Chương 2

MÃ HÓA ĐỐI XỨNG

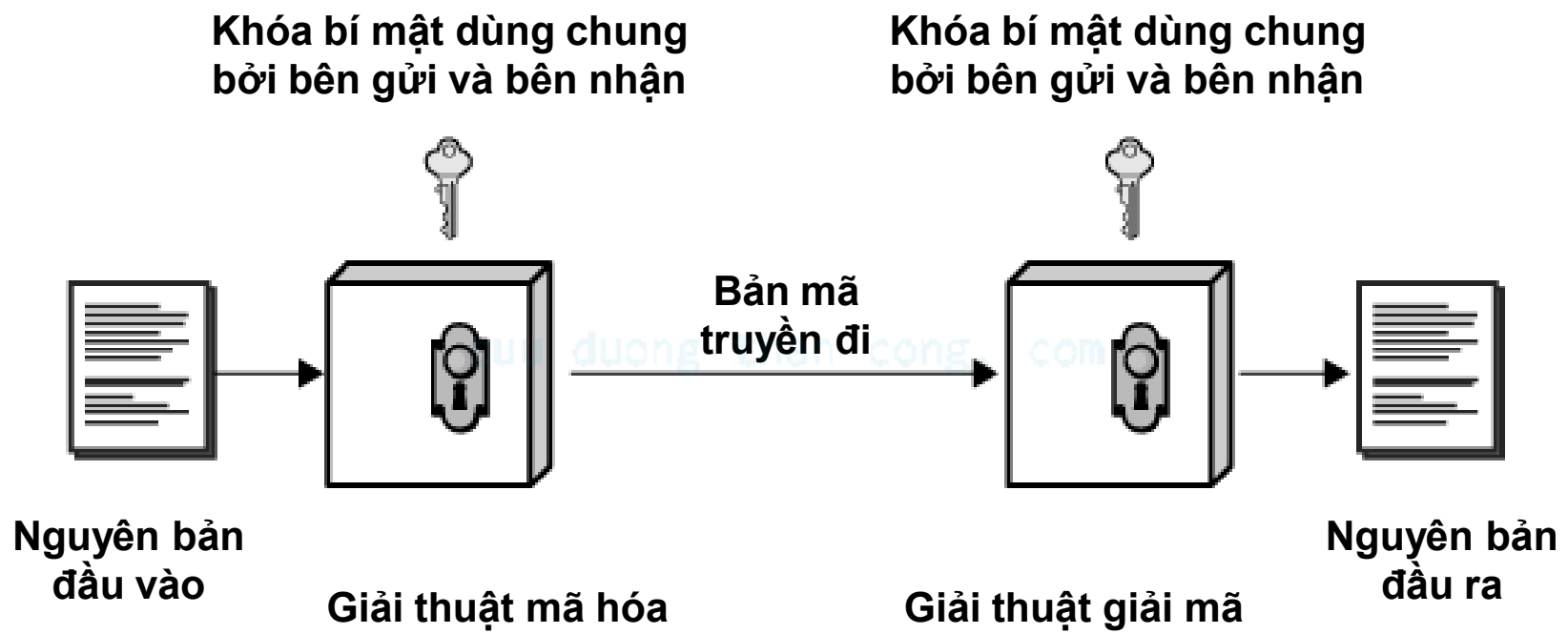
Hai kỹ thuật mã hóa chủ yếu

- Mã hóa đối xứng
  - Bên gửi và bên nhận sử dụng chung một khóa
  - Còn gọi là
    - Mã hóa truyền thống
    - Mã hóa khóa riêng / khóa đơn / khóa bí mật
  - Là kỹ thuật mã hóa duy nhất trước những năm 70
  - Hiện vẫn còn được dùng rất phổ biến
- Mã hóa khóa công khai (bất đối xứng)
  - Mỗi bên sử dụng một cặp khóa
    - Một khóa công khai + Một khóa riêng
  - Công bố chính thức năm 1976

Một số cách phân loại khác

- Theo phương thức xử lý
  - Mã hóa khối
    - Mỗi lần xử lý một khối nguyên bản và tạo ra khối bản mã tương ứng (chẳng hạn 64 hay 128 bit)
  - Mã hóa luồng
    - Xử lý dữ liệu đầu vào liên tục (chẳng hạn mỗi lần 1 bit)
- Theo phương thức chuyển đổi
  - Mã hóa thay thế
    - Chuyển đổi mỗi phần tử nguyên bản thành một phần tử bản mã tương ứng
  - Mã hóa hoán vị
    - Bố trí lại vị trí các phần tử trong nguyên bản

Mô hình hệ mã hóa đối xứng



Mã hóa

$Y = E_K(X)$

Giải mã

$X = D_K(Y)$

Mô hình hệ mã hóa đối xứng

- Gồm có 5 thành phần
  - Nguyên bản
  - Giải thuật mã hóa
  - Khóa bí mật
  - Bản mã
  - Giải thuật giải mã
- An ninh phụ thuộc vào sự bí mật của khóa, không phụ thuộc vào sự bí mật của giải thuật

Phá mã

- Là nỗ lực giải mã văn bản đã được mã hóa không biết trước khóa bí mật
- Có hai phương pháp phá mã
  - Vét cạn
    - Thử tất cả các khóa có thể
  - Thám mã
    - Khai thác những nhược điểm của giải thuật
    - Dựa trên những đặc trưng chung của nguyên bản hoặc một số cặp nguyên bản - bản mã mẫu

Phương pháp phá mã vét cạn

- Về lý thuyết có thể thử tất cả các giá trị khóa cho đến khi tìm thấy nguyên bản từ bản mã
- Dựa trên giả thiết có thể nhận biết được nguyên bản cần tìm
- Tính trung bình cần thử một nửa tổng số các trường hợp có thể
- Thực tế không khả khi nếu độ dài khóa lớn

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

Các kỹ thuật thám mã

- Chỉ có bản mã
  - Chỉ biết giải thuật mã hóa và bản mã hiện có
- Biết nguyên bản
  - Biết thêm một số cặp nguyên bản - bản mã
- Chọn nguyên bản
  - Chọn 1 nguyên bản, biết bản mã tương ứng
- Chọn bản mã
  - Chọn 1 bản mã, biết nguyên bản tương ứng
- Chọn văn bản
  - Kết hợp chọn nguyên bản và chọn bản mã

An ninh hệ mã hóa

- An ninh vô điều kiện
  - Bản mã không chứa đủ thông tin để xác định duy nhất nguyên bản tương ứng, bất kể với số lượng bao nhiêu và tốc độ máy tính thế nào
  - Chỉ hệ mã hóa độn một lần là an ninh vô điều kiện
- An ninh tính toán
  - Thỏa mãn một trong hai điều kiện
    - Chi phí phá mã vượt quá giá trị thông tin
    - Thời gian phá mã vượt quá tuổi thọ thông tin
  - Thực tế thỏa mãn hai điều kiện
    - Không có nhược điểm
    - Khóa có quá nhiều giá trị không thể thử hết

Mã hóa thay thế cổ điển

- Các chữ cái của nguyên bản được thay thế bởi các chữ cái khác, hoặc các số, hoặc các ký hiệu
- Nếu nguyên bản được coi như một chuỗi bit thì thay thế các mẫu bit trong nguyên bản bằng các mẫu bit của bản mã

Hệ mã hóa Caesar

- Là hệ mã hóa thay thế xuất hiện sớm nhất và đơn giản nhất
- Sử dụng đầu tiên bởi Julius Caesar vào mục đích quân sự
- Dịch chuyển xoay vòng theo thứ tự chữ cái
  - Khóa k là số bước dịch chuyển
  - Với mỗi chữ cái của văn bản
    - Đặt p = 0 nếu chữ cái là a, p = 1 nếu chữ cái là b,...
    - Mã hóa : $C = E(p) = (p + k) \bmod 26$
    - Giải mã : $p = D(C) = (C - k) \bmod 26$
- Ví dụ : Mã hóa "meet me after class" với k = 3

Phá mã hệ mã hóa Caesar

- Phương pháp vét cạn
  - Khóa chỉ là một chữ cái (hay một số giữa 1 và 25)
  - Thử tất cả 25 khóa có thể
  - Dễ dàng thực hiện
- Ba yếu tố quan trọng
  - Biết trước các giải thuật mã hóa và giải mã
  - Chỉ có 25 khóa để thử
  - Biết và có thể dễ dàng nhận ra được ngôn ngữ của nguyên bản
- Ví dụ : Phá mã "GCUA VQ DTGCM"

