An toàn Máy tính

Chủ đề: DES

1. Mật mã học
2. Mã khóa bí mật
3. DES

Một số phân loại

- Dựa vào số lượng khóa
  - Hàm băm/hash: không có khóa
  - Mã khóa bí mật/secret key: một khóa
  - Mã khóa công khai/Public key : hai khóa
- Kiểu thao tác mã hóa
  - Thay thế / biến đổi / tích
- Cách xử lý bản rõ
  - Mã khối / mã dòng

Khóa mã bí mật & thuật toán

- Khó giữ bí mật nếu dùng phổ biến:
  - Lần ngược
- Thương mại: được công bố công khai
  - Xem xét bởi nhiều tổ chức/cá nhân, tin cậy
- Quân sự: tránh lộ các thông tin, ý tưởng cho đối phương

Sơ đồ thám mã

- Chỉ biết bản mã / Ciphertext only:
  - Tìm đến khi được bản giải có nghĩa
  - Nói chung cần bản mã tương đối dài
- Biết bản rõ / Known plaintext:
  - Có thể truy cập bộ mã hóa, từ đó có các cặp <bản mã, bản rõ> tương ứng
  - Khá hiệu quả với hệ mã đơn ký tự
- Chosen plaintext:
  - Choose text, get encrypted
  - Useful if limited set of messages

Bảo mật tính toán & vô điều kiện

- Bảo mật vô điều kiện
  - Với năng lực tính toán vô hạn, hệ mã không bị phá vỡ
  - Từ bản mã, không trích rút được thông tin về bản rõ tương ứng
  - Vd sơ đồ một lần
- Bảo mật tính toán
  - Chi phí để phá mã lớn hơn giá trị của thông tin
  - Số lượng phép toán để phá mã rất lớn
  - Thời gian cần để phá mã vượt quá thời hạn hiệu lực/có ý nghĩa của thông tin

Brute Force

- Luôn có phương án phá mã: thử tất cả các khóa có thể
- Phương án tấn công cơ bản, tỷ lệ với kích thước khóa
- Giả định nhận biết được bản rõ

Key Size (bits)	Number of Alternative Keys	Time required at 1 encryption/μs	Time required at 10^6 encryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\ \mu s = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\ \mu s = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\ \mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\ \mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\ \mu s = 6.4 \times 10^{12}$ years	6.4×10^6 years

Thời gian tìm kiếm trung bình

Kích thước khóa (bit)	Số lượng khóa	Thời gian cần thiết (1 giải mã/μs)	Thời gian cần thiết (10^6 giải mã/μs)
32	$2^{32} = 4,3 \times 10^9$	2^{31} μs = 35,8 phút	2,15 ms
56	$2^{56} = 7,2 \times 10^{16}$	2^{55} μs = 1142 năm	10,01 giờ
128	$2^{128} = 3,4 \times 10^{38}$	2^{127} μs = $5,4 \times 10^{24}$ năm	$5,4 \times 10^{18}$ năm
168	$2^{168} = 3,7 \times 10^{50}$	2^{167} μs = $5,9 \times 10^{36}$ năm	$5,9 \times 10^{30}$ năm
26 ký tự (hoán vị)	$26! = 4 \times 10^{26}$	2×10^{26} μs = $6,4 \times 10^{12}$ năm	$6,4 \times 10^6$ năm

Khóa DES dài 56 bit
Khóa AES dài 128+ bit
Khóa 3DES dài 168 bit

Tuổi vũ trụ : ~ 10^{10} năm

Mã đơn ký tự - Monoalphabetic

- Thay vì đơn giản dịch chuyển như mã dịch chuyển

Có thể hoán vị (shuffle) các ký tự tùy ý

- Mỗi ký tự bản rõ ánh xạ với 1 ký tự bất kỳ khác nhau thuộc bản mã
- Khóa có thể xem như xâu 26 ký tự

Ví dụ

```
Plain:  abcdefghijklmnopqrstuvwxyz
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN
Plaintext:  ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA
```

Bảo mật của mã đơn ký tự

- Phân tích sơ bộ: số lượng khóa bây giờ là ?

$26! = 4 \times 10^{26}$ keys

- Có an toàn không ? (so sánh với mã chuyển dịch – chỉ có 26 khóa)
- Có thể thám mã dựa vào đặc trưng ngôn ngữ
  - Ngôn ngữ nói là dư thừa
  - Các ký tự không được dùng với tần suất như nhau

One-Time Pad

- Nếu sử dụng khóa thực sự ngẫu nhiên, độ dài bằng thông điệp thì có hệ mã bảo mật, gọi là One-Time pad
- Ví dụ: chuỗi ngẫu nhiên bit 0, 1 được XOR với bản rõ, khóa không lặp lại
- Không phá được bởi vì bản mã không mang thông tin thống kê gì liên quan đến bản rõ tương ứng
- Với bản rõ tùy ý, cần khóa ngẫu nhiên cùng độ dài => khó khăn trong việc sinh số lượng khóa lớn
- Vấn đề phân phối khóa an toàn

Mã tích

- Mật mã sử dụng phép thay thế hoặc hoán vị không bảo mật vì đặc trưng ngôn ngữ
- Do đó xem xét sử dụng nhiều hệ mã để tăng độ khó phá
  - 2 phép thay thế tạo nên phép thay thế phức tạp hơn
  - 2 phép hoán vị tạo nên phép hoán vị phức tạp hơn
  - Thay thế kết hợp hoán vị tạo nên mã an toàn hơn nữa
- Điều này là cầu nối chuyển tiếp mã cổ điển với mã hiện đại

1. Mật mã học
2. Mã khóa bí mật
3. DES

Mã khối (block) và mã dòng (stream)

- Mã khối xử lý thông điệp theo từng khối, mỗi khối được mã/giải mã
- Hình dung tương tự như là phép thế trên ký tự kích cỡ lớn (64-bits hoặc hơn)
- Mã dòng lại xử lý thông điệp theo từng bit/byte khi mã/giải mã
- Mã khối chiếm phần lớn trong các hệ mật mã hiện nay

Nguyên lý trong mã khối

- Phần lớn hệ mã khối đối xứng dựa trên cấu trúc mã **Feistel**
- Có thể xem mã khối như là phép thay thế kích cỡ lớn
- Nếu thiết lập đủ, cần bảng với 2^{64} dòng cho khối 64-bit
- Thay vào đó: tạo thành từ các khối bé hơn
- Áp dụng mã tích

Mã thay thế-hoán vị

- Đề xuất của Shannon kết nối phép toán thay thế-hoán vị

Substitution-permutation (S-P) networks [Shannon, 1949]

Các hệ mã hiện đại với tích phép toán thay thế-hoán vị

- Là ý tưởng cho xây dựng các hệ mã hiện đại
- S-P networks dựa trên 2 thao tác mã cơ bản
  - *Thay thế* (S-box)
  - *Hoán vị* (P-box)
- Làm biến đổi thông điệp theo 2 tác động
  - *Rối loạn (confusion)*
  - *Khuếch tán (diffusion*)

