



SECURITYBOX™



# AN TOÀN THÔNG TIN

**Giảng viên: Th.s Nguyễn Thu Hiền**

**Email: [nthien@uneti.edu.vn](mailto:nthien@uneti.edu.vn)**

**Tổ Mạng Máy Tính và Công Nghệ Đa Phương Tiện**

**Khoa Công nghệ Thông tin**

# CHƯƠNG 3: MÃ HÓA ĐỐI XỨNG HIỆN ĐẠI

- 3.1. Nguyên lý của các hệ mã hóa khối.**
- 3.2. Chuẩn mã hóa dữ liệu DES.**
- 3.3. Hệ mã hóa 3DES.**
- 3.4. Chuẩn mã hóa tiên tiến AES.**
- 3.5. Các hệ mã hóa khối khác.**
- 3.6. Các phương thức mã hóa liên hợp.**
- 3.7. Triển khai chức năng mã hóa.**
- 3.8. Quản lý và phân phối khóa**

## 3.3 Hệ mã hóa 3DES

- Mã hóa một khối dữ liệu 64 bit (bản rõ) thành một khối dữ liệu mới 64 bit (bản mã).
- **Mã hóa:** sử dụng 3 khóa ( $K_1$ ,  $K_2$ ,  $K_3$  - mỗi khóa có giá trị 56 bit), chạy 3 lần giải thuật DES.
  - $C = E_{K_3}[D_{K_2}[E_{K_1}[p]]]$
  - Thực hiện mã hóa DES với khóa  $K_1$ .
  - Giải mã DES với khóa  $K_2$
  - Mã hóa DES với khóa  $K_3$
- **Giải mã:**  $p = D_{K_1}[E_{K_2}[D_{K_3}[C]]]$ 
  - Giải mã với khóa  $K_3$ , mã hóa với khóa  $K_2$ , và giải mã với khóa  $K_1$ .

## 3.3 Hệ mã hóa 3DES (t)

- Tiêu chuẩn chỉ ra phương thức lựa chọn nhóm khóa ( $K_1$ ,  $K_2$ ,  $K_3$ ) như sau:
  - Lựa chọn 1:  $K_1$ ,  $K_2$ ,  $K_3$  là các khóa độc lập
  - Lựa chọn 2:  $K_1$ ,  $K_2$  là hai khóa độc lập và  $K_3 = K_1$
  - Lựa chọn 3:  $K_1 = K_2 = K_3$
- Lựa chọn 1 là phương thức mã hóa mạnh nhất với 168 bit khóa độc lập ( $168 = 3 \times 56$ )
- Lựa chọn 2 ít bảo mật hơn với 112 bit khóa ( $2 \times 56 = 112$  bit)
- Lựa chọn 3 chỉ tương đương với việc mã hóa DES 1 lần với 56 bit khóa.

## 3.3 Hệ mã hóa 3DES (t)

- Độ dài khóa thực tế là 168 bit
  - Không tồn tại  $K_4 = 56$  sao cho  $C = E_{K_4}(p)$
- Nếu dùng 2 lần DES trên hai khóa khác nhau  $K_1, K_2$ 
  - $C = E_{K_2}(E_{K_1}(p)) \Rightarrow X = E_{K_1}(p) = D_{K_2}(C)$
  - Dễ gặp tấn công ở mức trung gian  $\Rightarrow$  dễ tìm ra khóa để phá mã.
  - Nếu biết một cặp  $(p, C)$ 
    - Mã hóa  $p$  với  $2^{56}$  khóa và giải mã  $C$  với  $2^{56}$  khóa
    - So sánh tìm ra  $K_1$  và  $K_2$  tương ứng
    - Kiểm tra lại với 1 cặp  $(p, C)$  mới: nếu đúng  $\Rightarrow K_1$  và  $K_2$  là khóa

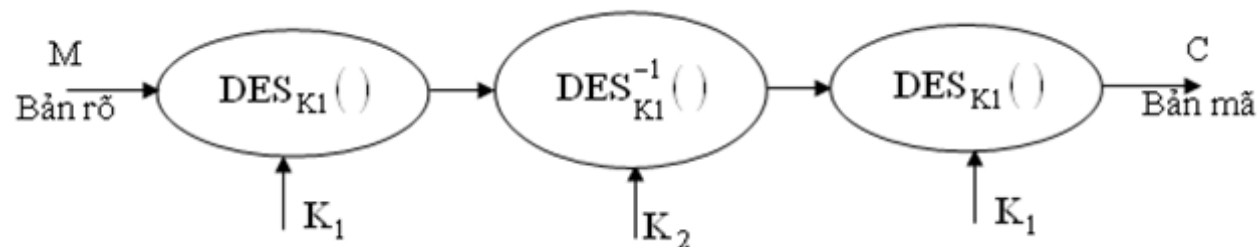
## 3.3 Hệ mã hóa 3DES (t)

❖ Có thể sử dụng 3DES với 2 khoá

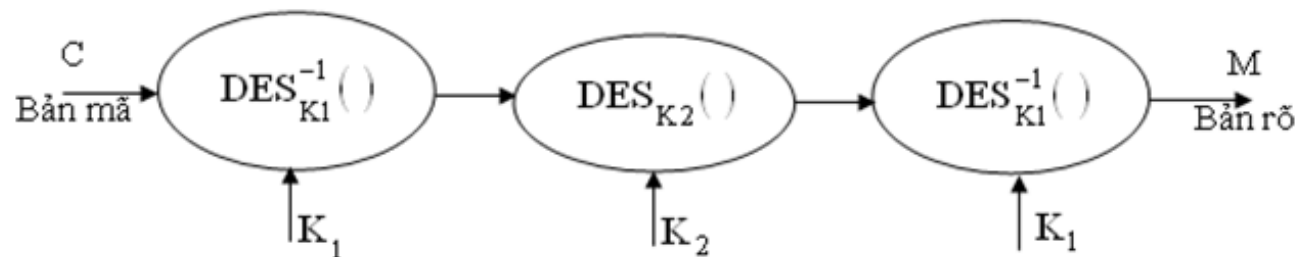
- Để tránh tấn công ở mức trung gian, cần sử dụng 3 mã, vậy nói chung có thể dùng 3 khoá khác nhau

- $C = EK_1[DK_2[EK_1[P]]]$

- Nếu  $K_1 = K_2$  thì tương đương làm việc với một lần DES



- Giải mã



## 3.4 Chuẩn mã hóa tiên tiến AES

- AES (Advanced Encryption Standard) được công nhận chuẩn mới năm 2001
- Tên giải thuật là Rijndael (Rijmen + Daemen)
- An toàn hơn và nhanh hơn 3DES
- Kích thước khối : 128 bit
- Kích thước khóa : 128/192/256 bit
- Số vòng : 10/12/14
- Cấu trúc mạng S-P, nhưng không theo hệ Feistel
  - Không chia mỗi khối làm đôi



## 3.4 Chuẩn mã hóa tiên tiến AES (t)

- Thuật toán AES khá phức tạp, được mô tả khái quát gồm 3 bước như sau:
  1. Bước khởi tạo: sử dụng phép AddRoundKey để kết hợp giữa dữ liệu cần được mã hóa  $p$  và khóa  $key$ .



