Mật mã đối xứng
Giải thuật DES

Phạm Nguyên Khang
BM. Khoa học máy tính
pnkhang@cit.ctu.edu.vn

Data Encryption Standard

- DES được công nhận vào năm 1977 bởi Viện nghiên cứu quốc gia về chuẩn của Mỹ (NIST – National Institut of Standards and Technology)
- Nguyên lý:
  - Sử dụng một khóa K tạo ra n khóa con K_1, K_2, …, K_n
  - Hoán vị dữ liệu (Initial Permutation)
  - Thực hiện n vòng lặp, ở mỗi vòng lặp
    - Dữ liệu được chia thành hai phần
    - Áp dụng phép toán thay thế lên một phần, phần còn lại giữ nguyên
    - Hoán vị 2 phần cho nhau (trái ⇔ phái)
  - Hoán vị dữ liệu (Final Permutation)

Simplified DES – Giới thiệu

- Giải thuật DES đơn giản hóa (S-DES) được phát triển bởi GS. Edwward Schaefer tại Đại học Santa Clara vào năm 1996.
- Giải thuật S-DES với ít tham số hơn DES, chỉ mang tính hàn lâm, giúp sinh viên có một khung nhìn tổng quát trước khi tìm hiểu giải thuật DES.
- Mật mã hóa: dùng khối bảng rõ 8-bit và khóa 10-bit, sản sinh khối bảng mã 8-bit.
- Giải mật mã: dùng khối bảng mã 8-bit và khóa 10-bit, sản sinh khối bảng rõ 8-bit.

S-DES – Quy trình chính

- Mật mã hóa:

  $Ciphertext = IP^{-1}(f_{k2}(SW(f_{k1}(IP(Plaintext)))))$

  Trong đó

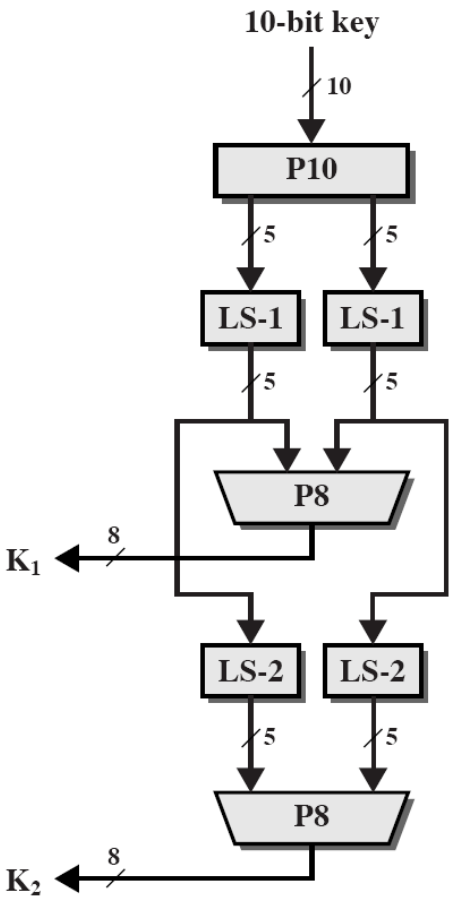
  $K_1 = P8(Shift(P10(key)))$

  $K_2 = P8(Shift(Shift(P10(key))))$

- Giải mật mã:

  $Plaintext = IP^{-1}(f_{k1}(SW(f_{k2}(IP(Ciphertext)))))$

S-DES – Sinh khóa



- Một khóa 10-bit được chia sẻ giữa người gởi và người nhận
- Từ khóa này, 2 khóa con được sinh ra để cung cấp cho các bước riêng biệt của quá trình mã hóa và giải mã.
- P10 có dạng:

  3 5 2 7 4 10 1 9 8 6
- P8 có dạng:

  6 3 7 4 8 5 10 9
- Ví dụ: khóa 1010000010

  P10: 10000 01100

  LS-1: 00001 11000

  P8 (K_1): 1010 0100

  LS-2: 00100 00011

  P8 (K_2): 0100 0011

Figure 3.2 Key Generation for Simplified DES

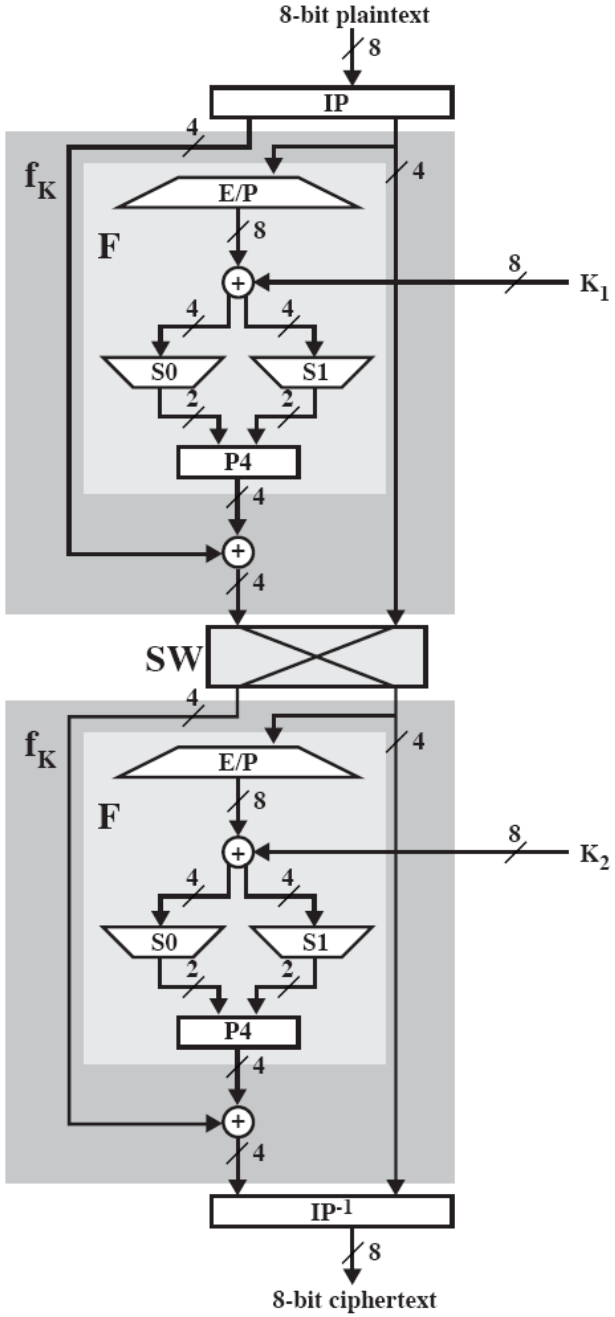


Figure 3.3 Simplified DES Encryption Detail

S-DES – Mật mã hóa

- IP: **2 6 3 1 4 8 5 7**
- IP^{-1}: **4 1 3 5 7 2 8 6**
- $f_K(L, R) = (L \oplus F(R, S_K), R)$
  S_K là khóa con (K_1 hoặc K_2)
- E/P: **4 1 2 3 2 3 4 1**
- P4 : **2 3 4 1**
- Hộp thay thế S-Box:
  - S_0
  - S_1
- SW: hoán vị hai nửa khối 4-bit

S-DES – Mật mã hóa

- Ghép bit 1, bit 4 làm hàng
- Ghép bit 2, bit 3 làm cột
- Tra bảng, đổi giá trị ra số nhị phân (2 bit)

- Ví dụ:
  - Đầu vào của S_0 là 0111
  - Bit 0 & 4: 01 ➔ hàng 1
  - Bit 2 & 3: 11 ➔ cột 3
  - Tra bảng được 0 ➔ 00

