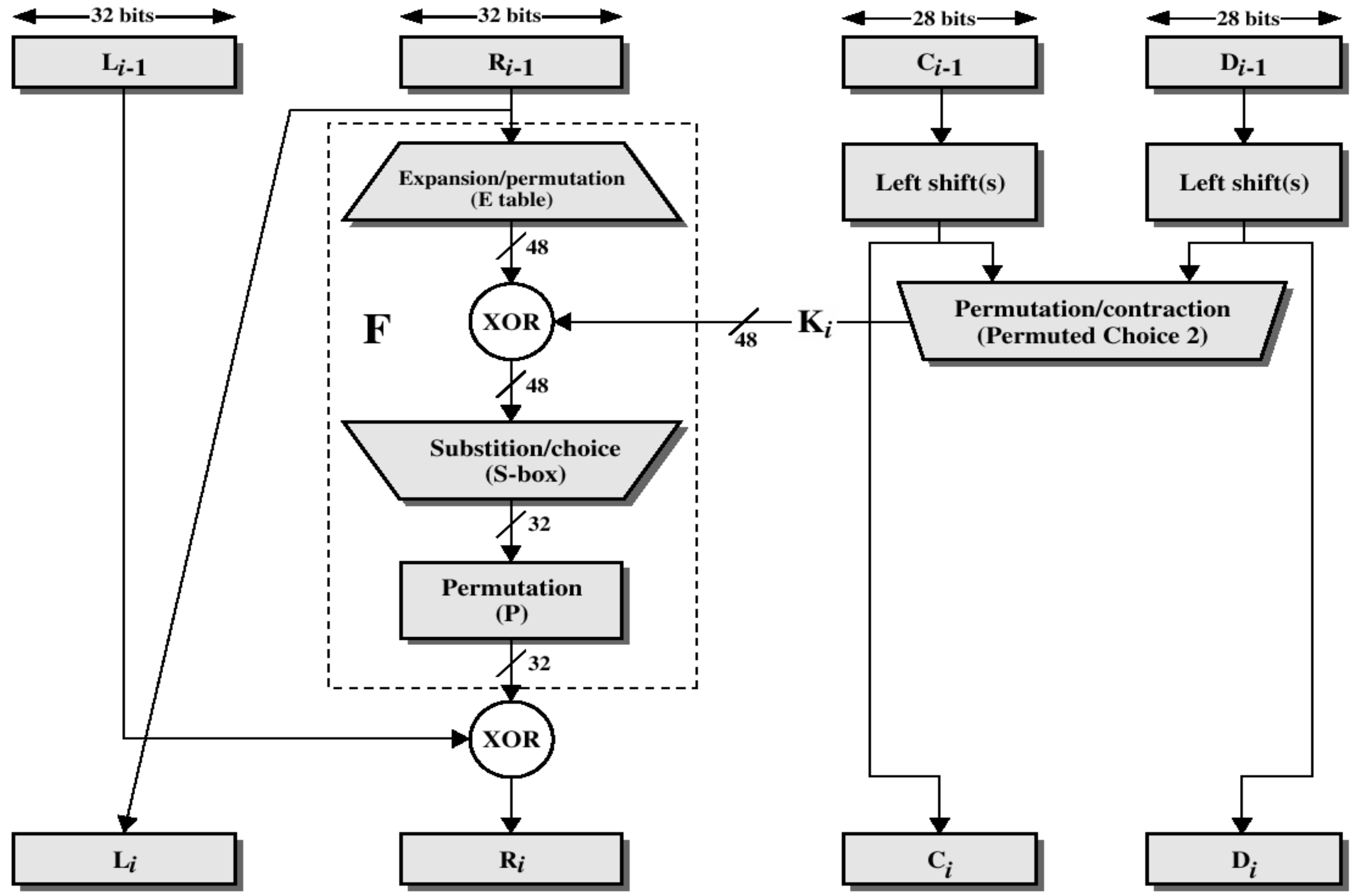
0x0000 0080 0000 0002

Note

The initial and final permutations are straight P-boxes that are inverses of each other.
They have no cryptography significance in DES.

Symmetrical

Block:DES-1977

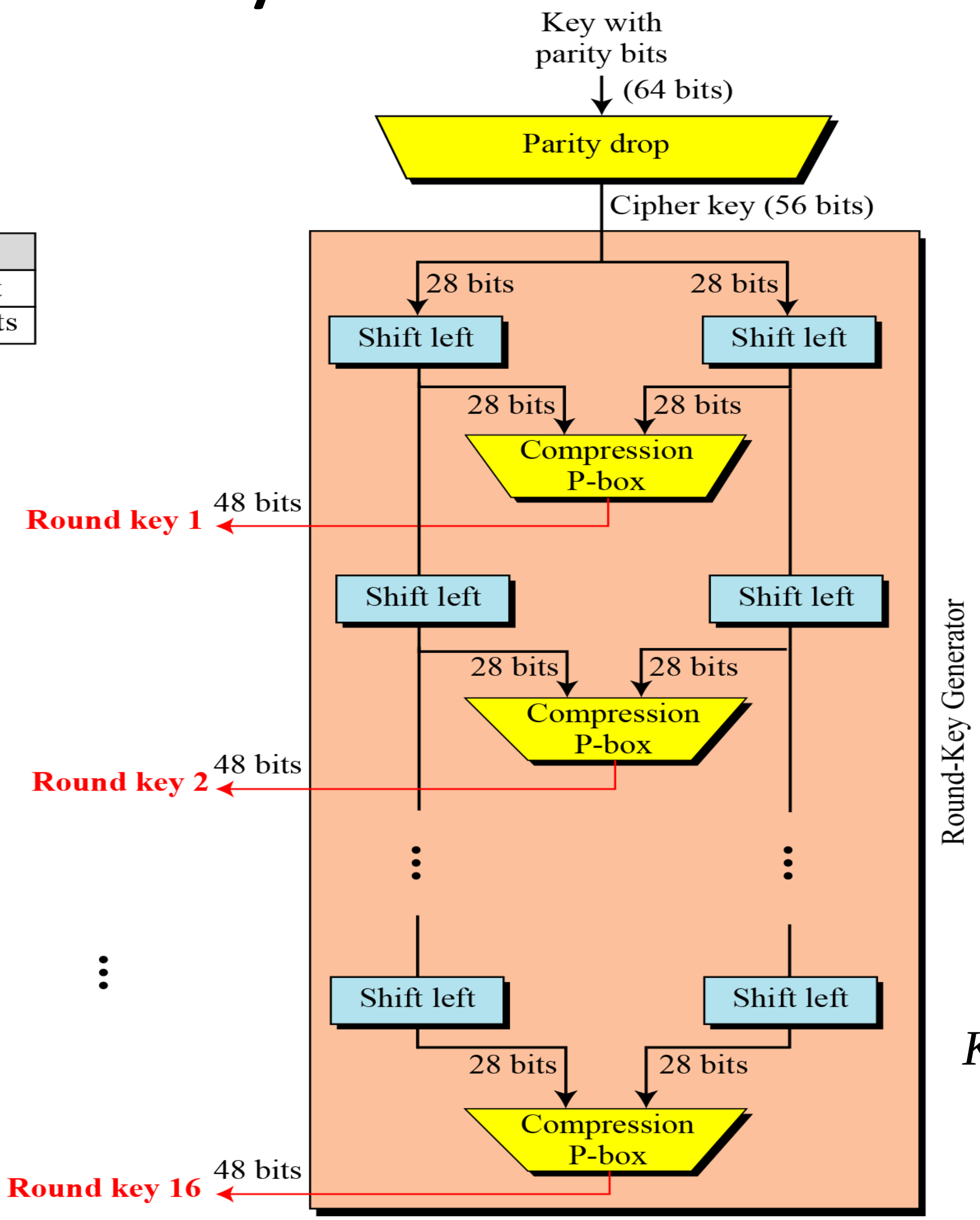


Single Round of DES Algorithm

Key Generation

Shifting

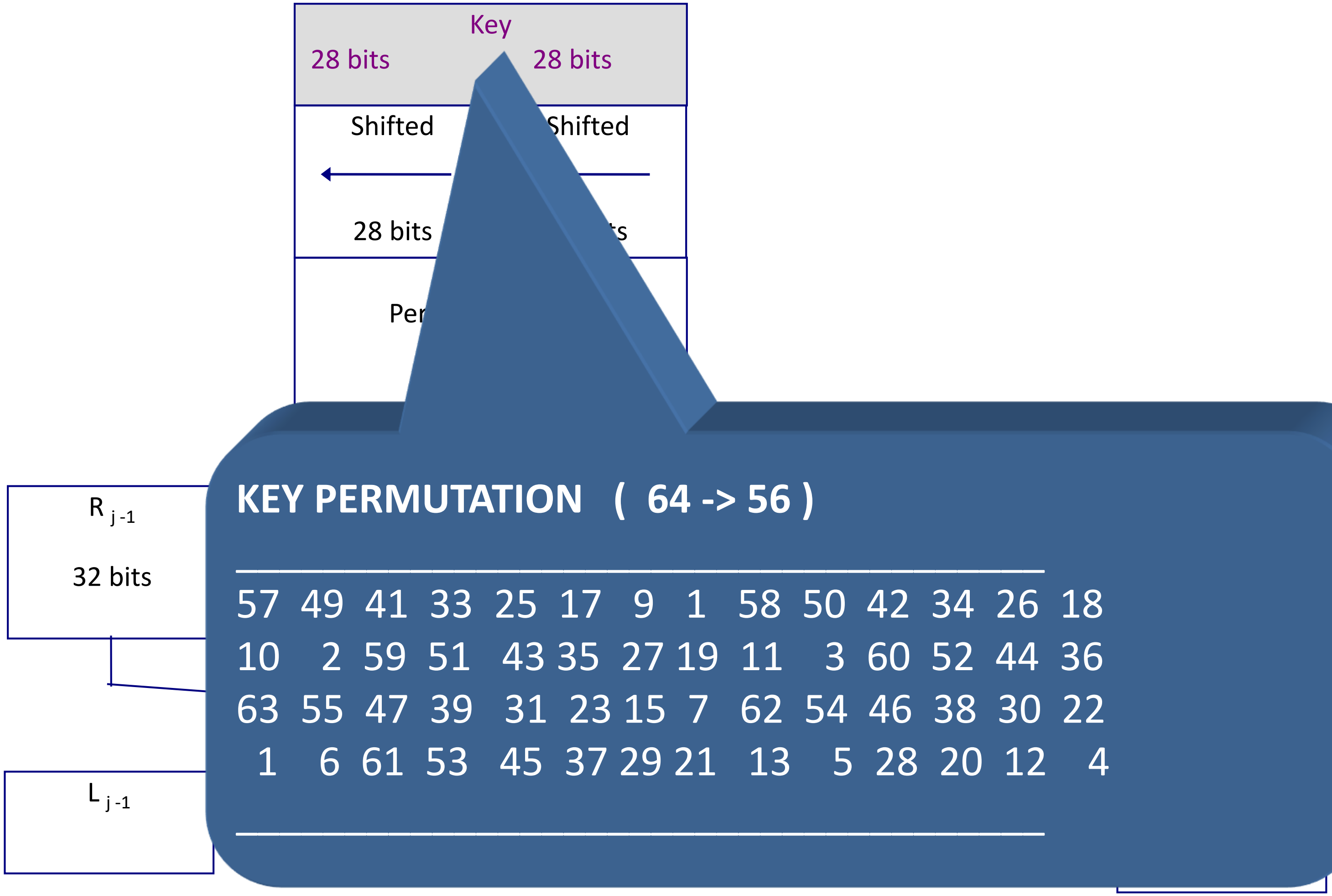
Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits



