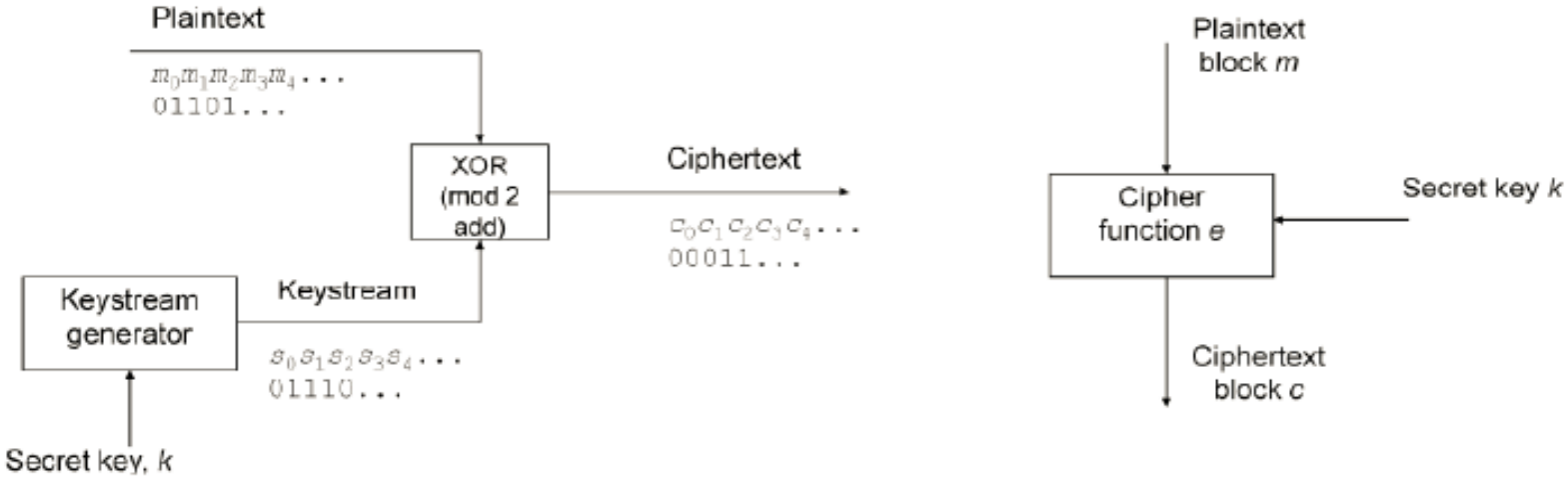
Block Ciphers, DES, 2DES, 3DES

Stream vs Block Ciphers



- Both are symmetric key ciphers
- **Stream ciphers** process messages a bit or byte at a time when en/decrypting, E.g. RC4, A5/1, A5/2
- **Block cipher** operates on fixed-length groups of bits, called blocks,
- Well-known block ciphers, E.g. DES, Triple DES, AES

Figure 4-2 General n-bit-n-bit Block Substitution (Shown with n=4)

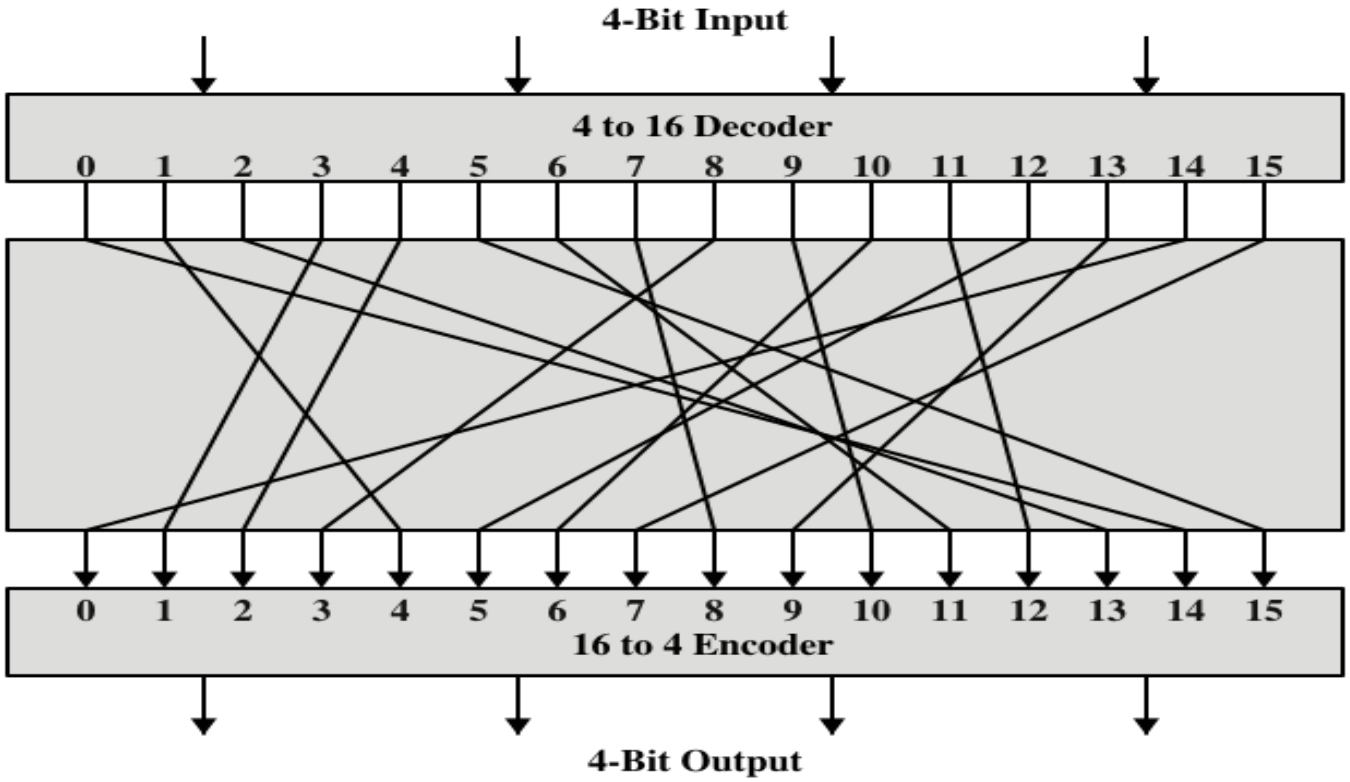


Table 4-1 Encryption and Decryption Tables for Substitution Cipher of Figure 4-2 (1 of 2)

Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111

Table 4-1 Encryption and Decryption Tables for Substitution Cipher of Figure 4-2 (2 of 2)

Plaintext	Ciphertext
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

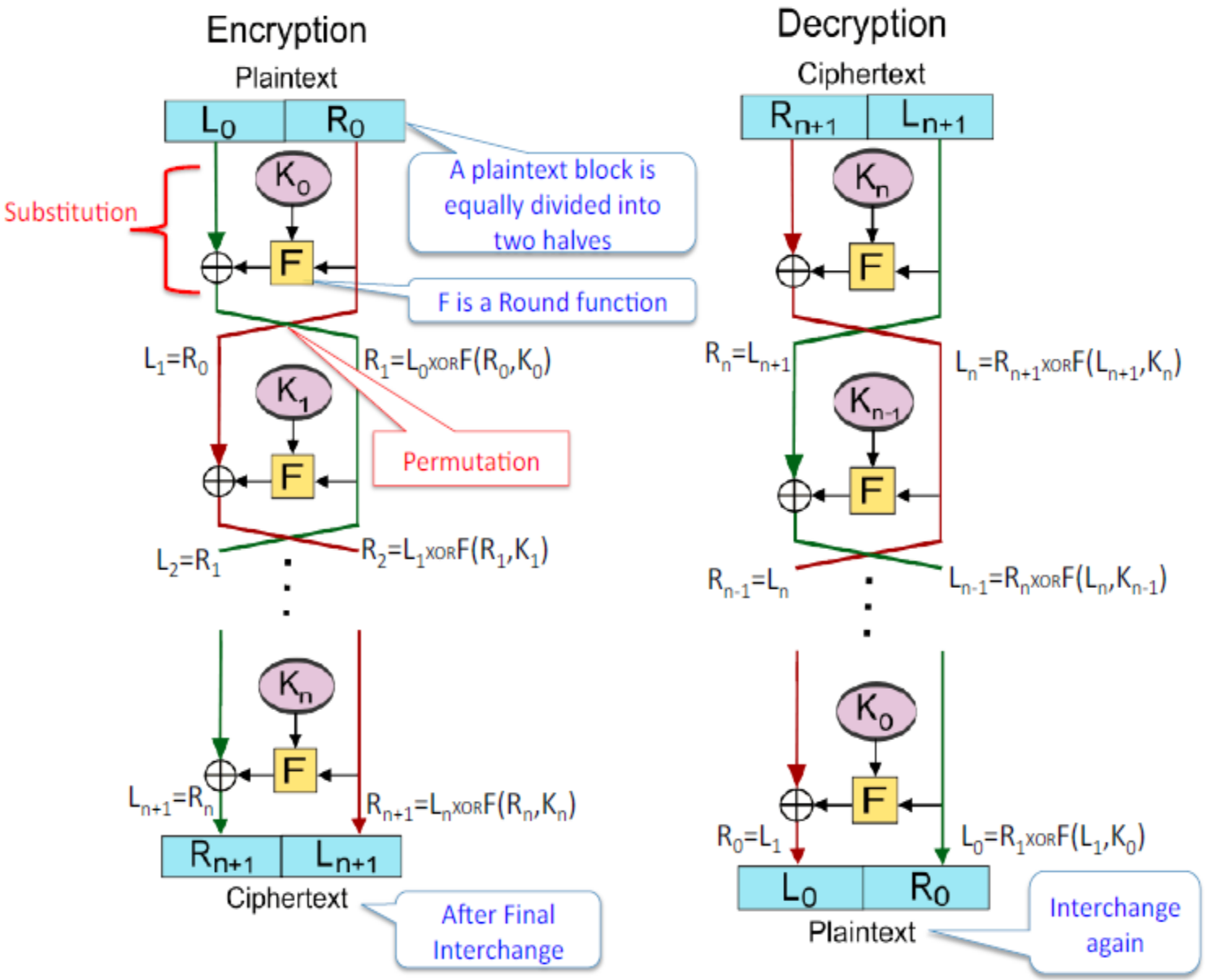
Plaintext	Ciphertext
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

