Designing Block Ciphers --- DES

B Tech III - ISC
Dhiren Patel
(March 2024)

Designing Block Ciphers

- Kerckhoff’s principle: The secrecy should be in the key, not in the encryption/decryption algorithms!!!
- Cipher (Encryptor/Decryptor)
- SKC and PKC

SKC – Symmetric Key Cryptography

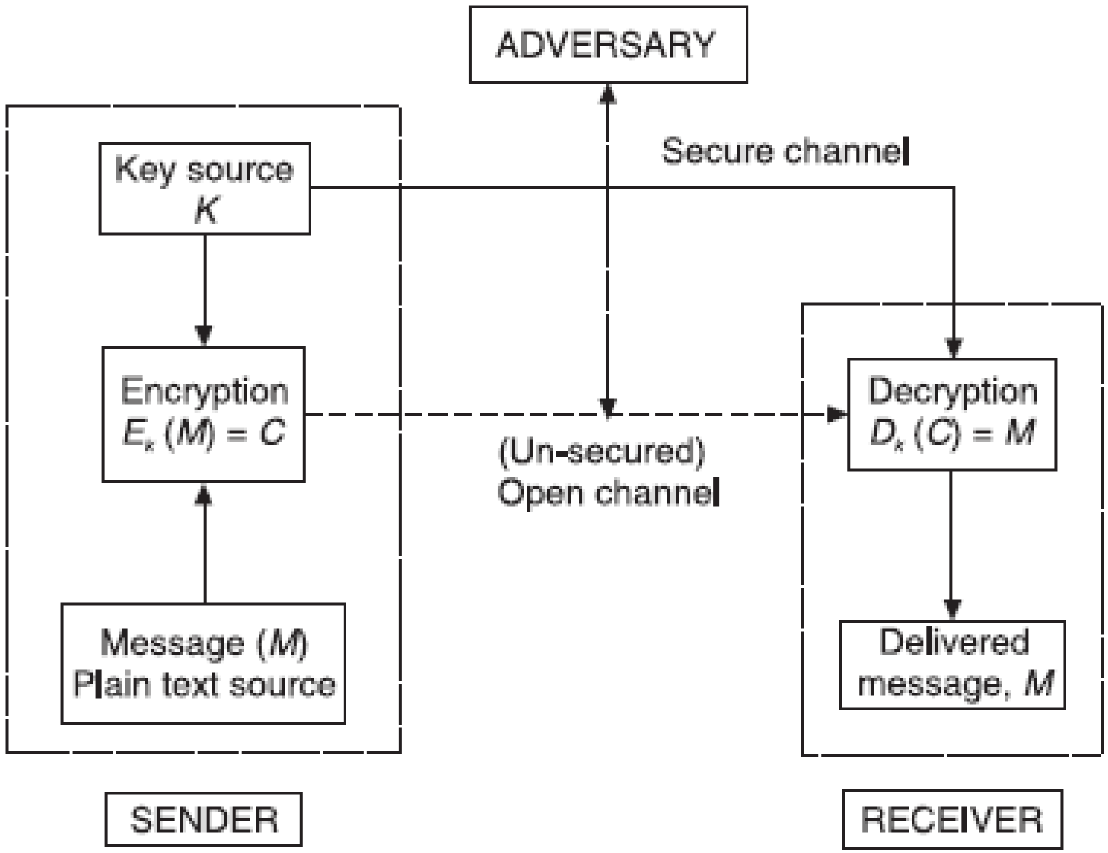


Figure 3.1 Communication using symmetric key cryptography ($k = k_1 = k_2$).

BLOCK CIPHER MODEL - SKC

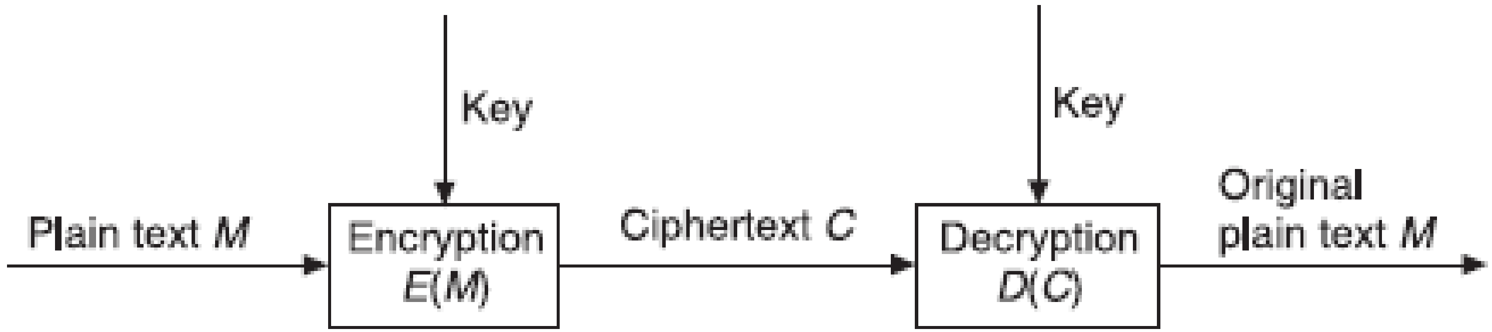


Figure 1.1 Encryption and decryption.

