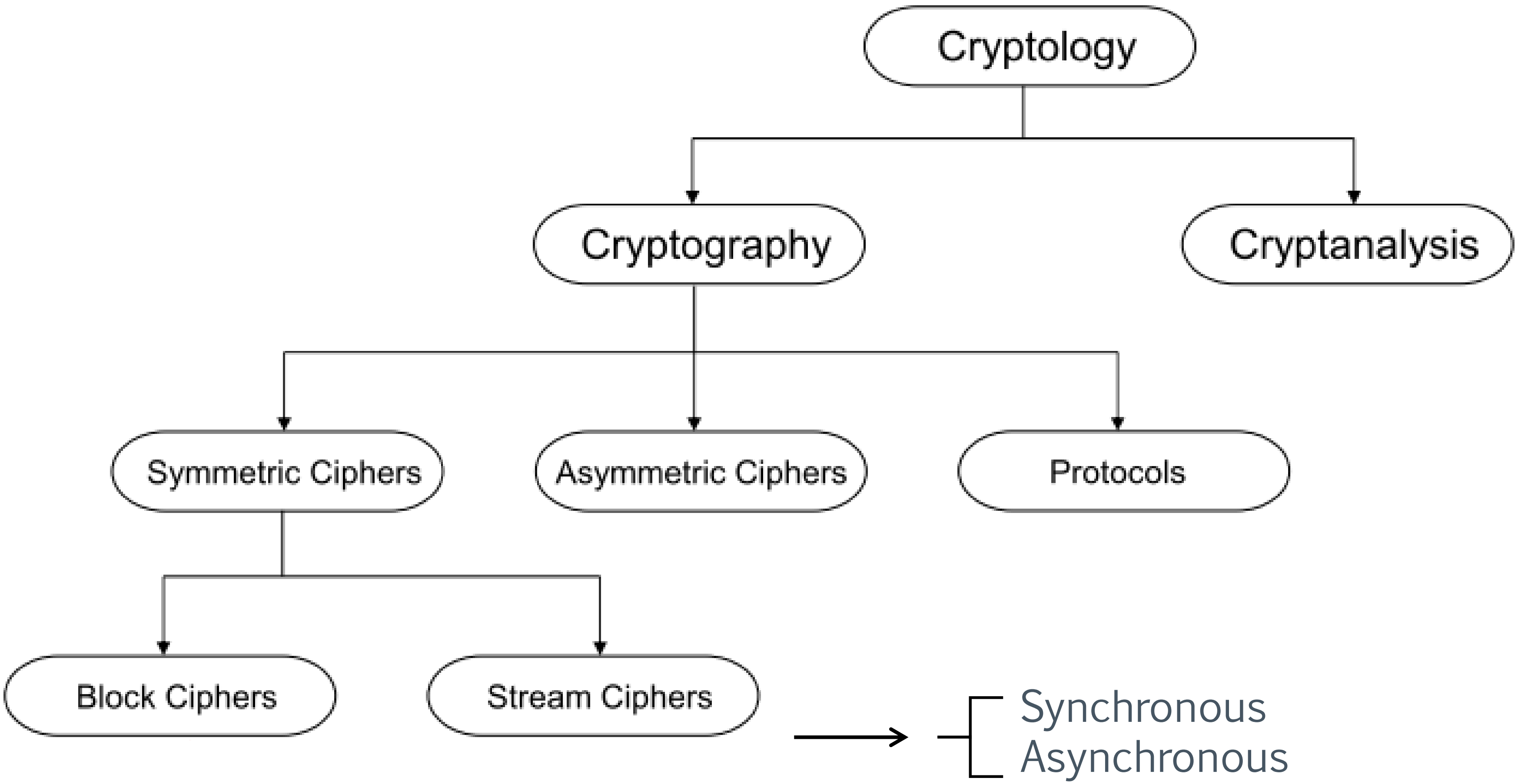
Chapter 3

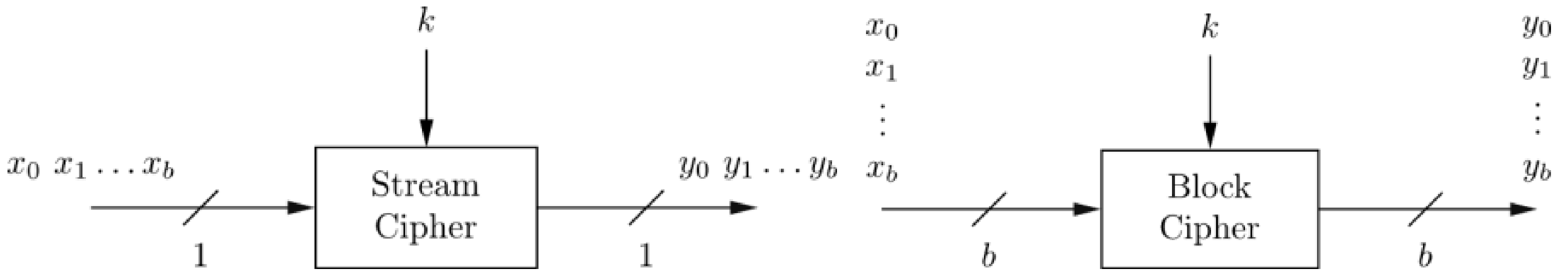
Block Ciphers and the Data Encryption Standard

Dr. Shin-Ming Cheng

Block vs Stream Ciphers



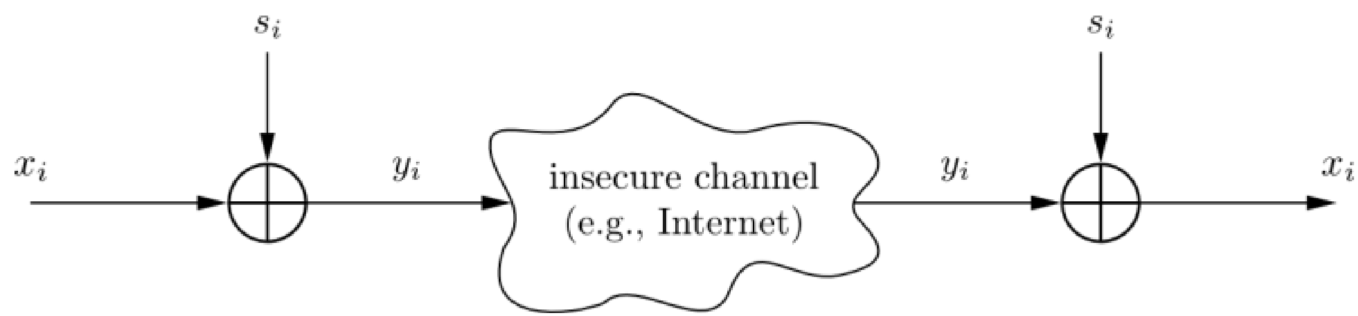
Stream Cipher vs. Block Cipher



- Stream Ciphers
  - Encrypt bits individually
  - Usually small and fast
    - common in embedded devices
      - A5/1 for GSM phones
- Block Ciphers:
  - Always encrypt a full block (several bits)
  - Are common for Internet applications