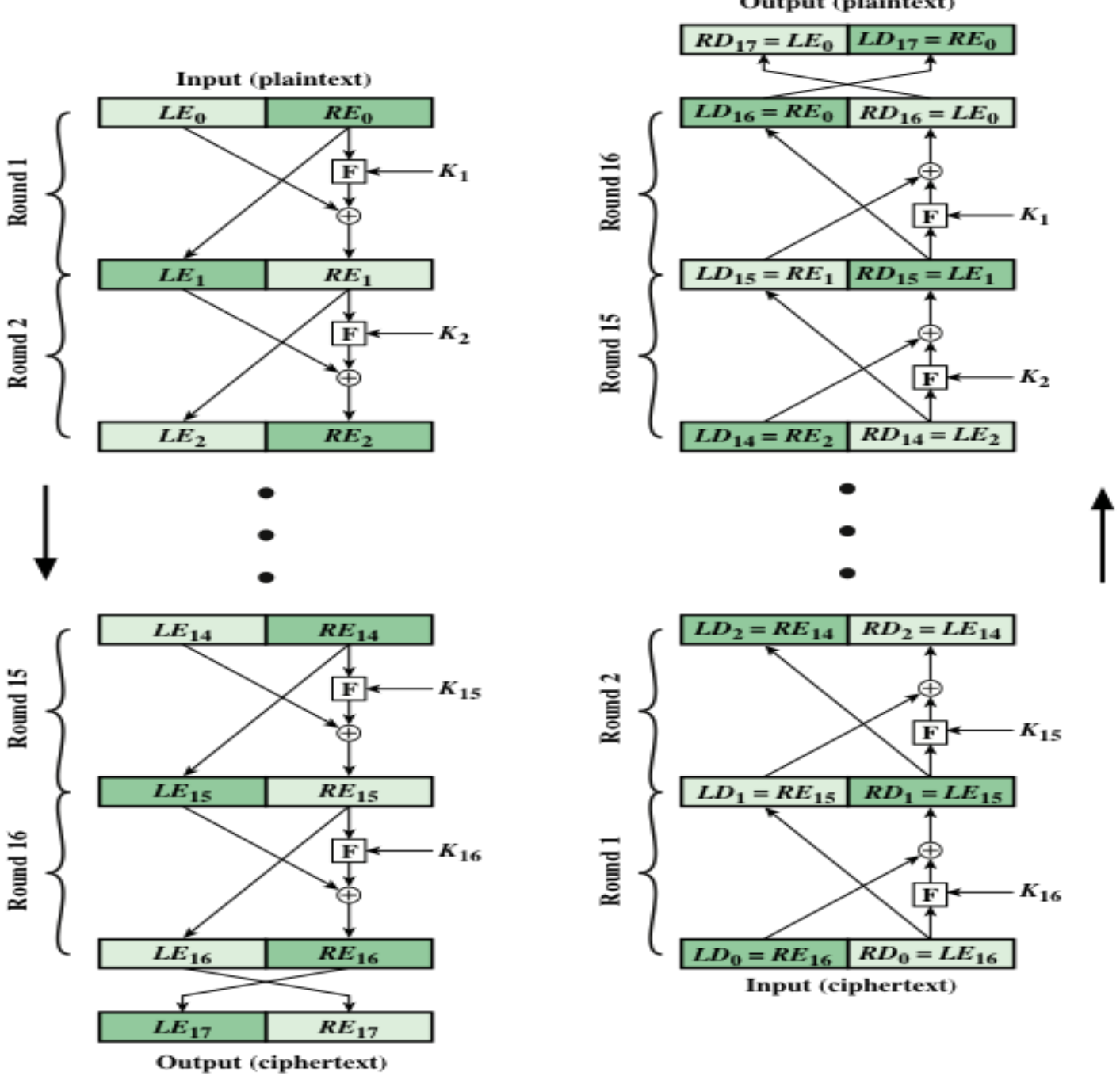
S-P Network

- Claude Shannon introduced the idea of substitution-permutation **S-P networks** in 1949
- Form a basis of modern block ciphers
- Introduced the ideas of **confusion** & **diffusion**
  - **Confusion** seeks to make the relationship between the statistics of the **ciphertext** and the value of the encryption **key** as complex as possible
  - **Diffusion** seeks to make the statistical relationship between the **plaintext** and **ciphertext** as complex as possible
- S-P Network consists of:
  - substitution (S-box) (improves confusion)
  - permutation (P-box) (improves diffusion)

Feistel Cipher *By German-born physicist **Horst Feistel***

- A cipher scheme with iterated use of Round function
- Properties:
  - Symmetric Key Cipher
  - Block Cipher
  - Product Cipher
- Combines multiple rounds of repeated operations to achieve high degree of Confusion and Diffusion
  - **Bit-shuffling** (often called permutation boxes or P-boxes)
  - **Simple non-linear functions** (often called substitution boxes or S-boxes)
  - **Linear mixing** (in the sense of modular algebra) using XOR





Feistel Cipher (1 of 2)

- Feistel proposed the use of a cipher that alternates substitutions and permutations
  - Substitutions
    - Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements
  - Permutation
    - No elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed

Feistel Cipher (2 of 2)

- Is a practical application of a proposal by Claude Shannon to develop a product cipher that alternates confusion and diffusion functions
- Is the structure used by many significant symmetric block ciphers currently in use

Diffusion and Confusion (1 of 2)

- Terms introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system
  - Shannon's concern was to thwart cryptanalysis based on statistical analysis
- Diffusion
  - The statistical structure of the plaintext is dissipated into long-range statistics of the ciphertext
  - This is achieved by having each plaintext digit affect the value of many ciphertext digits

Diffusion and Confusion (2 of 2)

- Confusion
  - Seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible
  - Even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key

Feistel Cipher Design Features (1 of 2)