## 3.4 Chuẩn mã hóa tiên tiến AES (t)

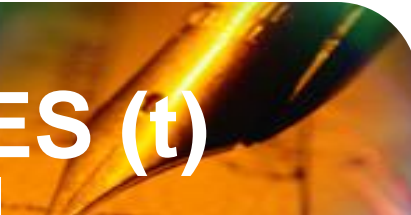
2. Bước lập mã hóa gồm 4 phép biến đổi lần lượt:

**SubBytes, ShiftRows, MixColumns, AddRoundKey**

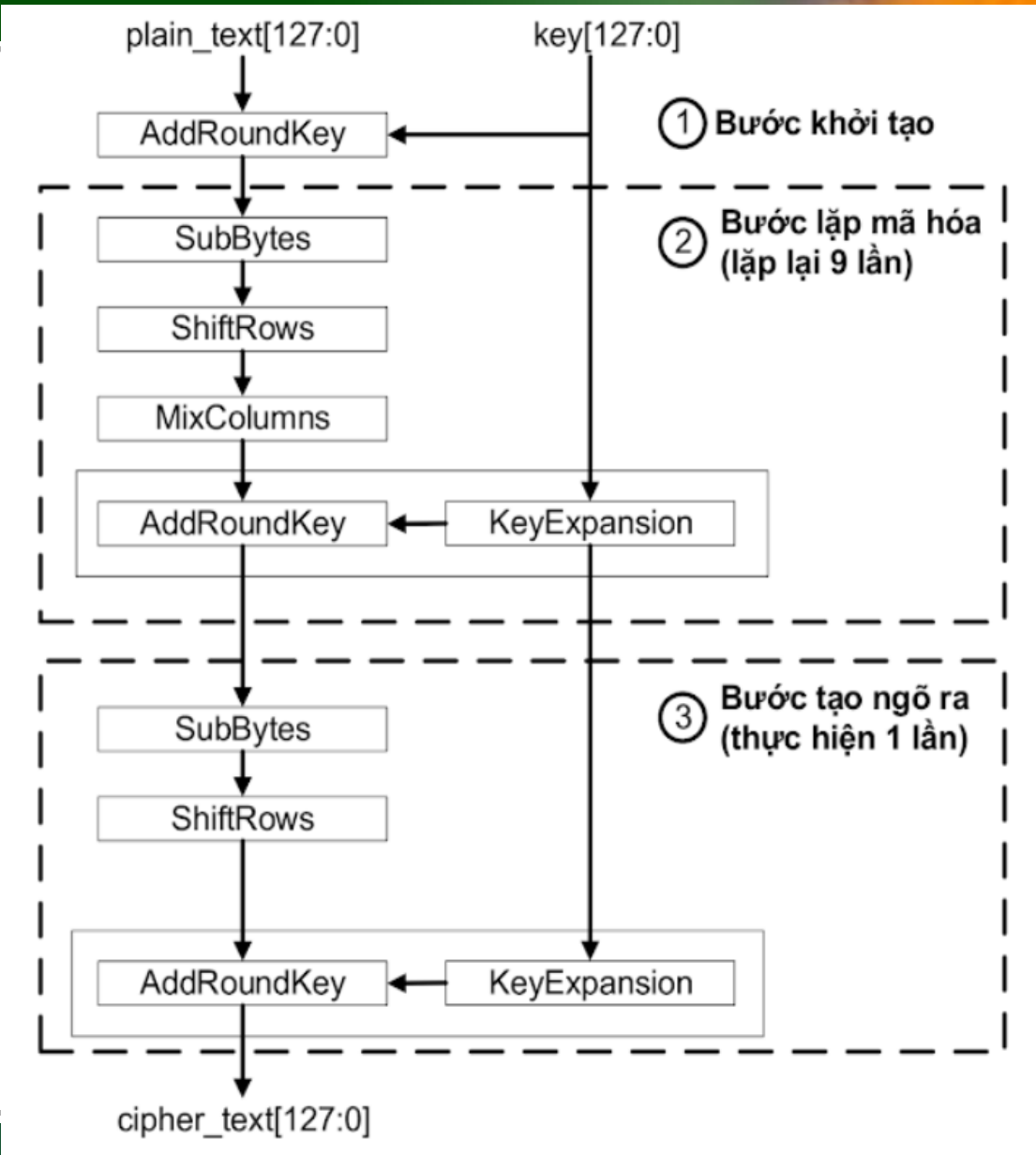
- Kết quả bước 1 được sử dụng để thực hiện tuần tự các chức năng SubBytes, ShiftRows, MixColumns và AddRoundKey.
- Bước này được lặp lại 9 lần.
- KeyExpansion thực hiện song song với bước AddRoundKey để tạo khóa vòng cho chức năng này.

## 3.4 Chuẩn mã hóa tiên tiến AES (t)

3. Bước tạo ngõ ra: gồm các phép biến đổi giống vòng lặp và không có phép MixColumns.
  - Sau 9 lần lặp ở bước 2, kết quả được sử dụng để thực hiện tuần tự các chức năng SubBytes, ShiftRows và AddRoundKey để tạo bản mã.



# ES (t)



## 3.4 Chuẩn mã hóa tiên tiến AES (t)

- Phép xử lý vòng (trừ vòng cuối cùng) được tạo nên từ 4 phép toán khác nhau:
  - AddRoundKey: cộng (XOR) khóa của chu kỳ vào trạng thái hiện hành. Độ dài của khóa của chu kỳ bằng với kích thước của trạng thái.
  - SubBytes: thay thế phi tuyến mỗi byte trong trạng thái hiện hành thông qua bảng thay thế (S-box).
  - MixColumns: trộn thông tin của từng cột trong trạng thái hiện hành. Mỗi cột được xử lý độc lập.
  - ShiftRows: dịch chuyển xoay vòng từng dòng của trạng thái hiện hành với di số khác nhau.