Hệ mã hóa đơn bảng

- Thay một chữ cái này bằng một chữ cái khác theo trật tự bất kỳ sao cho mỗi chữ cái chỉ có một thay thế duy nhất và ngược lại
- Khóa dài 26 chữ cái
- Ví dụ
  - Khóa

    a b c d e f g h i j k l m n o p q r s t u v w x y z

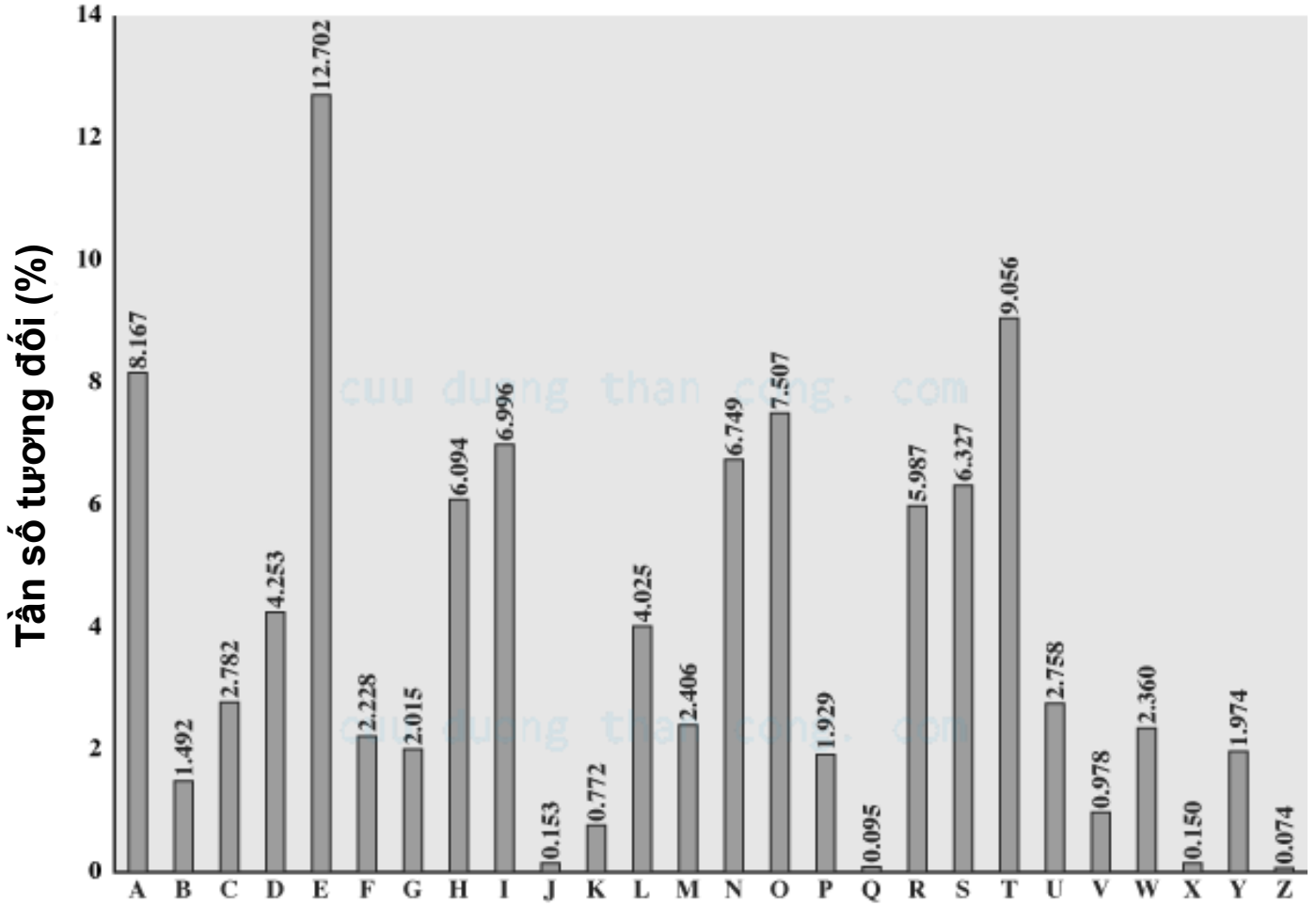
    M N B V C X Z A S D F G H J K L P O I U Y T R E W Q
  - Nguyên bản

    i love you

Phá mã hệ mã hóa đơn bảng

- Phương pháp vét cạn
  - Khóa dài 26 ký tự
  - Số lượng khóa có thể = 26! = 4×10^{26}
  - Rất khó thực hiện
- Khai thác những nhược điểm của giải thuật
  - Biết rõ tần số các chữ cái tiếng Anh
    - Có thể suy ra các cặp chữ cái nguyên bản - chữ cái bản mã
    - Ví dụ : chữ cái xuất hiện nhiều nhất có thể tương ứng với 'e'
  - Có thể nhận ra các bộ đôi và bộ ba chữ cái
    - Ví dụ bộ đôi : 'th', 'an', 'ed'
    - Ví dụ bộ ba : 'ing', 'the', 'est'

Các tần số chữ cái tiếng Anh



Ví dụ phá mã hệ đơn bảng

- Cho bản mã
  UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
  VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
  EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
- Tính tần số chữ cái tương đối
- Đoán P là e, Z là t
- Đoán ZW là th và ZWP là the
- Tiếp tục đoán và thử, cuối cùng được
  it was disclosed yesterday that several informal but
  direct contacts have been made with political
  representatives of the viet cong in moscow

Hệ mã hóa Playfair (1)

- Là một hệ mã hóa nhiều chữ
  - Giảm bớt tương quan cấu trúc giữa bản mã và nguyên bản bằng cách mã hóa đồng thời nhiều chữ cái của nguyên bản
- Phát minh bởi Charles Wheatstone vào năm 1854, lấy tên người bạn Baron Playfair
- Sử dụng 1 ma trận chữ cái 5x5 xây dựng trên cơ sở 1 từ khóa
  - Điền các chữ cái của từ khóa (bỏ các chữ trùng)
  - Điền nốt ma trận với các chữ khác của bảng chữ cái
  - I và J chiếm cùng một ô của ma trận

Hệ mã hóa Playfair (2)

- Ví dụ ma trận với từ khóa MONARCHY

M	**O**	**N**	**A**	**R**
C	**H**	**Y**	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

- Mã hóa 2 chữ cái một lúc
  - Nếu 2 chữ giống nhau, tách ra bởi 1 chữ điền thêm
  - Nếu 2 chữ nằm cùng hàng, thay bởi các chữ bên phải
  - Nếu 2 chữ nằm cùng cột, thay bởi các chữ bên dưới
  - Các trường hợp khác, mỗi chữ cái được thay bởi chữ cái khác cùng hàng, trên cột chữ cái cùng cặp

Phá mã hệ mã hóa Playfair

- An ninh đảm bảo hơn nhiều hệ mã hóa đơn chữ
- Có 26 x 26 = 676 cặp chữ cái
  - Việc giải mã từng cặp khó khăn hơn
  - Cần phân tích 676 tần số xuất hiện thay vì 26
- Từng được quân đội Anh, Mỹ sử dụng rộng rãi
- Bản mã vẫn còn lưu lại nhiều cấu trúc của nguyên bản
- Vẫn có thể phá mã được vì chỉ có vài trăm cặp chữ cái cần giải mã

Hệ mã hóa Vigenère

- Là một hệ mã hóa đa bảng
  - Sử dụng nhiều bảng mã hóa
  - Khóa giúp chọn bảng tương ứng với mỗi chữ cái
- Kết hợp 26 hệ Ceasar (bước dịch chuyển 0 - 25)
  - Khóa K = $k_1k_2...k_d$ gồm d chữ cái sử dụng lặp đi lặp lại với các chữ cái của văn bản
  - Chữ cái thứ i tương ứng với hệ Ceasar bước chuyển i
- Ví dụ
  - Khóa : deceptivedeceptivedeceptive
  - Nguyên bản : wearediscoveredsaveyourself
  - Bản mã : ZICVTWQNGRZGVTWAVZHCQYGLMGJ

Phá mã hệ mã hóa Vigenère

- Phương pháp vét cạn
  - Khó thực hiện, nhất là nếu khóa gồm nhiều chữ cái
- Khai thác những nhược điểm của giải thuật
  - Cấu trúc của nguyên bản được che đậy tốt hơn hệ Playfair nhưng không hoàn toàn biến mất
  - Chỉ việc tìm độ dài khóa sau đó phá mã từng hệ Ceasar
  - Cách tìm độ dài khóa
    - Nếu độ dài khóa nhỏ so với độ dài văn bản, có thể phát hiện 1 dãy văn bản lặp lại nhiều lần
    - Khoảng cách giữa 2 dãy văn bản lặp là 1 bội số của độ dài khóa
    - Từ đó suy ra độ dài khóa

Hệ mã hóa khóa tự động

- Vigenère đề xuất từ khóa không lặp lại mà được gắn vào đầu nguyên bản
  - Nếu biết từ khóa sẽ giải mã được các chữ cái đầu tiên
  - Sử dụng các chữ cái này làm khóa để giải mã các chữ các tiếp theo,...
- Ví dụ :
  - Khóa : deceptivewearediscoveredsav
  - nguyên bản : wearediscoveredsaveyourself
  - Mã hóa : ZICVTWQNGKZEIIGASXSTSLVVWLA
- Vẫn có thể sử dụng kỹ thuật thống kê để phá mã
  - Khóa và nguyên bản có cùng tần số các chữ cái

Độn một lần

- Là hệ mã hóa thay thế không thể phá được
- Đề xuất bởi Joseph Mauborgne
- Khóa ngẫu nhiên, độ dài bằng độ dài văn bản, chỉ sử dụng một lần
- Giữa nguyên bản và bản mã không có bất kỳ quan hệ nào về thống kê
- Với bất kỳ nguyên bản và bản mã nào cũng tồn tại một khóa tương ứng
- Khó khăn ở việc tạo khóa và đảm bảo phân phối khóa an ninh

Mã hóa hoán vị cổ điển

- Che đậy nội dung văn bản bằng cách sắp xếp lại trật tự các chữ cái
- Không thay đổi các chữ cái của nguyên bản
- Bản mã có tần số xuất hiện các chữ cái giống như nguyên bản