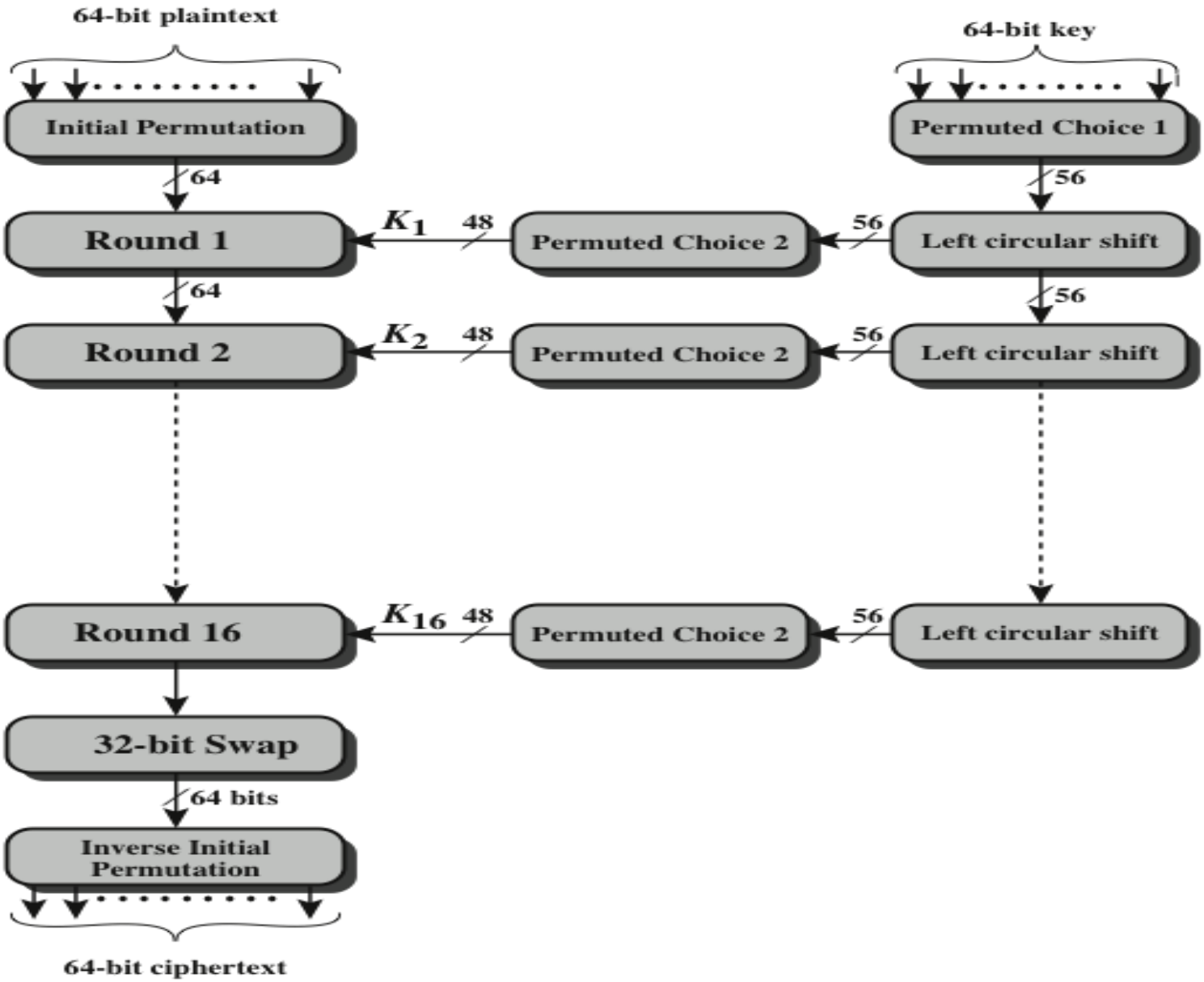
- Block size
  - Larger block sizes mean greater security but reduced encryption/decryption speed for a given algorithm
- Key size
  - Larger key size means greater security but may decrease encryption/decryption speeds
- Number of rounds
  - The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security
- Subkey generation algorithm
  - Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis

Feistel Cipher Design Features (2 of 2)

- Round function F
  - Greater complexity generally means greater resistance to cryptanalysis
- Fast software encryption/decryption
  - In many cases, encrypting is embedded in applications or utility functions in such a way as to preclude a hardware implementation; accordingly, the speed of execution of the algorithm becomes a concern
- Ease of analysis
  - If the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength

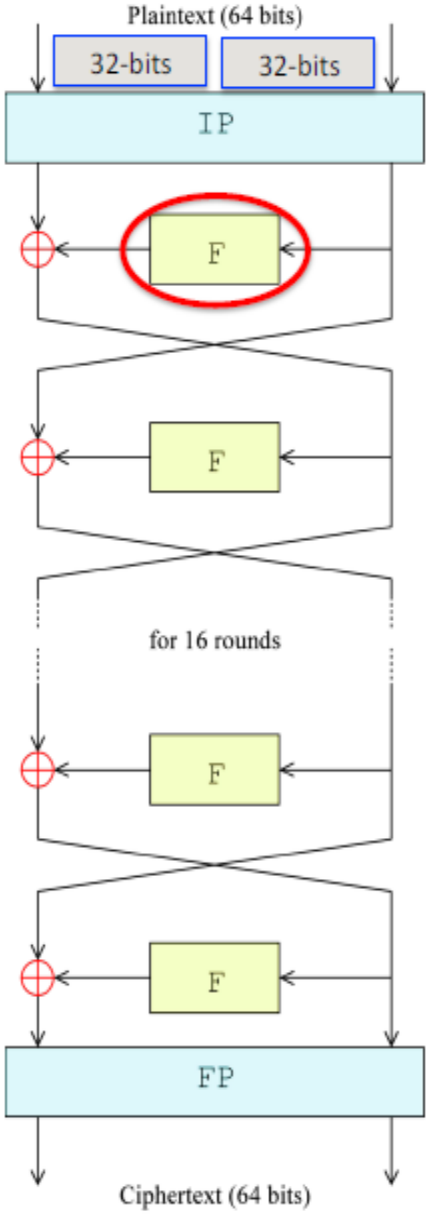
Data Encryption Standard (**DES**)

- Selected by the *National Bureau of Standards* as an official *Federal Information Processing Standard* (FIPS) for the United States in 1976
- In January, 1999, distributed.net and the Electronic Frontier Foundation collaborated to publicly break a DES key in 22 hours and 15 minutes
- Replaced by Triple DES (TDES or 3DES)
- 64-bits plaintext block and key size (out of which 56-bits of keys are effectively used)
- Based upon **Feistel Structure** (which ensures that decryption/ encryption are very similar processes — the only difference is that the sub-keys are applied in the reverse order when decrypting)

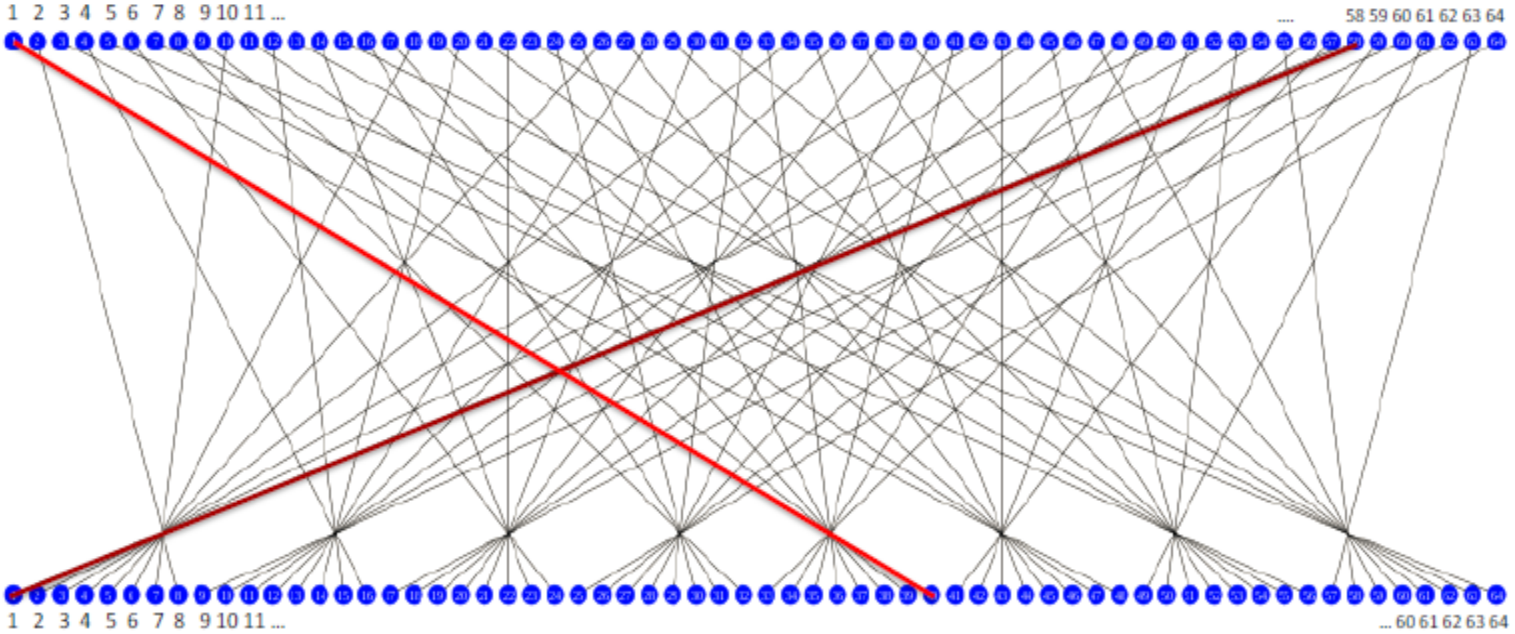


DES Overall Structure

- 64-bits input is divided into two halves (Feistel Structure)
- 16 identical stages of processing, termed as ***rounds***
- Initial and Final Permutation (IP and FP) inverses of each other
- The **F-function** scrambles half a block together with part of the key
- The output from the **F-function** is then combined with the other half of the block, and the halves are swapped before the next round.