- Encryption algorithm/decryption algorithm
- Secret Key(s)

Cipher – other properties

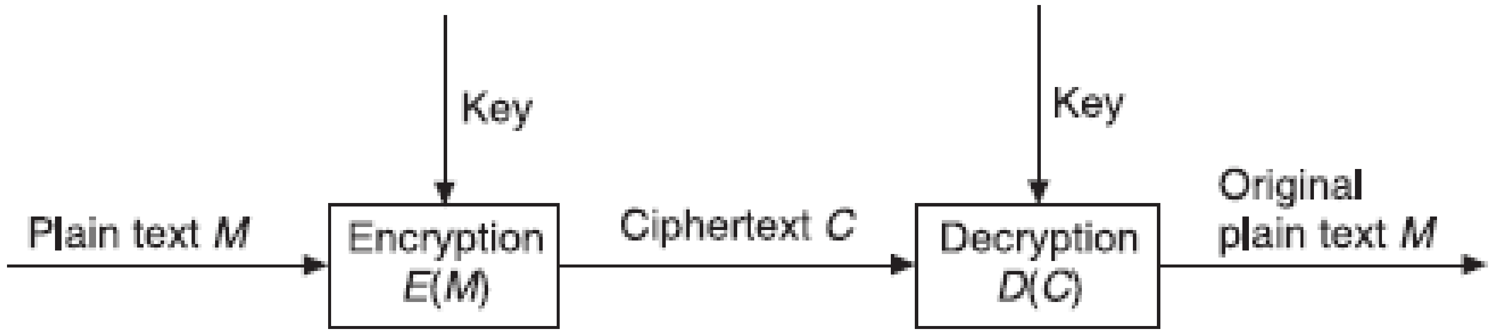


Figure 1.1 Encryption and decryption.

- Encryption algorithm/decryption algorithm
- Secret Key
- Block size <no of plain text bits encrypted at a time> (64, 128, ….)
- Key size <number of bits in key> (64, 128, 192 …)
- Confusion and Diffusion (Claude Shannon – 1949)

Confusion

- No clue regarding relationship between the cipher text and the key
- a single bit change in key - changes roughly half of the bits in corresponding cipher text, moreover positions of changed (flipped) bits are random!
- Substitution enhances confusion!!

Diffusion

- Concerned with the relationship between the plain text and the corresponding cipher text.
- Changing a single bit in a block of plain text will have the effect of changing each bit of cipher text with probability of 0.5
- However, this changes are scattered across the block of cipher text.
- between plain text and cipher text – no statistical relation…
- Transposition enhances diffusion!!!

Block Cipher Design

- A block cipher (is a function which maps) n-bit plaintext blocks to n-bit ciphertext blocks; n is called the *block length*.
  - E: $\{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$
- Use of plaintext and ciphertext blocks of equal size avoids data expansion.
- The function is parameterized by a k-bit key.

Block Cipher Design

- To allow unique decryption, the encryption function must be one-to-one (i.e., invertible)
- For n-bit plaintext and ciphertext blocks and a fixed key, the encryption function is a bijection (1-to-1 and on-to), defining a permutation on n-bit vectors.
- Smaller block size may be vulnerable to attacks based on statistical analysis.

Product cipher

- Combine both substitution (confusion) and transposition <permutation> (diffusion)
- Principal design template for symmetric block ciphers

Substitution box – S box

- Takes as input a string (binary) of length m
- Returns a string of length n
- Implemented using a table (array) of 2^m rows each containing an n-bit value.
- Input to S-box is used to index the table which returns n-bit output of S-box.
- Usually m = n (need not always)
- In DES (Data Encryption Standard); m > n.

Data Encryption Standard (DES)

- DES is a symmetric-key block cipher. This means it uses the same secret key for both encryption and decryption, and it encrypts data in fixed-sized blocks (64 bits in DES)
- Structure: DES is based on a Feistel network, a design commonly used in block ciphers. This network involves multiple rounds of substitutions and permutations to scramble the data and make it unintelligible without the key.

DES

- Key Length: One of the weaknesses of DES is its key length. It uses a 56-bit key, with 8 additional parity bits that are discarded during encryption/decryption.
- This short key length makes it vulnerable to brute-force attacks, where attackers can try every possible combination to crack the encryption.
- (Export Regulations (1977-1985) – outside USA; allowed key length was only 40 bits

e.g. 6×4-bit S-Box from DES (S_5)

S_5		Middle 4 bits of input															
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Outer bits	**00**	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

- selecting the row using the outer two bits (the first and last bits), and the column using the inner four bits.
- For example, an input "**0**1101**1**" has outer bits "**01**" and inner bits "1101"; the corresponding output would be "1001".

