ĐẠI HỌC KINH TẾ KỸ THUẬT CÔNG NGHIỆP
KHOA CÔNG NGHỆ THÔNG TIN

AN TOÀN THÔNG TIN

Ths. Nguyễn Hoàng Chiến

Bộ môn Mạng máy tính & công nghệ đa phương tiện

Nhchien@uneti.edu.vn

Năm học 2023

Tài liệu tham khảo

- Sách tham khảo chính
  - William Stallings. *Cryptography and Network Security : Principles and Practice, Fourth Edition*. Prentice Hall, 2005.
- Sách tham khảo phụ
  - Charlie Kaufman, Radia Perlman, Mike Speciner. *Network Security: Private Communication in a Public World, Second Edition*. Prentice Hall, 2002.
  - Matt Bishop. *Computer Security: Art and Science*. Addison Wesley, 2002.
  - Man Young Rhee. *Internet Security: Cryptographic principles, algorithms and protocols*. John Wiley & Sons, 2003.
- Website
  - *http://williamstallings.com*

Chương 1
Giới thiệu

Bối cảnh

- Nhu cầu đảm bảo an toàn thông tin có những biến đổi lớn
  - Trước đây
    - Chỉ cần các phương tiện vật lý và hành chính
  - Từ khi có máy tính
    - Cần các công cụ tự động bảo vệ tệp tin và các thông tin khác lưu trữ trong máy tính
  - Từ khi có các phương tiện truyền thông và mạng
    - Cần các biện pháp bảo vệ dữ liệu truyền trên mạng

Các khái niệm

- An toàn thông tin
  - Liên quan đến các yếu tố tài nguyên, nguy cơ, hành động tấn công, yếu điểm, và điều khiển
- An toàn máy tính
  - Các công cụ bảo vệ dữ liệu và phòng chống tin tặc
- An toàn mạng
  - Các biện pháp bảo vệ dữ liệu truyền trên mạng
- An toàn liên mạng
  - Các biện pháp bảo vệ dữ liệu truyền trên một tập hợp các mạng kết nối với nhau

Mục tiêu môn học

- Chú trọng an toàn liên mạng
- Nghiên cứu các biện pháp ngăn cản, phòng chống, phát hiện và khắc phục các vi phạm an toàn liên quan đến truyền tải thông tin

Đảm bảo an toàn thông tin

- Để thực hiện có hiệu quả cần đề ra một phương thức chung cho việc xác định các nhu cầu về an toàn thông tin
- Phương thức đưa ra sẽ xét theo 3 mặt
  - Hành động tấn công
  - Cơ chế an toàn
  - Dịch vụ an toàn

Dịch vụ an toàn

- Là một dịch vụ nâng cao độ an toàn của các hệ thống xử lý thông tin và các cuộc truyền dữ liệu trong một tổ chức
- Nhằm phòng chống các hành động tấn công
- Sử dụng một hay nhiều cơ chế an toàn
- Có các chức năng tương tự như đảm bảo an toàn tài liệu vật lý
- Một số đặc trưng của tài liệu điện tử khiến việc cung cấp các chức năng đảm bảo an toàn khó khăn hơn

Cơ chế an toàn

- Là cơ chế định ra để phát hiện, ngăn ngừa và khắc phục một hành động tấn công
- Không một cơ chế đơn lẻ nào có thể hỗ trợ tất cả các chức năng đảm bảo an toàn thông tin
- Có một yếu tố đặc biệt hậu thuẫn nhiều cơ chế an toàn sử dụng hiện nay là các kỹ thuật mật mã
- Môn học sẽ chú trọng lĩnh vực mật mã

Hành động tấn công

- Là hành động phá hoại an toàn thông tin của một tổ chức
- An toàn thông tin là những cách thức ngăn ngừa các hành động tấn công, nếu không được thì phát hiện và khắc phục hậu quả
- Các hành động tấn công có nhiều và đa dạng
- Chỉ cần tập trung vào những thể loại chung nhất
- Lưu ý : nguy cơ tấn công và hành động tấn công thường được dùng đồng nghĩa với nhau

Kiến trúc an toàn OSI

- Kiến trúc an toàn cho OSI theo khuyến nghị X.800 của ITU-T
- Định ra một phương thức chung cho việc xác định các nhu cầu về an toàn thông tin
- Cung cấp một cái nhìn tổng quan về các khái niệm môn học sẽ đề cập đến
- Chú trọng đến các dịch vụ an toàn, các cơ chế an toàn và các hành động tấn công

Các dịch vụ an toàn

- Theo X.800
  - Dịch vụ an toàn là dịch vụ cung cấp bởi một tầng giao thức của các hệ thống mở kết nối nhằm đảm bảo an toàn cho các hệ thống và các cuộc truyền dữ liệu
  - Có 5 loại hình
- Theo RFC 2828
  - Dịch vụ an toàn là dịch vụ xử lý hoặc truyền thông cung cấp bởi một hệ thống để bảo vệ tài nguyên theo một cách thức nhất định

Các dịch vụ an toàn X.800

- Xác thực
  - Đảm bảo thực thể truyền thông đúng là nó
- Điều khiển truy nhập
  - Ngăn không cho sử dụng trái phép tài nguyên
- Bảo mật dữ liệu
  - Bảo vệ dữ liệu khỏi bị tiết lộ trái phép
- Toàn vẹn dữ liệu
  - Đảm bảo nhận dữ liệu đúng như khi gửi
- Chống chối bỏ
  - Ngăn không cho bên liên quan phủ nhận hành động

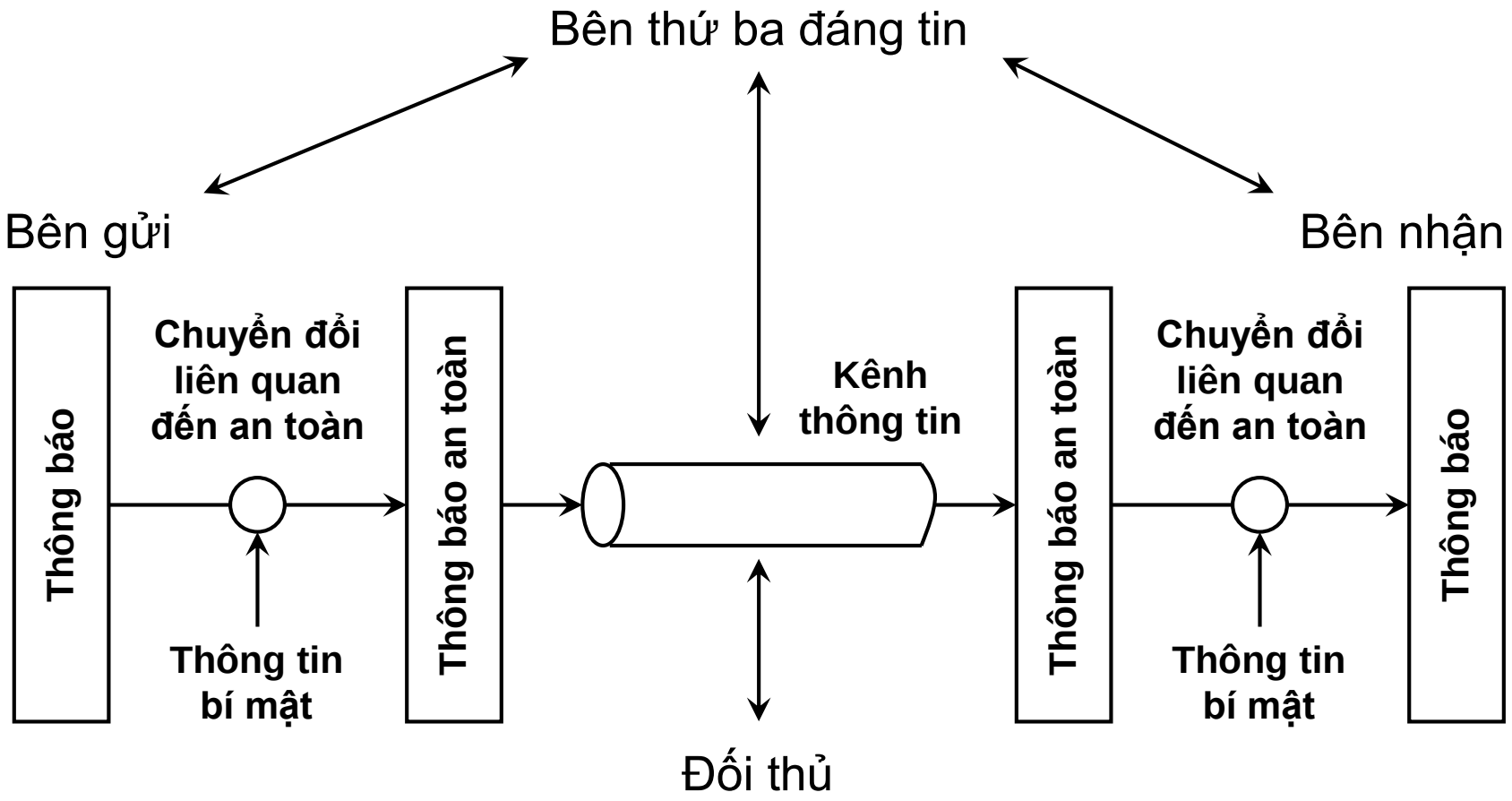
Các cơ chế an toàn X.800

- Các cơ chế an toàn chuyên biệt
  - Mã hóa, chữ ký số, điều khiển truy nhập, toàn vẹn dữ liệu, trao đổi xác thực, độn tin truyền, điều khiển định tuyến, công chứng
- Các cơ chế an toàn phổ quát
  - Tính năng đáng tin, nhãn an toàn, phát hiện sự kiện, dấu vết kiểm tra an toàn, khôi phục an toàn

Các hành động tấn công

- Các hành động tấn công thụ động
  - Nghe trộm nội dung thông tin truyền tải
  - Giám sát và phân tích luồng thông tin lưu chuyển
- Các hành động tấn công chủ động
  - Giả danh một thực thể khác
  - Phát lại các thông báo trước đó
  - Sửa đổi các thông báo đang lưu chuyển
  - Từ chối dịch vụ

Mô hình an toàn mạng



Mô hình an toàn mạng

- Yêu cầu
  - Thiết kế một giải thuật thích hợp cho việc chuyển đổi liên quan đến an toàn
  - Tạo ra thông tin bí mật (khóa) đi kèm với giải thuật
  - Phát triển các phương pháp phân bổ và chia sẻ thông tin bí mật
  - Đặc tả một giao thức sử dụng bởi hai bên gửi và nhận dựa trên giải thuật an toàn và thông tin bí mật, làm cơ sở cho một dịch vụ an toàn

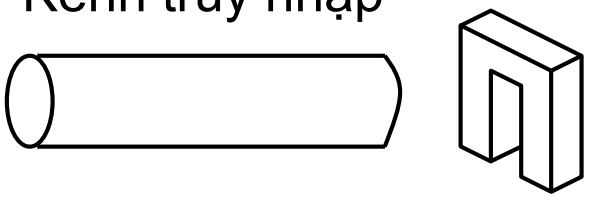
Mô hình an toàn truy nhập mạng

Đối thủ

- Con người

- Phần mềm

Kênh truy nhập



Chức năng gác cổng

Các tài nguyên tính toán (bộ xử lý, bộ nhớ, ngoại vi)

Dữ liệu

Các tiến trình

Phần mềm

Các điều khiển an toàn bên trong

Mô hình an toàn truy nhập mạng

- Yêu cầu
  - Lựa chọn các chức năng gác cổng thích hợp để định danh người dùng
  - Cài đặt các điều khiển an toàn để đảm bảo chỉ những người dùng được phép mới có thể truy nhập được vào các thông tin và tài nguyên tương ứng
- Các hệ thống máy tính đáng tin cậy có thể dùng để cài đặt mô hinh này

Chương 2

MÃ HÓA ĐỐI XỨNG

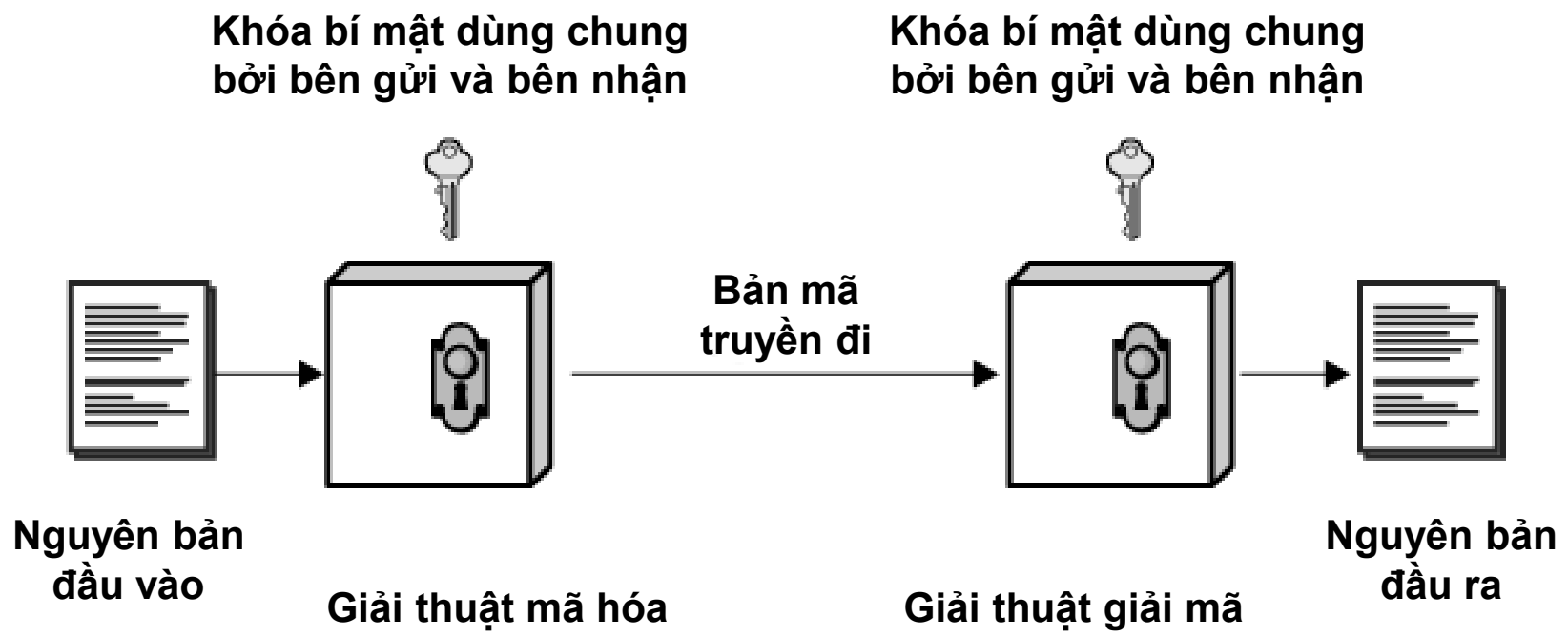
Hai kỹ thuật mã hóa chủ yếu

- 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 toàn 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 toàn hệ mã hóa

- An toàn 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 toàn vô điều kiện
- An toàn 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) mod 26
    - Giải mã : p = D(C) = (C - k) mod 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

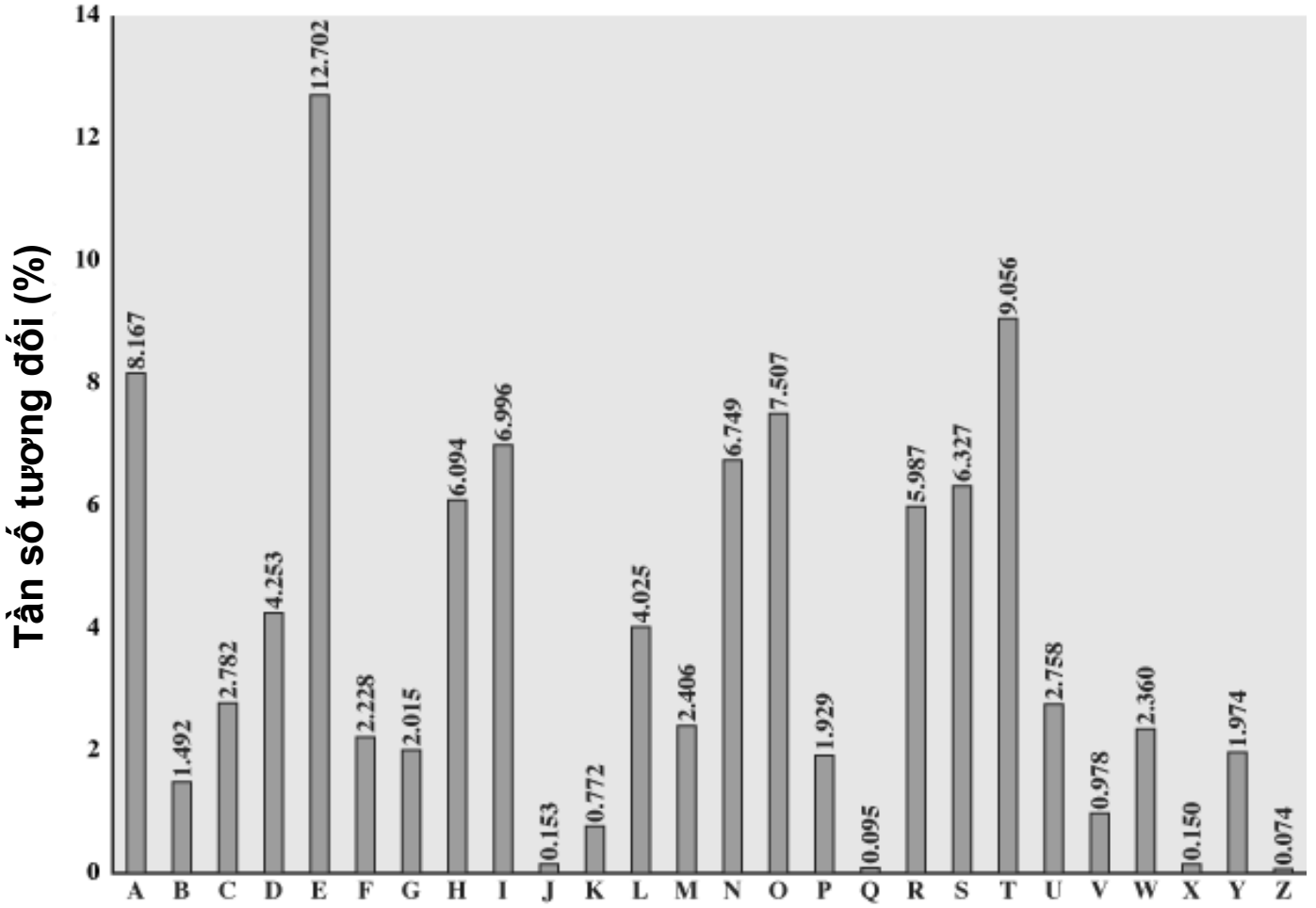
    M N B V C X Z A S D F G H J K L P O I U Y T R E W Q
  - Nguyên bản