Encryption and Decryption with Stream Ciphers

Plaintext x_i, ciphertext y_i, and key stream s_i consist of individual bits

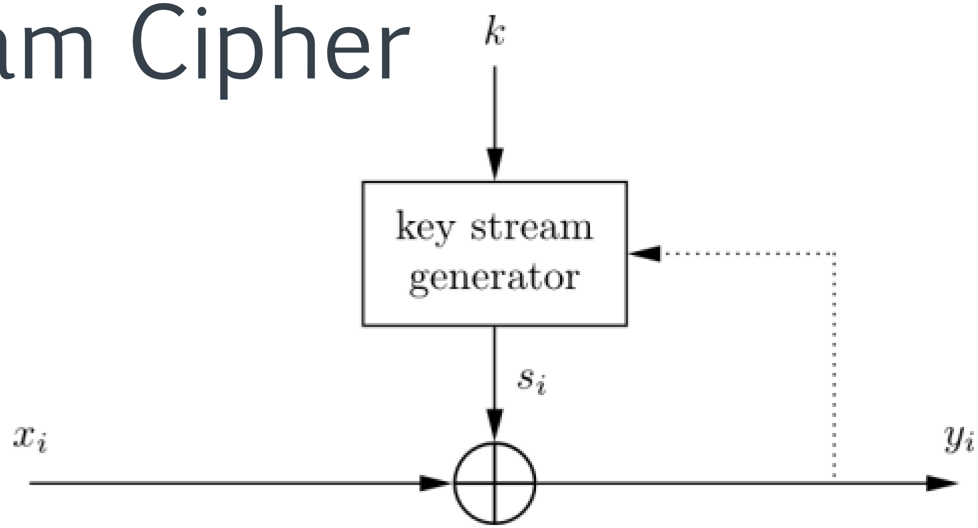


- Encryption and decryption are simple additions modulo 2 (aka XOR) Encrypt bits individually
- Encryption and decryption are the same functions

Encryption: $y_i = e_{s_i}(x_i) = x_i + s_i \bmod 2$ $\qquad x_i, y_i, s_i \in \{0, 1\}$

Decryption: $x_i = e_{s_i}(y_i) = y_i + s_i \bmod 2$

Synchronous vs. Asynchronous Stream Cipher



- Security of stream cipher depends entirely on the key stream s_i:
  - Should be random , i.e., $Pr(s_i = 0) = Pr(s_i = 1) = 0.5$
  - Must be reproducible by sender and receiver
- Synchronous Stream Cipher
  - Key stream depend only on the key
- Asynchronous Stream Ciphers
  - Key stream depends also on the ciphertext

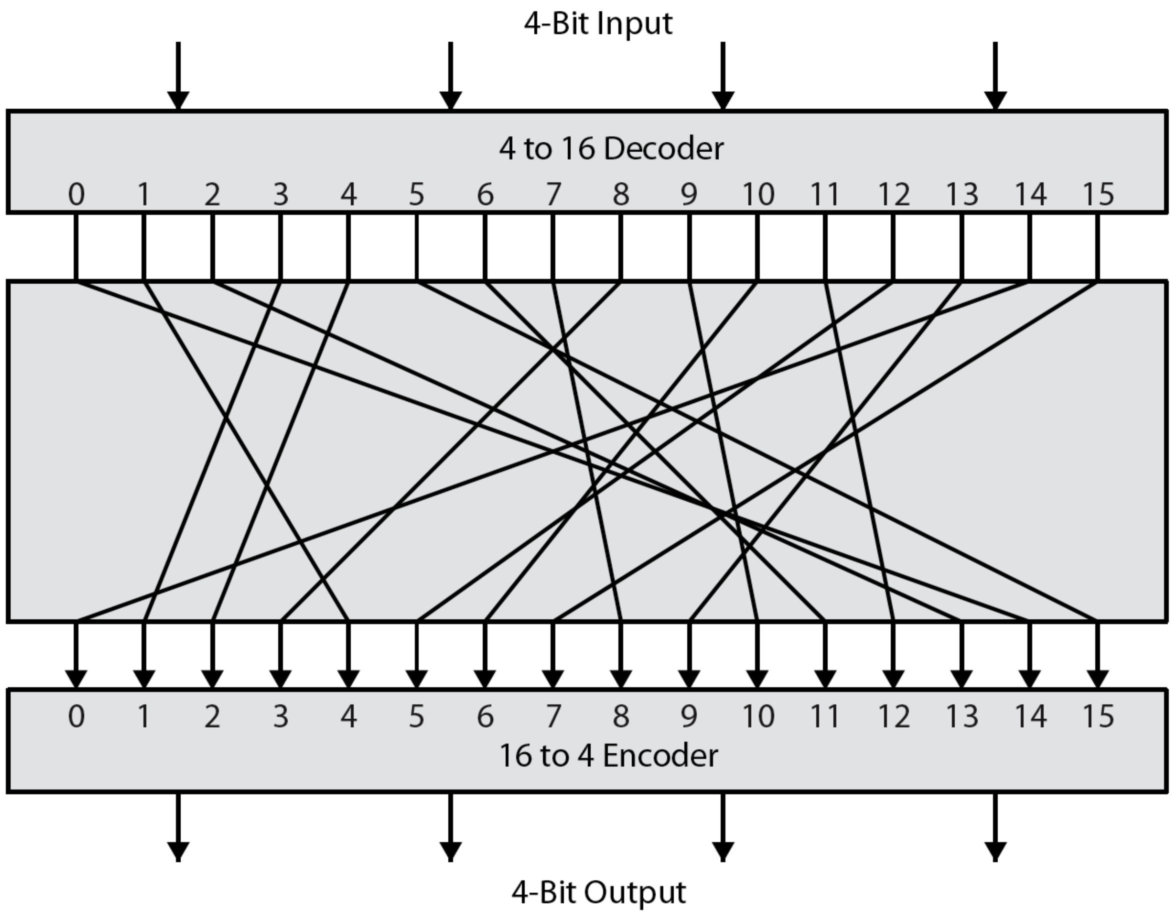
Modern Block Ciphers

- most widely used types of cryptographic algorithms
- provide secrecy /authentication services
- DES (Data Encryption Standard)
- illustrate block cipher design principles