- S_0

	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3
3	3	1	3	2

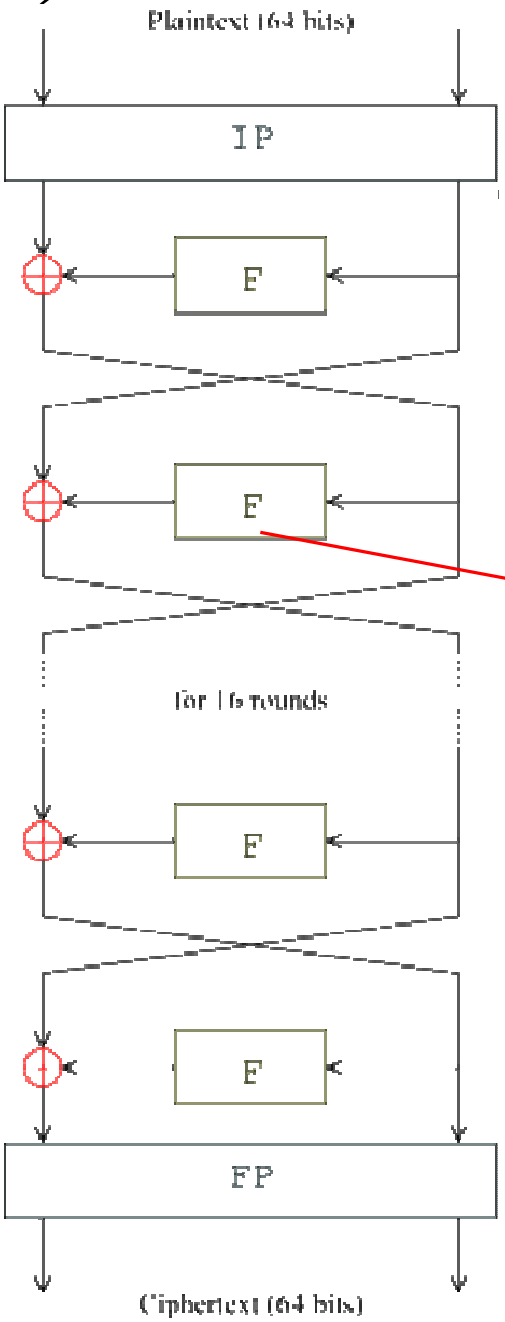
- S_1

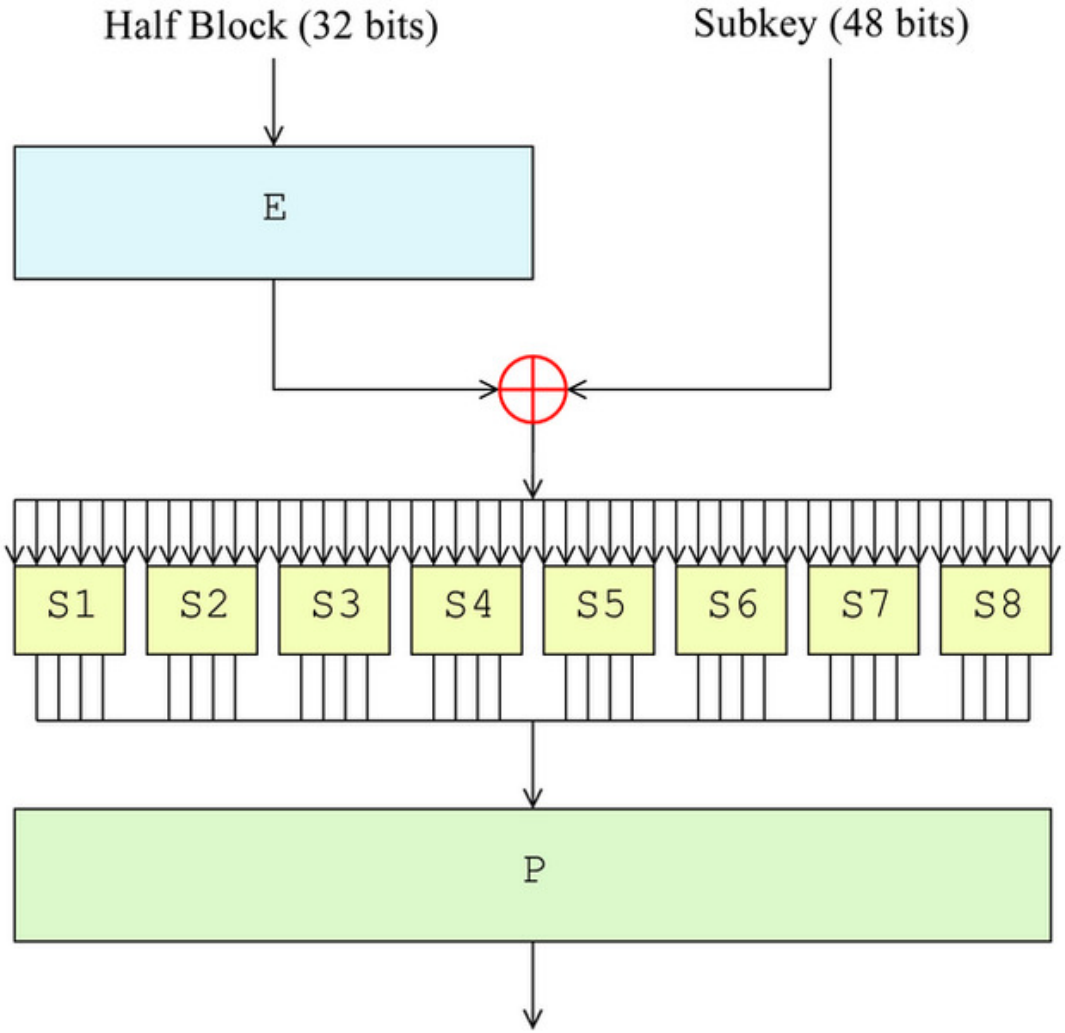
	0	1	2	3
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	2	1	0	3

DES

- Khóa
  - Lý thuyết: 56 bits = 7 bytes
  - Thực tế (trên Java) sử dụng 8 bytes (1 byte không sử dụng)
  - Sinh ra 16 khóa con K_1, K_2, …, K_{16}
- Khối:
  - 64 bits
- Số vòng lặp:
  - 16

Sơ đồ hoạt động





DES – Tóm tắt giải thuật

- Tạo 16 khóa con

  C [0]*D* [0] = *PC-1(KEY)*

  for *i* = 1 **to** 16

  C[i] = LeftShift[i](C[i-1])

  D[i] = LeftShift[i](D[i-1])

  K[i] = PC-2(C[i]D[i])

  end for

- Mã hóa khối dữ liệu

  L[0]R[0] = IP(plain block)

  for i=1 to 16

  L[i] = R[i-1]

  R[i] = L[i-1] XOR F(R[i-1], K[i])

  end for

  cipher block = FP(R[16]L[16])

DES – Tóm tắt giải thuật

- Giải mã khối dữ liệu

  R[16]L[16] = IP(cipher block)

  for i=1 to 16

    R[i-1] = L[i]

    L[i-1] = R[i] xor f(L[i], K[i])