Key generation

Symmetrical

Block:DES-1977

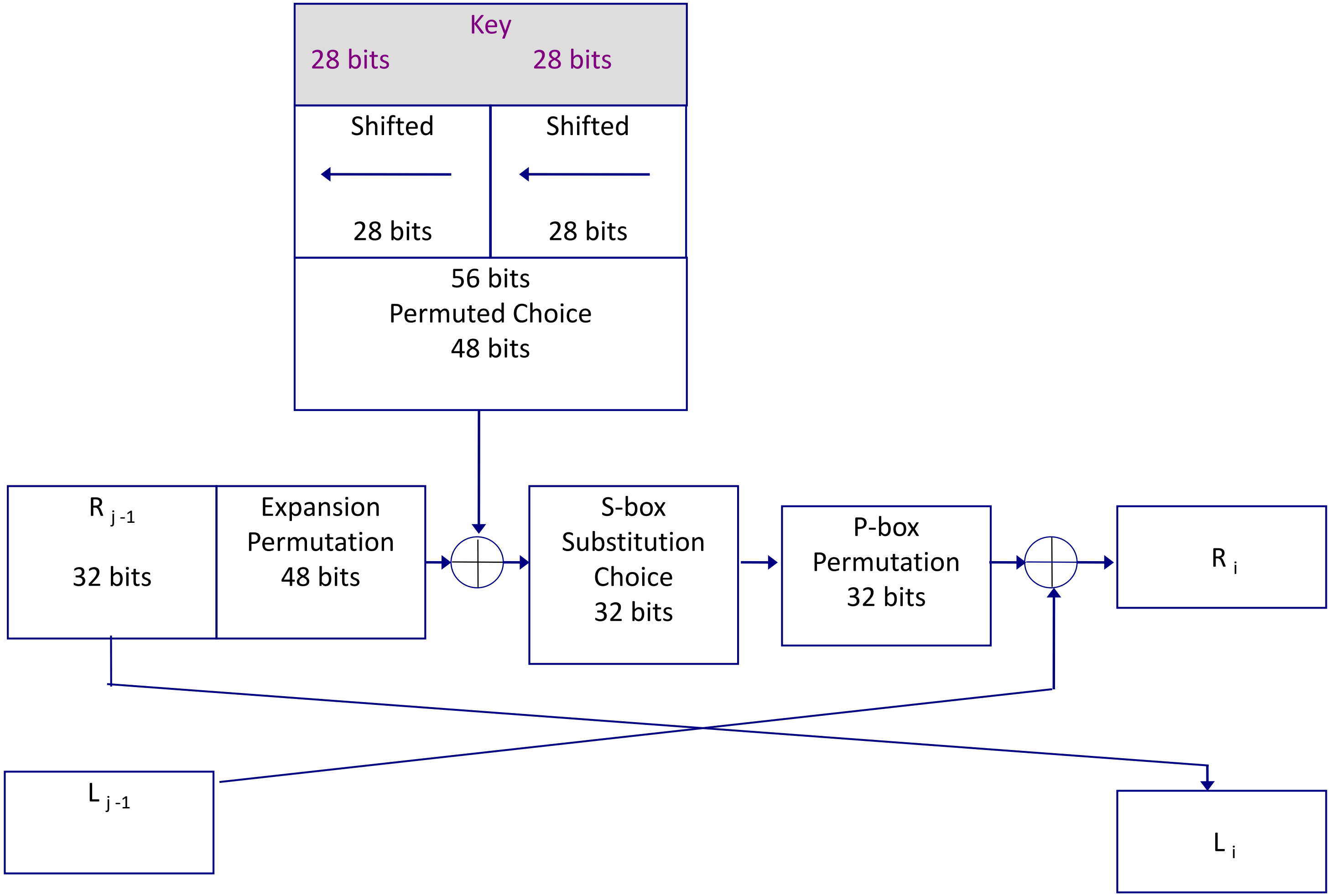


KEY PERMUTATION (64 -> 56)

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
1	6	61	53	45	37	29	21	13	5	28	20	12	4

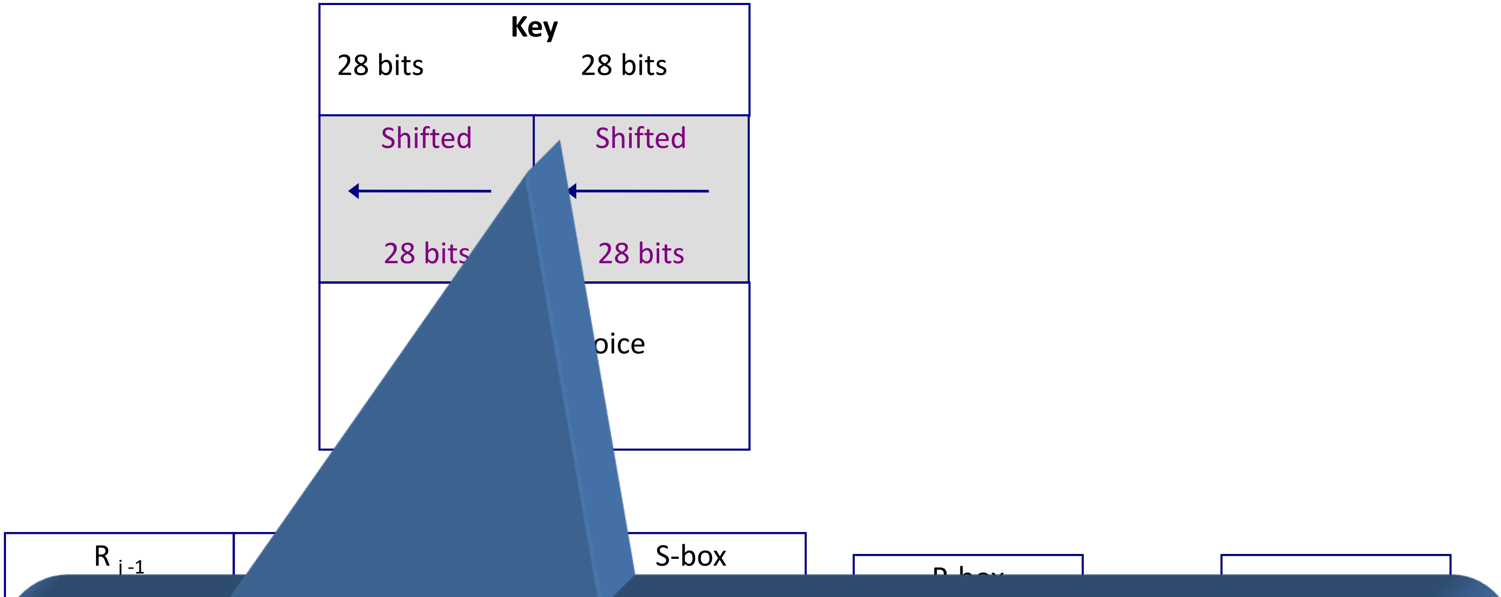
Symmetrical

Block:DES-1977



Symmetrical

Block:DES-1977

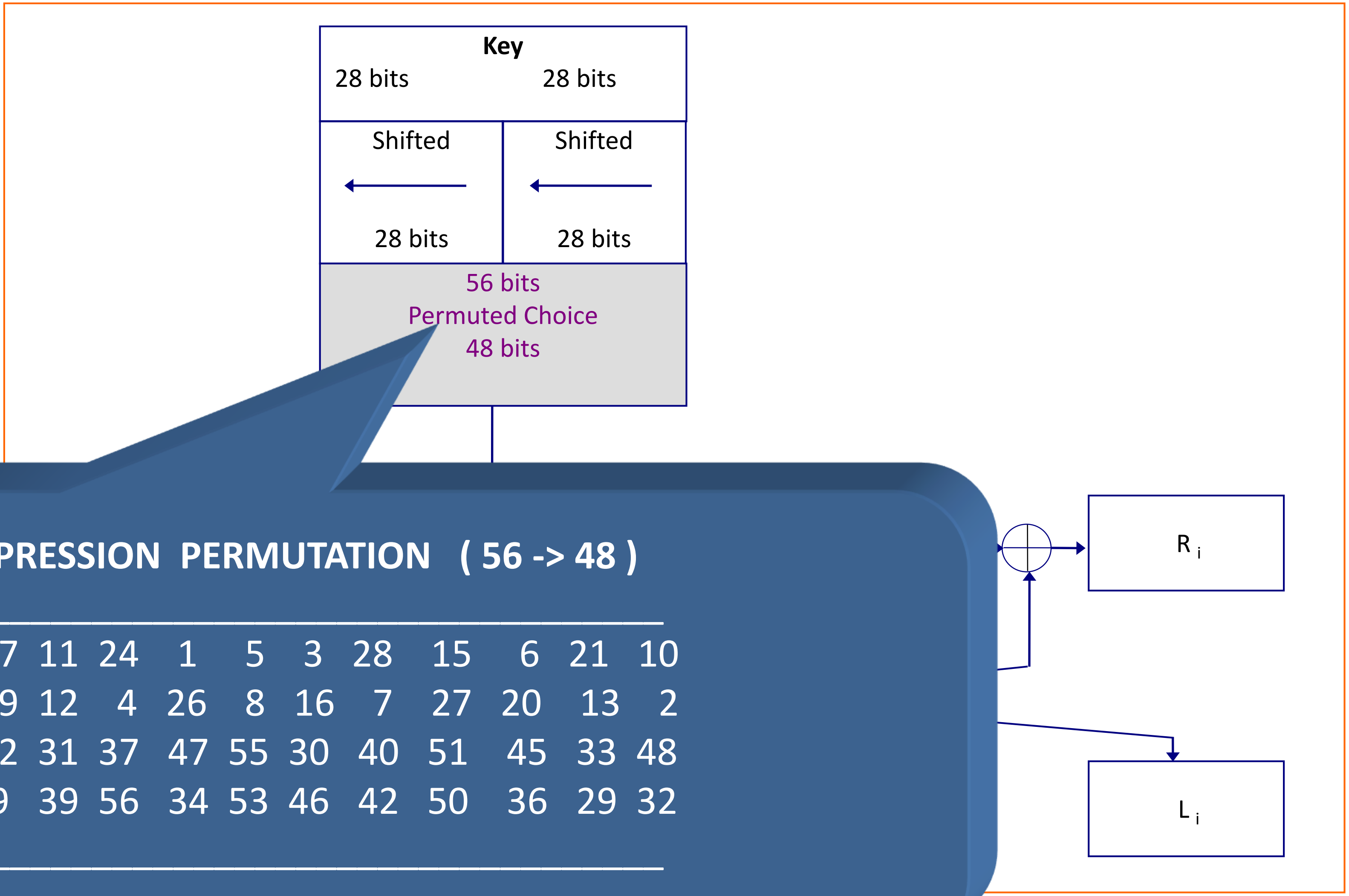


KEY SHIFTS PER ROUND

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
# of shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Symmetrical

Block:DES-1977



Key

28 bits 28 bits

Shifted

28 bits

Shifted

28 bits

56 bits

Permuted Choice

48 bits

R_i

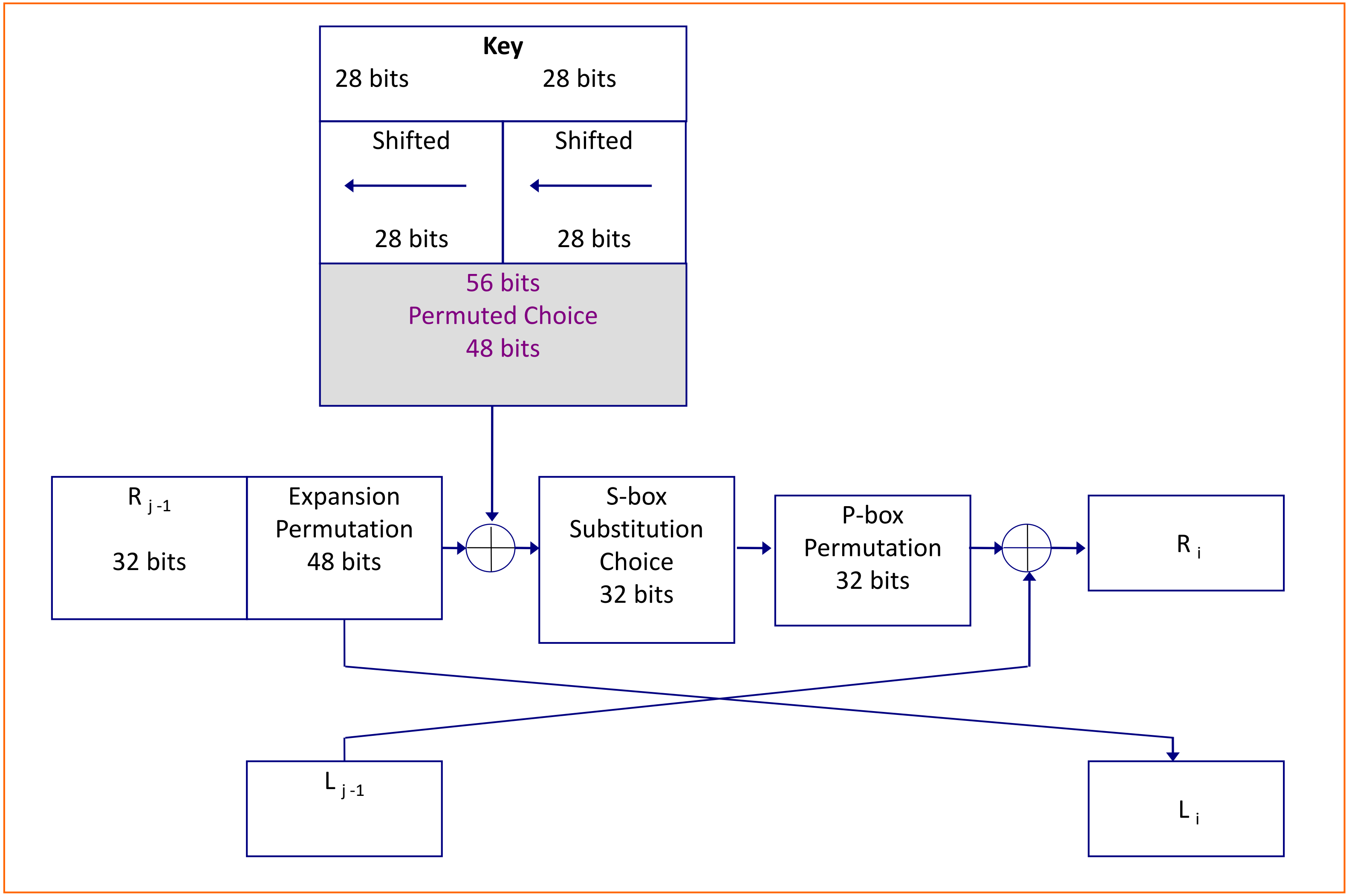
L_i

COMPRESSION PERMUTATION (56 -> 48)

