IN410
Cryptography and Secure Communications

Ahmad Fadlallah

Credits

- The following slides are the adaptation of
  - Lecture slides of Lawrie Brown based on William Stallings book “Cryptography and Network Security: Principles and Practice”
  - Lecture Slides of Dan Boneh – Stanford University
  - Lecture slides of Pawel Wocjan – University of Central California
  - Lecture Slides of Ahmed Serhrouchni – Telecom ParisTech
  - Others (References in the note section)

Course Syllabus

- Context
- History/ Classical ciphers
- **Symmetric Encryption ←**
- Asymmetric Encryption
- Cryptographic hash functions
- Key Distribution
- Security Protocols

Symmetric Encryption

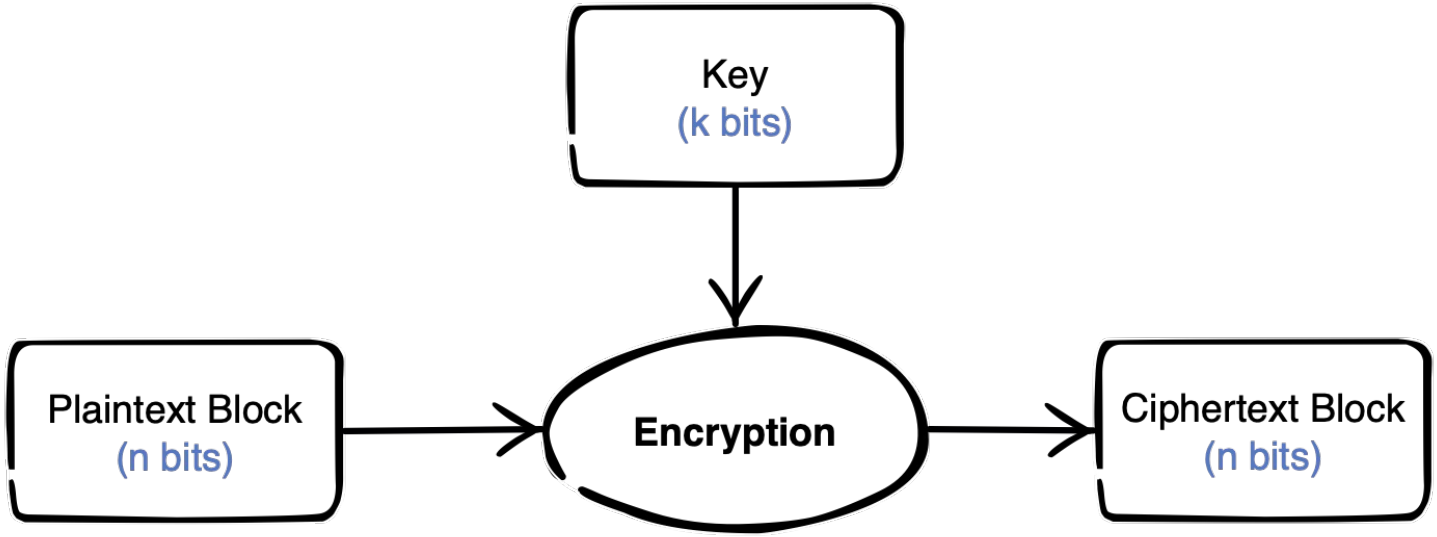
- **Block Ciphers ←**
- Operation Modes
- Stream Ciphers

Block Ciphers

Block Ciphers vs. Stream Ciphers

- **Block ciphers** process messages in blocks, each of which is then encrypted/decrypted
- Like a substitution on very big characters
  - 64-bits or more
- **Stream ciphers** process messages a bit or byte at a time when encrypting/decrypting
- Many current ciphers are block ciphers
  - Better analyzed
  - Broader range of applications
  - Possibility to use them as stream ciphers (operation modes)

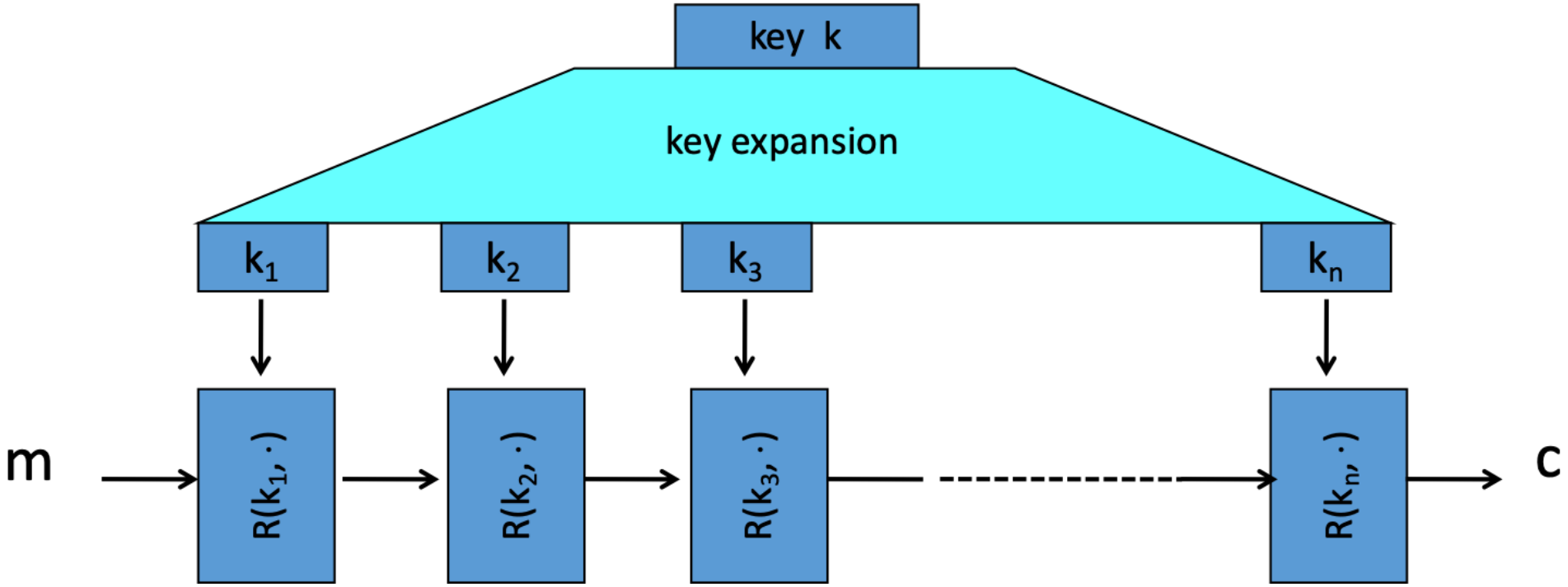
Block ciphers: Block diagram



- Examples:
  - DES: n=64 bits, k=56 bits
  - 3DES: n= 64 bits, k = 168 bits
  - AES: n=128 bits, k = 128, 192, 256 bits

Block Ciphers Built by Iteration

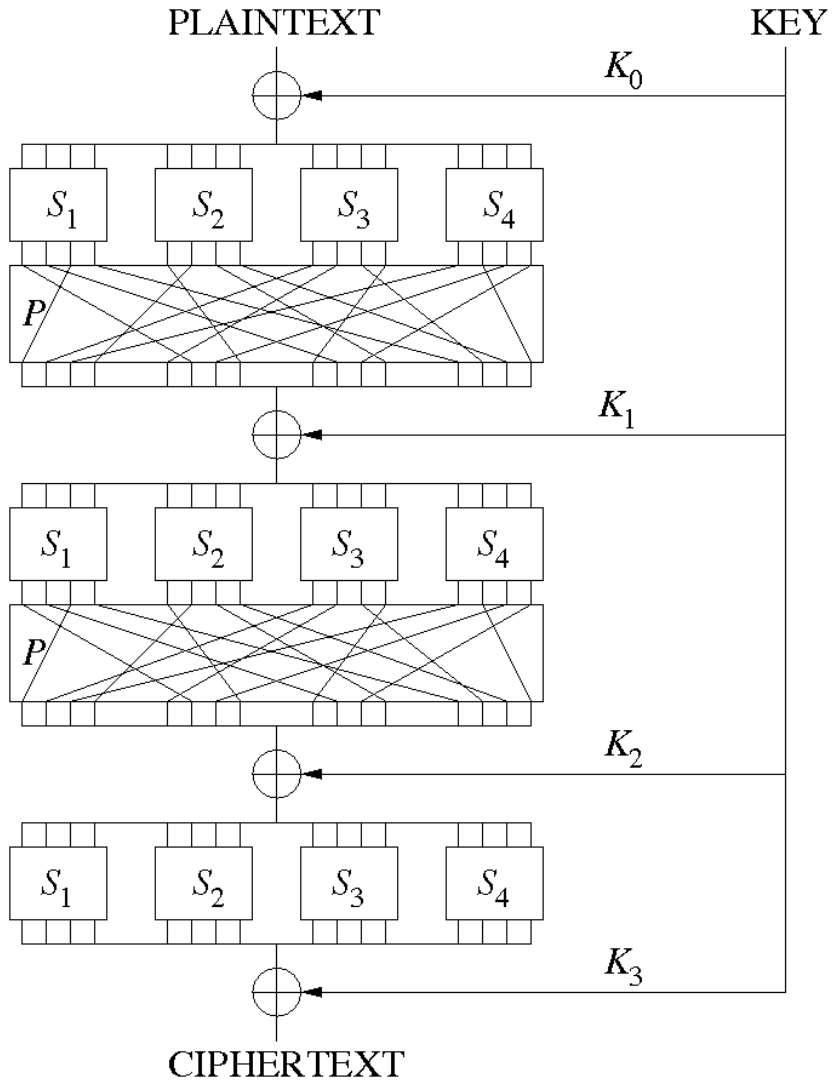
- $R(k_i, .)$ is called a round function
  - for 3DES (n=48), for AES-128 (n=10)