Block Cipher Principles

- Most symmetric block ciphers are based on a **Feistel Cipher Structure**
  - decrypt ciphertext to recover messages efficiently
- block ciphers look like an extremely large substitution
  - need table of 2^{64} entries for a 64-bit block
  - instead create from smaller building blocks
- use the idea of a product cipher

Ideal Block Cipher



DES History

- IBM developed Lucifer cipher
  - by team led by 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
- IBM submitted their revised Lucifer which was eventually accepted as the DES

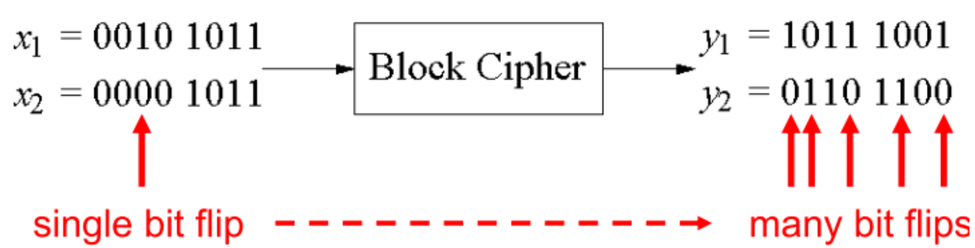
Block Cipher Primitives: Confusion and Diffusion

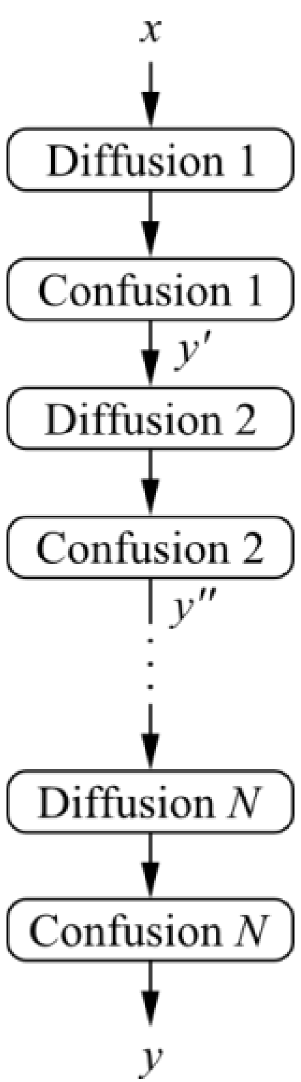
- Confusion:
  - An encryption operation where the **relationship between key and ciphertext is obscured.**
  - Substitution
- Diffusion:
  - An encryption operation where the **influence of one plaintext symbol is spread over many ciphertext symbols** with the goal of hiding statistical properties of the plaintext.
  - Bit permutation
- Both operations by themselves cannot provide security.
- The idea is to concatenate confusion and diffusion elements to build so called product ciphers

Product Ciphers

- Consist of rounds which are applied repeatedly to the data
- Reach excellent diffusion
  - changing of one bit of plaintext results on average in the change of half the output bits

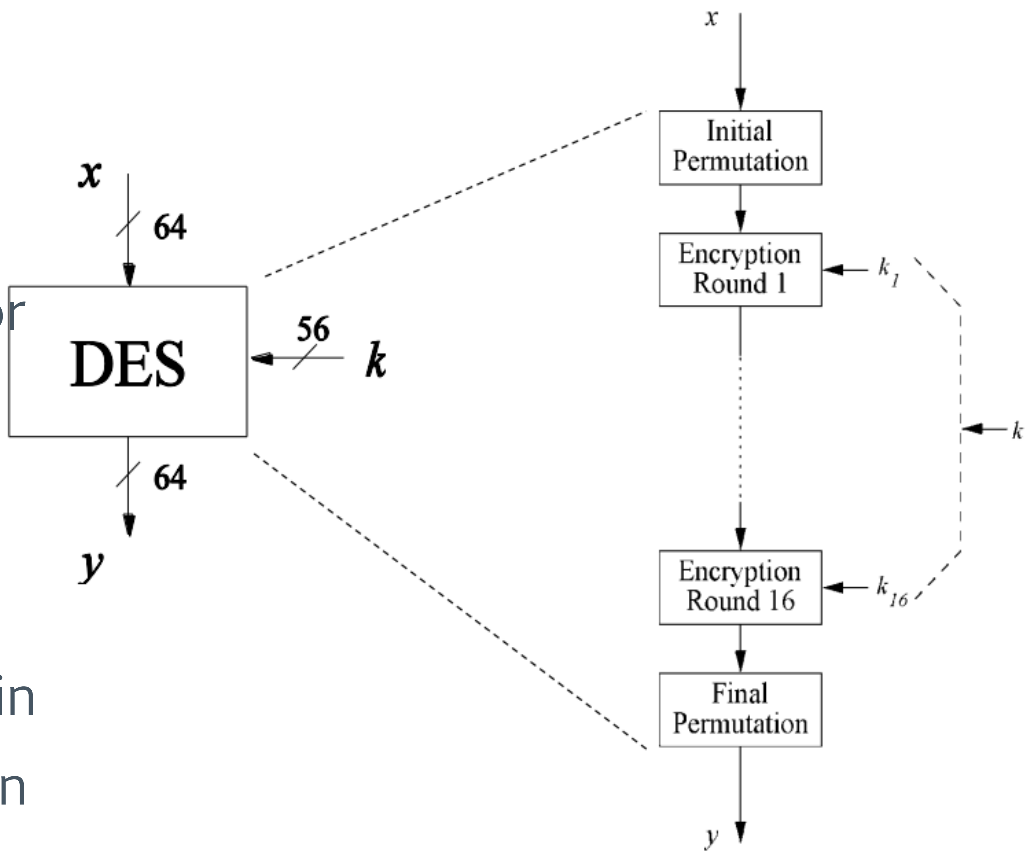
Example:





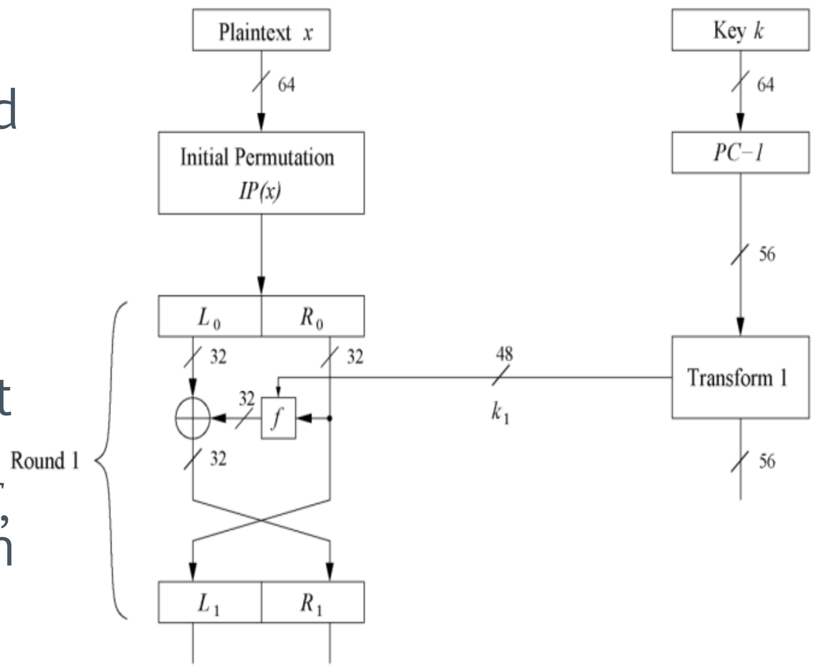
Overview of the DES Algorithm

- Encrypts blocks of size 64 bits
- Uses a key of size 56 bits
- Symmetric cipher
  - uses same key for encryption and decryption
- Uses 16 rounds which all perform the identical operation
  - Different subkey in each round derived from main key



The DES Feistel Network (1)

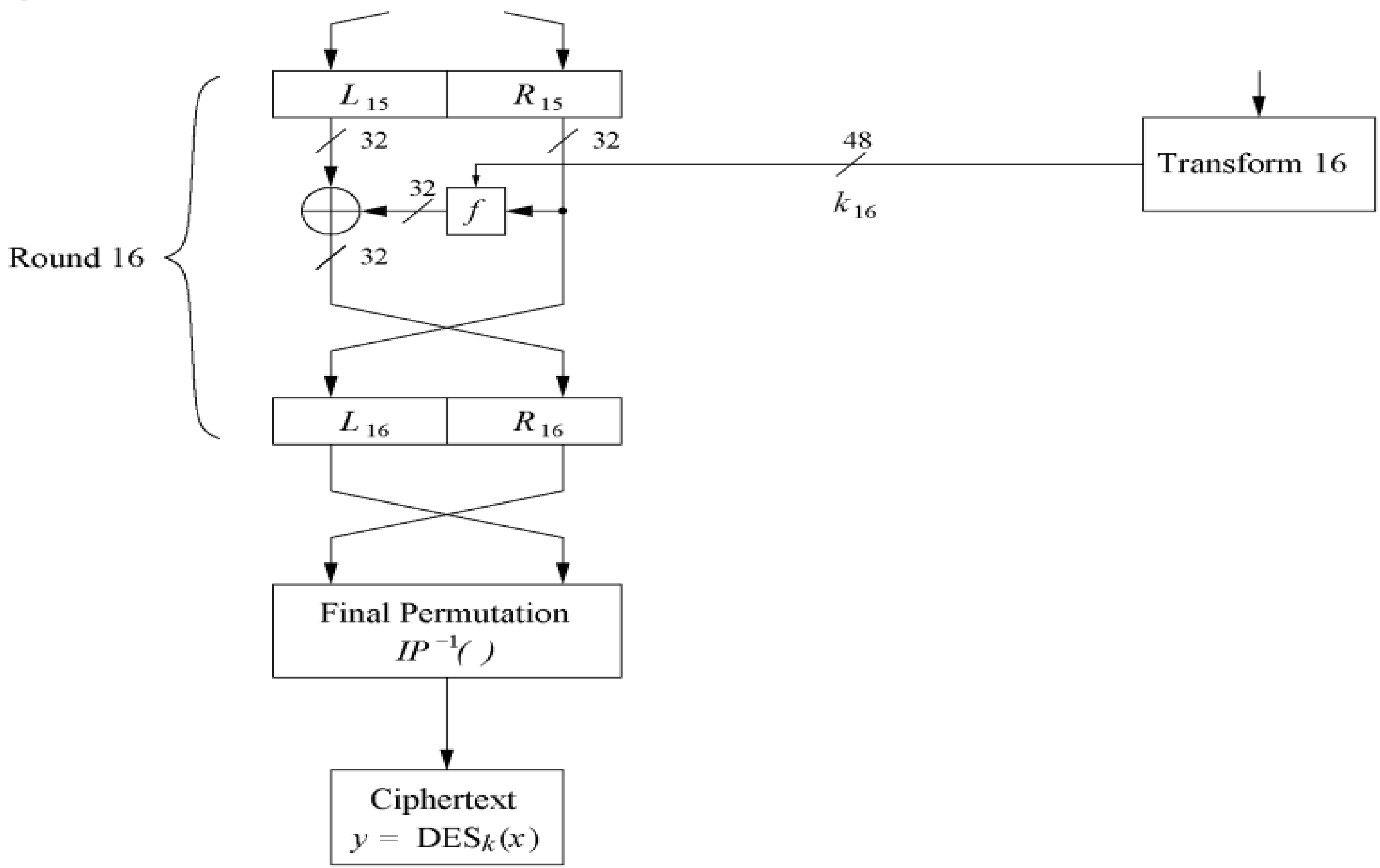
- DES structure is a Feistel network
  - Advantage: encryption and decryption differ only in key schedule
- Bitwise initial permutation, then 16 rounds
  - Plaintext is split into 32-bit halves L_i and R_i
  - R_i is fed into the function f, the output of which is then XORed with L_i
  - Left and right half are swapped
- Rounds can be expressed as:



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

The DES Feistel Network (2)

- L and R swapped again at the end of the cipher, i.e., after round 16 followed by a final permutation