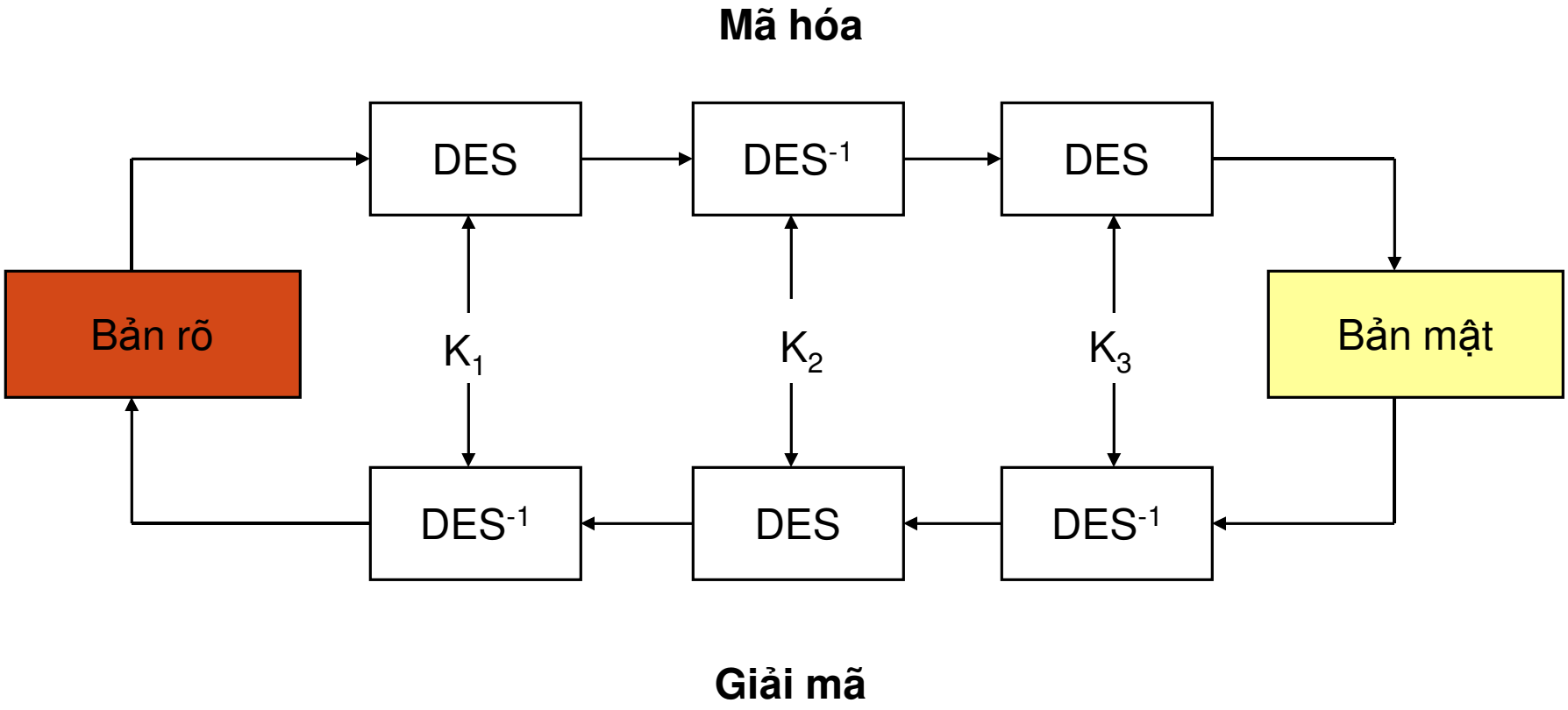
  end for

  plain block = FP(L[0]R[0])

DES – Đánh giá hiệu năng

- Khóa 56 bits → có $2^{56} = 7.2 * 10^{16}$ khóa
- Tấn công kiểu brute-force với 1 encryption/us mất 1142 năm
- Trên thực tế, với những thiết bị chuyên dụng và phần cứng đắt tiền (20 triệu USD vào năm 1977) có thể 'bẻ khóa' DES trong 10 giờ

An toàn hơn nữa với DES: 3-DES (TripleDES)

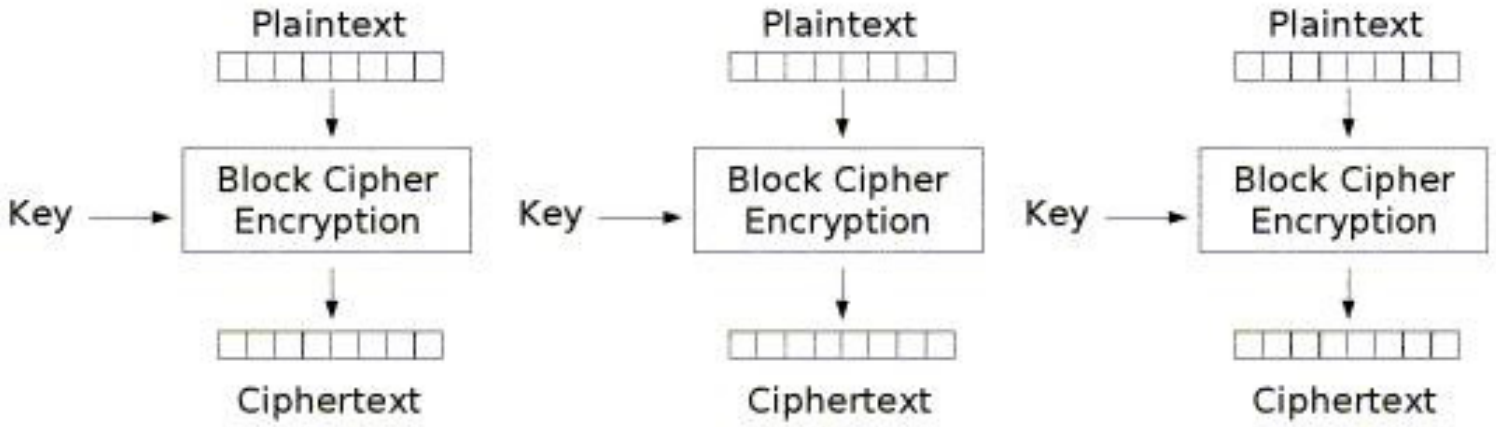


Giải thuật mã hóa khác

- Blowfish
  - Có thể hoạt động với bộ nhớ < 5KB
  - Kích thước khóa thay đổi, có thể đến 448 bit
- AES: Advanced Encryption Standard
- RC2 và RC4
  - Do Ron Rivest(Ron's code) đề nghị
  - Kích thước khóa từ 1 đến 2048 bit
- RC5
  - Kích thước khóa là một tham số đầu vào
- IDEA: International Data Encryption Algorithm
  - Khóa 128 bit, được sử dụng bởi PGP