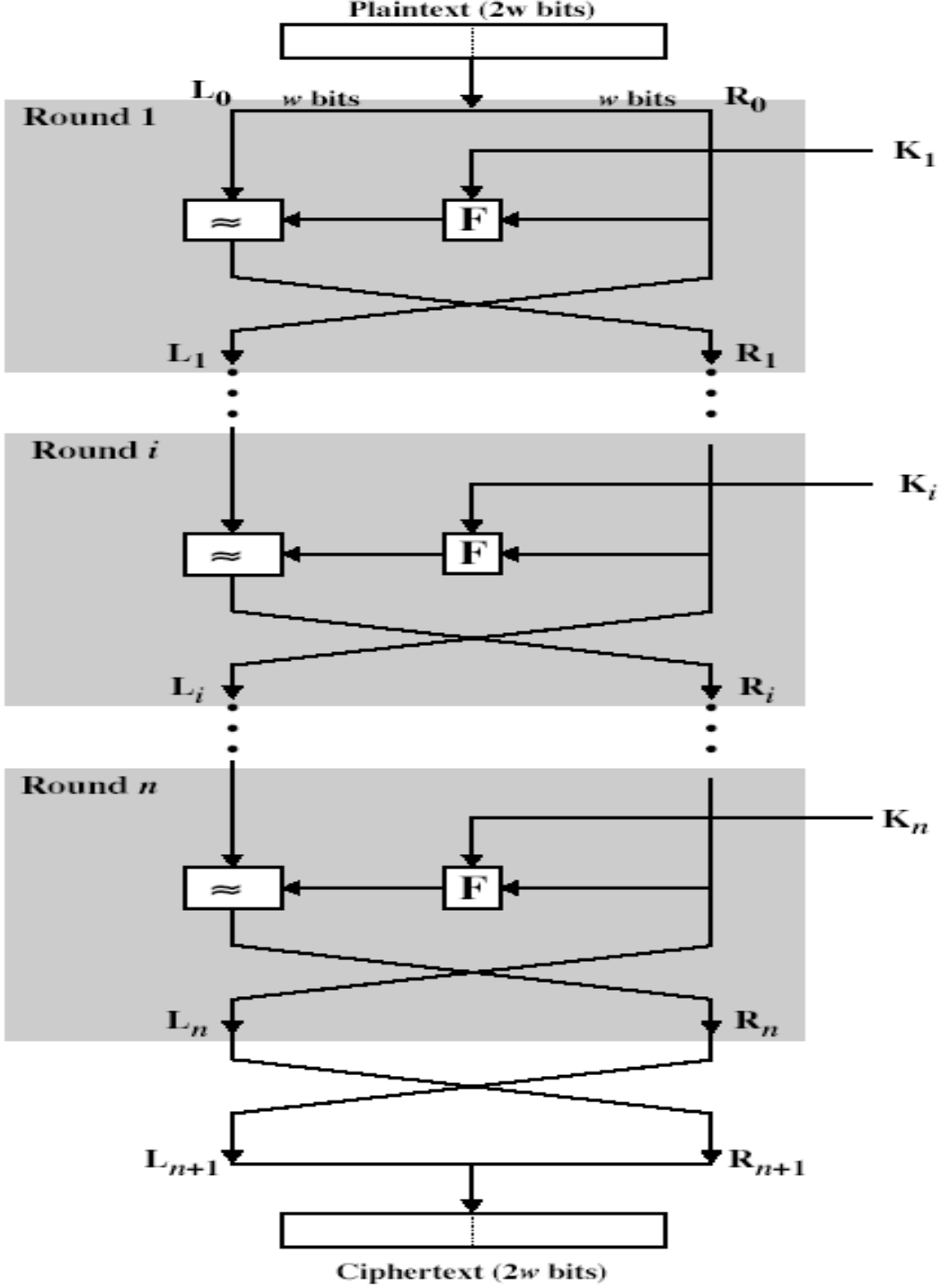
Rối loạn và khuếch tán

- Mã hóa cần làm mờ/rối tung hoàn toàn các tính chất thống kê của thông điệp gốc
- Cơ chế one-time pad có thể đáp ứng
- Cơ chế sử dụng hiệu quả trong thực tế là S-P networks:
  - **Khuếch tán (Diffusion)** – xóa đi cấu trúc thống kê của bản rõ truyền sang bản mã
  - **Làm rối (Confusion)** – làm cho các liên hệ giữa bản mã và khóa phức tạp nhất có thể

Cấu trúc mã Feistel

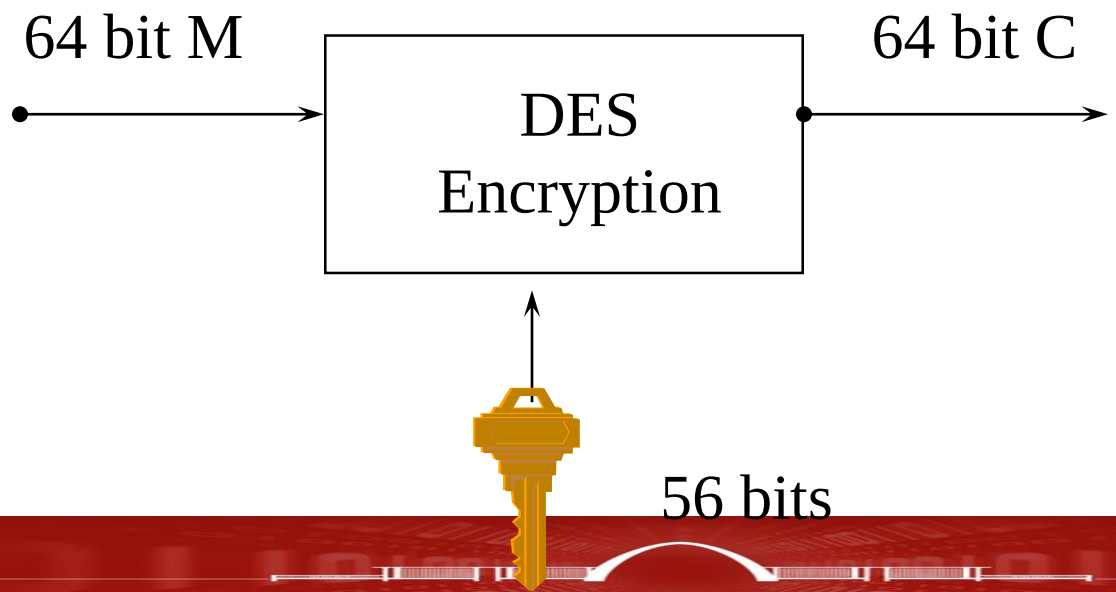
- **Mã Feistel** đã cài đặt ý tưởng S-P network của Shannon
  - Dựa trên mã tích có thể đảo ngược
- Quá trình xử lý dùng nhiều vòng lặp
  - Phân chia khối đầu vào (input) thành 2 nửa
  - Thực hiện thay thế (substitution) với nửa bên trái
  - Dựa trên hàm quay vòng với nửa bên phải và khóa phiên
  - Hoán vị các nửa

Cấu trúc Feistel

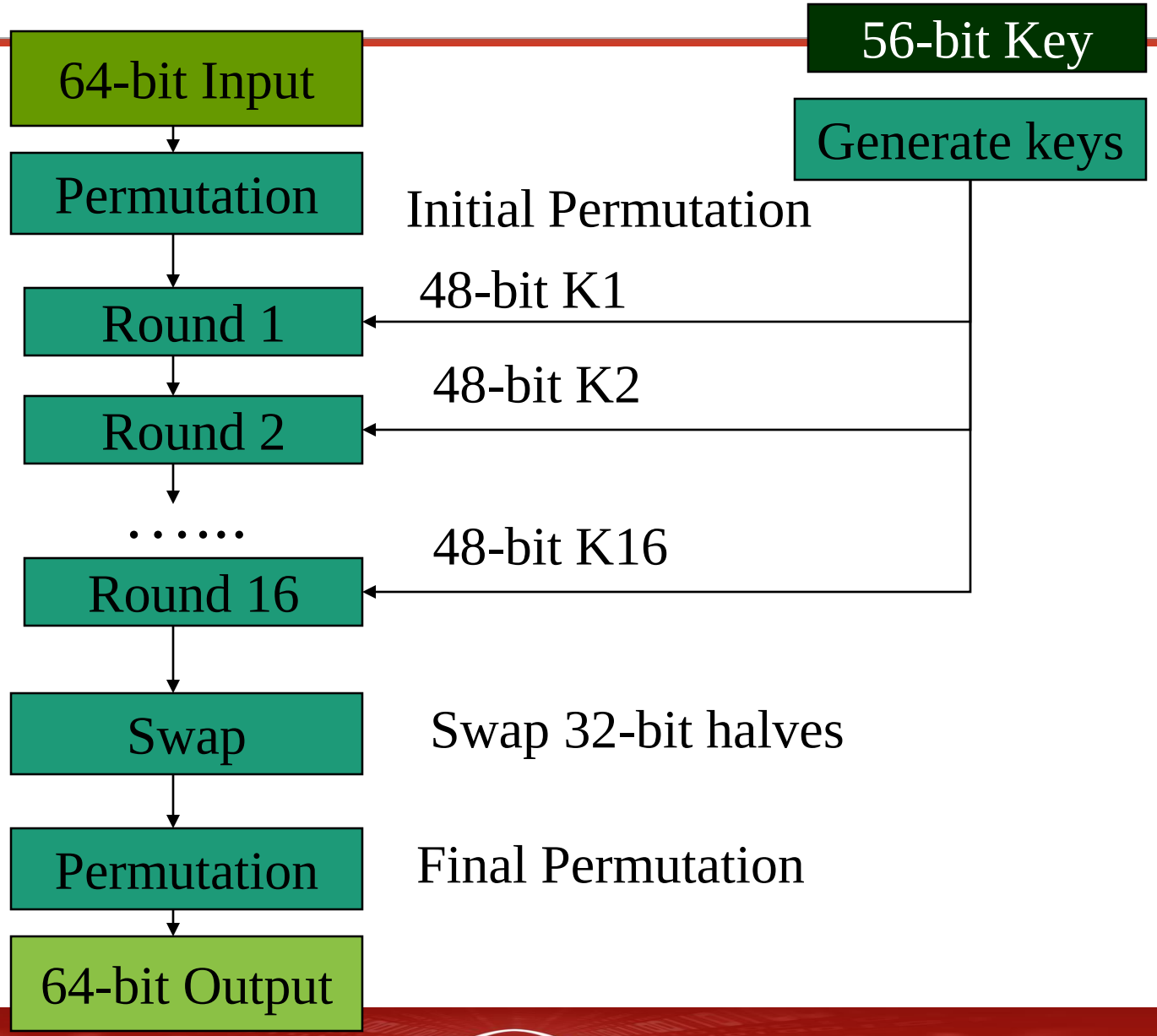


DES (Data Encryption Standard)