    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 toàn 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 toàn

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

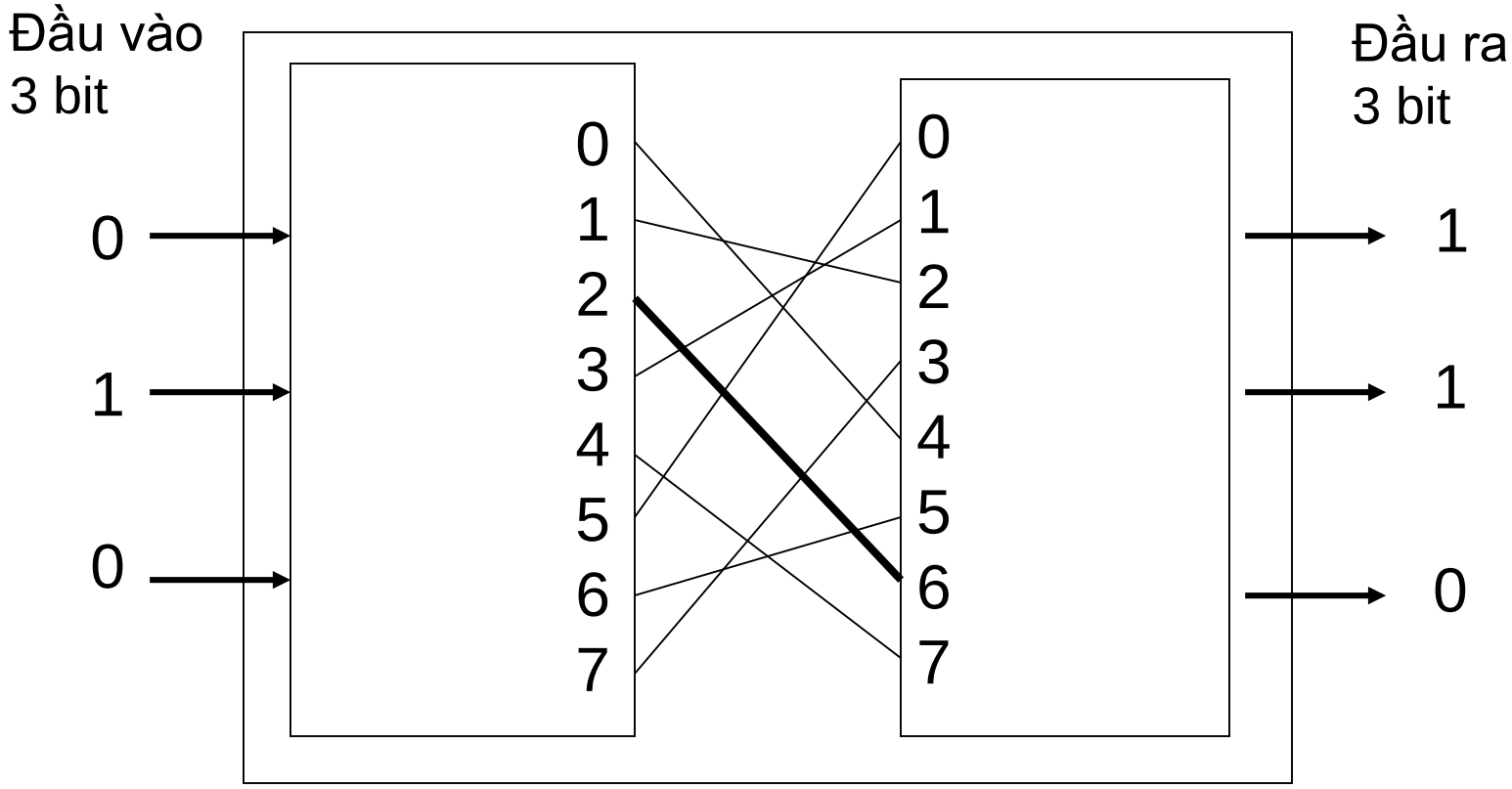
Mã hóa khối

- 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 ($\geq$ 64 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

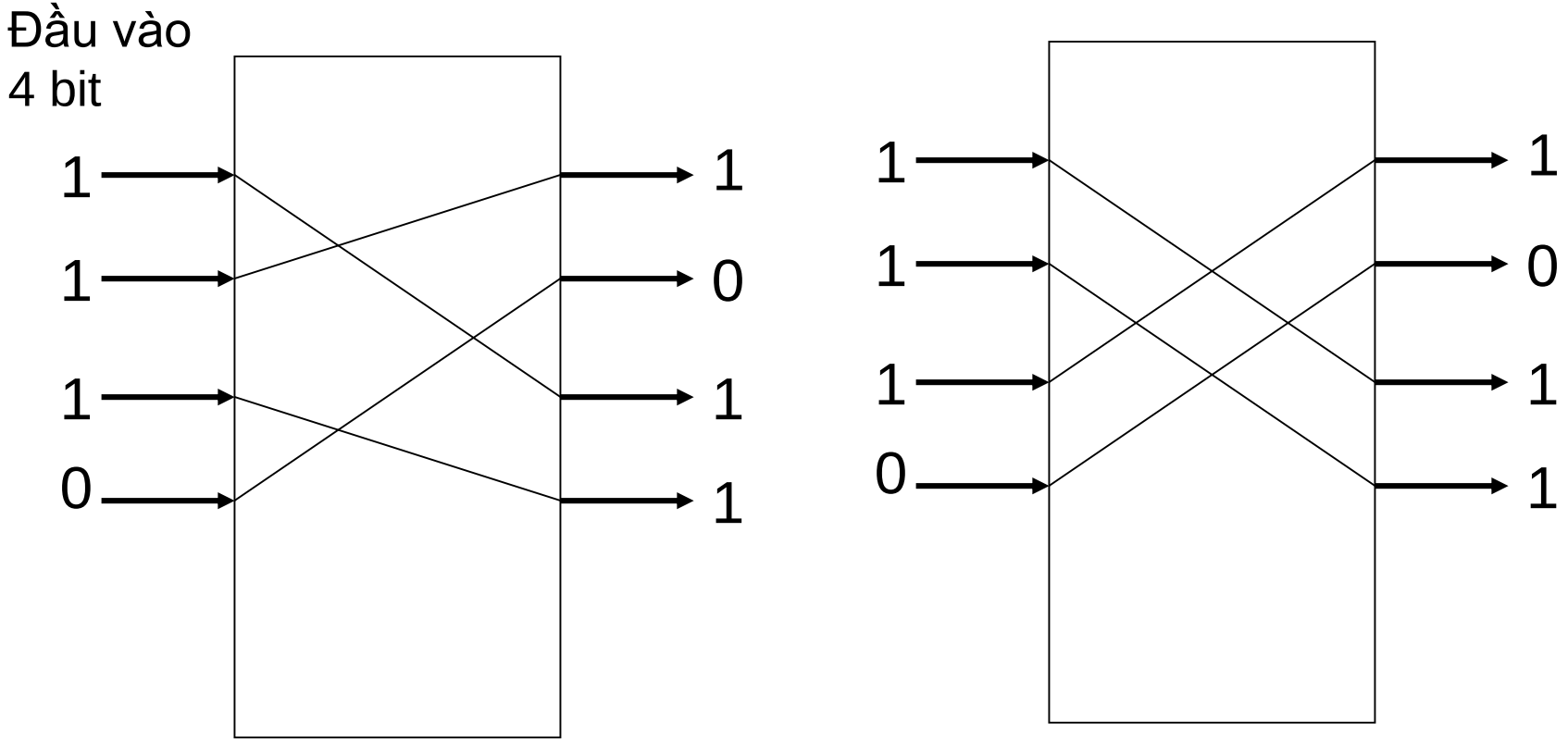
- Mạng thay thế (S) - hoán vị (P) đề xuất bởi Claude Shannon vào năm 1949
- Là cơ sở của các hệ mã hóa khối hiện đại
- Dựa trên 2 phép mã hóa cổ điển
  - Phép thay thế : Hộp S
  - Phép hoán vị : Hộp P
- Đan xen các chức năng
  - Khuếch tán : Hộp P (kết hợp với hộp S)
    - Phát tỏa cấu trúc thống kê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

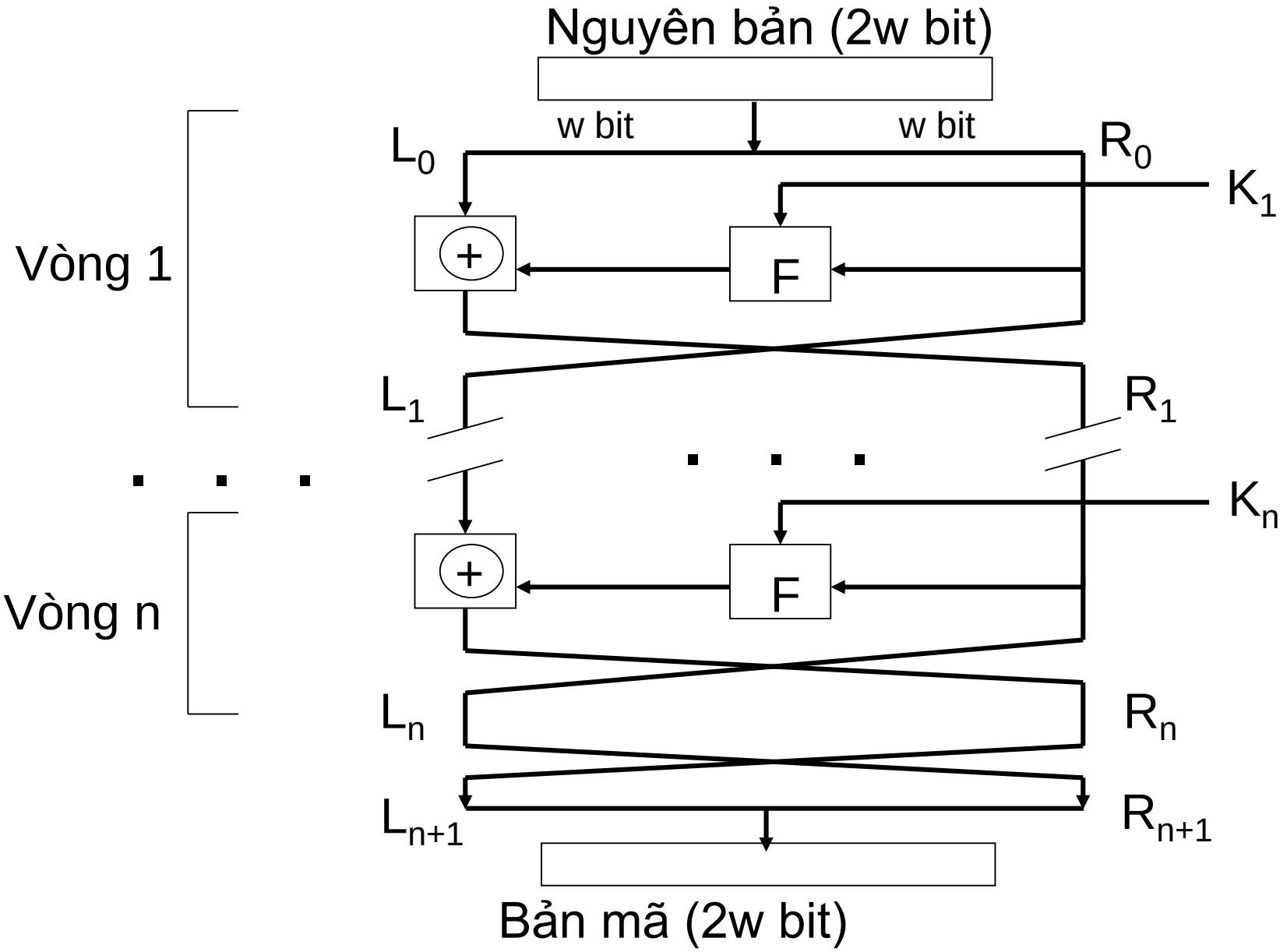
Hộp P



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


Nguyên bản (2w bit)
w bit
w bit
L_0
R_0
K_1
Vòng 1
+
F
L_1
R_1
K_n
Vòng n
L_n
R_n
L_{n+1}
R_{n+1}
Bản mã (2w bit)

Các đặc trưng hệ Feistel

- Độ dài khối
  - Khối càng lớn càng an toàn (thường 64 bit)
- Độ dài khóa
  - Khóa càng dài càng an toàn (thường 128 bit)
- Số vòng
  - Càng nhiều vòng càng an toàn (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