S-box

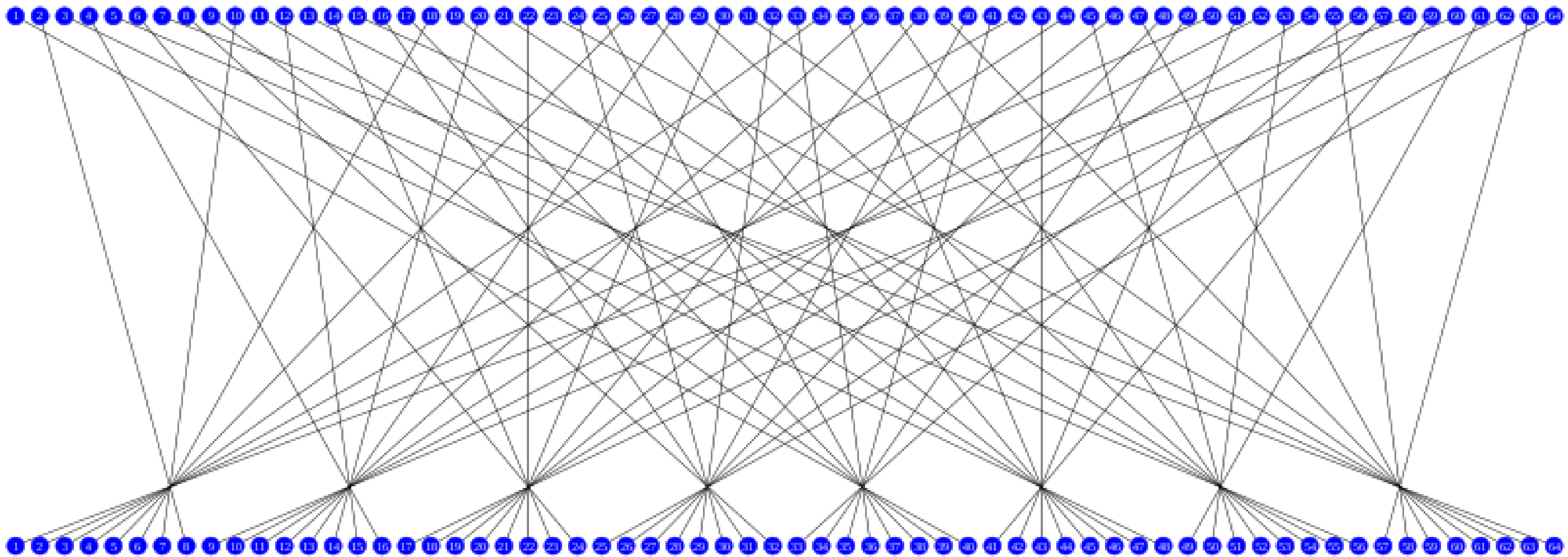
- Injects non-linearity into the design of cipher
- Absence of a linear relationship between any subset of bits in the plain text, cipher text, and the key.

Permutation box - P box

- Performs permutation or rearrangement of the bits in the input
- E.g. IP (initial permutation in DES – 64 bit), there is a 32bit P also!
- The meaning is as follows: the first bit of the output is taken from the 58th bit of the input; the second bit from the 50th bit, and so on, with the last bit of the output taken from the 7th bit of the input. This information is presented as a table for ease of presentation; it is a vector.

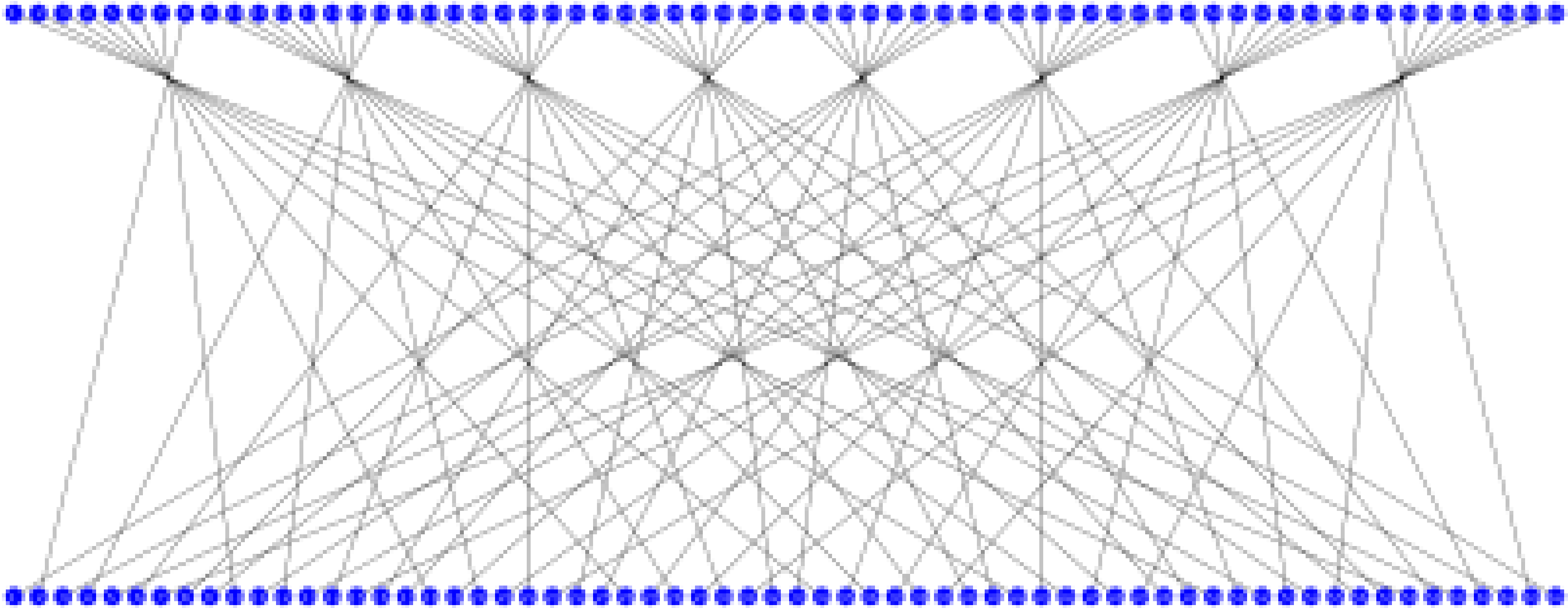
IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP in DES (P-box)



$FP = IP^{-1}$

- The final permutation is the inverse of the initial permutation (64-bit)

