CISC 468: CRYPTOGRAPHY

LESSON 5: BLOCK CIPHERS

Furkan Alaca

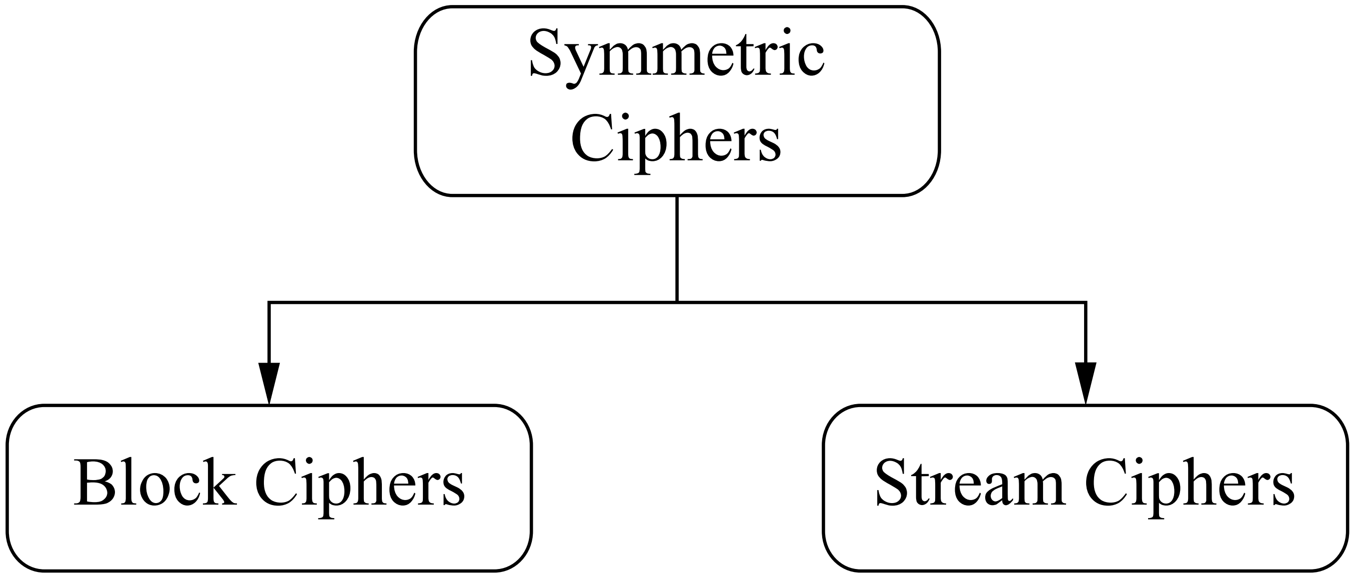
TODAY, WE WILL LEARN ABOUT...