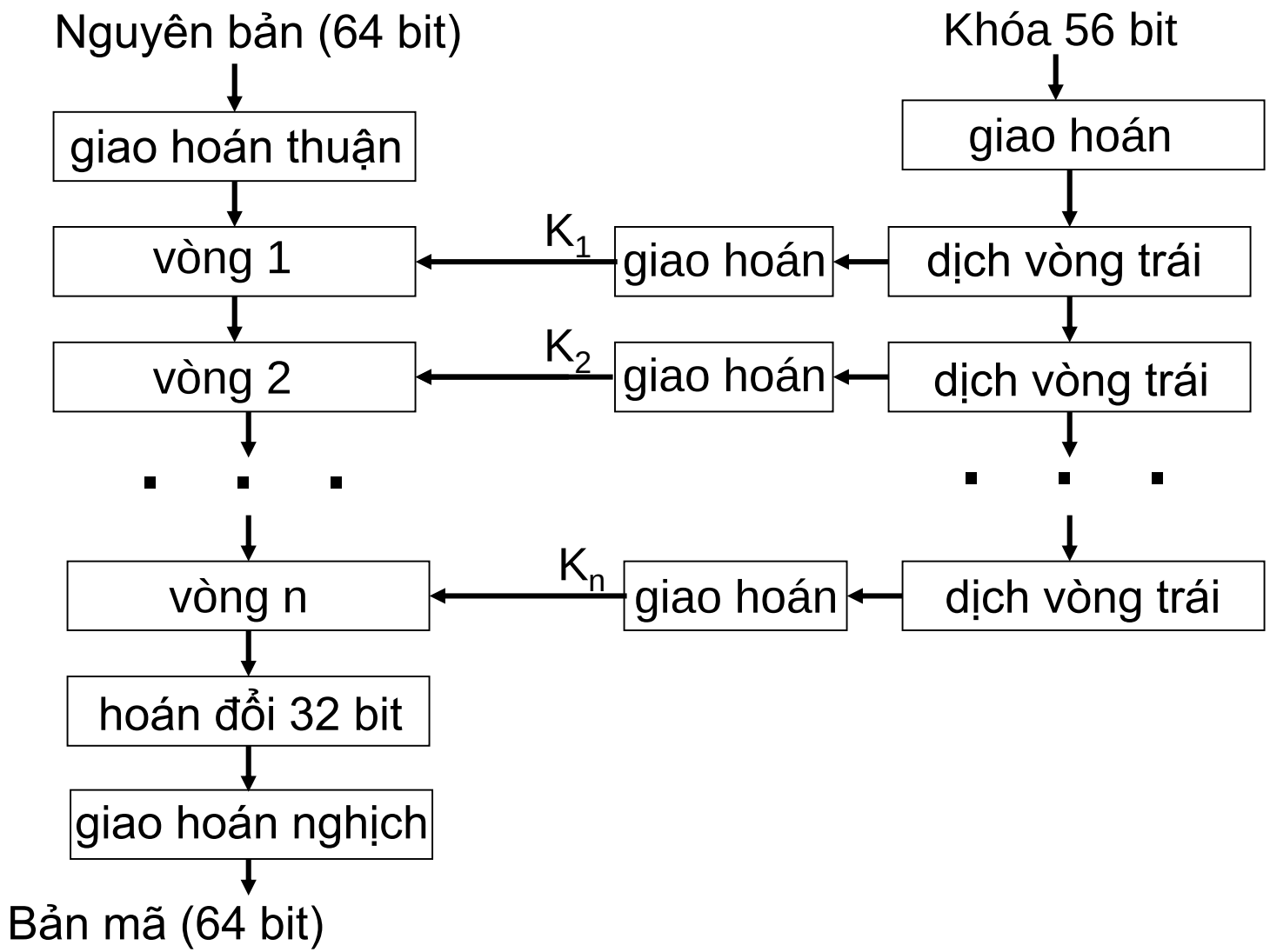
Giải mã Feistel

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

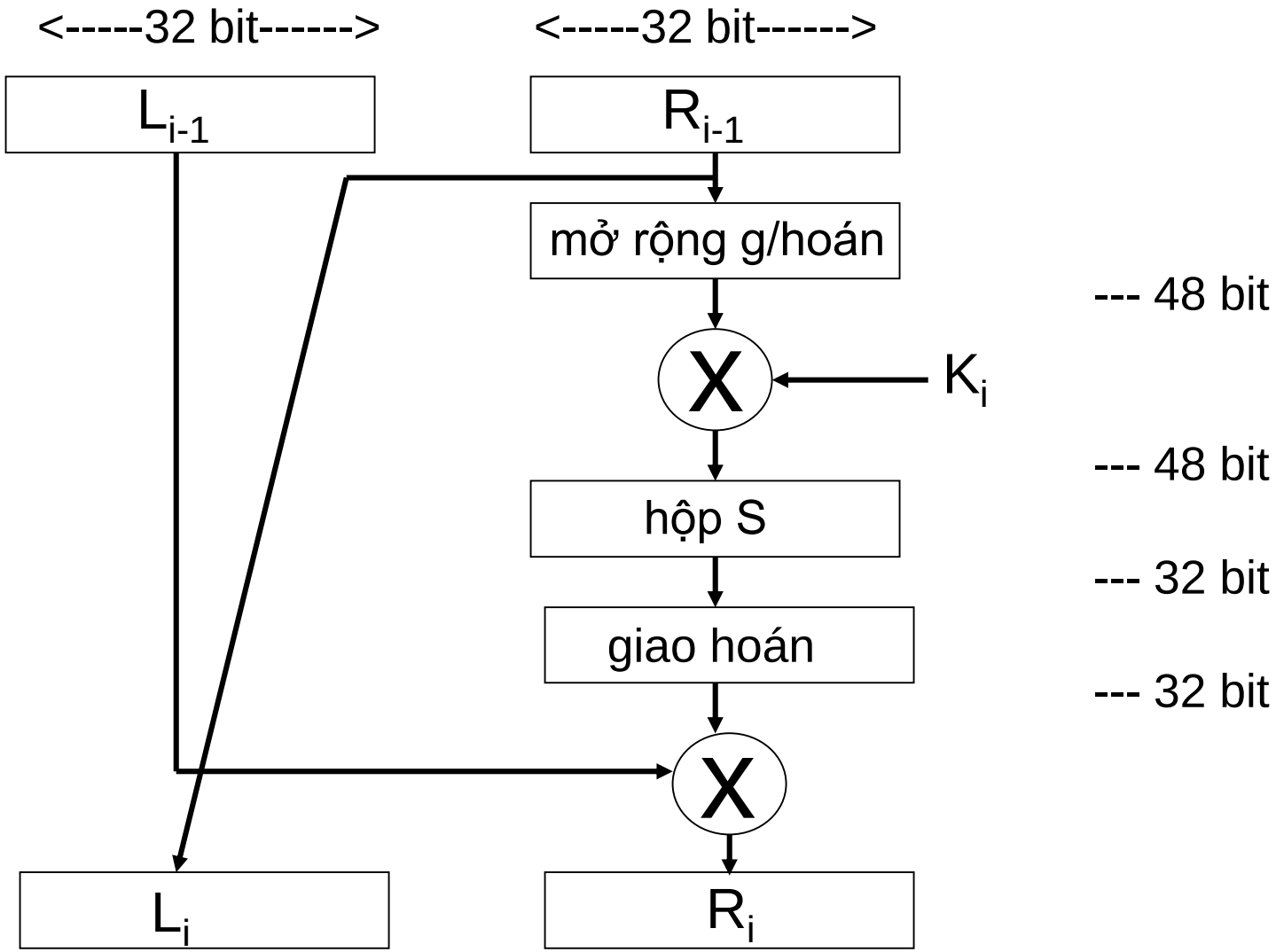
Chuẩn mã hóa dữ liệu

- 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 toàn

Giải thuật mã hóa DES



Một vòng DES


<-----32 bit------>
<-----32 bit------>
L_{i-1}
R_{i-1}
mở rộng g/hoán
X
K_i
hộp S
giao hoán
X
L_i
R_i
--- 48 bit
--- 48 bit
--- 32 bit
--- 32 bit

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 toàn 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"
  - $C = E_{K_2}(E_{K_1}(p)) \Rightarrow X = E_{K_1}(p) = D_{K_2}(C)$
  - 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 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

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 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

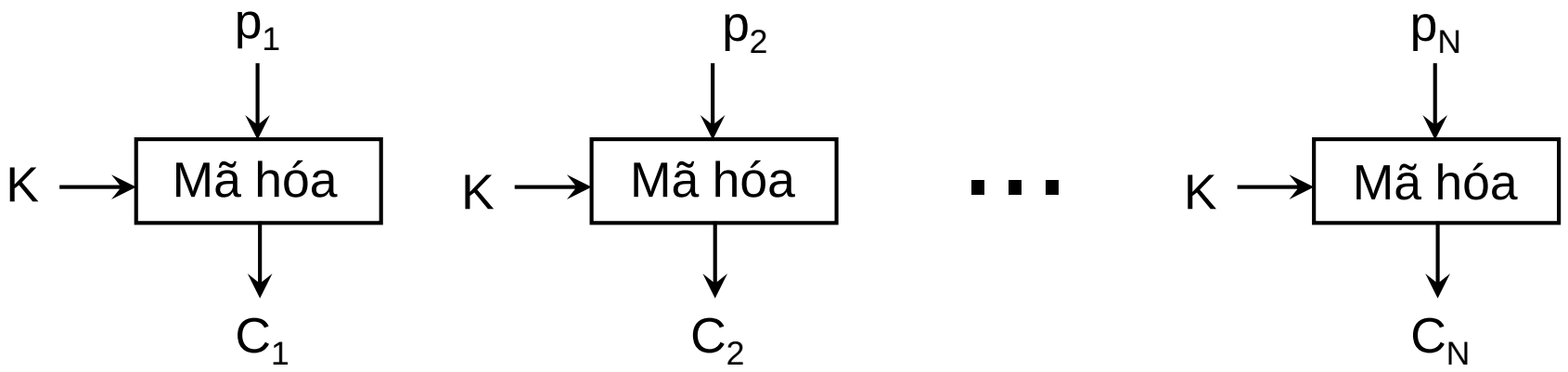
Các hệ mã hóa khối khác (2)