Hệ mã hóa hàng rào

- Viết các chữ cái theo đường chéo trên một số hàng nhất định
- Sau đó đọc theo từng hàng một
- Ví dụ
  - Nguyên bản : attack at midnight
  - Mã hóa với độ cao hàng rào là 2

    a t c a m d i h

    t a k t i n g t
  - Bản mã : ATCAMDIHTAKTINGT

Hệ mã hóa hàng

- Viết các chữ cái theo hàng vào 1 số cột nhất định
- Sau đó hoán vị các cột trước khi đọc theo cột
- Khóa là thứ tự đọc các cột
- Ví dụ
  - Khóa : 4 3 1 2 5 6 7
  - Nguyên bản : a t t a c k p
    o s t p o n e
    d u n t i l t
    w o a m x y z
  - Bản mã :

    TTNAAPTMTSUOAODWCOIXKNLYPETZ

Mã hóa tích hợp

- Các hệ mã hóa thay thế và hoán vị không an toàn vì những đặc điểm của ngôn ngữ
- Kết hợp sử dụng nhiều hệ mã hóa sẽ khiến việc phá mã khó hơn
  - Hai thay thế tạo nên một thay thế phức tạp hơn
  - Hai hoán vị tạo nên một hoán vị phức tạp hơn
  - Một thay thế với một hoán vị tạo nên một hệ mã hóa phức tạp hơn nhiều
- Là cầu nối từ các hệ mã hóa cổ điển đến các hệ mã hóa hiện đại

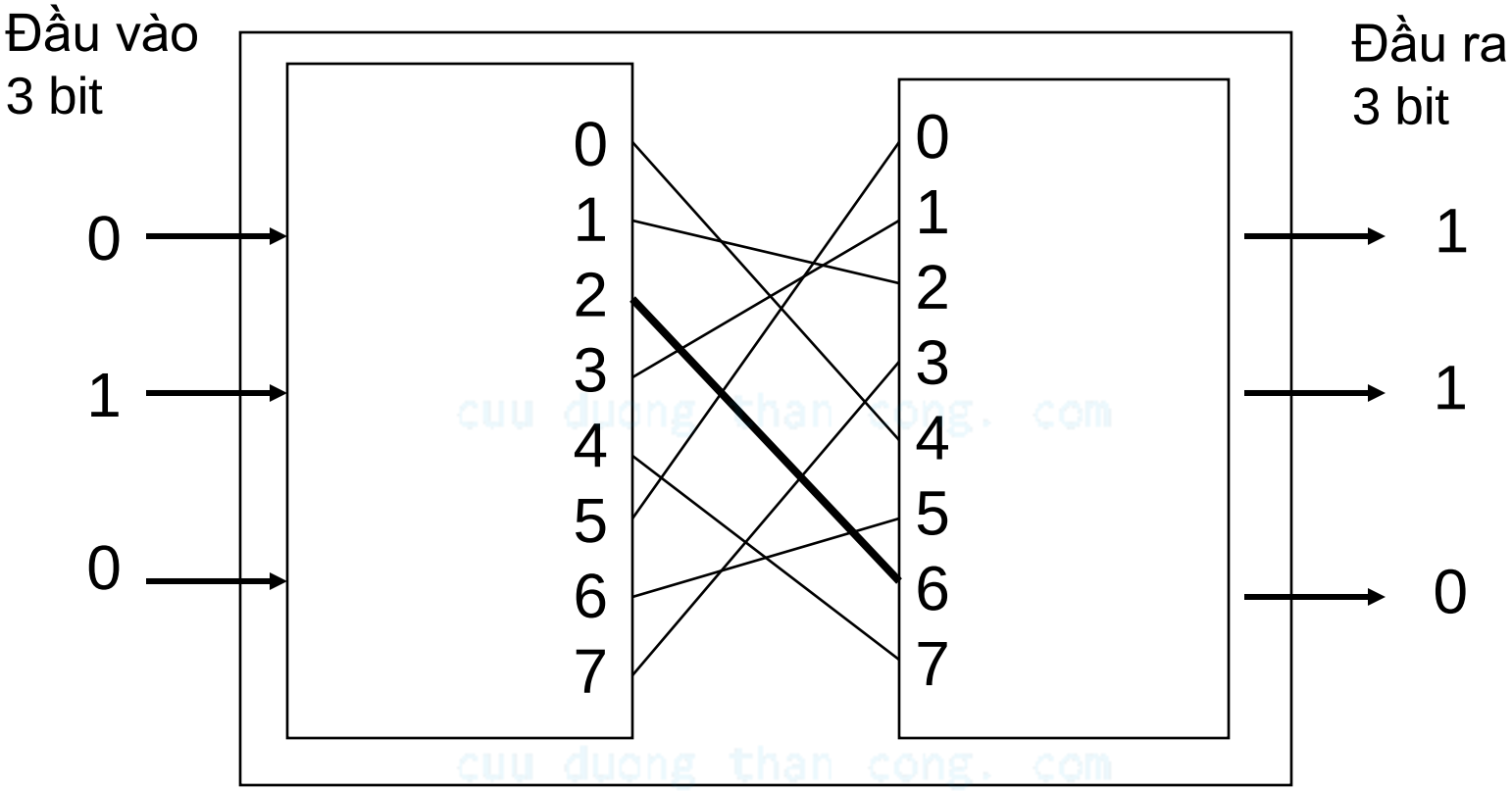
Mã hóa khối

- So với mã hóa luồng
  - Mã hóa khối xử lý thông báo theo từng khối
  - Mã hóa luồng xử lý thông báo 1 bit hoặc 1 byte mỗi lần
- Giống như thay thế các ký tự rất lớn ([redacted]4 bit)
  - Bảng mã hóa gồm 2^n đầu vào (n là độ dài khối)
  - Mỗi khối đầu vào ứng với một khối mã hóa duy nhất
    - Tính thuận nghịch
  - Độ dài khóa là n x 2^n bit quá lớn
- Xây dựng từ các khối nhỏ hơn
- Hầu hết các hệ mã hóa khối đối xứng dựa trên cấu trúc hệ mã hóa Feistel

Mạng S-P

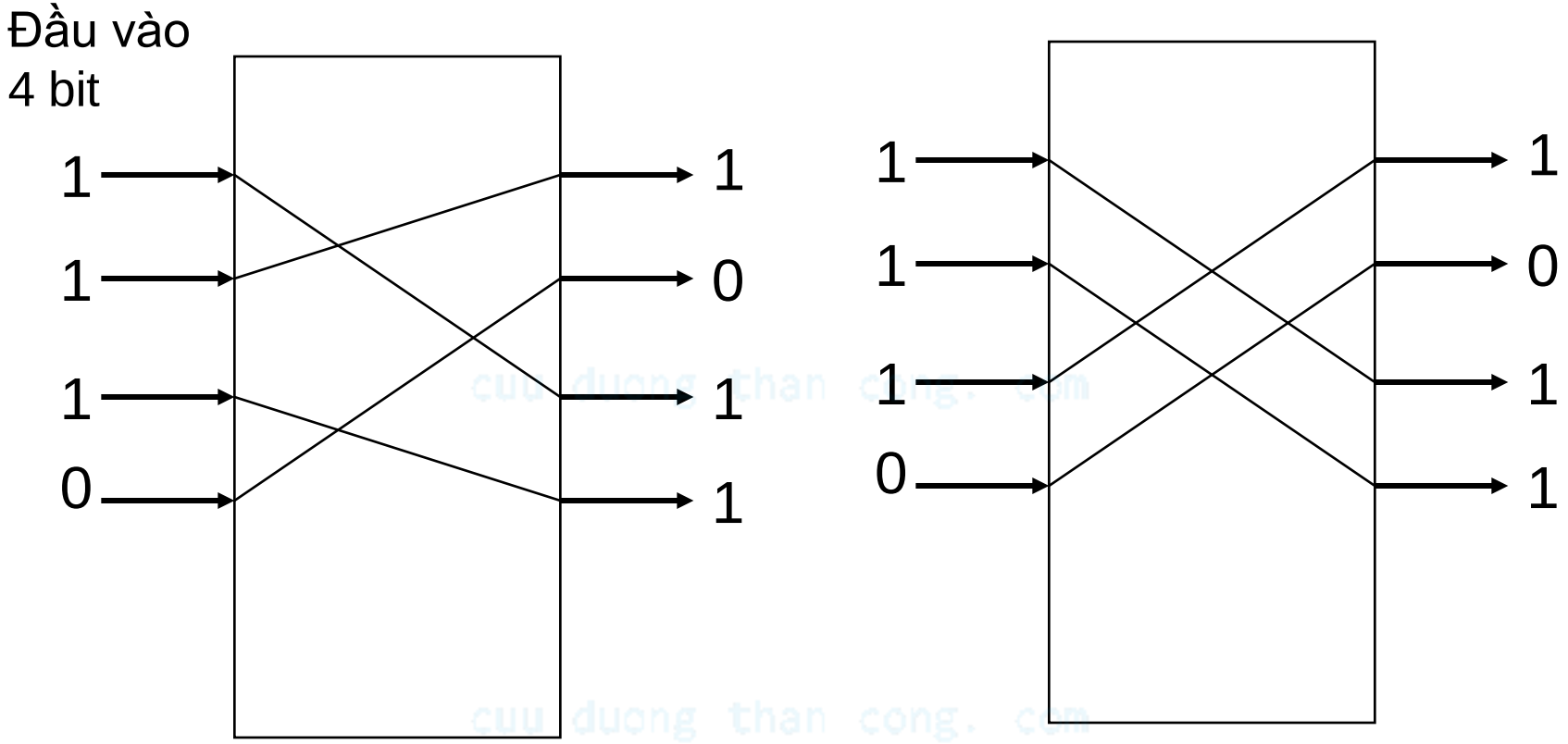
- Mạng thay thế (S) - hoán vị (P) đề xuất bởi Claude Shannon vào năm 1949
- Là cơ sở của các hệ mã hóa khối hiện đại
- Dựa trên 2 phép mã hóa cổ điển
  - Phép thay thế : Hộp S
  - Phép hoán vị : Hộp P
- Đan xen các chức năng
  - Khuếch tán : Hộp P (kết hợp với hộp S)
    - Phát tỏa cấu trúc thống kê của nguyên bản khắp bản mã
  - Gây lẫn : Hộp S
    - Làm phức tạp hóa mối quan hệ giữa bản mã và khóa

