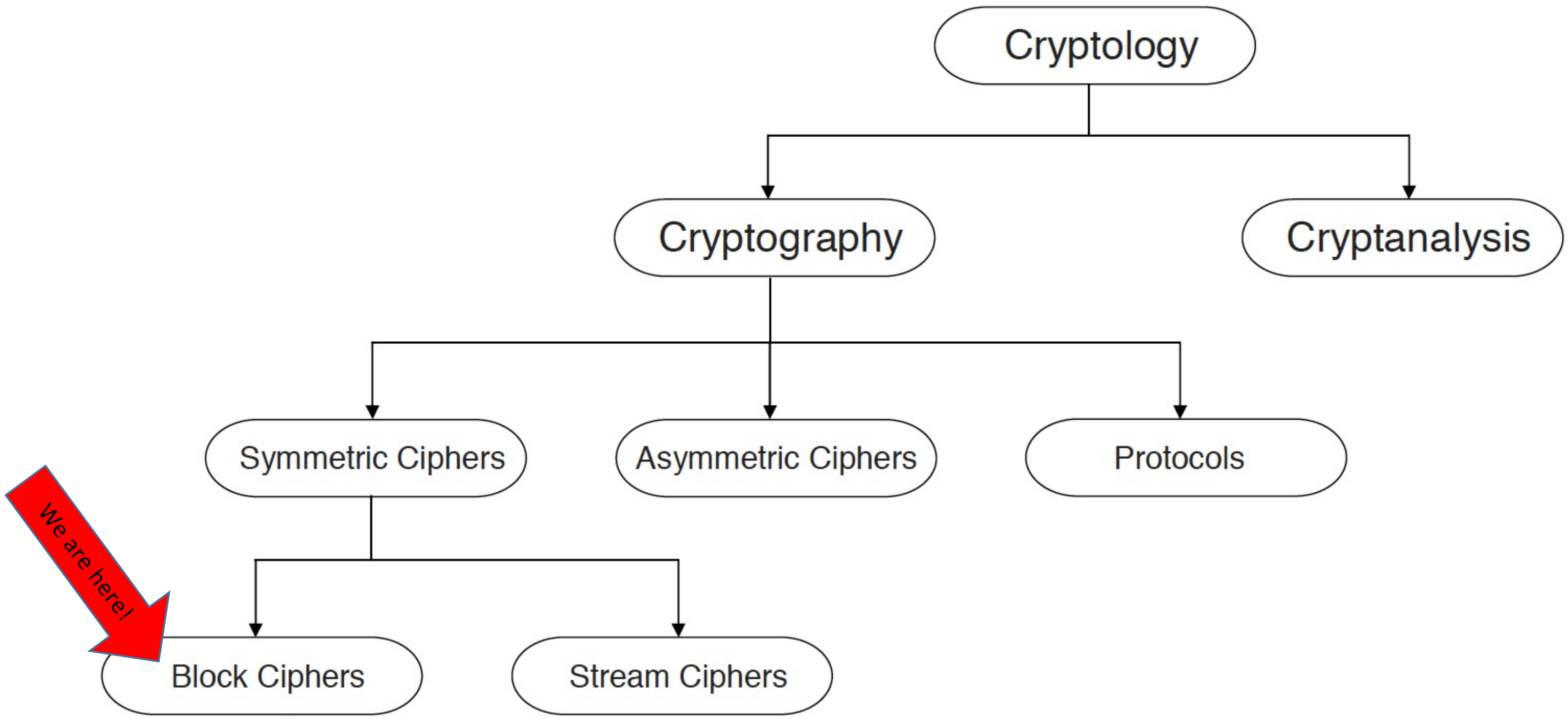
Chapter 3: Data Encryption Standard (DES) and Alternatives

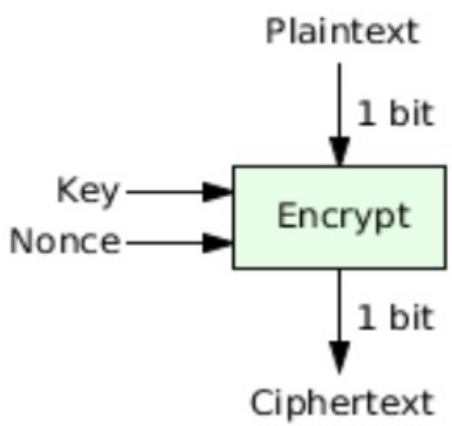
- Block cipher overview
- Feistel Schemes
- DES

Where are we now?

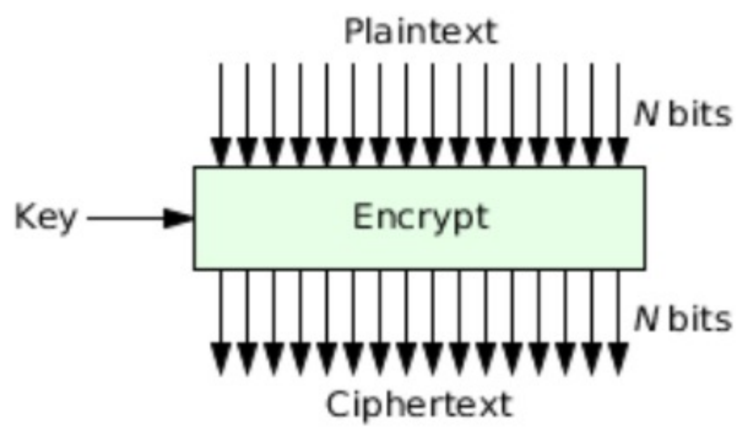


Stream Cipher vs. Block Cipher

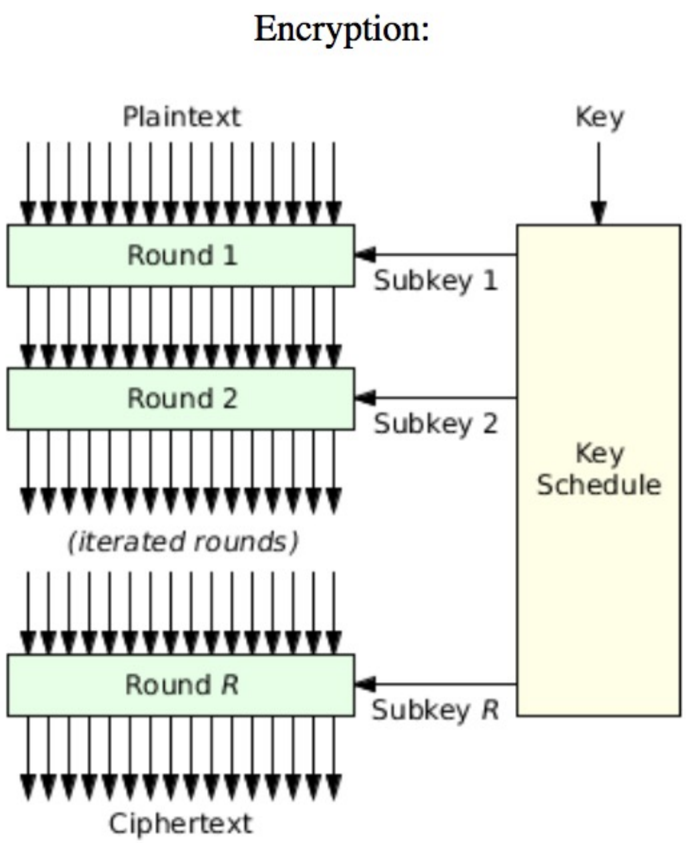
Stream Cipher:



Block Cipher:



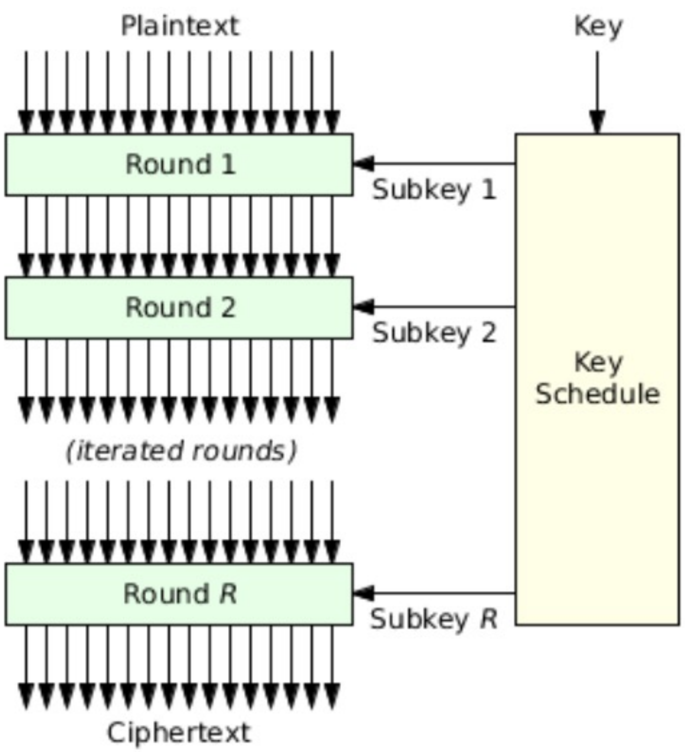
Block Cipher Architecture



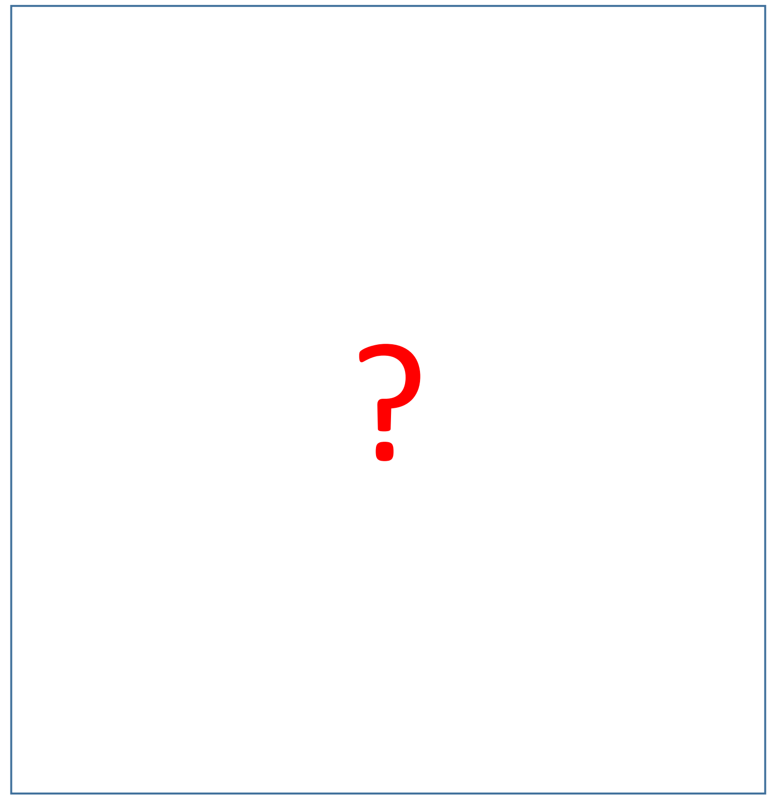
- Every block cipher used in practice consists of several **rounds**, each of which performs **identical** operations.
  - XOR, Substitution, Permutation
- From the **main key**, **subkeys** (or **round keys**) are generated by a key schedule algorithm so that each round uses a different key.

Block Cipher Architecture

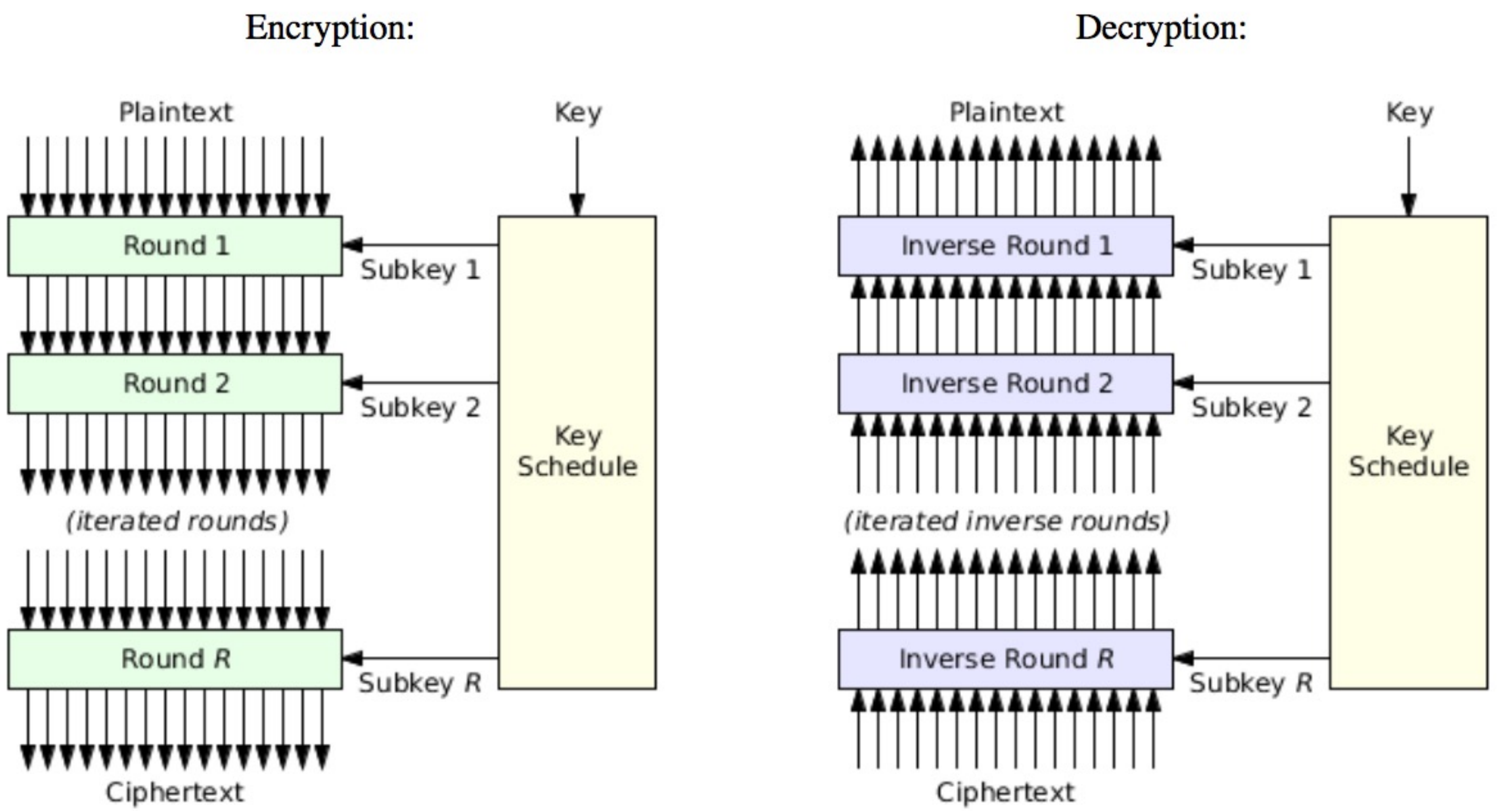
Encryption:



Decryption:

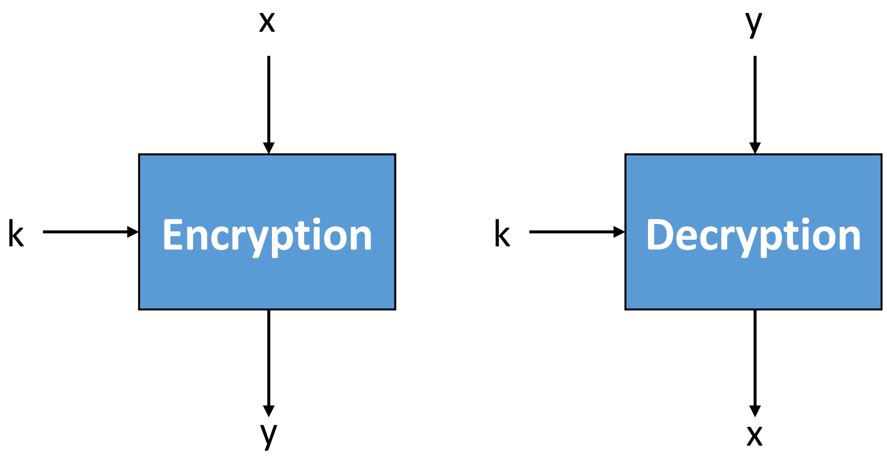


Block Cipher Architecture



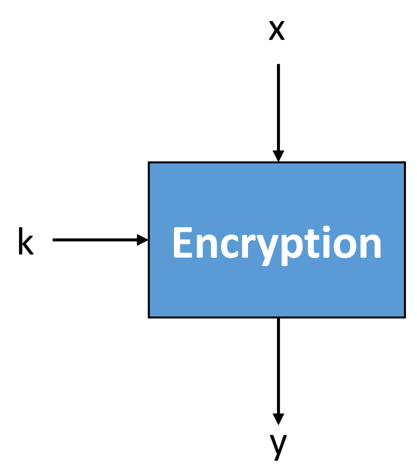
Notable Block Ciphers

Cipher	Block size (bits)	Key size (bits)
DES	64	56
3DES	64	112, 168
AES	128	128, 192, 256
Blowfish	64	32 -- 448
Twofish	128	-- 256
Serpent	128	128, 192, 256
…	…	…



Why Block Size Matters?

- The **block size** and **key size** are two critical parameters that define a block cipher.
  - **Security** and **performance** depend on both values.
  - Keys that are too short are susceptible to brute-force attacks.
  - Longer block or key sizes typically result in slower encryption and decryption processes.
  - A block size that is too short can also have implications for security, including vulnerabilities to attacks like **code book attacks**.



The Codebook Attack

- Consider an example of a block cipher with 16-bit blocks (and a possibly long key).
  1. There are only 65536 (2^{16}) possible ciphertexts exist.
  2. It may be feasible for an attacker to build a lookup table (aka a codebook) mapping each ciphertext block to its corresponding plaintext block.
  3. To decrypt an unknown ciphertext block, look up its corresponding plaintext block in the table.
- When 16-bit blocks are used, the lookup table needs only 2^{16} entries, which is manageable.
- But with 64-bit blocks, you'd have to store 2^{64} entries (a zetabit), which is not manageable.
  - Codebook attacks won't be an issue for larger blocks.

Codebook

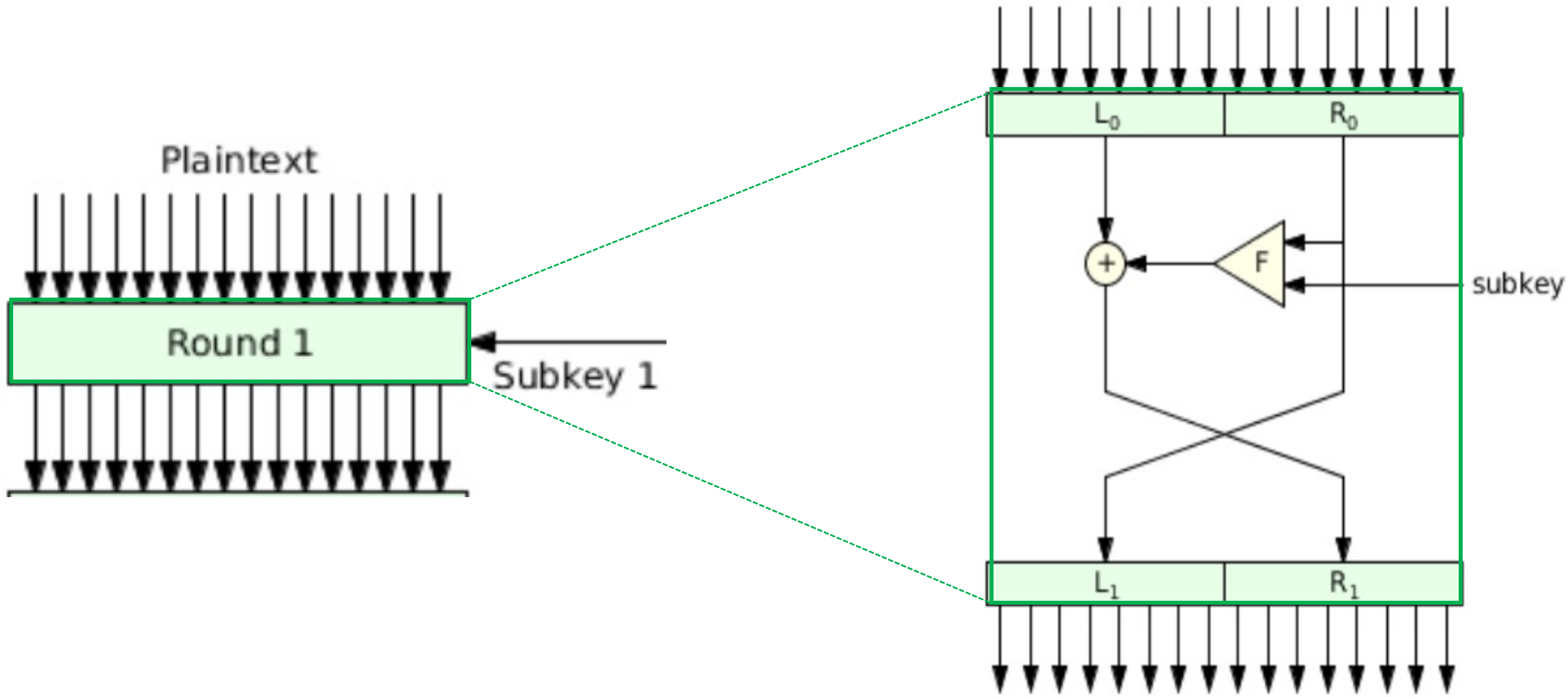
ciphertext	plaintext
1010111100101100	0011001011100101
0010111010010111	1101011100100011
….	….

The Feistel Structure

Cipher	Block size (bits)	Key size (bits)
DES (**F**)	64	56
3DES (**F**)	64	112, 168
AES	128	128, 192, 256
Blowfish (**F**)	64	32 -- 448
Twofish (**F**)	128	-- 256
Serpent	128	128, 192, 256
…	…	…

- The **Feistel** network structure is a cryptographic construction used in the design of block ciphers.
- It was introduced by **Horst Feistel** in the early 1970s and has been widely adopted due to its simplicity and effectiveness in creating **invertible** cryptographic functions.

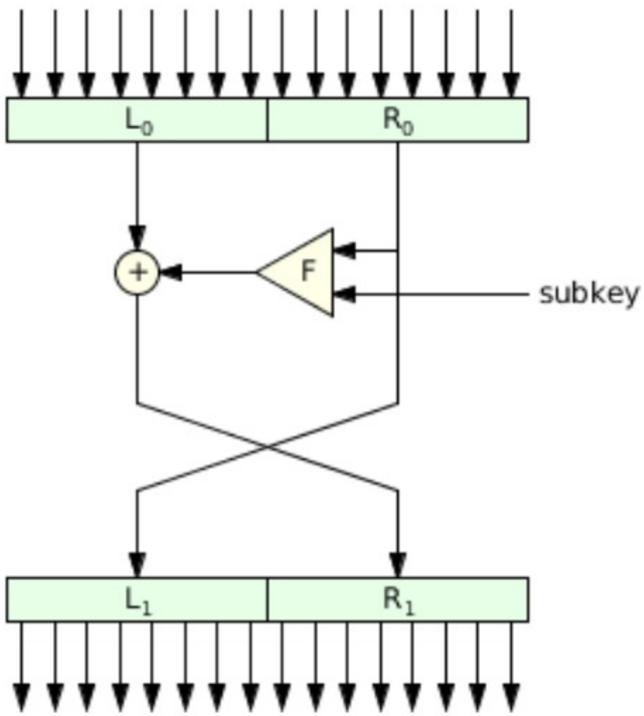
Feistel Schemes



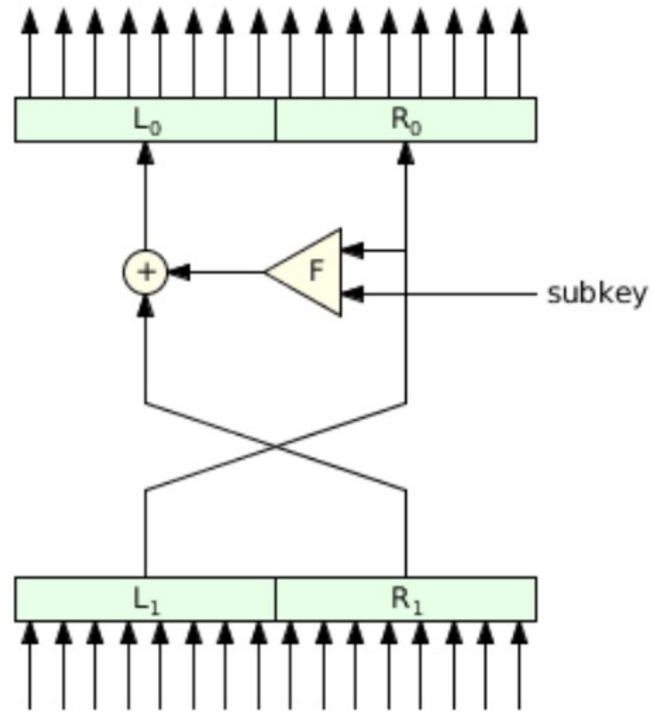
Feistel Schemes

Show that decryption works. You need to prove

1. L1 == R0
2. R1 + F(L1, subkey) == L0

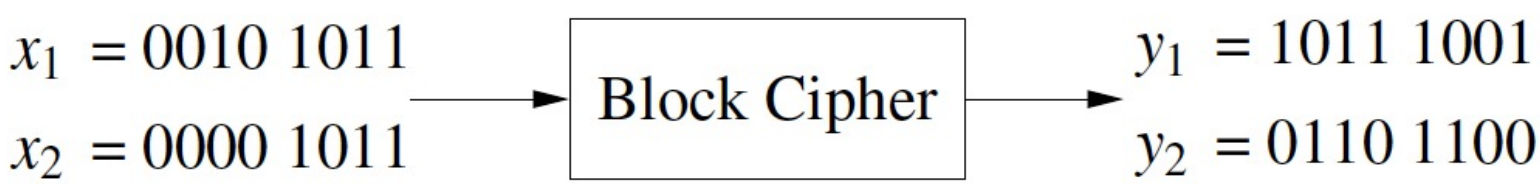


$L_1 = R_0$
$R_1 = L_0 + F(R_0, \text{subkey})$



Diffusion Property

- Changing of one bit of plaintext results *on average* in the change of half the output bits.
  - The second ciphertext looks statistically independent of the first one.
- Example: Assume a small block cipher with a block length of 8 bits. Encryption of two plaintexts x_1 and x_2