Shannon & Substitution-Permutation Ciphers

- Claude Shannon introduced the idea of Substitution-Permutation (S-P) networks in 1949
- Forms the basis of modern block ciphers
- S-P nets are based on the two primitive cryptographic operations
  - Substitution (S-box)
  - Permutation (P-box)
- Provide **Confusion** & **Diffusion** of message & key

Shannon & Substitution-Permutation Ciphers



Confusion and Diffusion

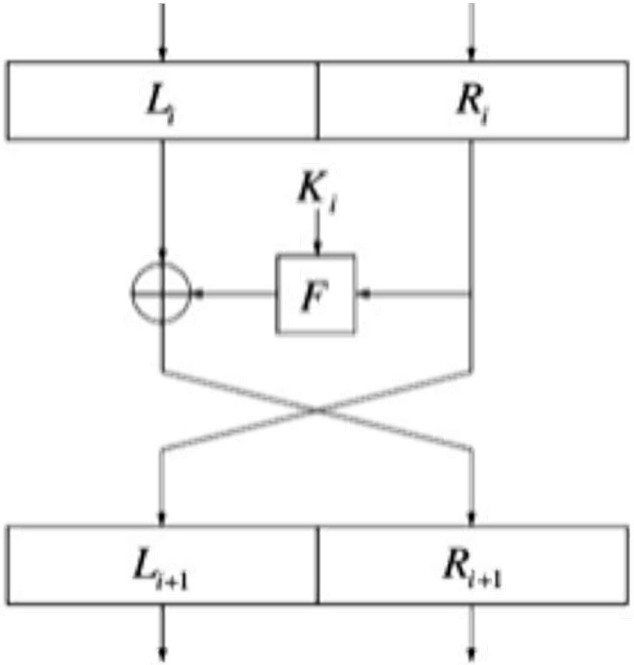
- Cipher needs to completely obscure statistical properties of original message
  - One-Time Pad (OTP) does this
- Shannon suggested combining S & P elements to obtain: **Confusion** and **Diffusion**
- **Confusion**: makes relationship between ciphertext and key as complex as possible
- **Diffusion**: dissipates statistical structure of plaintext over bulk of ciphertext

Confusion and Diffusion (cont'd)

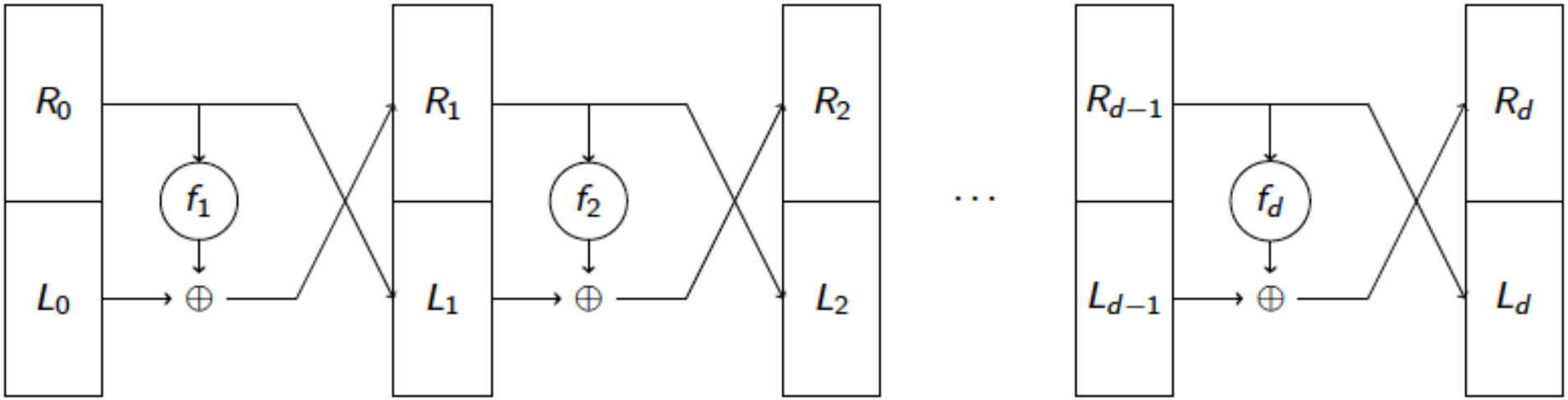
- **Confusion** => each binary digit (bit) of the ciphertext should depend on several parts of the key, obscuring the connections between the two.
- **Diffusion** => if we change a single bit of the plaintext, then (statistically) half of the bits in the ciphertext should change, and similarly, if we change one bit of the ciphertext, then approximately one half of the plaintext bits should change

Feistel Cipher Structure

- Horst Feistel devised the Feistel cipher
  - based on concept of invertible product cipher
- Input block split into two halves
  - Process through multiple rounds
    - Perform a substitution on left data half based on round function of right half & sub-key
    - then have permutation swapping halves
- Shannon's S-P net concept