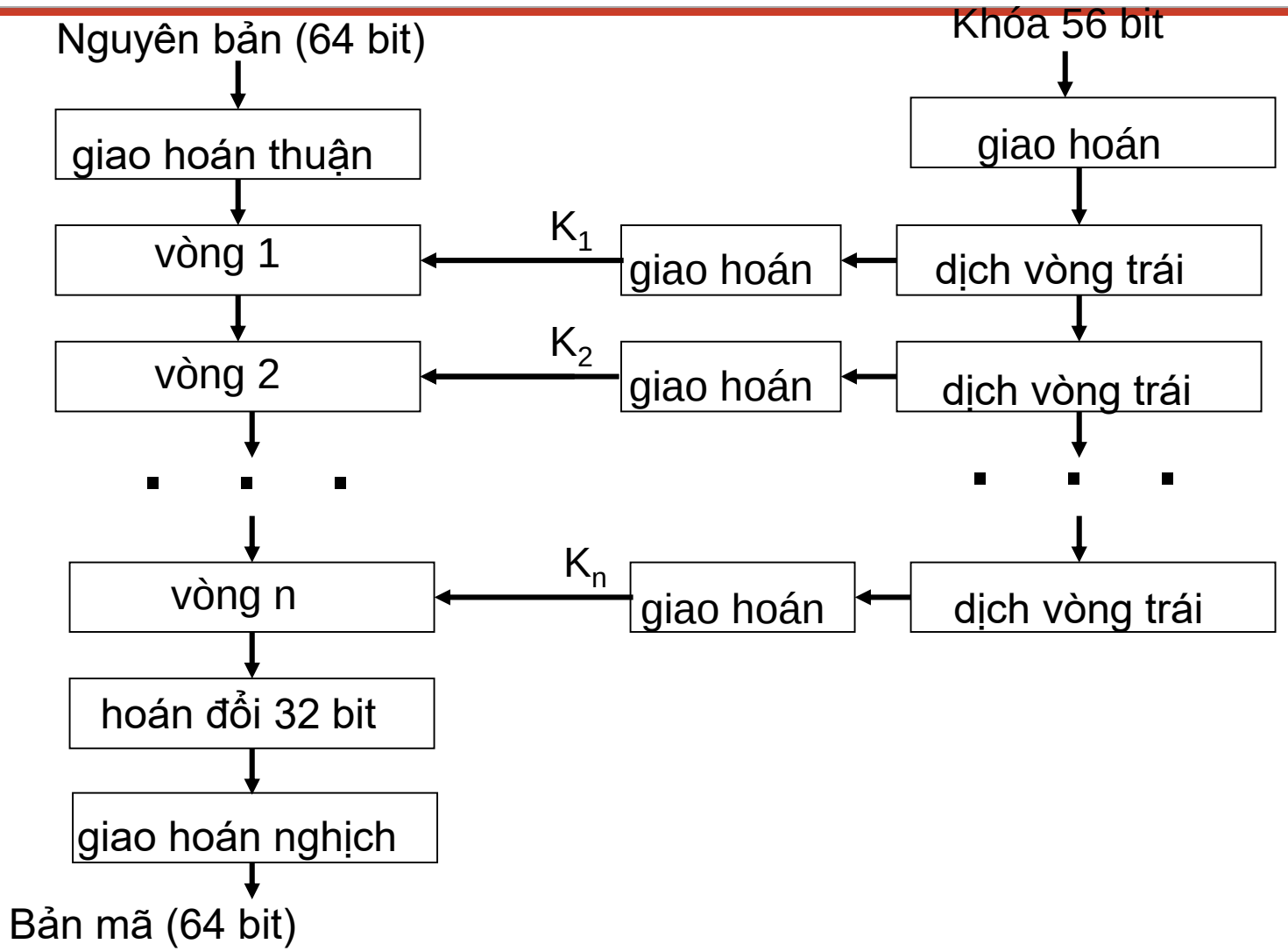
- Công bố 1977, chuẩn hóa 1979.
- Khóa : 64 bit (gồm 8-bit kiểm tra lỗi và 56-bit khóa)
  - Các bit thứ 8 của mỗi byte là parity bit.
- Đầu vào: xâu bản rõ 64 bit
- Đầu ra: xâu bản mã 64 bit.