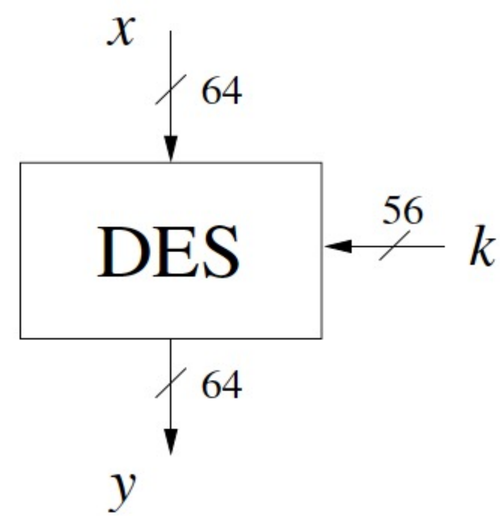


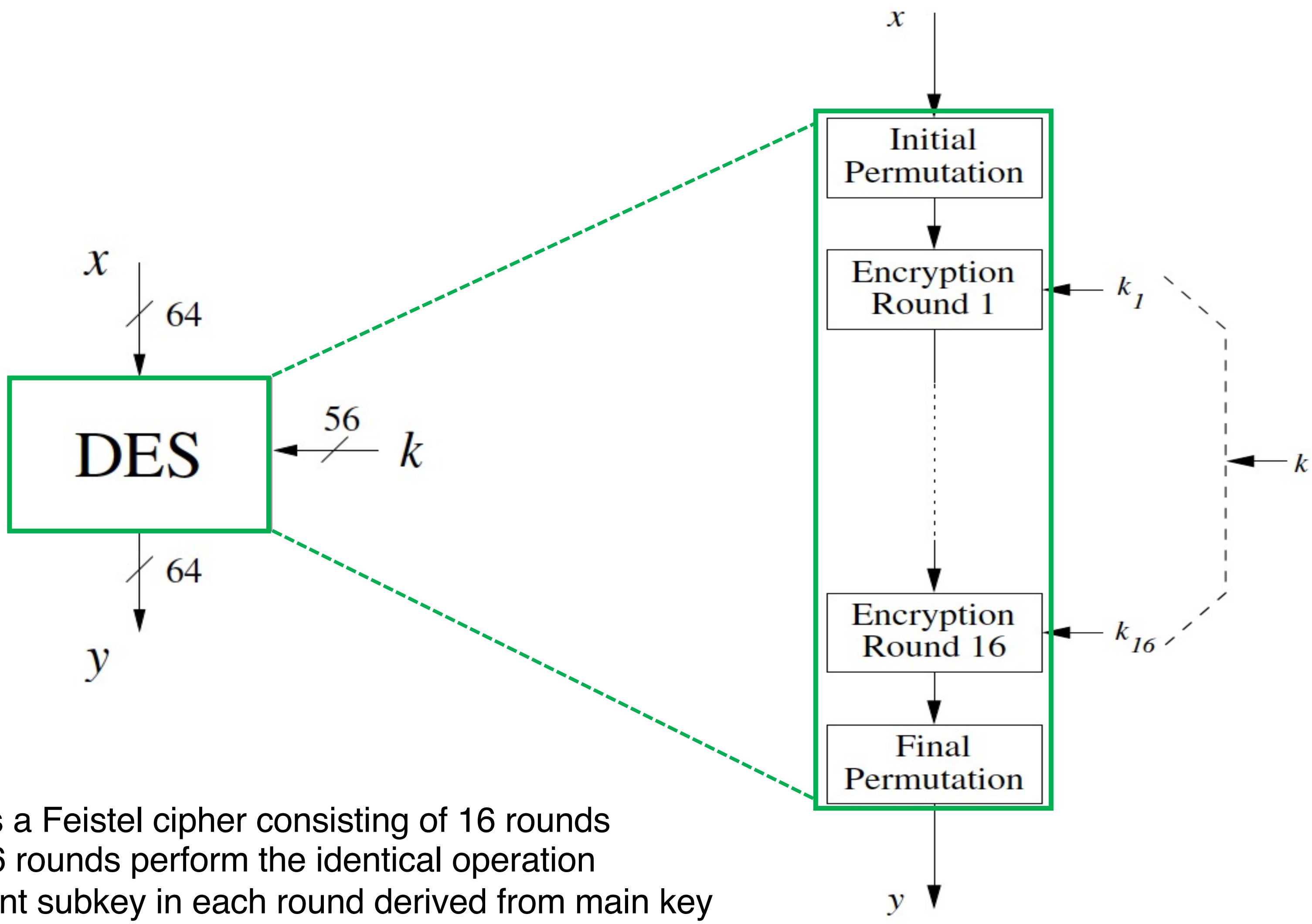
DES History

- In 1972, the *NBS* initiated a request for proposals for a standardized cipher in the USA.
  - US *National Bureau of Standards (NBS)* is now called the *National Institute of Standards and Technology (NIST)*
  - The goal was to find a single cryptographic algorithm which could be used for a variety of applications.
- In 1974, the NBS received the most promising candidate from a team of cryptographers working at IBM, called *Lucifer*.
- In order to investigate the security of the submitted ciphers, the NBS requested the help of the *National Security Agency (NSA) that* influenced changes to the IBM cipher which was rechristened DES.
- In 1977, the NBS finally released all specifications of the modified IBM cipher as the *Data Encryption Standard* to the public.
  - DES is specifically designed to withstand differential cryptanalysis, an attack not known to the public until 1990.
  - Allegedly, the NSA convinced IBM to reduce the Lucifer key length of 128 bit to 56 bit, which made the cipher much more vulnerable to brute-force attacks.

Overview of DES

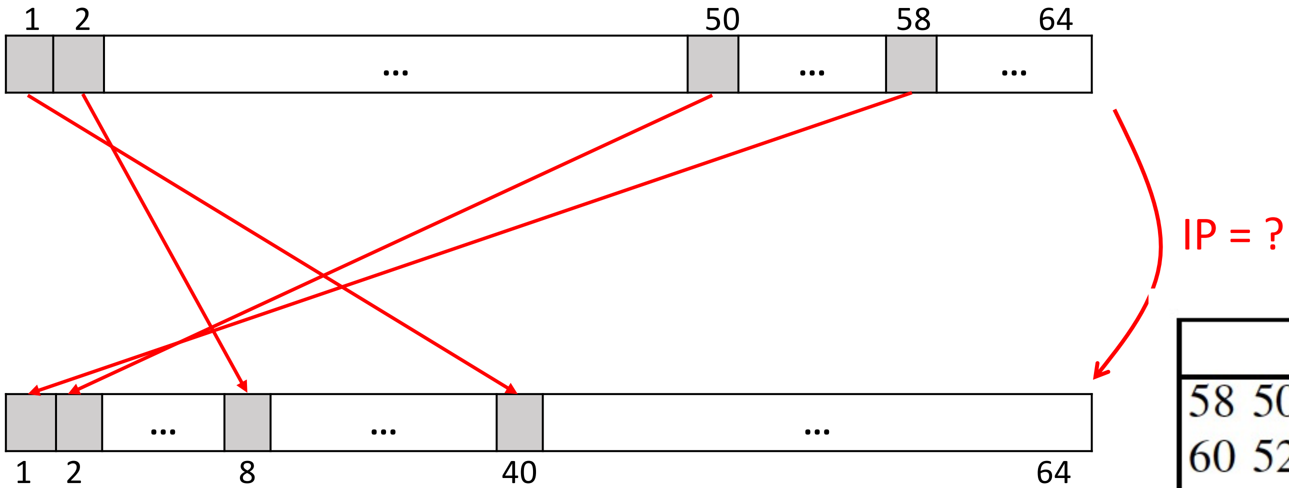
- Data Encryption Standard
  - The first standard cipher
  - Developed in the early 1970s at IBM based on Horst Feister (Lucifer)
- A symmetric cipher
  - the same key is used for encryption and decryption
- Encrypts blocks of length of 64 bits with a key of 56 bits .





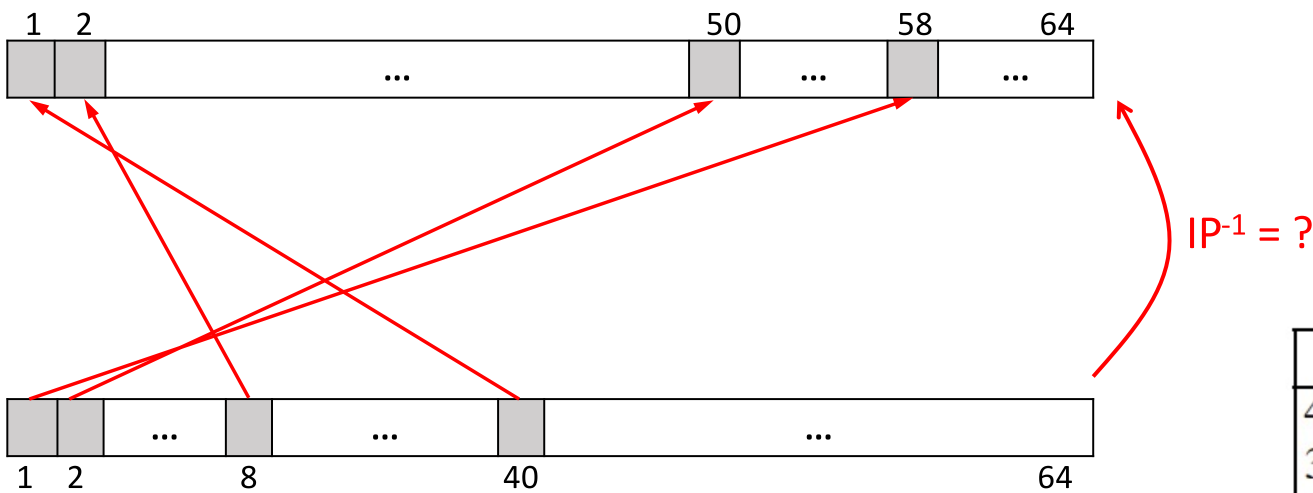
- DES is a Feistel cipher consisting of 16 rounds
- ALL 16 rounds perform the identical operation
- Different subkey in each round derived from main key

IP: 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

IP^{-1}: Final Permutation

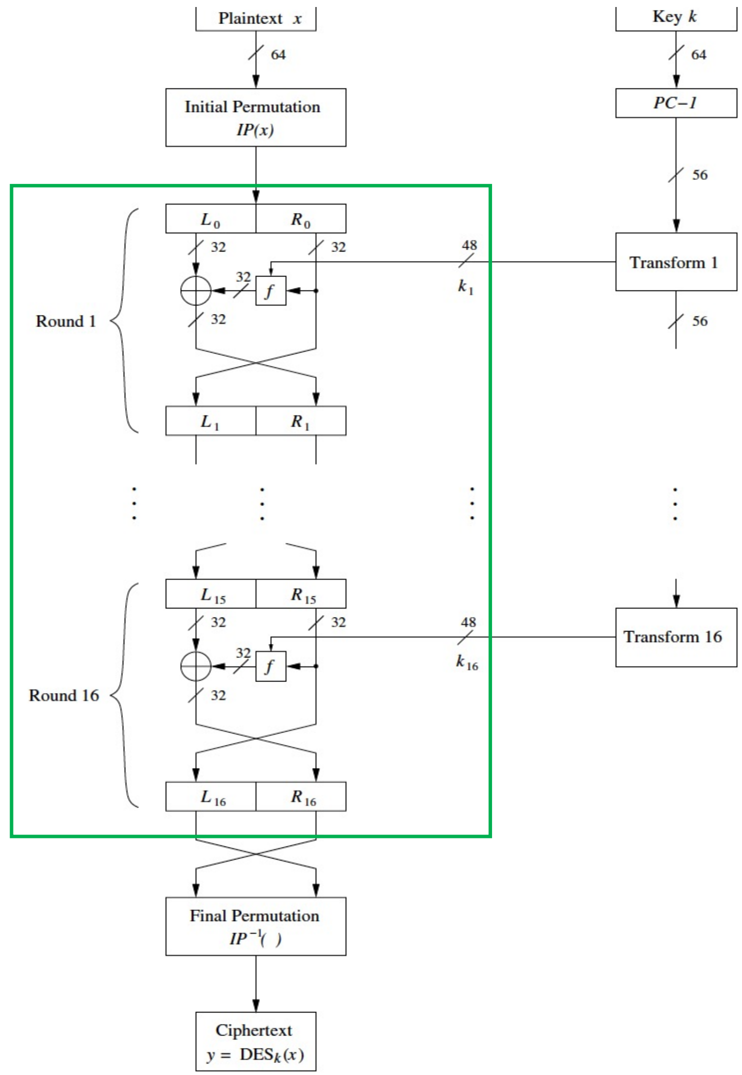


IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

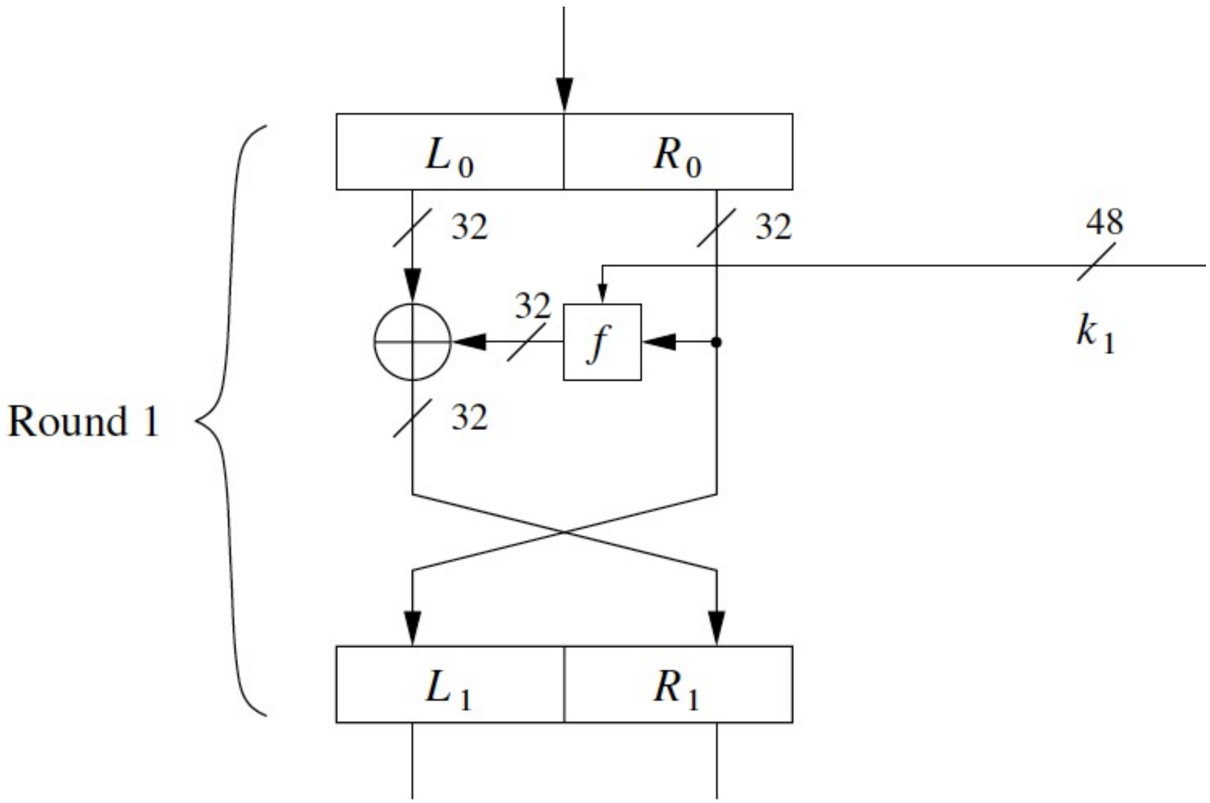
The DES Feistel Network

DES uses a Feistel structure.