1. How block ciphers and stream ciphers differ
2. Bit operations required

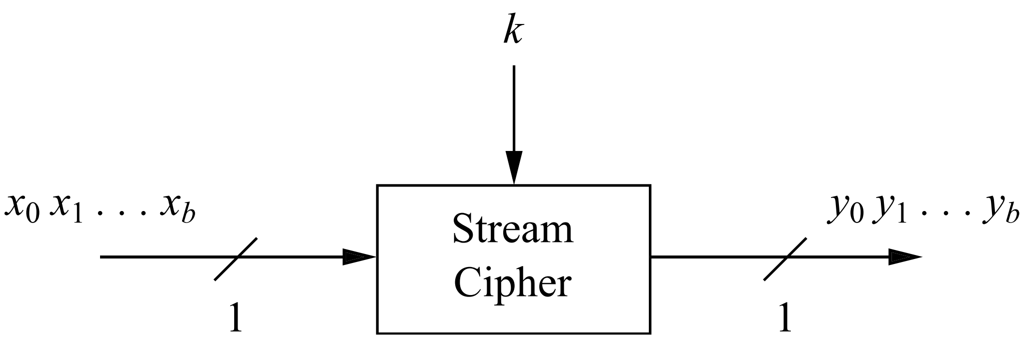
READINGS

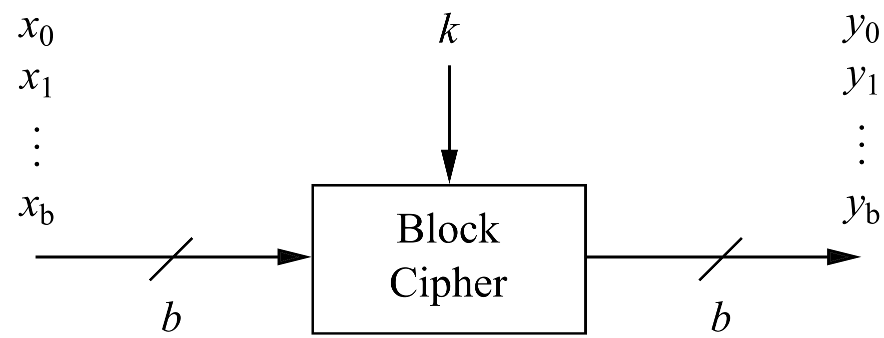
- Section 3.1: Introduction to DES, Paar & Pelzl
- Section 3.2: Overview of the DES Algorithm, Paar & Pelzl
- Section 3.3: Internal Structure of DES, Paar & Pelzl

SYMMETRIC CIPHERS



STREAM CIPHERS VS. BLOCK CIPHERS





BLOCK CIPHERS

- A block cipher is a *bijective function* from $\mathcal{P}$ (set of all possible plaintext blocks) to $\mathcal{C}$ (set of all possible ciphertext blocks)
- Its two inputs are a key and a fixed-size block of plaintext
- Its output is a fixed-size block of ciphertext
- The plaintext and ciphertext block size are equal (e.g., 128 bits), but key size and block size need not be equal
- In a later chapter we will deal with plaintext messages that exceed the block size

BLOCK CIPHER DESIGN PROPERTIES (1)

1. *Confusion:* Each bit of ciphertext depends on several parts of the key, thereby obscuring the relationship between the key and ciphertext. Commonly achieved via *substitution* operations.

BLOCK CIPHER DESIGN PROPERTIES (2)

2. *Diffusion:* Changing a single bit of plaintext should impact half of the ciphertext bits on average, thereby obscuring the relationship between plaintext and ciphertext. Commonly achieved via *bit permutation* operations.
   - Single flipped bit in the plaintext block x_1 results in a ciphertext block y_2 that appears statistically independent from the original ciphertext block y_1, e.g.,

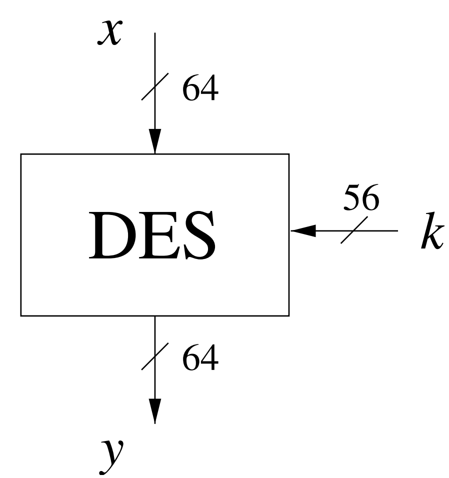
$$\begin{array}{l} x_1 = 0010\ 1011 \\ x_2 = 0000\ 1011 \end{array} \longrightarrow \boxed{\text{Block Cipher}} \longrightarrow \begin{array}{l} y_1 = 1011\ 1001 \\ y_2 = 0110\ 1100 \end{array}$$

THE DATA ENCRYPTION STANDARD (DES): HISTORY

- Published as Federal Information Processing Standard (FIPS) PUB 46 in 1977
- FIPS PUB 46 was revised in 1999 to recommend Triple DES
- Superceded in 2001 by the Advanced Encryption Standard (AES) published in FIPS PUB 186