Initial and Final Permutation

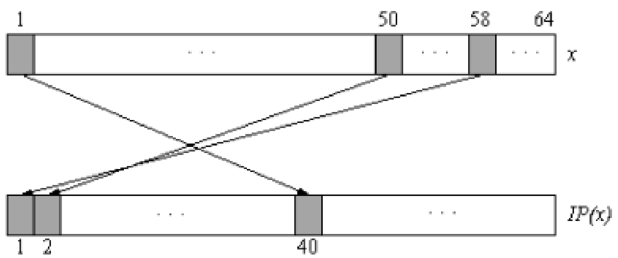
› Bitwise Permutations

› Inverse operations

– Described by tables IP and IP^{-1}

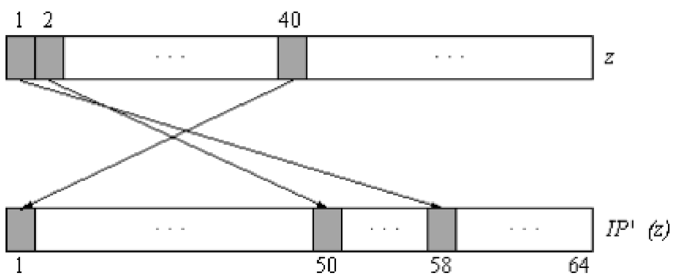
Initial Permutation

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



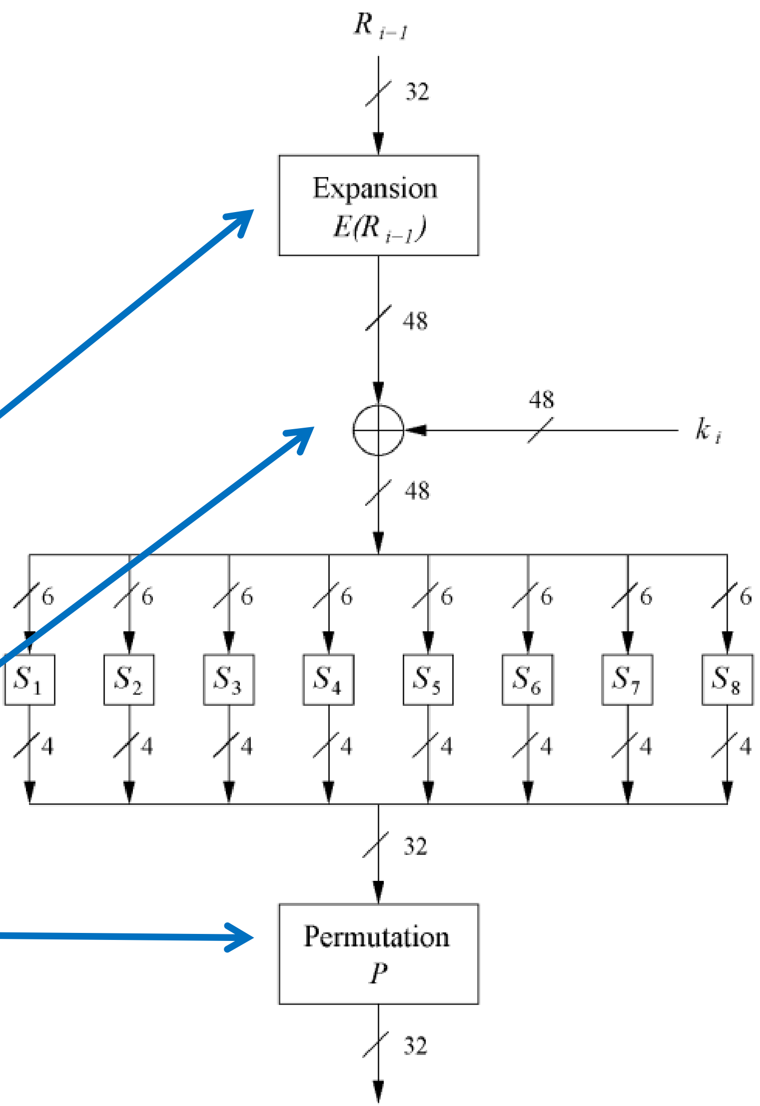
Final Permutation

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



The f-Function

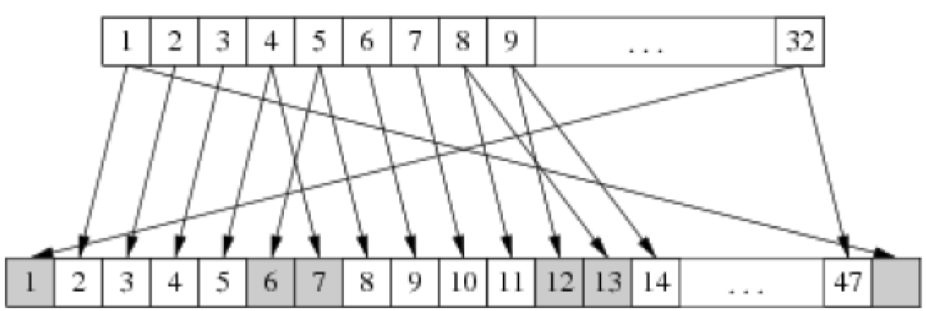
- Main operation of DES
- f-function inputs:
  - R_{i-1} and round key k_i
- 4 Steps:
  - Expansion E
  - XOR with round key
  - S-box substitution
  - Permutation