## 3.5 Các hệ mã hóa khối khác

### ➤ **IDEA (International Data Encryption Algorithm)**

- Khối 64 bit, khóa 128 bit, 8 vòng
- Theo cấu trúc mạng S-P, nhưng không theo hệ Feistel
  - Mỗi khối chia làm 4
- Rất an toàn
- Bản quyền bởi Ascom nhưng dùng miễn phí

### ➤ **Blowfish**

- Khối 64 bit, khóa 32-448 bit (ngầm định 128 bit), 16 vòng
- Theo cấu trúc hệ Feistel
- An toàn, khá nhanh và gọn nhẹ
- Tự do sử dụng

## 3.5 Các hệ mã hóa khối khác (t)

### ➤ RC5

- Phát triển bởi Ron Rivest
- Khối 32/64/128 bit, khóa 0-2040 bit, 0-255 vòng
- Đơn giản, thích hợp các bộ xử lý có độ rộng khác nhau
- Theo cấu trúc hệ Feistel

### ➤ CAST-128

- Phát triển bởi Carlisle Adams và Stafford Tavares
- Khối 64 bit, khóa 40-128 bit, 12/16 vòng
- Có 3 loại hàm vòng dùng xen kẽ
- Theo cấu trúc hệ Feistel
- Bản quyền bởi Entrust nhưng dùng miễn phí

## 3.6 Các phương thức mã hóa liên hợp

### ➤ ECB (Electronic Codebook)

- Mã hóa từng khối riêng rẽ

### ➤ CBC (Cipher Block Chaining)

- Khối nguyên bản hiện thời được XOR với khối bản mã trước đó

### ➤ CFB (Cipher Feedback)

- Mô phỏng mã hóa luồng (đơn vị s bit)
- s bit mã hóa trước được đưa vào thanh ghi đầu vào hiện thời



## 3.6 Các phương thức mã hóa liên hợp (t)

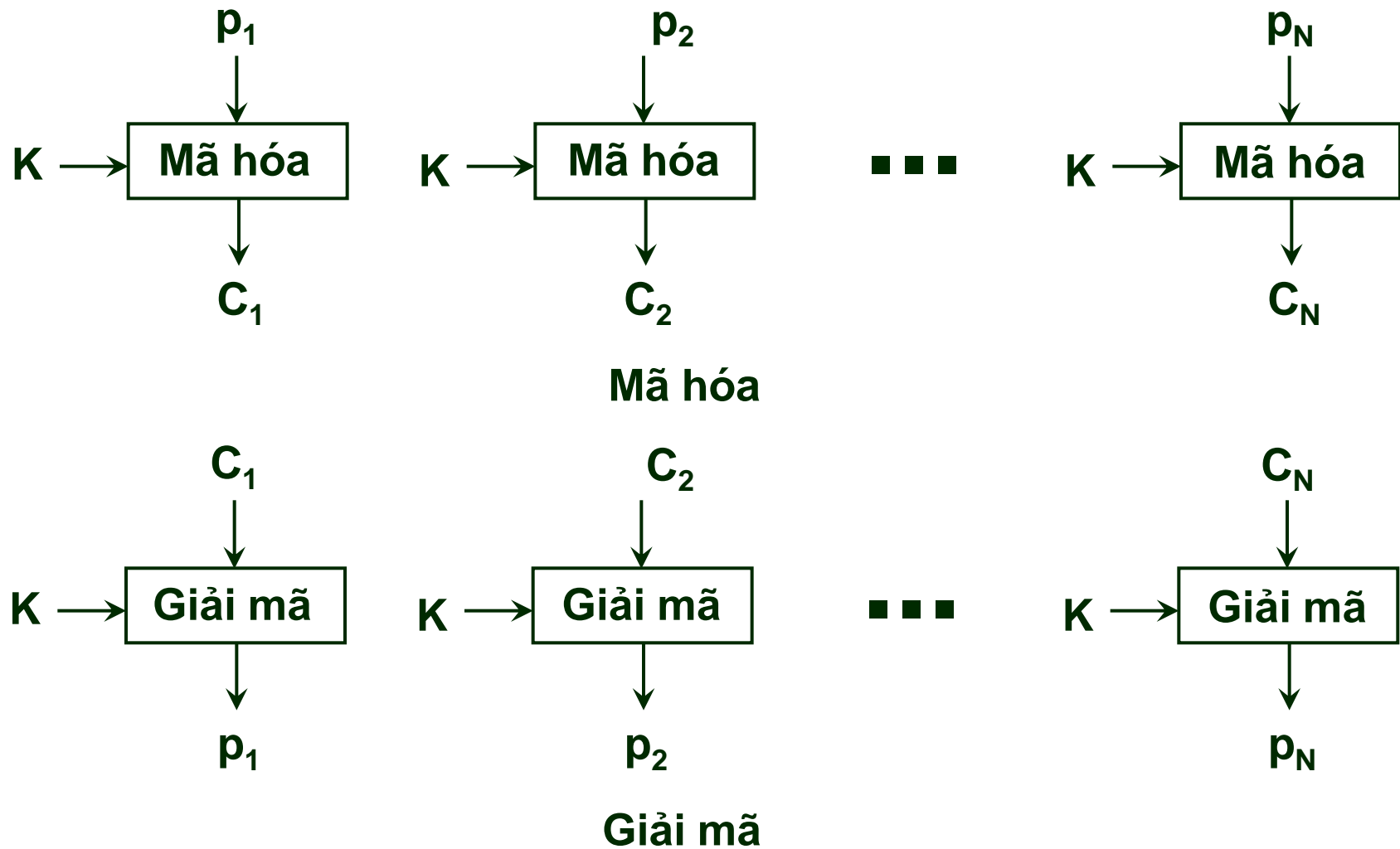
### ➤ OFB (Output Feedback)