Hộp S



Lưu ý : Hộp S có tính thuận nghịch

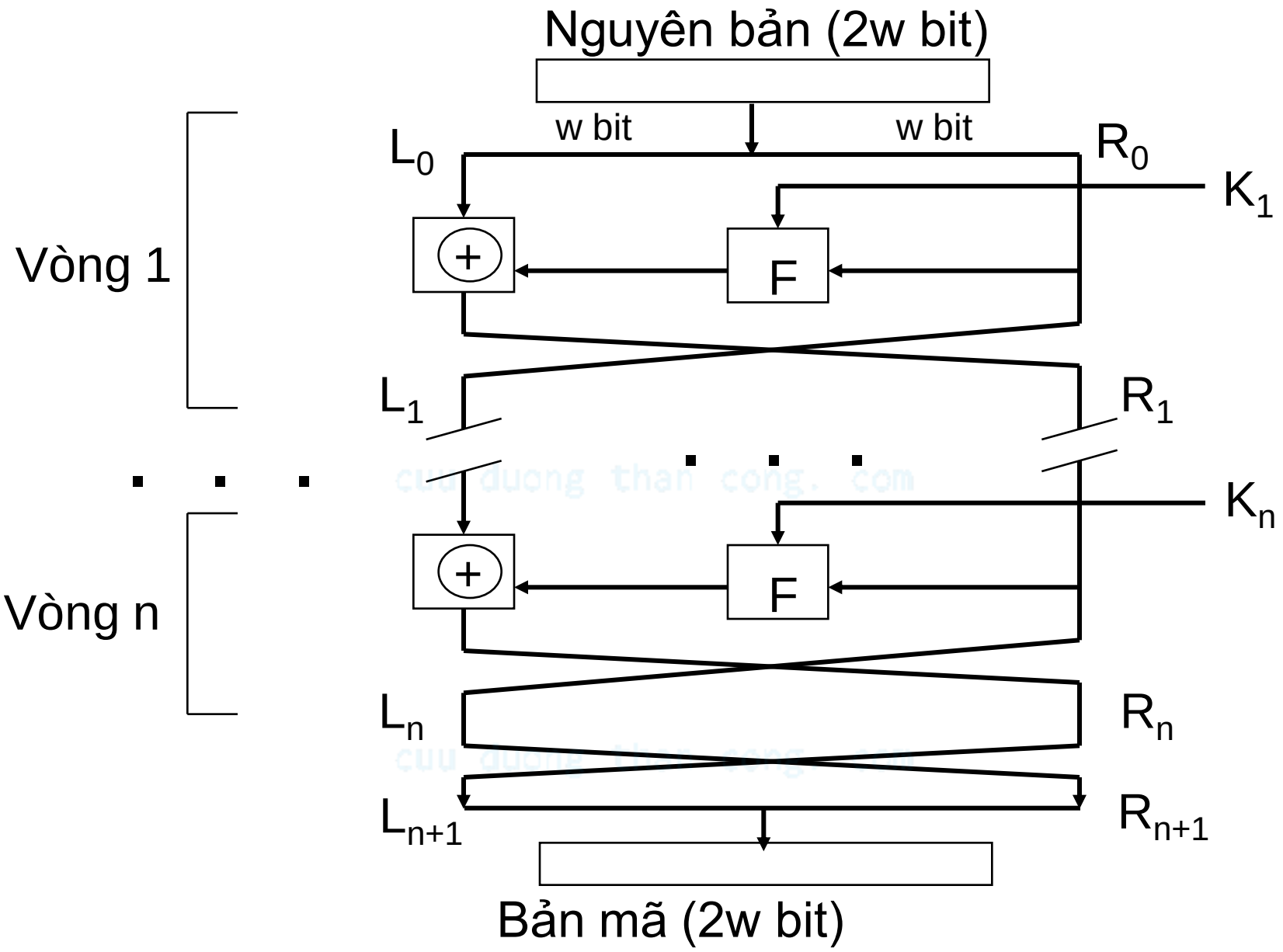
Hộp P



Lưu ý : Hộp P có tính thuận nghịch

Mã hóa Feistel

- Đề xuất bởi Horst Feistel dựa trên khái niệm hệ mã hóa tích hợp thuận nghịch của Shannon
- Phân mỗi khối dài 2w bit thành 2 nửa L_0 và R_0
- Xử lý qua n vòng
- Chia khóa K thành n khóa con $K_1, K_2, ..., K_n$
- Tại mỗi vòng i
  - Thực hiện thay thế ở nửa bên trái L_{i-1} bằng cách XOR nó với $F(K_i, R_{i-1})$
  - F thường gọi là hàm chuyển đổi hay hàm vòng
  - Hoán vị hai nửa L_i và R_i



Các đặc trưng hệ Feistel

- Độ dài khối
  - Khối càng lớn càng an ninh (thường 64 bit)
- Độ dài khóa
  - Khóa càng dài càng an ninh (thường 128 bit)
- Số vòng
  - Càng nhiều vòng càng an ninh (thường 16 vòng)
- Giải thuật sinh mã con
  - Càng phức tạp càng khó phá mã
- Hàm vòng
  - Càng phức tạp càng khó phá mã
- Ảnh hưởng đến cài đặt và phân tích

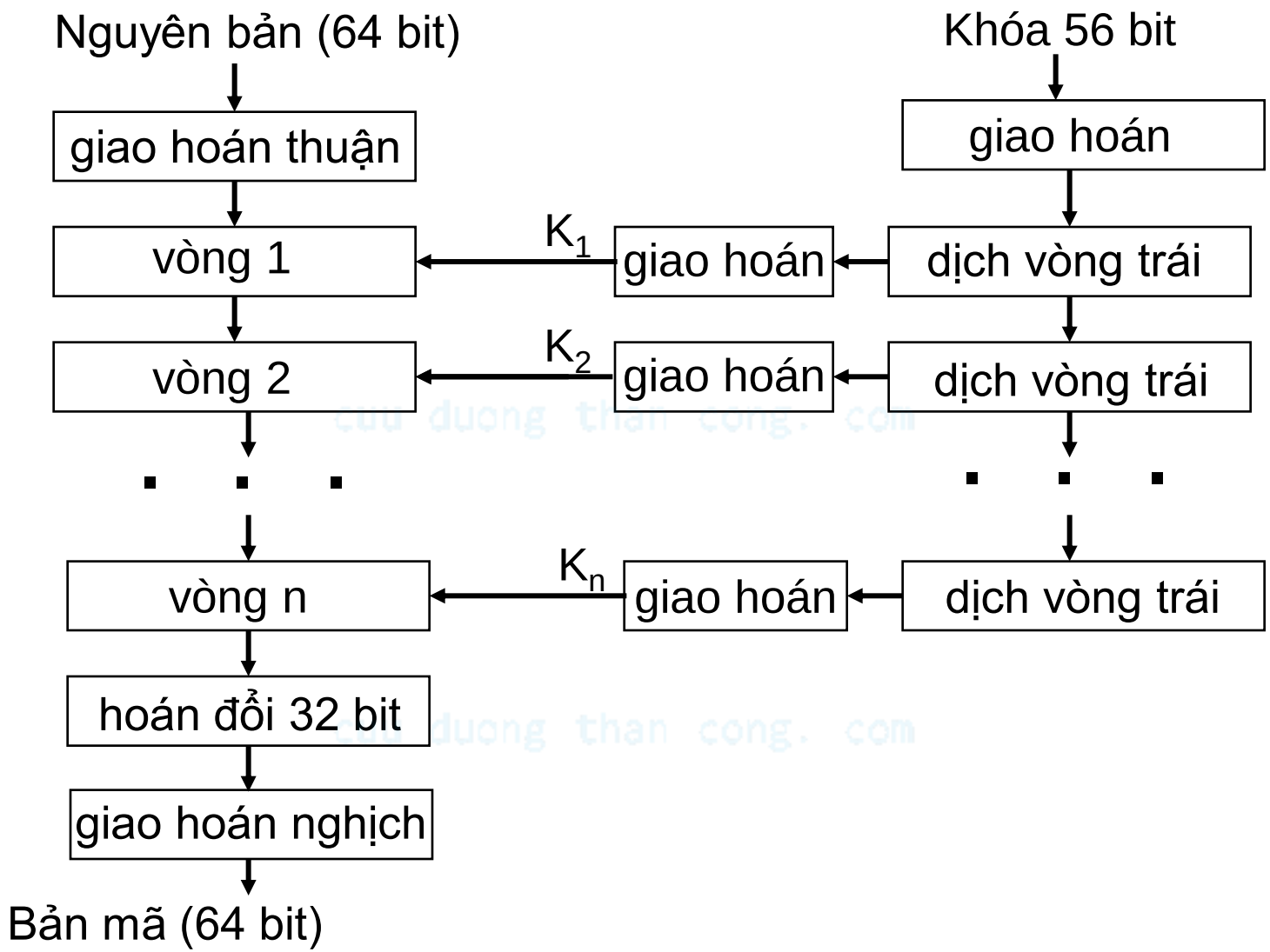
Giải mã Feistel

- Giống giải thuật mã hóa, chỉ khác
  - Bản mã là dữ liệu đầu vào
  - Các khóa con được dùng theo thứ tự ngược lại
- Tại mỗi vòng kết quả đầu ra chính là các dữ liệu đầu vào của quá trình mã hóa
  - Đối với quá trình mã hóa
    - $L_i = R_{i-1}$
    - $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
  - Đối với quá trình giải mã
    - $R_{i-1} = L_i$
    - $L_{i-1} = R_i \oplus F(L_i, K_i)$