Phương pháp mã hóa khối - ECB

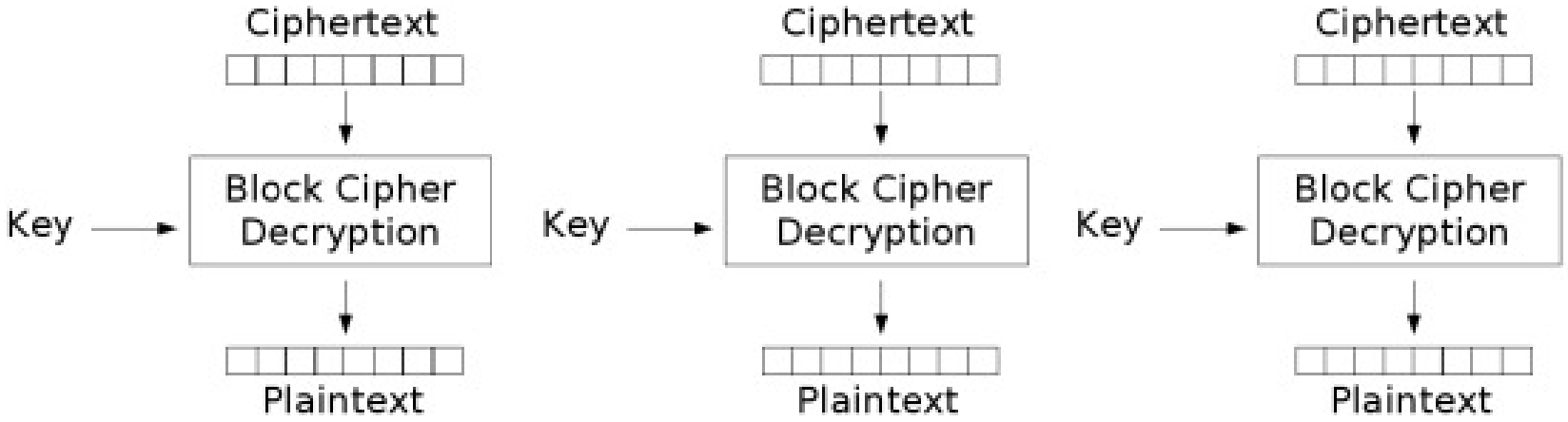
- ECB – Electronic Codebook
- Chia thông điệp thành các khối 64 bits, nhồi thêm dữ liệu vào khối cuối (nếu cần thiết)
- Mã hóa: $C_i = E_k(P_i)$



Electronic Codebook (ECB) mode encryption

Phương pháp mã hóa khối - ECB

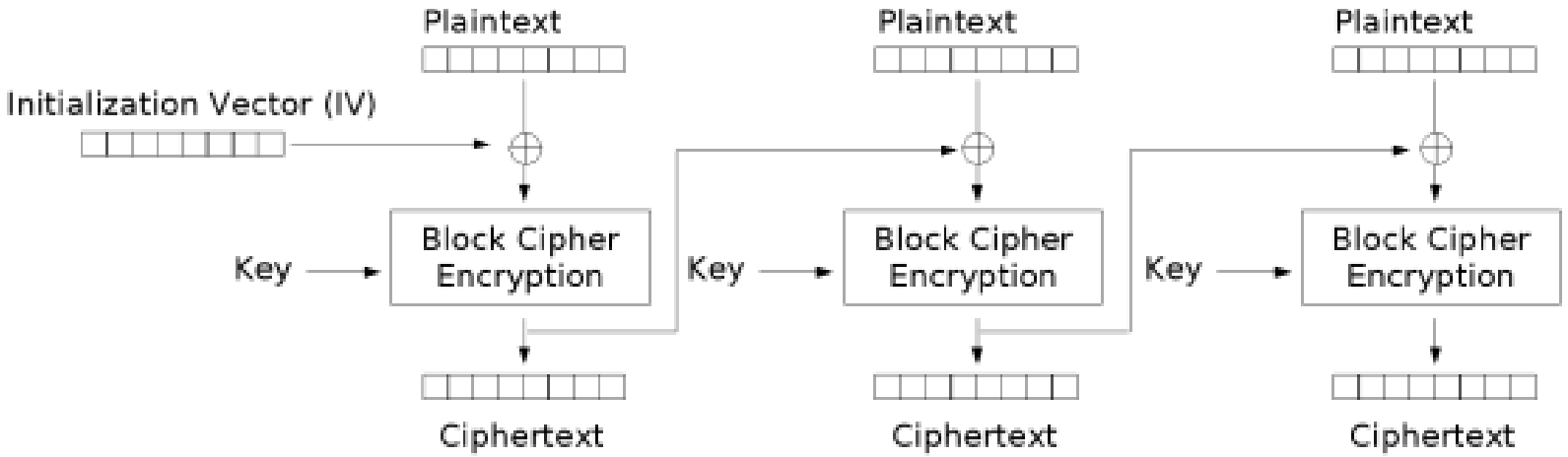
- Giải mã: $P_j = D_k(C_j)$
- Chỉ thích hợp cho việc mã hóa các thông điệp ngắn. Bảng mã của thông điệp dài có tính an toàn không cao.