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

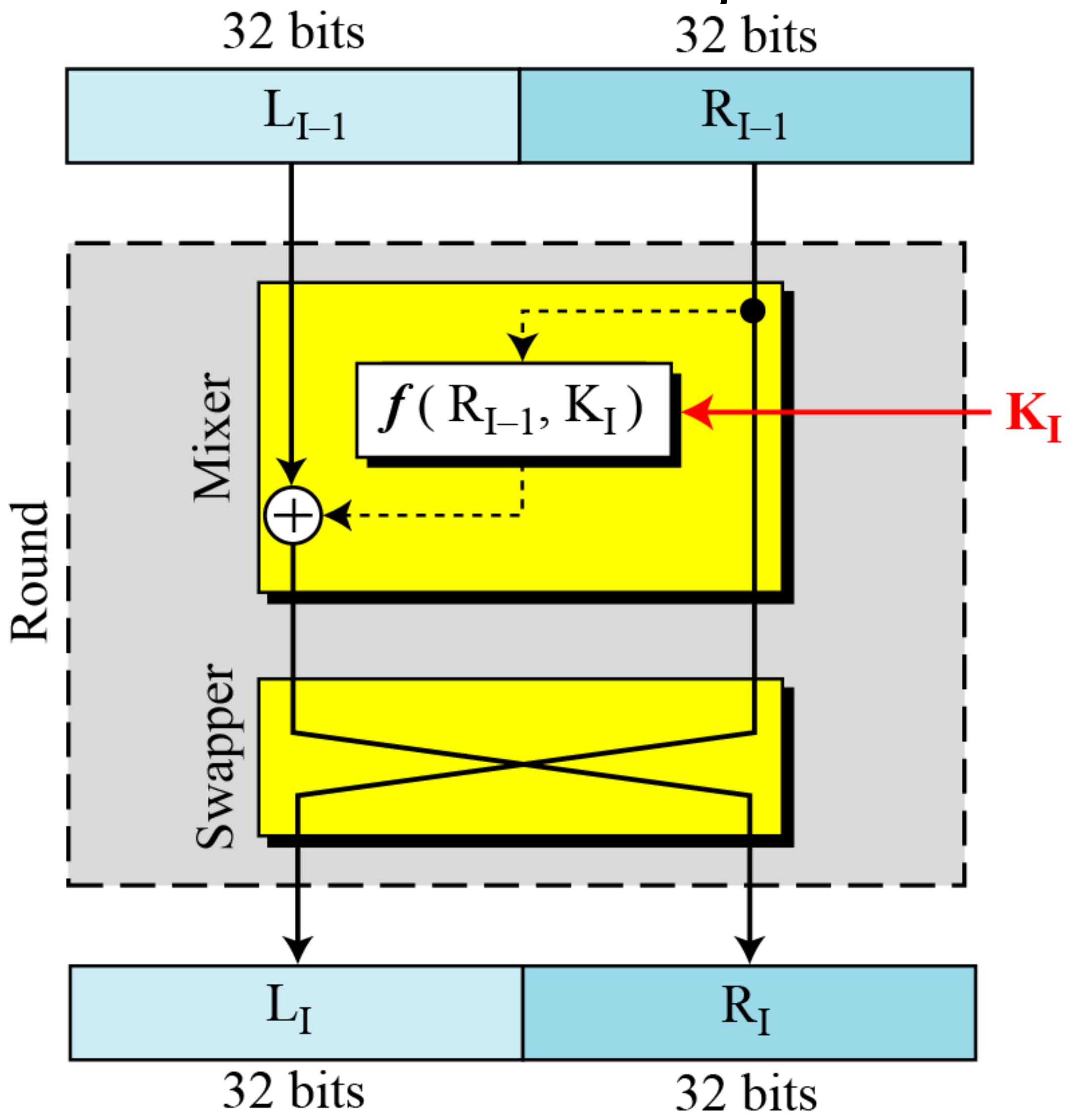
Symmetrical

Block:DES-1977



DES Rounds

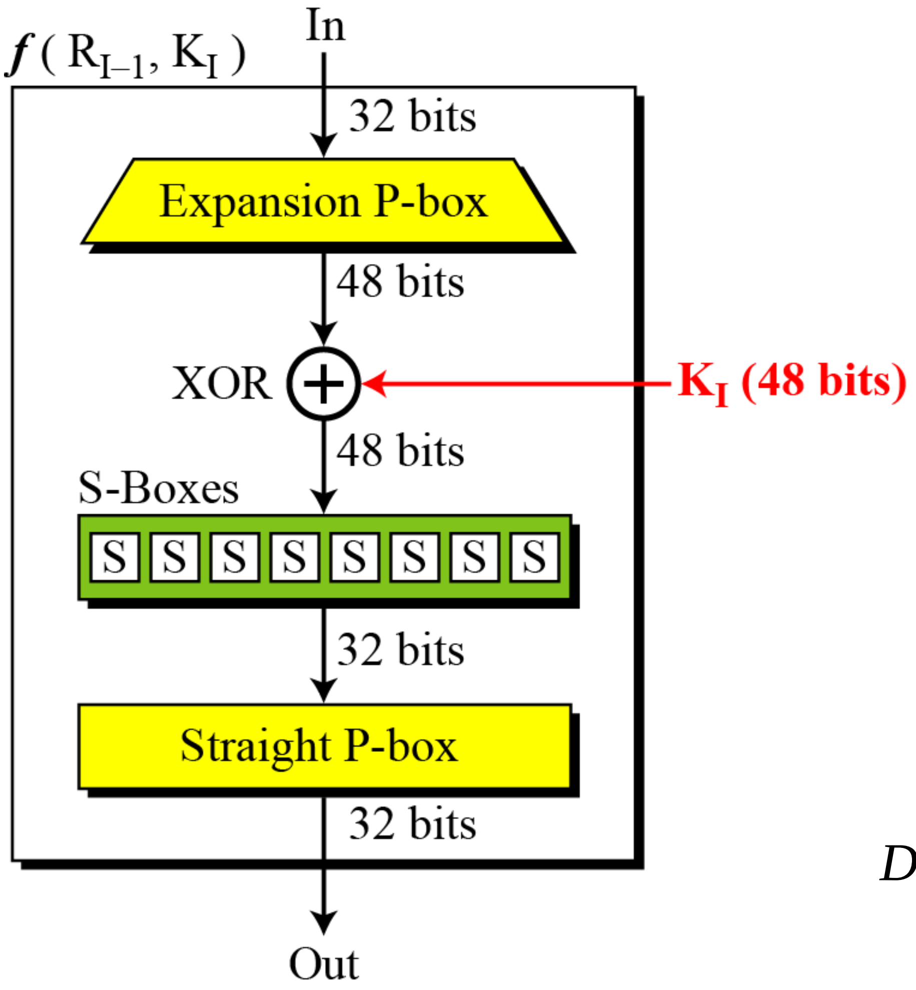
DES uses 16 rounds. Each round of DES is a Feistel cipher.



A round in DES (encryption site)

DES Function

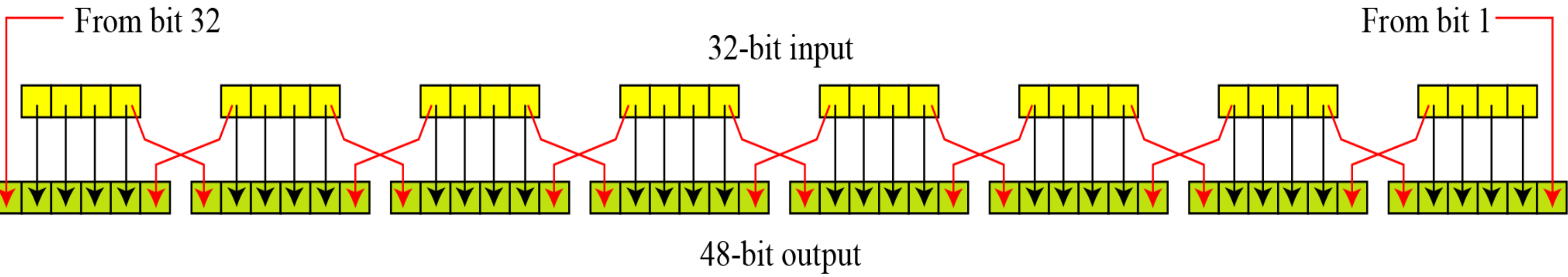
The heart of DES is the DES function. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



DES function

Expansion P-box

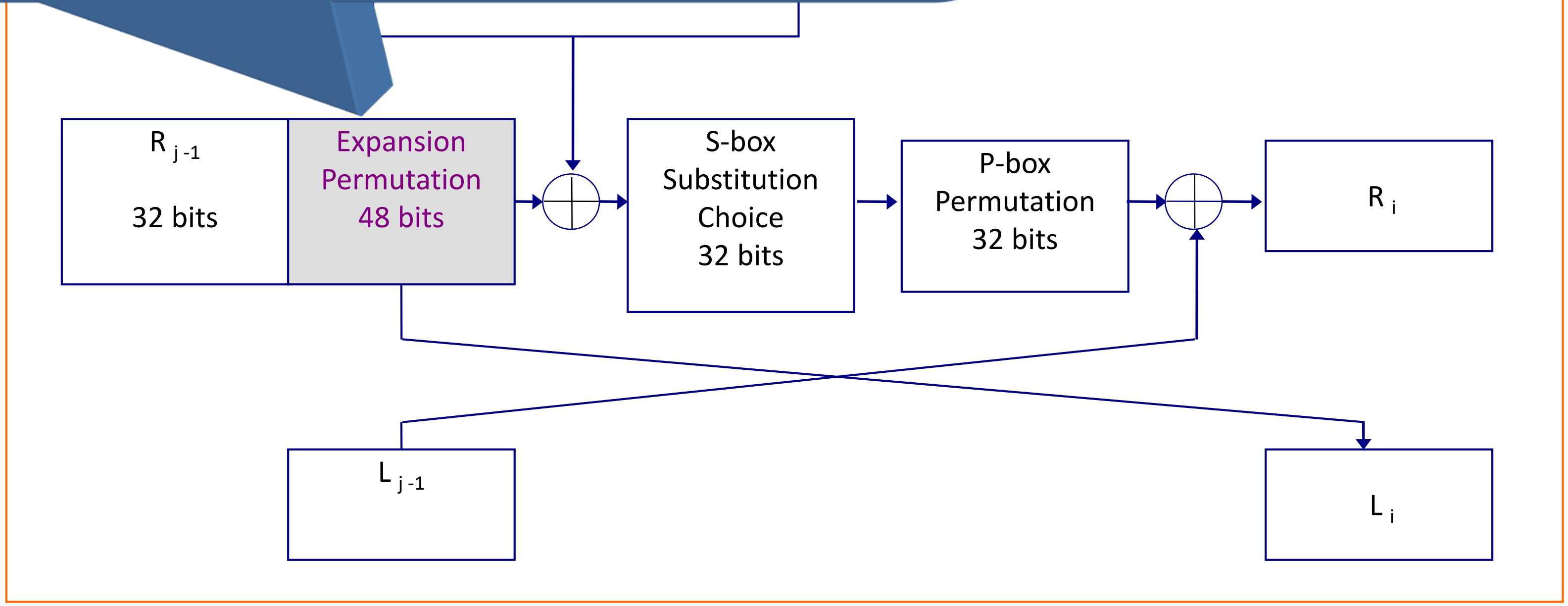
Since R_{I-1} is a 32-bit input and K_I is a 48-bit key, we first need to expand R_{I-1} to 48 bits.