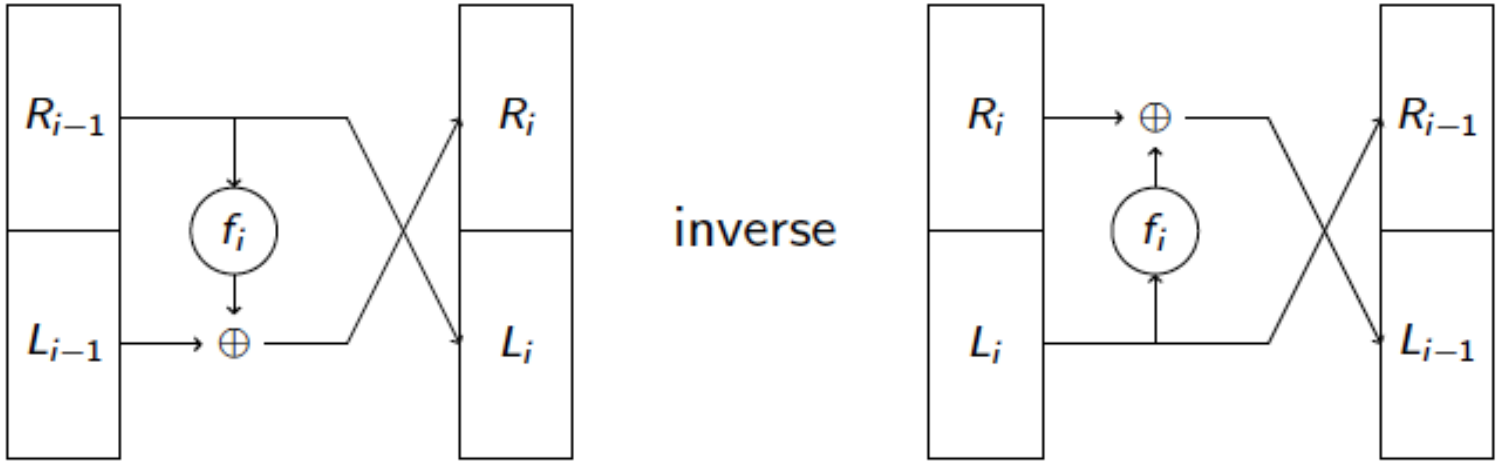


Feistel Cipher Structure



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

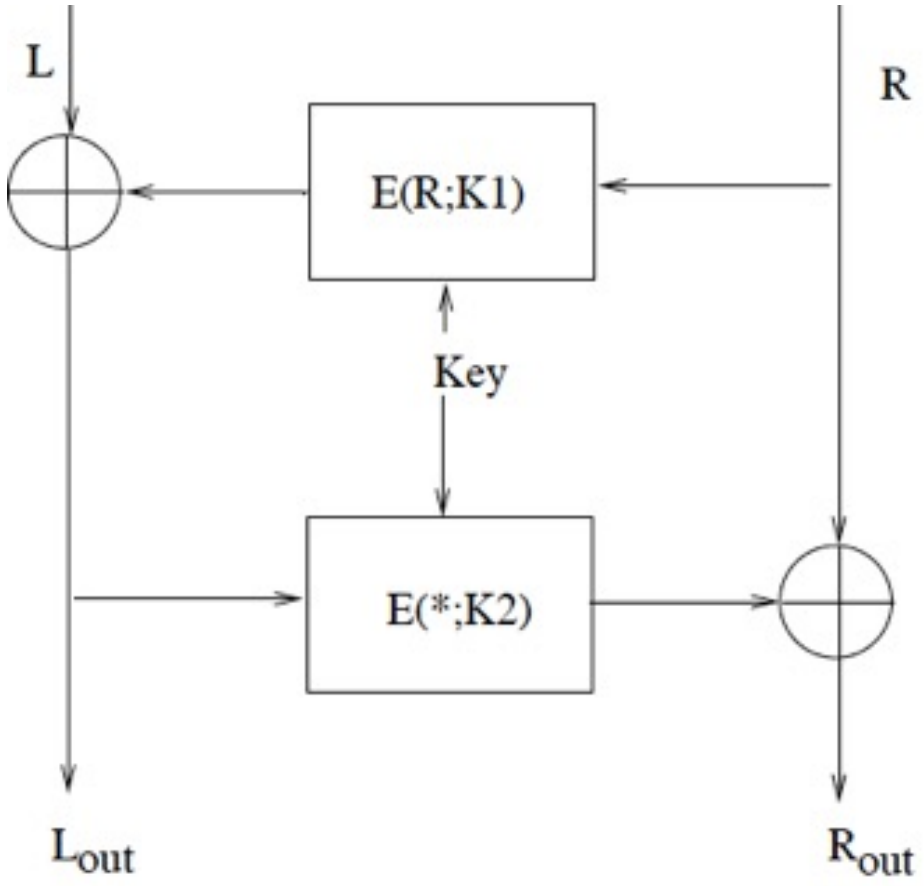
Feistel Cipher Structure (cont'd)



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

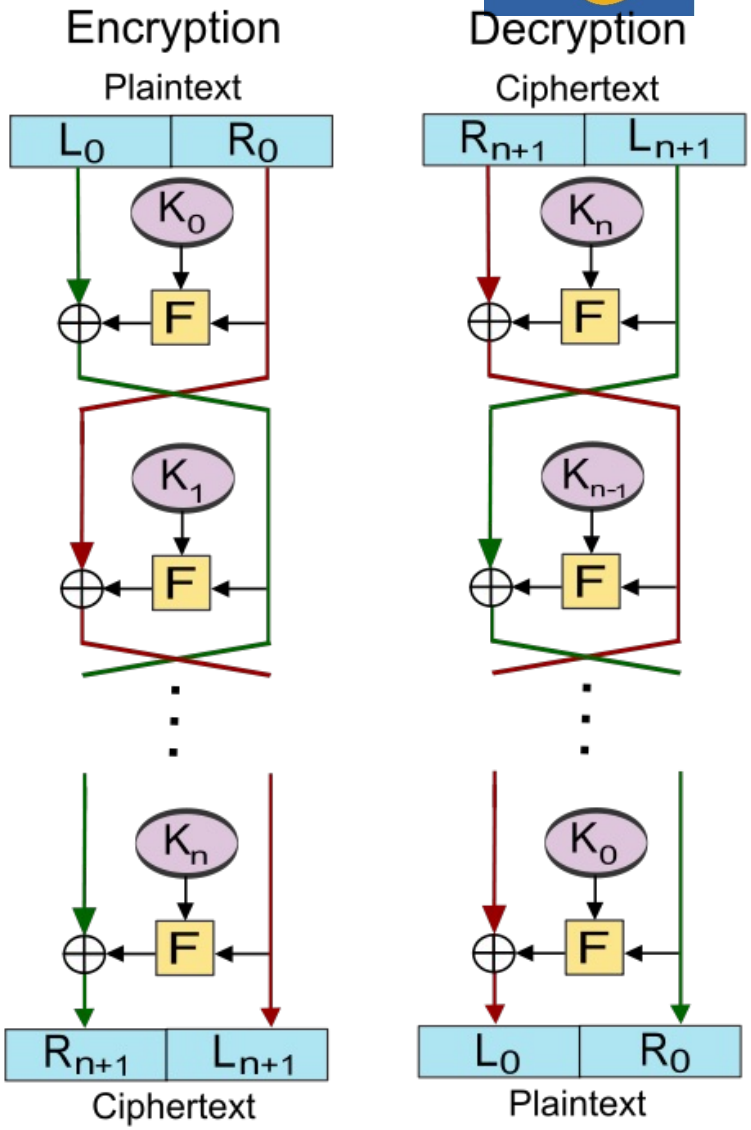
Exercise

- Calculate L_{out} and R_{out}
- Draw the decryption scheme.



Feistel Cipher Design Elements

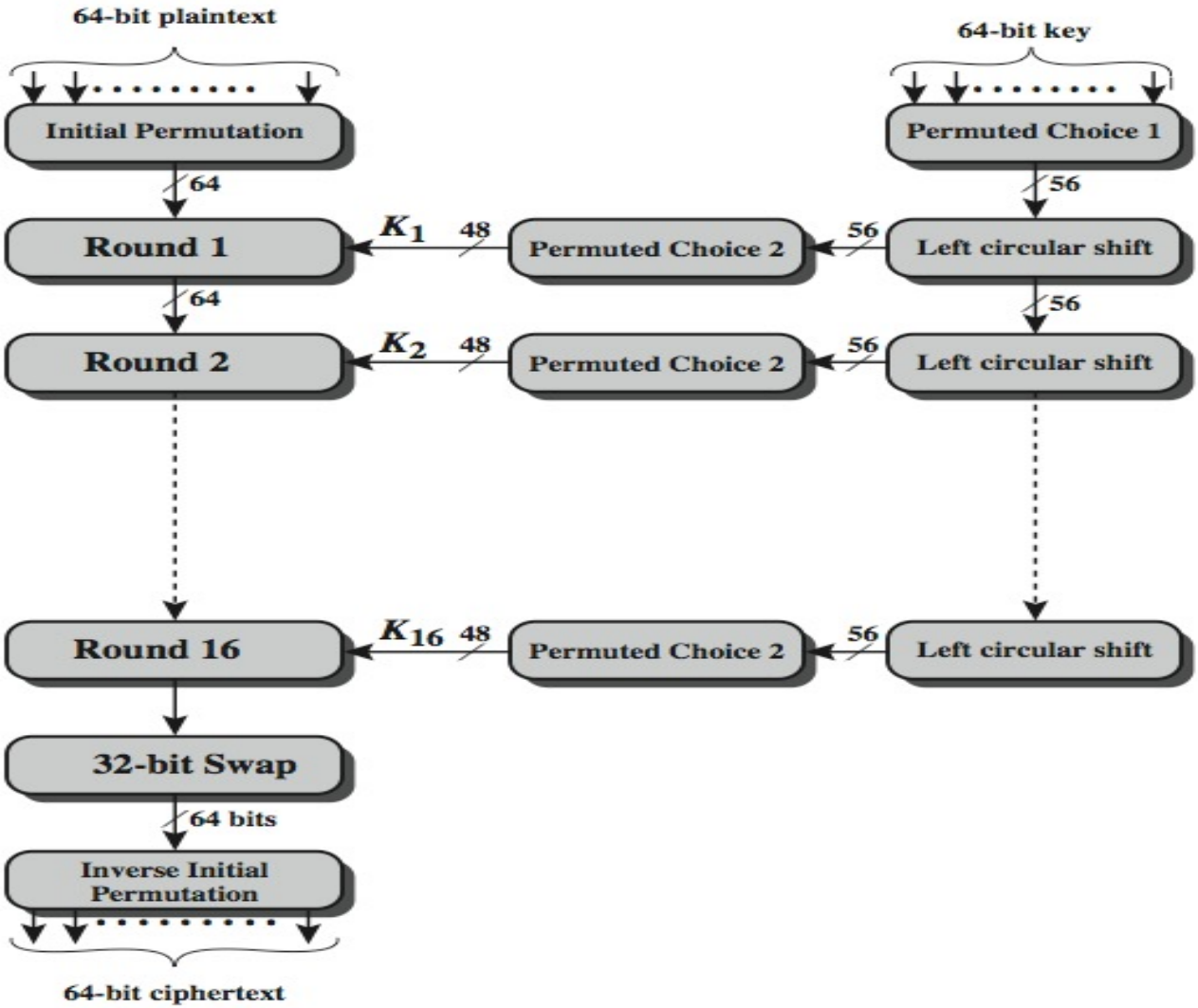
- Block size (greater => secure but slower)
- Key size (greater => secure but slower)
- Number of rounds (greater => secure but slower)
- Subkey generation algorithm
- Round function