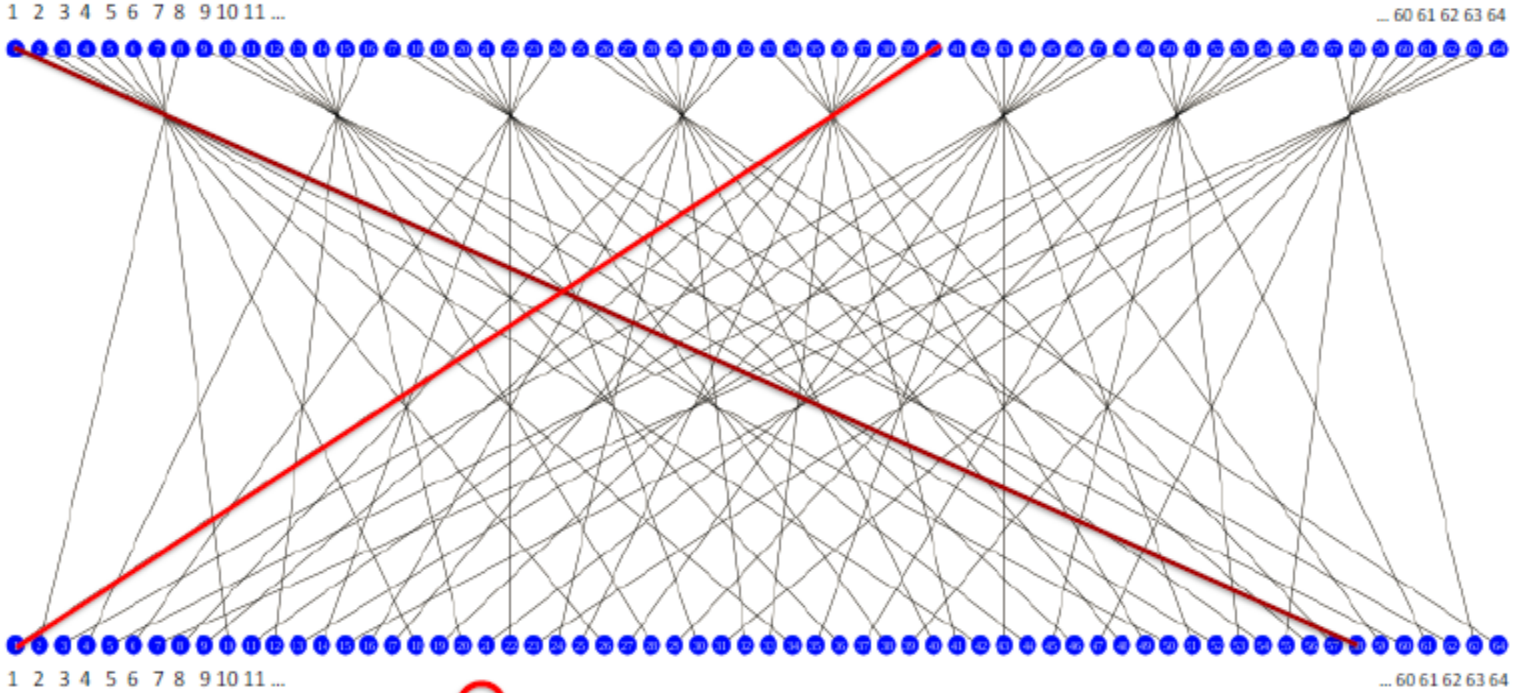


Initial Permutation



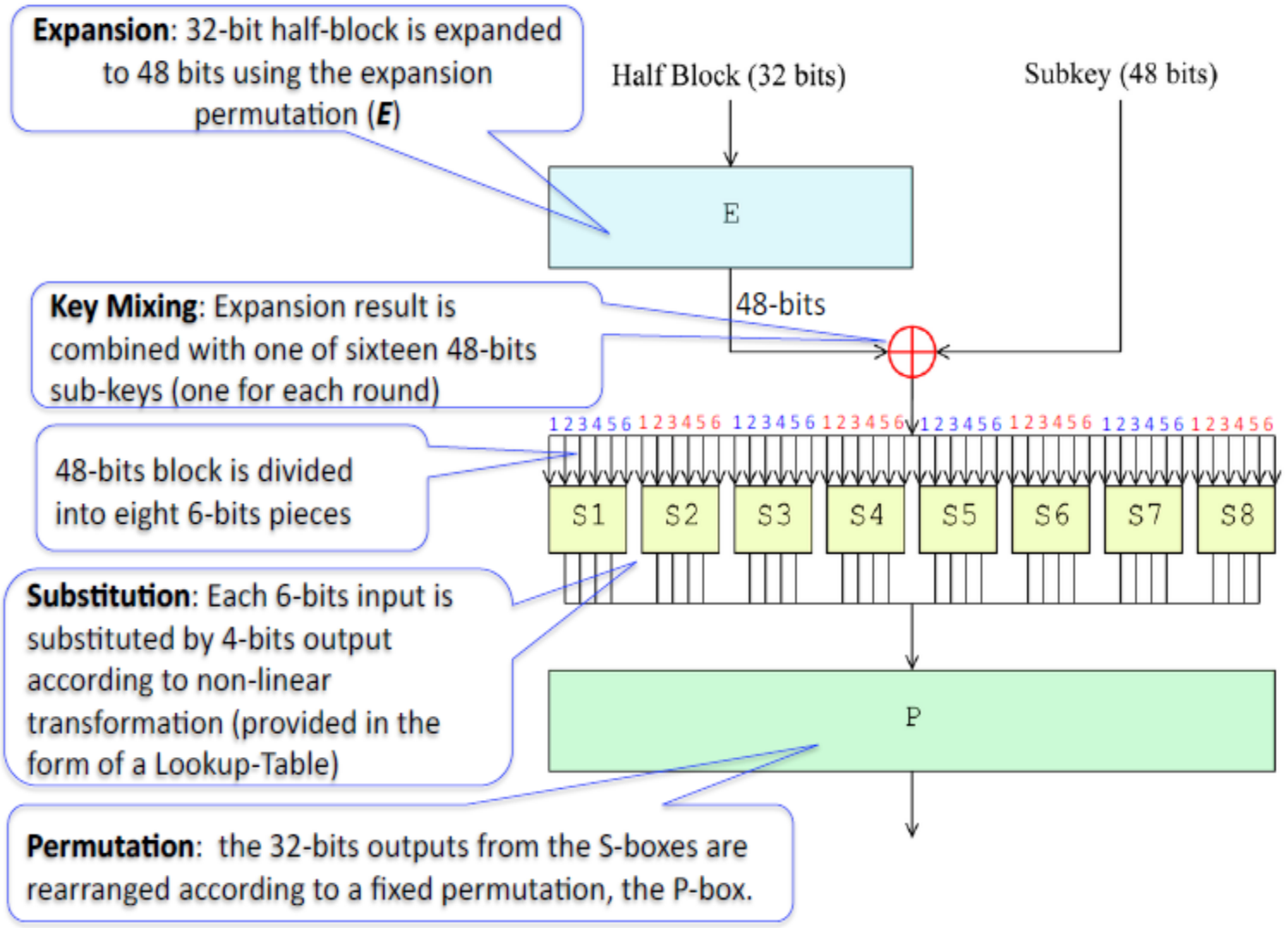
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Final Permutation