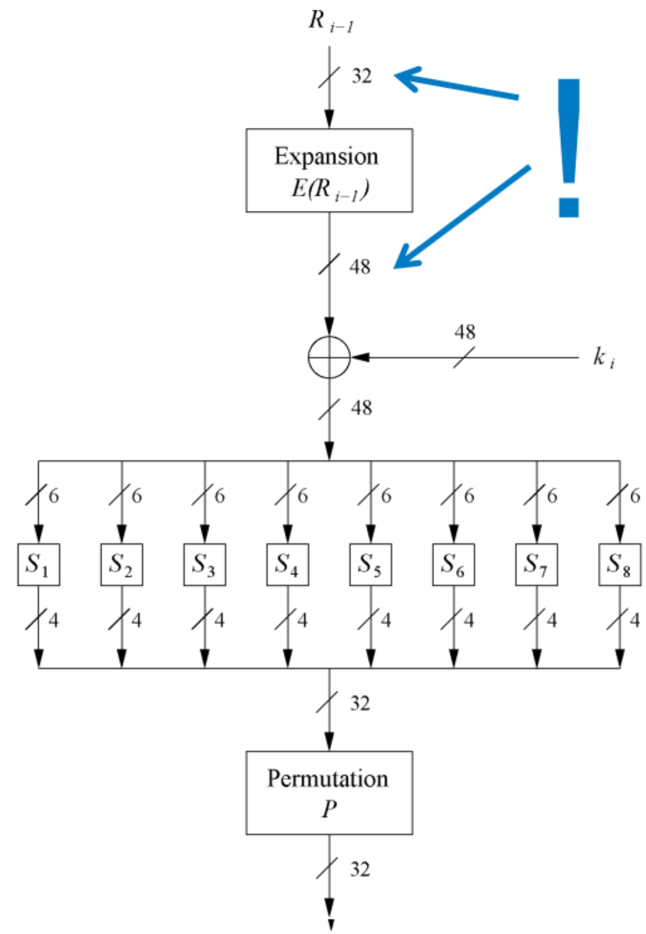


The Expansion Function E

- main purpose
  - increases diffusion

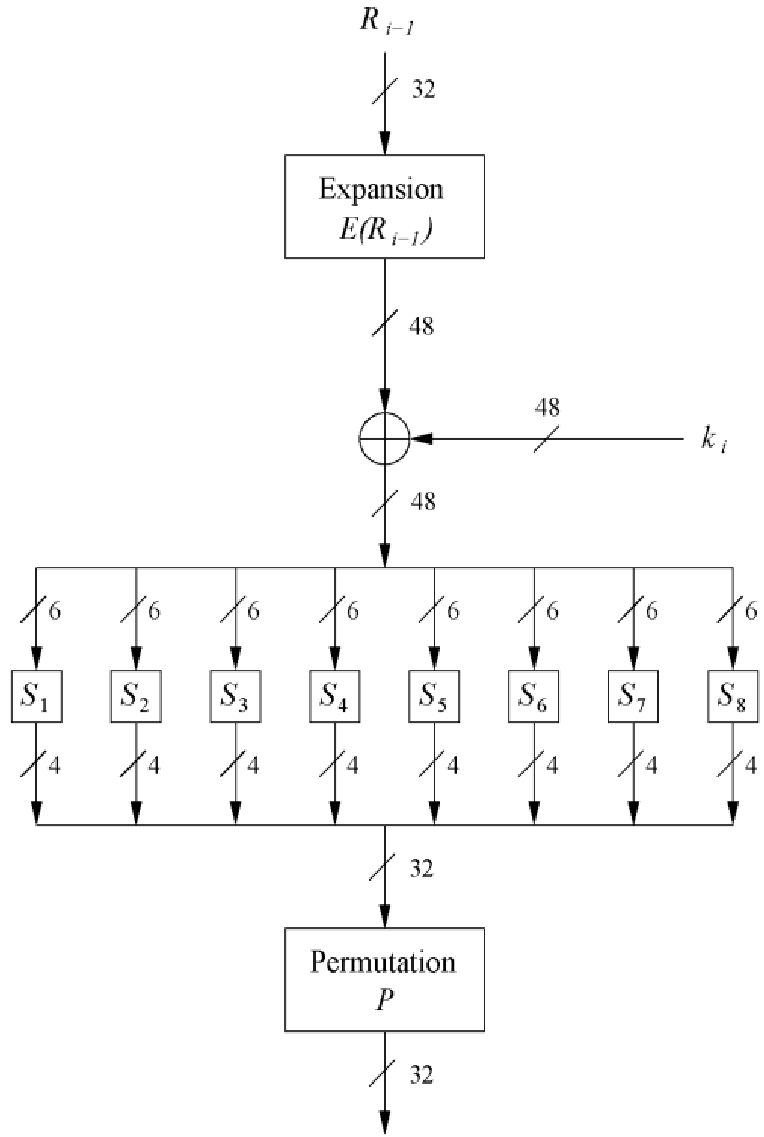
E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1





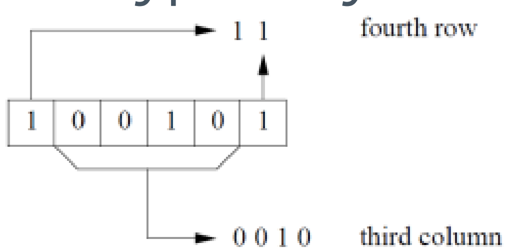
Add Round Key

- Bitwise XOR of the round key and the output of the expansion function E
- Round keys are derived from the main key in the DES keyschedule (in a few slides)

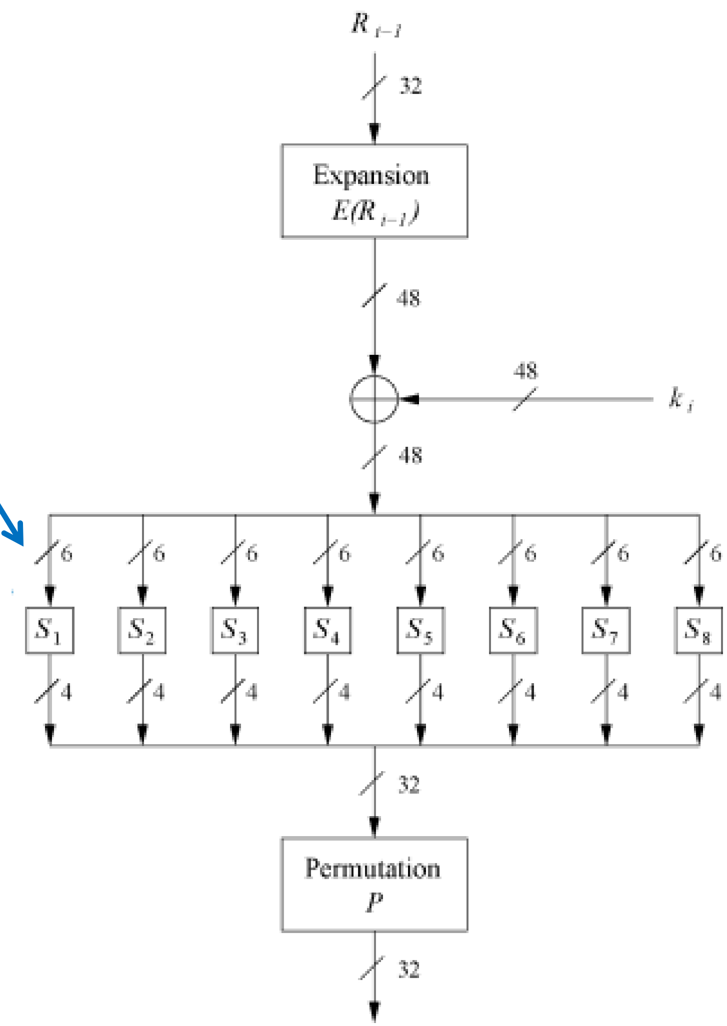


The DES S-Boxes

- Eight substitution tables
  - 6 bits of input, 4 bits of output
- Crucial element
  - Non-linear
  - resistant to differential cryptanalysis