- s bit trái đầu ra trước được đưa vào thanh ghi đầu vào hiện thời

### ➤ CTR (Counter)

- XOR mỗi khối nguyên bản với 1 giá trị thanh đếm mã hóa

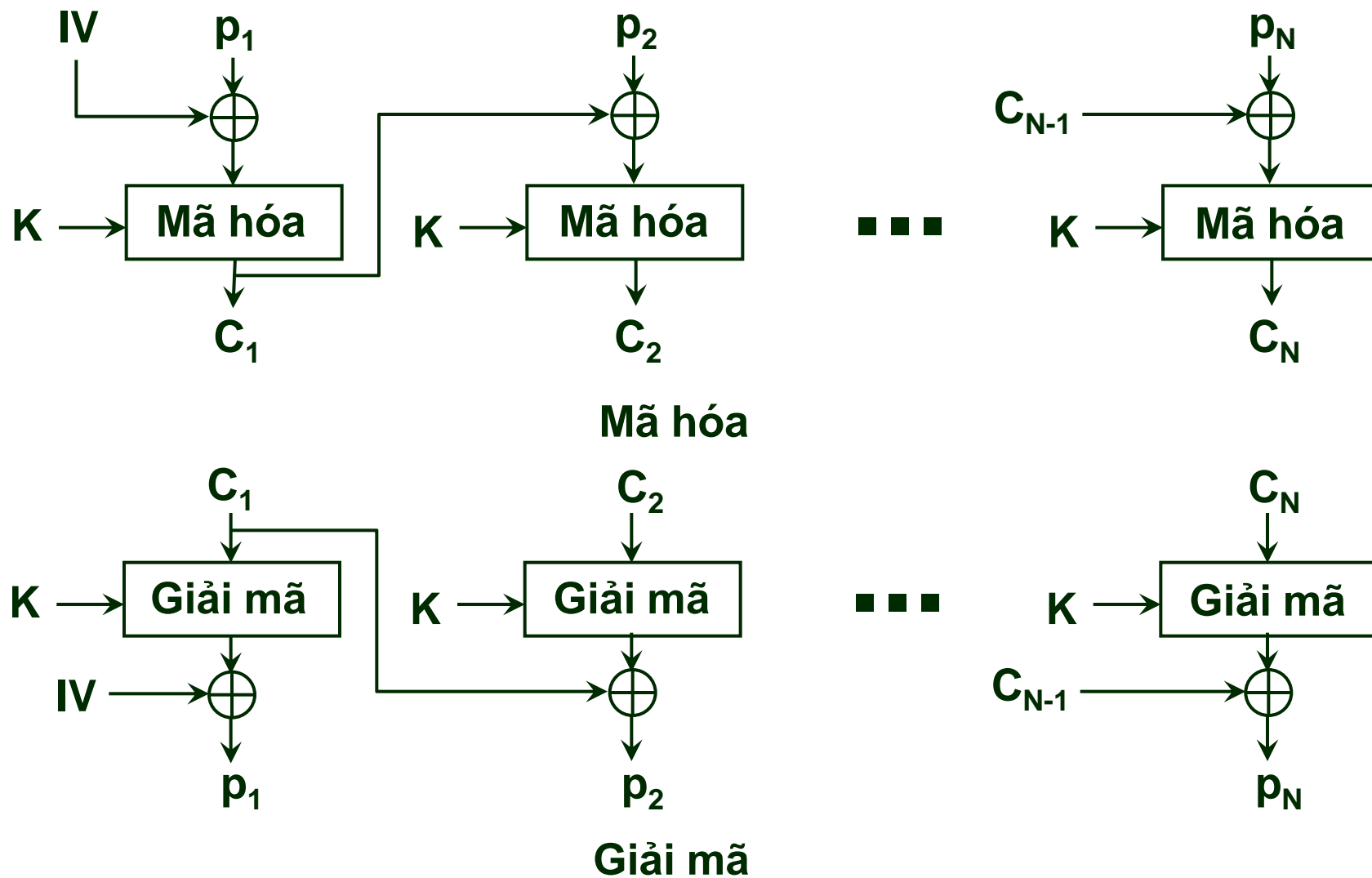
# Phương thức ECB



# Đánh giá ECB

- Những khối lặp lại trong nguyên bản có thể thấy được trong bản mã
- Nếu thông báo dài, có thể
  - Giúp phân tích phá mã
  - Tạo cơ hội thay thế hoặc bố trí lại các khối
- Nhược điểm do các khối được mã hóa độc lập
- Chủ yếu dùng để gửi thông báo có ít khối
  - Ví dụ gửi khóa