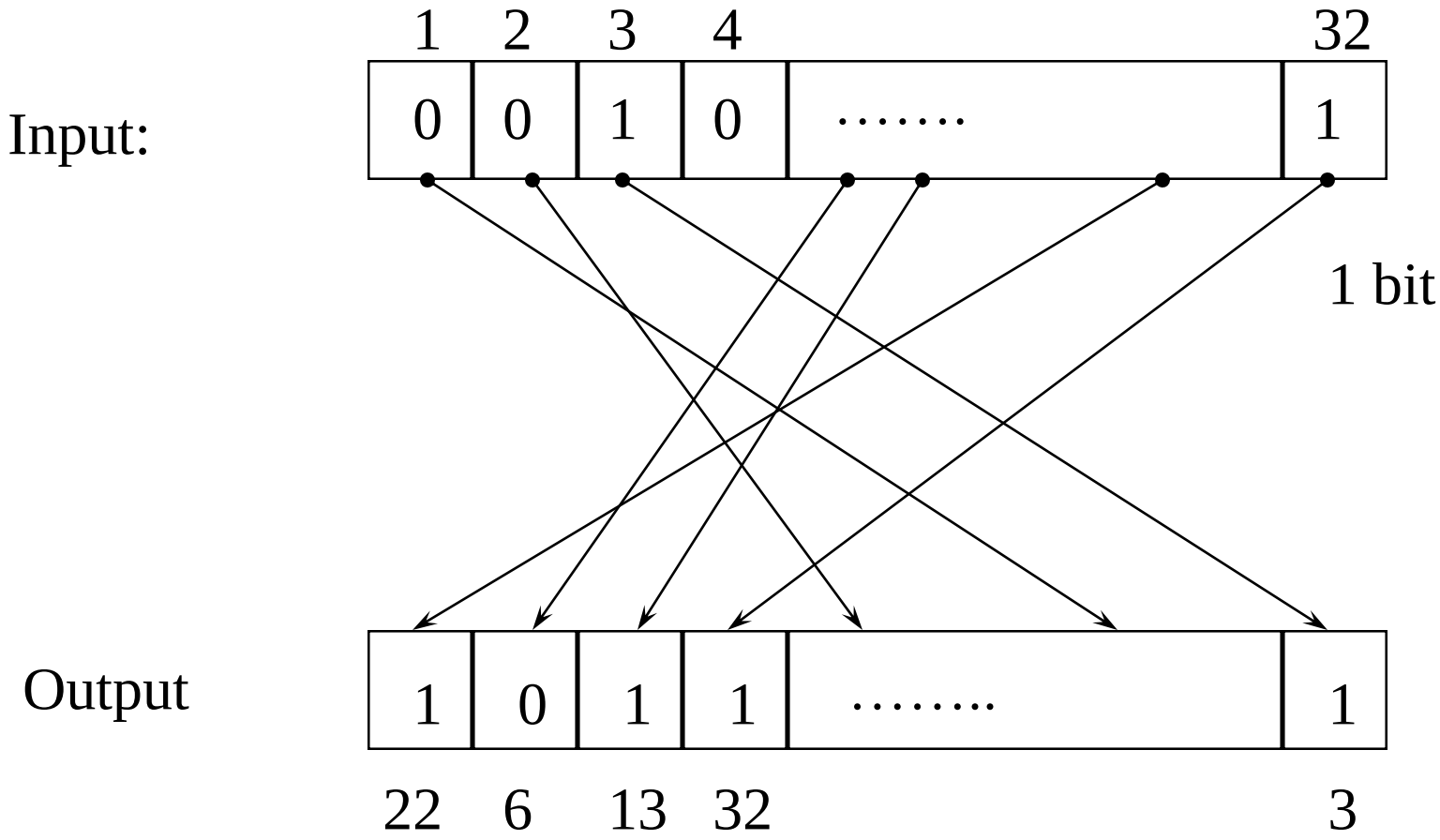
Mô tả mức cao của DES



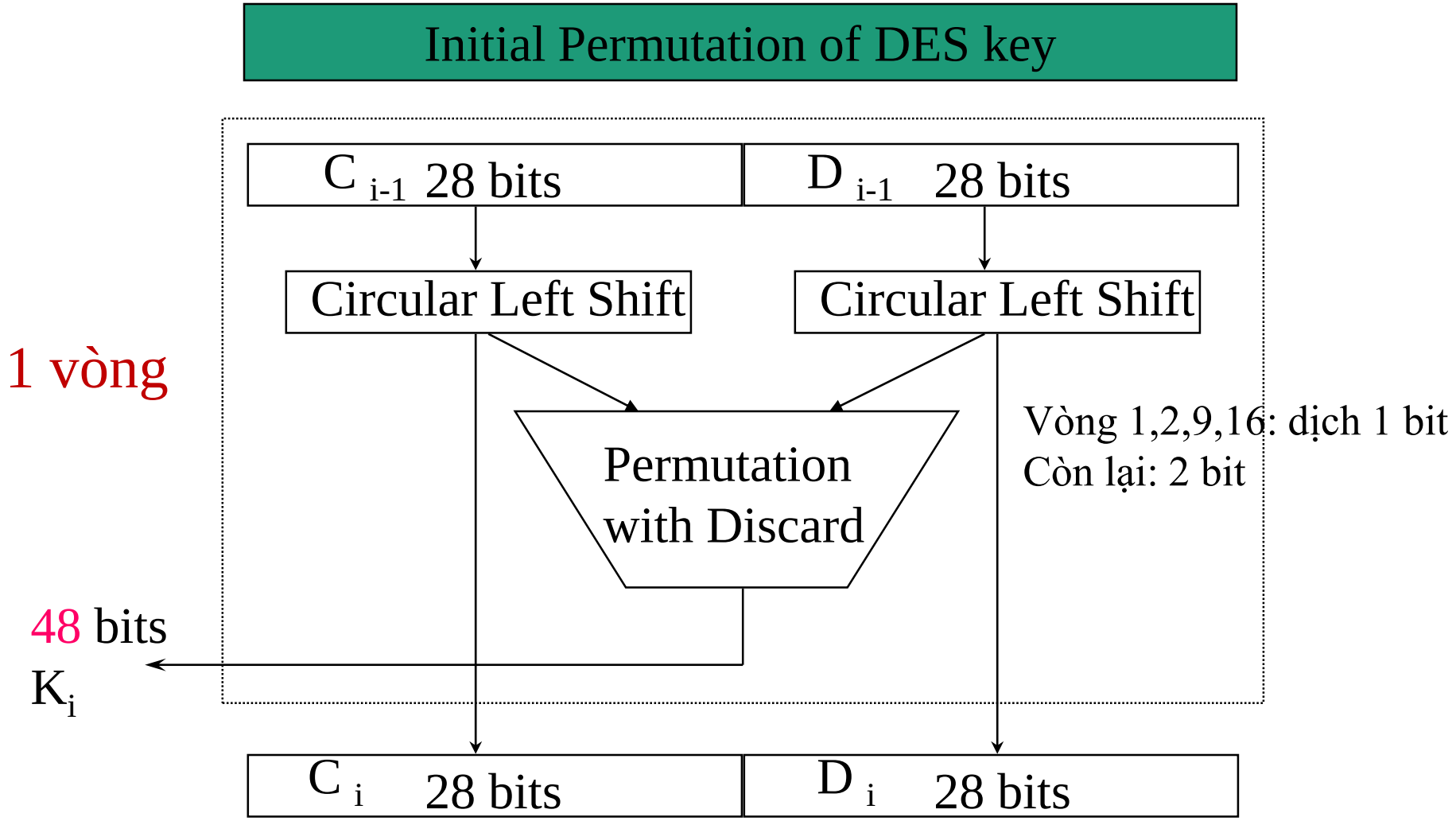
Giải thuật mã hóa DES



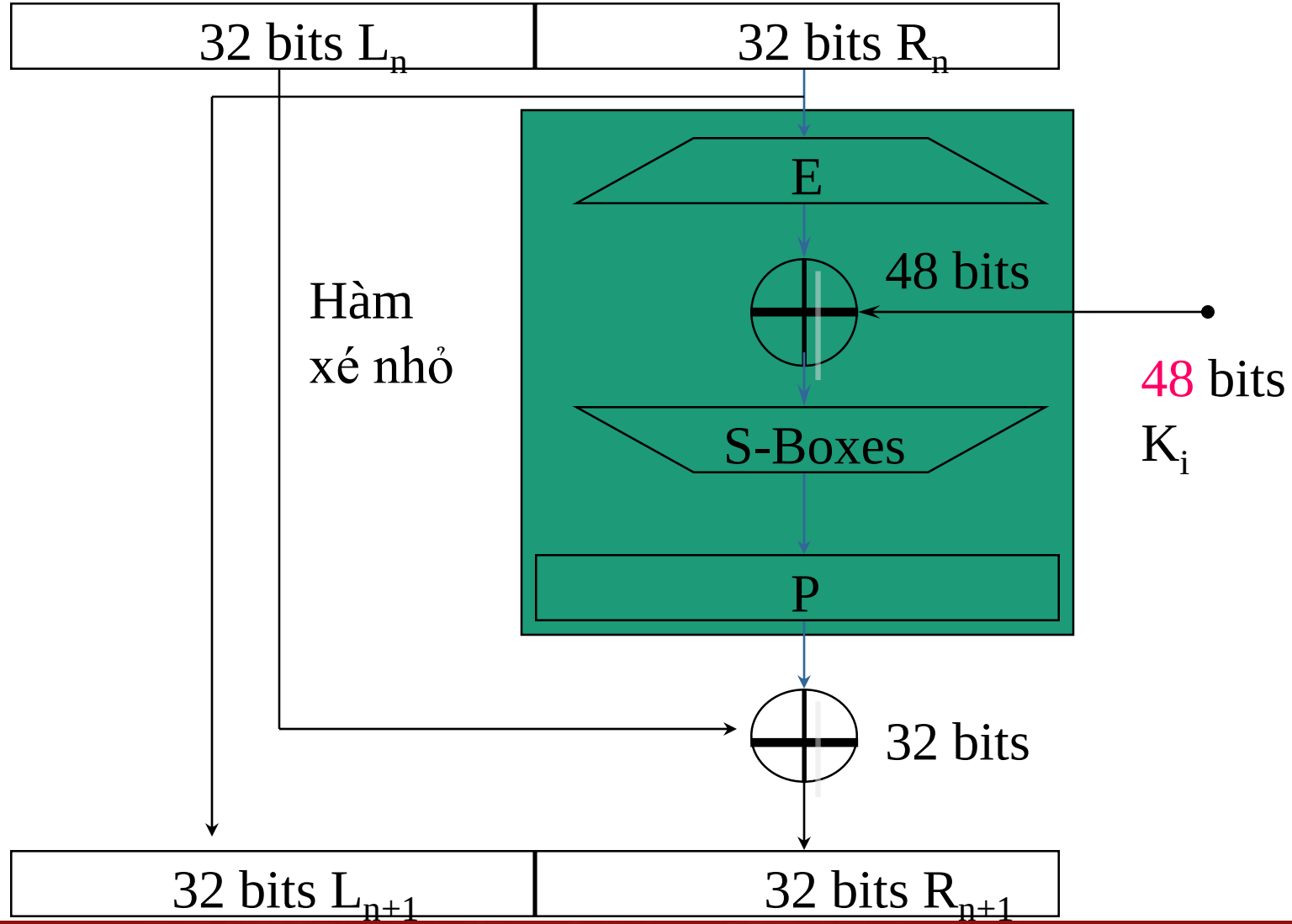
Hoán vị bit (1-to-1)