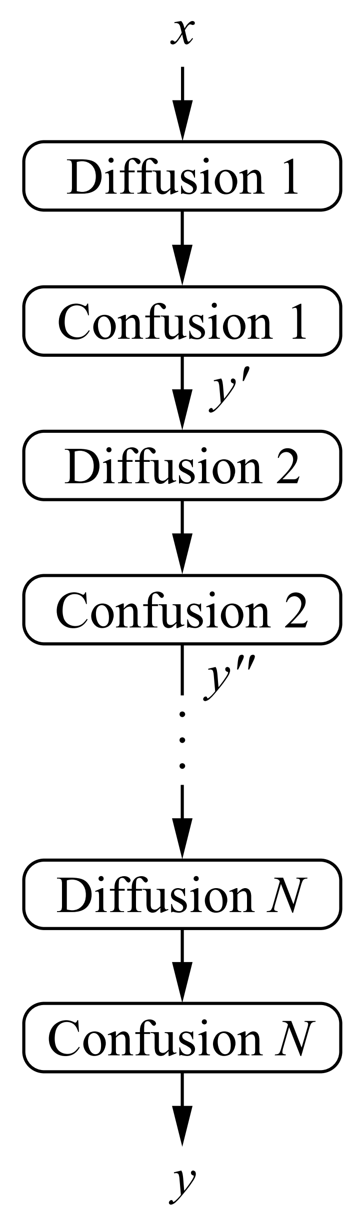
DES: BLOCK AND KEY SIZE, RELATED STRUCTURES

- DES is a block cipher with a block size of 64 bits and an effective key size of 56 bits
  - 64-bit key with every eighth bit used solely for parity-checking
- Design is related to two general structures: *product ciphers* and *Feistel ciphers*
  - Both structures involve repetition of a series of operations



PRODUCT CIPHERS

- A *product cipher* concatenates two or more operations such that the concatenation of operations is more secure than the individual operations
- In the following example, a confusion and diffusion operation are performed sequentially N times



FEISTEL CIPHERS

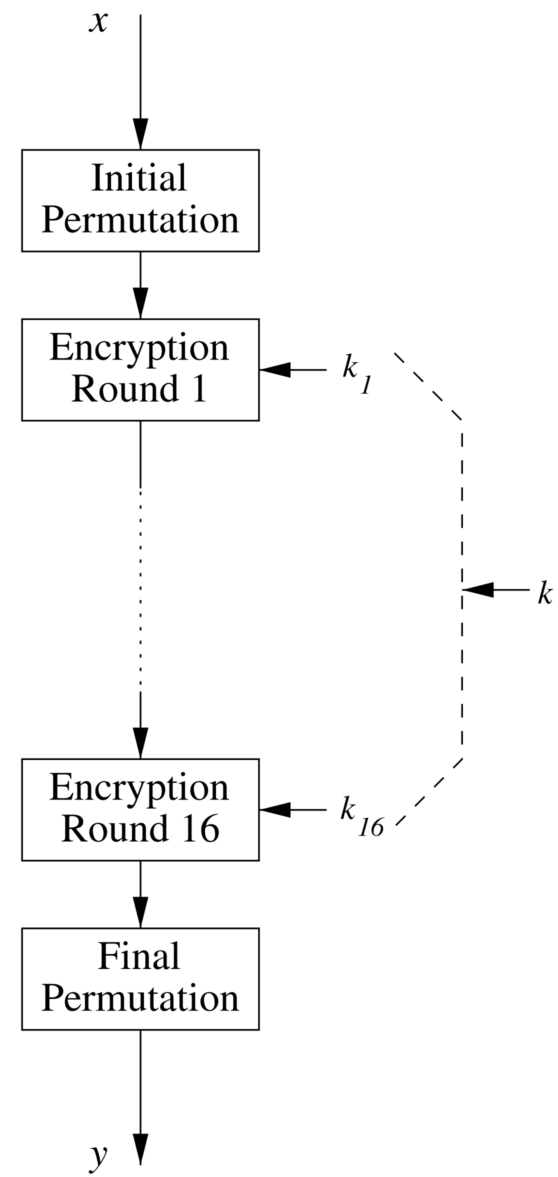
- A Feistel cipher is an iterative structure consisting of r repetitions of a *round function*
- Each round function performs an encryption operation using a subkey derived from the secret key using a *key schedule*
- Encryption and decryption are almost the same operation, with decryption requiring only a reversed key schedule
  - So the last round is reverted simply by repeating it