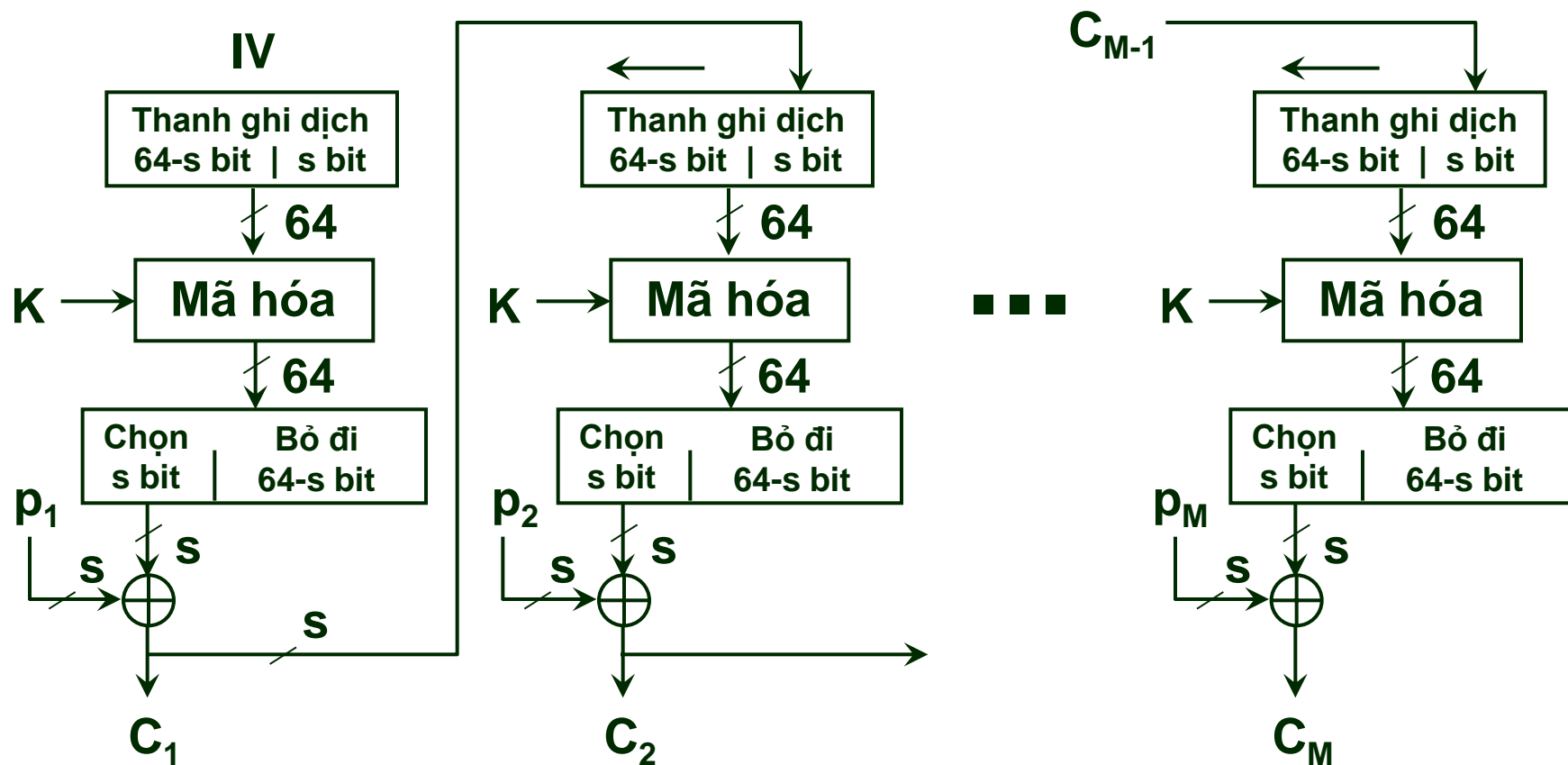
# Phương thức CBC



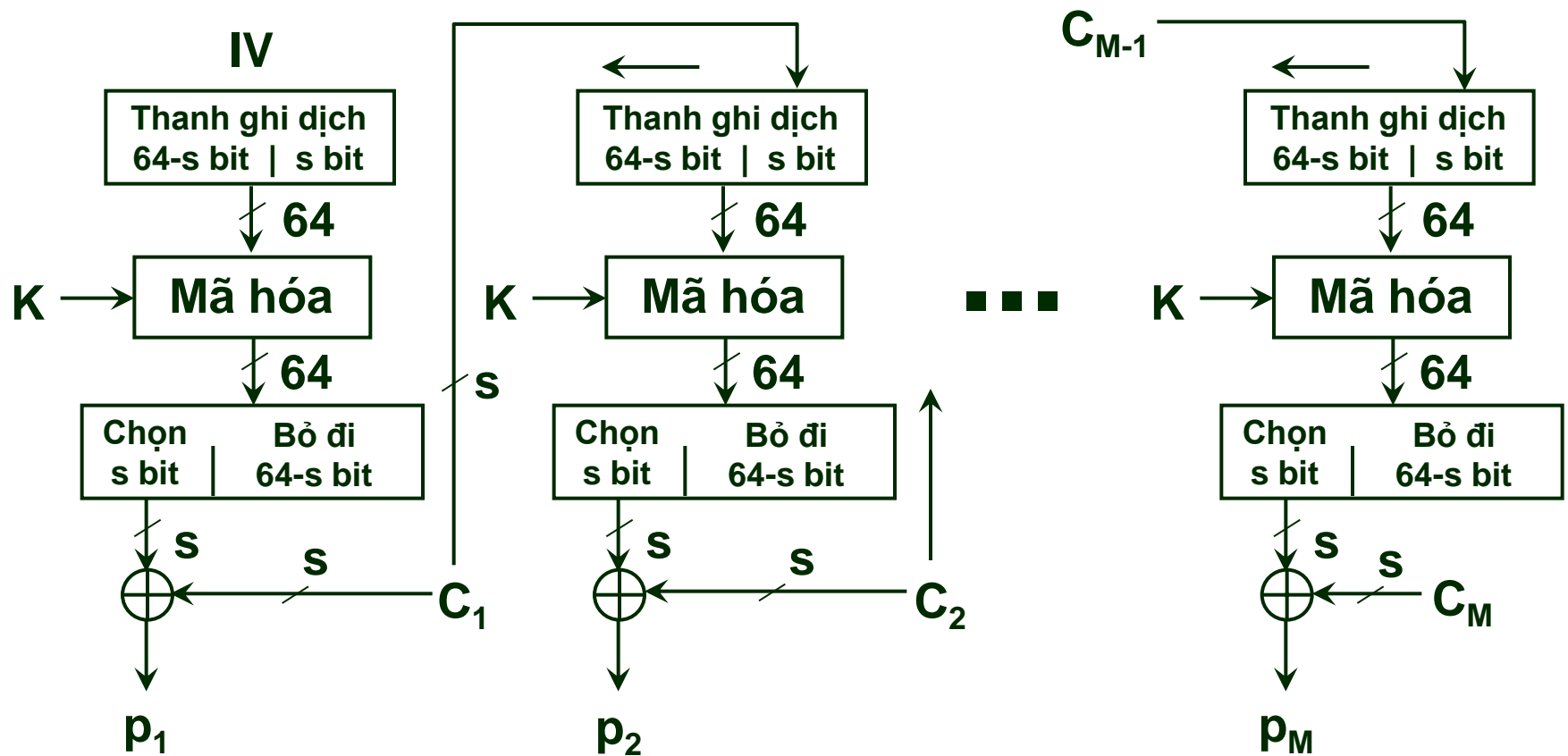
# Đánh giá CBC

- **Mỗi khối mã hóa phụ thuộc vào tất cả các khối nguyên bản trước đó**
  - Sự lặp lại các khối nguyên bản không thể hiện trong bản mã hóa
  - Thay đổi trong mỗi khối nguyên bản ảnh hưởng đến tất cả các khối bản mã về sau
- **Cần 1 giá trị đầu IV bên gửi và bên nhận đều biết**
  - Cần được mã hóa giống khóa
  - Nên khác nhau đối với các thông báo khác nhau
- **Cần xử lý đặc biệt khối nguyên bản không đầy đủ cuối cùng**
- **Dùng mã hóa dữ liệu lớn, xác thực**