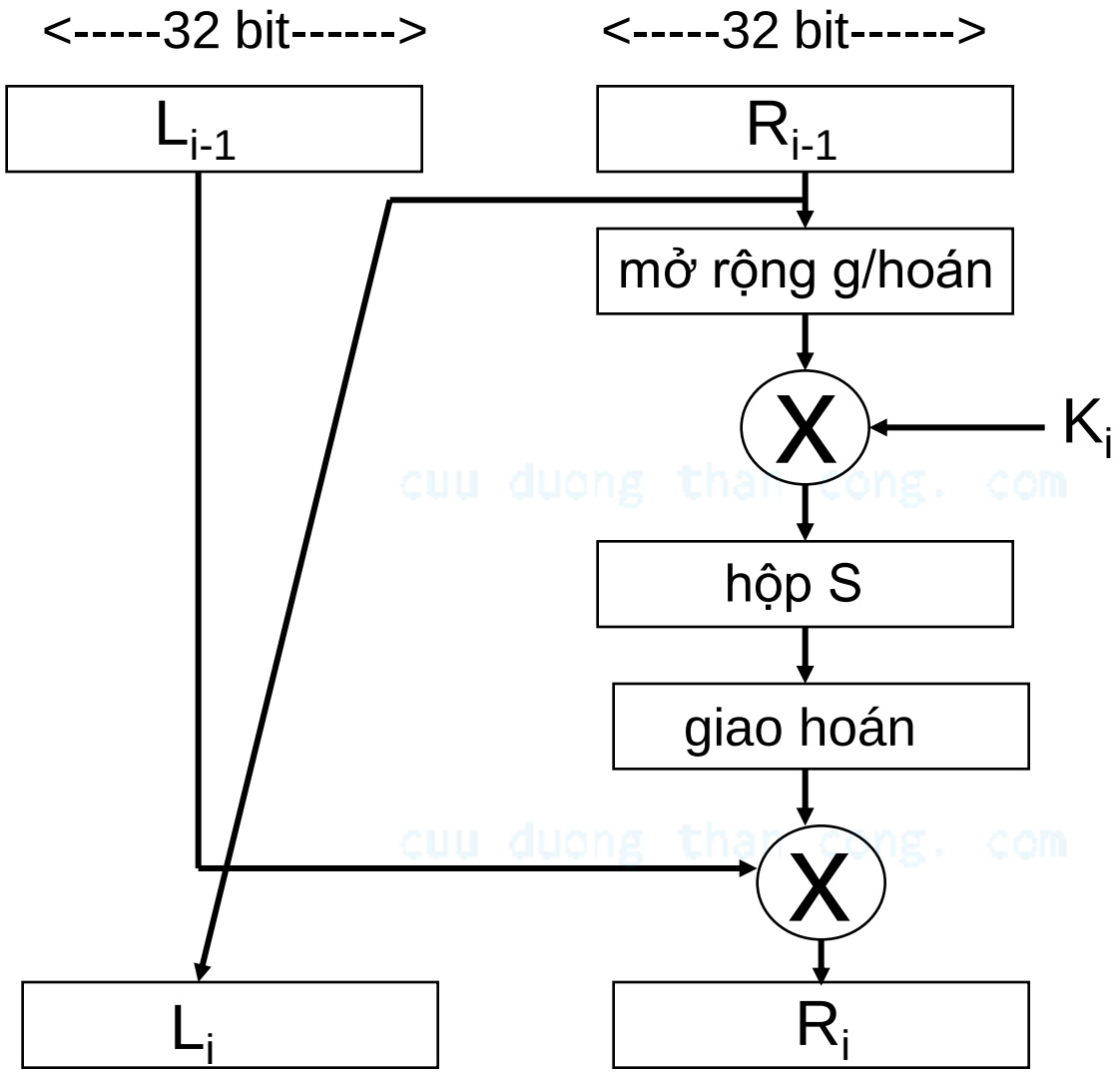
Chuẩn mã hóa dữ liệu

- DES (Data Encryption Standard) được công nhận chuẩn năm 1977
- Phương thức mã hóa được sử dụng rộng rãi nhất
- Tên giải thuật là DEA (Data Encryption Algorithm)
- Là một biến thể của hệ mã hóa Feistel, bổ xung thêm các hoán vị đầu và cuối
- Kích thước khối : 64 bit
- Kích thước khóa : 56 bit
- Số vòng : 16
- Từng gây nhiều tranh cãi về độ an ninh

Giải thuật mã hóa DES



Một vòng DES



Phá mã DES

- Khóa 56 bit có $2^{56} = 7,2 \times 10^{16}$ giá trị có thể
- Phương pháp vét cạn tỏ ra không thực tế
- 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
  - 1999 : 100000 máy tính phá mã trong 22 giờ
- Vấn đề còn phải nhận biết được nguyên bản
- Thực tế DES vẫn được sử dụng không có vấn đề
- 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 $C = E_{K_4}(p)$
- Vì sao 3 lần : tránh tấn công "gặp nhau ở giữa"
  - $C = E_{K_2}(E_{K_1}(p)) = E_{K_1}(p) = D_{K_2}(C)$
  - 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 OK thì K_1 và K_2 là khóa

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
- Cấu trúc mạng S-P, nhưng không theo hệ Feistel
  - Không chia mỗi khối làm đôi

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

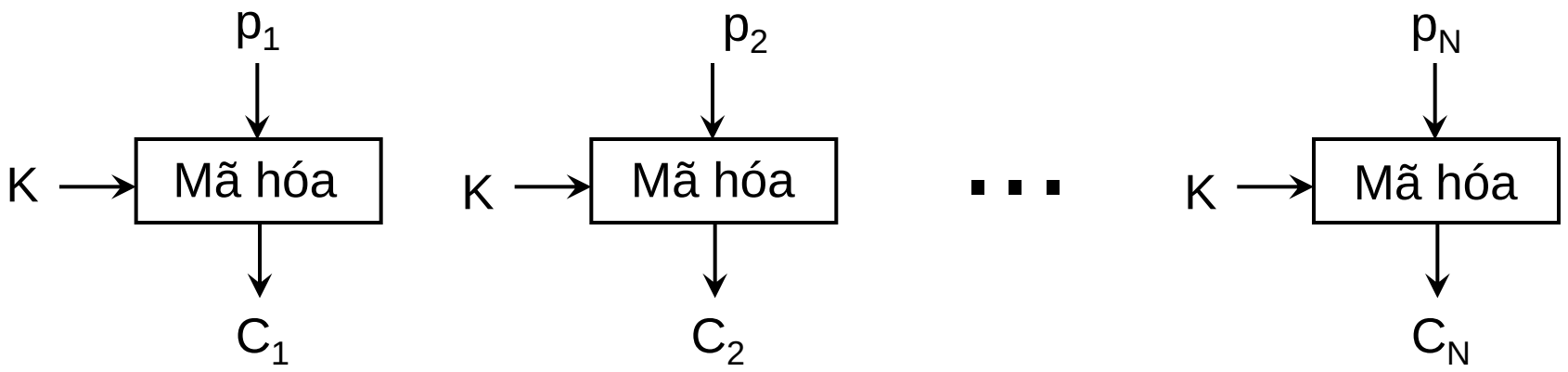
Các hệ mã hóa khối khác (2)