Expansion permutation

EXPANSION PERMUTATION (32 -> 48)

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	
17	24	25	26	27	28	29	28	29	30	31	32	1



XOR

Whitener (XOR)

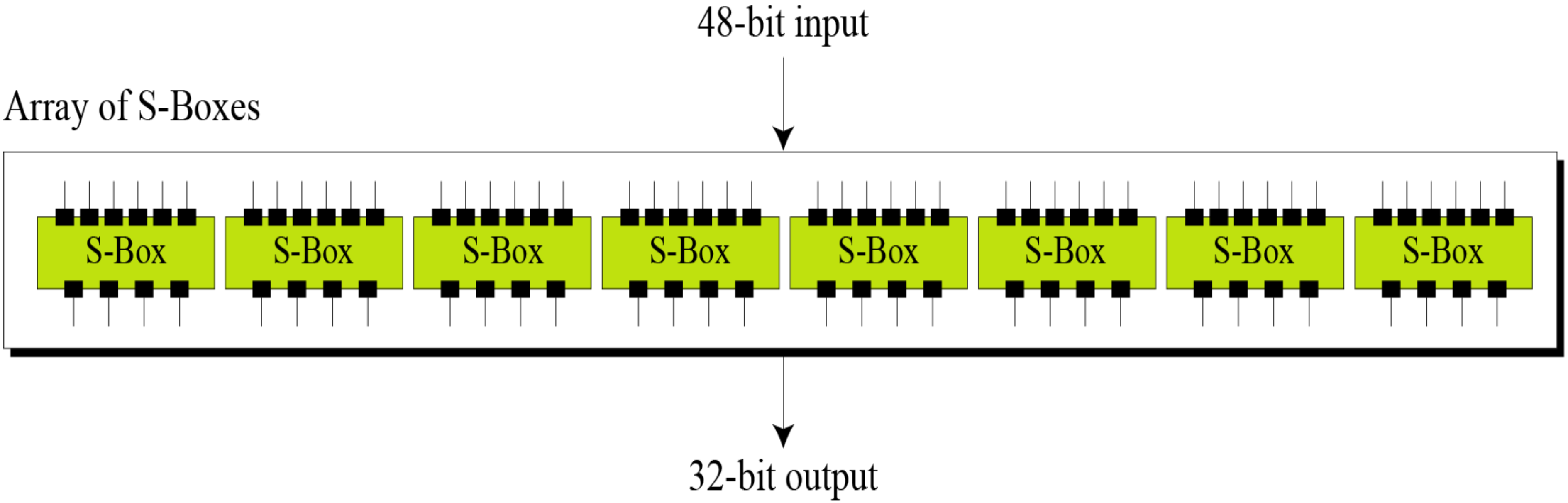
After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key.

Note that both **the right section and the key are 48-bits in length**. Also note that **the round key is used only in this operation**.

S-Boxes

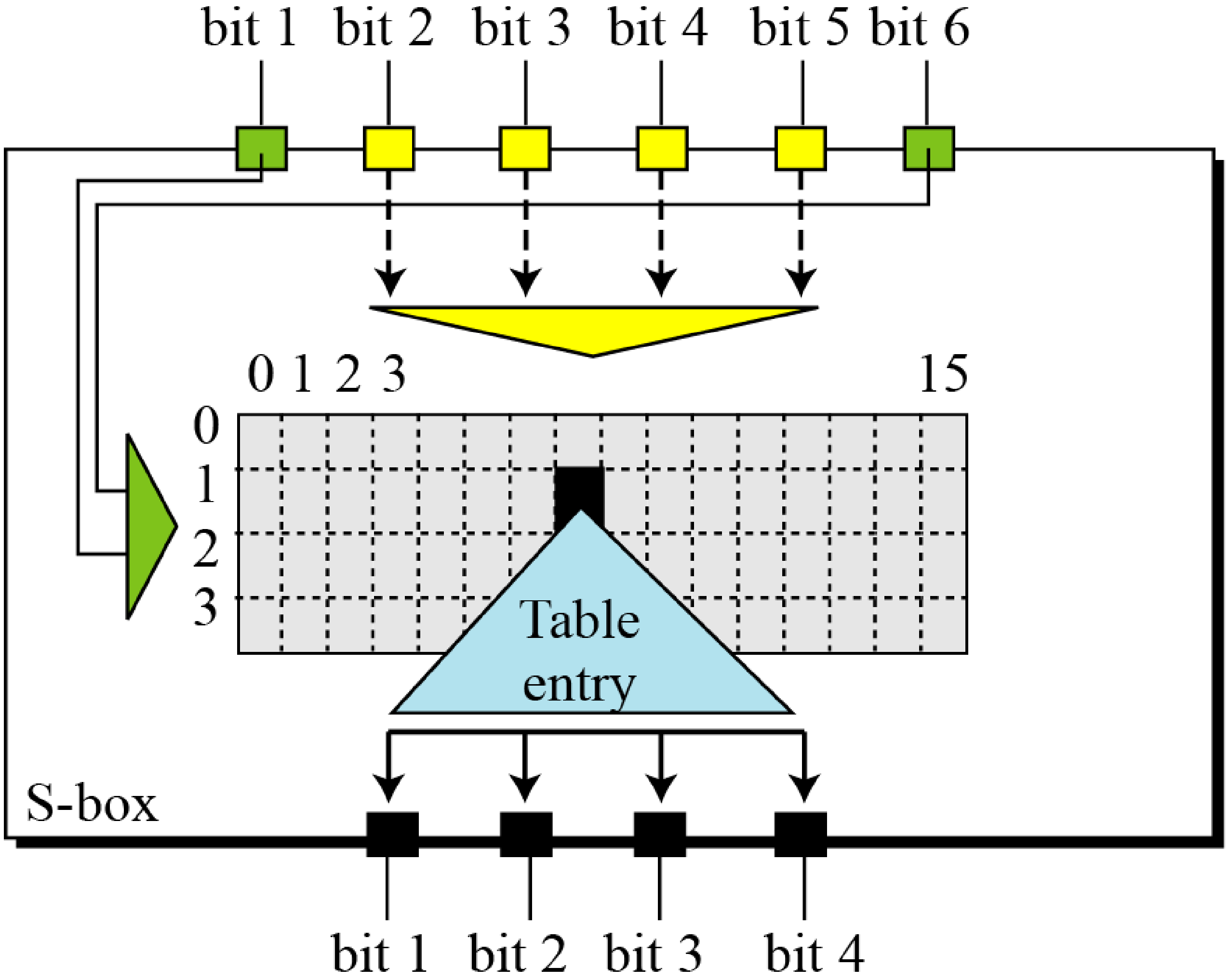
The S-boxes do the **real mixing (confusion).**

DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.



S-boxes

S-Box Rule


bit 1
bit 2
bit 3
bit 4
bit 5
bit 6
0
1
2
3
15
0
1
2
3
Table entry
S-box
bit 1
bit 2
bit 3
bit 4

*** s1 ***

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

*** s2 ***

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

*** s3 ***

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

*** s4 ***

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

*** s5 ***

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

*** s6 ***

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

*** s7 ***

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

*** s8 ***

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

$(15)_{10} =(1111)_2$

EXAMPLE S-BOX USAGE:

INPUT TO THE 5th. S-BOX IS ==> 110011

1 1   1

ST BITS

NS 3rd. ROW

R BITS

ORM 1001

$(9)_{10}$ WHICH MEANS 9th. COLUMN

3rd. ROW, 9th. COLUMN OF
THE 5th. S-BOX IS ==> 15, AND SO

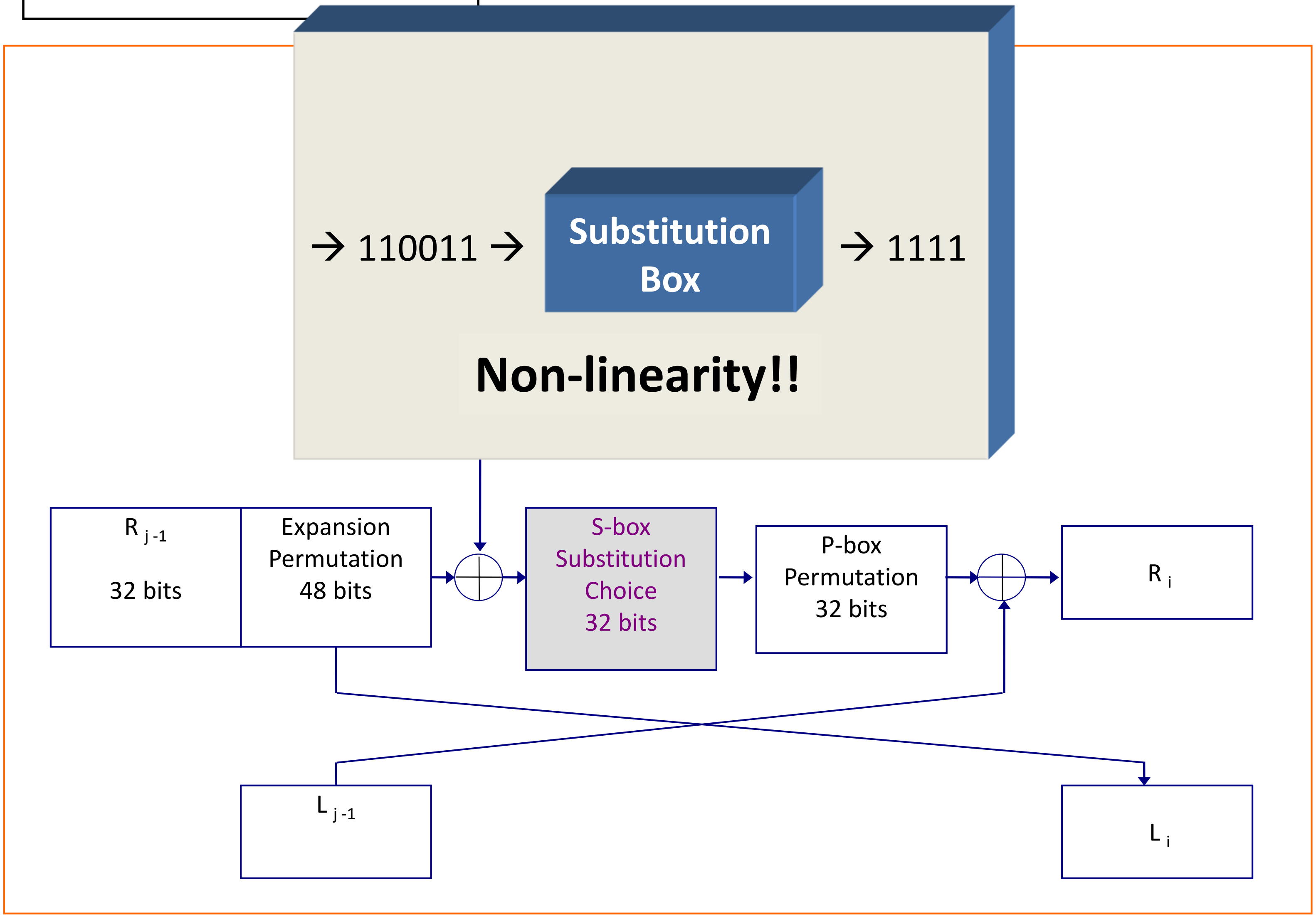
$\mathbf{(15)_{10} = (1111)_{2}}$

SO THE VALUE **1111**
IS SUBSTITUED FOR **110011**

Symmetrix

Block:DES-1977

Symmetrical



→ 110011 → Substitution Box → 1111

Non-linearity!!