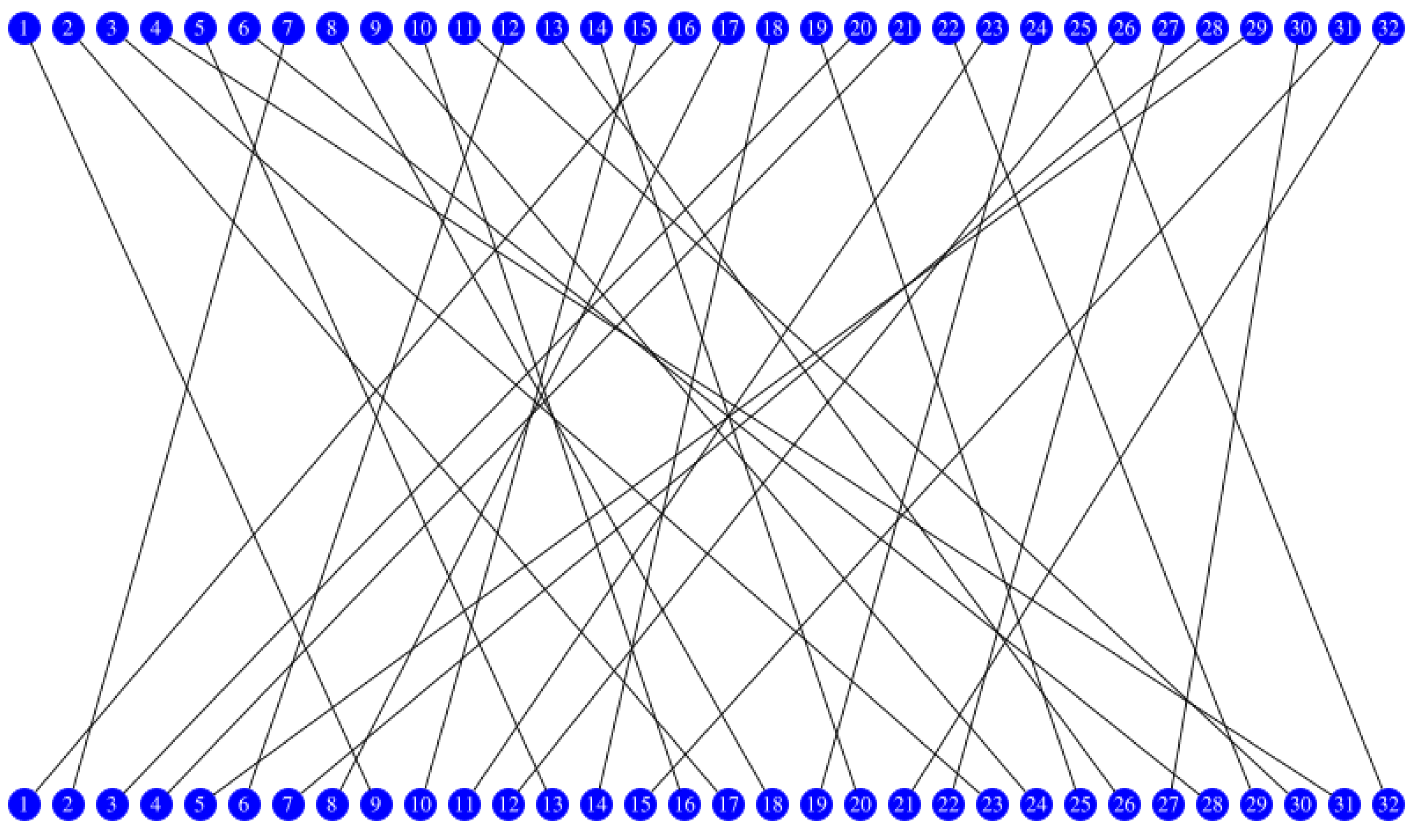


P-Box

- Diffuses or spreads contiguous bits of the input across the block
- Removing local effect i.e. certain bits of the output would not be a certain bits of input

P (different than IP)

- The P permutation shuffles the bits of a 32-bit half-block.



Fiestel Structure – DES SPN

- Hornst Fiestel – IBM (key designer of DES)

DES

- SPN – Substitution Permutation Network
- Block size of 64-bits
- Key – 56 bits (+8 bits parity check of each byte)
- Initial Permutation, Rounds, left-right swap, Final Permutation, Fiestel structure