STRUCTURE OF A FEISTEL CIPHER

- Input is an even-length block of plaintext, which divided into left and right equal-sized halves (L_0, R_0)
- Output is a block of ciphertext (L_r, R_r) produced by an r-round process
- Round i maps $(L_{i-1}, R_{i-1}) \xrightarrow{k_i} (L_i, R_i)$, where
  - k_i is a subkey derived from the secret key k
  - $L_i = R_{i-1}$ (copied from right half of previous round)
  - $R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$
    - f function provides the confusion and diffusion

DES: ITERATIVE STRUCTURE

- DES is a 16-round Feistel cipher
- Before inputing into the encryption rounds, the input plaintext x undergoes an Initial Permutation (IP)
- After the final encryption round, the output undergoes a Final Transformation (IP^{-1})
- Each encryption round uses a 48-bit round key $k_1, k_2, \ldots, k_{16}$ derived from the 56-bit key k



DES: INITIAL PERMUTATION

- IP and IP^{-1} do not add any security
  - Thought to make data fetches easier in 1970s hardware
  - Very efficient in hardware, but not in software
- Represented as table, but read as 1D-array from left to right and top to bottom

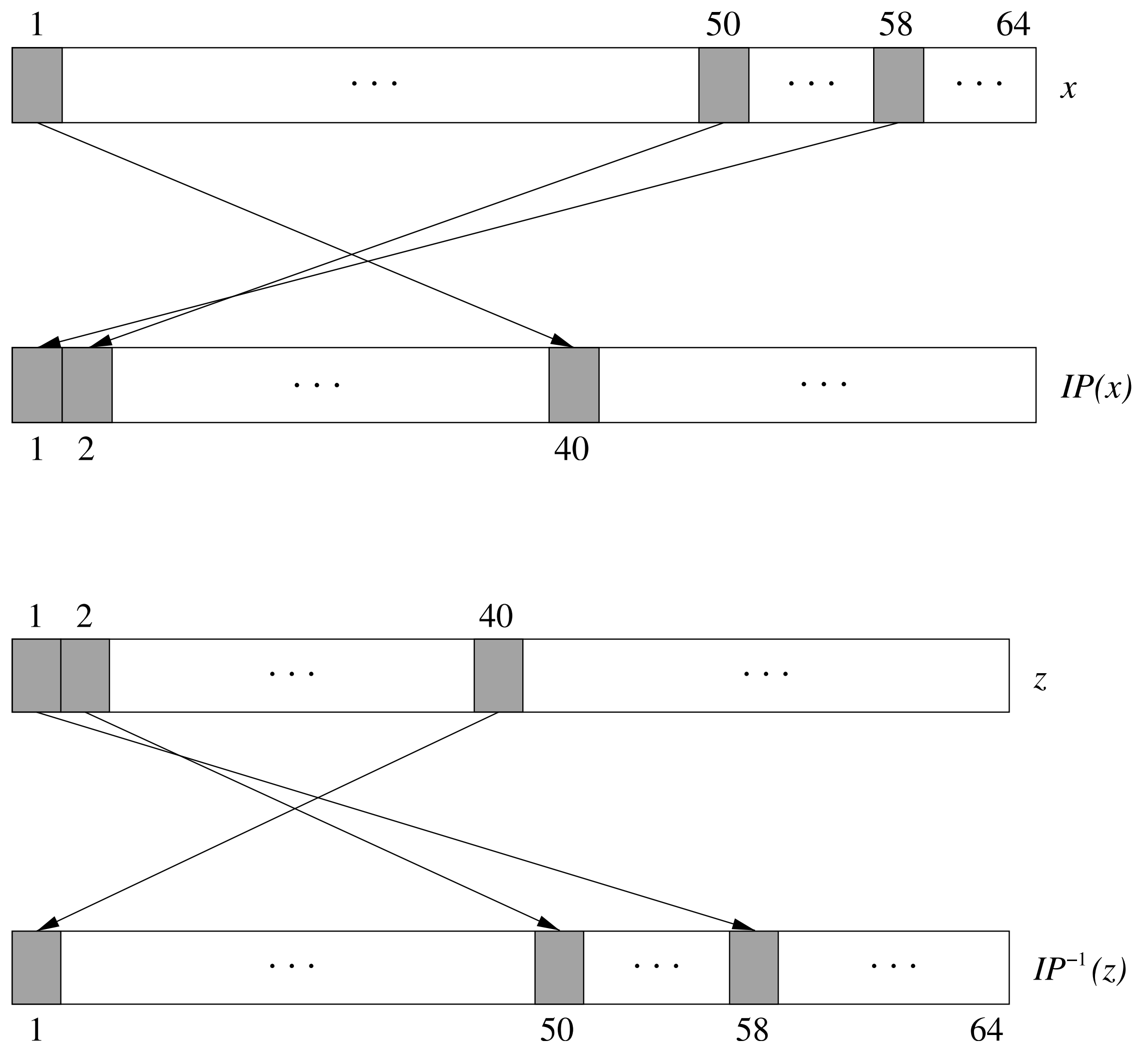
IP

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

IP^{-1}

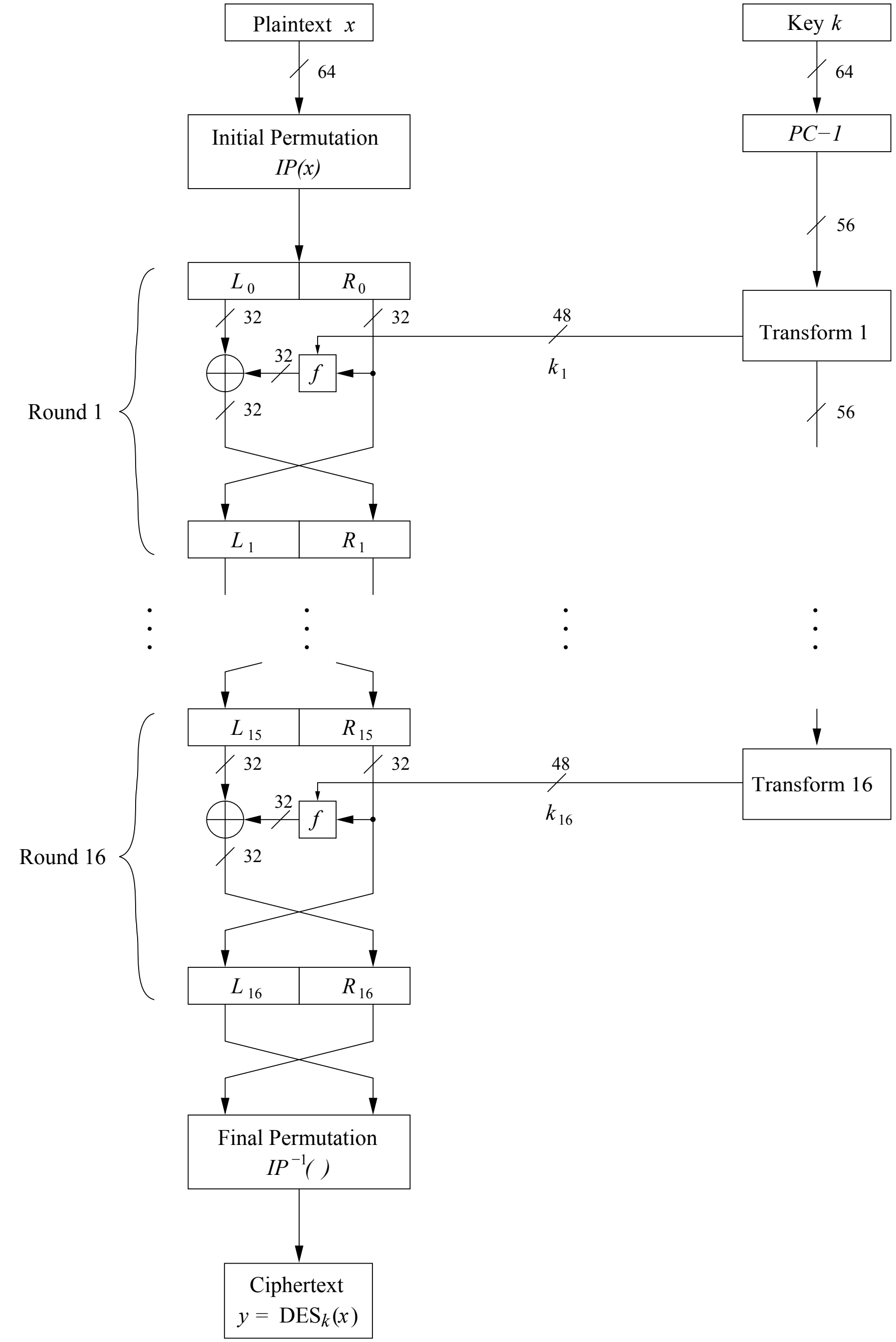
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