Data Encryption Standard (DES)

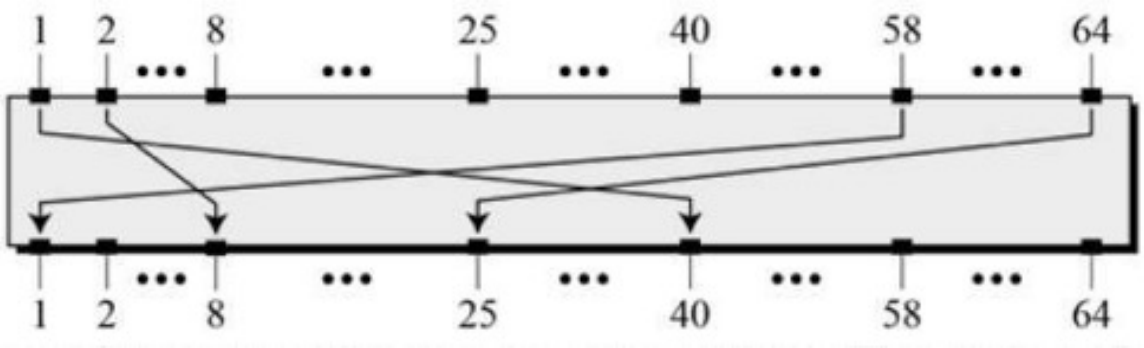
- Most widely used block cipher (until the appearance of AES in 2001)
- Originally developed by IBM (previous name: LUCIFER)
- Adopted in 1977 by NBS (now NIST) as *FIPS PUB 46*
- Encrypts 64-bit data using 56-bit key

DES Encryption Overview



Initial Permutation IP

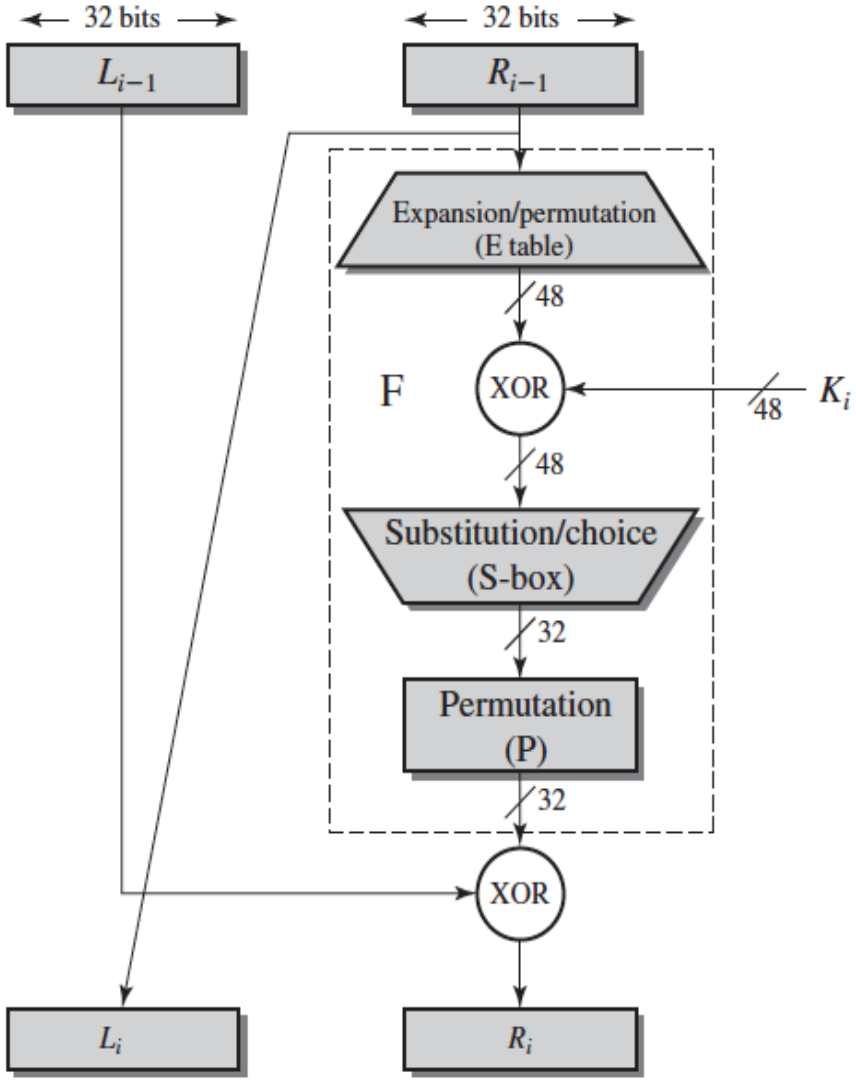
- First step
- IP reorders the input data bits
  - Even bits to LH half
  - Odd bits to RH half



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

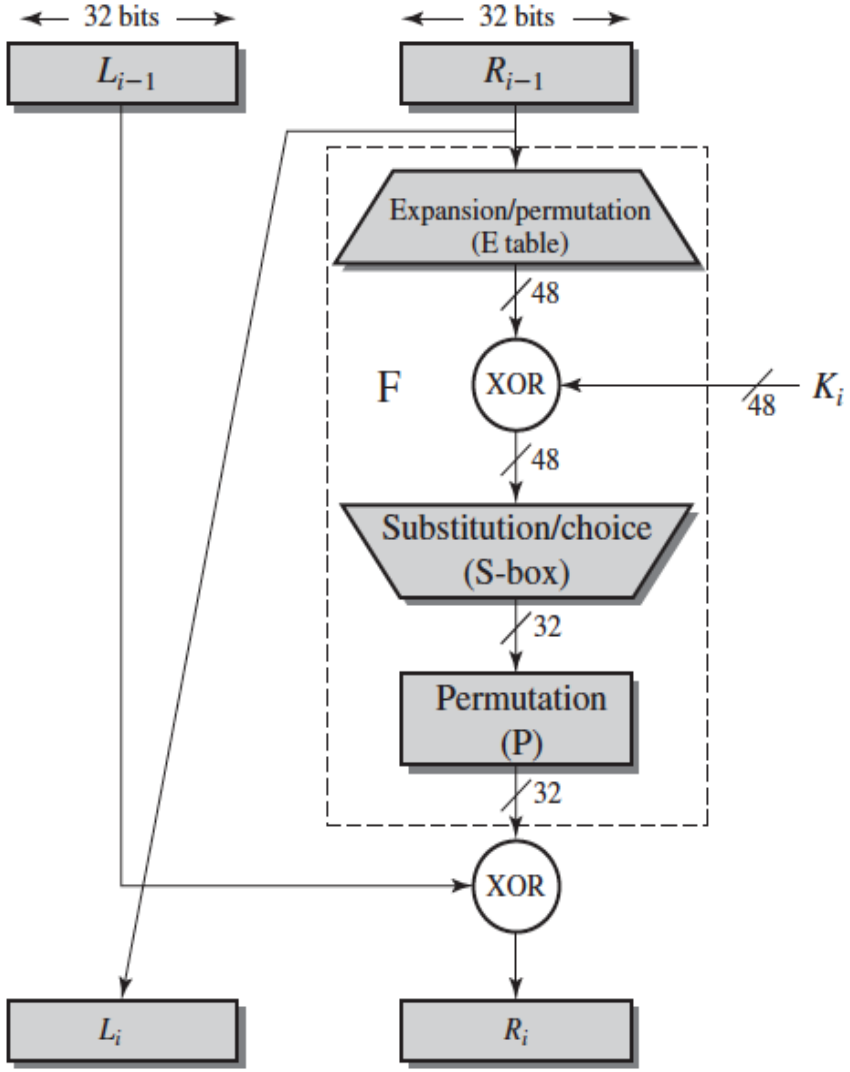
DES Round Structure

- Uses two 32-bit L & R halves
- Feistel Cipher
  - § $L_i = R_{i-1}$
  - § $R_i = L_{i-1} \oplus F(R_{i-1}, Ki)$
- F takes R half and 48-bit subkey
  - Expands R to 48-bits using Expansion/permutation table E
  - XOR with Subkey
  - Passes through 8 S-boxes to get 32-bit result
  - Permutes using permutation P