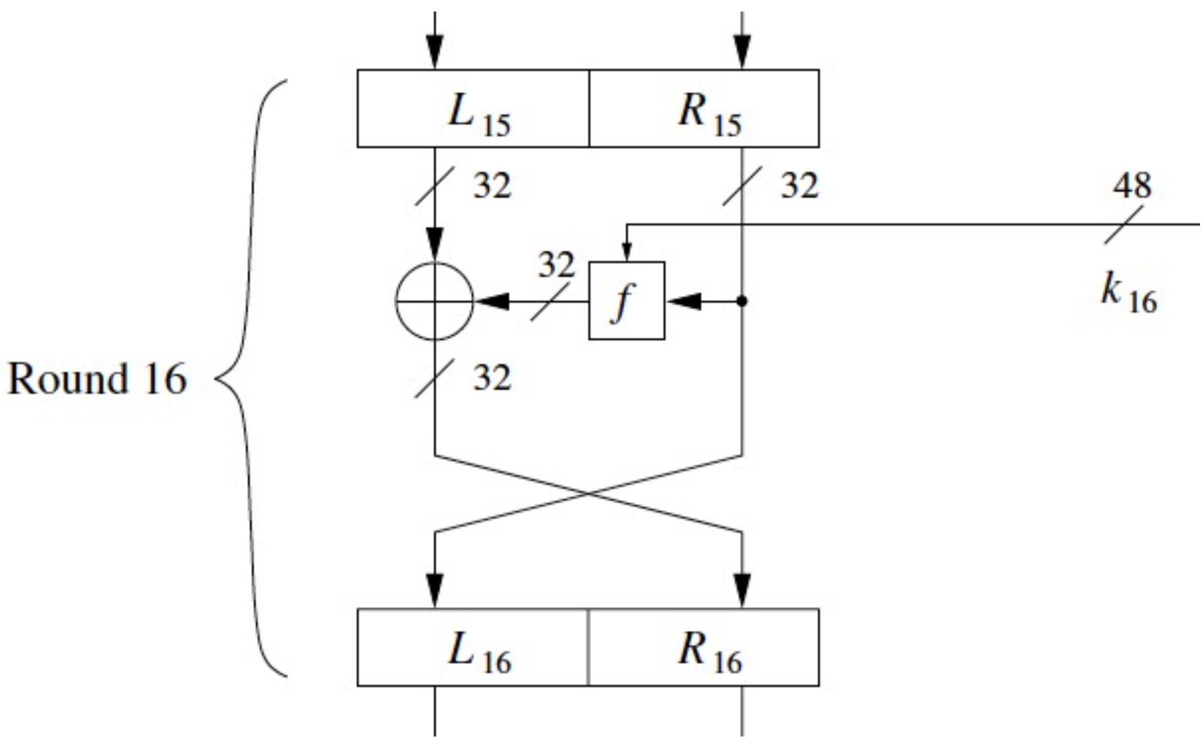
Advantage:
Encryption and decryption differ only in key schedule



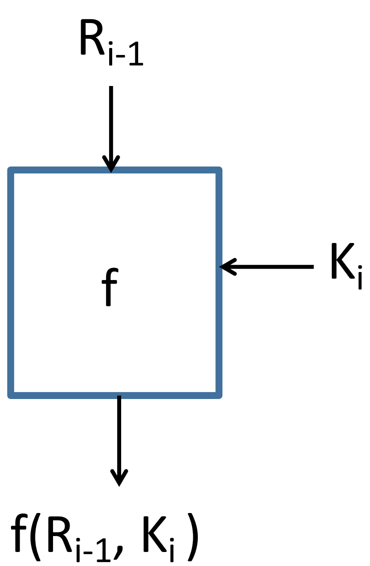
Round 1



Round 16



The F-function



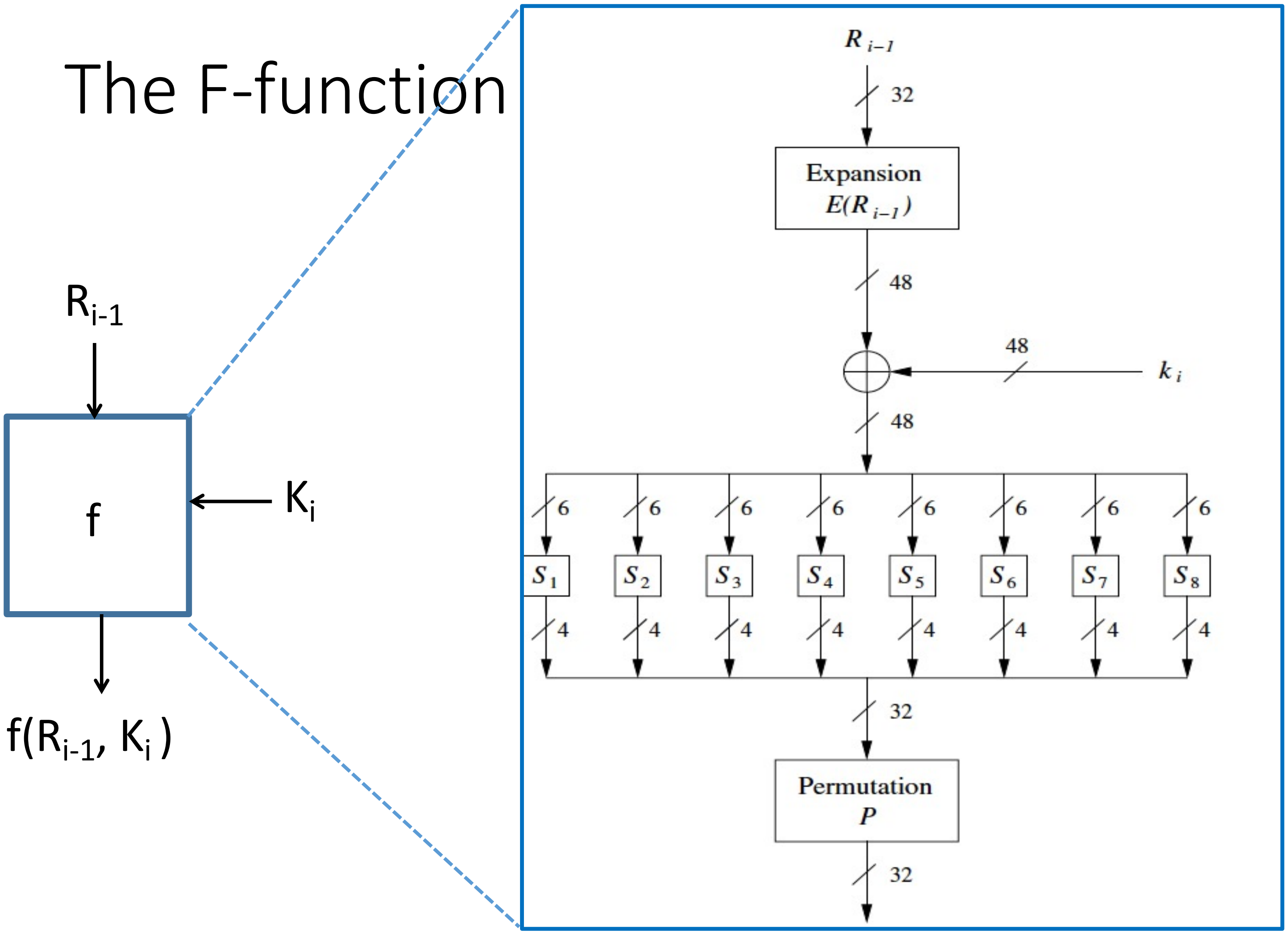
Each round **i** uses the same **f** function.

input:

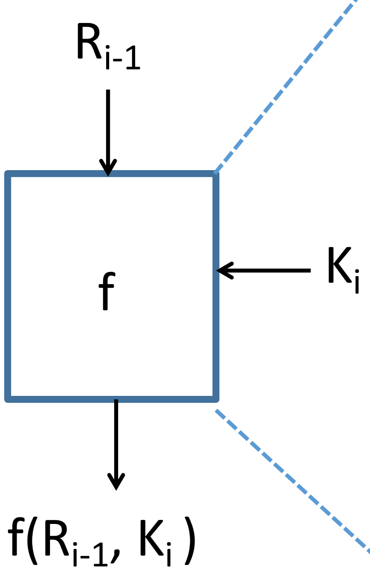
R_{i-1} right half of the output of the previous round

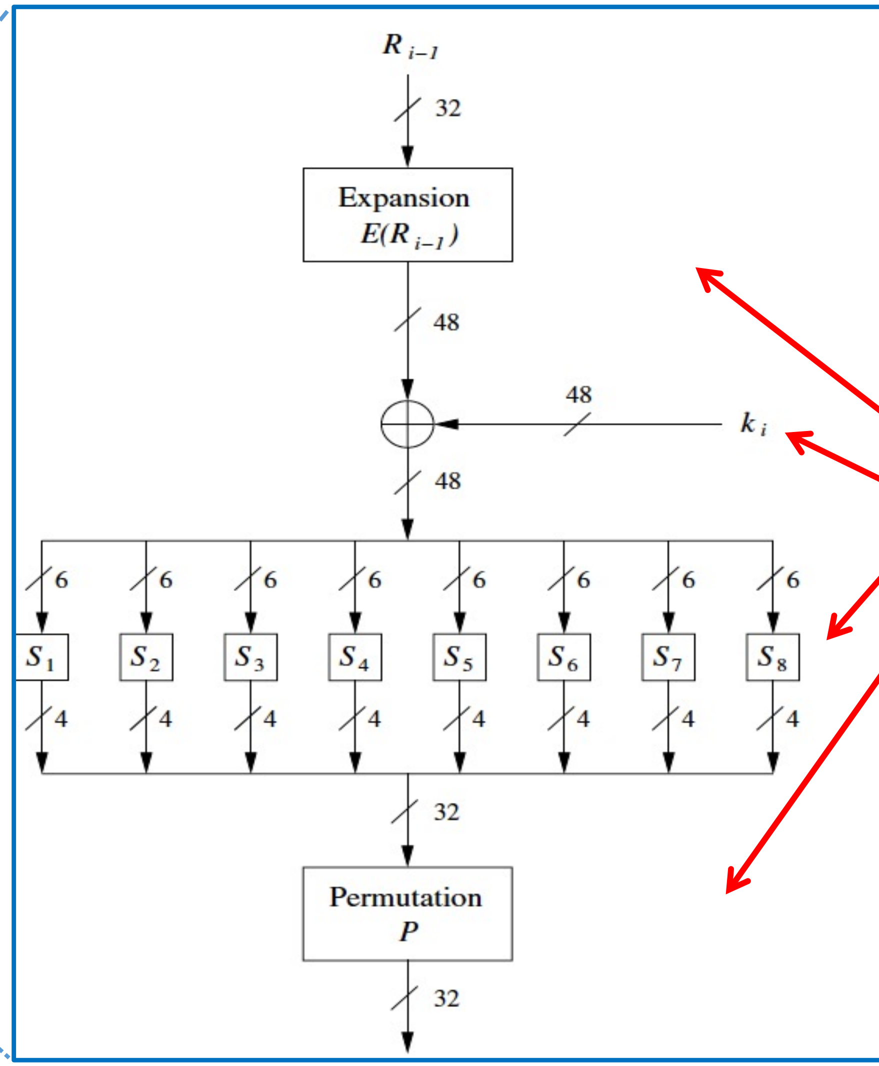
k_i current round-key

The F-function



The F-function



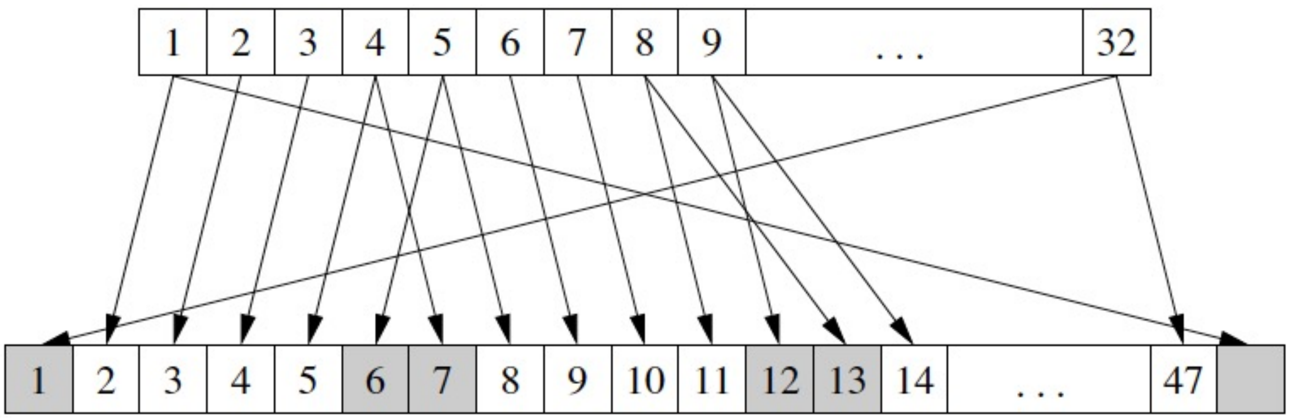


4 steps:

1. Expansion
2. XOR with k_i
3. S-box substitution
4. Permutation P

F-function: 1. Expansion

- First, the 32-bit input is expanded to 48 bits
  - by partitioning the input into eight 4-bit blocks
  - and by expanding each block to 6 bits.