- 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

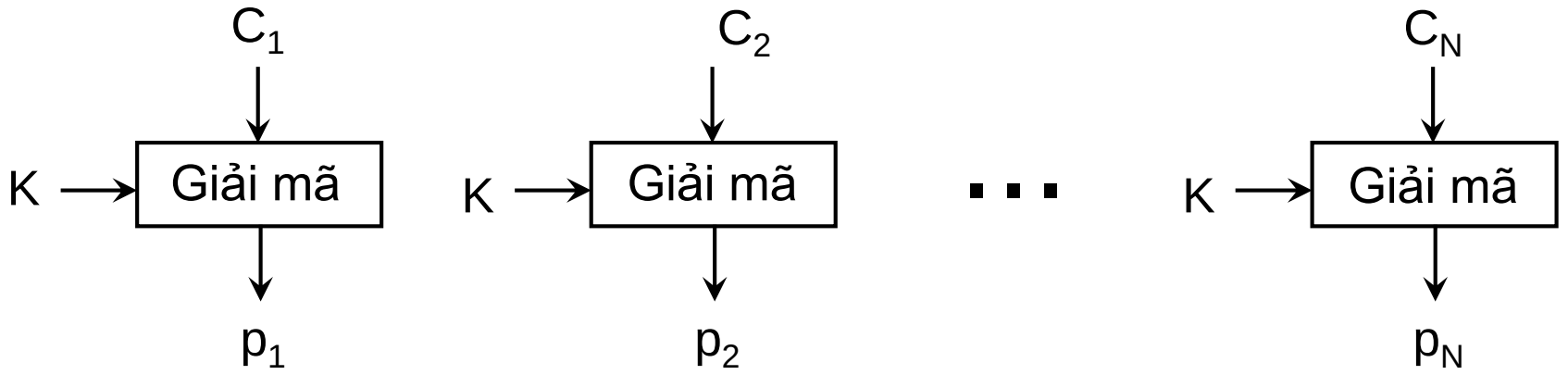
Các phương thức mã hóa khối

- 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

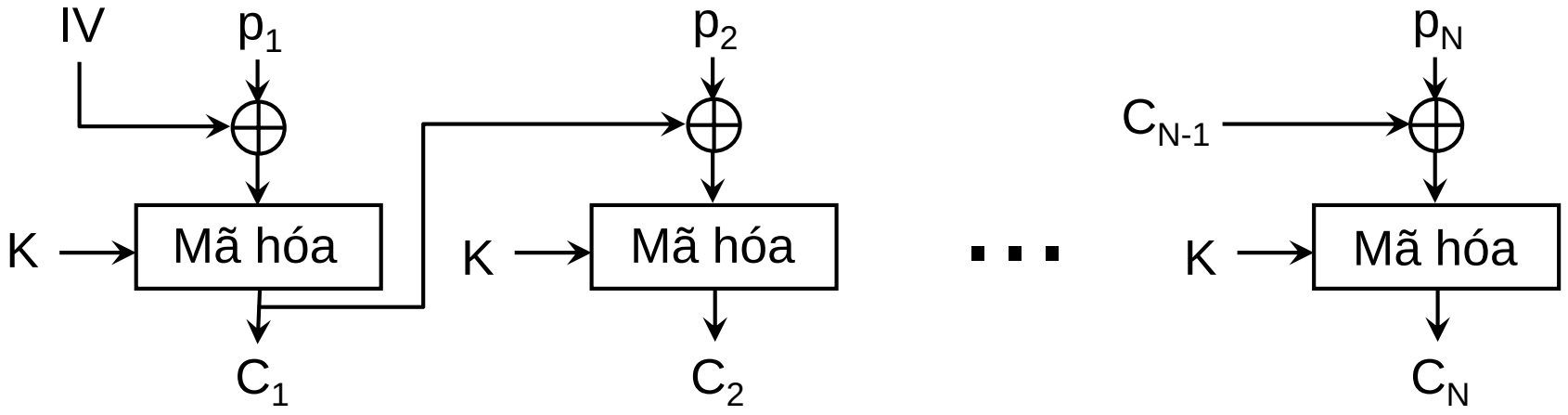


Giải mã

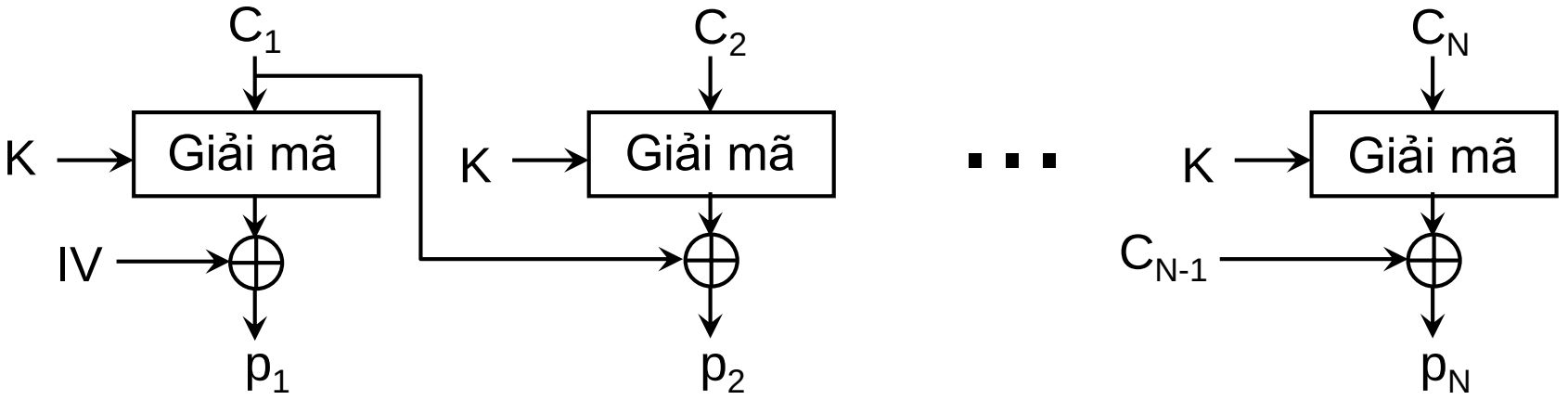
Đánh giá ECB

- 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

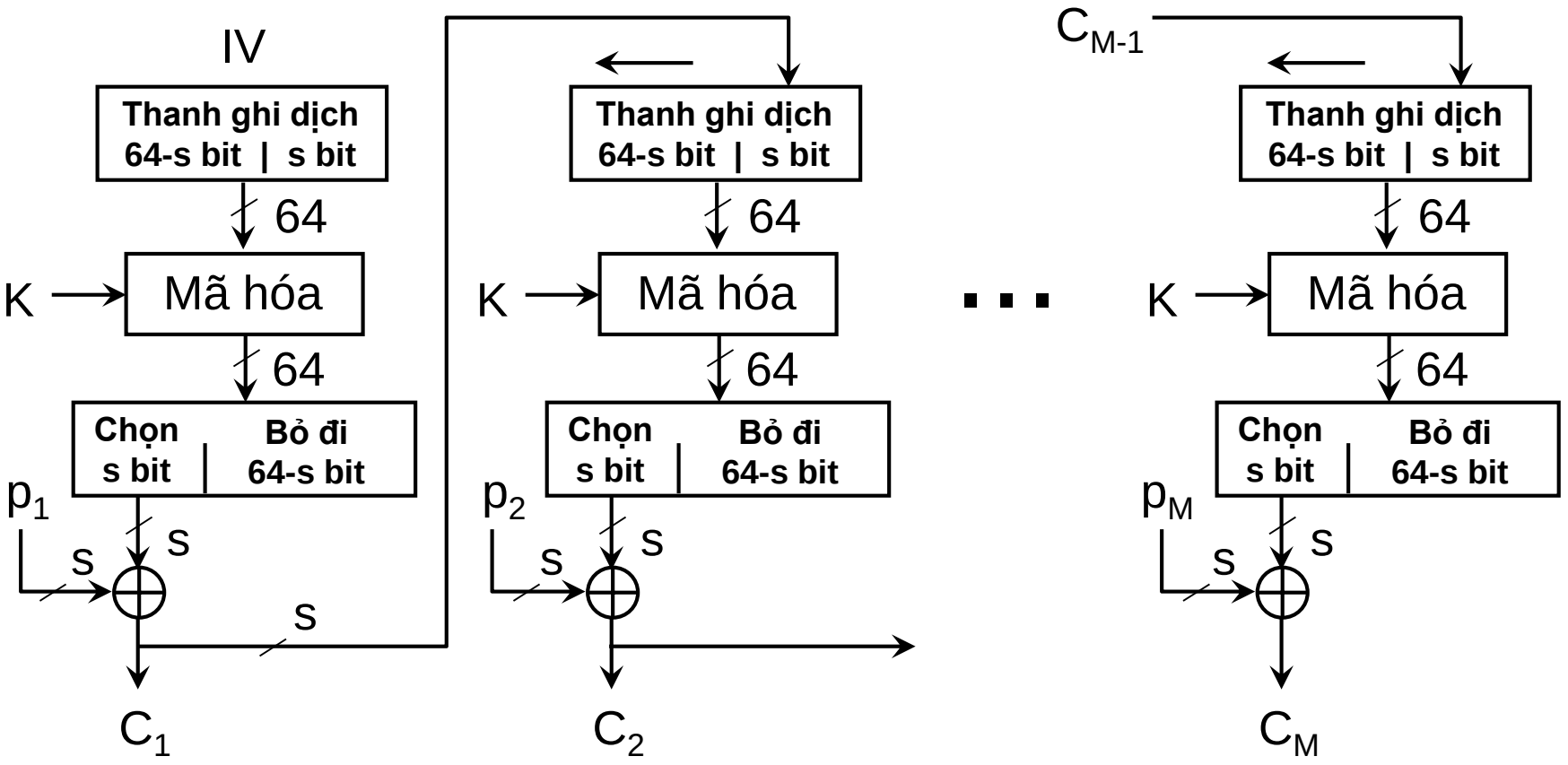


Giải mã

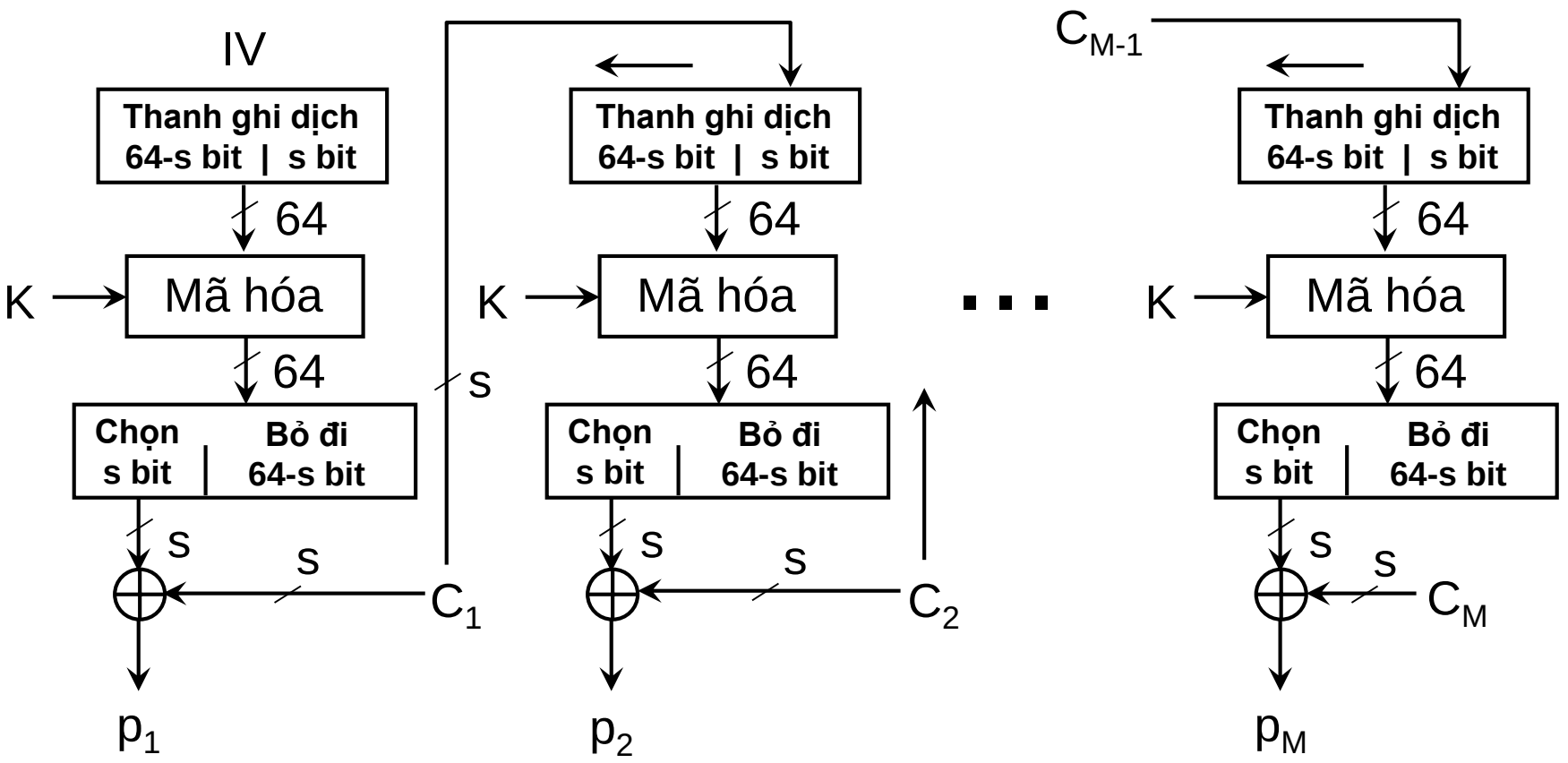
Đánh giá CBC

- 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

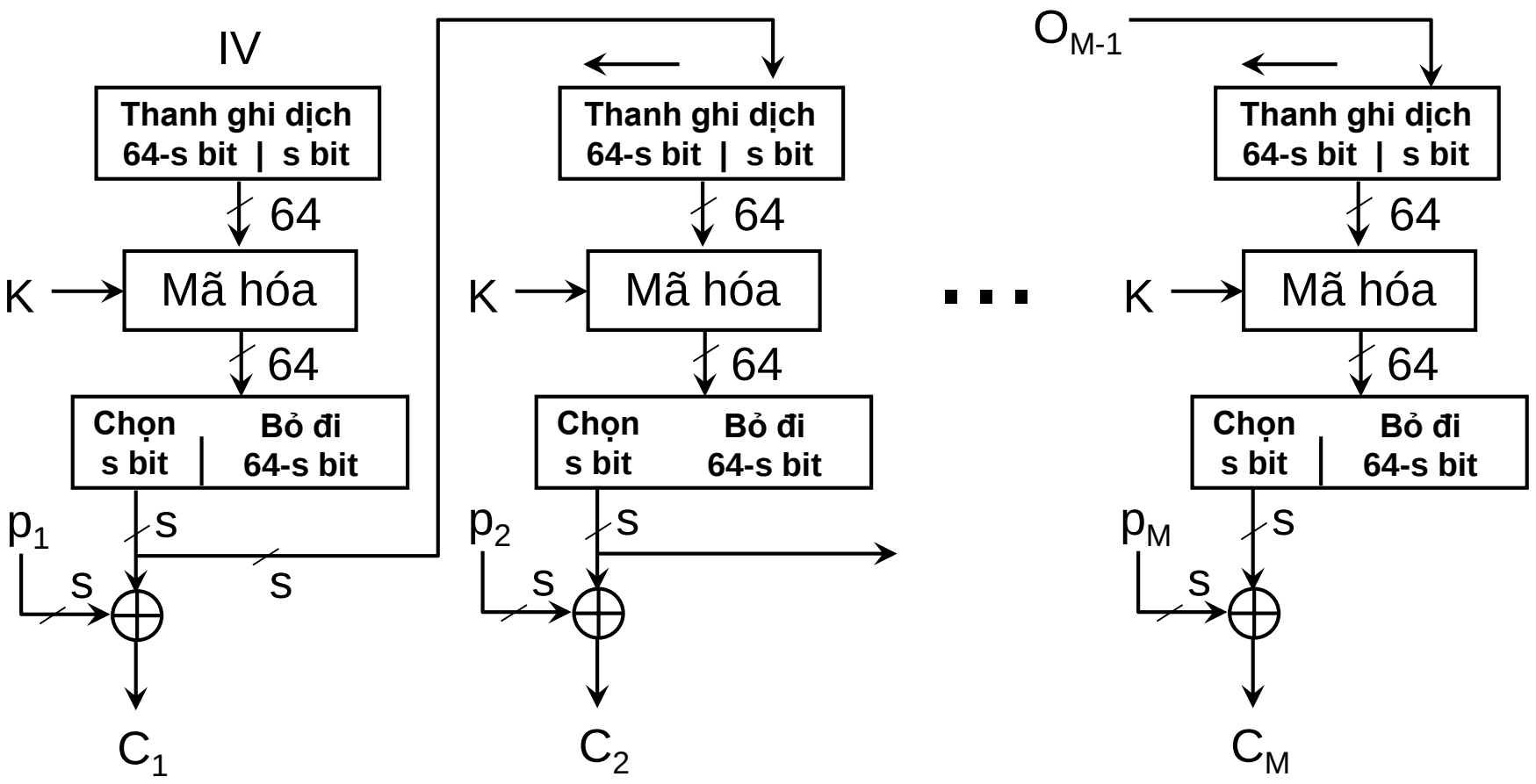
Giải mã CFB


IV
Thanh ghi dịch
64-s bit | s bit
64
K
Mã hóa
64
Chọn
s bit
Bỏ đi
64-s bit
s
s
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

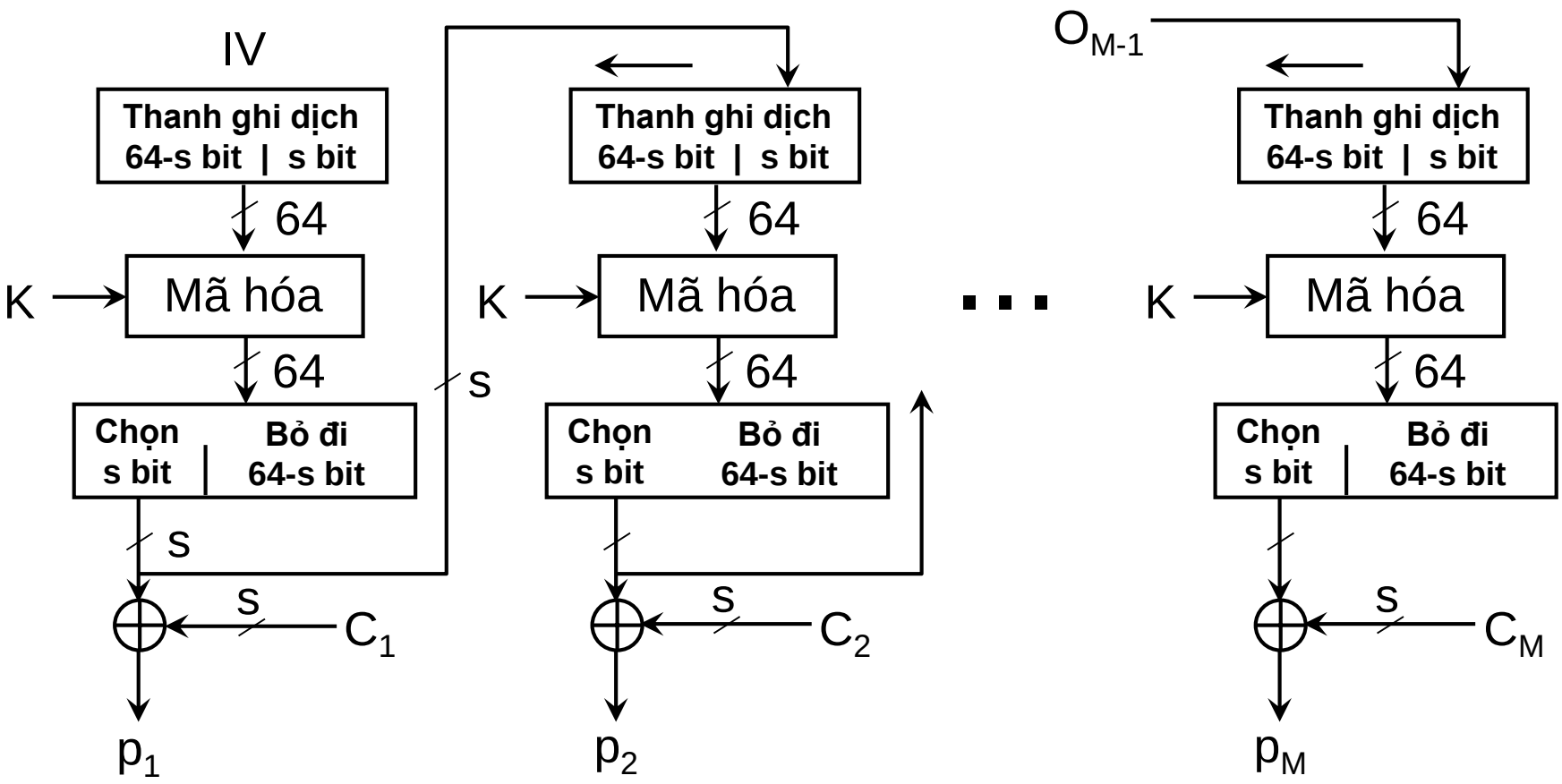
Đánh giá CFB

- 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

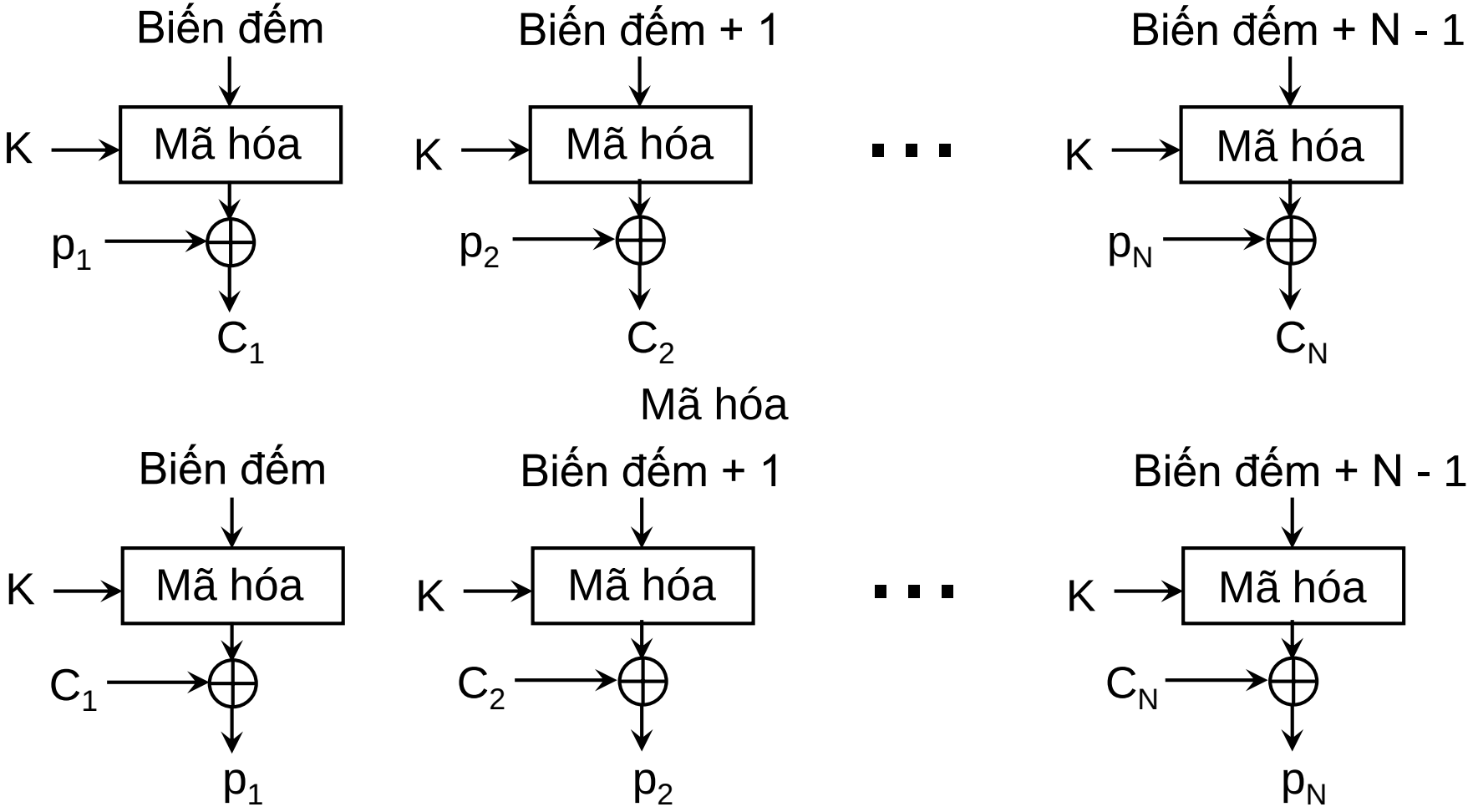
Giải mã OFB


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
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 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)

Bố trí công cụ 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

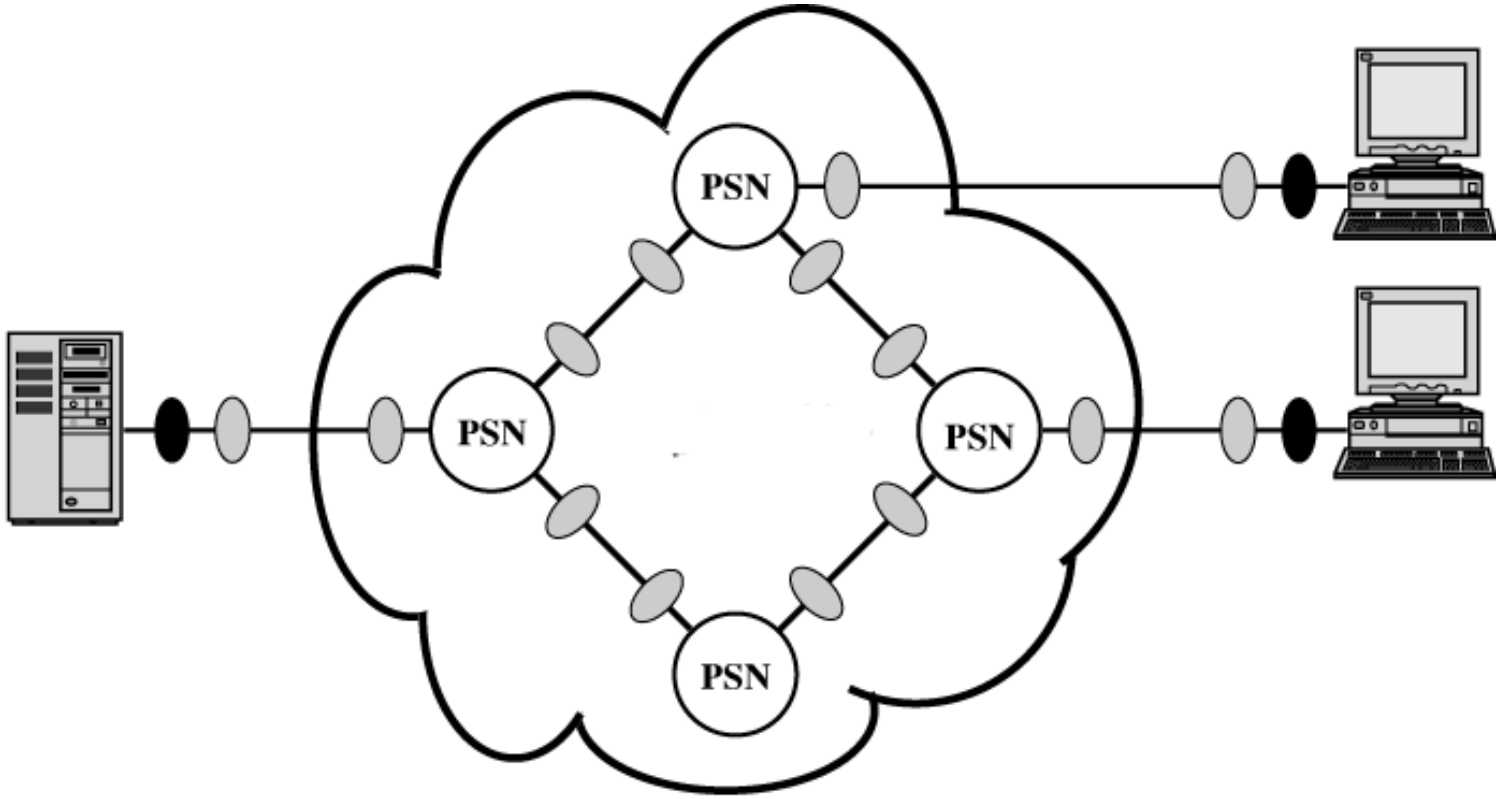
Mã hóa liên kết

- 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 và càng an toàn 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 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