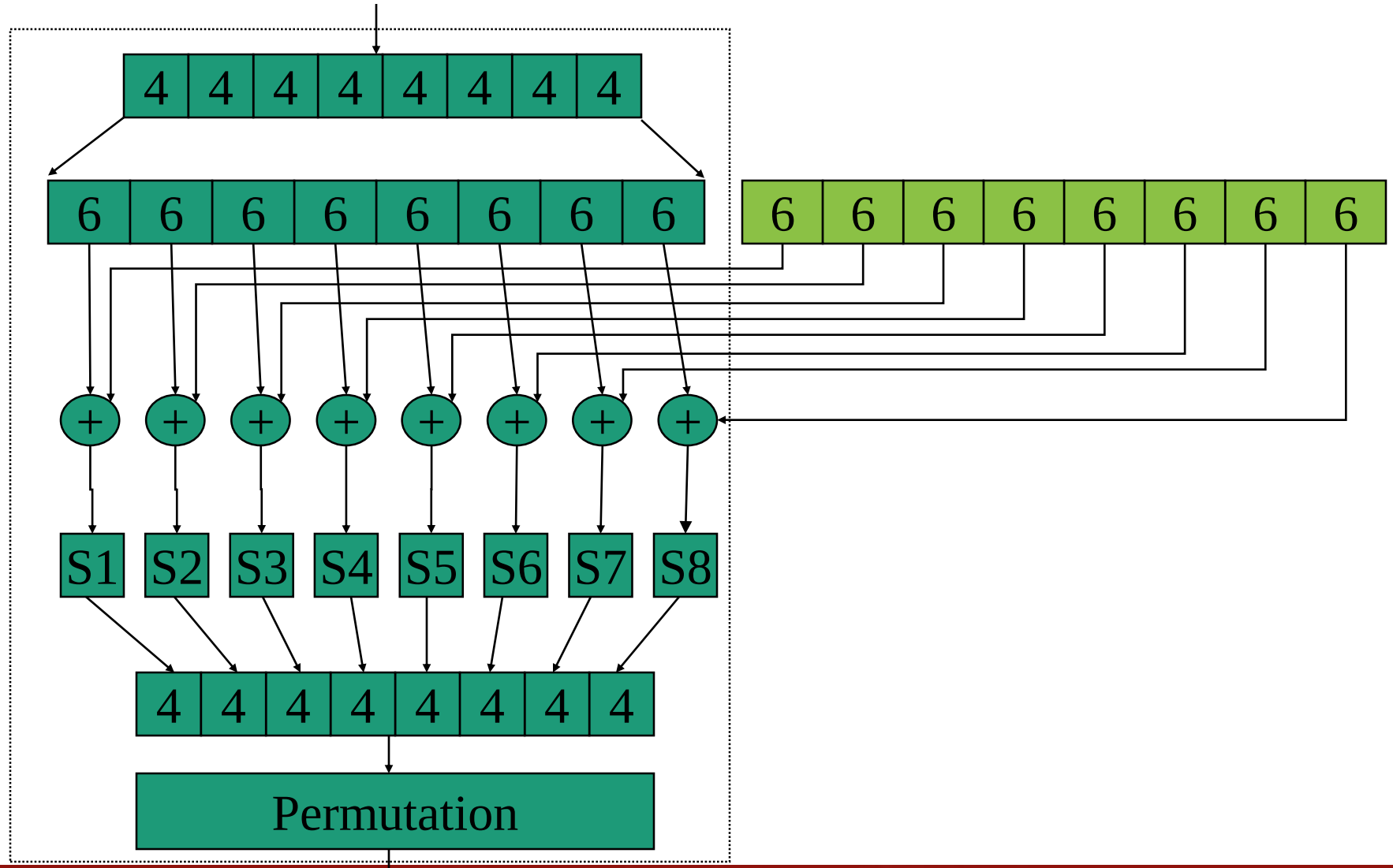
Sinh khóa phiên



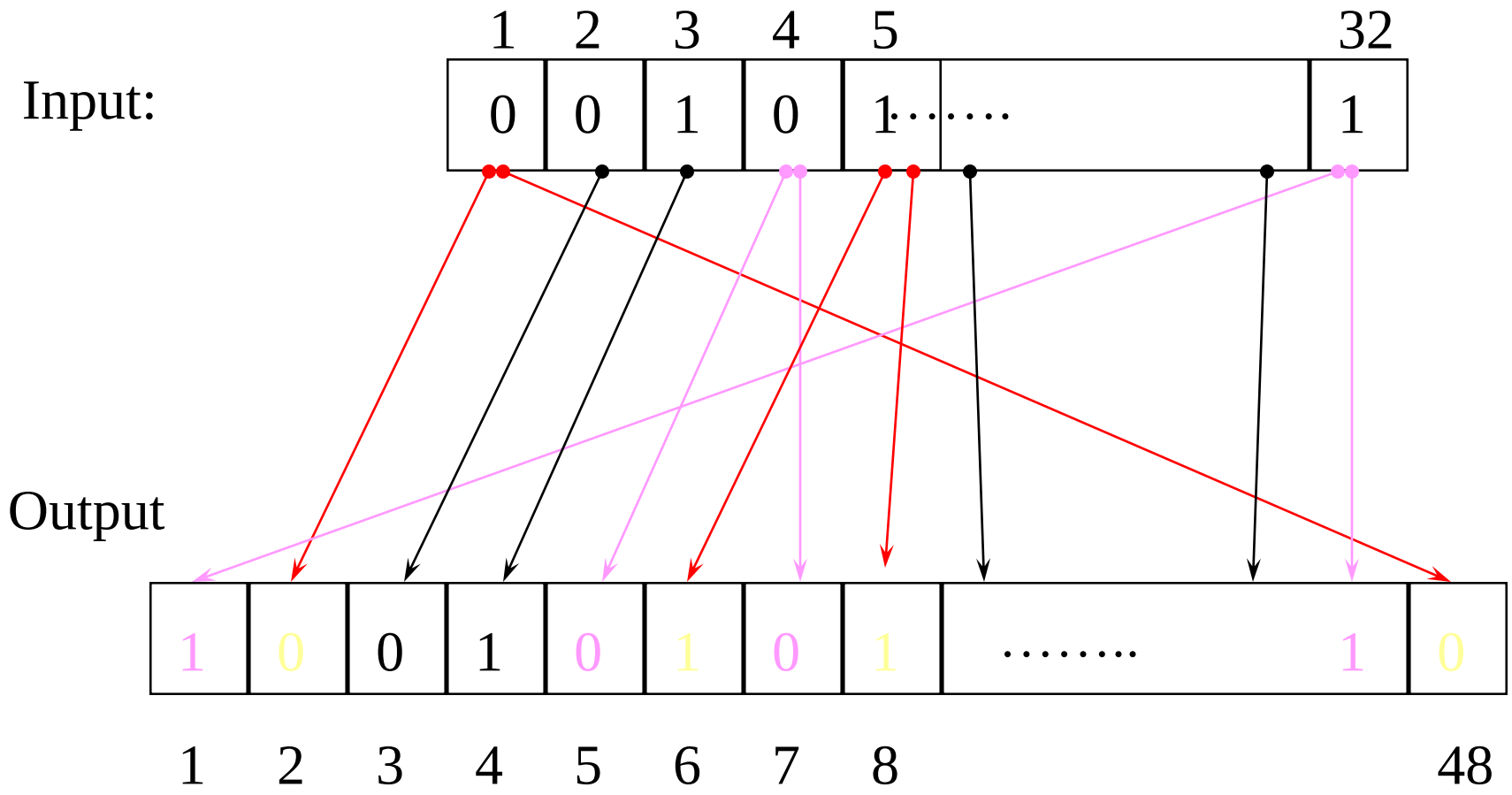
Một vòng lặp DES



Hàm xé nhỏ

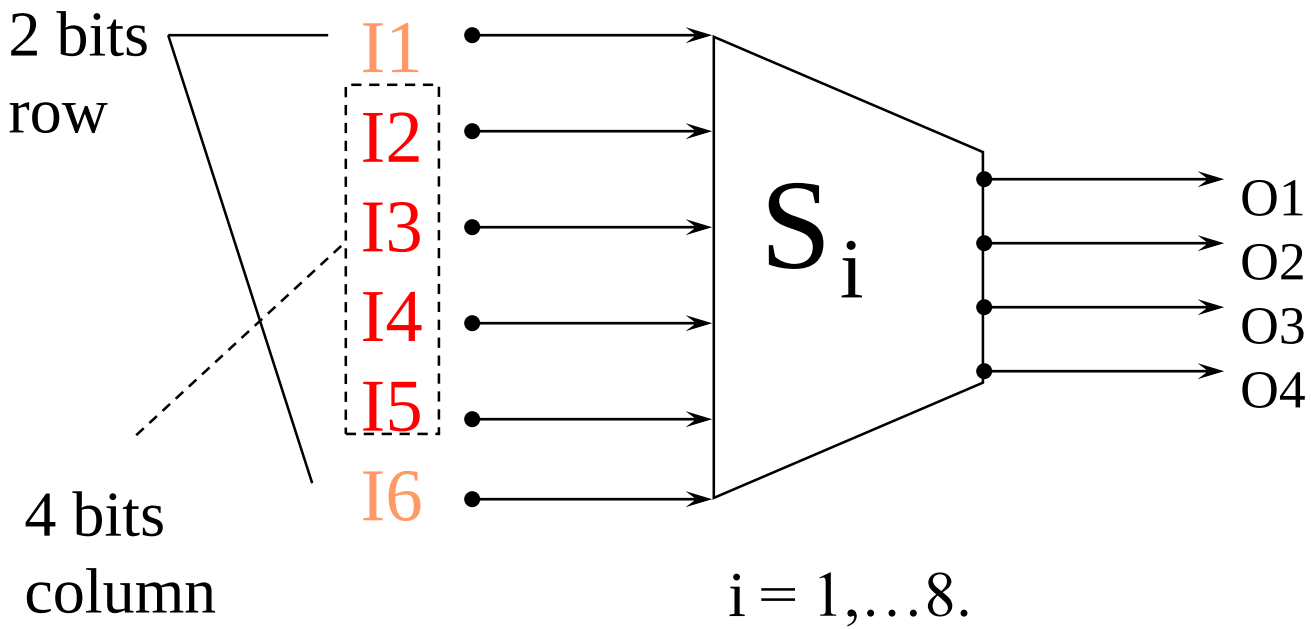

4 4 4 4 4 4 4 4
6 6 6 6 6 6 6 6
6 6 6 6 6 6 6 6
+ + + + + + + +
S1 S2 S3 S4 S5 S6 S7 S8
4 4 4 4 4 4 4 4
Permutation

Mở rộng bit (1-m)



Khối S-Box

- Khối 48 bits => khối 32 bits. (8*6 ==> 8*4)



Ví dụ S-Box

- Mỗi dòng: 1 hoán vị của bộ (0..15)

	0	1	2	3	4	5	6	7	8	9…. 15
0	14	4	13	1	2	15	11	8	3	
1	0	15	7	4	14	2	13	1	10	
2	4	1	14	8	13	6	2	11	15	
3	15	12	8	2	4	9	1	7	5	

Ví dụ: input: 100110 output: ???

Phá mã DES

- Khóa 56 bit có $2^{56} = 7,2 \times 10^{16}$ giá trị có thể
- Phương pháp vét cạn ?
- Tốc độ tính toán cao có thể phá được khóa
  - 1997 : 70000 máy tính phá mã DES trong 96 ngày
  - 1998 : Electronic Frontier Foundation (EFF) phá mã DES bằng máy chuyên dụng (250000$) trong < 3 ngày
- Vấn đề còn phải nhận biết được nguyên bản
- Nếu cần an ninh hơn : 3DES hay chuẩn mới AES

Hệ mã hóa 3DES

- Sử dụng 3 khóa và chạy 3 lần giải thuật DES
  - Mã hóa : $C = E_{K_3}[D_{K_2}[E_{K_1}[p]]]$
  - Giải mã : $p = D_{K_1}[E_{K_2}[D_{K_3}[C]]]$
- Độ dài khóa thực tế là 168 bit
  - Không tồn tại $K_4 = 56$ sao cho $C = E_{K_4}(p)$

Chuẩn mã hóa tiên tiến

- AES (Advanced Encryption Standard) được công nhận chuẩn mới năm 2001
- Tên giải thuật là Rijndael (Rijmen + Daemen)
- An ninh hơn và nhanh hơn 3DES
- Kích thước khối : 128 bit
- Kích thước khóa : 128/192/256 bit
- Số vòng : 10/12/14

Các hệ mã hóa khối khác (1)

- 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 ninh
  - 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 ninh, khá nhanh và gọn nhẹ
  - Tự do sử dụng

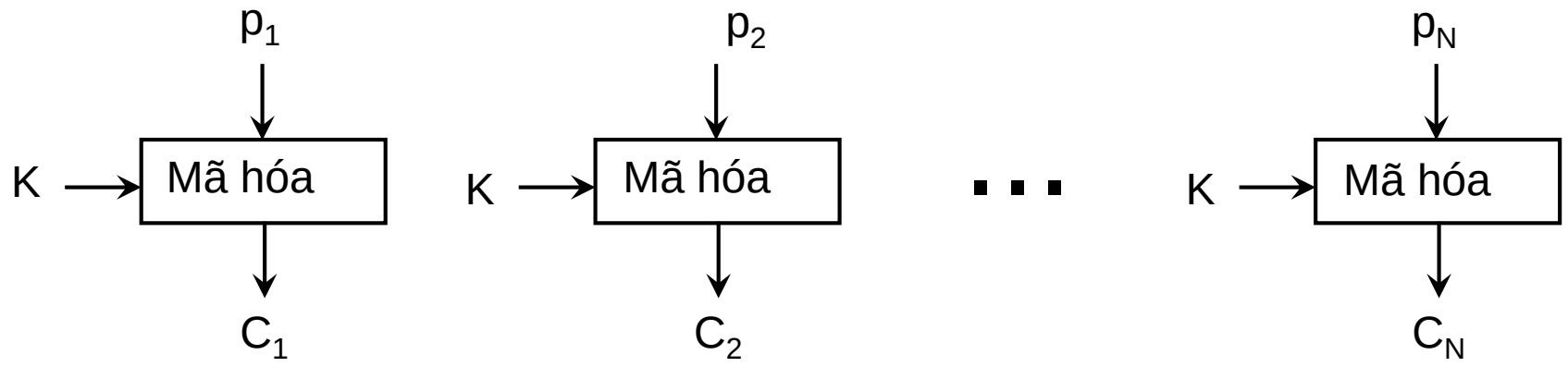
Các hệ mã hóa khối khác (2)