R j -1

32 bits

Expansion Permutation 48 bits

S-box Substitution Choice 32 bits

P-box Permutation 32 bits

R i

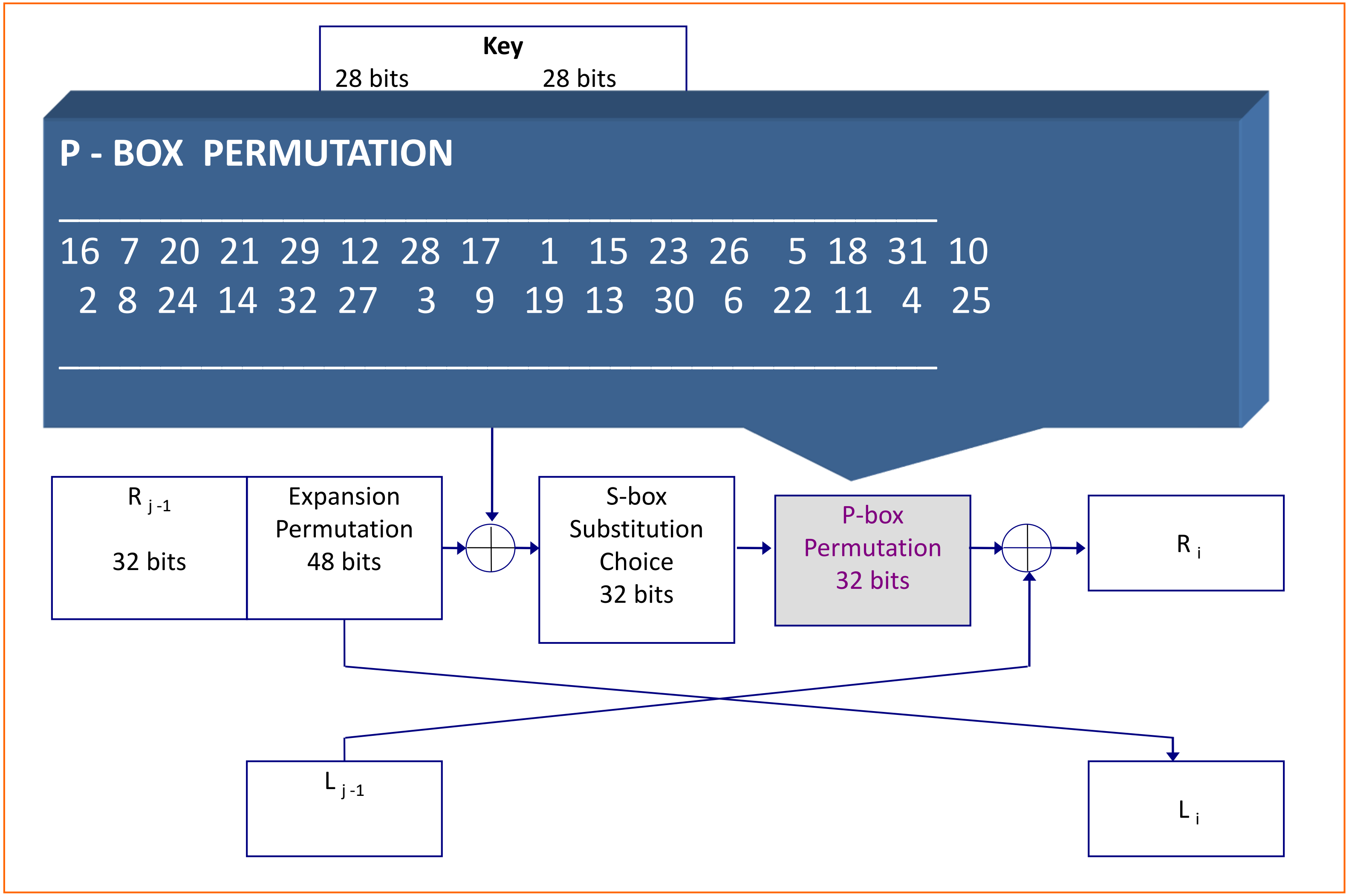
L j -1

L i

Symmetrix

Symmetrical

Block:DES-1977



P - BOX PERMUTATION

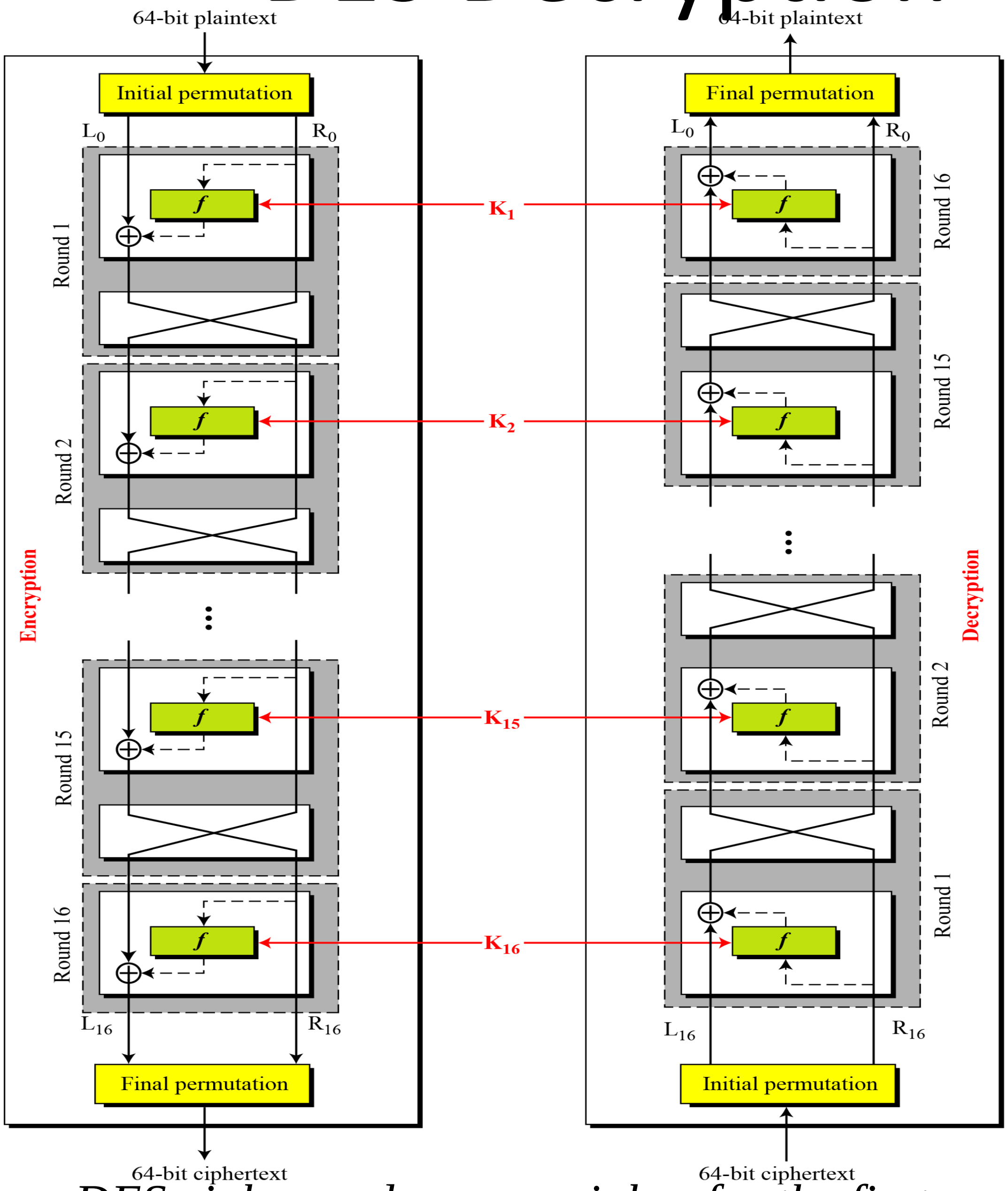
16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

Key
28 bits
28 bits
R_{j-1}
32 bits
Expansion Permutation 48 bits
S-box Substitution Choice 32 bits
P-box Permutation 32 bits
R_i
L_{j-1}
L_i

DES Decryption

- Decrypt must unwind steps of data computation
- With feistel design, do encryption steps again using subkeys in reverse order (SK16 … SK1)
  - IP undoes final FP step of encryption
  - 1st round with SK16 undoes 16th encrypt round
  - ….
  - 16th round with SK1 undoes 1st encrypt round
  - Then final FP undoes initial encryption IP
  - Thus recovering original data value

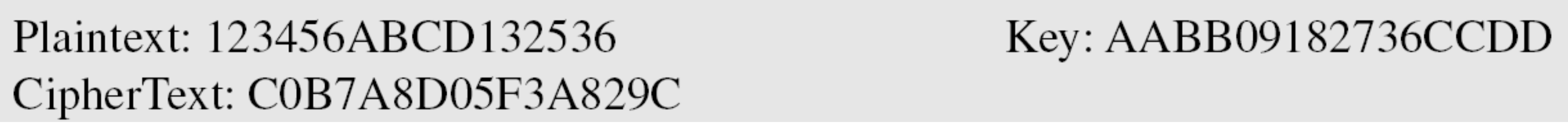
DES Decryption



DES cipher and reverse cipher for the first approach

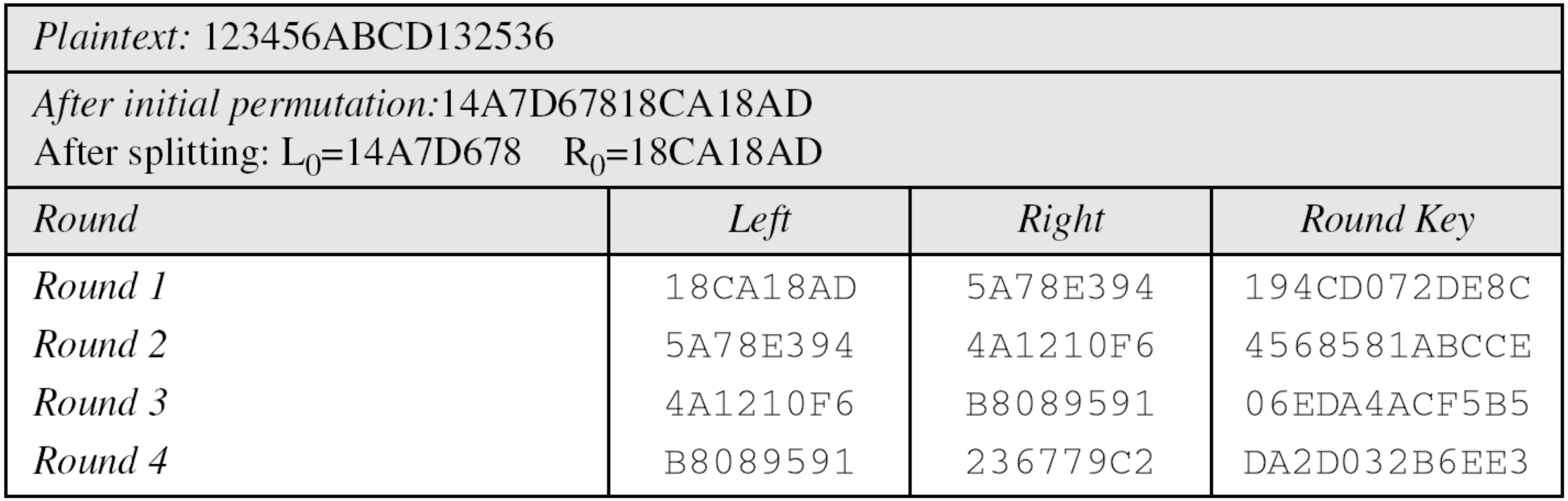
Example 2

We choose a random plaintext block and a random key, and determine what the ciphertext block would be (all in hexadecimal):