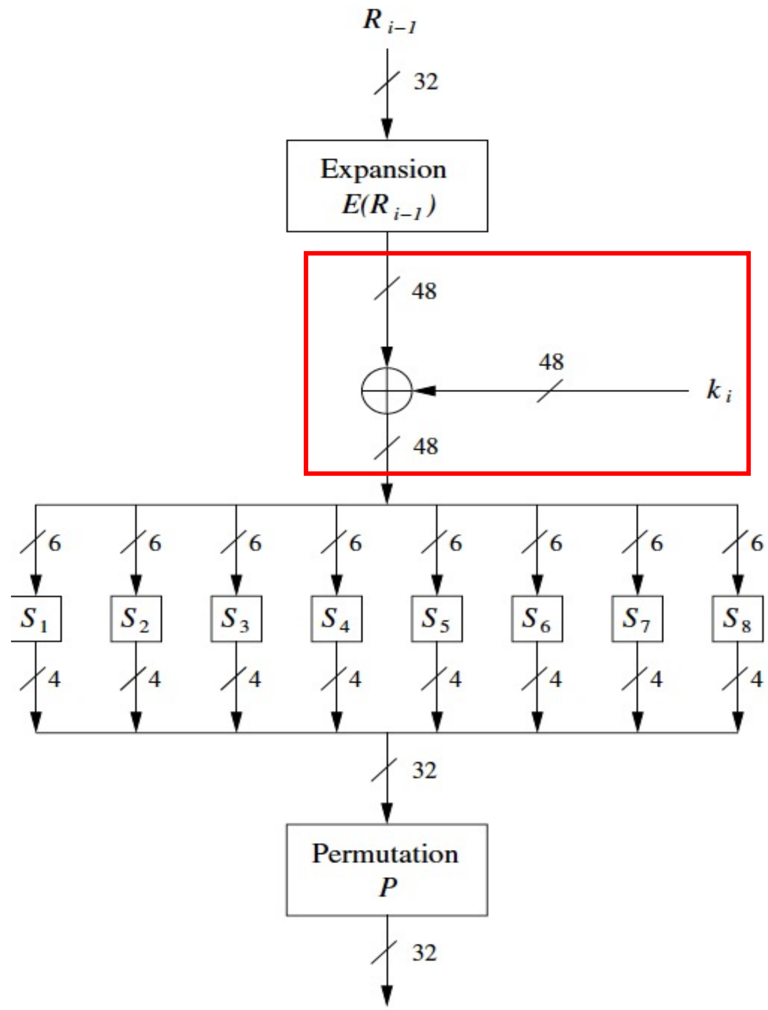
E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

F-function: 2. XOR

- Next, the 48-bit result of the expansion is XORed with the round key k_i

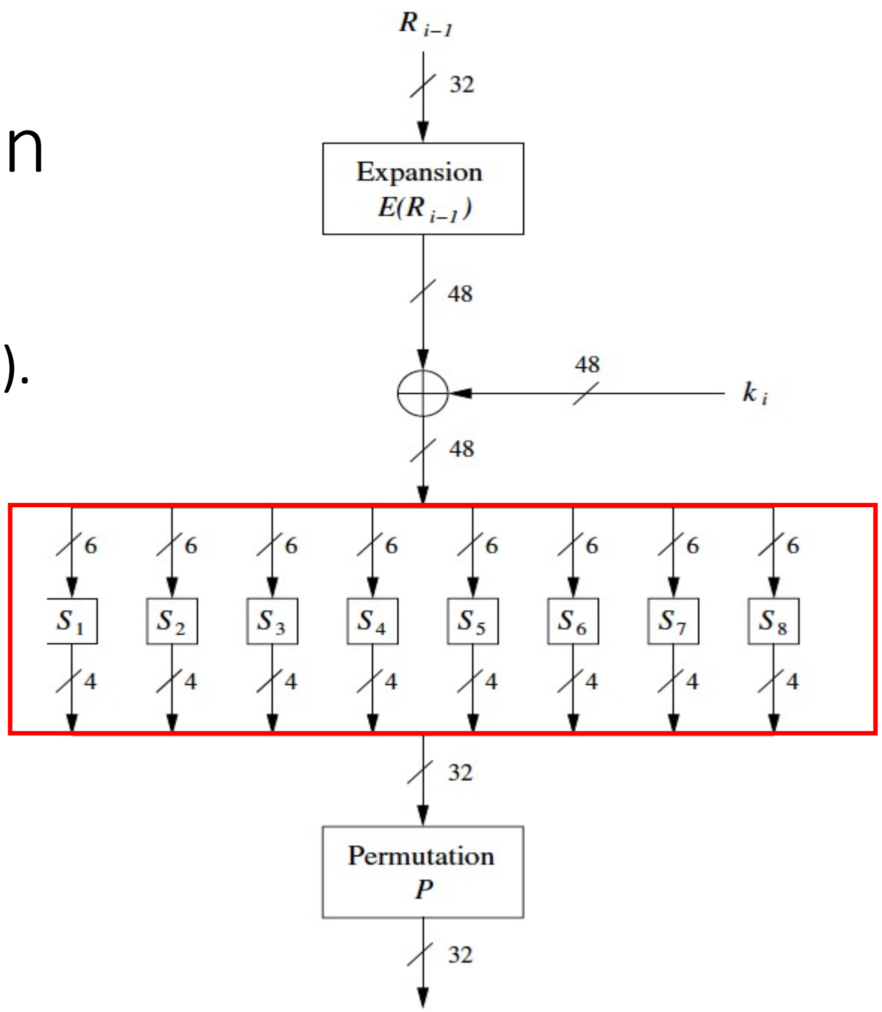
$$E(R_{i-1}) \oplus k_i$$

which is fed into the S-boxes.



F-function: 3. Substitution

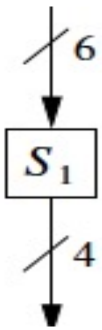
- The eight 6-bit blocks are fed into eight different substitution boxes (S-box).



F-function: 3. Substitution

- Each S-box is a lookup table that maps a 6-bit input to a 4-bit output.

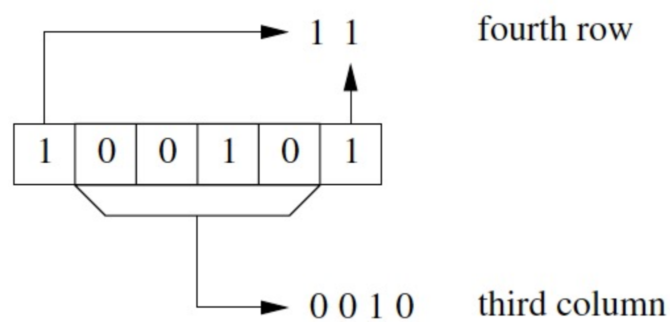
- S-box



S_1 box

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

How to read the table?



F-function: 3. Substitution

- There are 8 different S-boxes used in DES:

  $S_1, S_2, \ldots, S_8$

S_2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	01	08	14	06	11	03	04	09	07	02	13	12	00	05	10
1	03	13	04	07	15	02	08	14	12	00	01	10	06	09	11	05
2	00	14	07	11	10	04	13	01	05	08	12	06	09	03	02	15
3	13	08	10	01	03	15	04	02	11	06	07	12	00	05	14	09

S_3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	00	09	14	06	03	15	05	01	13	12	07	11	04	02	08
1	13	07	00	09	03	04	06	10	02	08	05	14	12	11	15	01
2	13	06	04	09	08	15	03	00	11	01	02	12	05	10	14	07
3	01	10	13	00	06	09	08	07	04	15	14	03	11	05	02	12

S_4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	07	13	14	03	00	06	09	10	01	02	08	05	11	12	04	15
1	13	08	11	05	06	15	00	03	04	07	02	12	01	10	14	09
2	10	06	09	00	12	11	07	13	15	01	03	14	05	02	08	04
3	03	15	00	06	10	01	13	08	09	04	05	11	12	07	02	14

S_5	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	02	12	04	01	07	10	11	06	08	05	03	15	13	00	14	09
1	14	11	02	12	04	07	13	01	05	00	15	10	03	09	08	06
2	04	02	01	11	10	13	07	08	15	09	12	05	06	03	00	14
3	11	08	12	07	01	14	02	13	06	15	00	09	10	04	05	03

S_6	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	01	10	15	09	02	06	08	00	13	03	04	14	07	05	11
1	10	15	04	02	07	12	09	05	06	01	13	14	00	11	03	08
2	09	14	15	05	02	08	12	03	07	00	04	10	01	13	11	06
3	04	03	02	12	09	05	15	10	11	14	01	07	06	00	08	13

S_7	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	04	11	02	14	15	00	08	13	03	12	09	07	05	10	06	01
1	13	00	11	07	04	09	01	10	14	03	05	12	02	15	08	06
2	01	04	11	13	12	03	07	14	10	15	06	08	00	05	09	02
3	06	11	13	08	01	04	10	07	09	05	00	15	14	02	03	12

S_8	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	02	08	04	06	15	11	01	10	09	03	14	05	00	12	07
1	01	15	13	08	10	03	07	04	12	05	06	11	00	14	09	02
2	07	11	04	01	09	12	14	02	00	06	10	13	15	03	05	08
3	02	01	14	07	04	10	08	13	15	12	09	00	03	05	06	11

Each small lookup table (4 by 16 = 64 bytes) is close to the max size that would fit on a single integrated chip in 1970s.

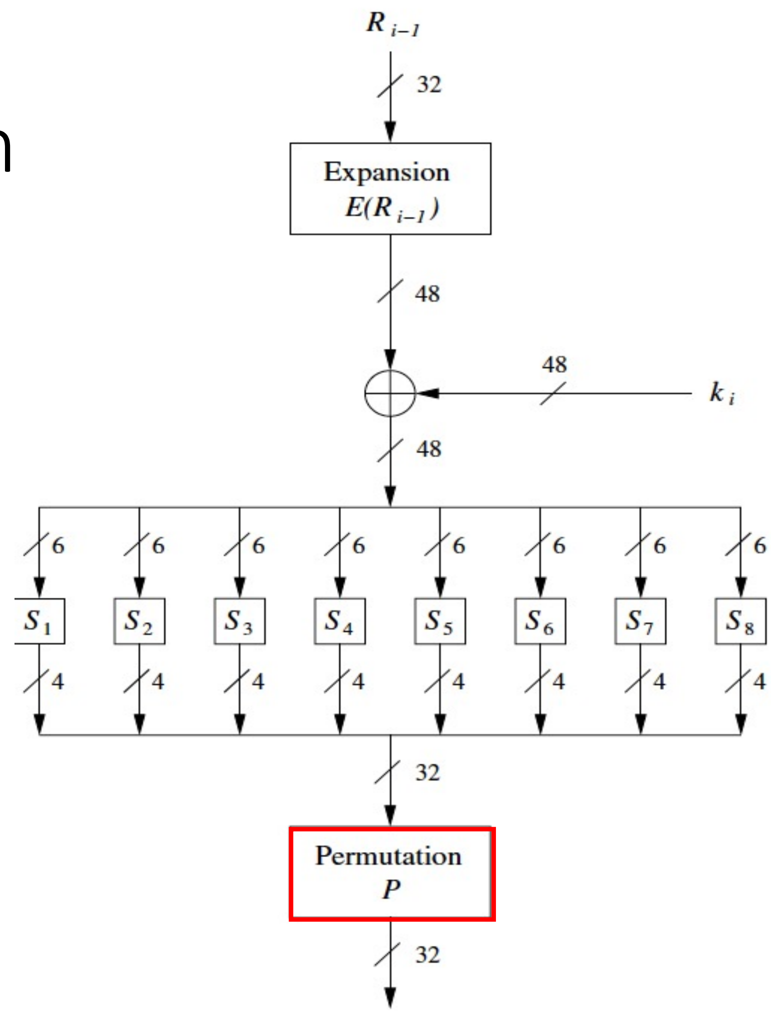
Design Criteria of S-boxes

1. Each S-box has six input bits and four output bits.
2. No single output bit should be too close to a linear combination of the input bits.
3. If the lowest and the highest bits of the input are fixed and the four middle bits are varied, each of the possible 4-bit output values must occur exactly once.
4. If two inputs to an S-box differ in exactly one bit, their outputs must differ in at least two bits.
5. If two inputs to an S-box differ in the two middle bits, their outputs must differ in at least two bits.
6. If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must be different.
7. ...