Electronic Codebook (ECB) mode decryption

Phương pháp mã hóa khối - CBC

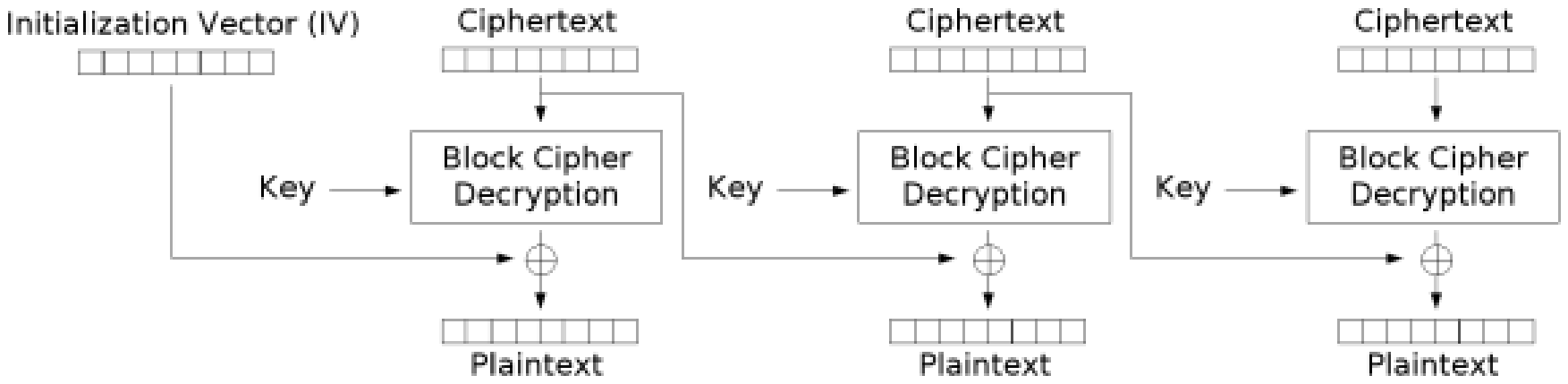
- CBC – Cipher Block Chaining
- Mã hóa: $C_j = E_k(C_{j-1}\ XOR\ P_j)$
- Cả hai phía mã hóa và giải mã đều dùng chung vector IV (initialization vector) để thao tác trên khối dữ liệu đầu



Cipher Block Chaining (CBC) mode encryption

Phương pháp mã hóa khối - CBC

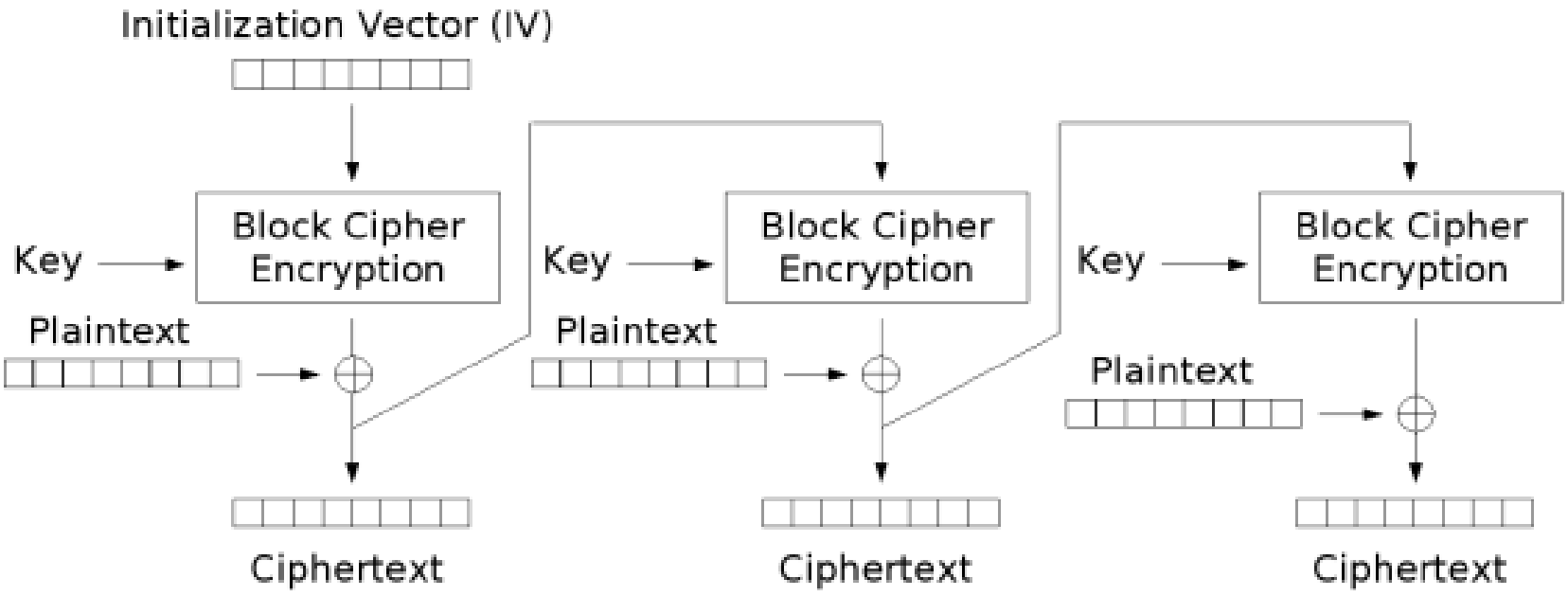
- Giải mã: $P_j = C_{j-1}$ XOR $D_k(C_j)$
- Chú ý khối đầu tiên:
  - $C_0 = E_k(IV \text{ XOR } P_j)$
  - $P_0 = IV \text{ XOR } D_k(C_1)$