Plaintext: 123456ABCD132536 Key: AABB09182736CCDD
CipherText: C0B7A8D05F3A829C

Table *Trace of data for Example 2*



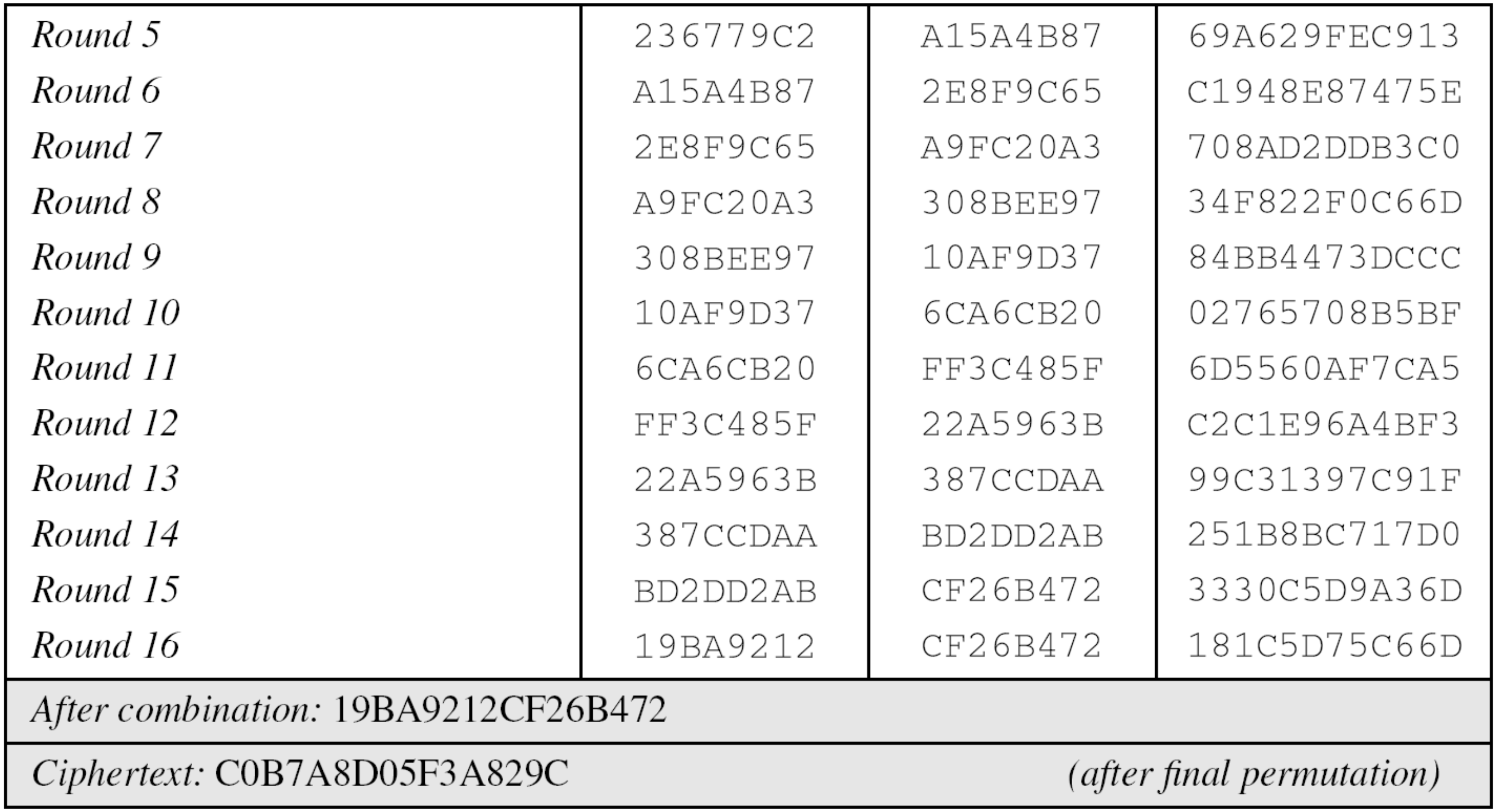
Plaintext: 123456ABCD132536

*After initial permutation:*14A7D67818CA18AD
After splitting: L_0=14A7D678 R_0=18CA18AD

Round	*Left*	*Right*	*Round Key*
Round 1	18CA18AD	5A78E394	194CD072DE8C
Round 2	5A78E394	4A1210F6	4568581ABCCE
Round 3	4A1210F6	B8089591	06EDA4ACF5B5
Round 4	B8089591	236779C2	DA2D032B6EE3

Example 2

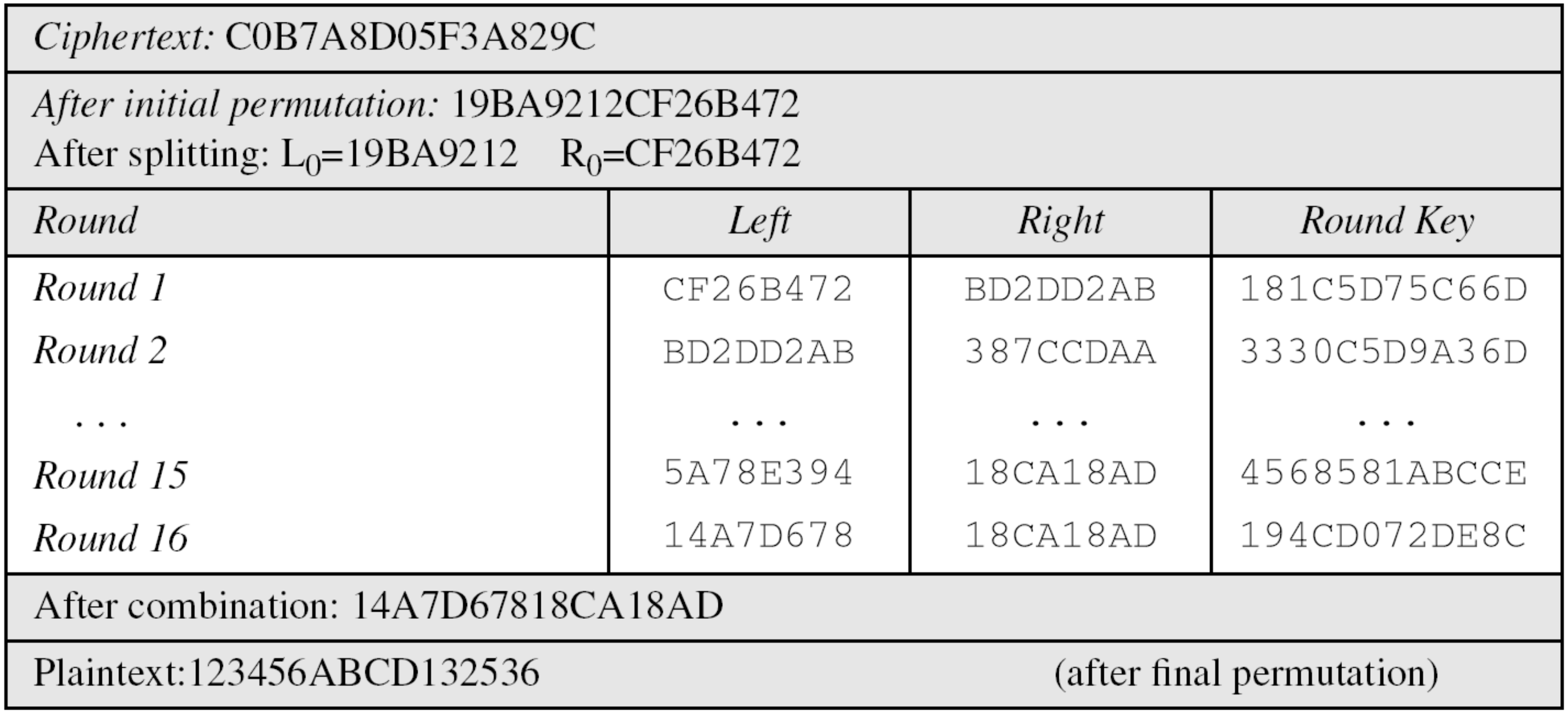
Table *Trace of data for Example 2(Continue)*



Round 5	236779C2	A15A4B87	69A629FEC913
Round 6	A15A4B87	2E8F9C65	C1948E87475E
Round 7	2E8F9C65	A9FC20A3	708AD2DDB3C0
Round 8	A9FC20A3	308BEE97	34F822F0C66D
Round 9	308BEE97	10AF9D37	84BB4473DCCC
Round 10	10AF9D37	6CA6CB20	02765708B5BF
Round 11	6CA6CB20	FF3C485F	6D5560AF7CA5
Round 12	FF3C485F	22A5963B	C2C1E96A4BF3
Round 13	22A5963B	387CCDAA	99C31397C91F
Round 14	387CCDAA	BD2DD2AB	251B8BC717D0
Round 15	BD2DD2AB	CF26B472	3330C5D9A36D
Round 16	19BA9212	CF26B472	181C5D75C66D
After combination: 19BA9212CF26B472			
Ciphertext: C0B7A8D05F3A829C			*(after final permutation)*

Example 3

Let us see how Bob, at the destination, can decipher the ciphertext received from Alice using the same key. Table shows some interesting points.



Ciphertext: C0B7A8D05F3A829C			
After initial permutation: 19BA9212CF26B472 After splitting: L_0=19BA9212 R_0=CF26B472			
Round	*Left*	*Right*	*Round Key*
Round 1	CF26B472	BD2DD2AB	181C5D75C66D
Round 2	BD2DD2AB	387CCDAA	3330C5D9A36D
. . .	. . .	. . .	. . .
Round 15	5A78E394	18CA18AD	4568581ABCCE
Round 16	14A7D678	18CA18AD	194CD072DE8C
After combination: 14A7D67818CA18AD			
Plaintext:123456ABCD132536			(after final permutation)

Animation Link: http://www.cs.bham.ac.uk/research/projects/lemsys/DES/DESPage.jsp

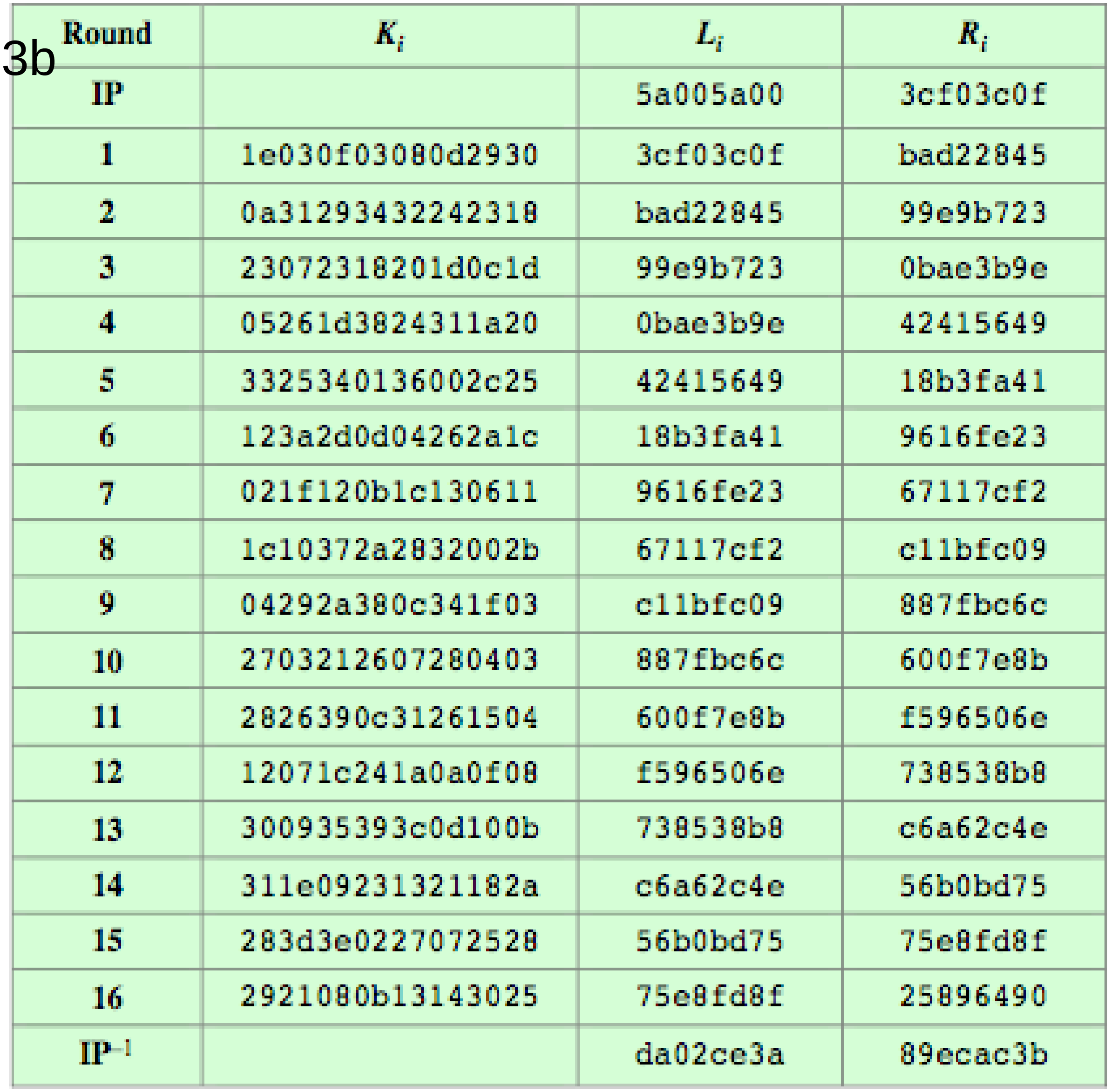
DES Example

Plaintext: 02468aceeca86420
Key: 0f1571c947d9e859
Ciphertext: da02ce3a89ecac3b

The first row shows the 32-bit values of the left and right halves of data after the initial permutation.
The next 16 rows show the results after each round.