INITIAL AND FINAL PERMUTATIONS: ILLUSTRATION



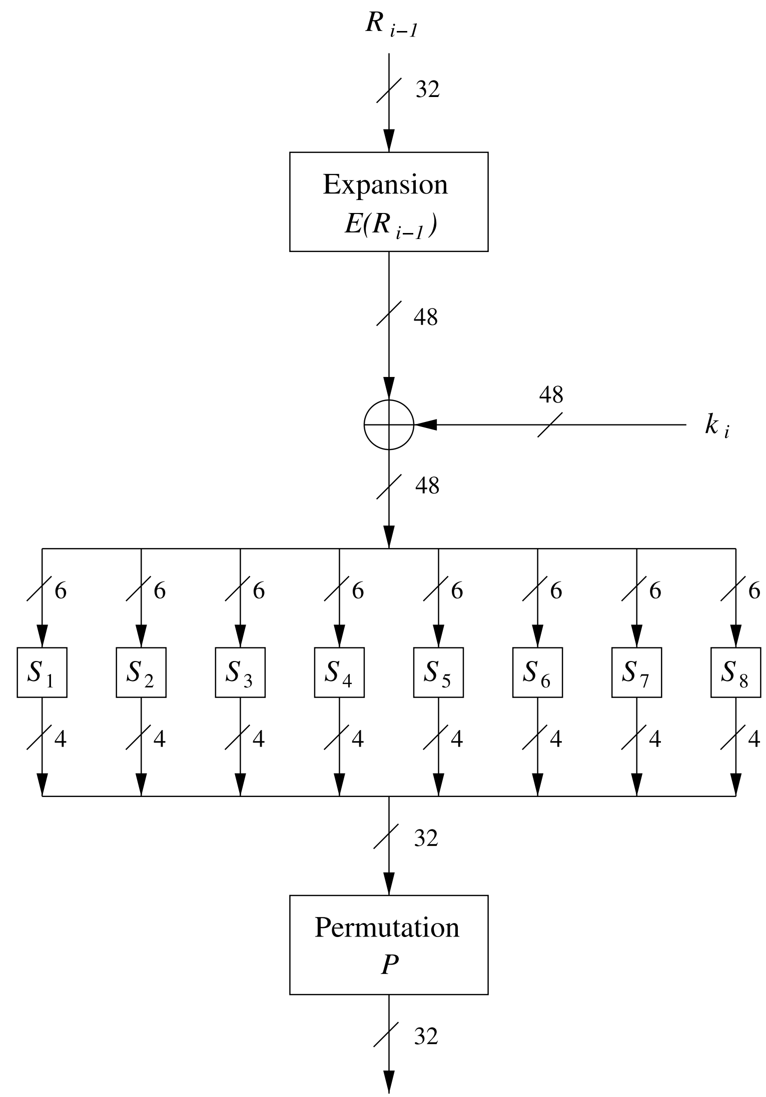
DES: THE f-FUNCTION

- Input of f function in round i is R_{i-1} and k_i
- Output can be thought of as a pseudorandom bit stream, and is XORd with L_{i-1}