- 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

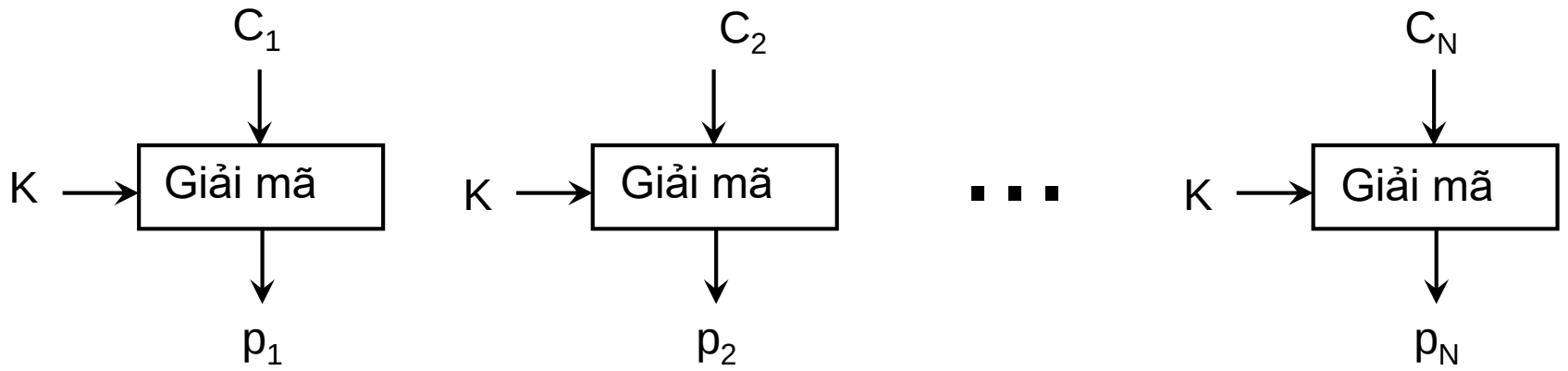
Các phương thức mã hóa khối

- 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
- OFB (Output Feeback)
  - 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