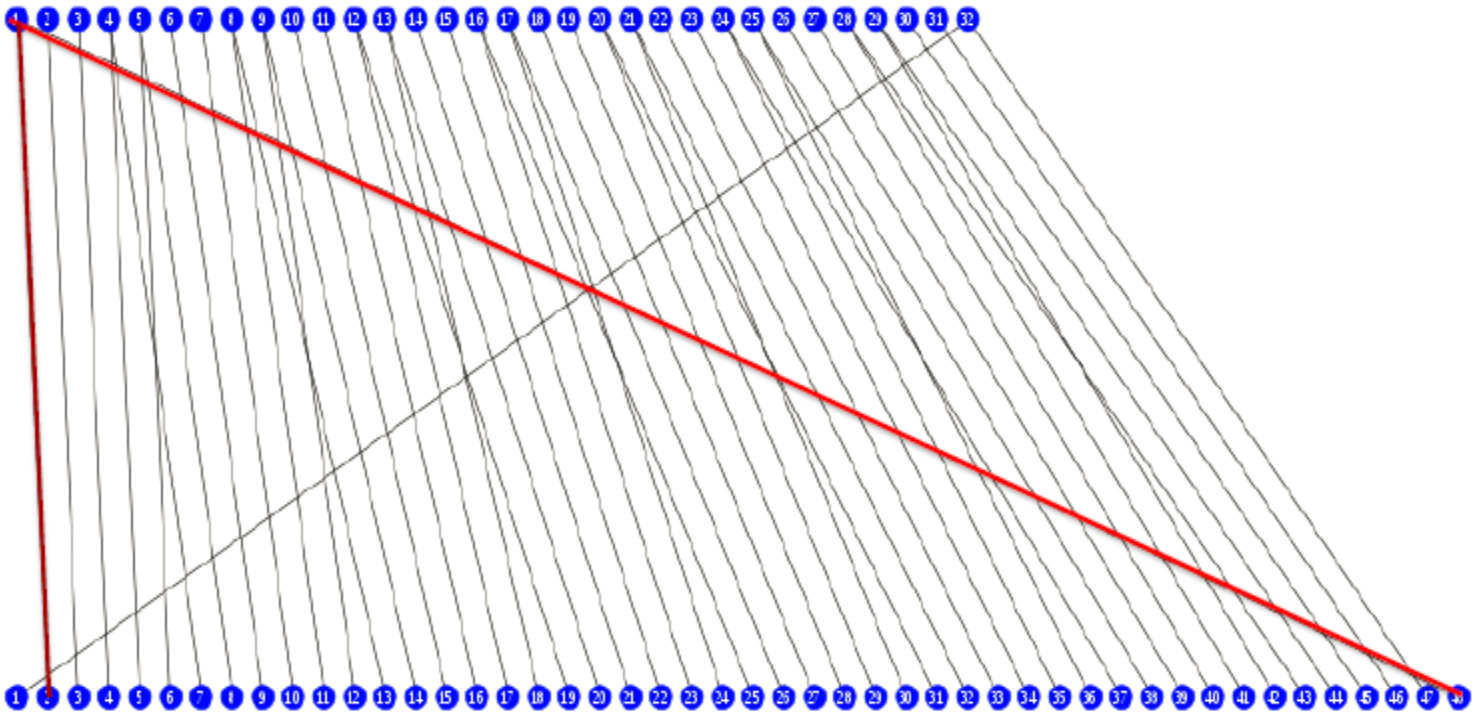


40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

The Round Function F



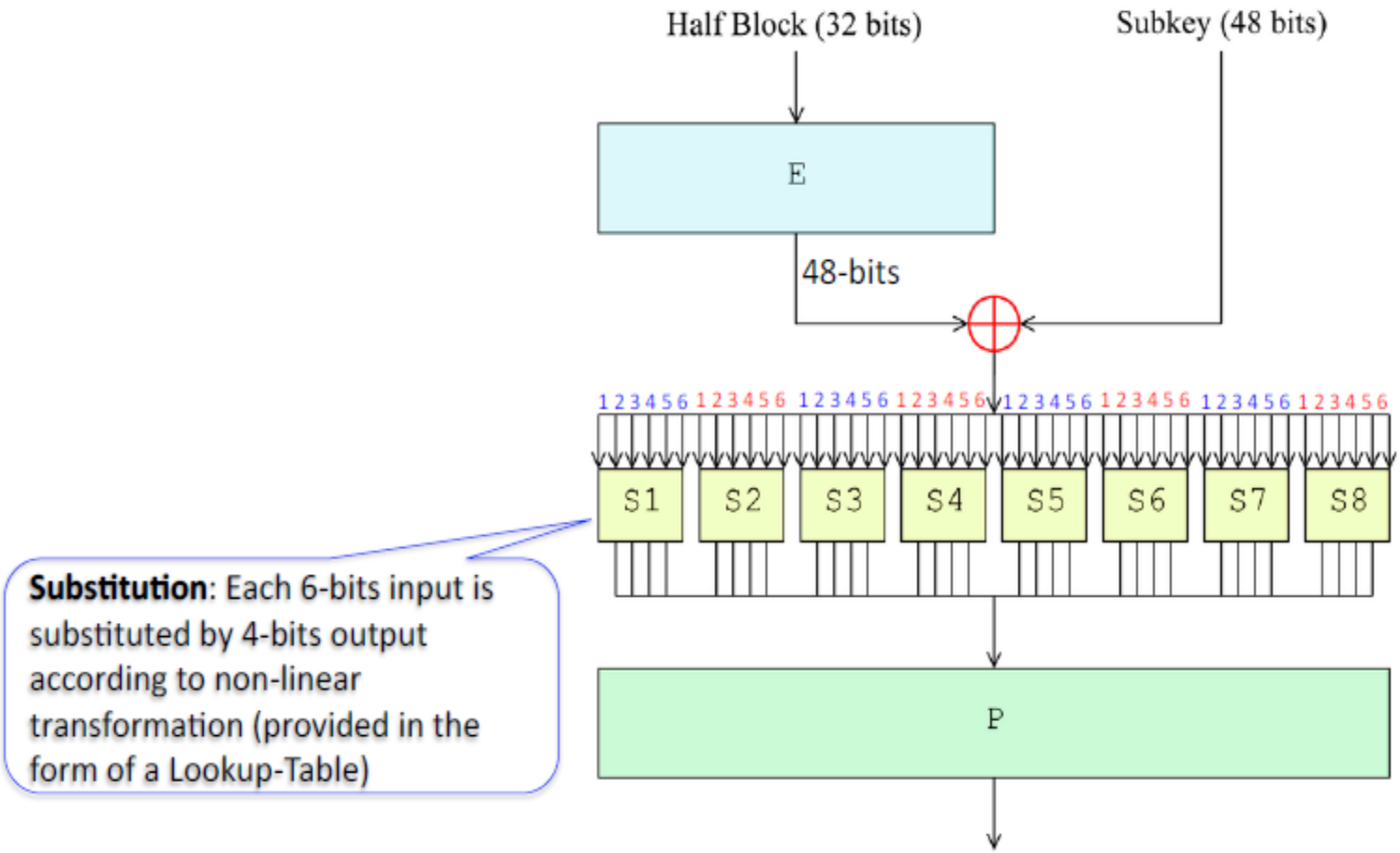
Expansion Function (E)



The expansion function is interpreted as for the initial and final permutations. Note that some bits from the input are duplicated at the output; e.g. the first bit of the input is duplicated in both the second and forty-eighth bit of the output. Thus, the 32-bit half-block is expanded to 48 bits.

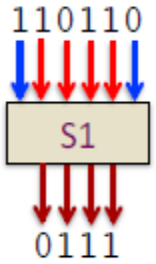
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

The Round Function F



Substitution

- Eight S-boxes each map 6 to 4 bits
- Each S-box is specified as a unique 4x16 table
  - each row is a permutation of 0-15 (0000 → 1111)
  - outer bits 1 & 6 of input are used to select one of the four rows
  - inner 4 bits of input are used to select a column