Also shown is the value of the 48-bit subkey generated for each round.

The final row shows the left and right-hand values after the inverse initial permutation. These two values combined form the ciphertext.



Round	K_i	L_i	R_i
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP^{-1}		da02ce3a	89ecac3b

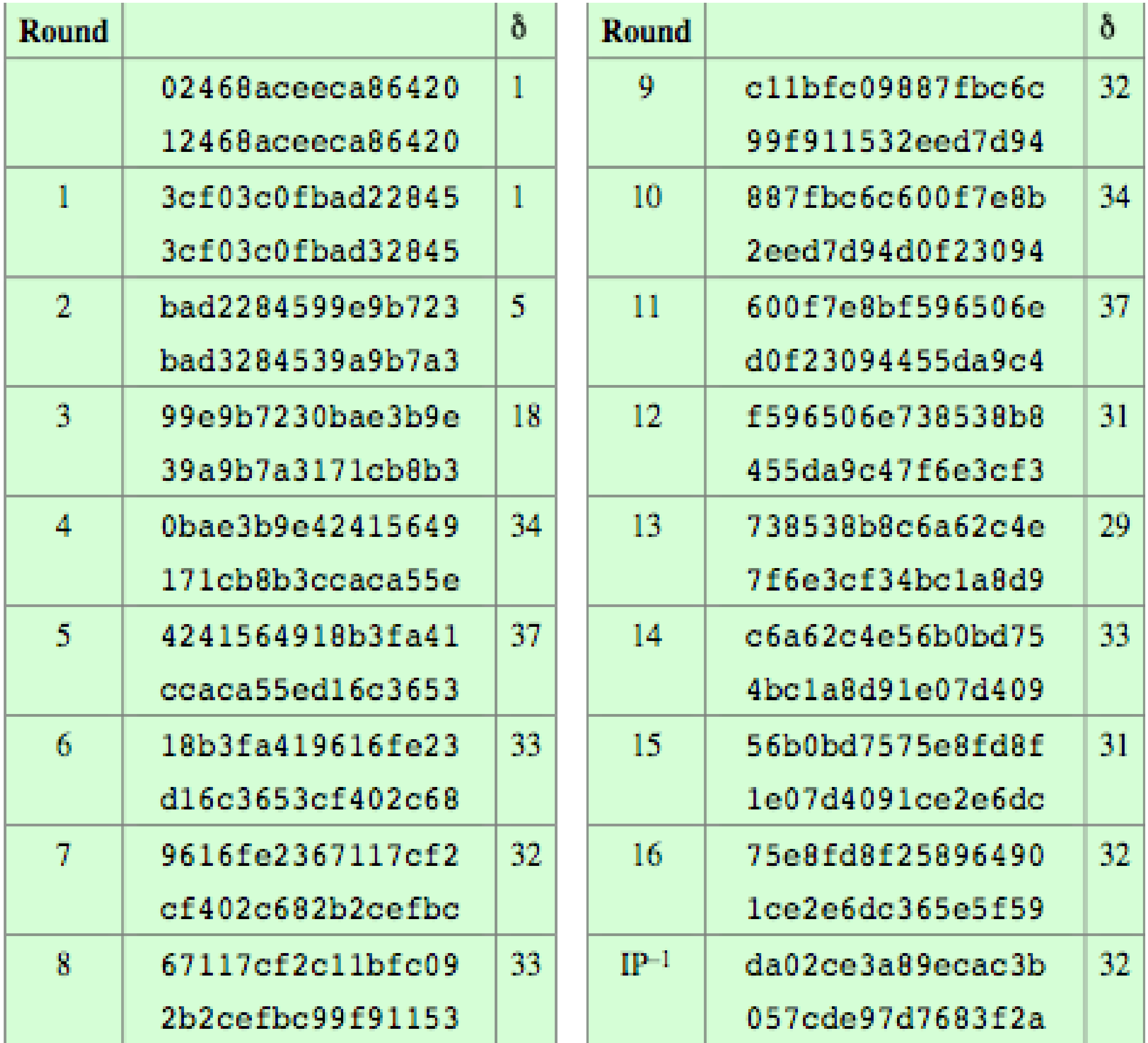
Avalanche Effect

A desirable property of any encryption algorithm is that **a small change in either the plaintext or the key should produce a significant change in the ciphertext.**

- Key desirable property of encryption algorithm
- Where a change of **one** input or key bit results in changing approx **half** output bits
- Making attempts to “home-in” by guessing keys impossible
- DES exhibits strong avalanche

Avalanche in DES

Table shows the result when the fourth bit of the plaintext is changed.
The plaintext is 12468aceeca86420.
The second column shows the intermediate 64-bit values at the end of each round for the two plaintexts.
The third column shows the number of bits that differ between the two intermediate values.
The table shows that after just three rounds, 18 bits differ between the two blocks.



Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

Completeness Effect

Completeness effect means that each bit of the ciphertext needs to depend on many bits on the plaintext.

An Exercise

Q. Explain why the cipher $e_k(m) = k \oplus m$ and $d_k(c) = k \oplus c$ defined by XOR of bit strings is not secure against a chosen plaintext attack.

Demonstrate your attack by finding the private key used to encrypt the 16-bit ciphertext c = 1001010001010111 if you know that the corresponding plaintext is p=0010010000101100.

A.

c	1	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1
k	**1**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**
p	0	0	1	0	0	1	0	0	0	0	1	0	1	1	0	0

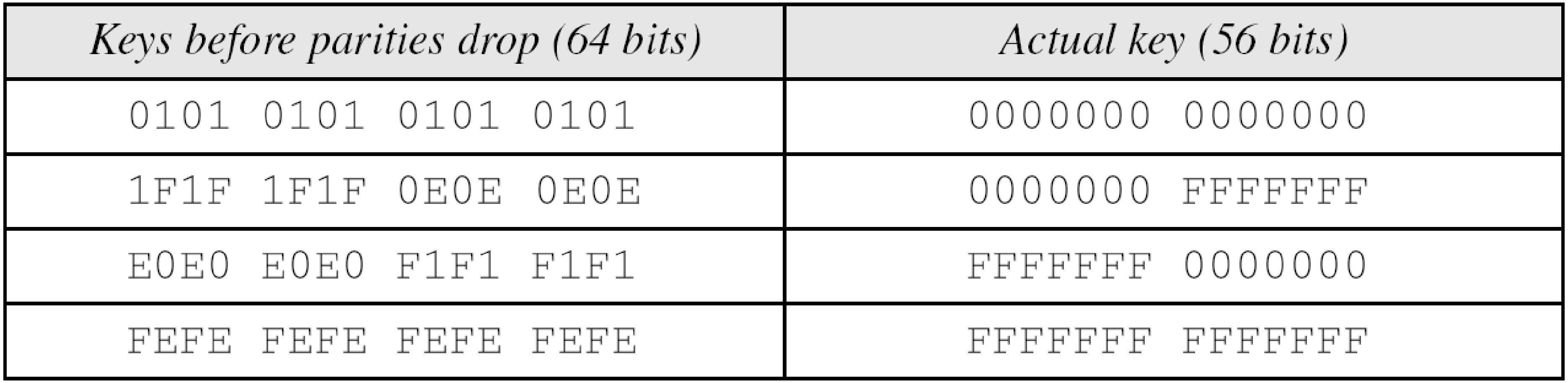
DES Weaknesses

During the last few years critics have found some weaknesses in DES.

Weaknesses in Cipher Design

1. Weaknesses in S-boxes
2. Weaknesses in P-boxes
3. Weaknesses in Key

Table 6.18 *Weak keys*



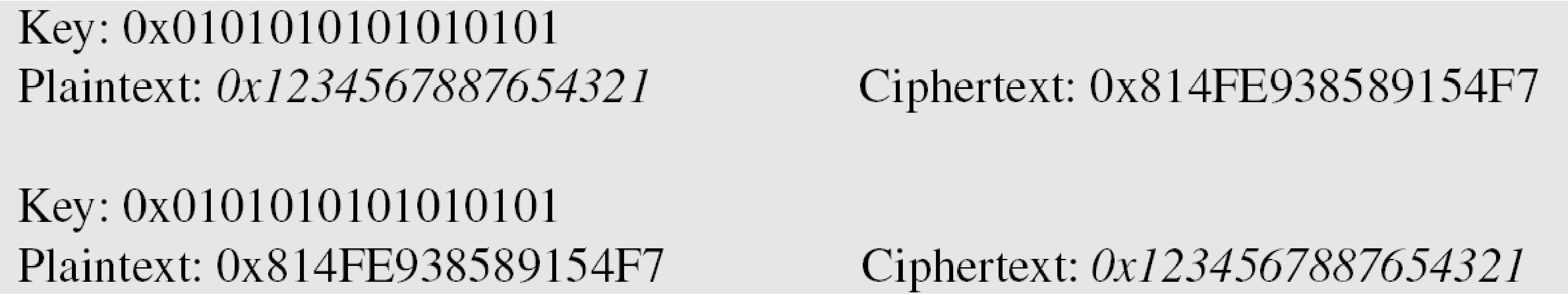
Keys before parities drop (64 bits)	*Actual key (56 bits)*
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

Strength of DES – Key Size

- 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
- brute force search looks hard
- recent advances have shown is possible
  - in 1997 on Internet in a few months
  - in 1998 on dedicated h/w (Electronic Frontier Foundation –EFF, "DES cracker" machine, $250,000) in a few days
  - in 1999 above combined in 22hrs!
- still must be able to recognize plaintext
- must now consider alternatives to DES

Example – DES Weaknesses

Let us try the first weak key in the Table to encrypt a block two times. After two encryptions with the same key the original plaintext block is created. Note that we have used the encryption algorithm two times, not one encryption followed by another decryption.

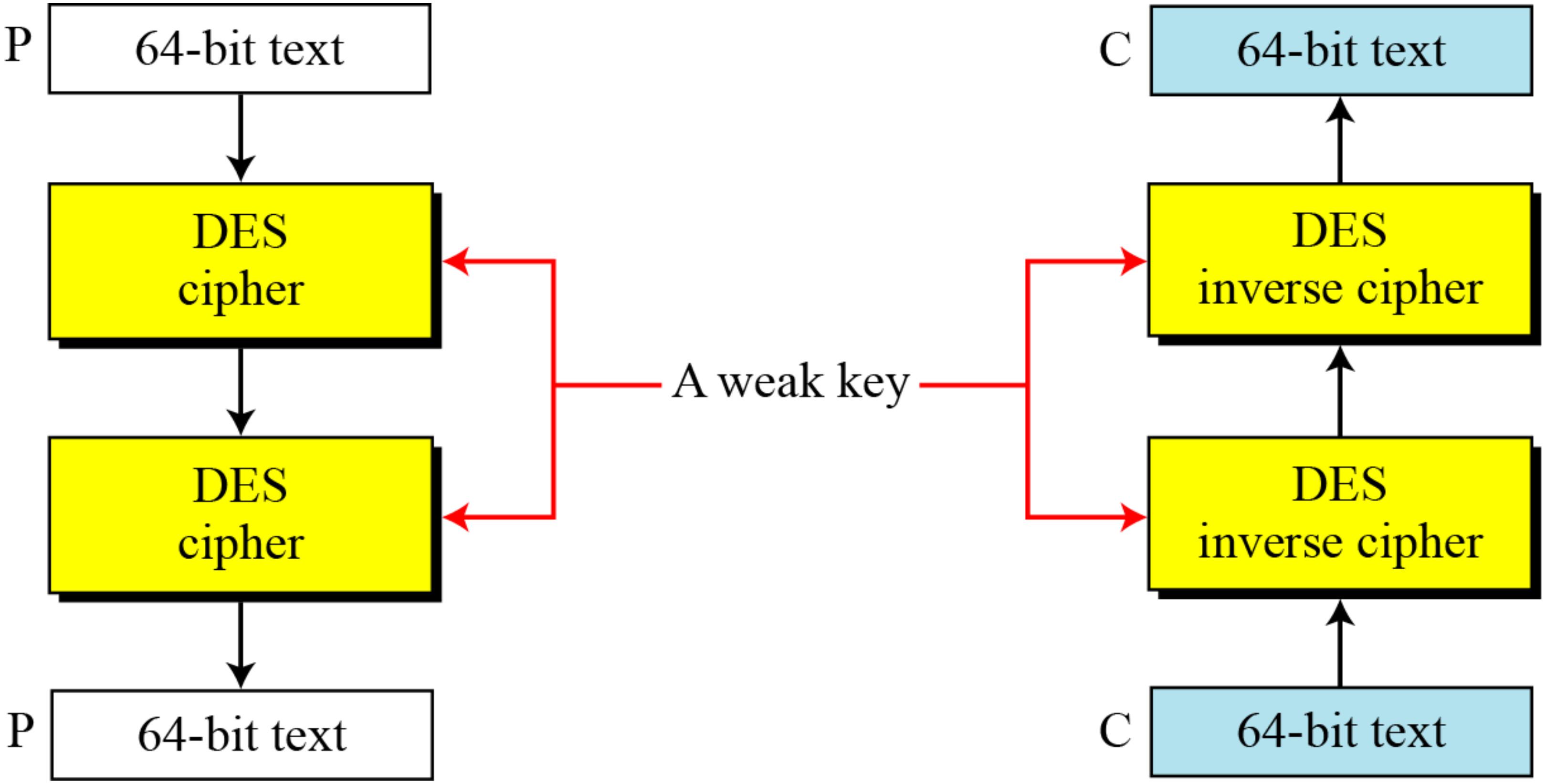


Key: 0x0101010101010101
Plaintext: *0x1234567887654321*   Ciphertext: 0x814FE938589154F7