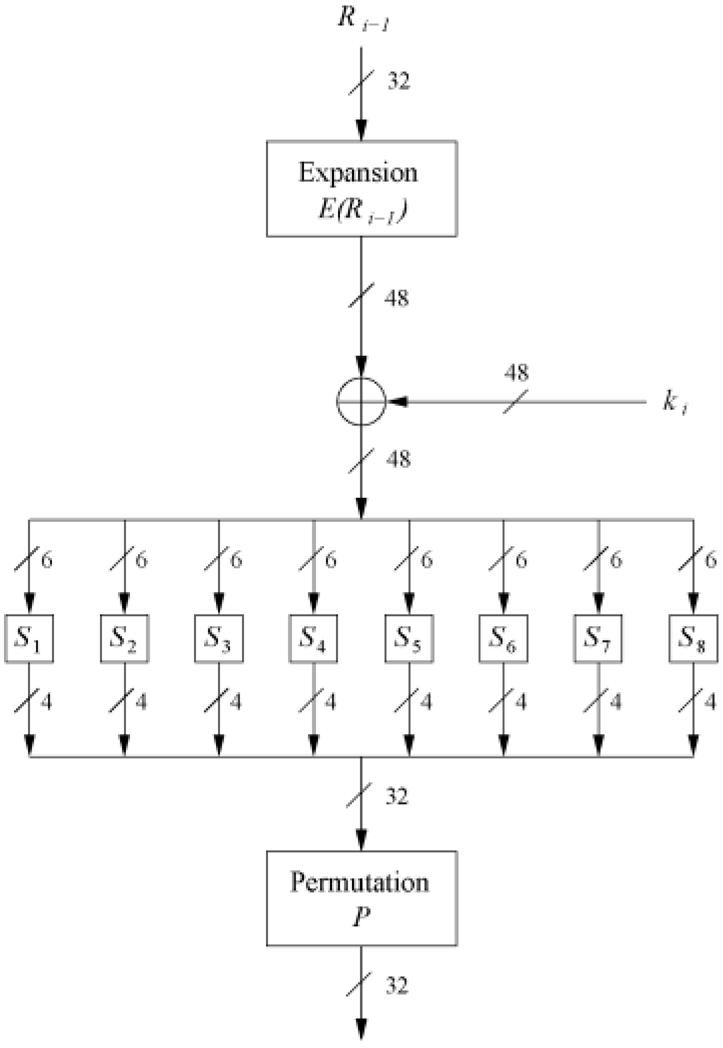


S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13



The Permutation P

- Bitwise permutation
  - Introduces diffusion
  - Output bits of one S-Box effect several S-Boxes in next round
- Diffusion by E, S-Boxes and P guarantees
  - after Round 5 every bit is a function of each key bit and each plaintext bit

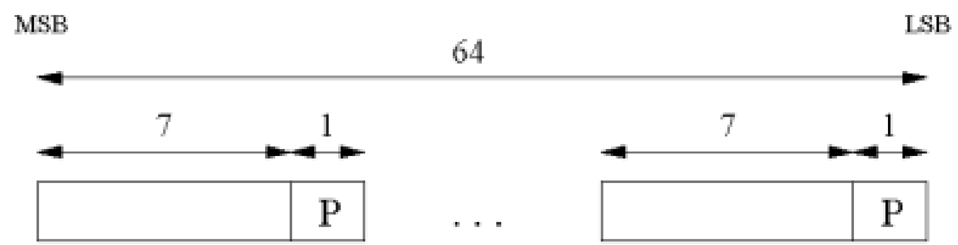


P							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Key Schedule (1/2)

- Derives 16 round keys (or subkeys) k_i of 48 bits each from the original 56 bit key
- The input key size of the DES is 64 bit ->56 bit key and 8 bit parity:

$PC-1$							
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

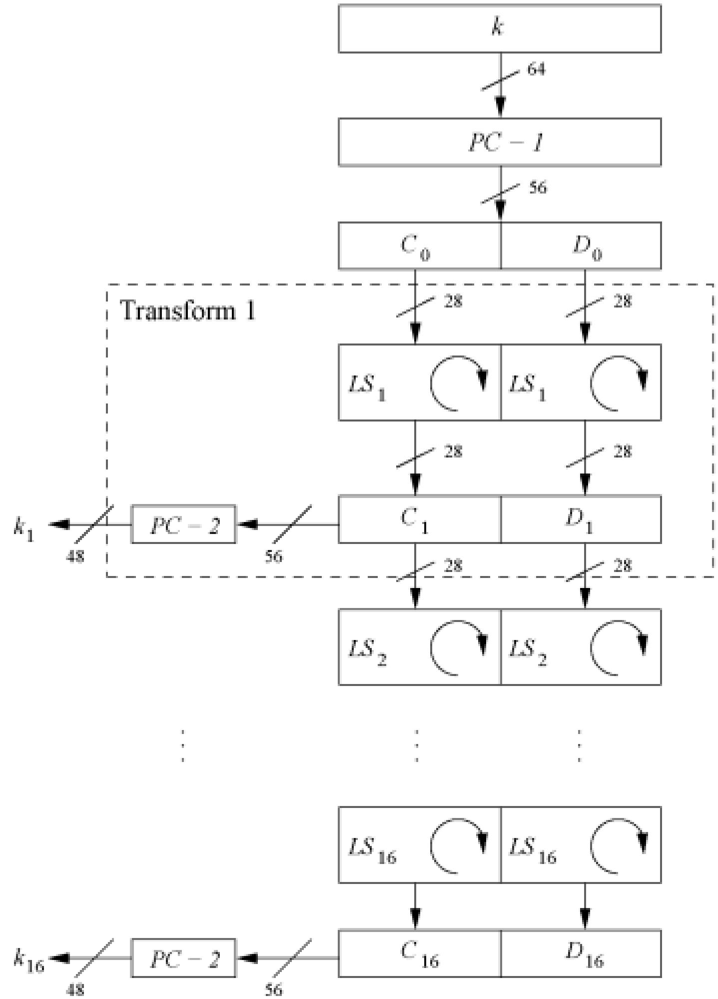


P = parity bit

  - Parity bits are removed in a first permuted choice PC –1: the bits 8, 16, 24, 32, 40, 48, 56 and 64 are not used at all)