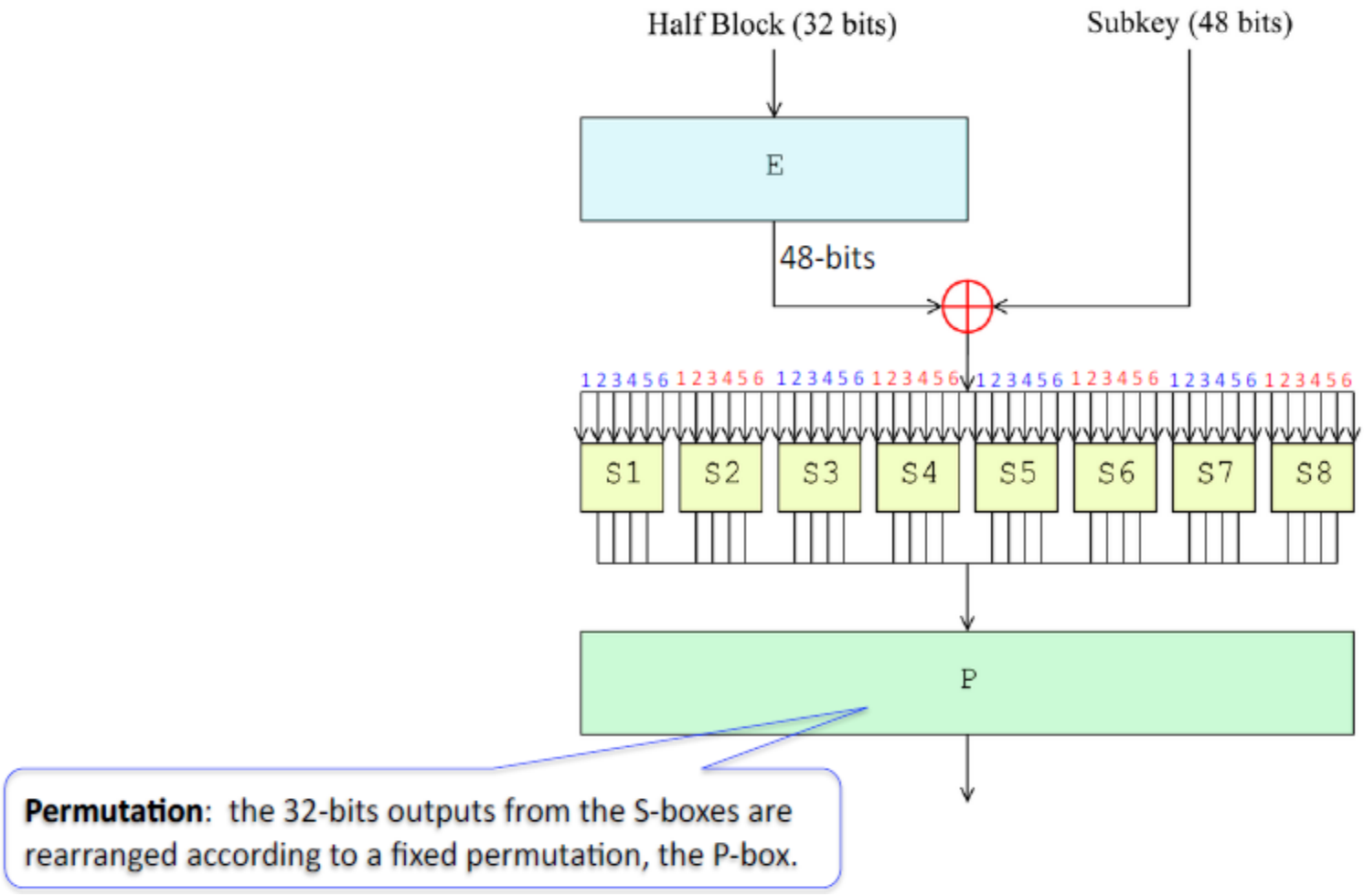


Row **10**
Column **1011**

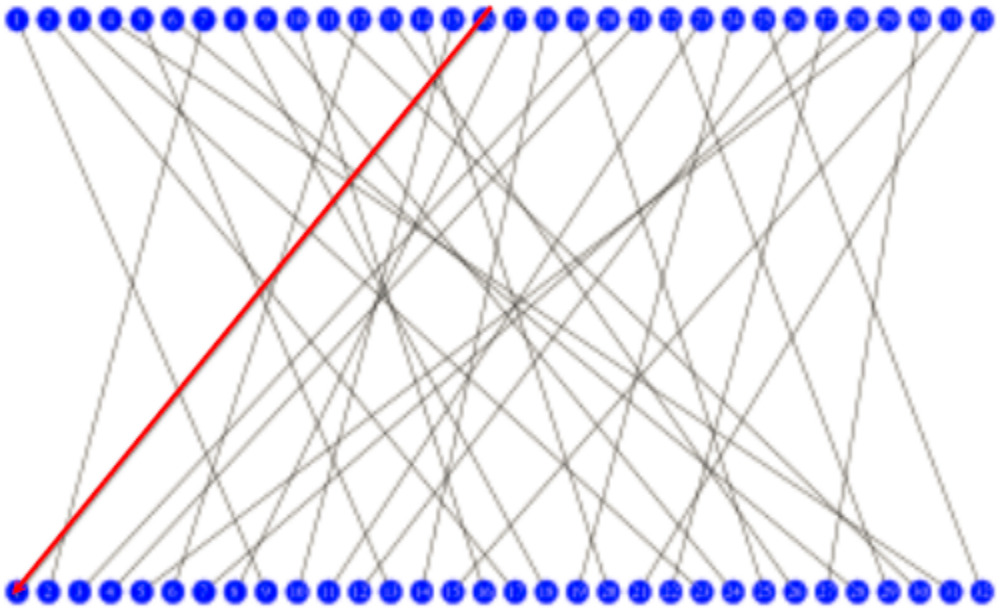
Table for S1

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	6	5	3	8
10	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
11	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

The Round Function F

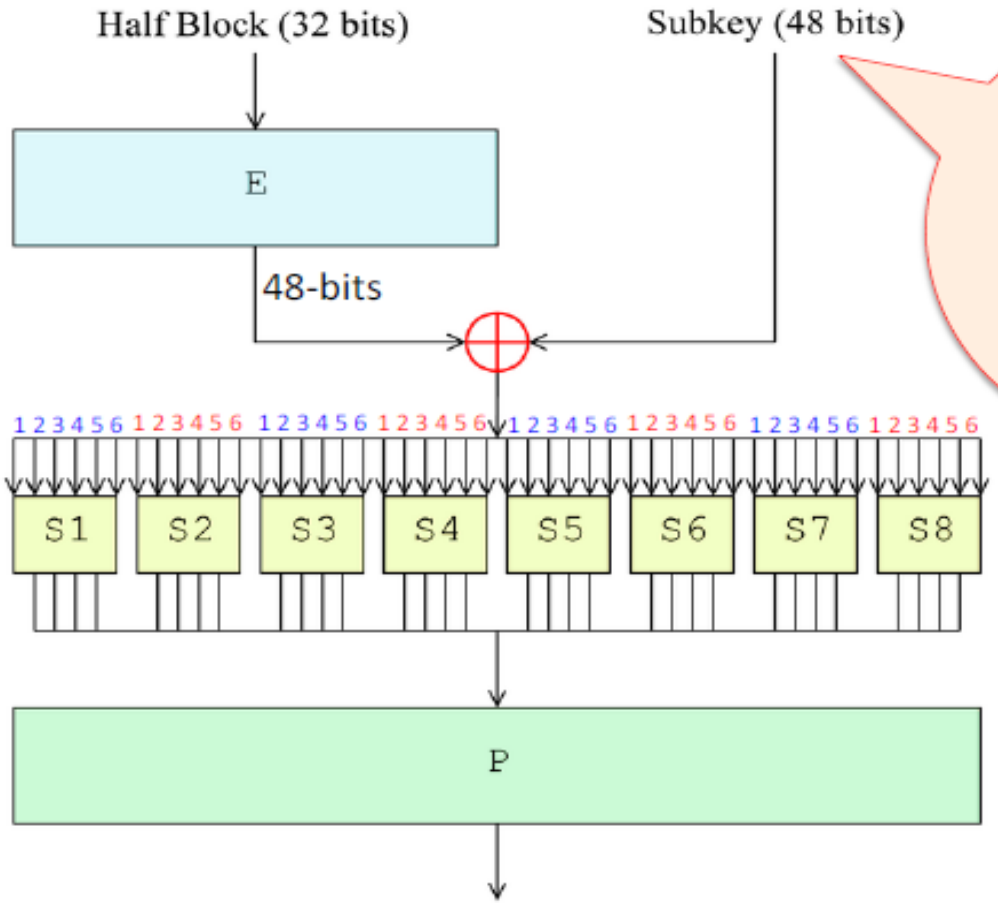


Permutation (**P**)



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

The Round Function F

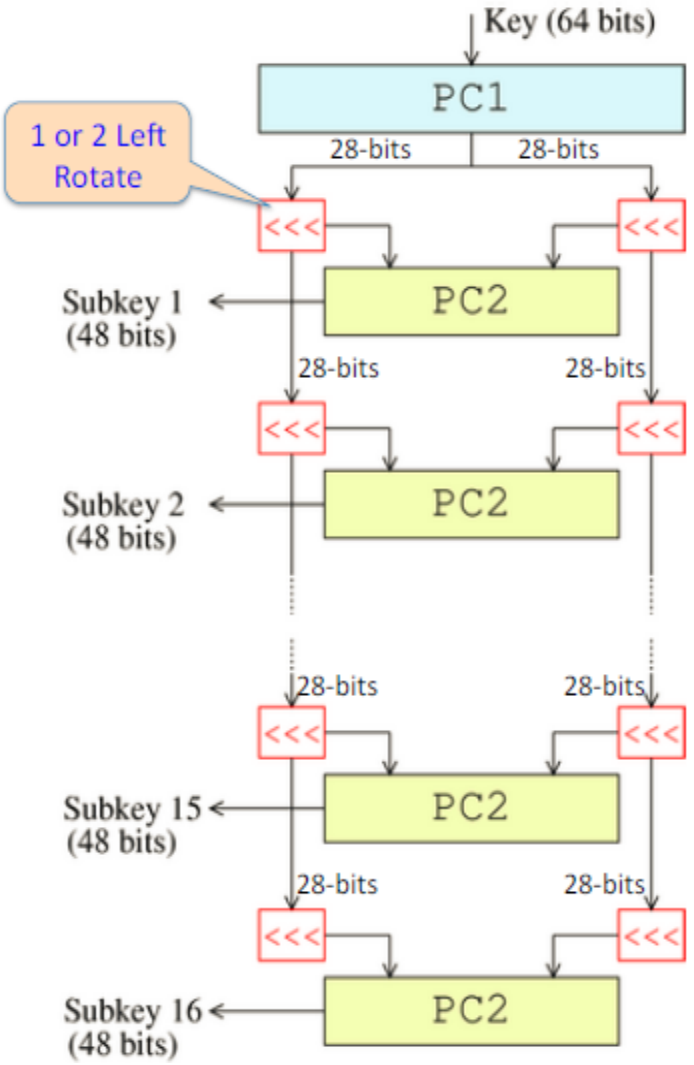


Initially we had a 64-bits key.

Where this 48-bits key came from???