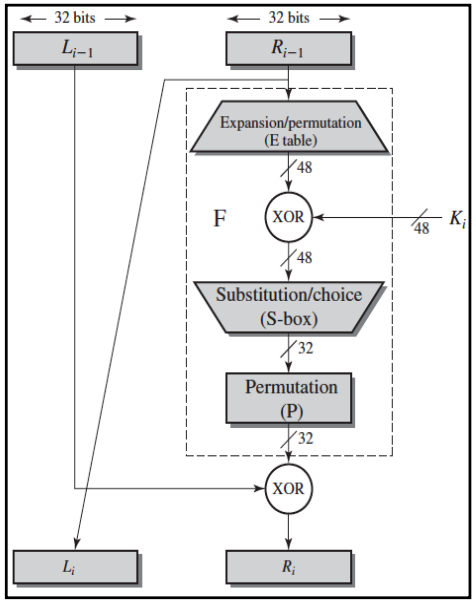


DES Round Structure

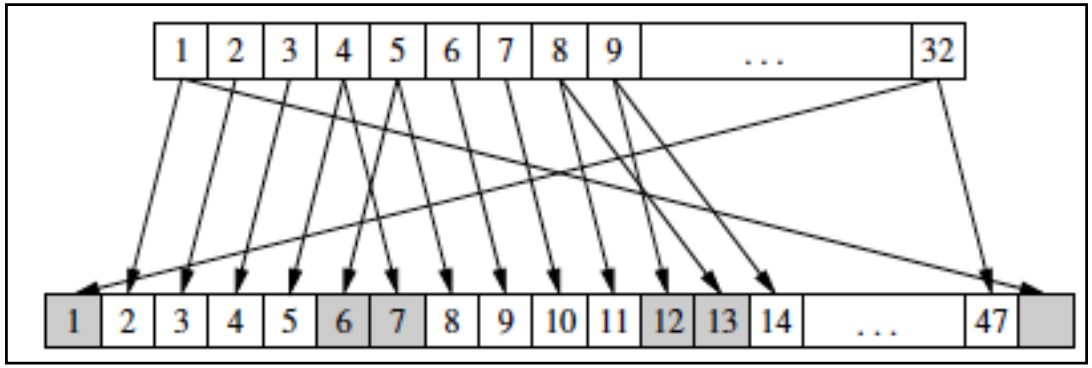
How are confusion and diffusion achieved in DES?



Expansion and Permutation (E)

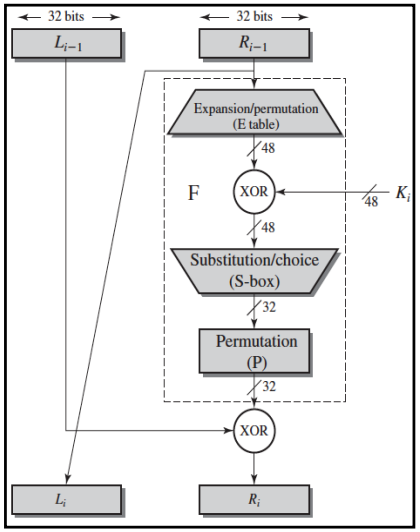


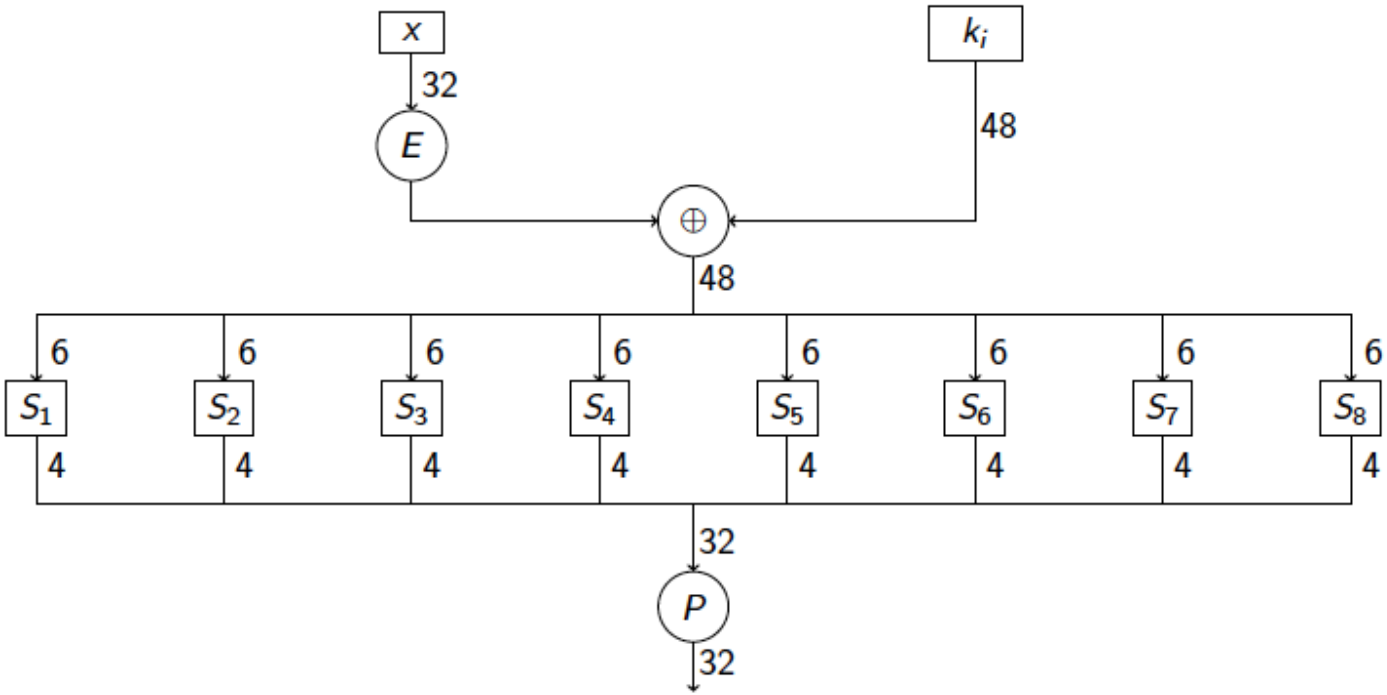
32	1	2	3	4	**5**
4	5	6	7	8	**9**
8	9	10	11	12	**13**
12	13	14	15	16	**17**
16	17	18	19	20	**21**
20	21	22	23	24	**25**
24	25	26	27	28	**29**
28	29	30	31	32	**1**



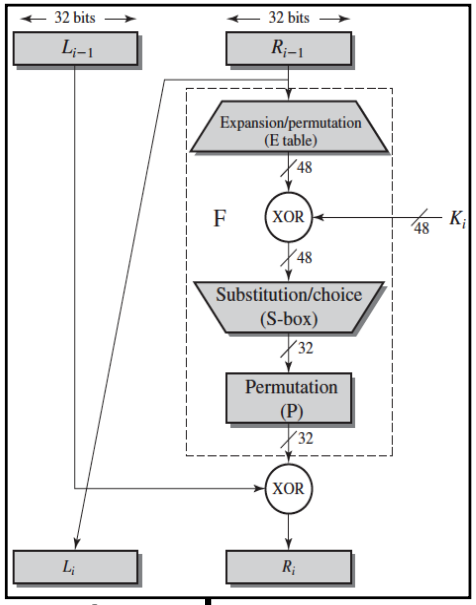
Substitution Boxes (S)

- Eight S-boxes which map 6 to 4 bits
- Implemented as lookup tables (faster)





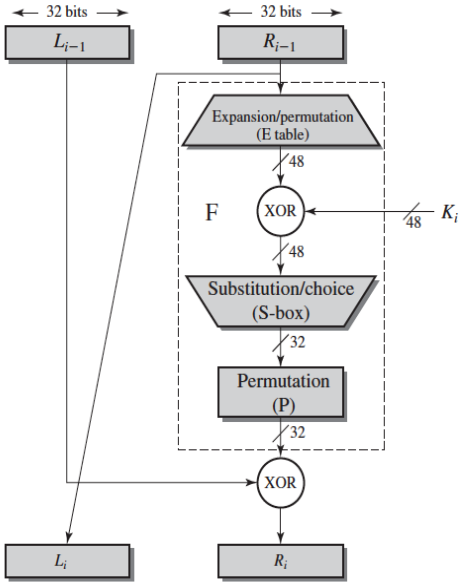
Substitution Boxes (S)



