

The background features a gradient from dark grey on the left to light blue on the right. Overlaid on this are various 3D-rendered numbers in shades of grey and blue, some appearing to float or be attached to the surface. A solid blue horizontal line is positioned in the upper left quadrant.

# Block Cipher

---

# Block cipher

- Plaintext and ciphertext consist of fixed-sized blocks
- Ciphertext obtained from plaintext by iterating a **round function**
- Input to round function consists of *key* and *output* of previous round
- Usually implemented in software




# Feistel Cipher

- Type of block cipher not a specific type
- Structure:
  - Text
  - subkeys

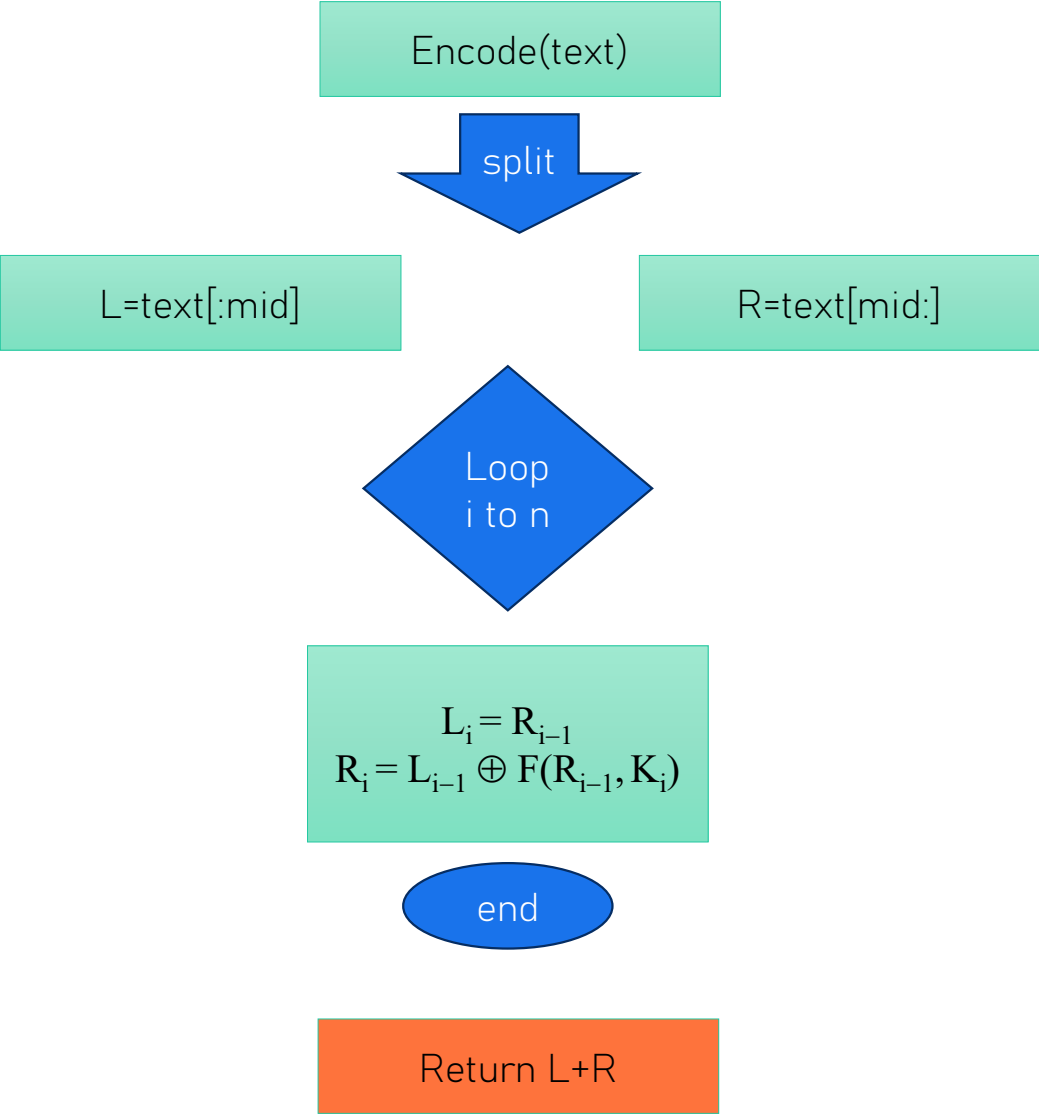
---

# Feistel Cipher

- Encryption:
  - Generate subkeys
  - Encode text
  - Split the plaintext into left and right halves:  $P = (L_0, R_0)$
  - Make rounds (loop) at each round  $i = 1, 2, \dots, n$ , compute
    - $L_i = R_{i-1}$
    - $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$  where  $F$  round function make xor
  - Ciphertext is  $L+R$
  - Return ciphertext

- 
- decryption:
    - Split the ciphertext into left and right halves:  $\mathbf{P} = (\mathbf{L}_0, \mathbf{R}_0)$
    - Make rounds (loop) at each round  $i = n, n-1, \dots, 1$ , compute
      - $\mathbf{L}_i = \mathbf{R}_{i-1}$
      - $\mathbf{R}_i = \mathbf{L}_{i-1} \oplus F(\mathbf{R}_{i-1}, \mathbf{K}_i)$  where  $F$  round function make xor
    - plaintext is  $\mathbf{L} + \mathbf{R}$
    - Return plaintext

—



---

# DES data encryption standard

## Structure

- 8 byte text
- 8 byte key
- Initial permutation IP (text)
- Final or reverse permutation FP (text)
- Expand permutation EP (right half text)
- PC1, PC2 (key)
- S POX permutation (EP result)
- P permutation (S POX result)

---

# DES

Steps:

- Encrypt:
  - Convert text to bits
  - Generate 16 subkey from key
  - Split text to L, R
  - 16 round i: 1 to 16 each round:
    - Feistel to R and subkey I
    - Calc new L,R
  - Combine L, R
  - Return combined result



---

# DES decrypt

Steps:

- Decrypt:
  - Generate 16 subkey from key
  - Split text to L, R
  - 16 round i: 16 to 1 each round:
    - Feistel to R and subkey I
    - Calc new L,R
  - Combine L, R
  - Return combined result