F-function: 4. Permutation

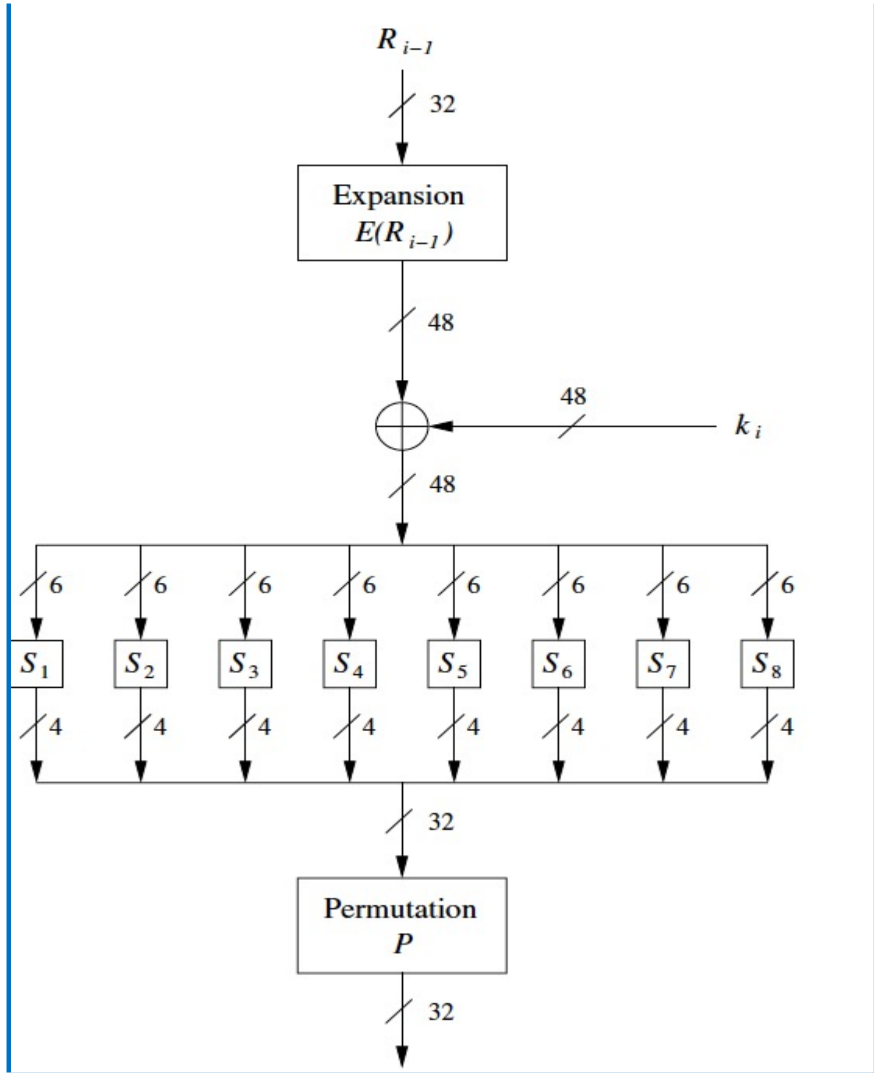
- Finally, the 32-bit output is permuted bitwise according to the P permutation:

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



Avalanche Effect

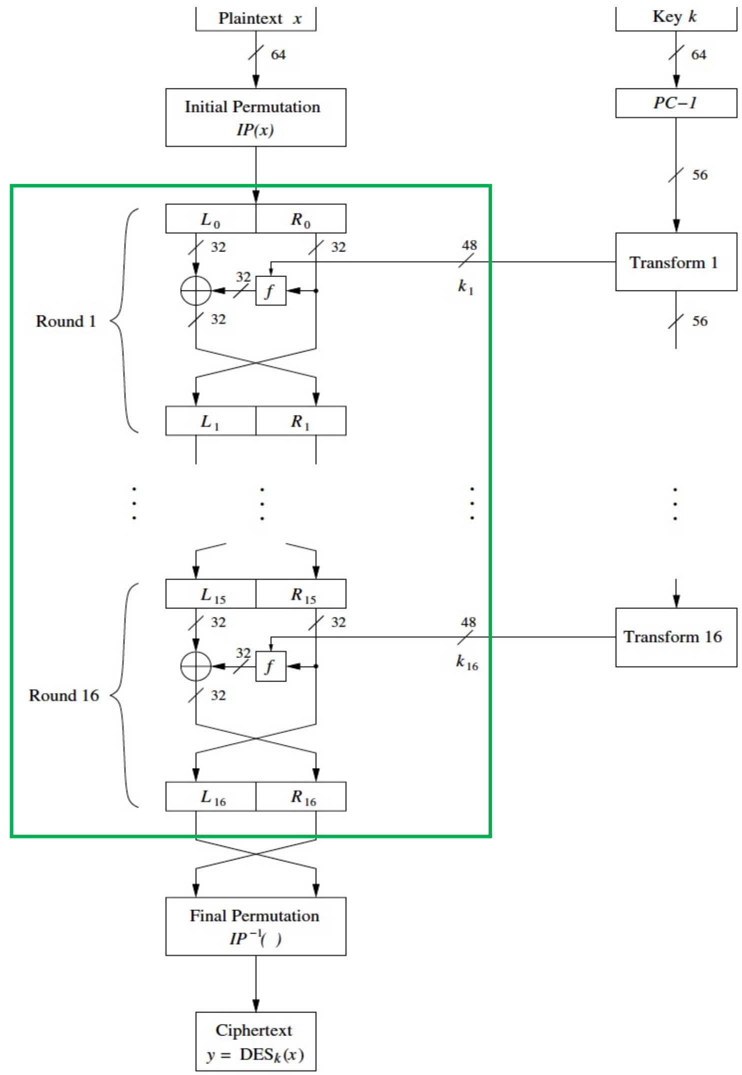
- The diffusion caused by the expansion, S-boxes and the permutation *P* guarantees that every bit at the end of the **fifth** round is a function of every plaintext bit and every key bit.

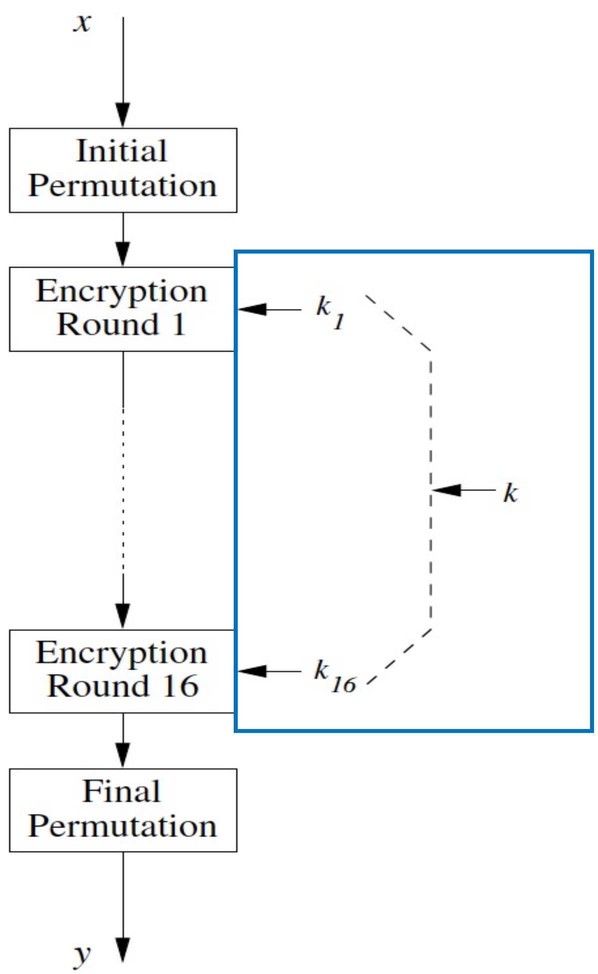


DES review

DES uses a Feistel structure.

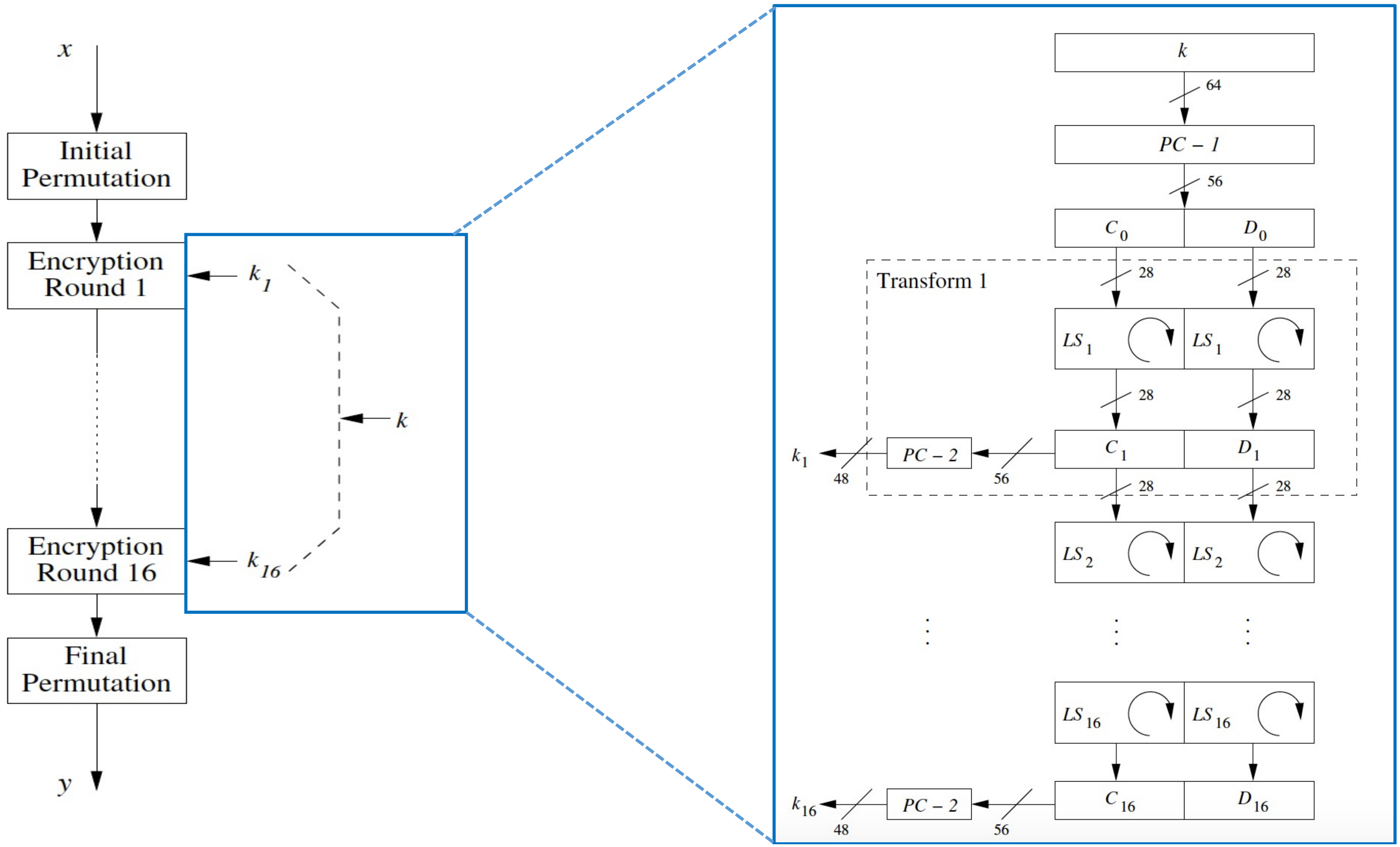
Advantage:
Encryption and decryption differ only in key schedule





Key Schedule

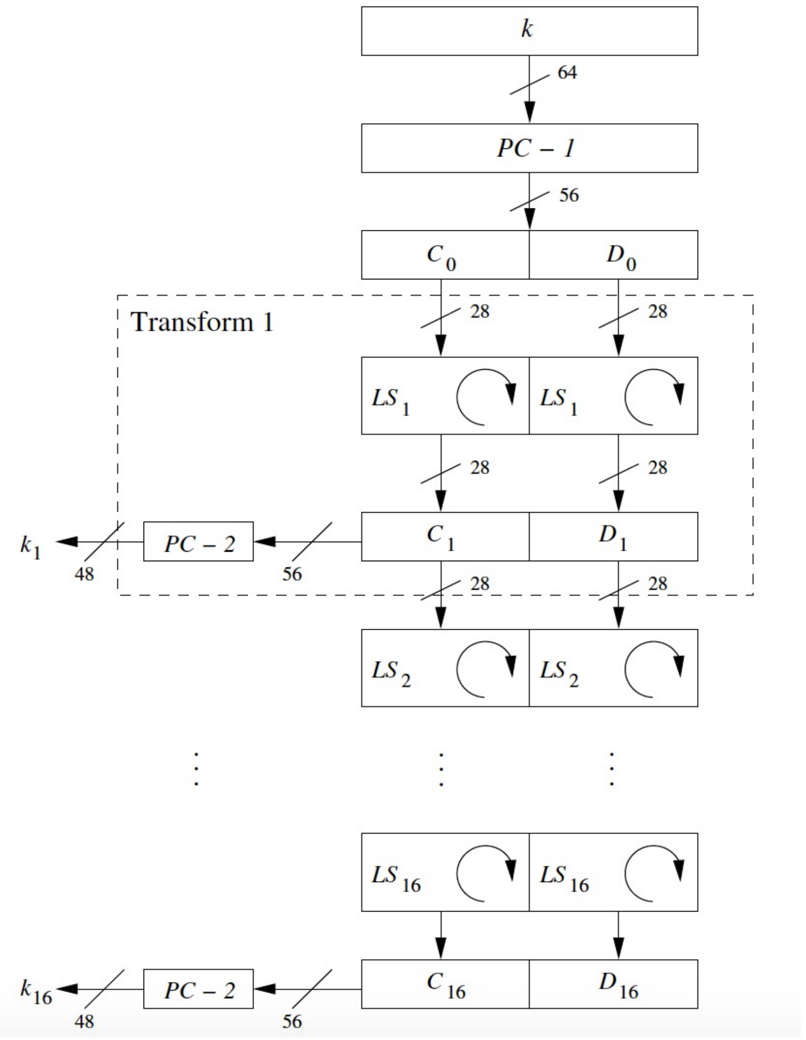
- Key scheduling algorithm derives 16 round keys k_i from the original key k
- k is 56-bits main key
- Each k_i is 48 bits round key (subkey)



x
Initial Permutation
Encryption Round 1
k 1
k
Encryption Round 16
k 16
Final Permutation
y
k
64
PC – 1
56
C 0
D 0
Transform 1
28
LS 1
C 1
D 1
PC – 2
k 1
48
LS 2
LS 16
C 16
D 16
k 16