DES: INTERNAL STRUCTURE OF f-FUNCTION

- E function expands the 32-bit input R_{i-1} to 48 bits
- The expanded input is XORd with the 48-bit subkey k_i
- The result is split into eight 6-bit chunks
- Each chunk is fed into an *S-box* that outputs 4 bits
- The P function permutes the 32-bit output from the S-boxes



DES: EXPANSION PERMUTATION OF f-FUNCTION

- The expansion permutation E partitions the 32-bit input into eight 4-bit chunks and expands each chunk to 6 bits
- Increases diffusion, since 16 of the 32 the input bits will each influence two output locations

E

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

DES: SUBSTITUTION BOXES (S-BOXES)

- Each S-box is a lookup table that maps a 6-bit input to a 4-bit output
- Provide confusion
- Each of the eight S-boxes are different, and carefully designed to meet a number of criteria
  - e.g., if two inputs to an S-box differ in exactly one bit, their outputs must differ in at least two bits
- Most crucial element of DES, since they provide non-linearity:

$$S(a) \oplus S(b) \neq S(a \oplus b).$$

DES: S-BOX S_1

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

- For a given 6-bit input:
  - The most- and least-significant bits select the row
  - The middle four bits select the column
  - The resulting cell in the table is the decimal representation of the 4-bit value to be substituted

DES: S-BOX S_1 (EXAMPLE)

S_1	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

Example. For a 6-bit input $b = 100101_2$:

- Select row $11_2 = 3$
- Select column $0010_2 = 2$
- 4-bit output is $08 = 1000_2$

DES: P-PERMUTATION OF f-FUNCTION

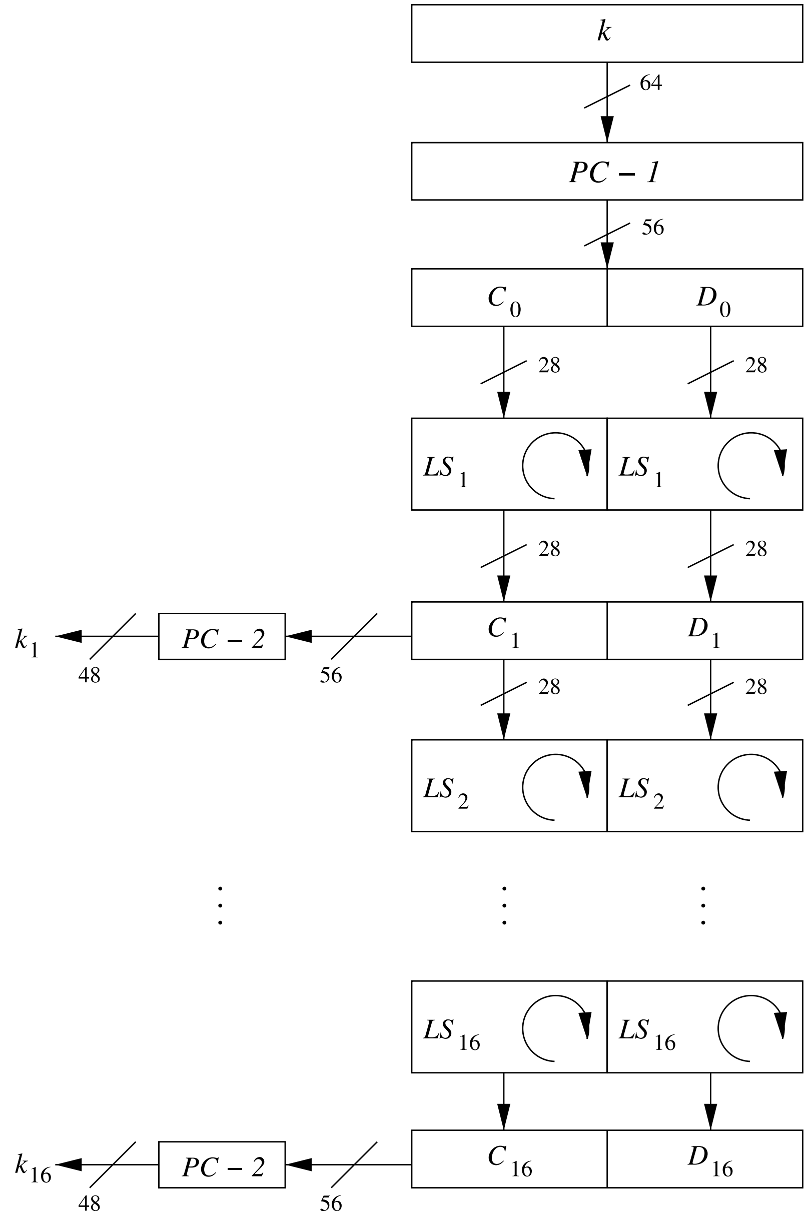
- The 32-bit output from the S-boxes is permuted so that the output bits from each S-box affects multiple S-boxes in the next round
- Introduces diffusion