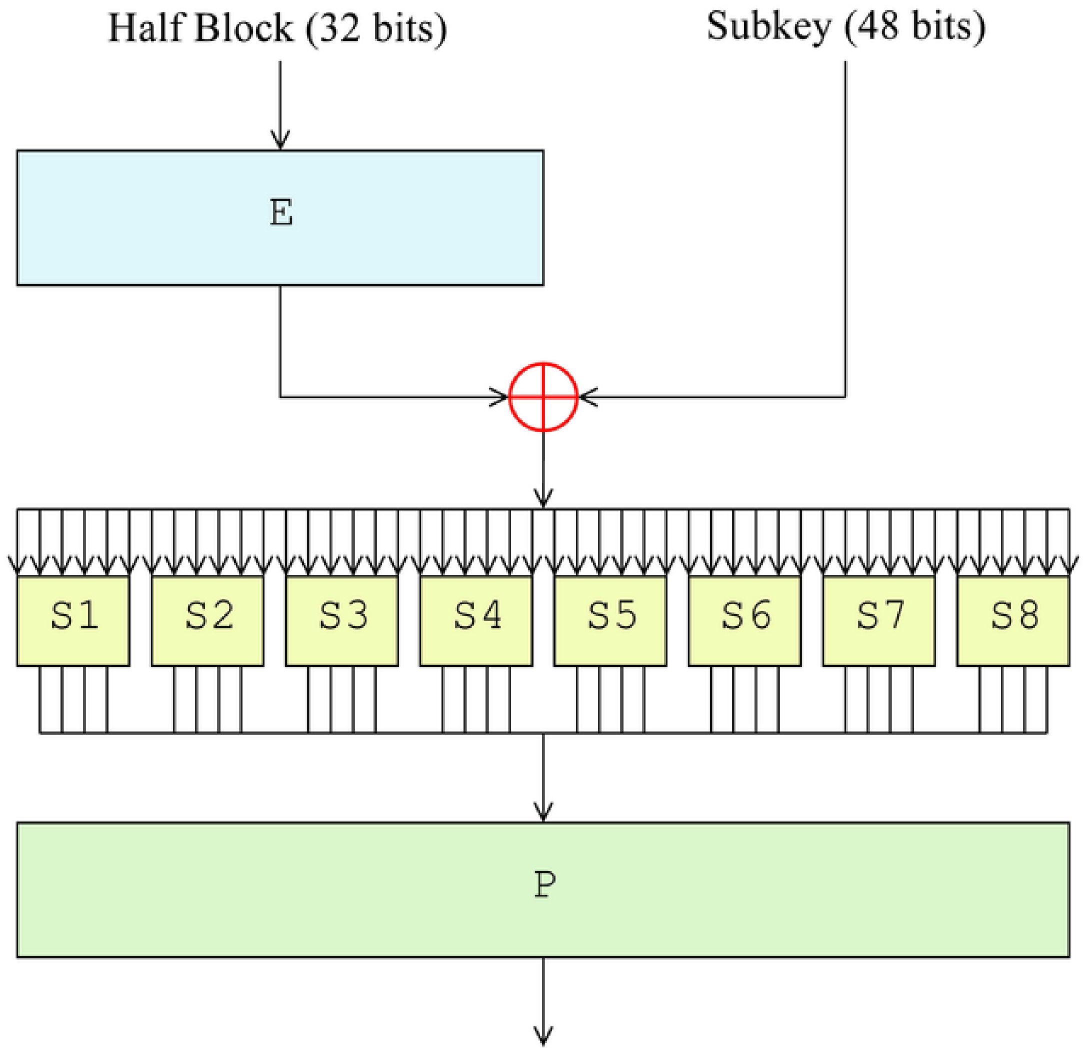
DES

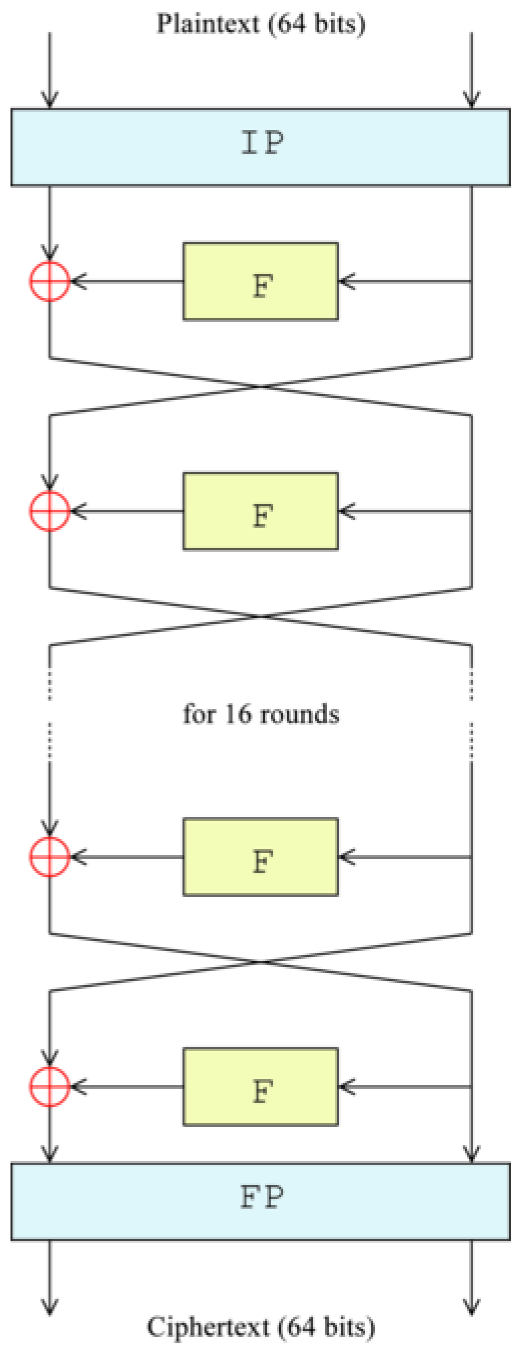
- there are 16 identical stages of processing, termed *rounds*.
- an initial and final permutation, termed *IP* and *FP*
- *F-function*

DES

- DES operates on the 64-bit blocks using *key* sizes of 56- bits. (+8 parity bits)
- Each block of 64 bits is divided into two blocks of 32 bits each, a left half block **L** and a right half **R**.
- Generate 16 subkeys from given key.
- Go through 16 iterations (Rounds)
- $\boldsymbol{L_n = R_{n-1}}$
  $\boldsymbol{R_n = L_{n-1} + f(R_{n-1}, K_n)}$

DES SPN




Plaintext (64 bits)
IP
F
F
for 16 rounds
F
F
FP
Ciphertext (64 bits)