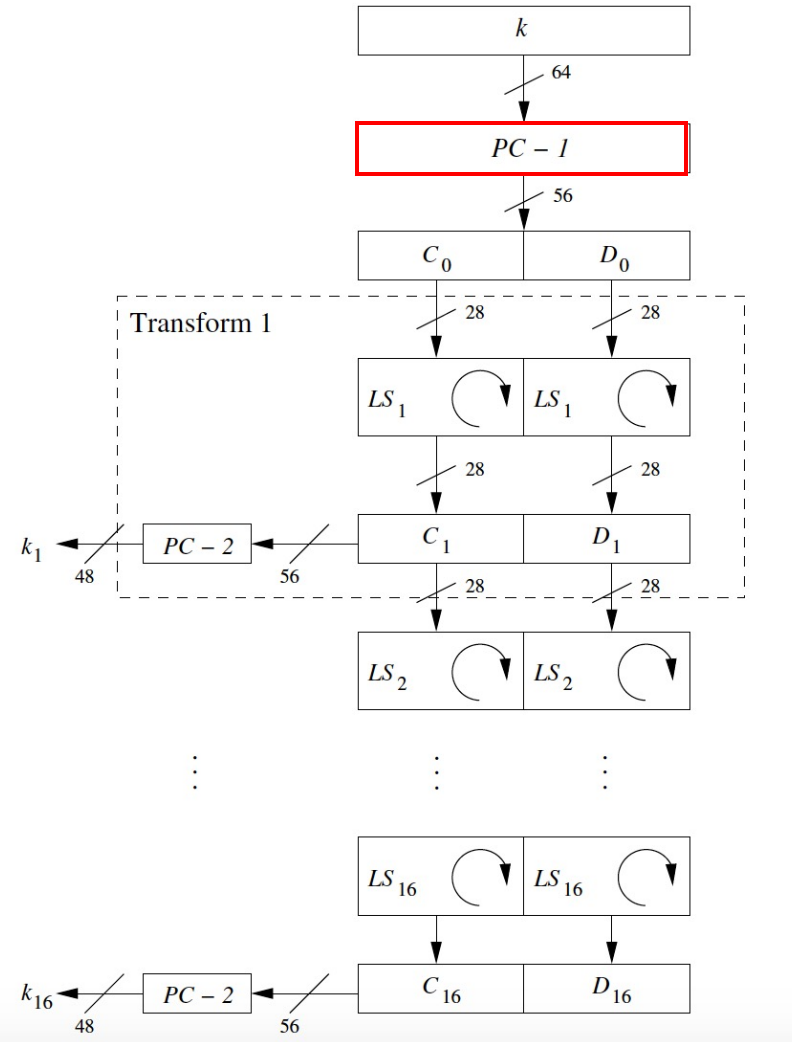
Key Schedule for Encryption

- Consists of 16 rounds
- Each round generate a round-key
- Operations used are
  - permutations
    - PC-1
    - PC-2
  - Left-Shift

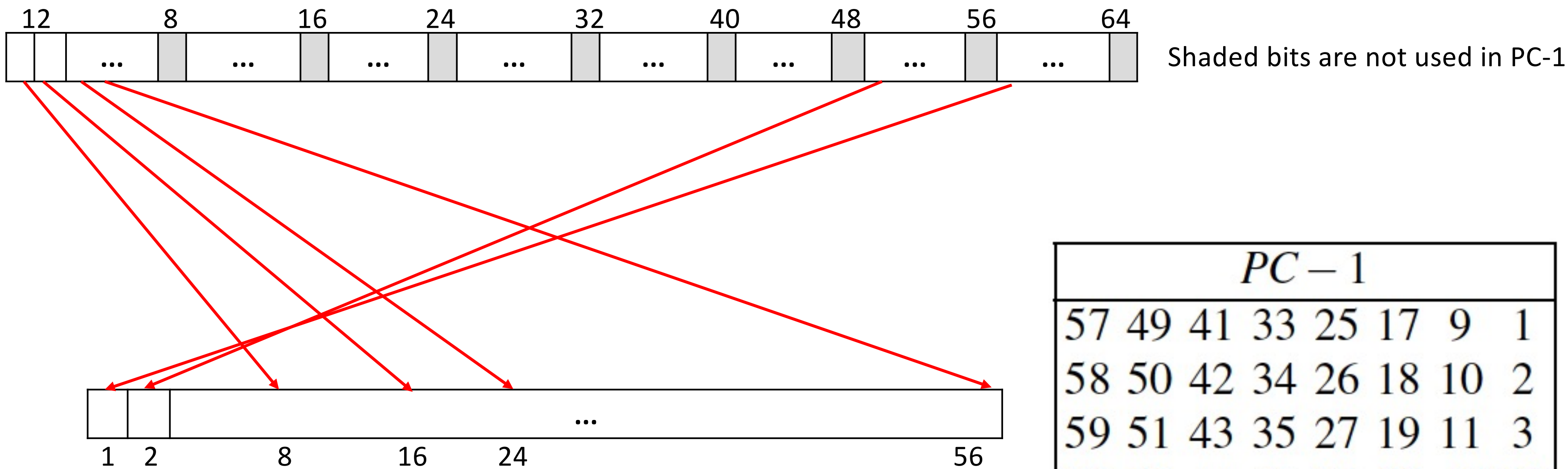


Key Schedule: PC-1

- **PC-1** stands for **permuted choice-1**
- Initial k is 64-bits, but every 8th bit is ignored.
  - Every eighth bit is used as an odd parity bit over the preceding seven bits
  - Odd parity is an extra bit added to a word used for error-checking in network communication. The extra bit ensures that the number of 1's in the word is odd.
- The eight parity bits are not actual key bits and do not increase the security.
- DES is a **56-bit cipher**, not a 64 bit one.



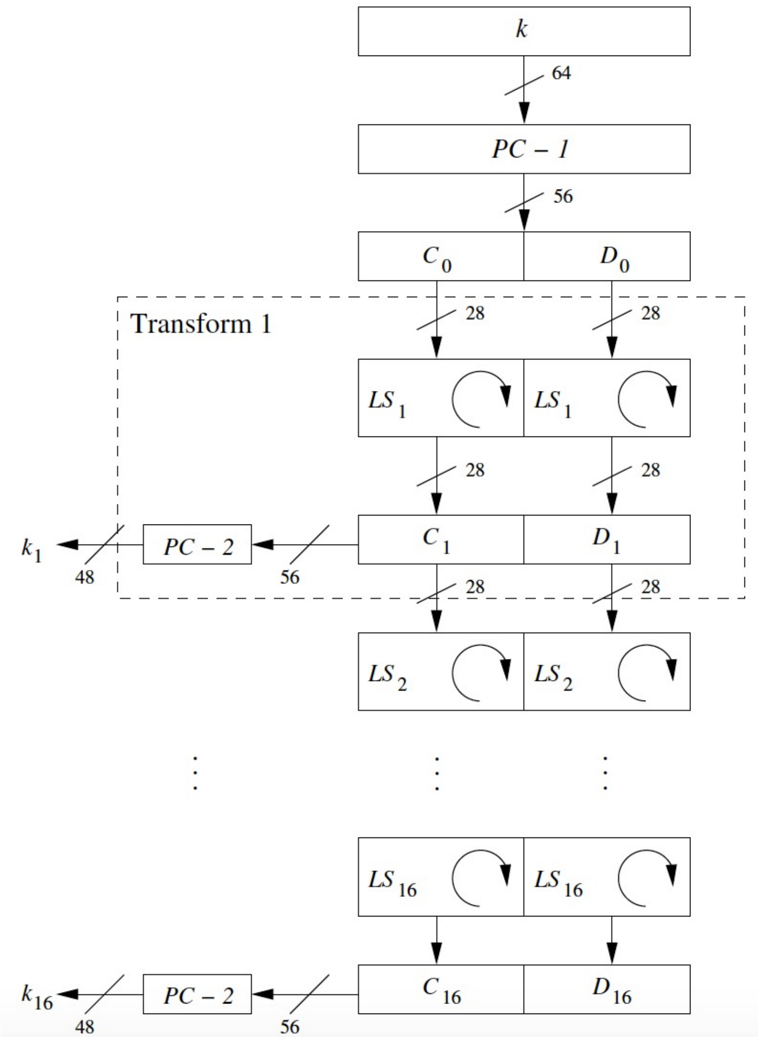
Key Schedule: 1. Initial key Permutation PC-1



$PC-1$							
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

Key Schedule: 2. Shift (Rotation)

- The resulting 56-bit key is split into two halves C_0 and D_0
- The two 28-bit halves are cyclically shifted, i.e., rotated, left by one or two bit positions:
  - In rounds ***i = 1, 2, 9, 16***, the two halves are rotated left by **one** bit.
  - In the other rounds ***where i != 1, 2, 9, 16***, the two halves are rotated left by **two** bits.
- The total number of rotation positions = 4*1+ 12*2 = 28
  - After round 16, the output bits are the same as the input bits right before round 1 starts. $C_{16} = C_0$ and $D_{16} = D_0$. (why?)

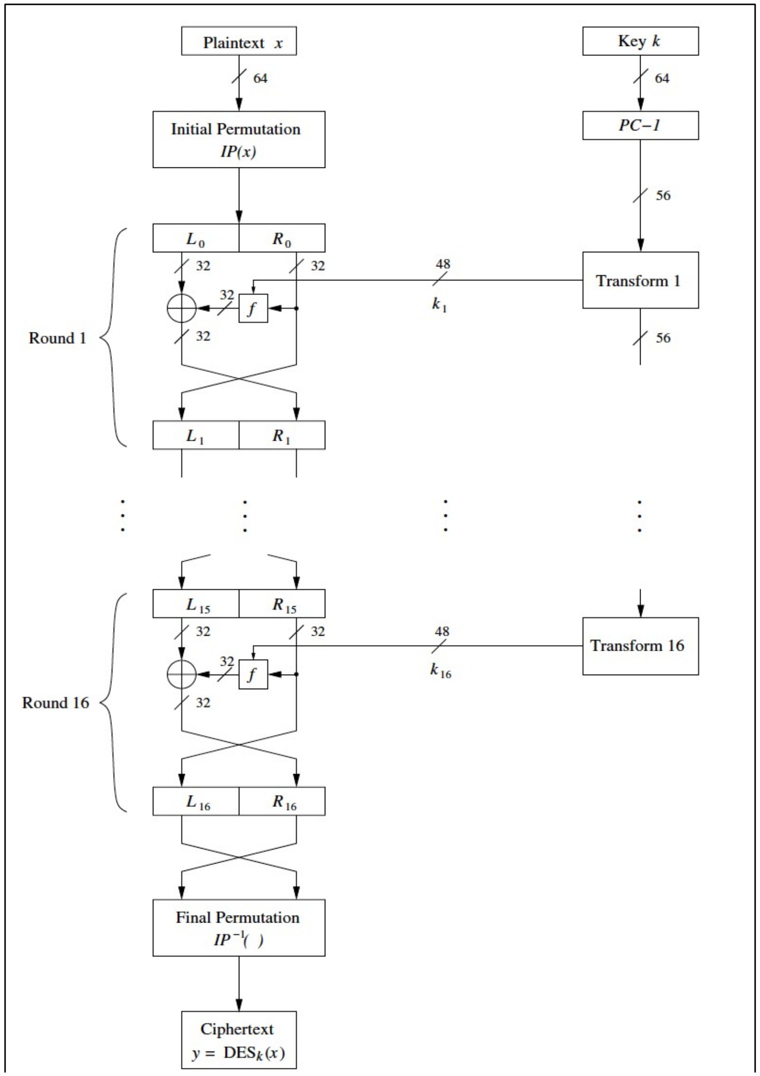


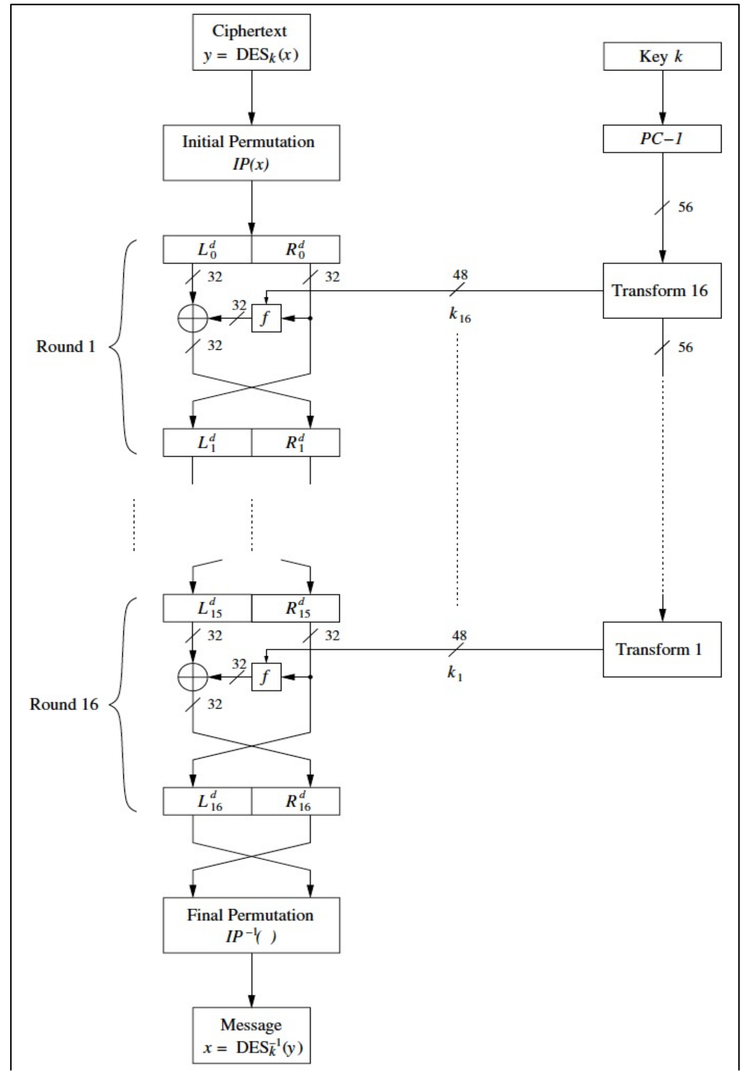
Key Schedule: 3. Permuted-Choice 2 (PC-2)

- In each round i, PC−2 permutes the 56 input bits coming from C_i and D_i and ignores 8 of them.