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

DES: KEY SCHEDULE

- The key schedule derives 16 subkeys k_i from the secret key



DES: KEY SCHEDULE (INITIAL KEY PERMUTATION)

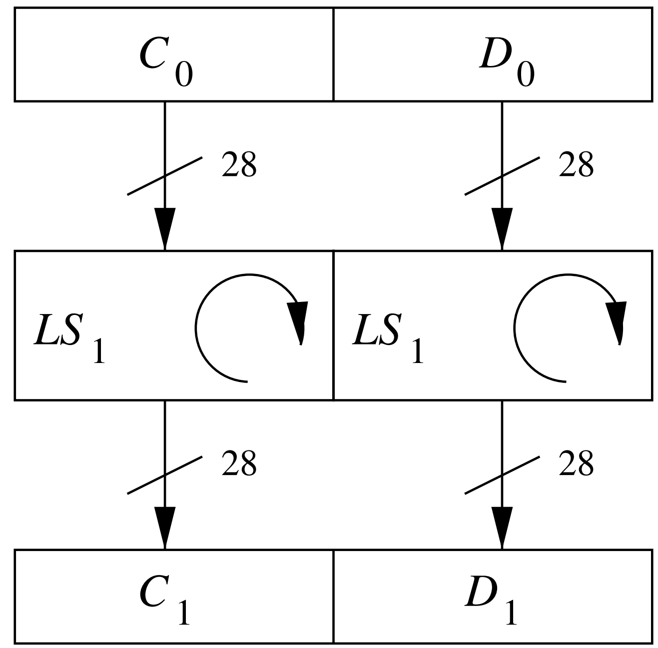
- The initial key permutation $PC-1$ reduces the 64-bit key to 56 bits by omitting every eigth bit
- The result is split into two halves C_0 and D_0

$PC-1$

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

DES: KEY SCHEDULE (SUBKEYS)

- In each round i, the two halves (C_i, D_i) are each circularly shifted to the left by:
  - One position if $i = 1, 2, 9, 16$
  - Two positions if $i \neq 1, 2, 9, 16$
- The total number of bit positions shifted is $4 \times 1 + 12 \times 2 = 28$, so $C_0 = C_{16}$ and $D_0 = D_{16}$



DES: KEY SCHEDULE (ROUND KEY PERMUTATION)

- $PC-2$ permutes and reduces (C_i, D_i) to output a 48-bit k_i
- Designed so that each bit in the key is used in approximately 14 of the 16 round keys

$PC-2$

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

NEXT TIME

- DES decryption
- Security of DES
- Triple DES

RECAP

- The security of block ciphers relies on operations that provide diffusion and confusion
- Non-linear operations are crucial for cryptographic strength
- Many block ciphers (including Feistel ciphers, and others) use an iterated design that repeatedly applies a round function, using a derived subkey for each round
- DES is a Feistel cipher and was the dominant symmetric encryption algorithm from the mid-1970s to mid-1990s