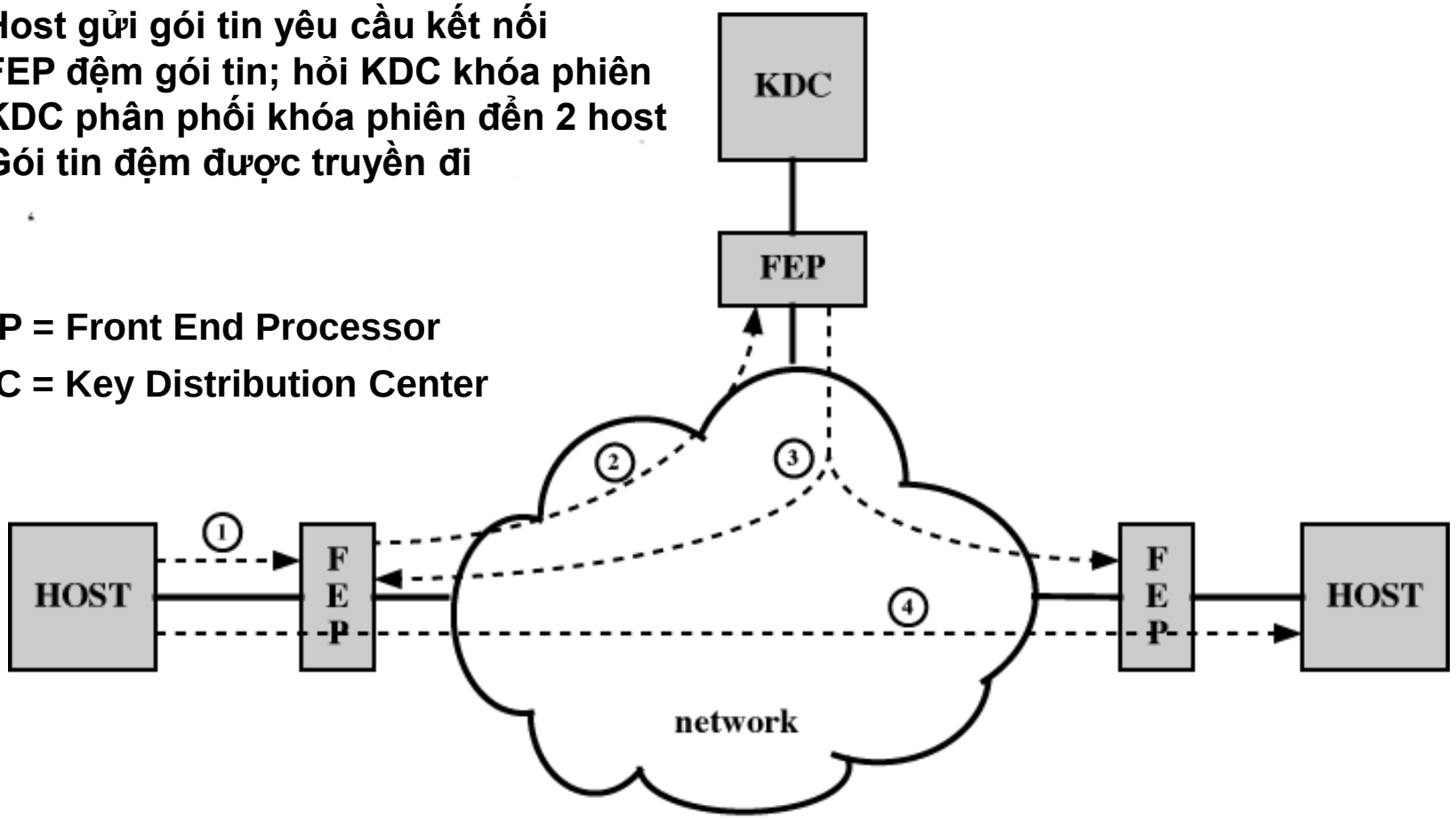
- Khóa có thể được chọn bởi bên A và gửi theo đường vật lý đến bên B
- Khóa có thể được chọn bởi một bên thứ ba, sau đó gửi theo đường vật lý đến A và B
- Nếu A và B đã có một khóa dùng chung thì một bên có thể gửi khóa mới đến bên kia, sử dụng khóa cũ để mã hóa khóa mới
- Nếu mỗi bên A và B đều có một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



Chương 3

MẬT MÃ KHÓA CÔNG KHAI

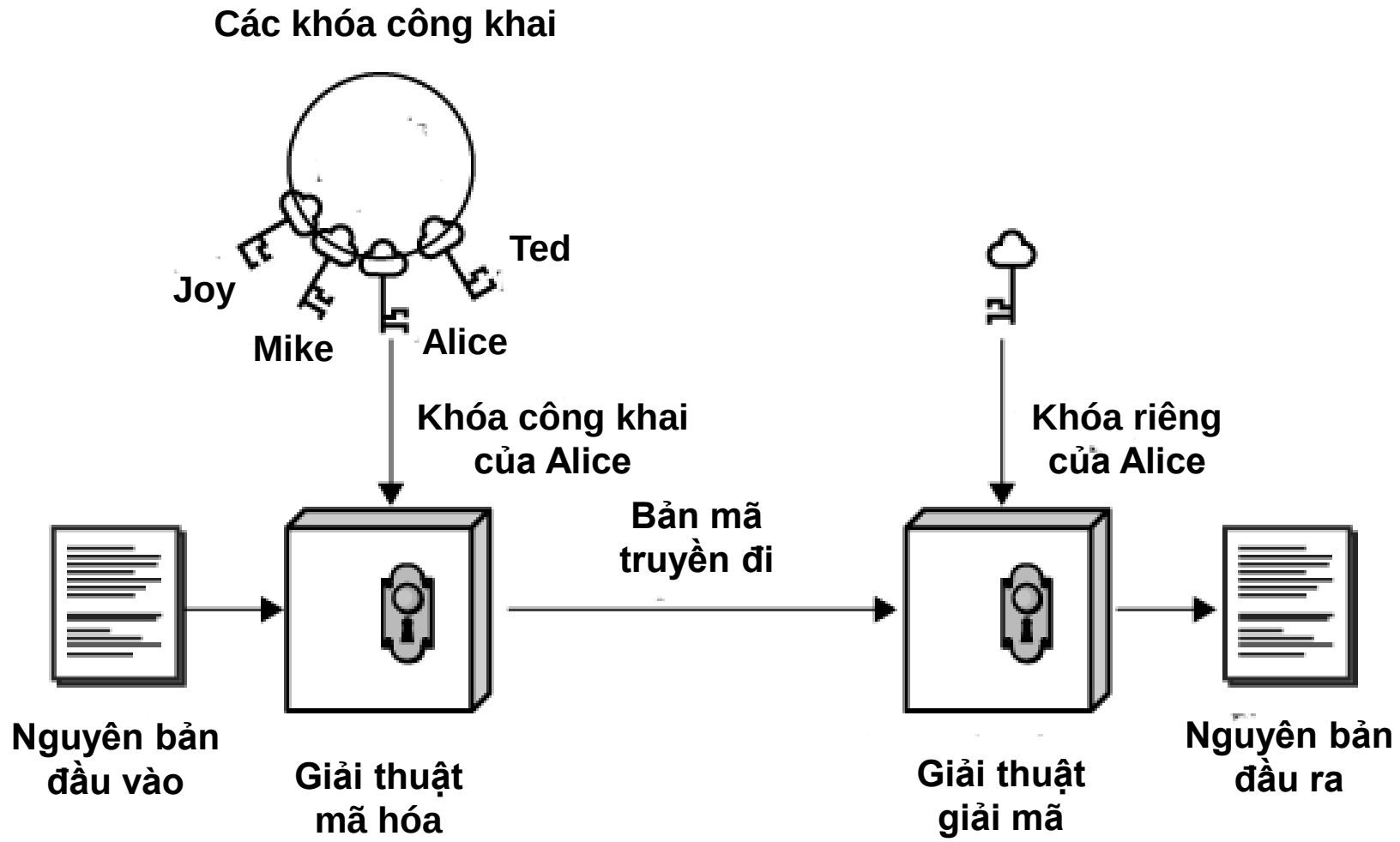
Giới thiệu

- Những hạn chế của mật mã đối xứng
  - Vấn đề phân phối khóa
    - Khó đảm bảo chia sẻ mà không làm lộ khóa bí mật
    - Trung tâm phân phối khóa có thể bị tấn công
  - Không thích hợp cho chữ ký số
    - Bên nhận có thể làm giả thông báo nói nhận được từ bên gửi
- Mật mã khóa công khai đề xuất bởi Whitfield Diffie và Martin Hellman vào năm 1976
  - Khắc phục những hạn chế của mật mã đối xứng
  - Có thể coi là bước đột phá quan trọng nhất trong lịch sử của ngành mật mã
  - Bổ xung chứ không thay thế mật mã đối xứng

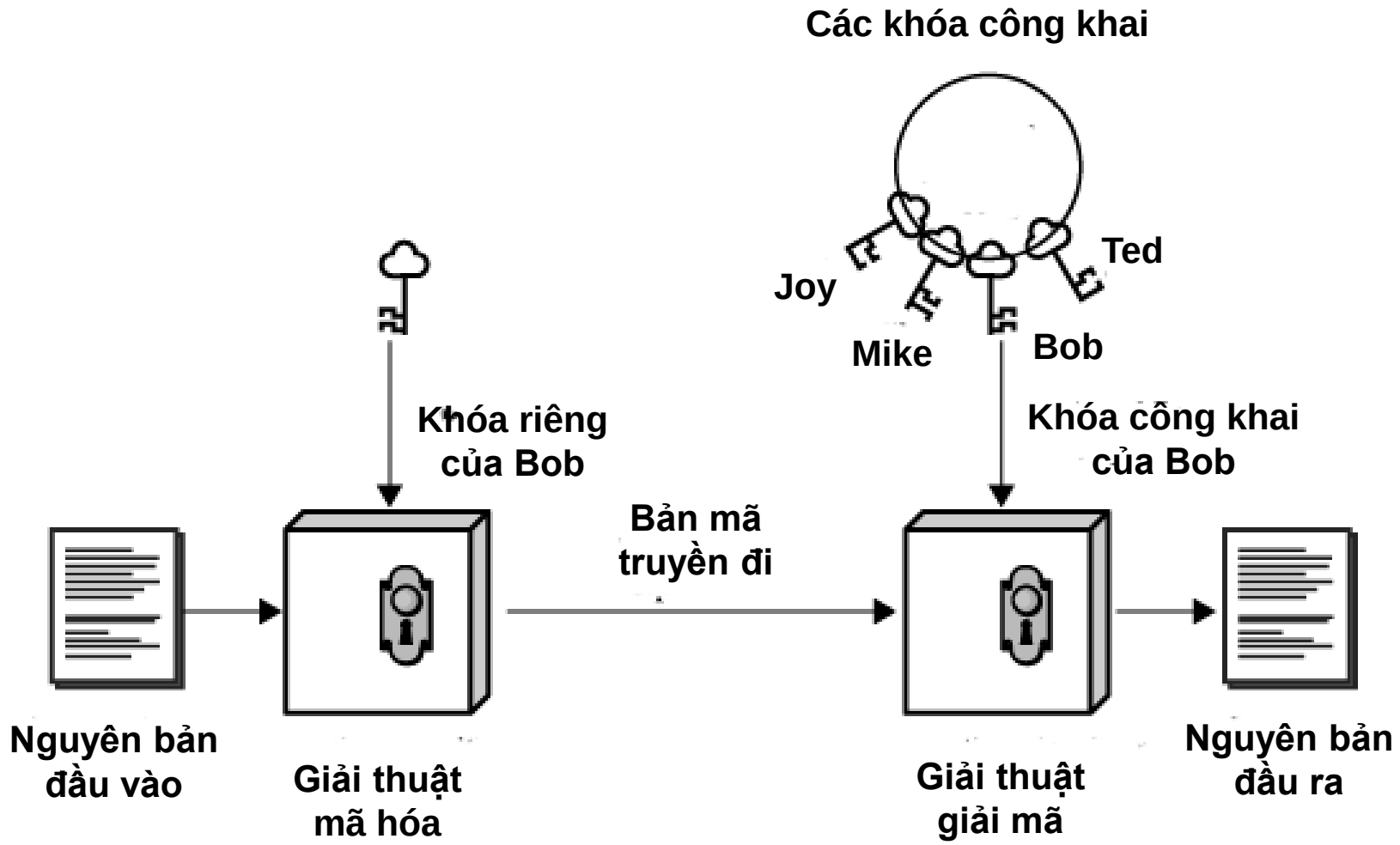
Đặc điểm mật mã khóa công khai

- Còn gọi là mật mã hai khóa hay bất đối xứng
- Các giải thuật khóa công khai sử dụng 2 khóa
  - Một khóa công khai
    - Ai cũng có thể biết
    - Dùng để mã hóa thông báo và thẩm tra chữ ký
  - Một khóa riêng
    - Chỉ nơi giữ được biết
    - Dùng để giải mã thông báo và ký (tạo ra) chữ ký
- Có tính bất đối xứng
  - Bên mã hóa không thể giải mã thông báo
  - Bên thẩm tra không thể tạo chữ ký

Mã hóa khóa công khai



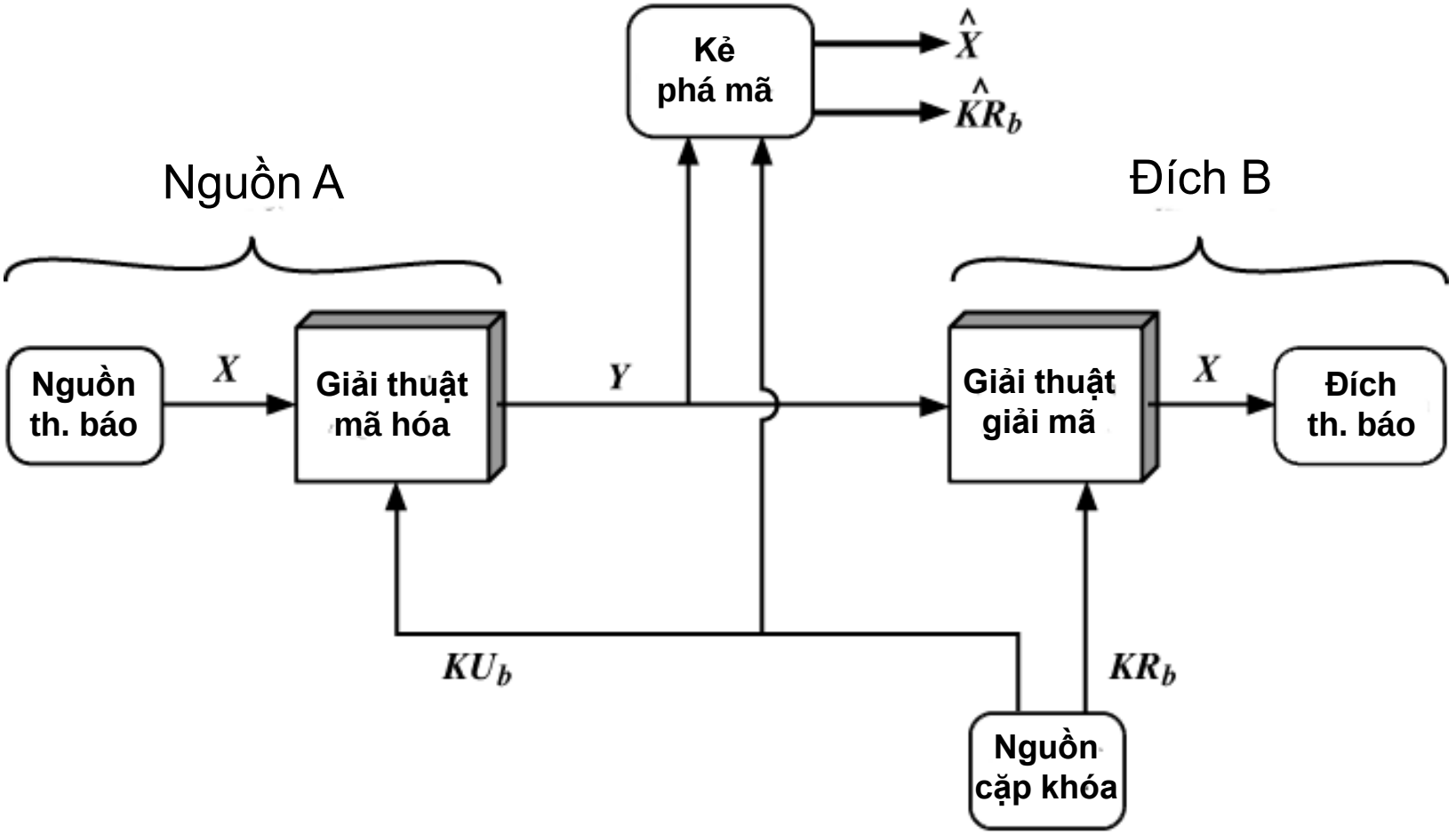
Xác thực


Các khóa công khai
Joy
Ted
Mike
Bob
Khóa riêng
của Bob
Khóa công khai
của Bob
Bản mã
truyền đi
Nguyên bản
đầu vào
Giải thuật
mã hóa
Giải thuật
giải mã
Nguyên bản
đầu ra

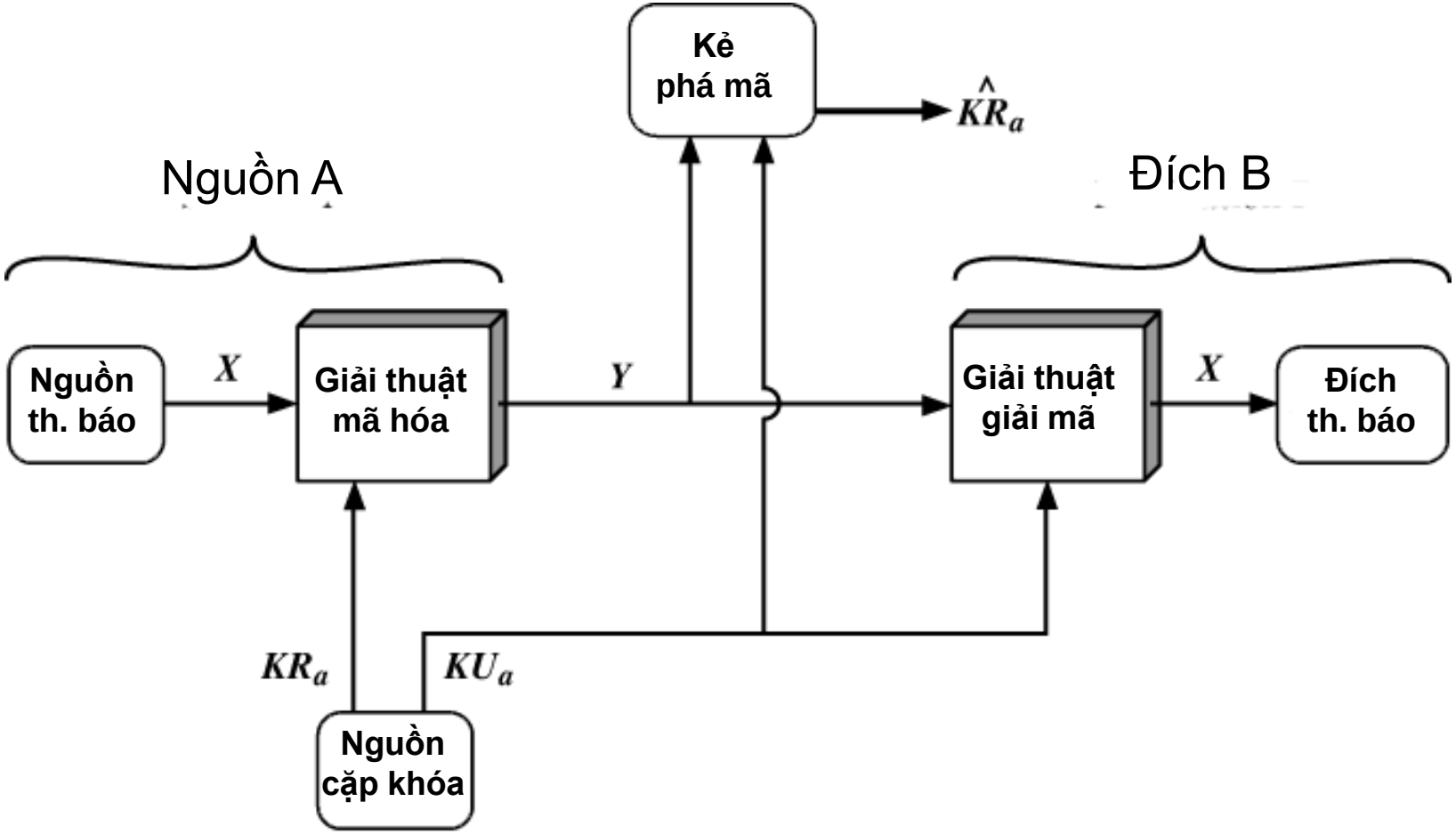
Ứng dụng mật mã khóa công khai

- Có thể phân ra 3 loại ứng dụng
  - Mã hóa/giải mã
    - Đảm bảo sự bí mật của thông tin
  - Chữ ký số
    - Hỗ trợ xác thực văn bản
  - Trao đổi khóa
    - Cho phép chia sẻ khóa phiên trong mã hóa đối xứng
- Một số giải thuật khóa công khai thích hợp cho cả 3 loại ứng dụng; một số khác chỉ có thể dùng cho 1 hay 2 loại