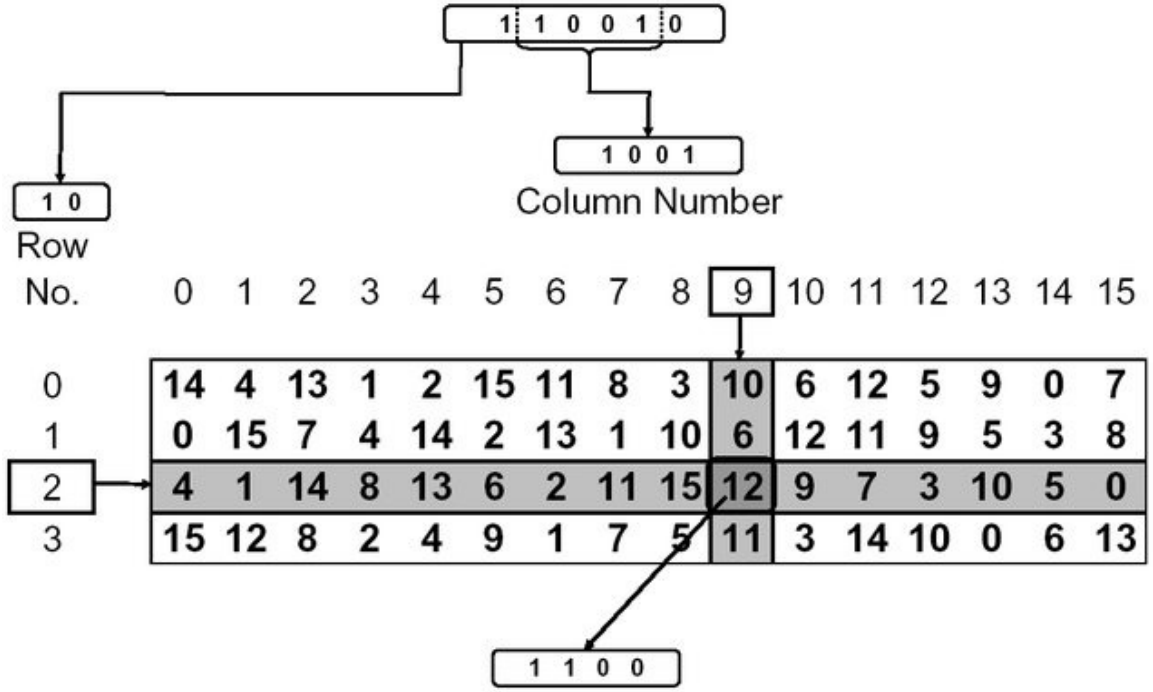
- Substitutes 6-bits input with 4-bits output
- Lookup table: 16 columns and 4 rows
  - 64 cells corresponds to all possible 2^6 input values
- Output directly read based on the input value

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

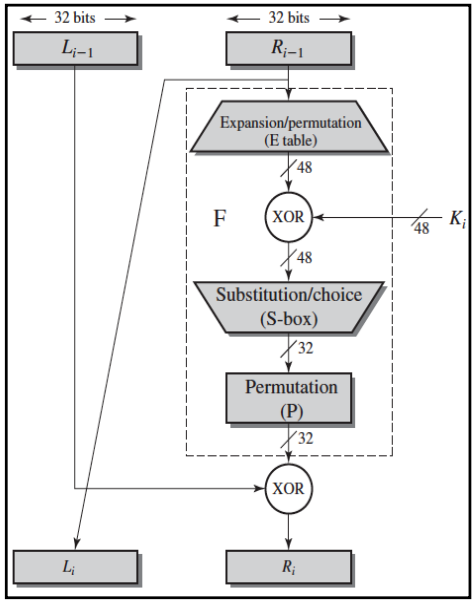
Substitution Boxes (S)



- Given a 6-bit input, the 4-bit output is found by selecting
  - Row index using the outer two bits (first and last bits)
  - The column using the inner four bits



Permutation (P-Box)



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

Final Permutation

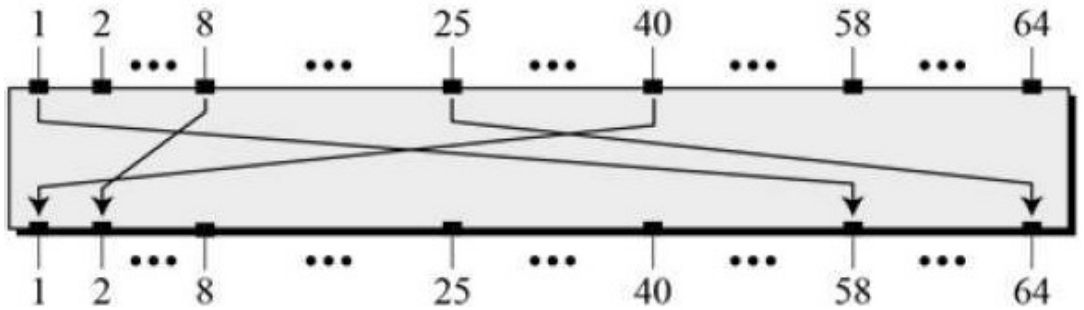
- Inverse the Initial Permutation (IP^{-1})