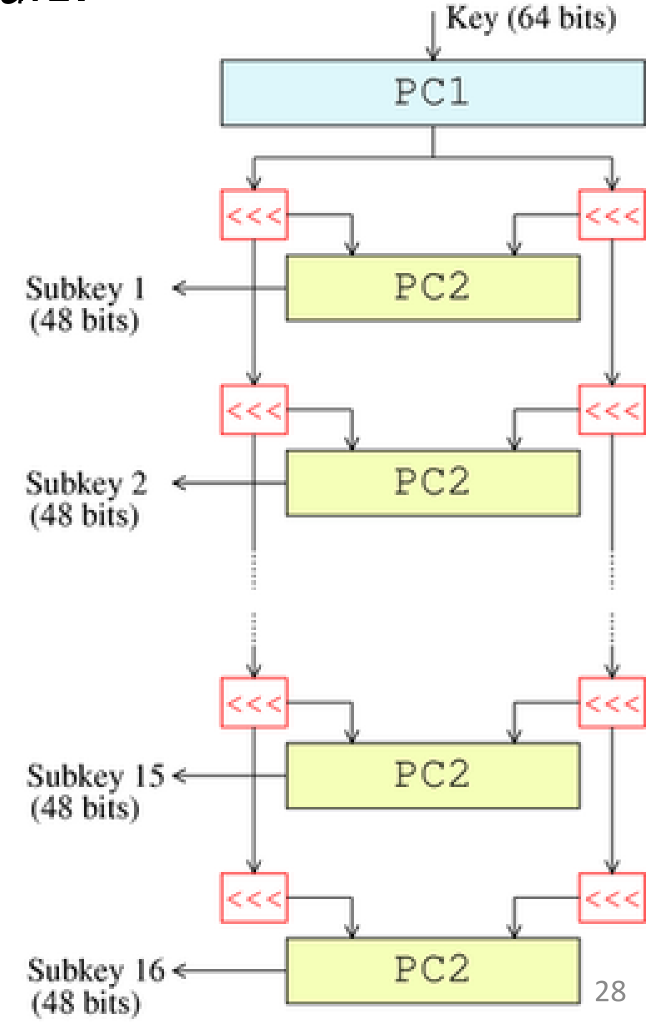
Key permutation (64 → 56)

The 64-bit key is permuted to generate 56-bit key. (It is to note that bit 08, 16, 24, 32, 40, 48, 56, 64 are ignored)

57	**49**	**41**	**33**	**25**	**17**	**09**
01	**58**	**50**	**42**	**34**	**26**	**18**
10	**02**	**59**	**51**	**43**	**35**	**27**
19	**11**	**03**	**60**	**52**	**44**	**36**
63	**55**	**47**	**39**	**31**	**23**	**15**
07	**62**	**54**	**46**	**38**	**30**	**22**
14	**06**	**61**	**53**	**45**	**37**	**29**
21	**13**	**05**	**28**	**20**	**12**	**04**

Key schedule of DES

- 16 nos. of 48-bit subkeys — one for each round — are derived from the main key using the *key schedule*.



Key schedule

- Next, split this key into left and right halves, C_0 and D_0, where each half has 28 bits.
- With C_0 and D_0 defined, create sixteen blocks C_n and D_n, 1<=n<=16.
- Each pair of blocks C_n and D_n is formed from the previous pair C_{n-1} and D_{n-1}, respectively using the following left shift schedule.
- C_3 and D_3 are obtained from C_2 and D_2, respectively, by two left shifts, and C_{16} and D_{16} are obtained from C_{15} and D_{15}, respectively, by one left shift

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Key schedule - **PC-2**

- Form the keys (48-bit) $\boldsymbol{K_n}$, for 1<=$\boldsymbol{n}$<=16, by applying the following permutation table to each of the concatenated pairs $\boldsymbol{C_n D_n}$. Each pair has 56 bits, but **PC-2** only uses 48 of these.

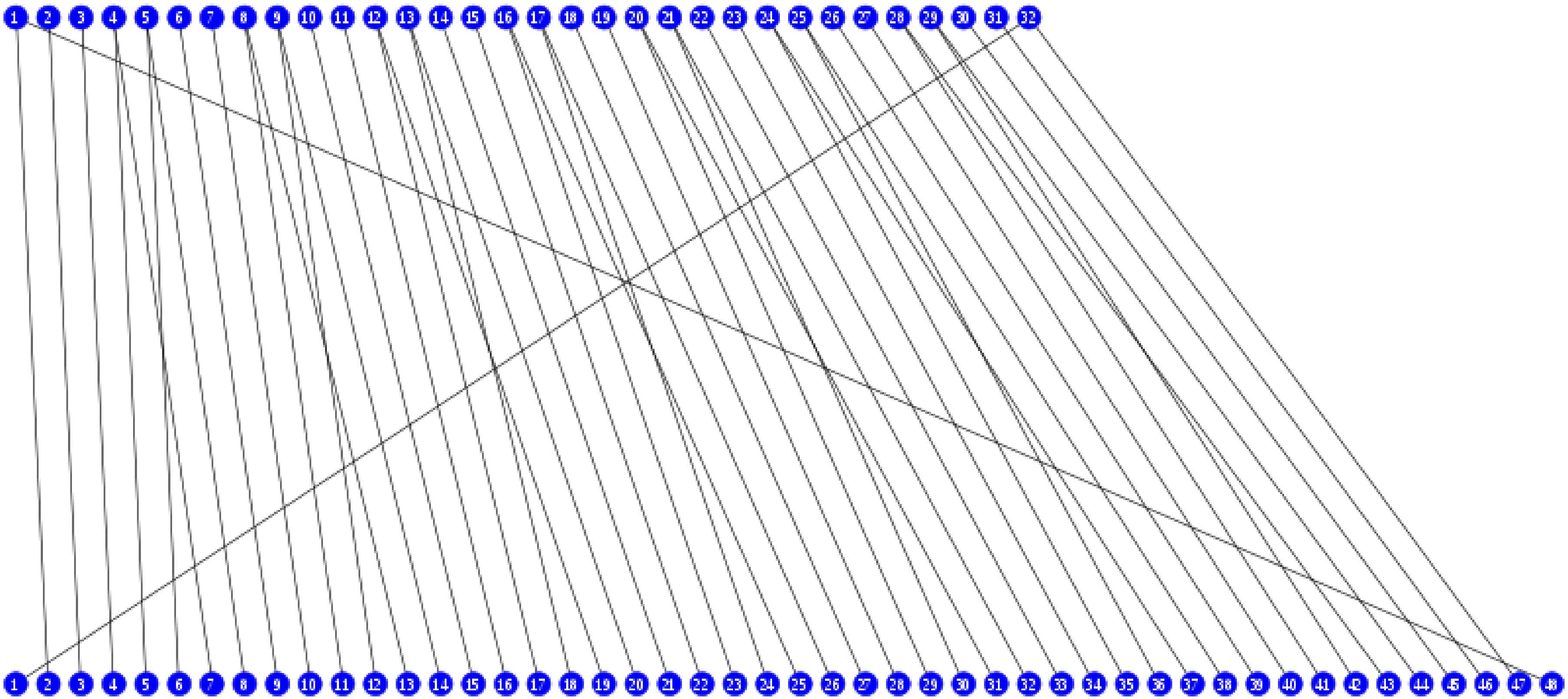
14	**17**	**11**	**24**	**01**	**05**
03	**28**	**15**	**06**	**21**	**10**
23	**19**	**12**	**04**	**26**	**08**
16	**07**	**27**	**20**	**13**	**02**
41	**52**	**31**	**37**	**47**	**55**
30	**40**	**51**	**45**	**33**	**48**
44	**49**	**39**	**56**	**34**	**53**
46	**42**	**50**	**36**	**29**	**32**