- 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

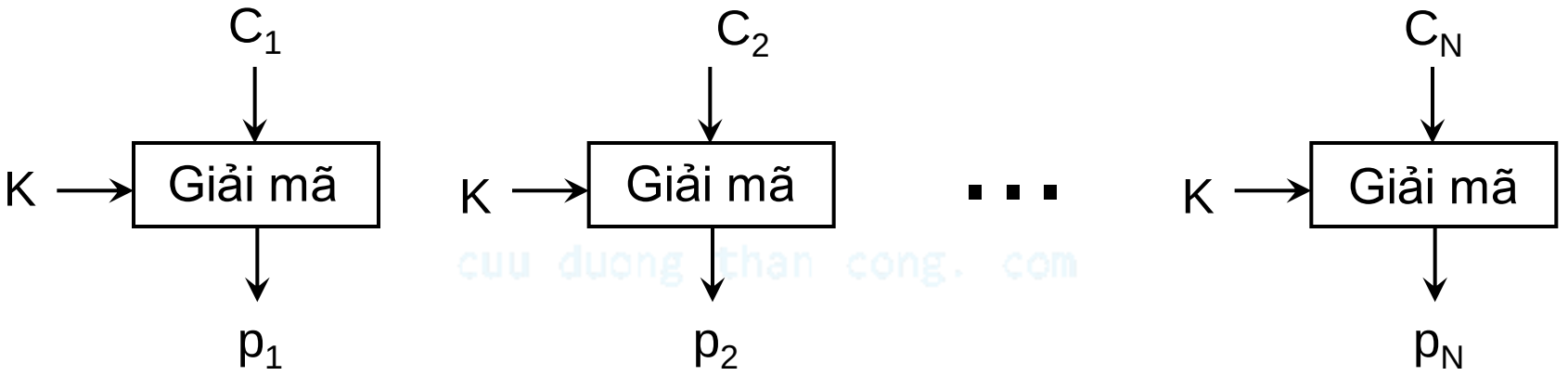
Các phương thức mã hóa khối

- 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

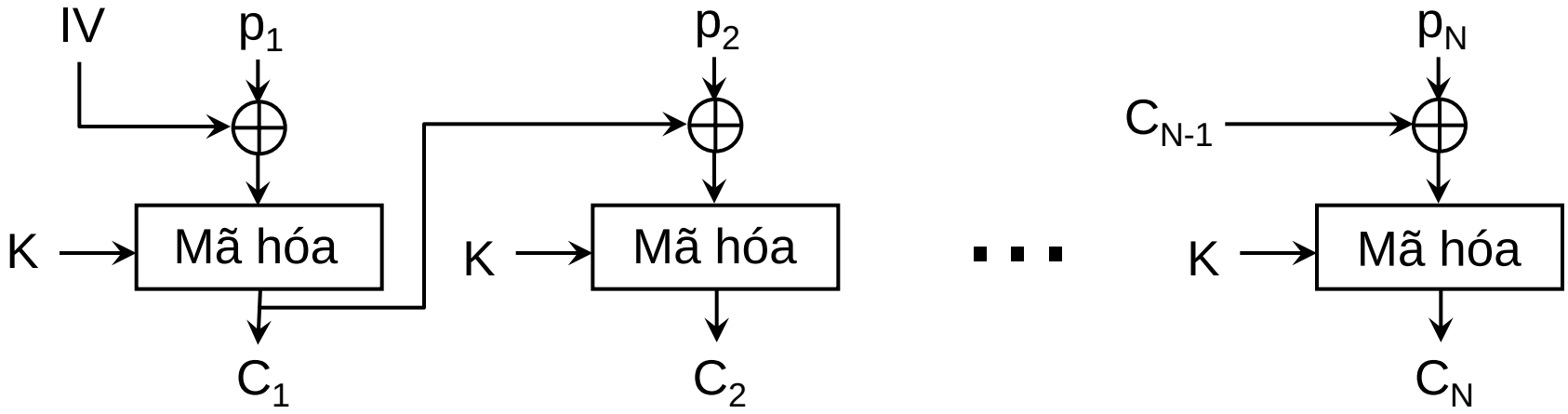


Giải mã

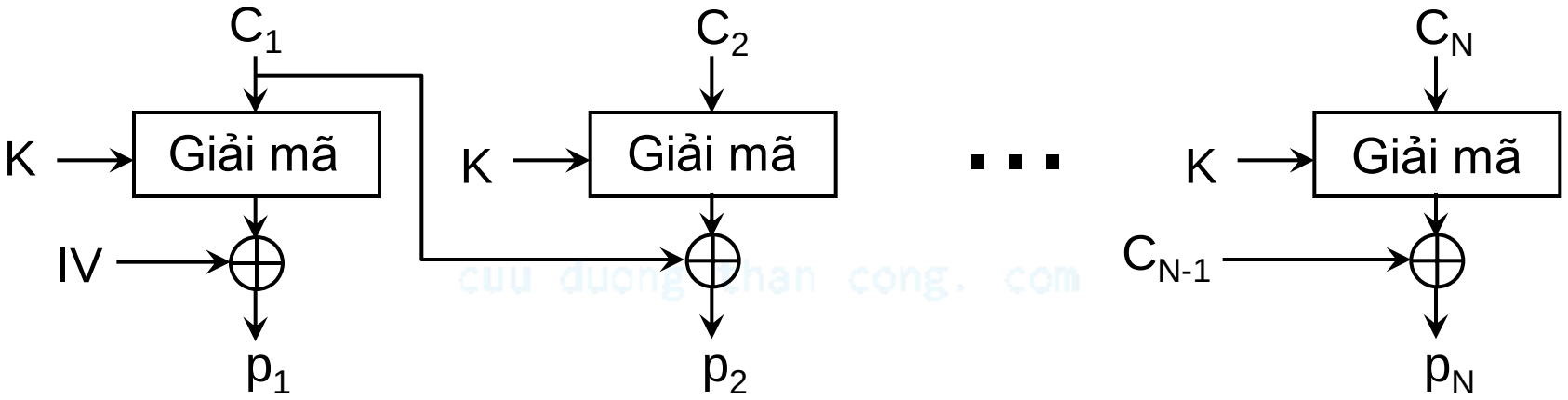
Đánh giá ECB

- 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

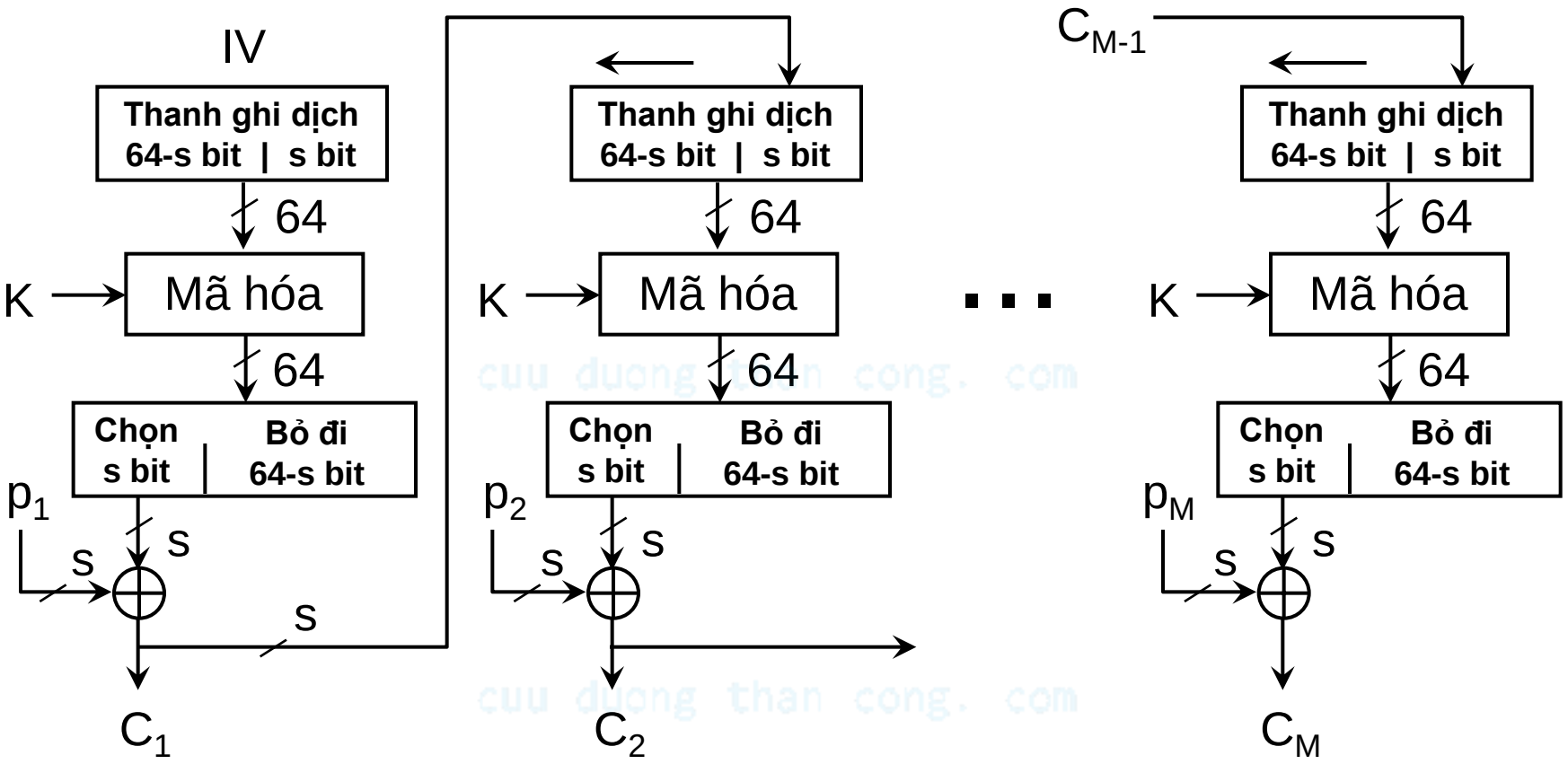


Giải mã

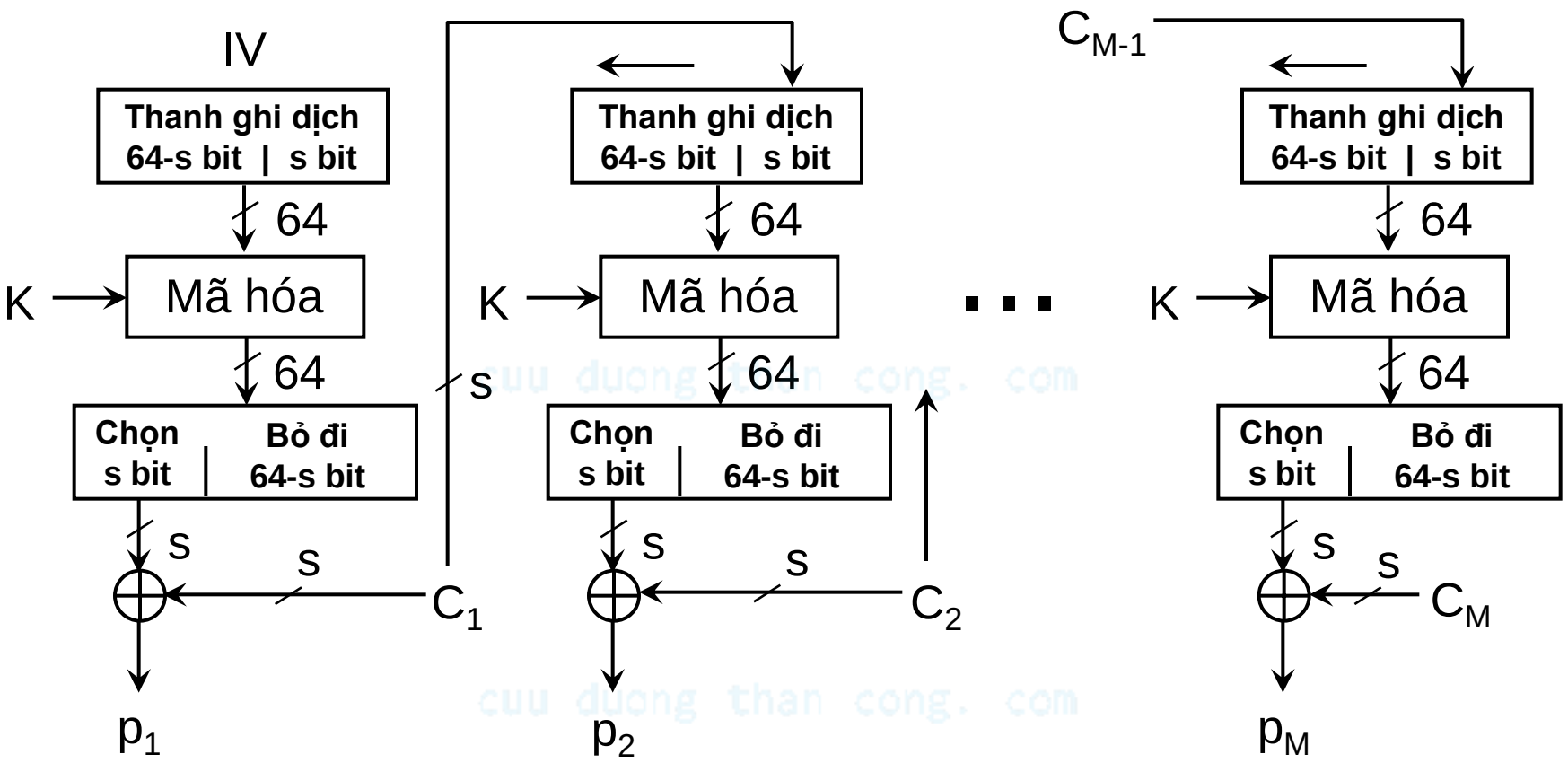
Đánh giá CBC

- 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



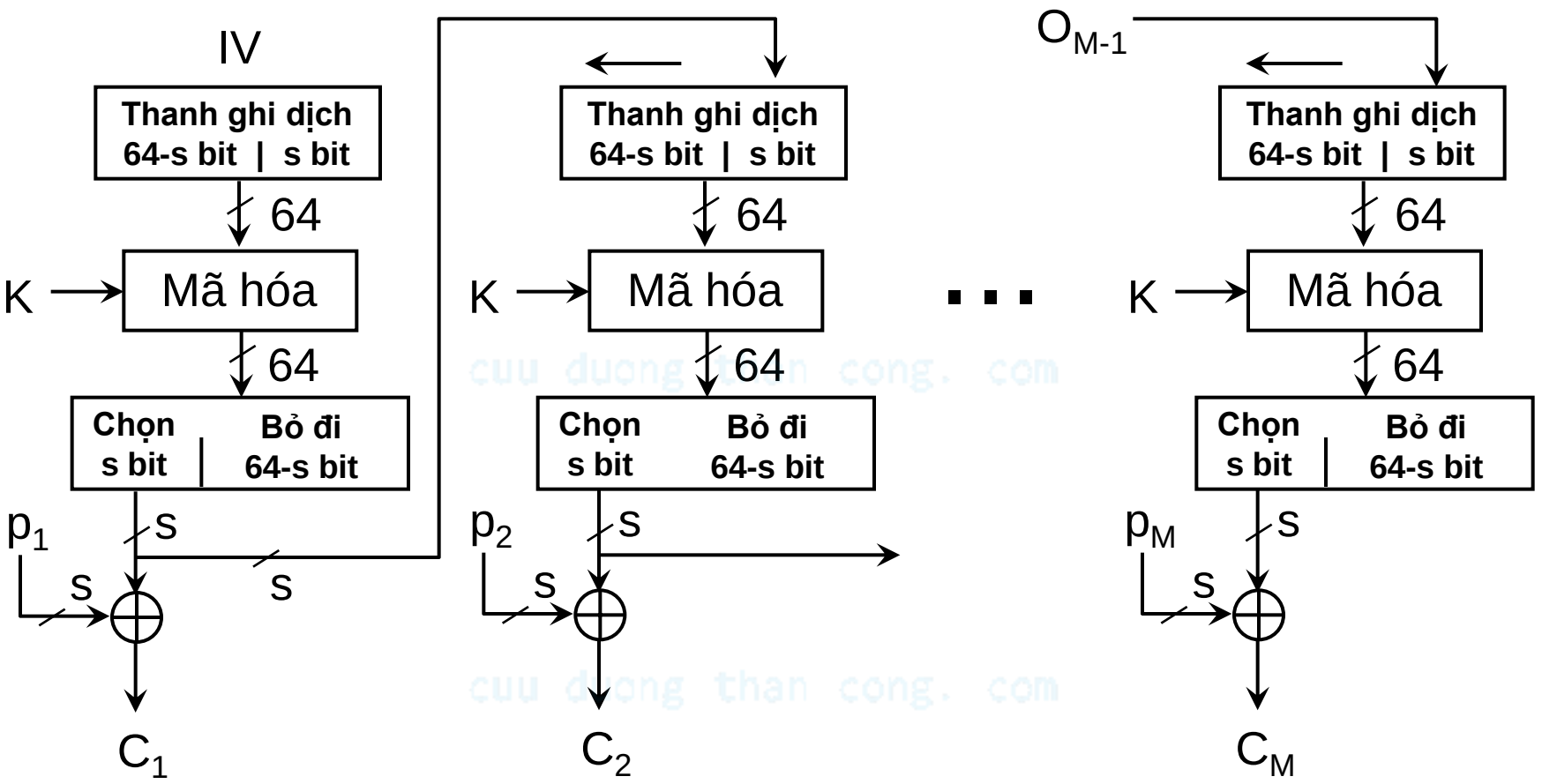
Giải mã CFB



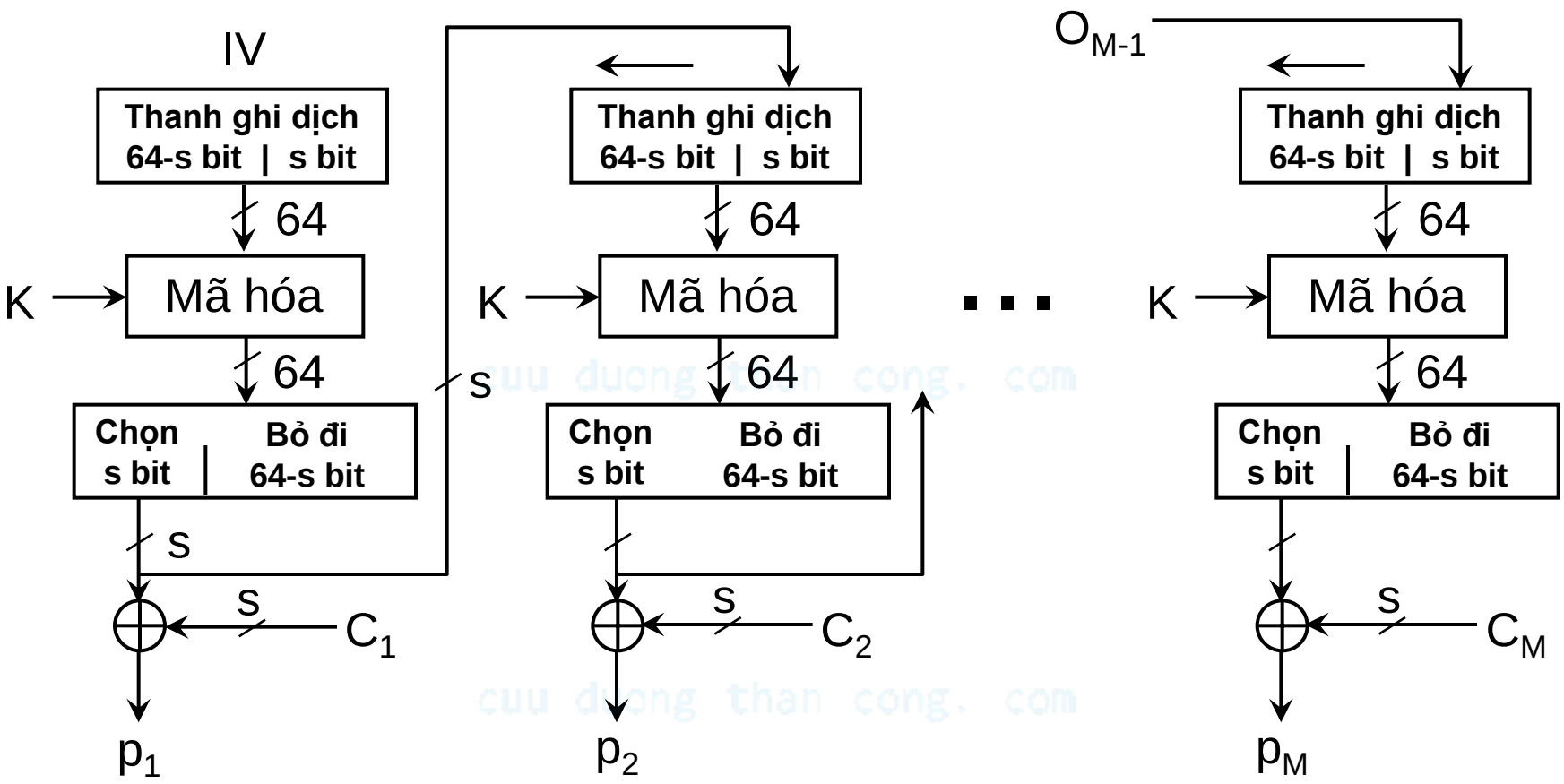
