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

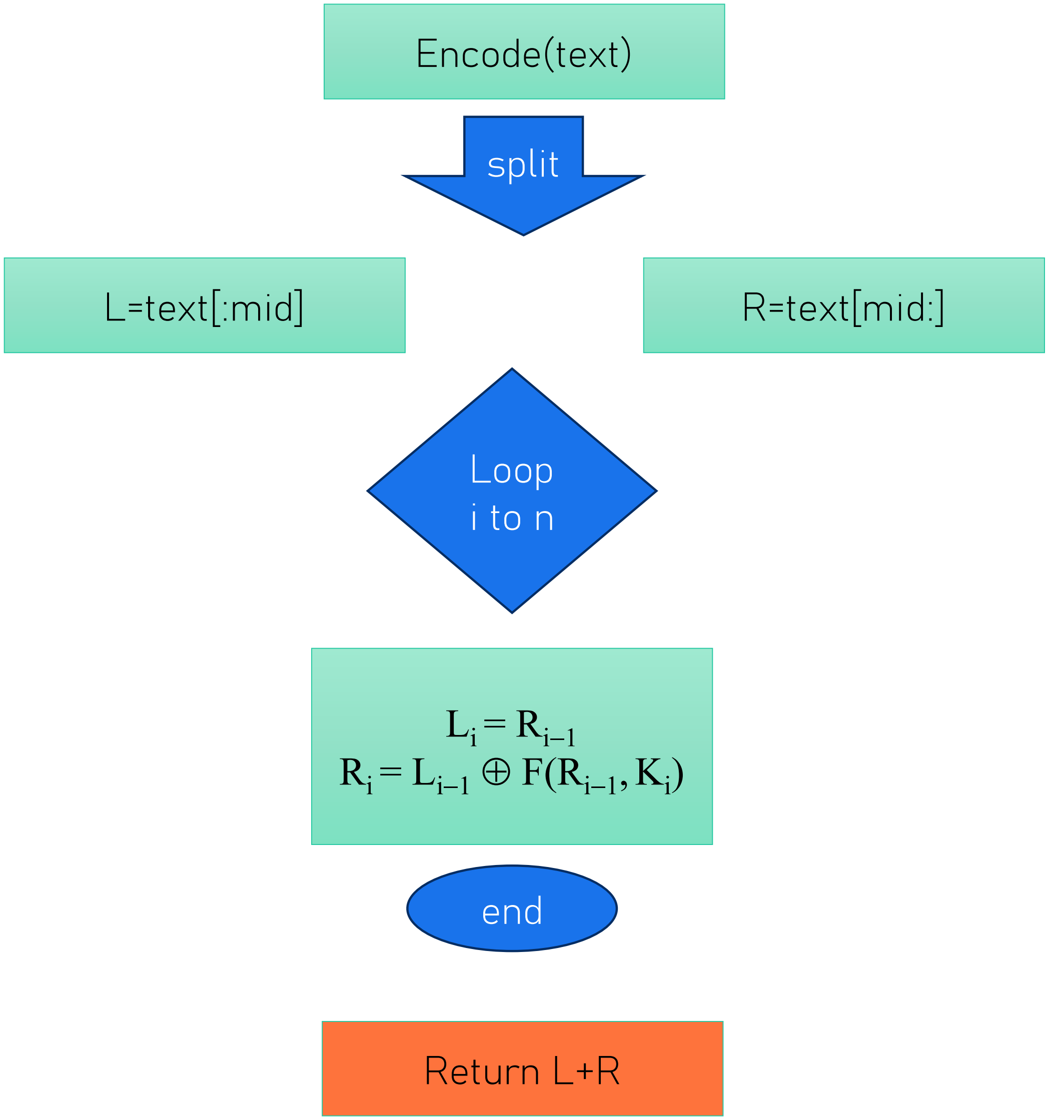
Feistel Cipher

- 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 $i = 1, 2, ..., n$, 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: P = (L_0, R_0)
    - Make rounds (loop) at each round $i = n, n\text{-}1, \ldots, 1$, compute
        - $L_i = R_{i-1}$
        - $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$ where F round function make xor
    - plaintext is L+R
    - Return plaintext



Encode(text)

split

L=text[:mid]

R=text[mid:]

Loop

i to n

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

end

Return L+R

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
  - Permute IP (initial P)
  - Generate 16 subkey from key
  - Split text to L, R
  - 16 round i: 1 to 16 each round:
    - $L_i = R_{i-1}$
    - $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$ where F round function make xor
  - Combine L, R
  - Permute FP (final or IN_INV P)

- Round function $F(R_{i-1}, K_i)$ steps:
  - Permute EP (expand P)
  - Xor R with key
  - Execute S pox
  - Permute P permutation

DES

Steps:

- decrypt:
  - Permute IP (initial P)
  - Generate 16 subkey from key
  - Split text to L, R
  - 16 round i: 16 to 1 each round:
    - $R_i = L_{i-1}$
    - $L_i = R_{i-1} \oplus F(L_{i-1}, K_i)$ where F round function make xor
  - Combine L, R
  - Permute FP (final or IN_INV P)