Cipher Block Chaining (CBC) mode decryption

Phương pháp mã hóa khối - CFB

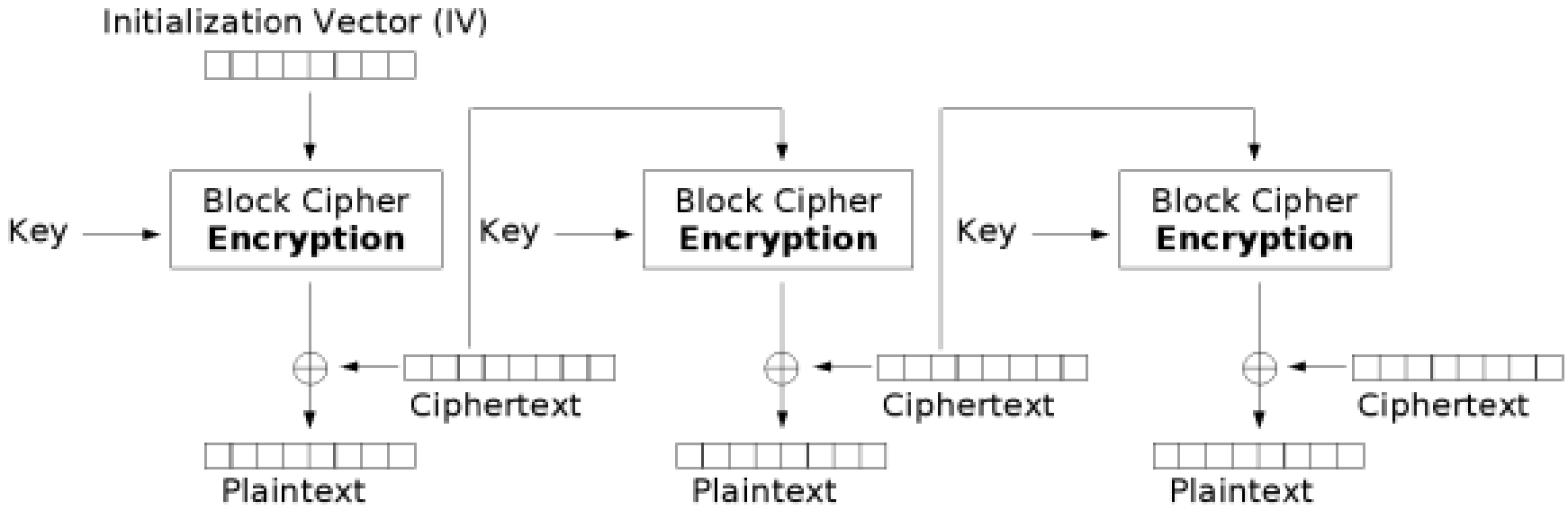
- CFB – Cipher FeedBack
- Mã hóa: $C_j = P_j$ XOR $E_k(C_{j-1})$