Mã hóa

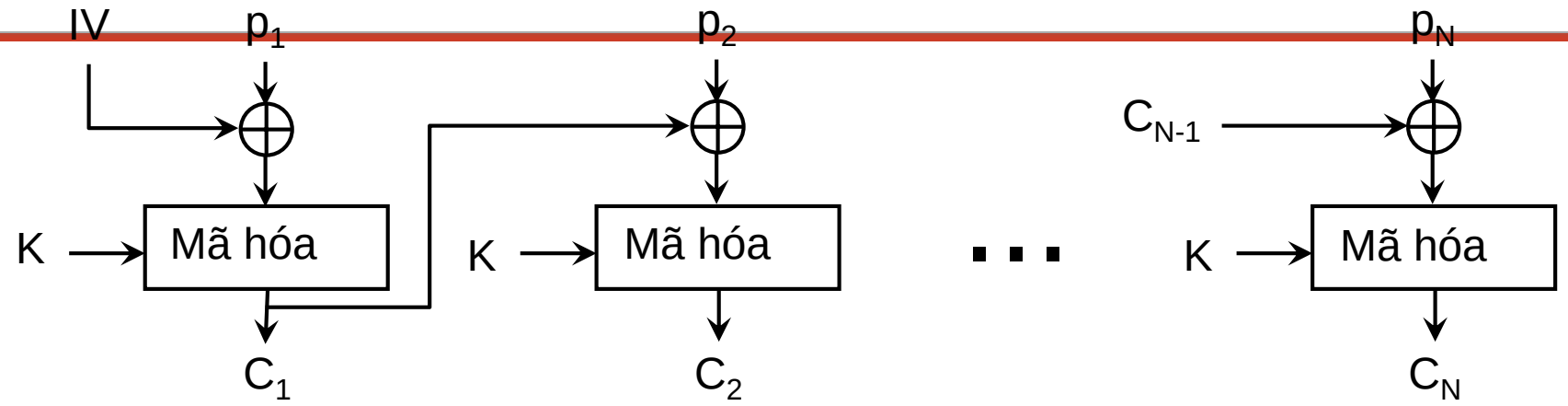


Giải mã

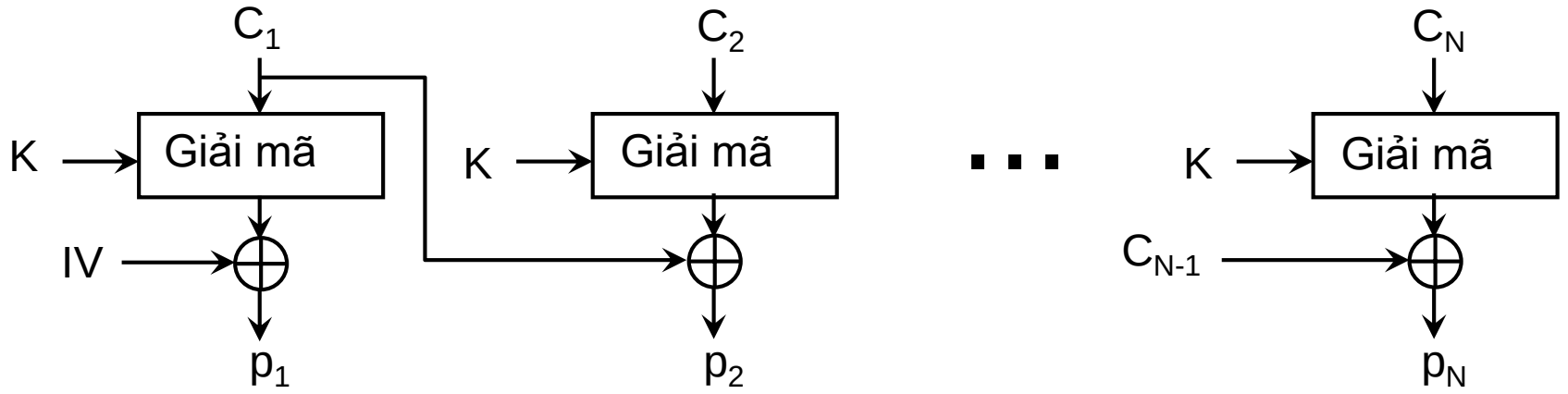
Đánh giá ECB

- 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



Mã hóa

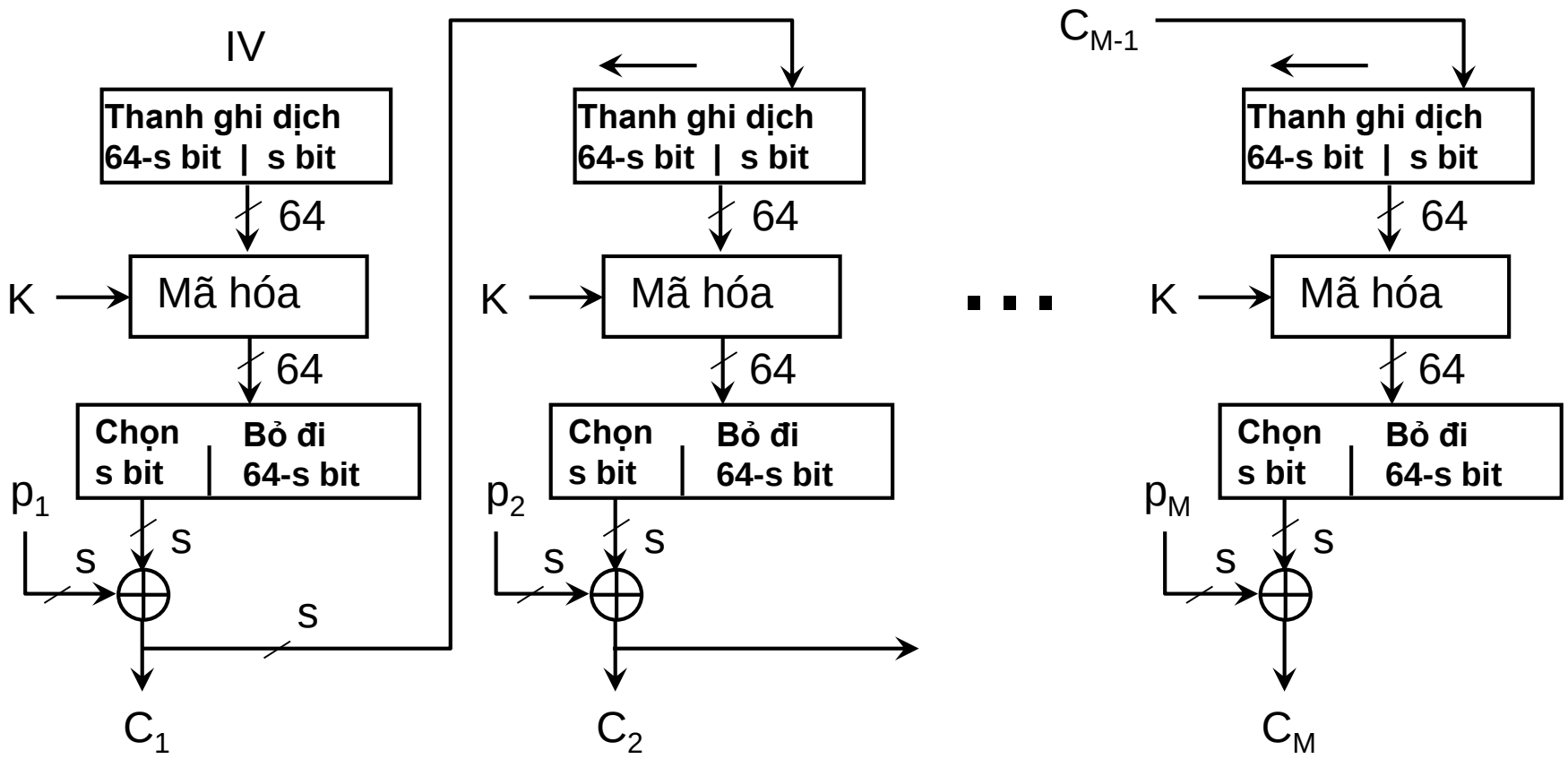


Giải mã

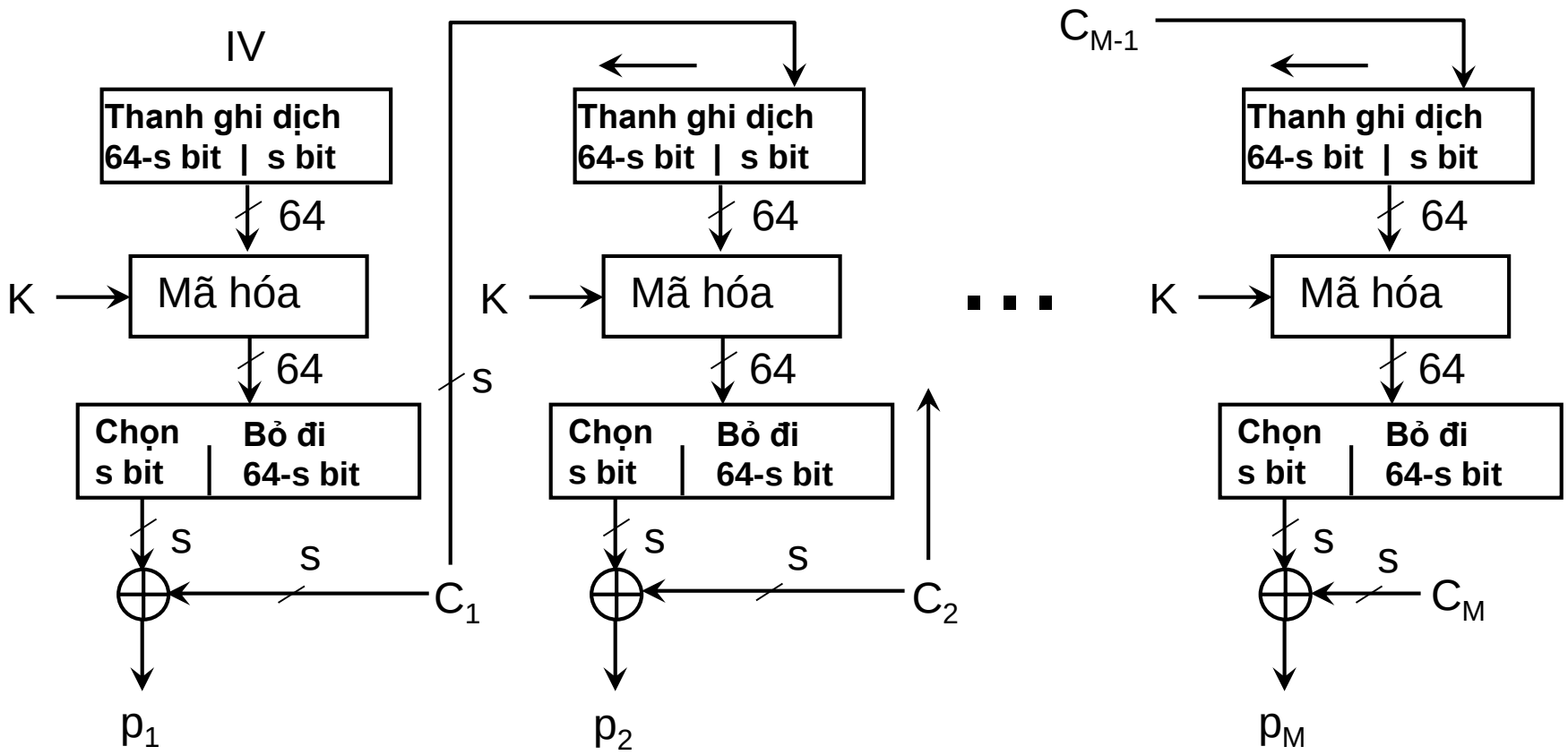
Đánh giá CBC

- 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


IV
Thanh ghi dịch
64-s bit | s bit
64
K
Mã hóa
64
Chọn
s bit
Bỏ đi
64-s bit
p1
s
s
s
C1
Thanh ghi dịch
64-s bit | s bit
64
K
Mã hóa
64
Chọn
s bit
Bỏ đi
64-s bit
p2
s
s
C2
CM-1
Thanh ghi dịch
64-s bit | s bit
64
K
Mã hóa
64
Chọn
s bit
Bỏ đi
64-s bit
pM
s
s
CM

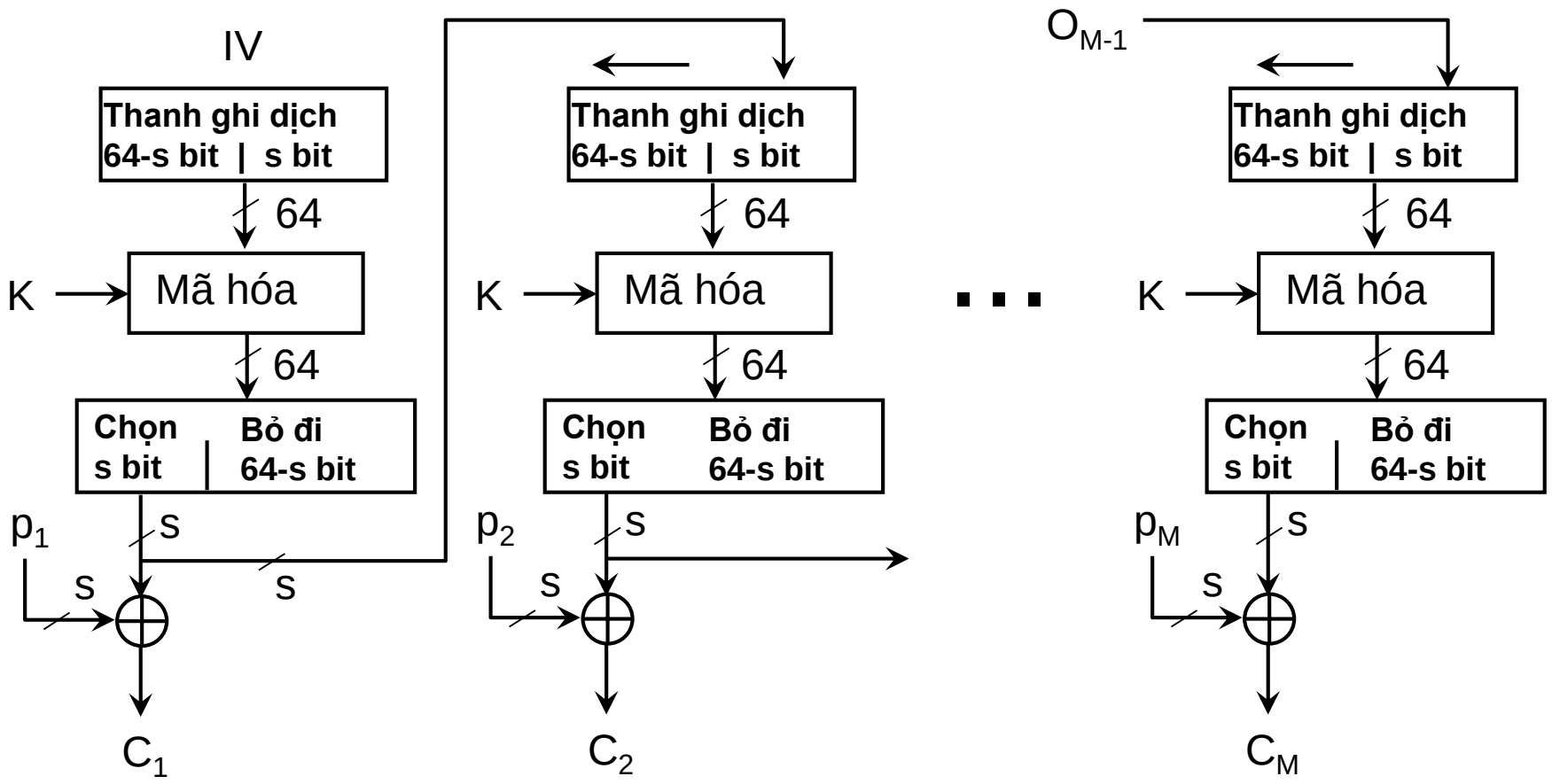
Giải mã CFB


IV
Thanh ghi dịch
64-s bit | s bit
64
K
Mã hóa
64
Chọn
s bit
Bỏ đi
64-s bit
s
s
s
C1
p1
Thanh ghi dịch
64-s bit | s bit
64
K
Mã hóa
64
Chọn
s bit
Bỏ đi
64-s bit
s
s
C2
p2
CM-1
Thanh ghi dịch
64-s bit | s bit
64
K
Mã hóa
64
Chọn
s bit
Bỏ đi
64-s bit
s
s
CM
pM