F -function

- operates on half a block (32 bits) at a time and consists of four stages:
- —The Feistel function (F-function) of DES
- *Expansion* — the 32-bit half-block is expanded to 48 bits using the *expansion permutation*, denoted *E* in the diagram, by duplicating half of the bits. The output consists of eight 6-bit(8*6=48bits) pieces, each containing a copy of 4 corresponding input bits, plus a copy of the immediately adjacent bit from each of the input pieces to either side.

E - function

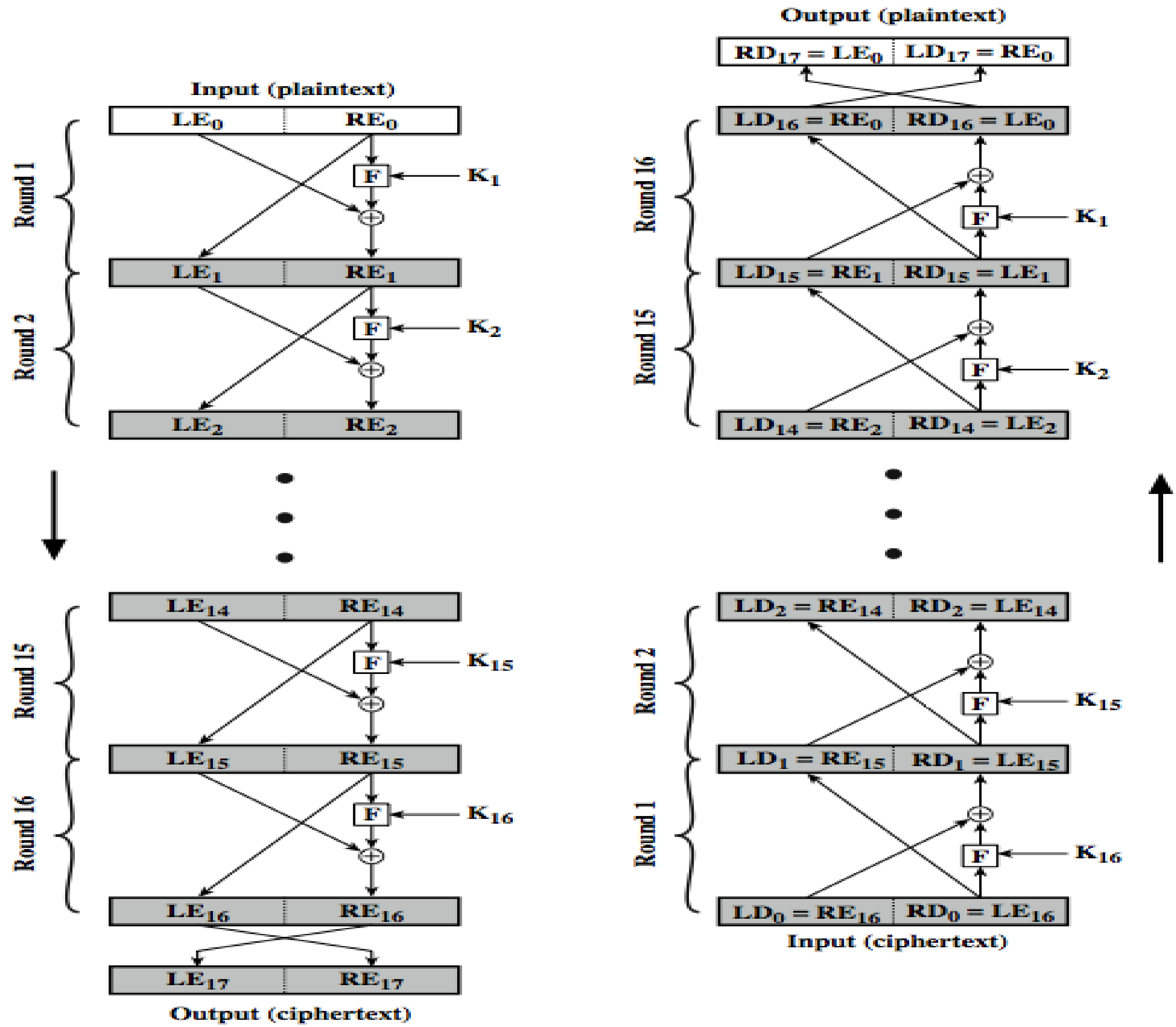
- Some bits from the input are duplicated at the output; e.g. the fifth bit of the input is duplicated in both the sixth and eighth bit of the output. Thus, the 32-bit half-block is expanded to 48 bits.