$PC-2$							
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

Encryption vs. Decryption

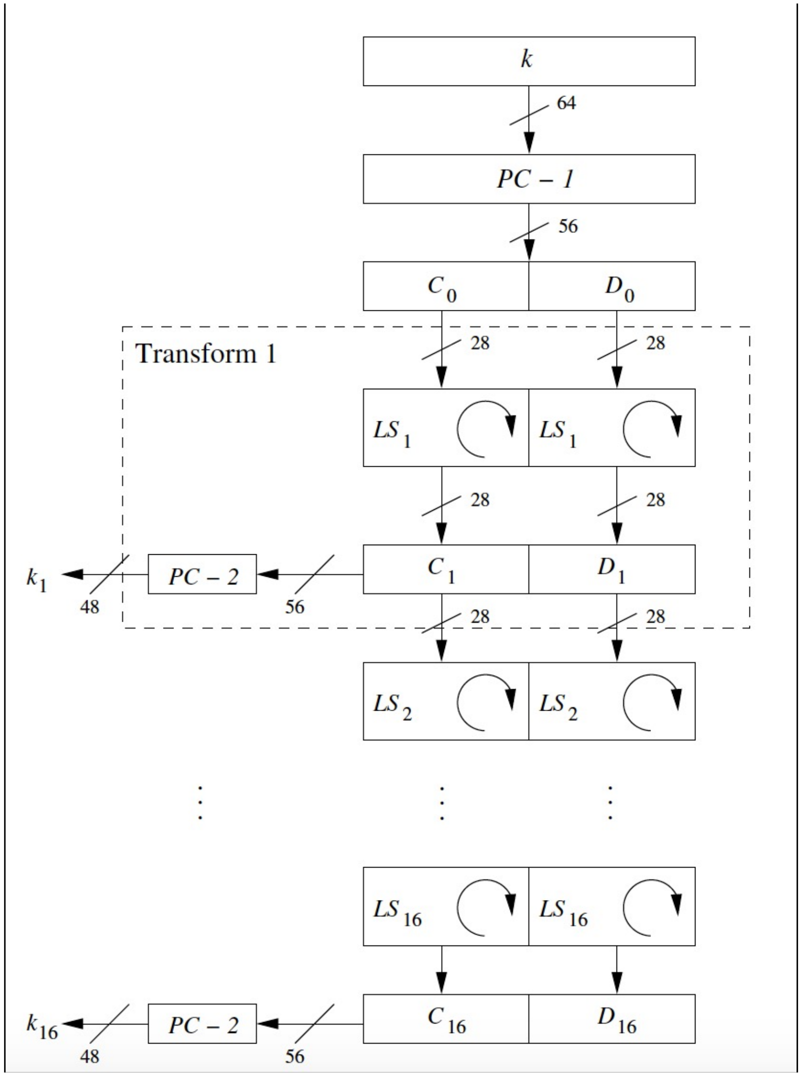




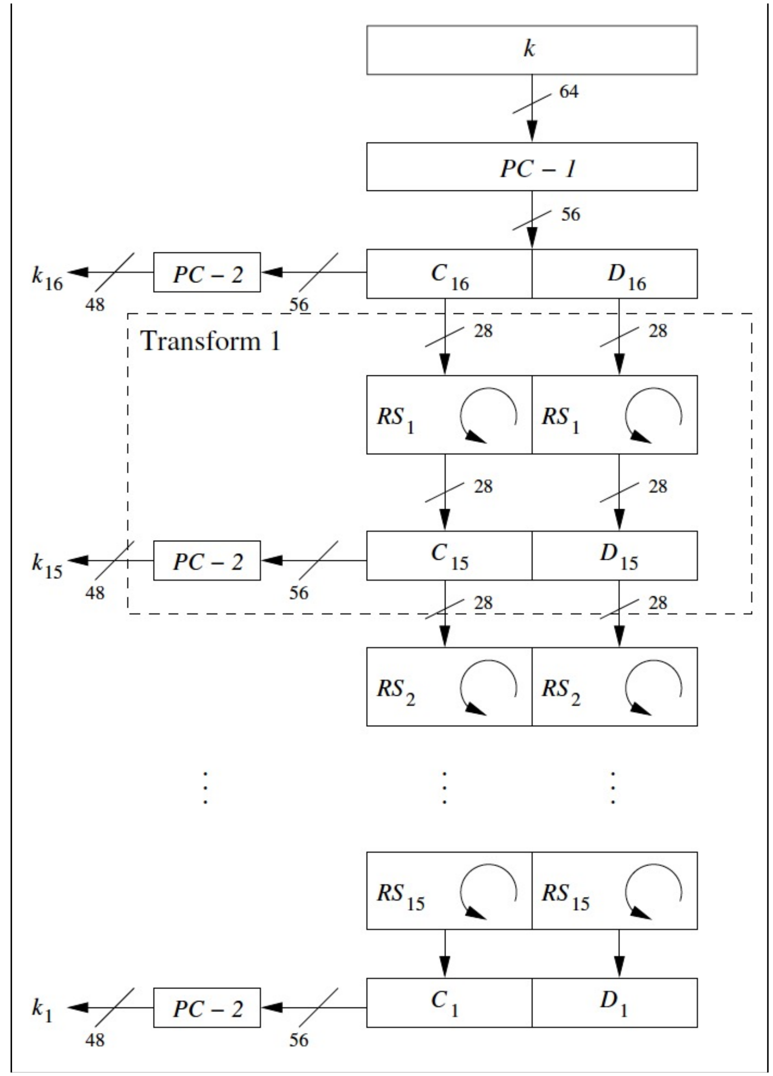
Why Decryption Works?

- In the last round of encryption,
  - $L_{16} = R_{15}$
  - $R_{16} = L_{15} + f(R_{15}, k_{16})$
- In the first round of decryption,
  - $L_0^d = R_{16}$
  - $R_0^d = L_{16}$
  - $L_1^d = R_0^d$
  - $R_1^d = L_0^d + f(R_0^d, k_{16})$
  - Therefore,
  - $L_1^d = R_{15}$
  - $R_1^d = L_{15} + \underbrace{f(R_{15}, k_{16}) + f(L_{16}, k_{16})}_{\text{cancelled out}} = L_{15}$

- Similarly, in the ith round of decryption,
  - $L_i^d = R_{16-i}$
  - $R_i^d = L_{16-i}$
- In particular, in the last round of decryption, we have
  - $L_{16}^d = R_0$
  - $R_{16}^d = L_0$
- Finally, (R_0, L_0) is swapped and IP^{-1} is applied.
- $IP^{-1}(L_0,R_0) = IP^{-1}(IP(x)) = x$



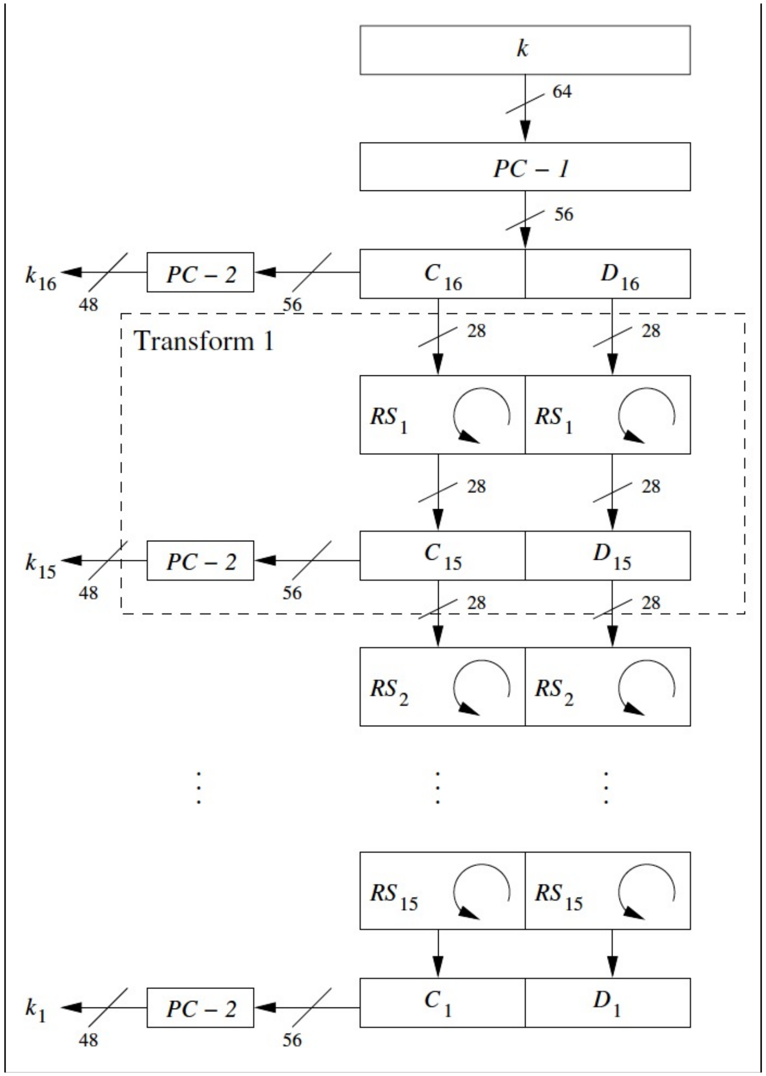
Key Schedule for Encryption and Decryption



Reversed key schedule for decryption of DES:

- PC-1
- PC-2
- RS (right shifting):
- The first subkey k_{16} is obtained without shifting.
- After that shift two halves by
- 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1 bits.
- E.g. (C_{15}, D_{15}) is obtained by right-shifting (C_{16}, D_{16}) by 1 position.

- In total, 27 bit positions are right-shifted



Security of DES

- After the proposal of DES two major criticisms arose:
  - The key space is too small.
    - The original cipher proposed by IBM had 128 bits.
  - The design criteria of the S-boxes was kept secret:
    - “Are there any hidden analytical attacks (backdoors), only known to DES designers?”
- Analytical Attacks: DES is highly resistant to all known cryptanalysis that have been published later than DES.
- Exhaustive key search
  - Input: at least one pair of plaintext–ciphertext (x,y)
  - Attack: Test all 2^{56} possible keys until the following condition is fulfilled:

$$DES^{-1}_{k_i}(y) = x, \quad i = 0, 1, \ldots, 2^{56} - 1.$$

  - Relatively easy given today's computer technology

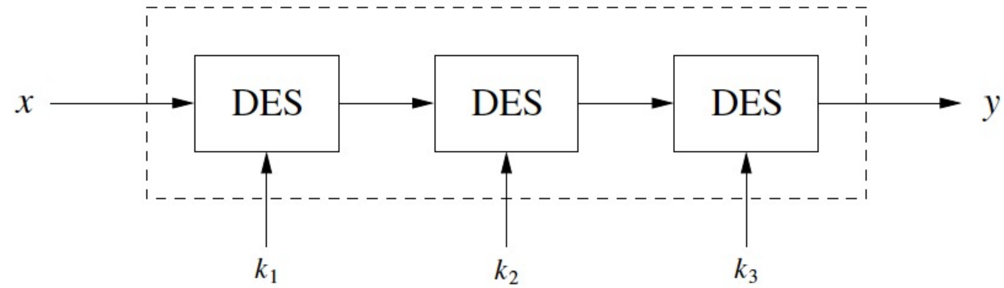
Cracking DES

- In 1977, Whitfield **Diffie** and Martin **Hellman** estimated that it was possible to build an exhaustive key search machine for approximately **$20,000,000**.
- In 1993, Michael **Wiene**r proposed the design of a very efficient key-search machine and he estimated the cost of his design at approximately **$1,000,000**, and the time required to find the key at **1.5 days**.
- In 1998, the EFF (Electronic Frontier Foundation) built the hardware machine **Deep Crack**, which performed a brute-force attack against DES in **56 hours.**
    - The average search time of Deep Crack was **15 days**, and the machine was built for less than **$250,000**.
    - The successful break with Deep Crack was considered the official demonstration that DES is no longer secure against determined attacks by many people.

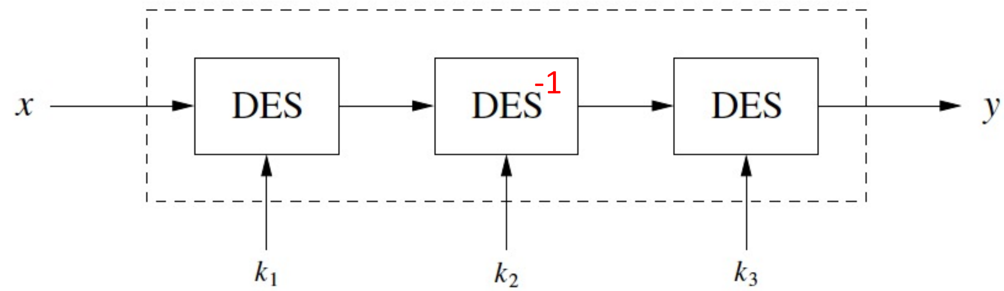
Deep Crack

DES Alternatives:

Option #1. 3DES



Option #2: A variant of 3DES:



Option #3: Key whitening - add keys k_1 and k_2 before and after encryption of DES.

Disadvantages of DES and DES Alternatives:

- Not efficient for software implementation
  - Its design is outdated by today's standards
- block size 64 bits – too short in certain applications
- Therefore, DES has been replaced by AES.