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

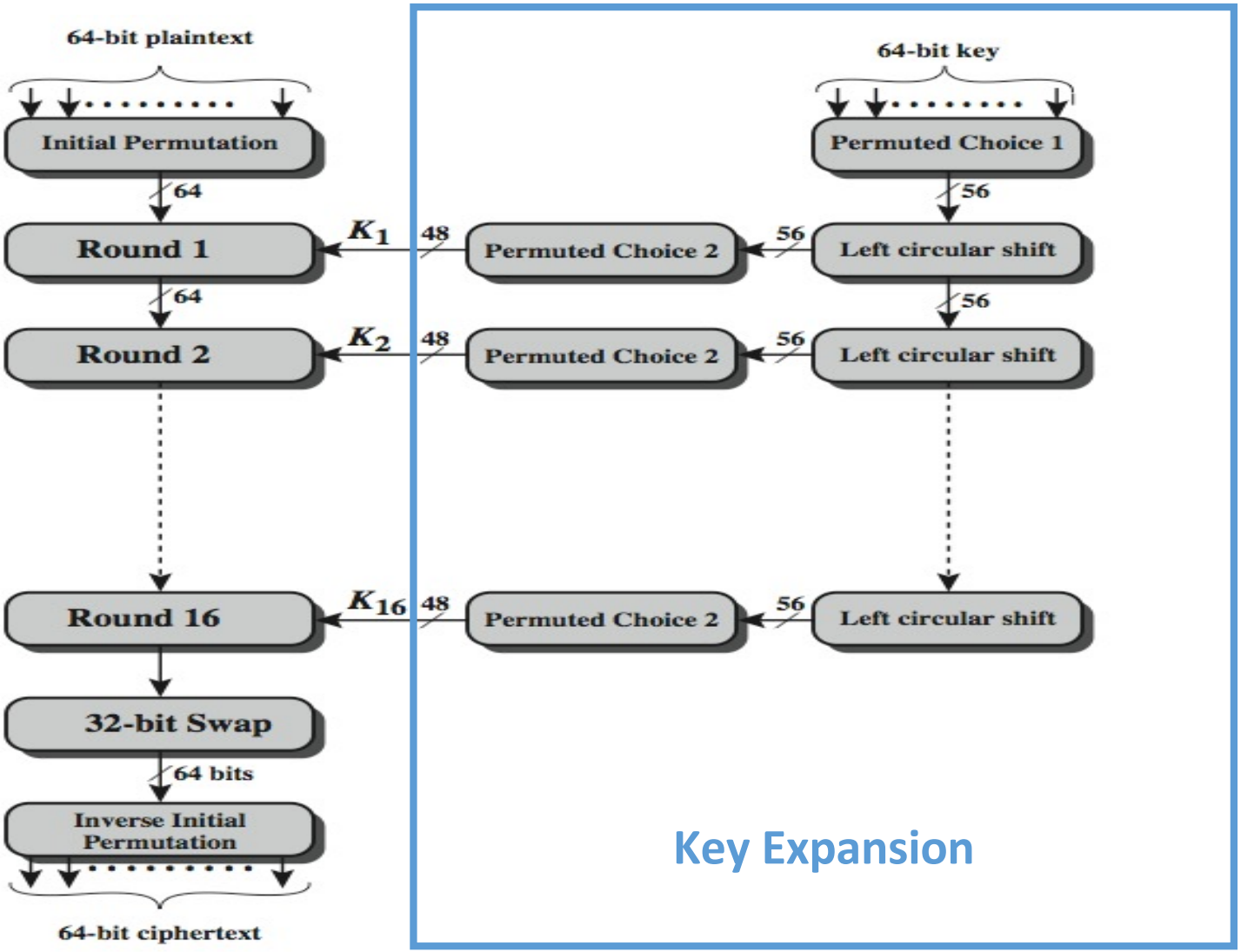
Initial Permutation Table

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

Final Permutation table

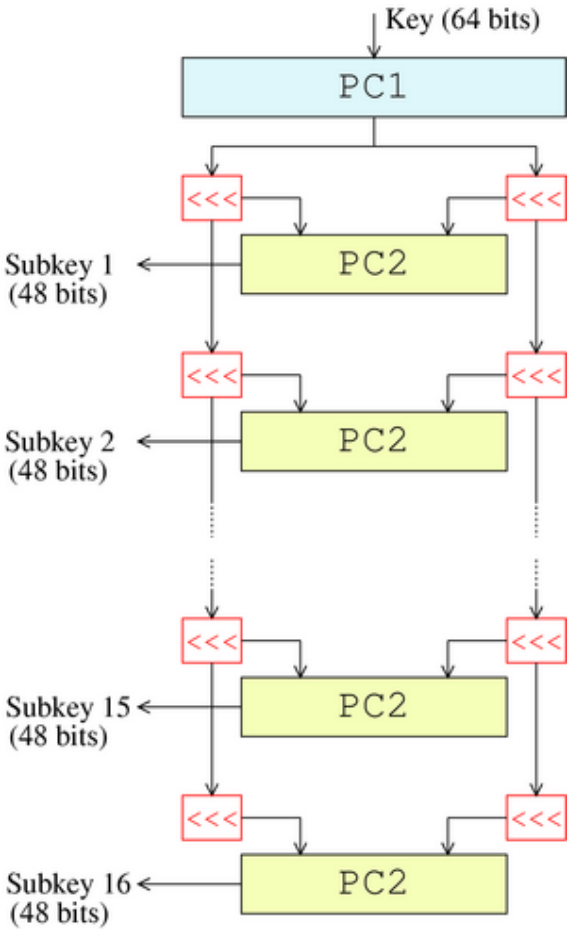


DES Encryption Overview - Revisited

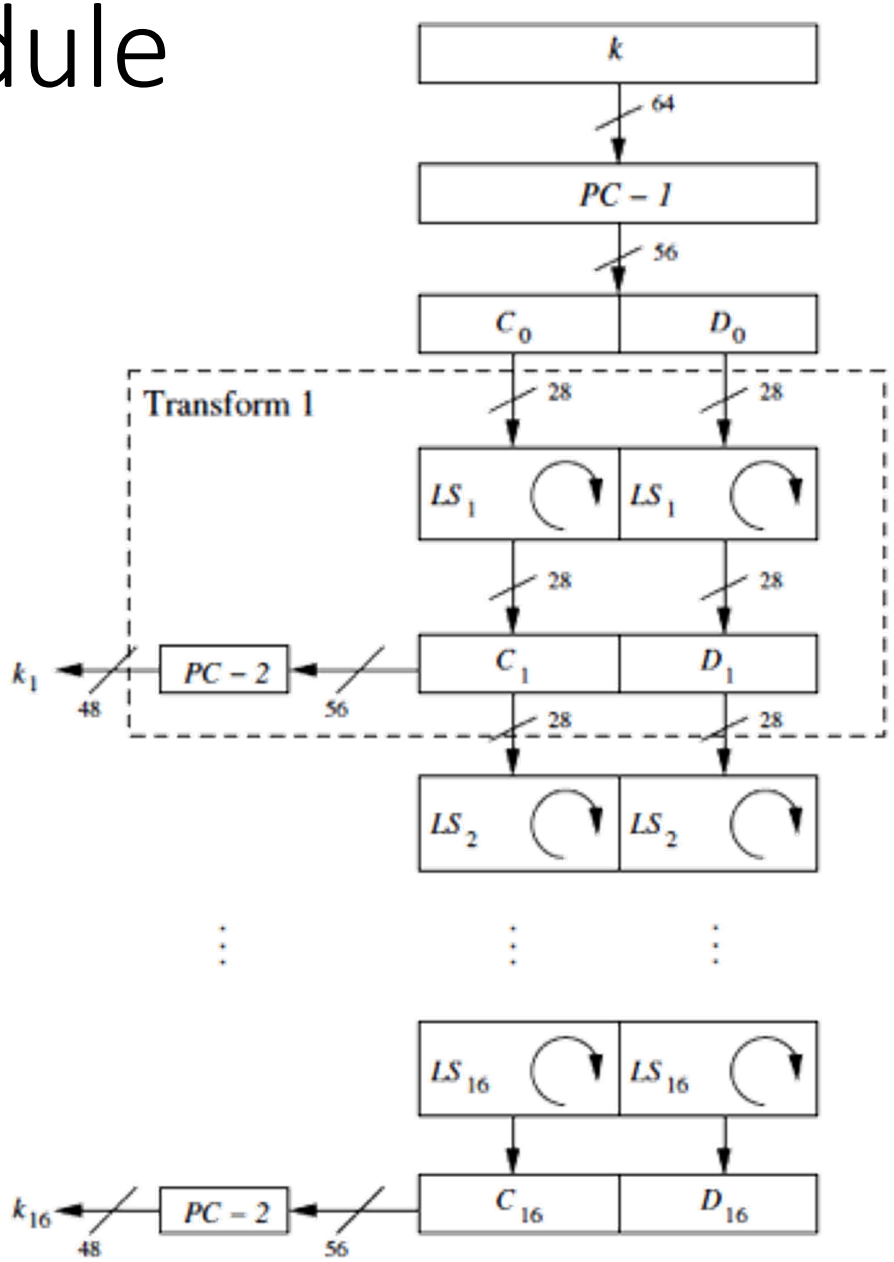


DES Key Schedule

- Forms **Round subkeys**
  - initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
  - **16 stages** consisting of:
    - Rotating each half separately either 1 or 2 places depending on the key rotation schedule K
    - Selecting 24-bits from each half & permuting them by PC2 for use in round function F



DES Key Schedule



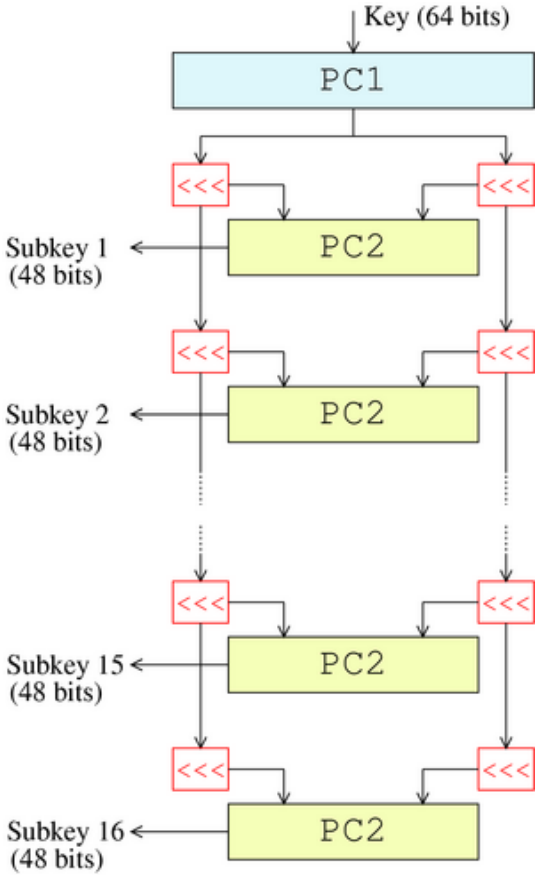
DES Key Schedule

Permutation Choice – 1 (PC1)

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

Permutation Choice – 2 (PC2)

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

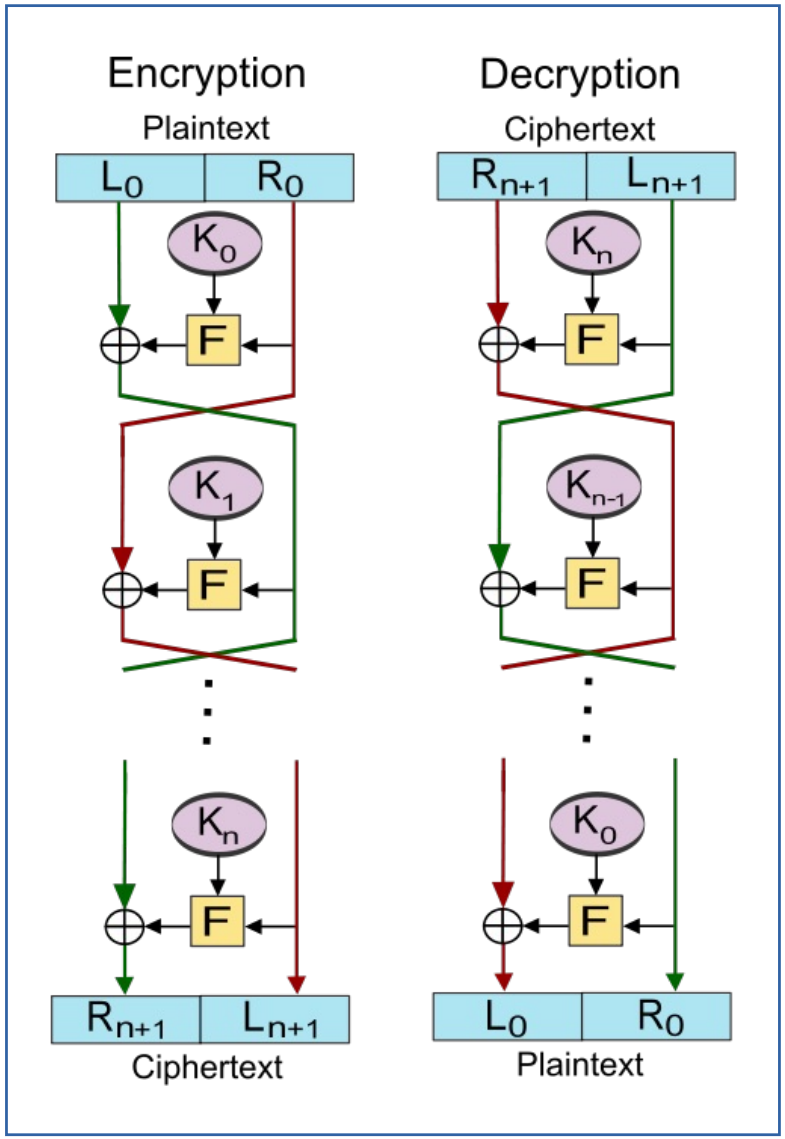


Round Nb	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Bits Rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