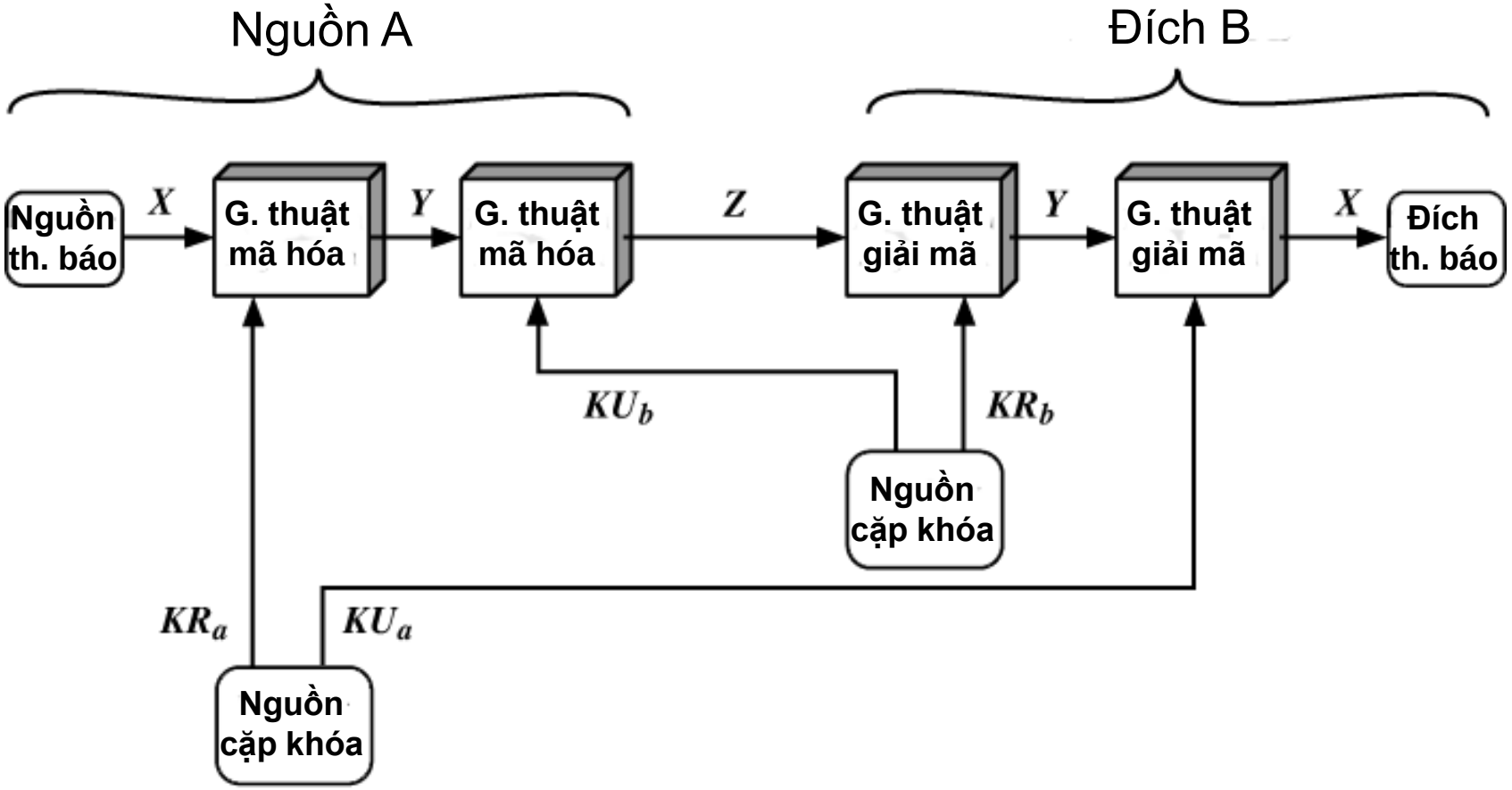
Mô hình đảm bảo bí mật



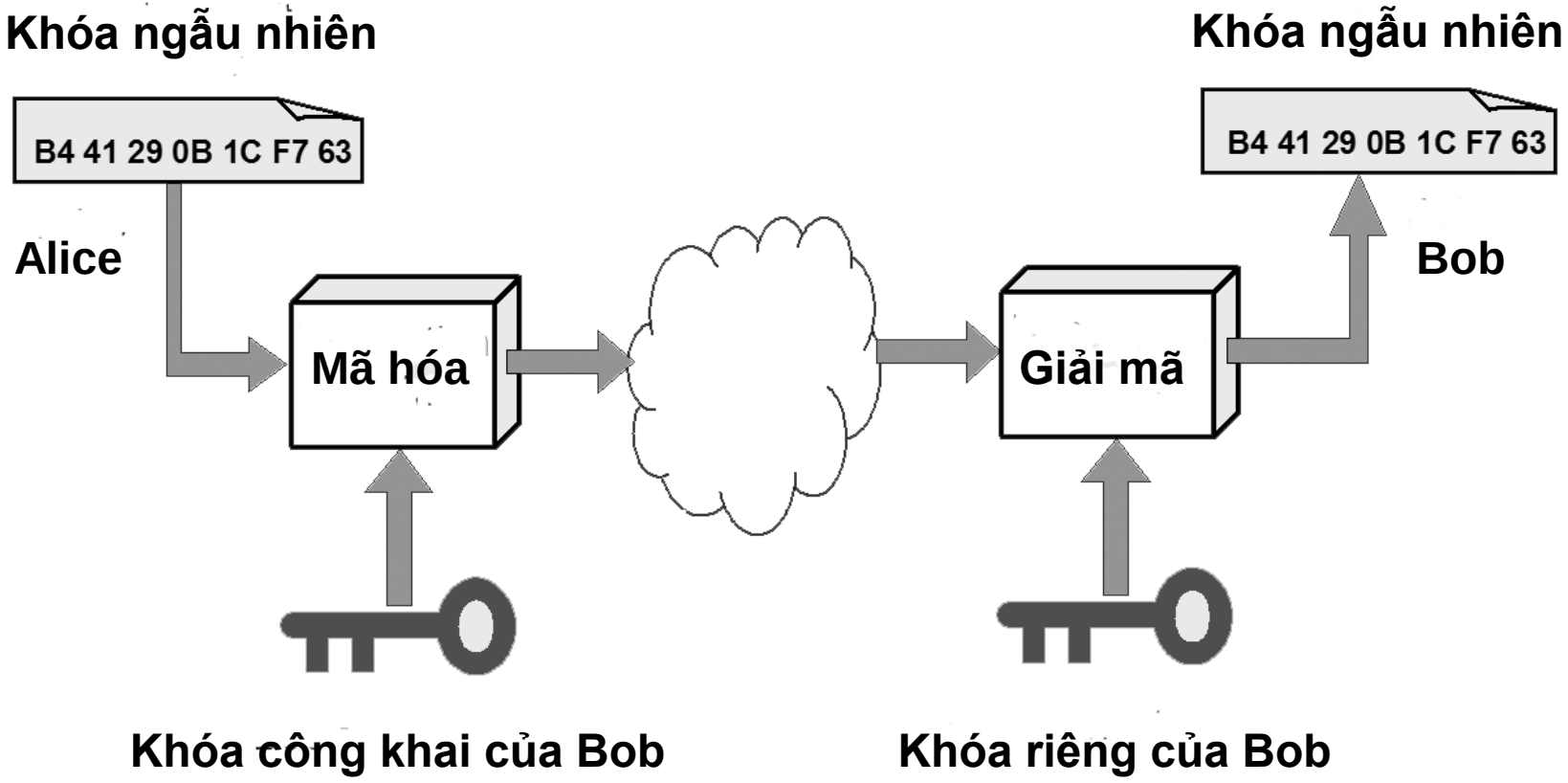
Mô hình chứng thực



Mô hình kết hợp



Trao đổi khóa



Các điều kiện cần thiết

- Bên B dễ dàng tạo ra được cặp (KU_b, KR_b)
- Bên A dễ dàng tạo ra được $C = E_{KU_b}(M)$
- Bên B dễ dàng giải mã $M = D_{KR_b}(C)$
- Đối thủ không thể xác định được KR_b khi biết KU_b
- Đối thủ không thể xác định được M khi biết KU_b và C
- Một trong hai khóa có thể dùng mã hóa trong khi khóa kia có thể dùng giải mã
  - $M = D_{KR_b}(E_{KU_b}(M)) = D_{KU_b}(E_{KR_b}(M))$
  - Không thực sự cần thiết

Hệ mã hóa RSA

- Đề xuất bởi Ron Rivest, Adi Shamir và Len Adleman (MIT) vào năm 1977
- Hệ mã hóa khóa công khai phổ dụng nhất
- Mã hóa khối với mỗi khối là một số nguyên < n
  - Thường kích cỡ n là 1024 bit ≈ 309 chữ số thập phân
- Đăng ký bản quyền năm 1983, hết hạn năm 2000
- An toàn vì chi phí phân tích thừa số của một số nguyên lớn là rất lớn

Tạo khóa RSA

- Mỗi bên tự tạo ra một cặp khóa công khai - khóa riêng theo các bước sau :
  - Chọn ngẫu nhiên 2 số nguyên tố đủ lớn $p \neq q$
  - Tính $n = pq$
  - Tính $\Phi(n) = (p-1)(q-1)$
  - Chọn ngẫu nhiên khóa mã hóa e sao cho $1 < e < \Phi(n)$ và $\gcd(e, \Phi(n)) = 1$
  - Tìm khóa giải mã $d \leq n$ thỏa mãn $e.d \equiv 1 \bmod \Phi(n)$
- Công bố khóa mã hóa công khai KU = {e, n}
- Giữ bí mật khóa giải mã riêng KR = {d, n}
  - Các giá trị bí mật p và q bị hủy bỏ

Thực hiện RSA

- Để mã hóa 1 thông báo nguyên bản M, bên gửi thực hiện
  - Lấy khóa công khai của bên nhận KU = {e, n}
  - Tính $C = M^e \bmod n$
- Để giải mã bản mã C nhận được, bên nhận thực hiện
  - Sử dụng khóa riêng KR = {d, n}
  - Tính $M = C^d \bmod n$
- Lưu ý là thông báo M phải nhỏ hơn n
  - Phân thành nhiều khối nếu cần

Vì sao RSA khả thi

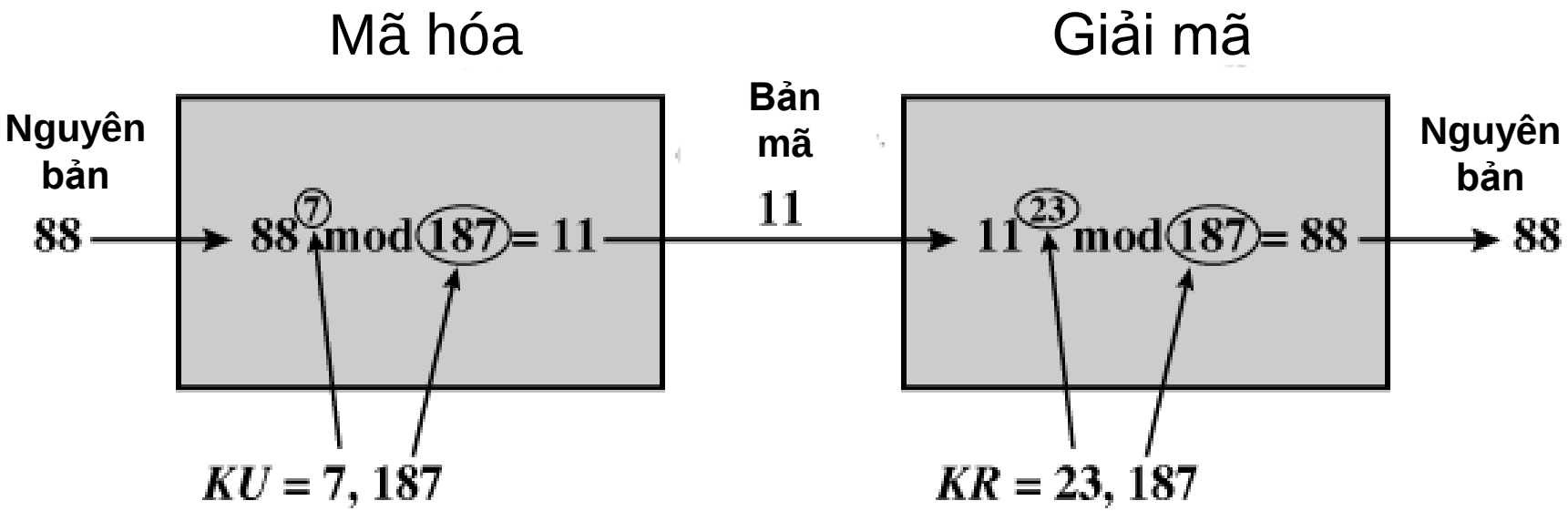
- Theo định lý Euler
  - $\forall$ a, n : gcd(a, n) = 1 $\Rightarrow$ $a^{\Phi(n)} \bmod n = 1$
  - Φ(n) là số các số nguyên dương nhỏ hơn n và nguyên tố cùng nhau với n
- Đối với RSA có
  - n = pq với p và q là các số nguyên tố
  - $\Phi(n) = (p - 1)(q - 1)$
  - $ed \equiv 1 \bmod \Phi(n) \Rightarrow \exists$ số nguyên k : $ed = k\Phi(n) + 1$
  - M < n
- Có thể suy ra
  - $C^d \bmod n = M^{ed} \bmod n = M^{k\Phi(n) + 1} \bmod n = M \bmod n = M$

Ví dụ tạo khóa RSA

- Chọn 2 số nguyên tố p = 17 và q = 11
- Tính n = pq = 17 $\times$ 11 = 187
- Tính Φ(n) = (p - 1)(q - 1) = 16 $\times$ 10 = 160
- Chọn e : gcd(e, 160) = 1 và 1 < e < 160; lấy e = 7
- Xác định d : de $\equiv$ 1 mod 160 và d $\leq$ 187

  Giá trị d = 23 vì 23 $\times$ 7 = 161 = 1 $\times$ 160 + 1
- Công bố khóa công khai KU = {7, 187}
- Giữ bí mật khóa riêng KR = {23, 187}
  - Hủy bỏ các giá trị bí mật p = 17 và q = 11

Ví dụ thực hiện RSA



Chọn tham số RSA

- Cần chọn p và q đủ lớn
- Thường chọn e nhỏ
- Thường có thể chọn cùng giá trị của e cho tất cả người dùng
- Trước đây khuyến nghị giá trị của e là 3, nhưng hiện nay được coi là quá nhỏ
- Thường chọn e = $2^{16} - 1 = 65535$
- Giá trị của d sẽ lớn và khó đoán

An toàn của RSA

- Khóa 128 bit là một số giữa 1 và một số rất lớn

  340.282.366.920.938.000.000.000.000.000.000.000.000

- Có bao nhiêu số nguyên tố giữa 1 và số này

  $\approx n / \ln(n) = 2^{128} / \ln(2^{128}) \approx$
  3.835.341.275.459.350.000.000.000.000.000.000.000

- Cần bao nhiêu thời gian nếu mỗi giây có thể tính được 10^{12} số

  Hơn 121,617,874,031,562,000 năm (khoảng 10 triệu lần tuổi của vũ trụ)

- An toàn nhưng cần đề phòng những điểm yếu

Phá mã RSA

- Phương pháp vét cạn
  - Thử tất cả các khóa riêng có thể
    - Phụ thuộc vào độ dài khóa
- Phương pháp phân tích toán học
  - Phân n thành tích 2 số nguyên tố p và q
  - Xác định trực tiếp $\Phi(n)$ không thông qua p và q
  - Xác định trực tiếp d không thông qua $\Phi(n)$
- Phương pháp phân tích thời gian
  - Dựa trên việc đo thời gian giải mã
  - Có thể ngăn ngừa bằng cách làm nhiễu

Phân tích thừa số RSA

- An toàn của RSA dựa trên độ phức tạp của việc phân tích thừa số n
- Thời gian cần thiết để phân tích thừa số một số lớn tăng theo hàm mũ với số bit của số đó
  - Mất nhiều năm khi số chữ số thập phân của n vượt quá 100 (giả sử làm 1 phép tính nhị phân mất 1 ηs)
- Kích thước khóa lớn đảm bảo an toàn cho RSA
  - Từ 1024 bit trở lên
  - Gần đây nhất năm 1999 đã phá mã được 512 bit (155 chữ số thập phân)