Đánh giá CFB

- 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



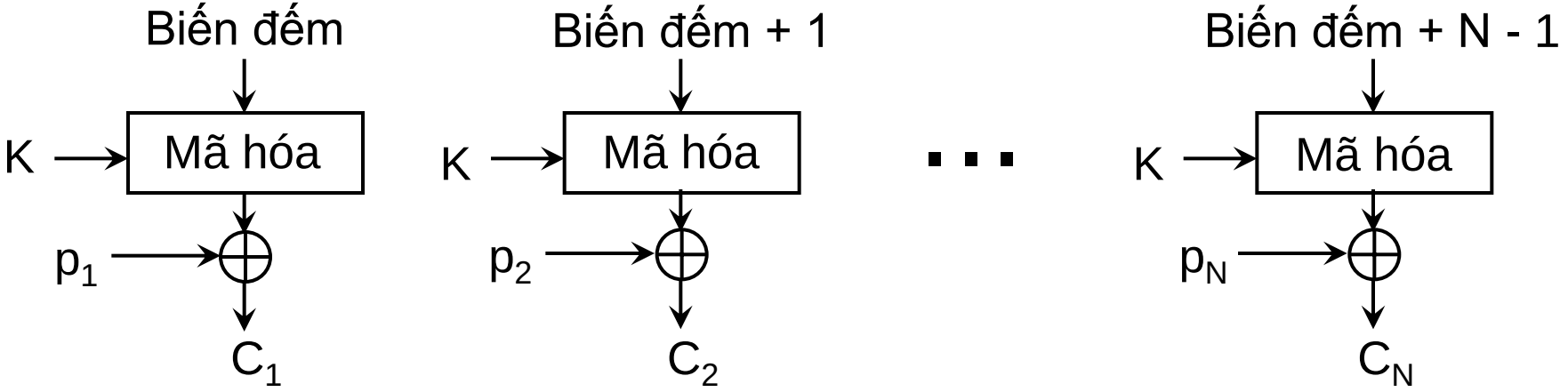
Giải mã OFB



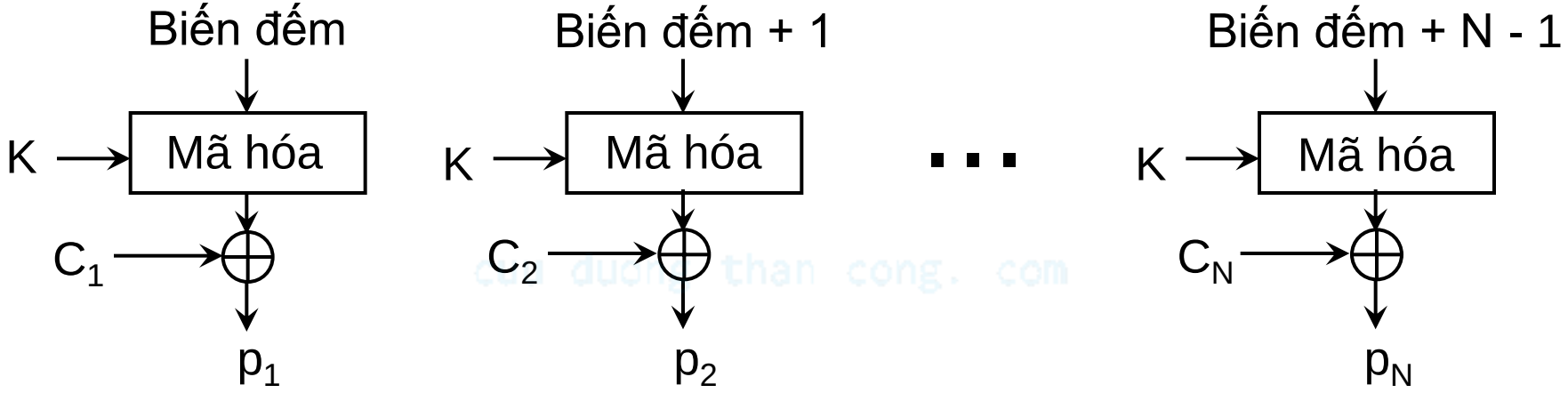
Đánh giá OFB

- 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



Mã hóa



Giải mã

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)

Bố trí công cụ mã hóa

- Giải pháp hữu hiệu và phổ biến nhất chống lại các mối đe dọa đến an ninh 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

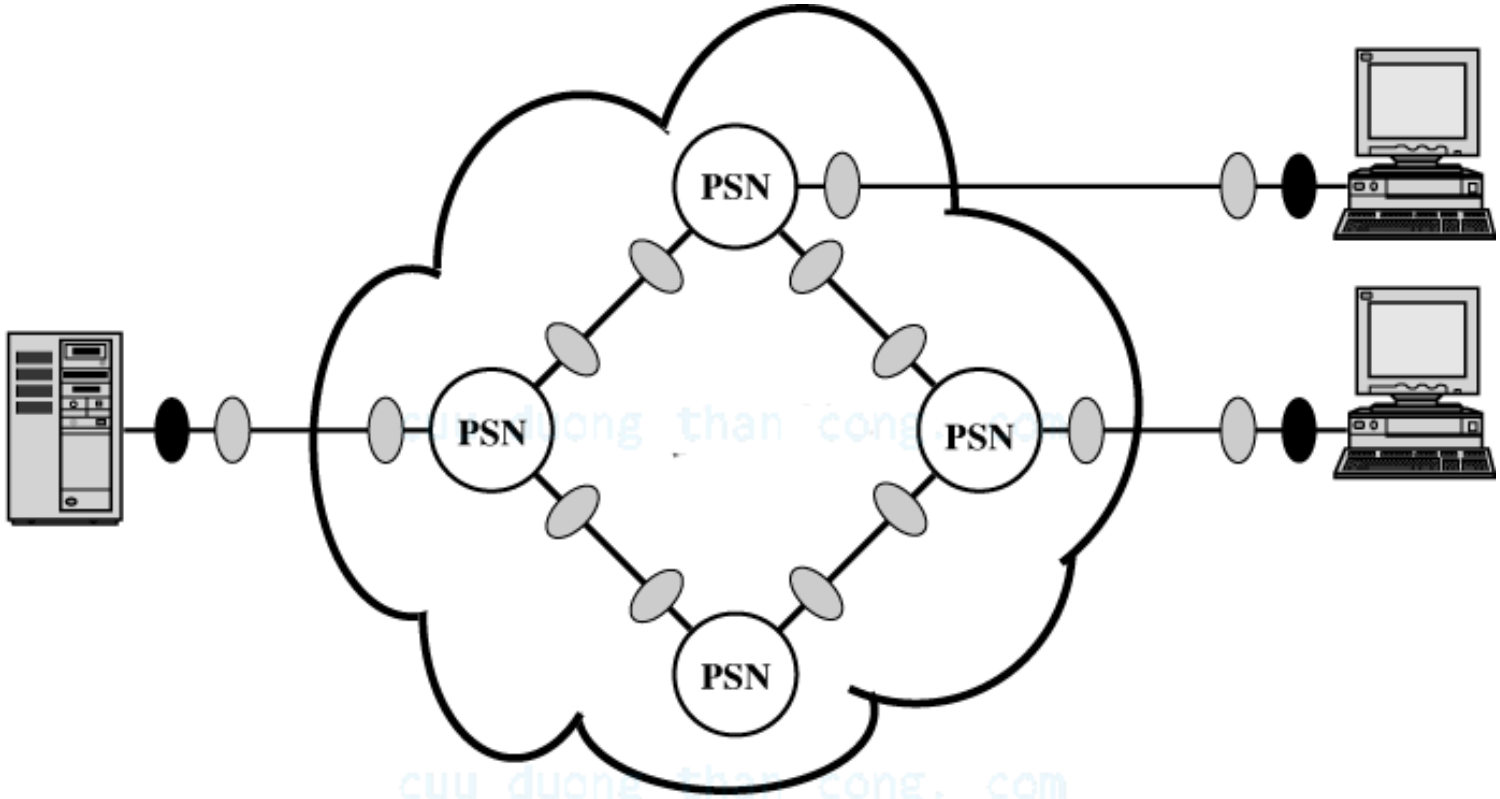
Mã hóa liên kết

- Công cụ mã hóa được sắp đặt ở 2 đầu của mọi liên kết có nguy cơ bị tấn công
- Đảm bảo an ninh 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 ninh 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 và càng an ninh nhưng càng phức tạp với nhiều thực thể và khóa

Kết hợp các phương án mã hóa



Công cụ mã hóa đầu cuối

Công cụ mã hóa liên kết

PSN : Packet-switching node

Quản lý khóa bí mật

- Vấn đề đối với mã hóa đối xứng là làm sao phân phối khóa an ninh đến các bên truyền tin
  - Thường hệ thống mất an ninh 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 ninh

Các cách phân phối khóa

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