Key: 0x0101010101010101
Plaintext: 0x814FE938589154F7   Ciphertext: *0x1234567887654321*

Example – DES Weaknesses

Double encryption and decryption with a weak key



Example – DES Weaknesses

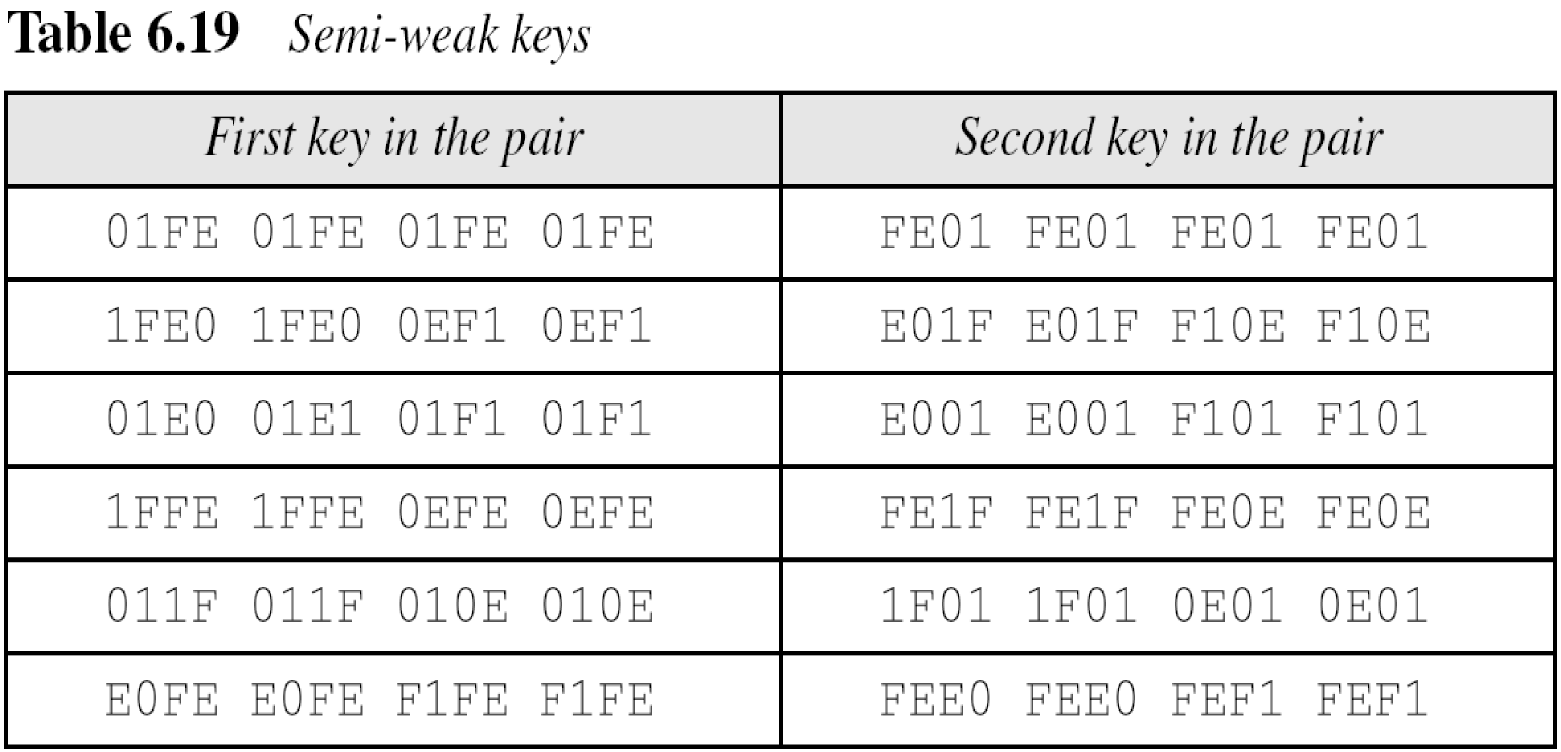
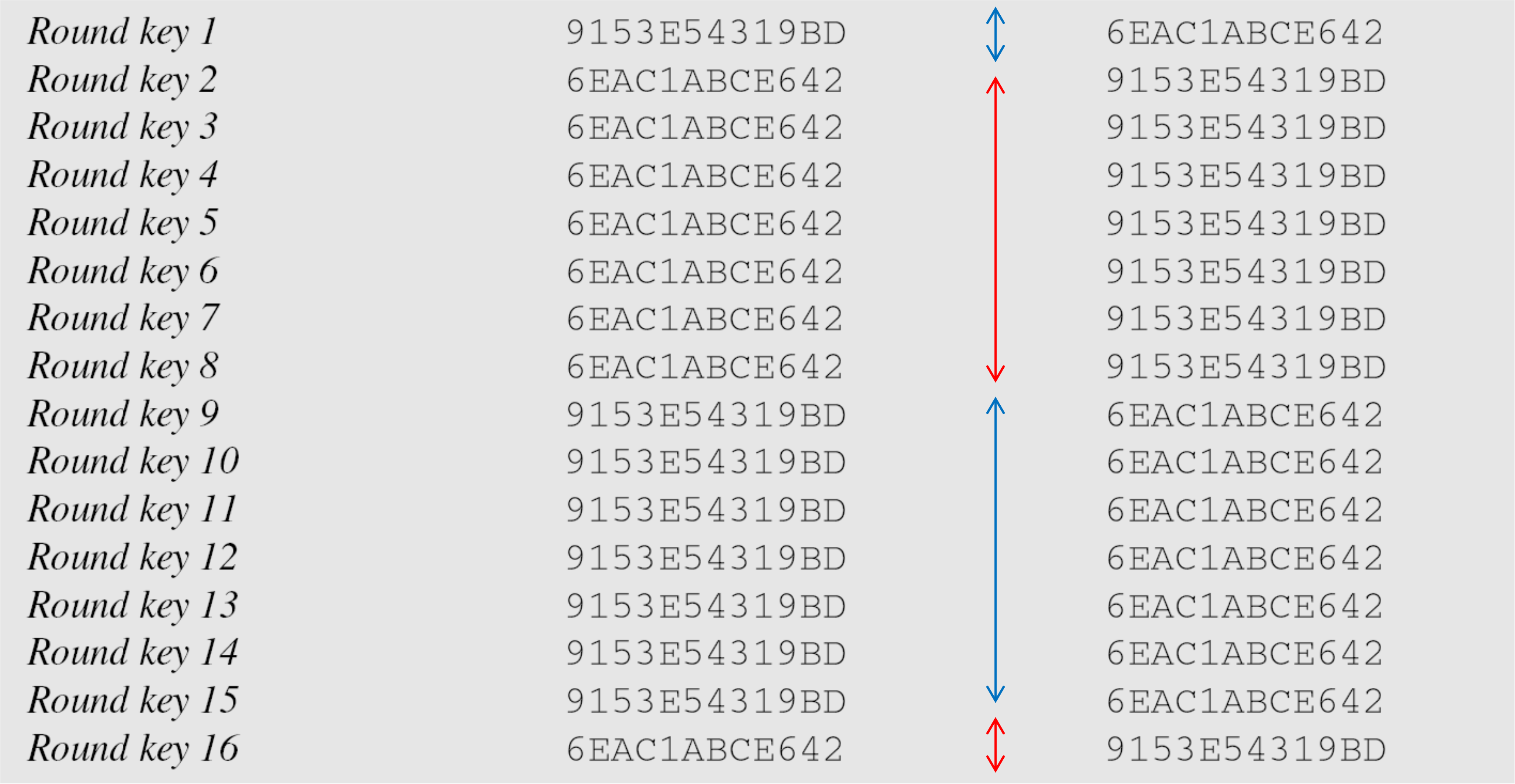


Table 6.19 *Semi-weak keys*

First key in the pair	*Second key in the pair*
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E
01E0 01E1 01F1 01F1	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1

Example – DES Weaknesses



Round key 1	9153E54319BD	6EAC1ABCE642
Round key 2	6EAC1ABCE642	9153E54319BD
Round key 3	6EAC1ABCE642	9153E54319BD
Round key 4	6EAC1ABCE642	9153E54319BD
Round key 5	6EAC1ABCE642	9153E54319BD
Round key 6	6EAC1ABCE642	9153E54319BD
Round key 7	6EAC1ABCE642	9153E54319BD
Round key 8	6EAC1ABCE642	9153E54319BD
Round key 9	9153E54319BD	6EAC1ABCE642
Round key 10	9153E54319BD	6EAC1ABCE642
Round key 11	9153E54319BD	6EAC1ABCE642
Round key 12	9153E54319BD	6EAC1ABCE642
Round key 13	9153E54319BD	6EAC1ABCE642
Round key 14	9153E54319BD	6EAC1ABCE642
Round key 15	9153E54319BD	6EAC1ABCE642
Round key 16	6EAC1ABCE642	9153E54319BD

Security of DES

- Now have several analytic attacks on DES
- These utilise some deep structure of the cipher
  - by gathering information about encryptions
  - can eventually recover some/all of the sub-key bits
  - if necessary then exhaustively search for the rest
- Generally these are statistical attacks
  - differential cryptanalysis
  - linear cryptanalysis
  - related key attacks

Brute-Force Attack

We have discussed the weakness of short cipher key in DES.

Combining this weakness with the key complement weakness, it is clear that DES can be broken using 2^{55} encryptions.

Multiple Encryption with DES

- In 2001, NIST published the Advanced Encryption Standard (AES) to replace DES.
- But users in commerce and finance are not ready to give up on DES.
- As a temporary solution to DES's security problem, one may encrypt a message (with DES) multiple times using multiple keys:
  - 2DES is not much securer than the regular DES
  - So, 3DES with either 2 or 3 keys is used

3DES with 2 keys

- A straightforward implementation would be :

$$c := E_{k_1}\left(E_{k_2}\left(E_{k_1}(m)\right)\right)$$

- In practice : $c := E_{k_1}\left(D_{k_2}\left(E_{k_1}(m)\right)\right)$
  - Also referred to as EDE encryption
- Reason : if $k_1 = k_2$, then 3DES = 1DES.

  Thus, a 3DES software can be used as a single-DES.
- Standardized in ANSI X9.17 & ISO 8732.
- No practical attacks are known.

3DES with 3 keys

- Encryption: $c := E_{k_3}\left(D_{k_2}\left(E_{k_1}(m)\right)\right).$
- If $k_1 = k_3,$ it becomes 3DES with 2 keys.
- If $k_1 = k_2 = k_3,$ it becomes the regular DES.
- So, it is backward compatible with both 3DES with 2 keys and the regular DES.
- Some internet applications adopt 3DES with three keys; e.g. PGP and S / MIME.

Summary

- Have considered:
  - Block vs stream ciphers
  - Feistel cipher design & structure
  - DES
    - details
    - strength
  - DES attack types
  - DES implementation types