Hệ trao đổi khóa Diffie-Hellman

- Giải thuật mật mã khóa công khai đầu tiên
- Đề xuất bởi Whitfield Diffie và Martin Hellman vào năm 1976
  - Malcolm Williamson (GCHQ - Anh) phát hiện trước mấy năm nhưng đến năm 1997 mới công bố
- Chỉ dùng để trao đổi khóa bí mật một cách an toàn trên các kêch thông tin không an toàn
- Khóa bí mật được tính toán bởi cả hai bên
- An toàn phụ thuộc vào độ phức tạp của việc tính log rời rạc

Thiết lập Diffie-Hellman

- Các bên thống nhất với nhau các tham số chung
  - q là một số nguyên tố đủ lớn
  - α là một nguyên căn của q
    - α mod q, α^2 mod q,..., α^{q-1} mod q là các số nguyên giao hoán của các số từ 1 đến q - 1
- Bên A
  - Chọn ngẫu nhiên làm khóa riêng $X_A < q$
  - Tính khóa chung $Y_A = \alpha^{X_A} \bmod q$
- Bên B
  - Chọn ngẫu nhiên làm khóa riêng $X_B < q$
  - Tính khóa chung $Y_B = \alpha^{X_B} \bmod q$

Trao đổi khóa Diffie-Hellman

- Tính toán khóa bí mật
  - Bên A biết khóa riêng X_A và khóa công khai Y_B

$$K = Y_B^{X_A} \bmod q$$

  - Bên B biết khóa riêng X_B và khóa công khai Y_A

$$K = Y_A^{X_B} \bmod q$$

- Chứng minh

$$\begin{aligned} Y_A^{X_B} \bmod q &= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\ &= \alpha^{X_A X_B} \bmod q \\ &= \alpha^{X_B X_A} \bmod q \\ &= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\ &= Y_B^{X_A} \bmod q \end{aligned}$$

Ví dụ Diffie-Hellman

- Alice và Bob muốn trao đổi khóa bí mật
- Cùng chọn q = 353 và α = 3
- Chọn ngẫu nhiên các khóa riêng
  - Alice chọn X_A = 97, Bob chọn X_B = 233
- Tính toán các khóa công khai
  - $Y_A = 3^{97} \bmod 353 = 40$ (Alice)
  - $Y_B = 3^{233} \bmod 353 = 248$ (Bob)
- Tính toán khóa bí mật chung
  - $K = Y_B^{X_A} \bmod 353 = 248^{97} \bmod 353 = 160$ (Alice)
  - $K = Y_A^{X_B} \bmod 353 = 40^{233} \bmod 353 = 160$ (Bob)

Hạn chế của khóa công khai

- Tốc độ xử lý
  - Các giải thuật khóa công khai chủ yếu dùng các phép nhân chậm hơn nhiều so với các giải thuật đối xứng
  - Không thích hợp cho mã hóa thông thường
  - Thường dùng trao đổi khóa bí mật đầu phiên truyền tin
- Tính xác thực của khóa công khai
  - Bất cứ ai cũng có thể tạo ra một khóa công bố đó là của một người khác
  - Chừng nào việc giả mạo chưa bị phát hiện có thể đọc được nội dung các thông báo gửi cho người kia
  - Cần đảm bảo những người đăng ký khóa là đáng tin

Chương 4

XÁC THỰC & CHỮ KÝ SỐ

Vấn đề xác thực

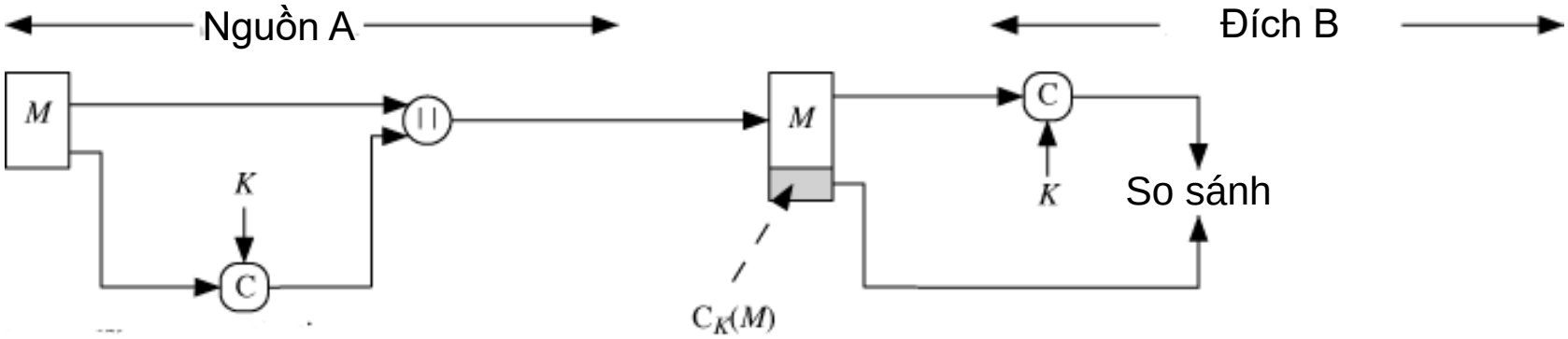
- Các tiêu chuẩn cần xác minh
  - Thông báo có nguồn gốc rõ ràng chính xác
  - Nội dung thông báo toàn vẹn không bị thay đổi
  - Thông báo được gửi đúng trình tự và thời điểm
- Mục đích để chống lại hình thức tấn công chủ động (xuyên tạc dữ liệu và giao tác)
- Các phương pháp xác thực thông báo
  - Mã hóa thông báo
  - Sử dụng mã xác thực thông báo (MAC)
  - Sử dụng hàm băm

Xác thực bằng cách mã hóa

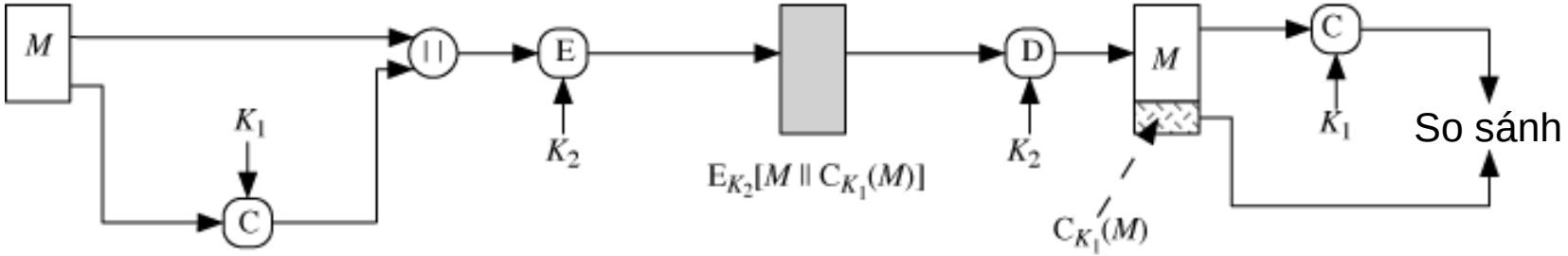
- Sử dụng mã hóa đối xứng
  - Thông báo gửi từ đúng nguồn vì chỉ có người gửi đó mới biết khóa bí mật dùng chung
  - Nội dung không thể bị thay đổi vì nguyên bản có cấu trúc nhất định
  - Các gói tin được đánh số thứ tự và mã hóa nên không thể thay đổi trình tự và thời điểm nhận được
- Sử dụng mã hóa khóa công khai
  - Không chỉ xác thực thông báo mà còn tạo chữ ký số
  - Phức tạp và mất thời gian hơn mã hóa đối xứng

Mã xác thực thông báo (MAC)

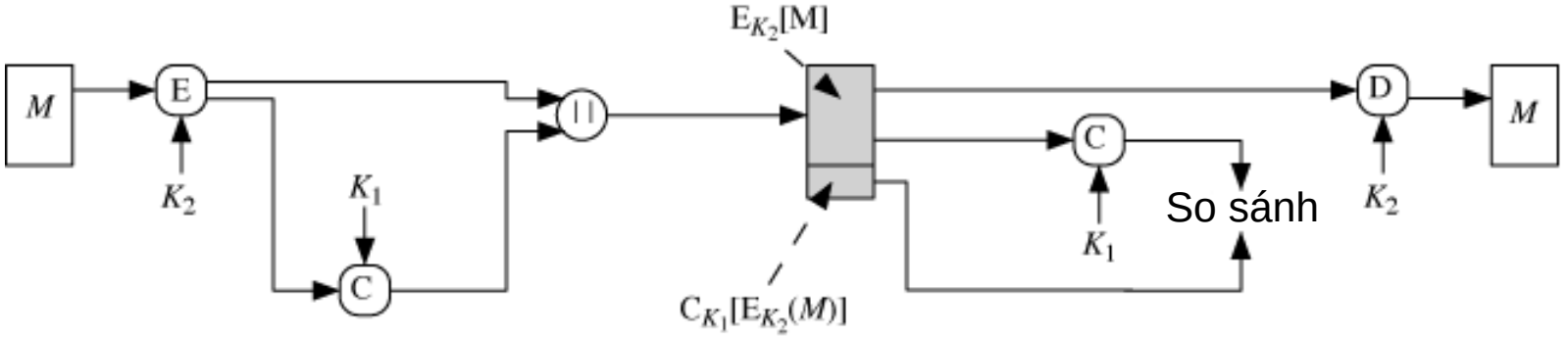
- Khối kích thước nhỏ cố định gắn vào thông báo tạo ra từ thông báo đó và khóa bí mật chung
- Bên nhận thực hiện cùng giải thuật trên thông báo và khóa để so xem MAC có chính xác không
- Giải thuật tạo MAC giống như giải thuật mã hóa nhưng không cần nghịch được
- Có thể nhiều thông báo cùng có chung MAC
  - Nhưng nếu biết một thông báo và MAC của nó, rất khó tìm ra một thông báo khác có cùng MAC
  - Các thông báo có cùng xác suất tạo ra MAC
- Đáp ứng 3 tiêu chuẩn xác thực



a) Xác thực thông báo



b) Xác thực thông báo và bảo mật; MAC gắn vào nguyên bản

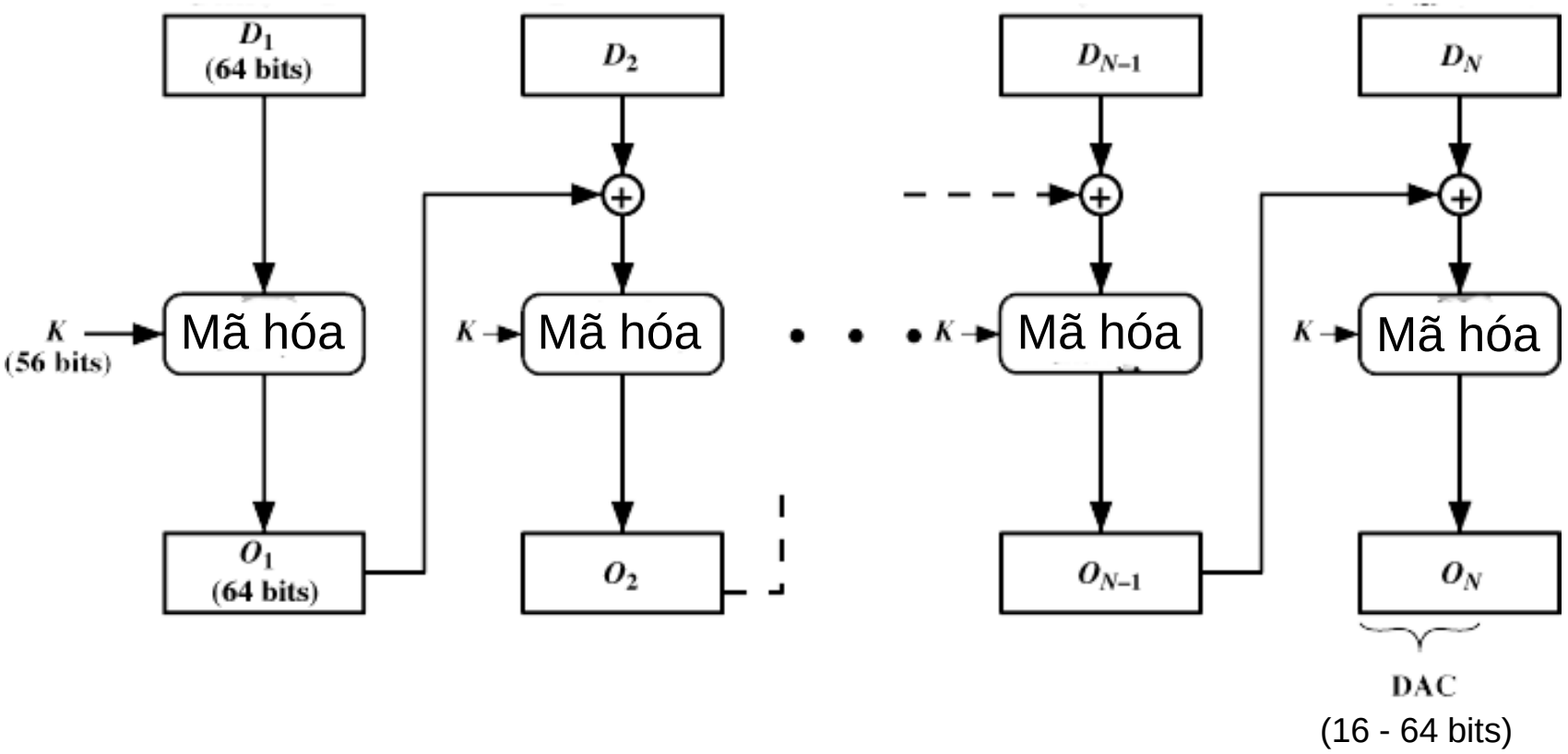


c) Xác thực thông báo và bảo mật; MAC gắn vào bản mã

Vì sao dùng MAC

- Nhiều trường hợp chỉ cần xác thực, không cần mã hóa tốn thời gian và tài nguyên
  - Thông báo hệ thống
  - Chương trình máy tính
- Tách riêng các chức năng bảo mật và xác thực sẽ khiến việc tổ chức linh hoạt hơn
  - Chẳng hạn mỗi chức năng thực hiện ở một tầng riêng
- Cần đảm bảo tính toàn vẹn của thông báo trong suốt thời gian tồn tại không chỉ khi lưu chuyển
  - Vì thông báo có thể bị thay đổi sau khi giải mã

MAC dựa trên DES (DAC)

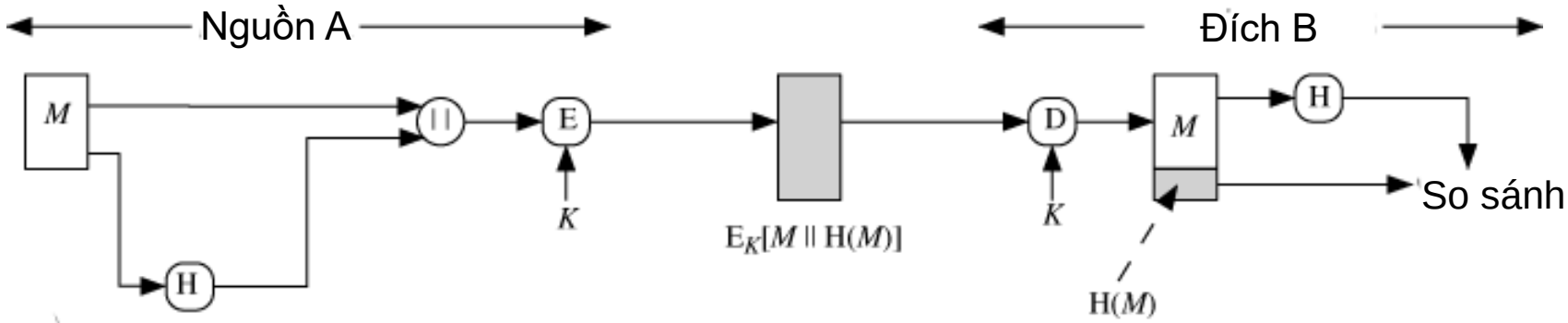


Hàm băm

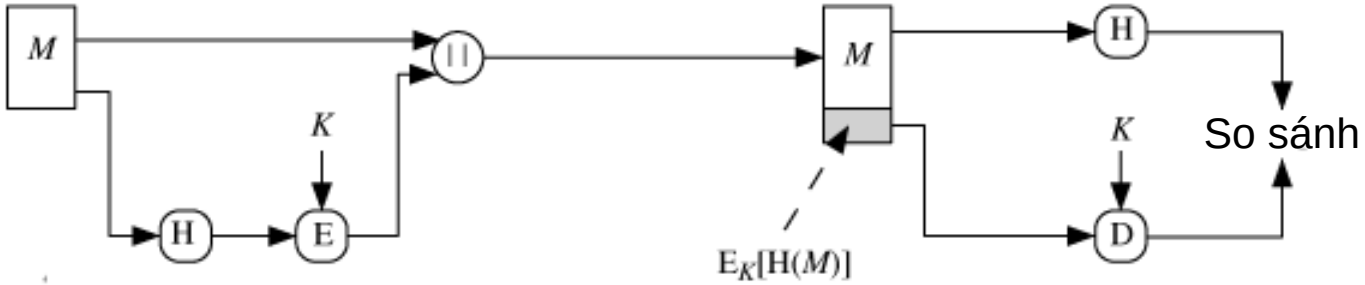
- Tạo ra một giá trị băm có kích thước cố định từ thông báo đầu vào (không dùng khóa)

$$h = H(M)$$

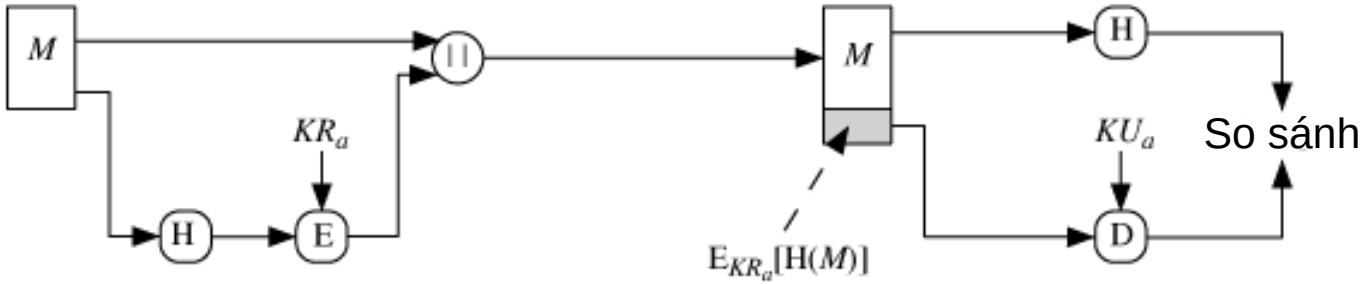
- Hàm băm không cần giữ bí mật
- Giá trị băm gắn kèm với thông báo dùng để kiểm tra tính toàn vẹn của thông báo
- Bất kỳ sự thay đổi M nào dù nhỏ cũng tạo ra một giá trị h khác