# Mã hóa CFB



# Giải mã CFB

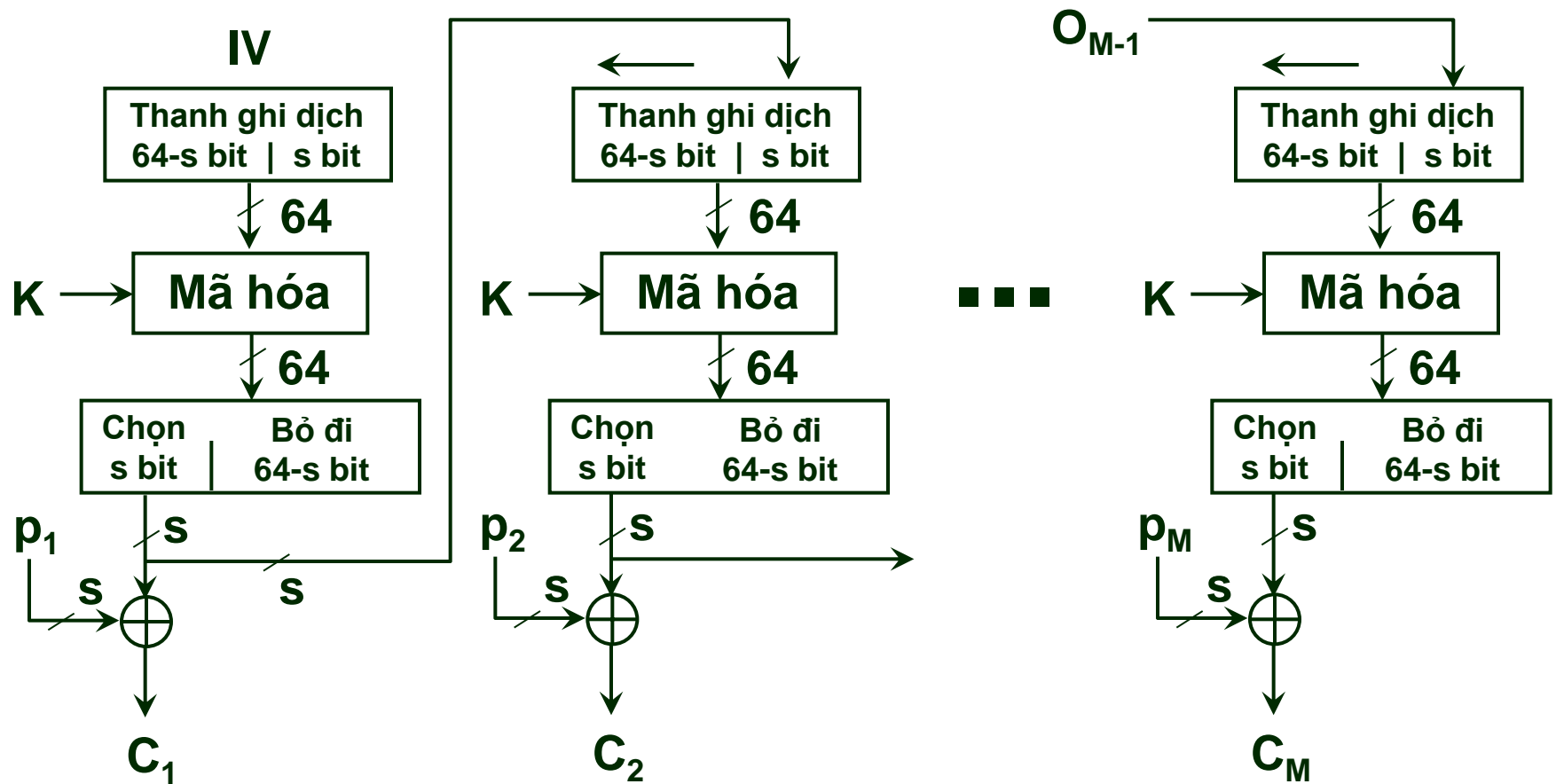




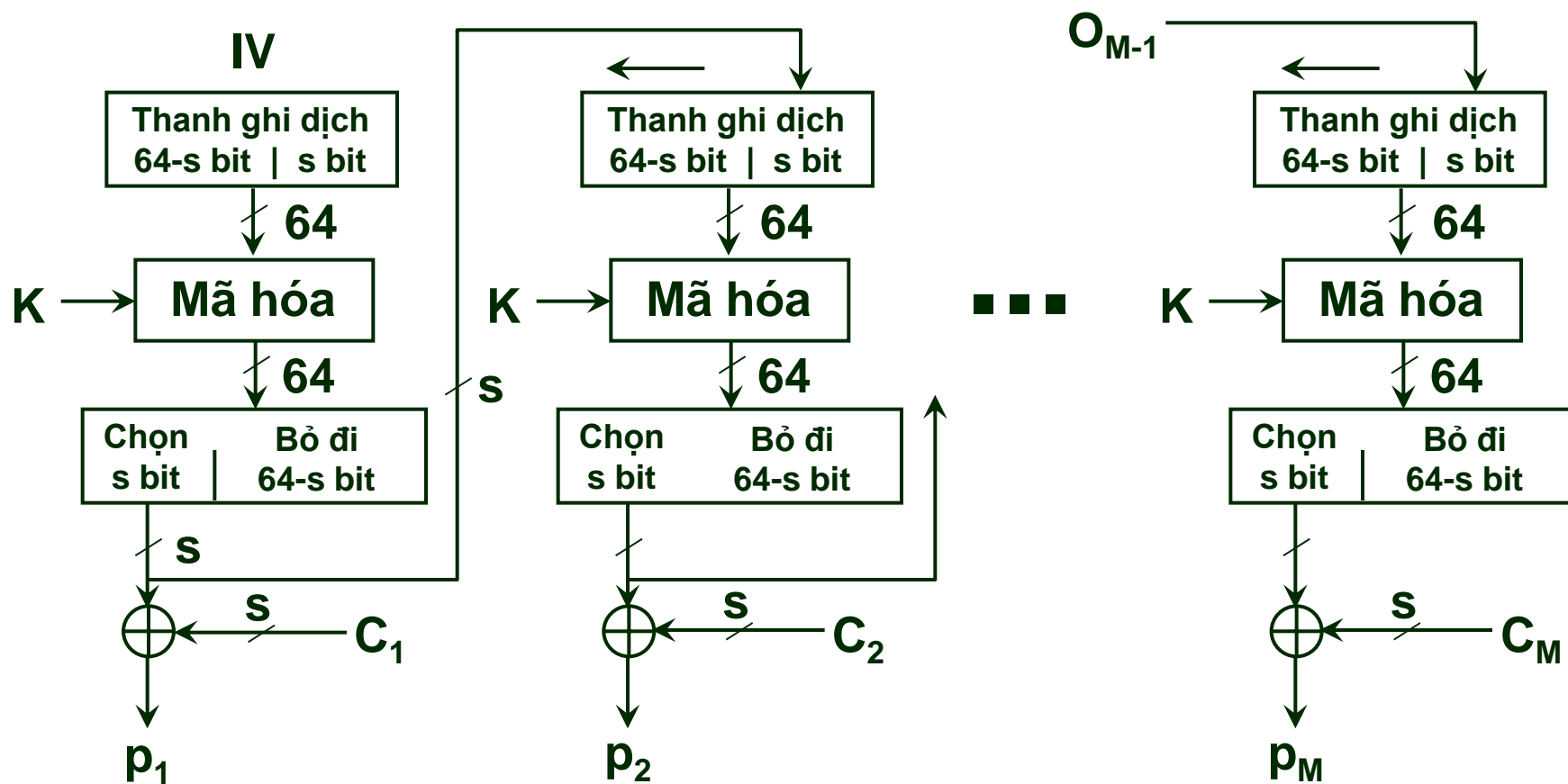
# Đánh giá CFB

- Thích hợp khi dữ liệu nhận được theo từng đơn vị bit hay byte
- Không cần đệm thông báo để làm tròn khối
- Cho phép số lượng bit bất kỳ
  - Ký hiệu CFB-1, CFB-8, CFB-64,...
- Là phương thức luồng phổ biến nhất
- Dùng giải thuật mã hóa ngay cả khi giải mã
- Lỗi xảy ra khi truyền 1 khối mã hóa sẽ lan rộng sang các khối tiếp sau.