F-function

- *Substitution* — after mixing in the subkey, the block is divided into eight 6-bit pieces before processing by the *S-boxes*.
- Each of the eight S-boxes replaces its six input bits with four output bits according to a non-linear transformation, provided in the form of a lookup table.
- The S-boxes provide the core of the security of DES — without them, the cipher would be linear, and trivially breakable.
- *Permutation* — finally, the 32 outputs from the S-boxes are rearranged according to a fixed permutation, the *P-box*.
- This is designed so that, after expansion, each S-box's output bits are spread across 6 different S boxes in the next round.

Feistel Cipher Structure (DES – encryption and decryption)



Security of block ciphers

- The objective of a block cipher is to provide confidentiality.
- The corresponding objective of an adversary is to recover plaintext from ciphertext.
- The best measure of security for practical ciphers is the complexity of the best known attack. Various aspects of such complexity may be distinguished as follows:
  - Data Complexity
  - Storage Complexity
  - Processing Complexity
- Cost of attack v/s value of information!!!

Security of Block Ciphers

- A block cipher is *totally broken* if a key can be found, and *partially broken* if an adversary is able to recover part of the plaintext (but not the key) from ciphertext.
- To evaluate block cipher security, it is customary to always assume that an adversary
  (i) has access to all data transmitted over the ciphertext channel;
  (ii) knows all details of the encryption function except the secret key

Weak keys in DES

- few specific keys termed "weak keys" and "semi-weak keys".
- These are keys that cause the encryption mode of DES to act identically to the decryption mode of DES (albeit potentially that of a different key).
- four keys are weak and twelve keys are semi-weak
- DES *weak keys* produce sixteen identical subkeys.

Weak keys

- This occurs when the key bits are:
- (0x0101010101010101)
- (0xFEFEFEFEFEFEFEFE)
- (0xE0E0E0E0F1F1F1F1)
- (0x1F1F1F1F0E0E0E0E)
- If an implementation does not consider the parity bits, the corresponding keys with the inverted parity bits may also work as weak keys:
- all zeros (0x0000000000000000)
- all ones (0xFFFFFFFFFFFFFFFF)
- (0xE1E1E1E1F0F0F0F0)
- (0x1E1E1E1E0F0F0F0F)

Weak/Semi-weak keys

- Using weak keys, the outcome of the Permuted Choice 1 (PC1) in the DES key schedule leads to round keys being either all zeros, all ones or alternating zero-one patterns.
- Since all the subkeys are identical, and DES is a Feistel network, the encryption function is self-inverting; that is, encrypting twice produces the original plaintext.

Semi weak keys

- There are six semi-weak key pairs:
- 0x011F011F010E010E and 0x1F011F010E010E01
- 0x01E001E001F101F1 and 0xE001E001F101F101
- 0x01FE01FE01FE01FE and 0xFE01FE01FE01FE01
- 0x1FE01FE00EF10EF1 and 0xE01FE01FF10EF10E
- 0x1FFE1FFE0EFE0EFE and 0xFE1FFE1FFE0EFE0E
- 0xE0FEE0FEF1FEF1FE and 0xFEE0FEE0FEF1FEF1

History - DES

- Late 1960s: IBM develops a cipher called LUCIFER as part of a research project.
- Early 1970s: IBM modifies LUCIFER for commercialization, collaborating with the National Security Agency (NSA) on some aspects of the design. This modified version is submitted as a proposal for a national encryption standard.
- 1972: The National Bureau of Standards (NBS) recognizes the need for a government-wide standard for encrypting unclassified sensitive information.
- 1976 (November): DES, based on the modified LUCIFER, is finally adopted as the Federal Information Processing Standard (FIPS PUB 46) by the NBS. There were concerns about the NSA's influence on the design, with some suspecting a backdoor, but these were never proven.
- 1977 (January): DES is officially published as a standard - FIPS 46.

History - DES

- 1980s and 1990s: DES enjoys widespread adoption due to its government backing. However, concerns about its key length (56 bits) begin to emerge as computing power increases.
- 1999: A variant called Triple DES (using three DES encryptions) is introduced in FIPS PUB 46-3 to address key length limitations.
- 2002: DES is officially superseded by the Advanced Encryption Standard (AES) after a public competition. AES offers a longer key length and stronger security.
- 2005: FIPS PUB 46-3 (including Triple DES) is withdrawn, although Triple DES remains approved for specific government uses until 2030.

DES

- Short key length makes DES vulnerable to brute-force attacks, where attackers can try every possible combination to crack the encryption.
- (Export Regulations (1977-1985) – outside USA; allowed key length was only 40 bits
- AES (Advanced Encryption Standard) have replaced DES for most applications.