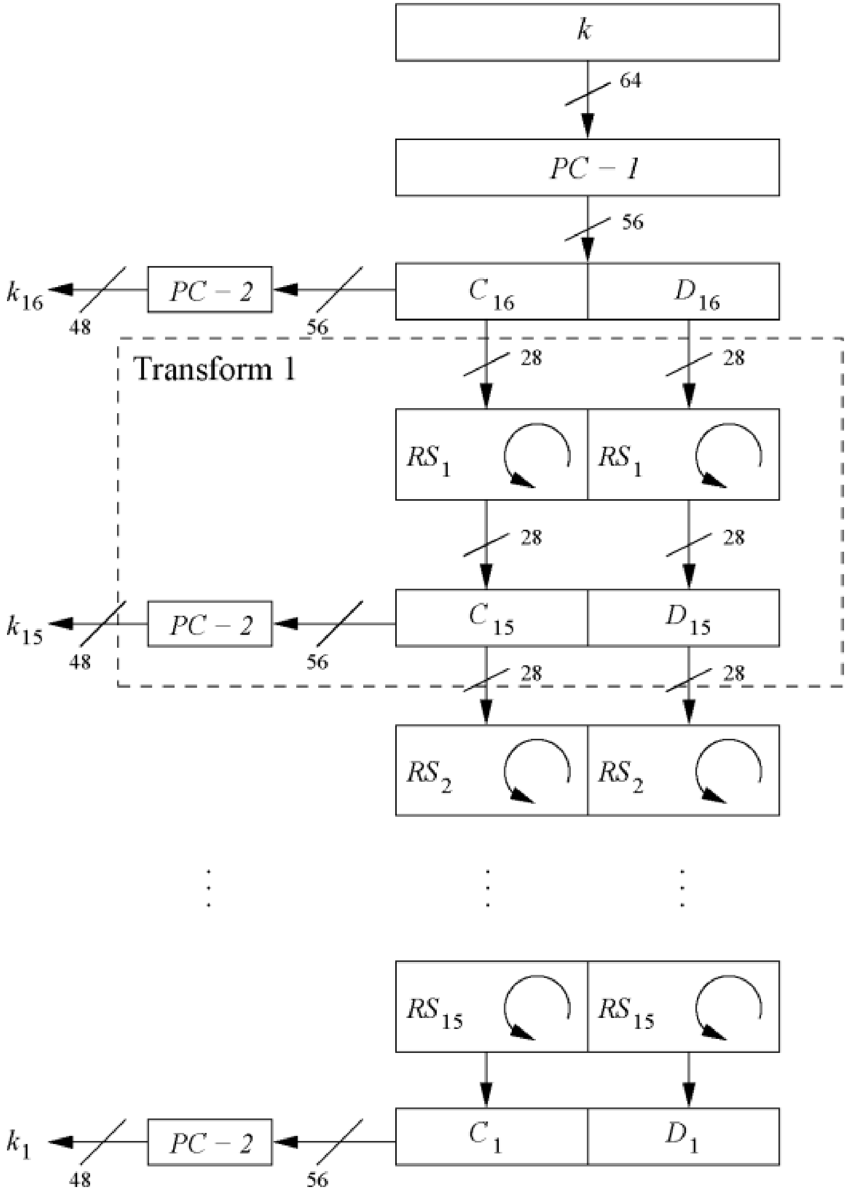
Key Schedule (2/2)

- Split key into 28-bit halves C_0 and D_0
  - In rounds i = 1, 2, 9, 16, the two halves are each rotated left by one bit
  - In all other rounds where the two halves are each rotated left by two bits
- In each round i permuted choice PC–2 selects a permuted subset of 48 bits of C_i and D_i as round key k_i,
  - each k_i is a permutation of k !
- The total number of rotations:
  - 4 X 1 + 12 X 2 = 28 => D_0=D_{16} and C_0 = C_{16}



Decryption

- Generate the same 16 round keys in reverse order
- Reversed key schedule:
  - As D_0=D_{16} and C_0 = C_{16} the first round key can be generated by applying PC – 2 right after PC – 1 No rotation in round 1
- One bit rotation to the right in rounds 2, 9 and 16
- Two bit rotations to the right in all other rounds



Security of DES (1/2)

- Major criticisms
  - Key space is too small (2^{56} keys)
  - S-box design criteria have been kept secret
    - Are there any hidden analytical attacks (backdoors), only known to the NSA?
- Exhaustive key search:
  - For a given pair of plaintext-ciphertext (x, y)
  - Test all 2^{56} keys until the condition $DES_k^{-1}(y) = x$ is fulfilled
  - Relatively easy given today's computer technology

Security of DES (2/2)

- Analytical Attacks:
  - DES is highly resistent to both differential and linear cryptanalysis, which have been published years later than the DES.
    - This means IBM and NSA had been aware of these attacks for 15 years!
  - So far there is no known analytical attack which breaks DES in realistic scenarios.

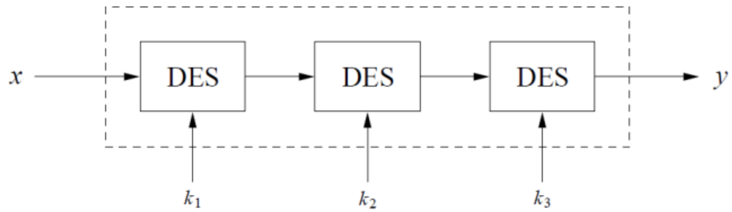
History of Attacks on DES

Year	Proposed / implemented DES Attack
1977	Diffie & Hellman, (under-)estimate the costs of a key search machine
1990	Biham & Shamir propose differential cryptanalysis (2^{47} chosen ciphertexts)
1993	Mike Wiener proposes design of a very efficient key search machine: Average search requires 36h. Costs: $1,000,000
1993	Matsui proposes linear cryptanalysis (2^{43} chosen ciphertexts)
Jun. 1997	DES Challenge I broken, 4.5 months of distributed search
Feb. 1998	DES Challenge II--1 broken, 39 days (distributed search)
Jul. 1998	DES Challenge II--2 broken, key search machine *Deep Crack* built by the Electronic Frontier Foundation (EFF): 1800 ASICs with 24 search engines each, Costs: $250 000, 15 days average search time (required 56h for the Challenge)
Jan. 1999	DES Challenge III broken in 22h 15min (distributed search assisted by *Deep Crack*)
2006-2008	Reconfigurable key search machine *COPACOBANA* developed at the Universities in Bochum and Kiel (Germany), uses 120 FPGAs to break DES in 6.4 days (avg.) at a cost of $10 000.

Triple DES – 3DES

- Triple encryption using DES is often used in practice to extend the effective key length of DES to 112.

$$y = DES_{k_3}(DES_{k_2}(DES_{k_1}(x)))$$



- Alternative version of 3DES: $y = DES_{k_3}(DES^{-1}_{k_2}(DES_{k_1}(x)))$.
- Choosing $k_1 = k_2 = k_3$ performs single DES encryption
- No practical attack known today
- Used in many legacy applications,
  - banking systems

Lessons Learned

- DES was the dominant symmetric encryption algorithm from the mid-1970s to the mid-1990s.
  - Since 56-bit keys are no longer secure, the Advanced Encryption Standard (AES) was created
- Standard DES with 56-bit key length can be broken relatively easily nowadays through an exhaustive key search
- DES is quite robust against known analytical attack
  - In practice it is very difficult to break the cipher with differential or linear cryptanalysis
- By encrypting with DES three times in a row, triple DES (3DES) is created, against which no practical attack is currently known