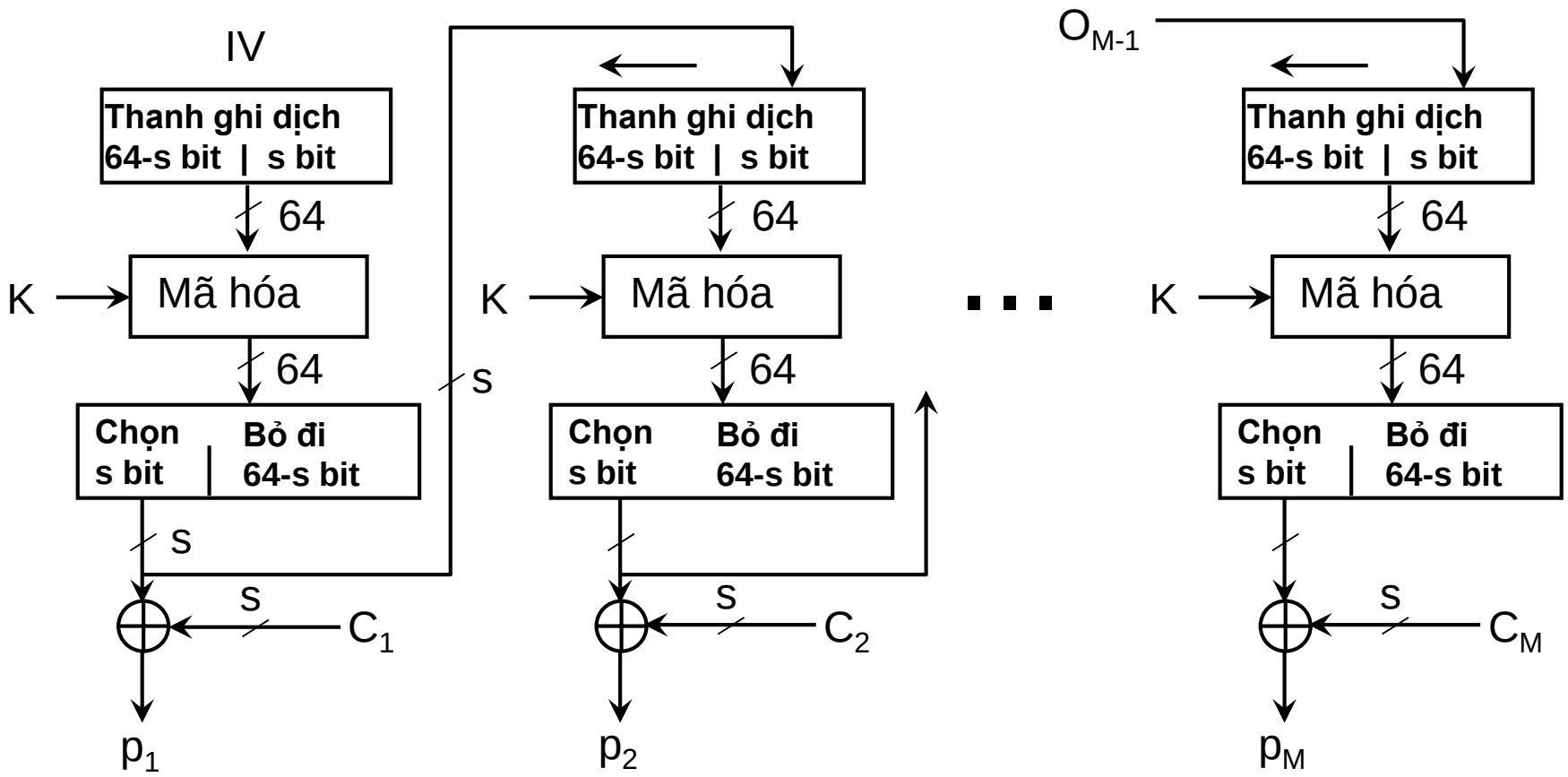
Đánh giá CFB

- Thích hợp khi dữ liệu nhận được theo từng đơn vị bit hay byte
- Không cần độn 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


IV
Thanh ghi dịch
64-s bit | s bit
64
K
Mã hóa
64
Chọn
s bit
Bỏ đi
64-s bit
p1
s
s
s
C1
Thanh ghi dịch
64-s bit | s bit
64
K
Mã hóa
64
Chọn
s bit
Bỏ đi
64-s bit
p2
s
s
C2
OM-1
Thanh ghi dịch
64-s bit | s bit
64
K
Mã hóa
64
Chọn
s bit
Bỏ đi
64-s bit
pM
s
s
CM

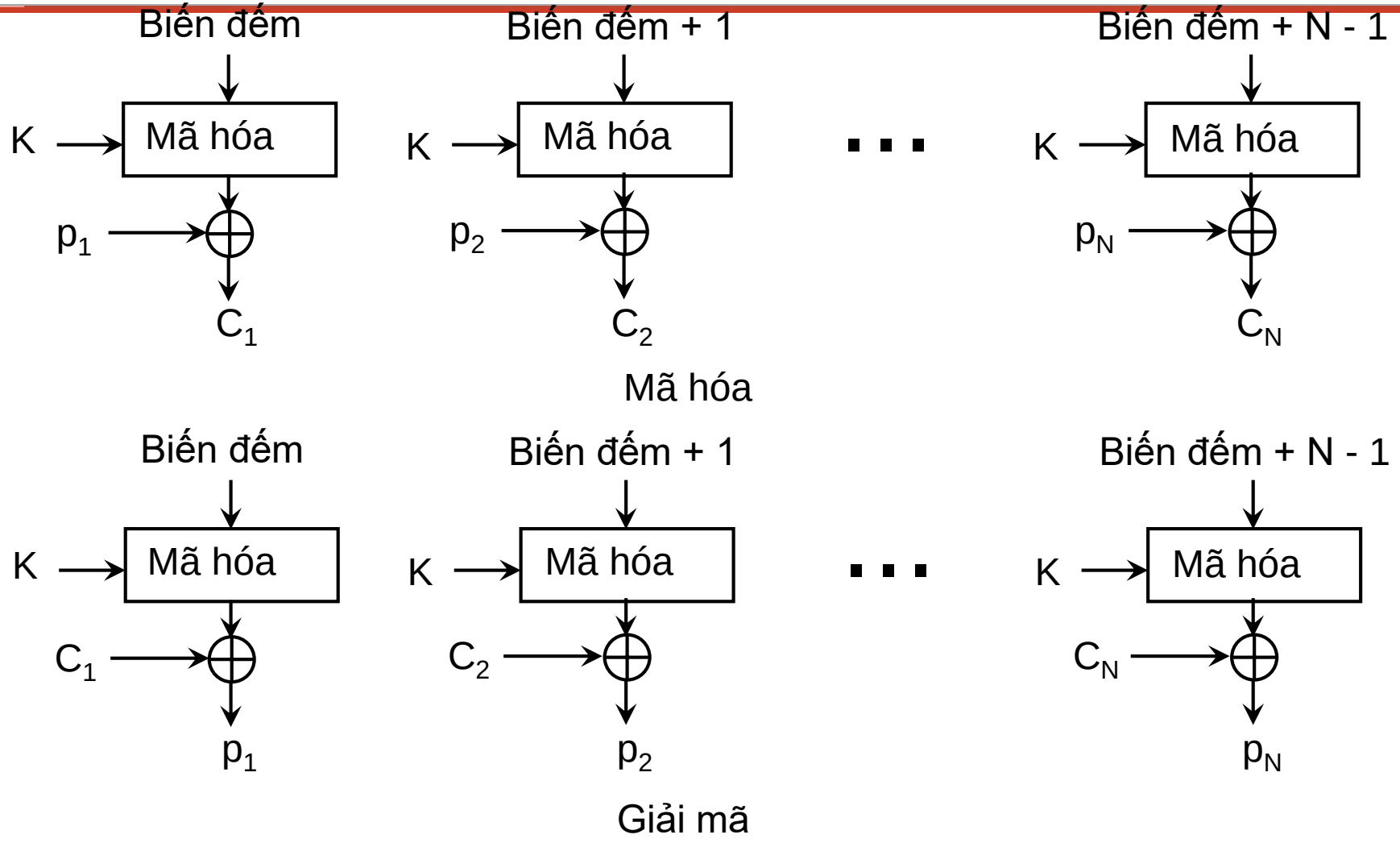
Giải mã OFB


IV
Thanh ghi dịch
64-s bit | s bit
64
K
Mã hóa
64
Chọn
s bit
Bỏ đi
64-s bit
s
s
C1
p1
s
Thanh ghi dịch
64-s bit | s bit
64
K
Mã hóa
64
Chọn
s bit
Bỏ đi
64-s bit
s
C2
p2
OM-1
Thanh ghi dịch
64-s bit | s bit
64
K
Mã hóa
64
Chọn
s bit
Bỏ đi
64-s bit
s
CM
pM

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 ninh 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)