# Mã hóa OFB



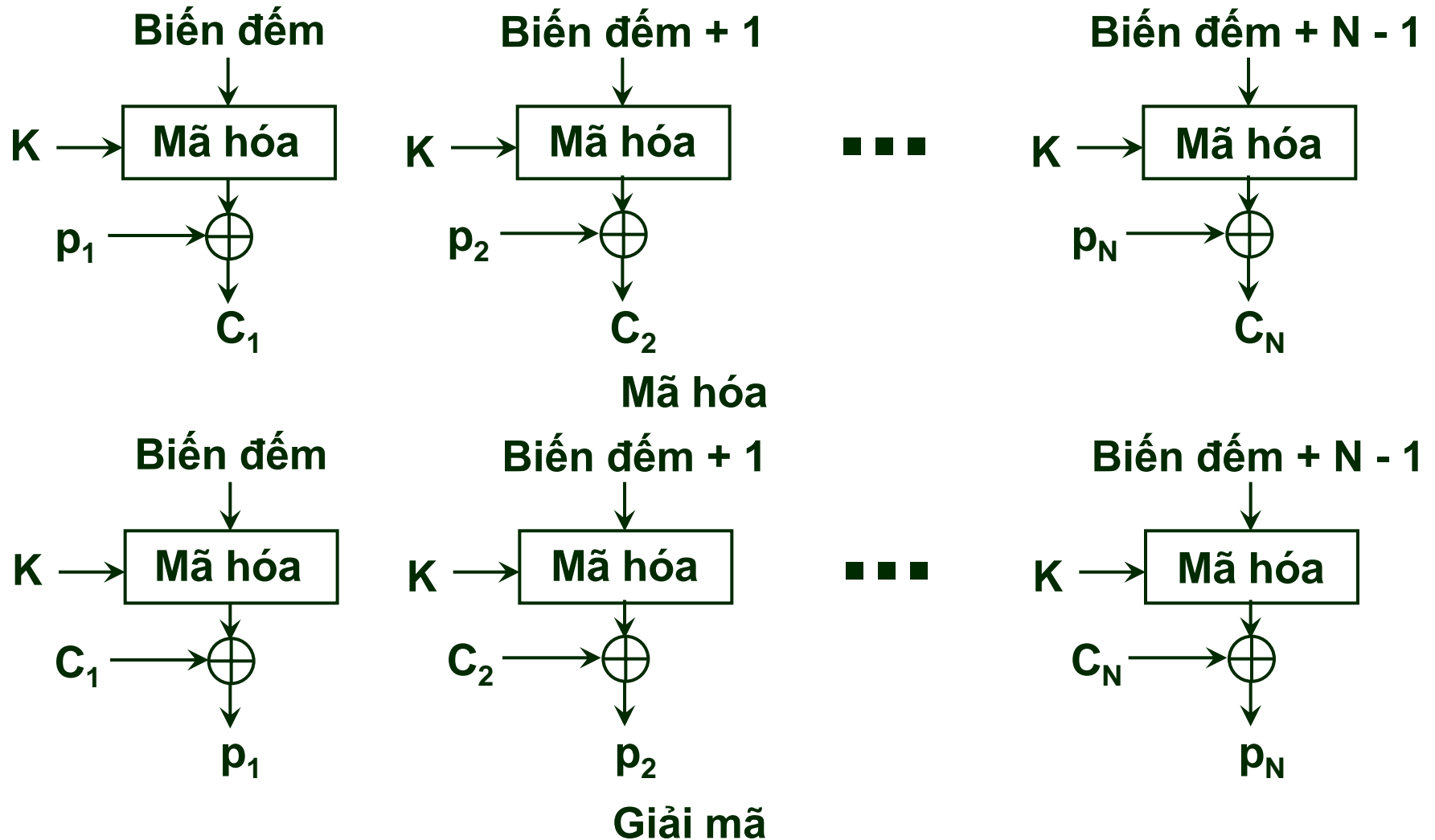
# Giải mã OFB



# Đánh giá OFB

- Tương tự CFB chỉ khác là phản hồi lấy từ đầu ra giải thuật mã hóa, độc lập với thông báo.
- Không bao giờ sử dụng lại cùng khóa và IV.
- Lỗi truyền 1 khối mã hóa không ảnh hưởng đến các khối khác.
- Thông báo dễ bị sửa đổi nội dung.
- Chỉ nên dùng OFB-64.
- Có thể tiết kiệm thời gian bằng cách thực hiện giải thuật mã hóa trước khi nhận được dữ liệu.

# Phương thức CTR



# Đánh giá CTR

- Hiệu quả cao
  - Có thể thực hiện mã hóa (hoặc giải mã) song song
  - Có thể thực hiện giải thuật mã hóa trước nếu cần
- Có thể xử lý bất kỳ khối nào trước các khối khác
- An toàn không kém gì các phương thức khác
- Đơn giản, chỉ cần cài đặt giải thuật mã hóa, không cần đến giải thuật giải mã
- Không bao giờ sử dụng lại cùng giá trị khóa và biến đếm (tương tự OFB)

## 3.7 Triển khai chức năng mã hóa

- Giải pháp hữu hiệu và phổ biến nhất chống lại các mối đe dọa đến an toàn mạng là mã hóa.
- Để thực hiện mã hóa, cần xác định
  - Mã hóa những gì
  - Thực hiện mã hóa ở đâu
- Có 2 phương án cơ bản
  - Mã hóa liên kết
  - Mã hóa đầu cuối