Key Scheduling (Key generation for each of 16 Round)

- 56 bits of the key are selected from the initial 64 by Permuted Choice 1 (PC-1)
- The 56 bits are then divided into two 28-bit halves
- both halves are rotated left by one or two bits
- 48 subkey bits are selected by Permuted Choice 2 (PC-2) — 24 bits from the left half, and 24 from the right.
- The rotations (denoted by "<<<" in the diagram) mean that a different set of bits is used in each subkey



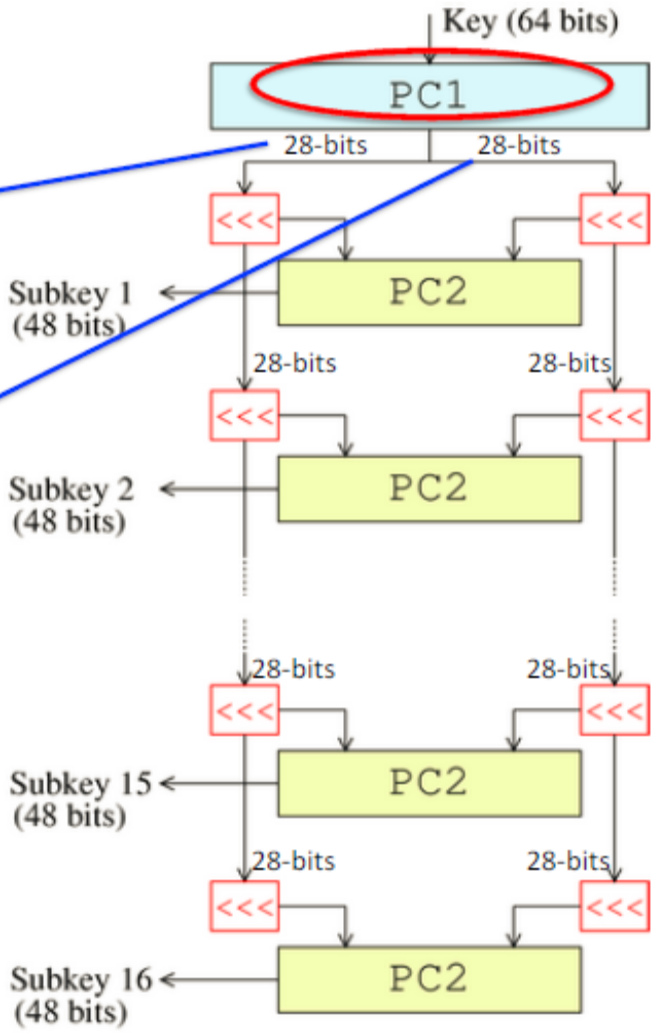
Permuted Choice1 (PC1)

Left side substitution Table for 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

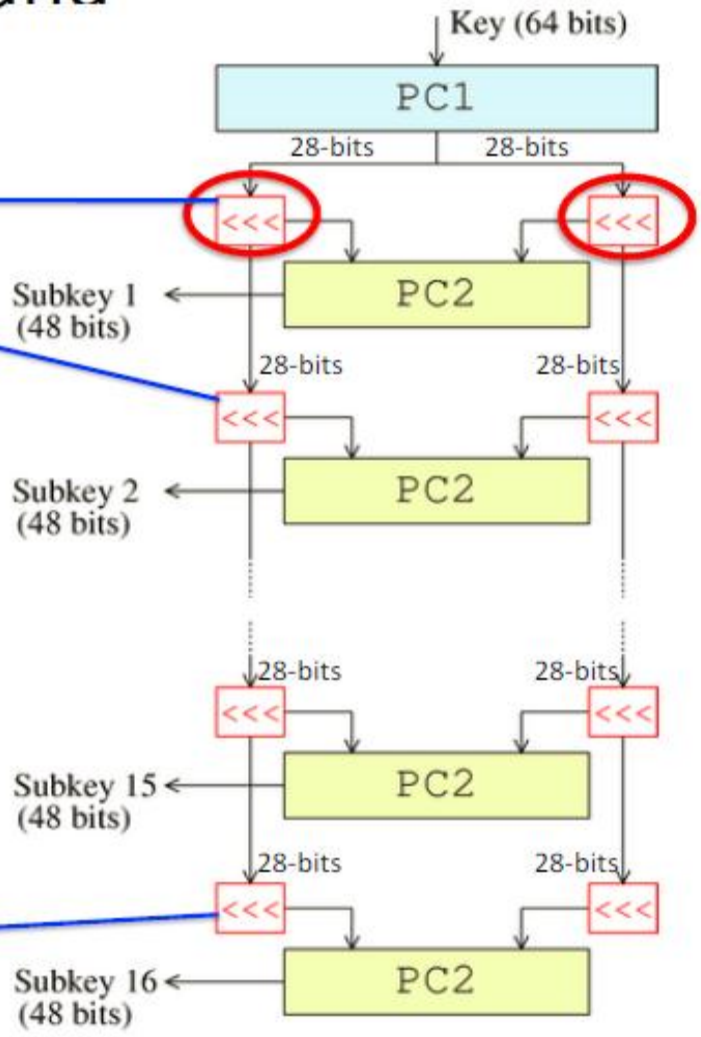
Right side substitution Table for PC1

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



Left Rotation for each round

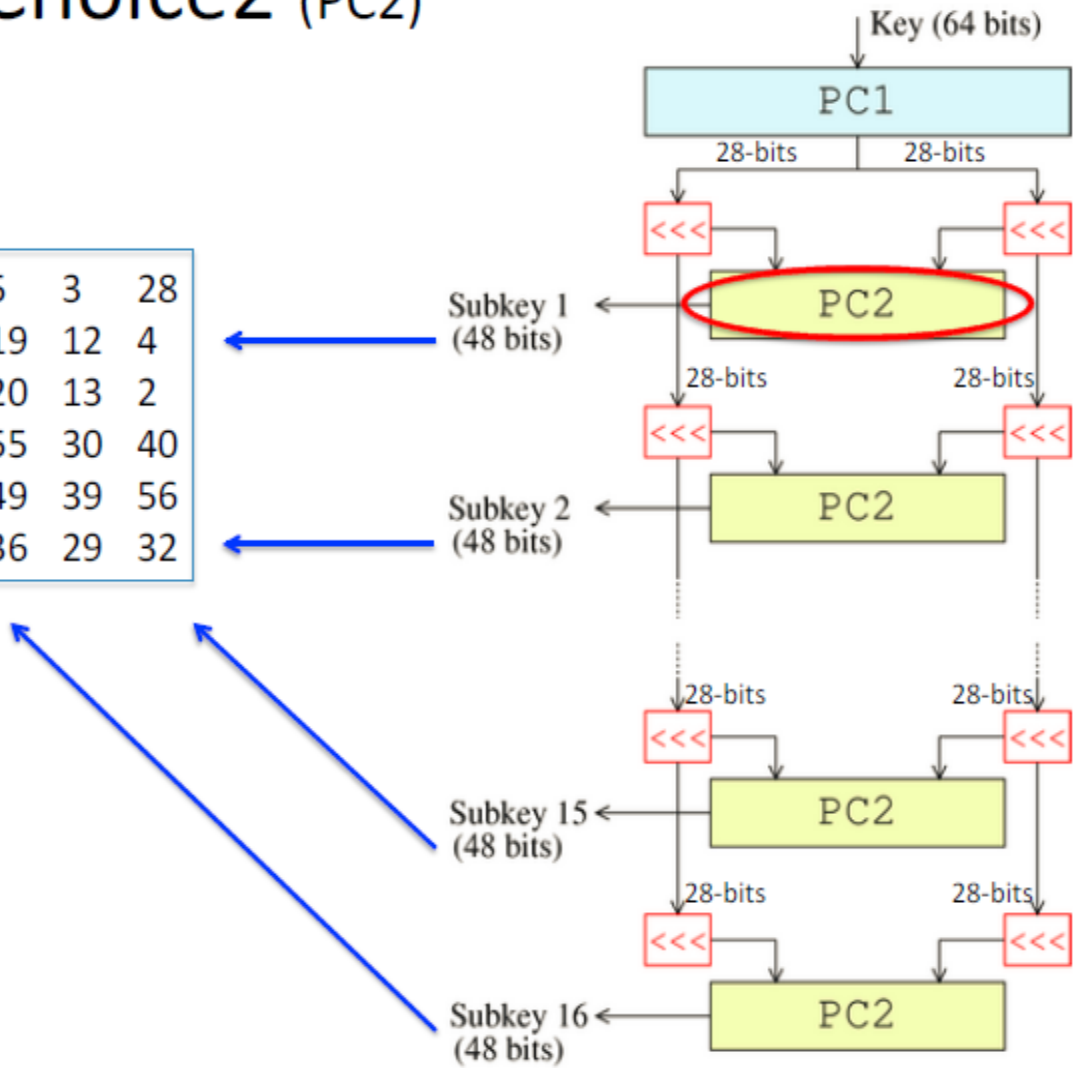
Round No.	No. of Rotations
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1



Permuted Choice2 (PC2)

Table for 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



Avalanche Effect in **DES**

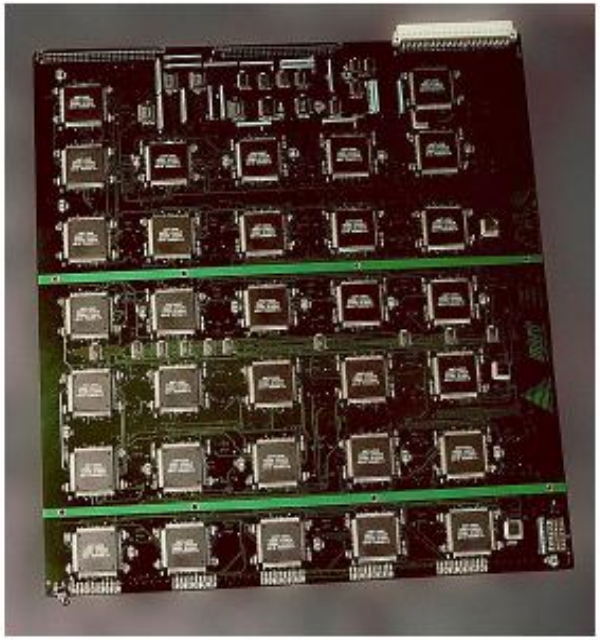
- Minor change in the (plaintext or key) results in a big (reasonable) change in the ciphertext.
- A good cryptosystem with greater **diffusion** and **confusion** results strong Avalanche Effect. (E.g. DES)
  - **1 bit change** in the **plaintext** (on average) **affects 34 bits** in the ciphertext
  - **1 bit change** in the **Key** (on average) **affects 35 bits** in the ciphertext

DES is not secure now!