Cipher Feedback (CFB) mode encryption

Phương pháp mã hóa khối - CFB

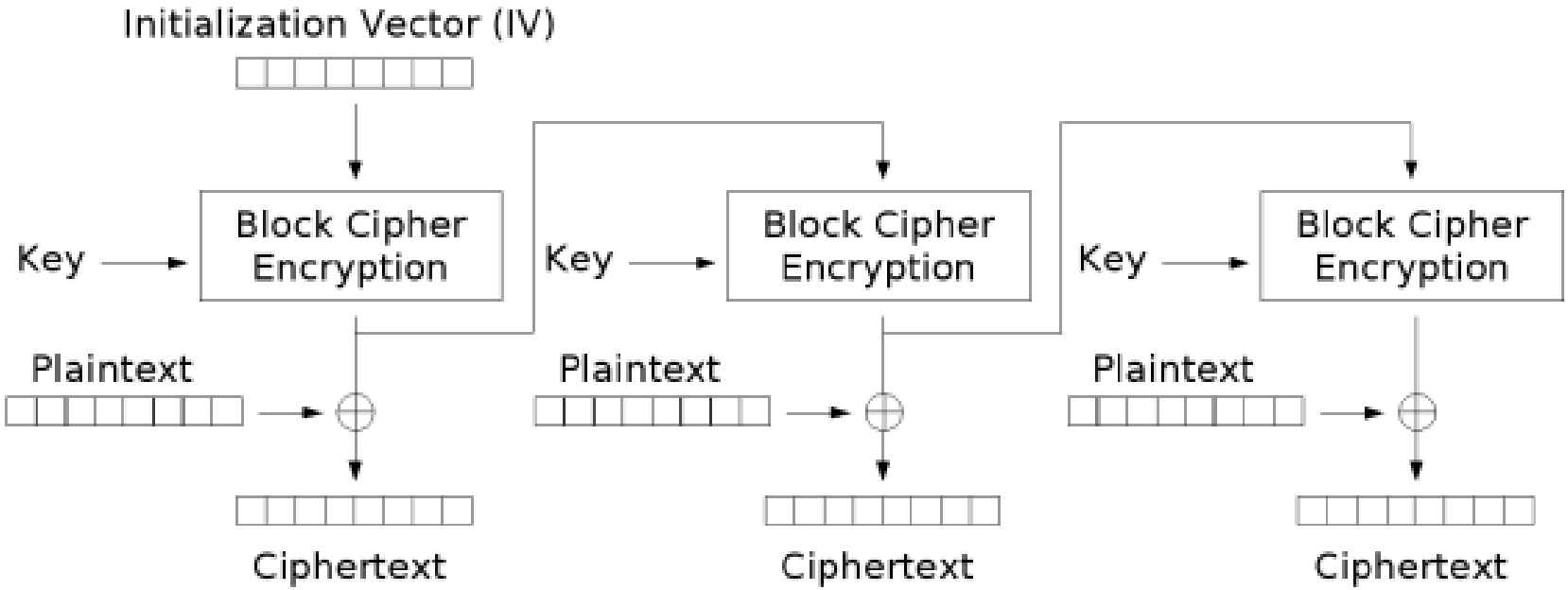
- Giải mã: $P_j = C_j$ XOR $D_k(C_{j-1})$



Cipher Feedback (CFB) mode decryption

Phương pháp mã hóa khối - OFB

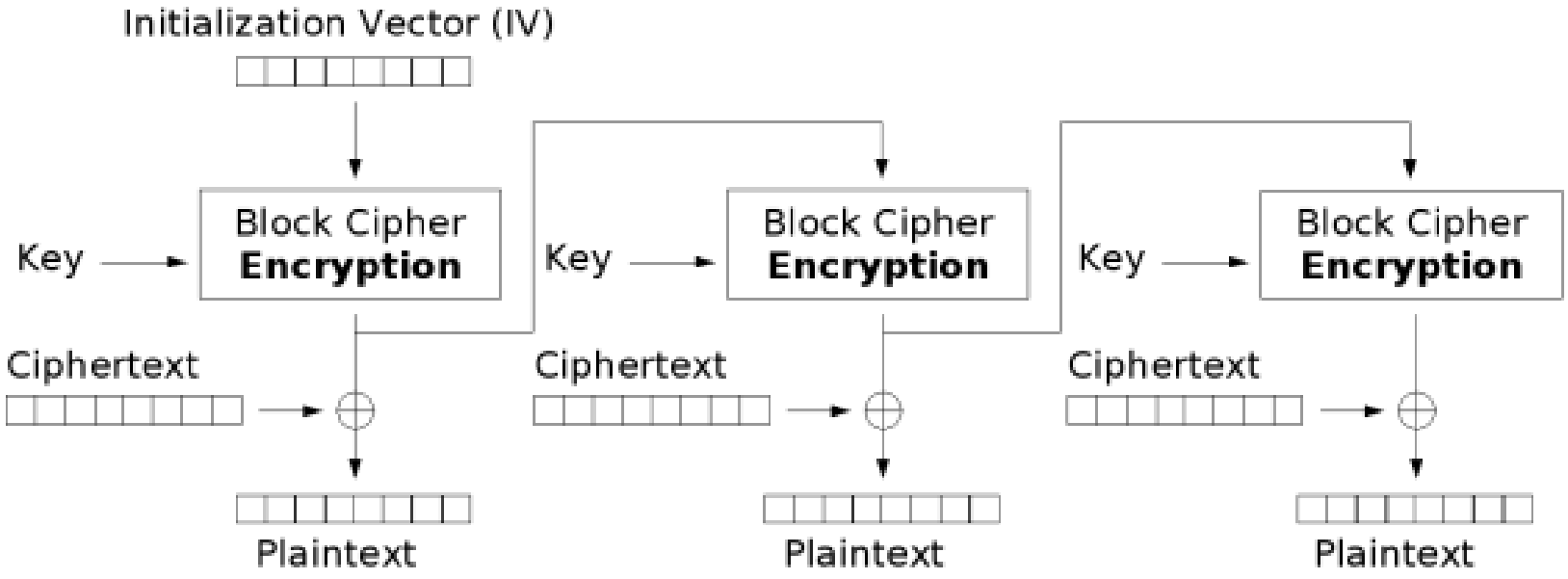
- OFB – Output FeedBack
- Mã hóa



Output Feedback (OFB) mode encryption

Phương pháp mã hóa khối - OFB

- Giải mã



Output Feedback (OFB) mode decryption