# Mã hóa liên kết

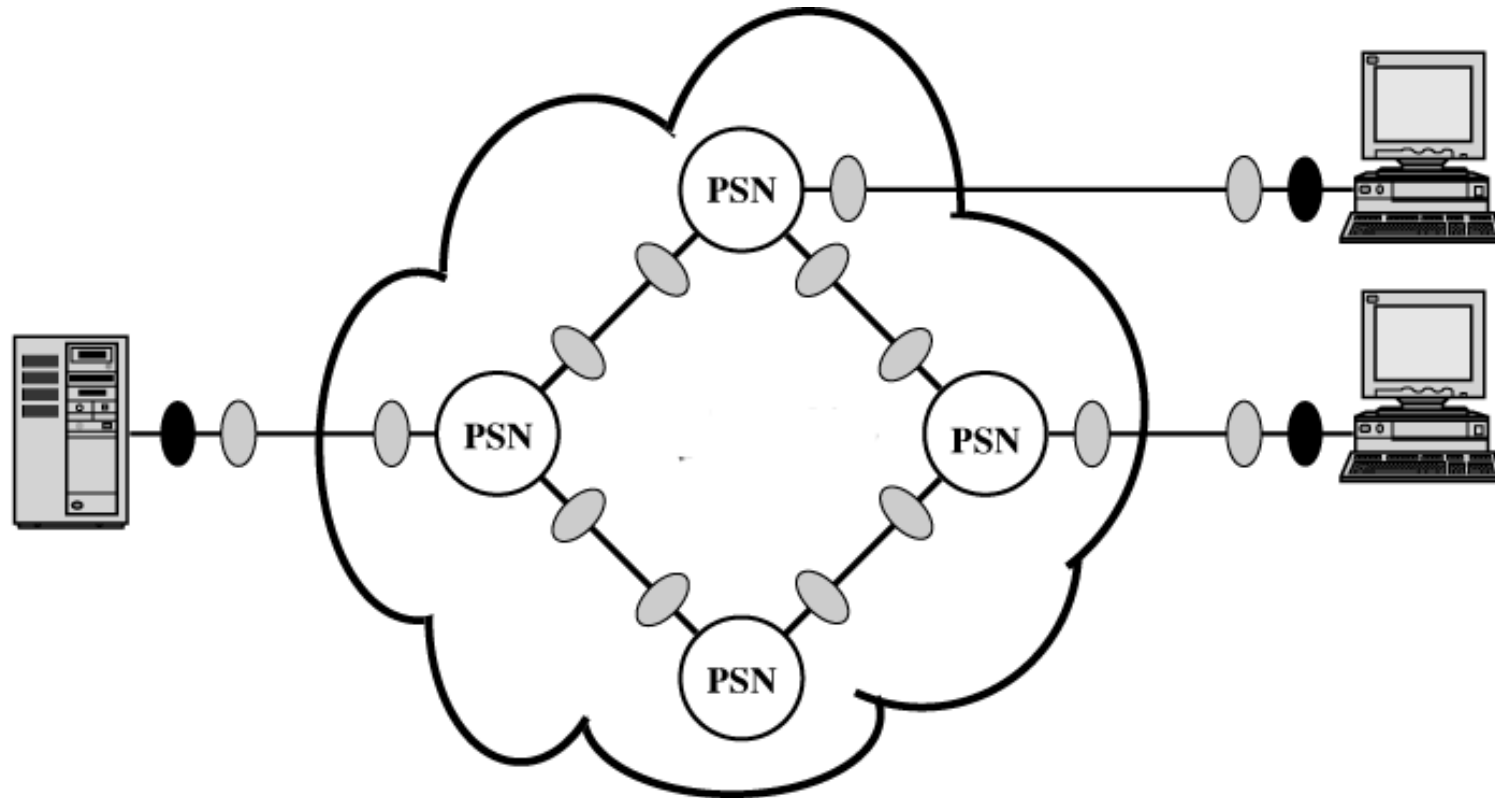
- Công cụ mã hóa được sắp đặt ở 2 đầu của mọi liên kết có nguy cơ bị tấn công.
- Đảm bảo an toàn việc lưu chuyển thông tin trên tất cả các liên kết mạng.
- Các mạng lớn cần đến rất nhiều công cụ mã hóa.
- Cần cung cấp rất nhiều khóa.
- Nguy cơ bị tấn công tại mỗi chuyển mạch
  - Các gói tin cần được mã hóa mỗi khi đi vào một chuyển mạch gói để đọc được địa chỉ ở phần đầu
- Thực hiện ở tầng vật lý hoặc tầng liên kết

# Mã hóa đầu cuối



- Quá trình mã hóa được thực hiện ở 2 hệ thống đầu cuối
- Đảm bảo an toàn dữ liệu người dùng
- Chỉ cần một khóa cho 2 đầu cuối
- Đảm bảo xác thực ở mức độ nhất định
- Mẫu lưu chuyển thông tin không được bảo vệ
  - Các phần đầu gói tin cần được truyền tải tường minh
- Thực hiện ở tầng mạng trở lên
  - Càng lên cao càng ít thông tin cần mã hóa, càng an toàn nhưng càng phức tạp với nhiều thực thể, khóa.

# Kết hợp các phương án mã hóa



● Công cụ mã hóa đầu cuối

○ Công cụ mã hóa liên kết

PSN : Packet-switching node

## 3.8 Quản lý và phân phối khóa

- Phân phối khóa an toàn đến các bên truyền tin:
  - Thường hệ thống mất an toàn là do không quản lý tốt việc phân phối khóa bí mật
- Phân cấp khóa
  - Khóa phiên (tạm thời)
    - Dùng mã hóa dữ liệu trong một phiên kết nối
    - Hủy bỏ khi hết phiên
  - Khóa chủ (lâu dài)
    - Dùng để mã hóa các khóa phiên, đảm bảo phân phối chúng một cách an toàn

# Các cách phân phối khóa



- Khóa được chọn bởi bên A, gửi theo đường vật lý đến bên B.
- Khóa được chọn bởi một bên thứ ba, gửi theo đường vật lý đến A và B.
- Nếu A và B có một khóa dùng chung: một bên có thể gửi khóa mới đến bên kia, sử dụng khóa cũ để mã hóa khóa mới.
- Nếu A và B đều có kênh mã hóa đến bên thứ ba C: C gửi khóa theo các kênh mã hóa đó đến A và B

# Phân phối khóa tự động

1. Host gửi gói tin yêu cầu kết nối
2. FEP đệm gói tin; hỏi KDC khóa phiên
3. KDC phân phối khóa phiên đến 2 host
4. Gói tin đệm được truyền đi

**FEP = Front End Processor**

**KDC = Key Distribution Center**