Key Size (bits)	Number of Alternative Keys	Time required at 1 decryption /μs	Time required at 10^6 decryptions /μs
56-bits	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\ \mu s = 1142$ years	**10.01 hours**

Practical DES Crackers

- The first DES Challenge was solved in 96 days by the DESCHALL Project led by Rocke Verser in Colorado.
- EFF DES cracker won RSA Laboratory's "**DES Challenge II-2**" by successfully finding a DES key in 56 hours.
  - (nicknamed "Deep Crack")
  - 29 circuit boards of 64 chips each
  - Cost: US$250,000
  - Key search speed over 90 billion keys / second



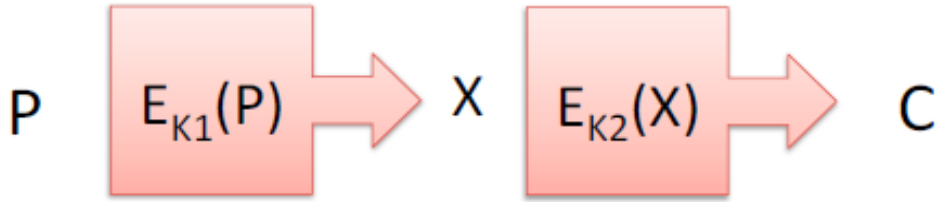
two-sided DES Cracker circuit board fitted with 64 Deep Crack chips

Finding the solution

- We simply need to increase the key size of DES to defend a brute force attack
- Is it worth to apply DES twice (2-Des) or thrice (3-DES)?
- Applying DES twice with two keys (K_1 and K_2) 56-bits each

  $\text{Encryption: } C = E_{K2}(E_{K1}(P))$

  $\text{Decryption: } P = D_{K1}(D_{K2}(C))$
- Key size: 56 x 2 = 112 bits = 2^{112} different keys
- Apparently much secure against brute force attack

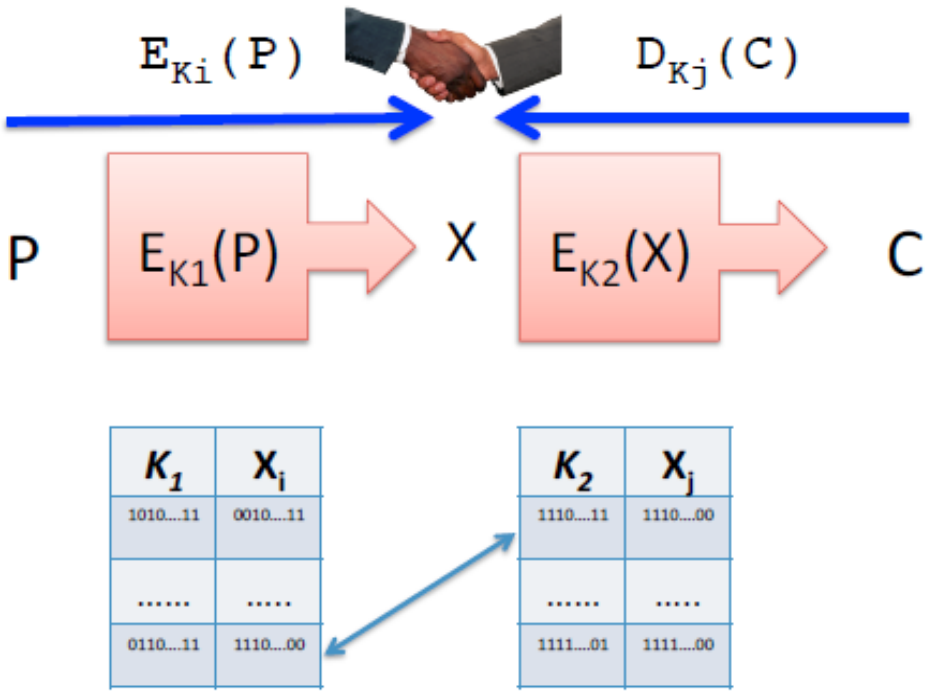


Are we really safe now ???

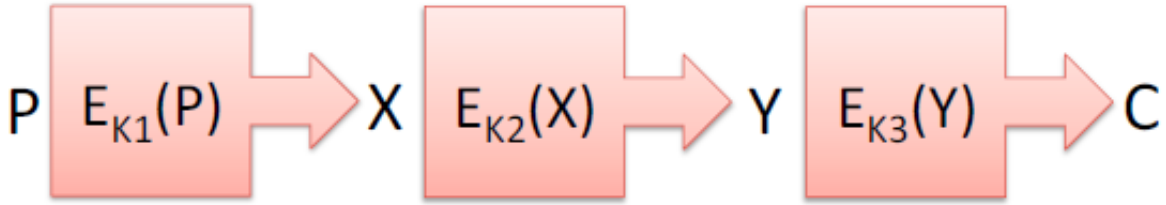
Verifying the security of 2DES

- Walk in the opposite directions and Meet in the Middle
- Assumption: Given a know (plaintext, ciphertext) sample encrypted with the same Key

1. Encrypt P with all 2^{56} possible keys and store all X's in a table
2. Decrypt C with all 2^{56} possible keys and store all X's in another table
3. Try find the exact mach of X's in the two tables
4. If a match found then use the corresponding keys as K1 and K2 to decrypt the actual ciphertext
5. If works then it takes $O(2^{56})$ to break 2-DES which is nearly same as breaking the simple DES



Trying 3 DES



- We simply add another DES round in 2DES making it 3DES
- Is it worth to apply DES thrice (3-DES)?
- Meet-in-the-Middle attack will extremely difficult if X and Y are not known

$$\text{Encryption: } C = E_{K3}(E_{K2}(E_{K1}(P)))$$

$$\text{Decryption: } P = D_{K1}(D_{K2}(D_{K3}(C)))$$

- Key size: 56 x 3 = 168 bits = 2^{168} different keys
- Apparently much secure against brute force attack (currently no practical attack is know)
- PGP and S/MIME (internet applications) use 3DES with 3-keys

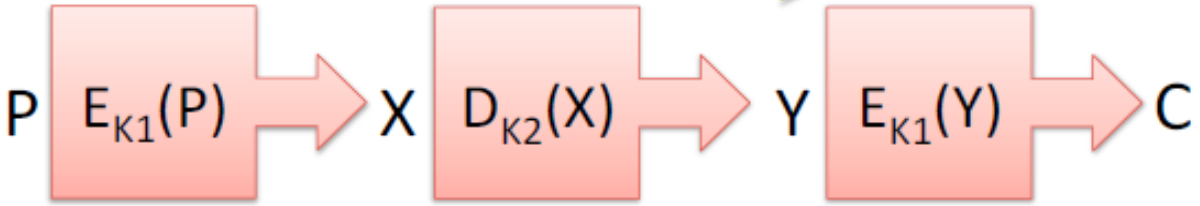
Different Key Options

- If all keys are same then 3DES = DES

$$\text{Encryption: } C = E_{K1}(D_{K1}(E_{K1}(P))) = E_{K1}(P)$$

$$\text{Decryption: } P = D_{K1}(C)$$

No practical attacks are known on 2-keys implementation



- If using two different keys as shown above then

$$\text{Encryption: } C = E_{K1}(D_{K2}(E_{K1}(P)))$$

$$\text{Decryption: } P = D_{K1}(E_{K2}(D_{K1}(P)))$$

  - Still very secure (harder to break because of **forward-reverse-forward** process)
  - Key size = 112 → 2^{112} Key space

Block Ciphers Modes of Operation

- The way in which the block cipher scheme is implemented

1. ECB (**E**lectronic **C**ode **B**ook)
2. CBC (**C**ipher **B**lock **C**haining)
3. PCBC (**P**ropagating **C**ipher **B**lock **C**haining)
4. CFB (**C**ipher **F**eedback)
5. OFB (**O**utput **F**eedback)
6. CTR (**C**ounter)

Assignment #2

- CLASSIFY different Block Cipher modes of Operations for DES algorithms. Highlight their main area of applications and advantages and disadvantages of each method. [CLO-2] [C3].
-