DES Decryption

- Decrypt must unwind steps of data computation
- With Feistel design, do encryption steps again using subkeys in reverse order (SK_{16} … SK_1)
  - IP undoes final FP step of encryption
  - 1st round with SK_{16} undoes 16th encrypt round
  - ….
  - 16th round with SK_1 undoes 1st encrypt round
  - Final Permutation undoes initial encryption IP
  - thus recovering original data value

DES Decryption



Avalanche Effect

- Key desirable property of encryption algorithm
- A change of one input or key bit results in changing approx. half output bits
- Making attempts to “home-in” by guessing keys impossible

Avalanche Effect in DES

Round		δ
	02468aceeca86420	1
	12468aceeca86420	
1	3cf03c0fbad22845	1
	3cf03c0fbad32845	
2	bad2284599e9b723	5
	bad3284539a9b7a3	
3	99e9b7230bae3b9e	18
	39a9b7a3171cb8b3	
4	0bae3b9e42415649	34
	171cb8b3ccaca55e	
5	4241564918b3fa41	37
	ccaca55ed16c3653	
6	18b3fa419616fe23	33
	d16c3653cf402c68	
7	9616fe2367117cf2	32
	cf402c682b2cefbc	
8	67117cf2c11bfc09	33
	2b2cefbc99f91153	

Round		δ
9	c11bfc09887fbc6c	32
	99f911532eed7d94	
10	887fbc6c600f7e8b	34
	2eed7d94d0f23094	
11	600f7e8bf596506e	37
	d0f23094455da9c4	
12	f596506e738538b8	31
	455da9c47f6e3cf3	
13	738538b8c6a62c4e	29
	7f6e3cf34bc1a8d9	
14	c6a62c4e56b0bd75	33
	4bc1a8d91e07d409	
15	56b0bd7575e8fd8f	31
	1e07d4091ce2e6dc	
16	75e8fd8f25896490	32
	1ce2e6dc365e5f59	
IP^{-1}	da02ce3a89ecac3b	32
	057cde97d7683f2a	

Attacking DES – DES Challenge

PT	=	The_unkn	own_mess	age_is:_	XXX...
CT	=	c_1	c_2	c_3	c_4

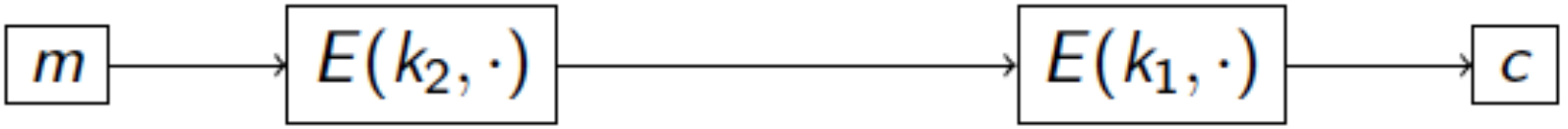
Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 Decryptions/s	Time Required at 10^{13} Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour

Multiple Encryption & DES

- A replacement for DES was needed
  - theoretical attacks that can break it
  - demonstrated exhaustive key search attacks
- AES is a new cipher alternative
- Prior to this alternative was to **use multiple encryption with DES** implementations

Double-DES?

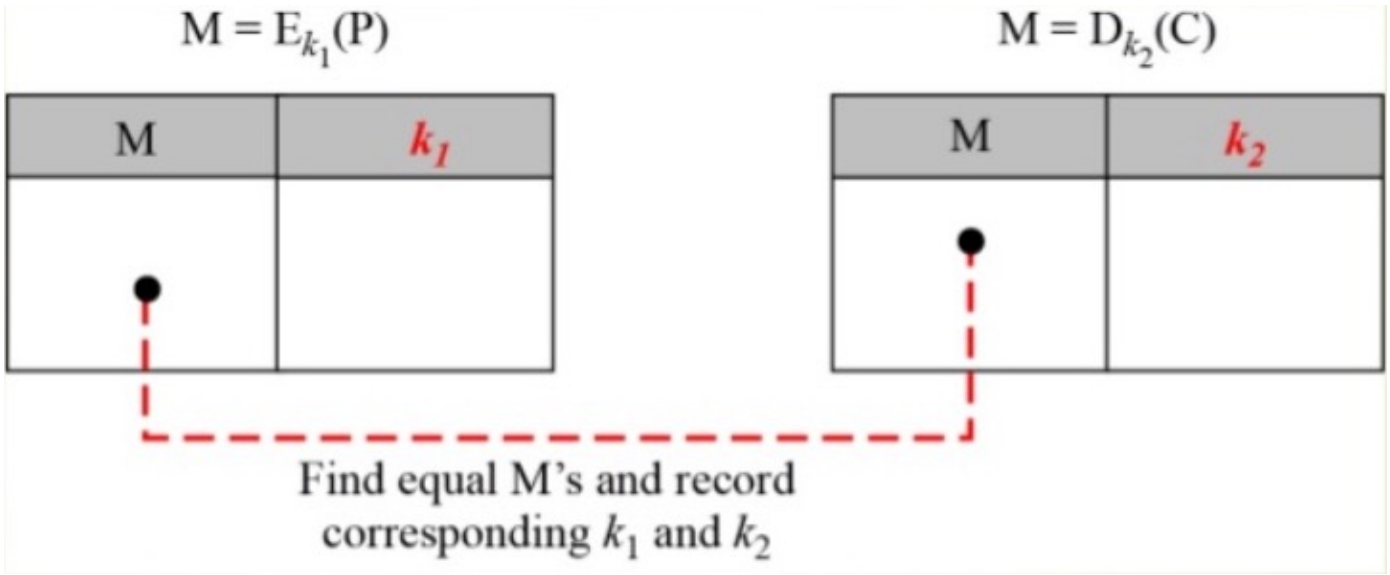
- Could use 2 DES encrypts on each block
  - $C = E_{K_1}(E_{K_2}(M))$



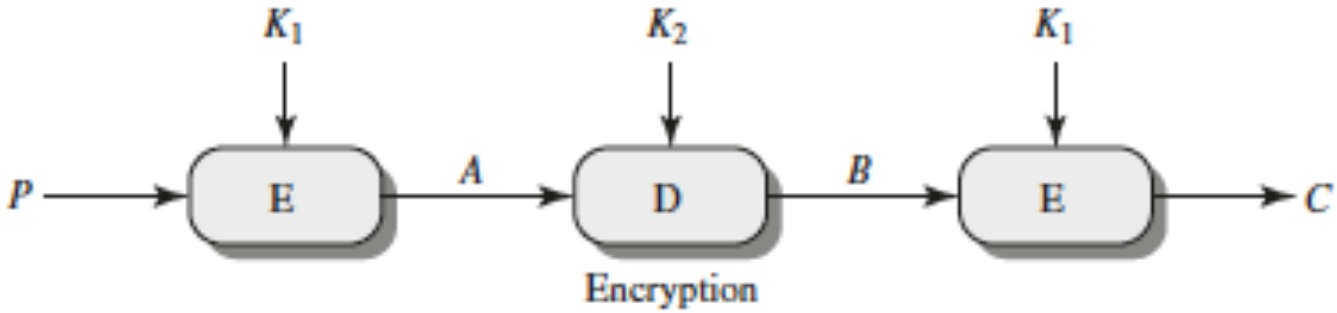
- Key length is 112 bits
  - Exhaustive search attack => 2^{112} time
- Question: is there a better attack?

Double DES?
Meet-In-The-Middle attack

- Step 1: Encrypt P with all possible values of K_2
  - build table (2^{56} entries)
  - Sort the table by the value of the second column
- Step 2: for all $k \in \{0,1\}_{56}$ do:
  - test if D(k, C) is in 2nd column.
  - if so then $E(k_i, M) = D(k, C) \Rightarrow (k_i, k) = (k_2, k_1)$



Triple-DES with Two-Keys



- Can use 2 keys (or three keys) with E-D-E sequence
  - $C = E_{K_1}(D_{K_2}(E_{K_1}(P)))$
  - $C = E_{K_3}(D_{K_2}(E_{K_1}(P)))$
  - Encrypt & decrypt are equivalent in security
  - if $K_1 = K_2$ then can work with single DES
- No current known practical attacks