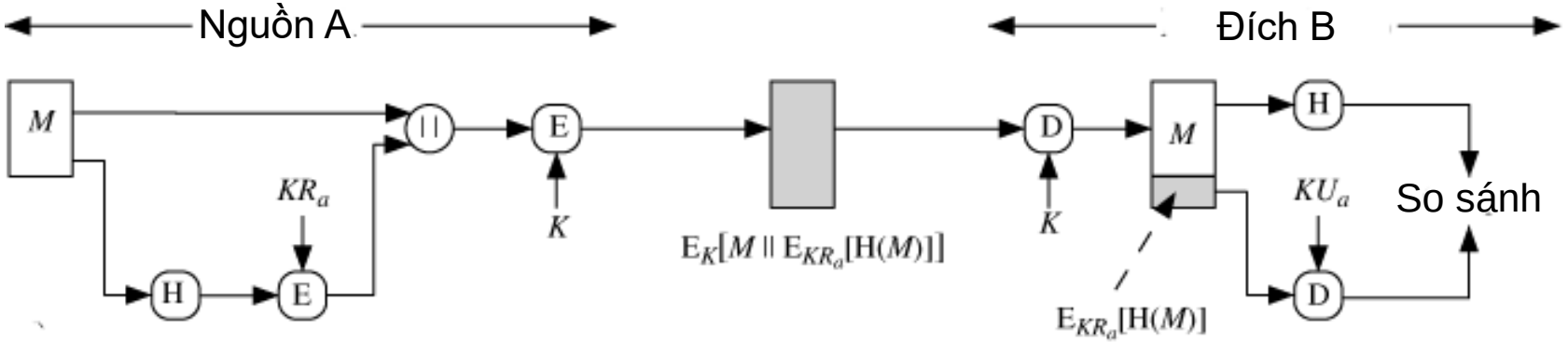
a) Xác thực thông báo và bảo mật; mã băm gắn vào nguyên bản



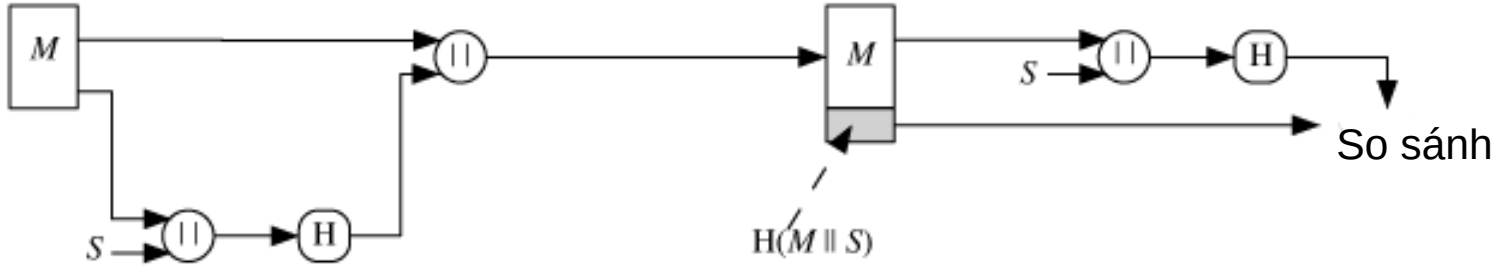
b) Xác thực thông báo; mã băm được mã hóa sử dụng phương pháp đối xứng



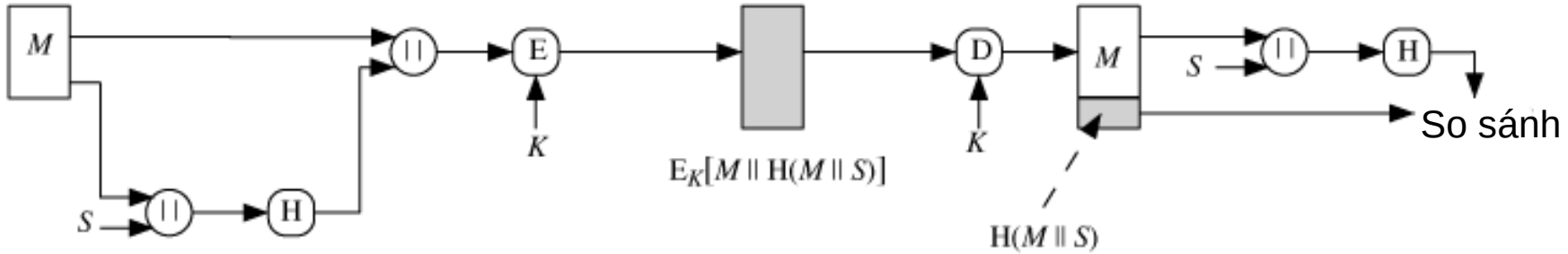
c) Xác thực thông báo; mã băm được mã hóa sử dụng phương pháp khóa công khai



d) Xác thực bằng mã hóa khóa công khai và bảo mật bằng mã hóa đối xứng



e) Xác thực không cần mã hóa nhờ hai bên chia sẻ một giá trị bí mật chung

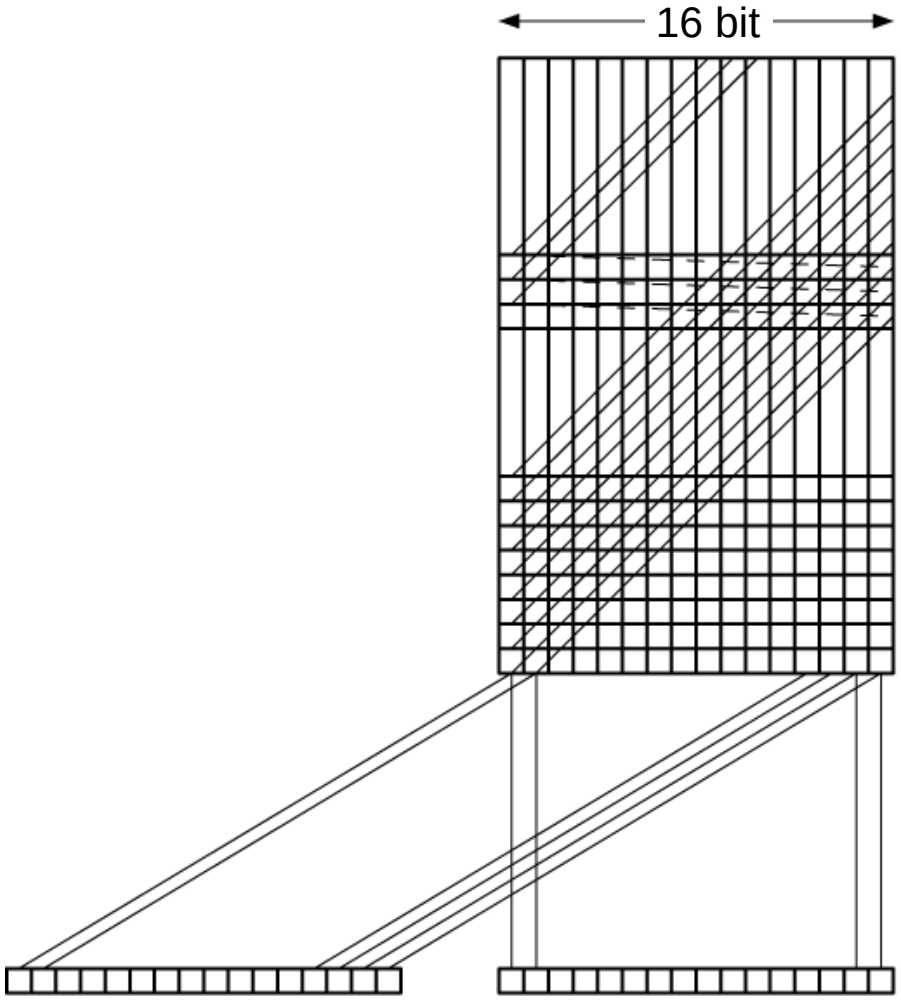


f) Xác thực nhờ một giá trị bí mật chung; bảo mật bằng phương pháp đối xứng

Yêu cầu đối với hàm băm

- Có thể áp dụng với thông báo M có độ dài bất kỳ
- Tạo ra giá trị băm h có độ dài cố định
- H(M) dễ dàng tính được với bất kỳ M nào
- Từ h rất khó tìm được M sao cho H(M) = h
  - Tính một chiều
- Từ M_1 rất khó tìm được M_2 sao cho $H(M_2) = H(M_1)$
  - Tính chống xung đột yếu
- Rất khó tìm được (M_1, M_2) sao cho $H(M_1) = H(M_2)$
  - Tính chống xung đột mạnh

Các hàm băm đơn giản



Kiểu tấn công ngày sinh

- Nghịch lý ngày sinh
  - Trong 23 người, xác suất tìm ra 1 người khác có cùng ngày sinh với A là ≈ 6%
  - Xác suất 2 trong 23 người có cùng ngày sinh là ≈ 50%
- Cách thức tấn công mã băm m bit
  - Tạo ra $2^{m/2}$ biến thể đồng nghĩa của thông báo hợp lệ
  - Tạo ra $2^{m/2}$ biến thể của thông báo giả mạo
  - So sánh 2 tập thông báo với nhau tìm ra 1 cặp có cùng mã băm (xác suất > 0,5 theo nghịch lý ngày sinh)
  - Để người gửi ký biến thể hợp lệ, rồi dùng chữ ký gắn vào biến thể giả mạo

An toàn hàm băm và MAC

- Kiểu tấn công vét cạn
  - Với hàm băm, nỗ lực phụ thuộc độ dài m của mã băm
    - Độ phức tạp của tính một chiều và tính chống xung đột yếu là 2^m; của tính chống xung đột mạnh là $2^{m/2}$
    - 128 bit có thể phá được, thường dùng 160 bit
  - Với MAC, nỗ lực phụ thuộc vào độ dài k của khóa và độ dài n của MAC
    - Độ phức tạp là $\min(2^k, 2^n)$
    - Ít nhất phải là 128 bit
- Kiểu thám mã
  - Hàm băm thường gồm nhiều vòng như mã hóa khối nên có thể tập trung khai thác điểm yếu hàm vòng

Chữ ký số

- Xác thực thông báo không có tác dụng khi bên gửi và bên nhận muốn gây hại cho nhau
  - Bên nhận giả mạo thông báo của bên gửi
  - Bên gửi chối là đã gửi thông báo đến bên nhận
- Chữ ký số không những giúp xác thực thông báo mà còn bảo vệ mỗi bên khỏi bên kia
- Chức năng chữ ký số
  - Xác minh tác giả và thời điểm ký thông báo
  - Xác thực nội dung thông báo
  - Là căn cứ để giải quyết tranh chấp

Yêu cầu đối với chữ ký số

- Phụ thuộc vào thông báo được ký
- Có sử dụng thông tin riêng của người gửi
  - Để tránh giả mạo và chối bỏ
- Tương đối dễ tạo ra
- Tương đối dễ nhận biết và kiểm tra
- Rất khó giả mạo
  - Bằng cách tạo thông báo khác có cùng chữ ký số
  - Bằng cách tạo chữ ký số theo ý muốn cho thông báo
- Thuận tiện trong việc lưu trữ

Chữ ký số trực tiếp

- Chỉ liên quan đến bên gửi và bên nhận
- Với mật mã khóa công khai
  - Dùng khóa riêng ký toàn bộ thông báo hoặc giá trị băm
  - Có thể mã hóa sử dụng khóa công khai của bên nhận
  - Quan trọng là ký trước mã hóa sau
- Chỉ có tác dụng khi khóa riêng của bên gửi được đảm bảo an toàn
  - Bên gửi có thể giả vờ mất khóa riêng
    - Cần bổ xung thông tin thời gian và báo mất khóa kịp thời
  - Khóa riêng có thể bị mất thật
    - Kẻ cắp có thể gửi thông báo với thông tin thời gian sai lệch

Chữ ký số gián tiếp

- Có sự tham gia của một bên trọng tài
  - Nhận thông báo có chữ ký số từ bên gửi, kiểm tra tính hợp lệ của nó
  - Bổ xung thông tin thời gian và gửi đến bên nhận
- An toàn phụ thuộc chủ yếu vào bên trọng tài
  - Cần được bên gửi và bên nhận tin tưởng
- Có thể cài đặt với mã hóa đối xứng hoặc mã hóa khóa công khai
- Bên trọng tài có thể được phép nhìn thấy hoặc không nội dung thông báo

Các kỹ thuật chữ ký số gián tiếp

(a) Mã hóa đối xứng, trọng tài thấy thông báo

(1) $X \rightarrow A : M \| E_{K_{XA}}[ID_X \| H(M)]$

(2) $A \rightarrow Y : E_{K_{AY}}[ID_X \| M \| E_{K_{XA}}[ID_X \| H(M)] \| T]$

(b) Mã hóa đối xứng, trọng tài không thấy thông báo

(1) $X \rightarrow A : ID_X \| E_{K_{XY}}[M] \| E_{K_{XA}}[ID_X \| H(E_{K_{XY}}[M])]$

(2) $A \rightarrow Y : E_{K_{AY}}[ID_X \| E_{K_{XY}}[M] \| E_{K_{XA}}[ID_X \| H(E_{K_{XY}}[M])] \| T]$

(c) Mã hóa khóa công khai, trọng tài không thấy thông báo

(1) $X \rightarrow A : ID_X \| E_{KR_X}[ID_X \| E_{KU_Y}[E_{KR_X}[M]]]$

(2) $A \rightarrow Y : E_{KR_A}[ID_X \| E_{KU_Y}[E_{KR_X}[M]] \| T]$

Ký hiệu : X = Bên gửi M = Thông báo

Y = Bên nhận T = Nhãn thời gian

A = Trọng tài

Chương 5

CÁC ỨNG DỤNG XÁC THỰC

Giới thiệu

- Mục đích của các ứng dụng xác thực là hỗ trợ xác thực và chữ ký số ở mức ứng dụng
- Phân làm 2 loại chính
  - Dựa trên mã hóa đối xứng
    - Dịch vụ Kerberos
    - Giao thức Needham-Schroeder
  - Dựa trên khóa công khai được chứng thực
    - Dịch vụ X.509
    - Hệ thống PGP

Kerberos

- Hệ thống dịch vụ xác thực phát triển bởi MIT
- Nhằm đối phó với các hiểm họa sau
  - Người dùng giả danh là người khác
  - Người dùng thay đổi địa chỉ mạng của client
  - Người dùng xem trộm thông tin trao đổi và thực hiện kiểu tấn công lặp lại
- Bao gồm 1 server tập trung có chức năng xác thực người dùng và các server dịch vụ phân tán
  - Tin cậy server tập trung thay vì các client
  - Giải phóng chức năng xác thực khỏi các server dịch vụ và các client

Ký hiệu

- C : Client
- AS : Server xác thực
- V : Server dịch vụ
- ID_C : Danh tính người dùng trên C
- ID_V : Danh tính của V
- P_C : Mật khẩu của người dùng trên C
- AD_C : Địa chỉ mạng của C
- K_V : Khóa bí mật chia sẻ bởi AS và V
- ‖ : Phép ghép
- TGS : Server cấp thẻ
- TS : Nhãn thời gian

Một hội thoại xác thực đơn giản

- Giao thức

  (1) $C \rightarrow AS : ID_C \| P_C \| ID_V$

  (2) $AS \rightarrow C :$ Thẻ

  (3) $C \rightarrow V : ID_C \| \text{Thẻ}$

  $\text{Thẻ} = E_{K_V}[ID_C \| AD_C \| ID_V]$

- Hạn chế
  - Mật khẩu truyền từ C đến AS không được bảo mật
  - Nếu thẻ chỉ sử dụng được một lần thì phải cấp thẻ mới cho mỗi lần truy nhập cùng một dịch vụ
  - Nếu thẻ sử dụng được nhiều lần thì có thể bị lấy cắp để sử dụng trước khi hết hạn
  - Cần thẻ mới cho mỗi dịch vụ khác nhau

Hội thoại xác thực Kerberos 4

(a) Trao đổi với dịch vụ xác thực : để có thẻ cấp thẻ

(1) $C \rightarrow AS : \ ID_C \parallel ID_{tgs} \parallel TS_1$

(2) $AS \rightarrow C : \ E_{K_C}[K_{C,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Hạn_2 \parallel Thẻ_{tgs}]$

$Thẻ_{tgs} = E_{K_{tgs}}[K_{C,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Hạn_2]$

(b) Trao đổi với dịch vụ cấp thẻ : để có thẻ dịch vụ

(3) $C \rightarrow TGS : \ ID_V \parallel Thẻ_{tgs} \parallel Dấu_C$

(4) $TGS \rightarrow C : \ E_{K_{C,tgs}}[K_{C,V} \parallel ID_V \parallel TS_4 \parallel Thẻ_V]$

$Thẻ_V = E_{K_V}[K_{C,V} \parallel ID_C \parallel AD_C \parallel ID_V \parallel TS_4 \parallel Hạn_4]$

$Dấu_C = E_{K_{C,tgs}}[ID_C \parallel AD_C \parallel TS_3]$

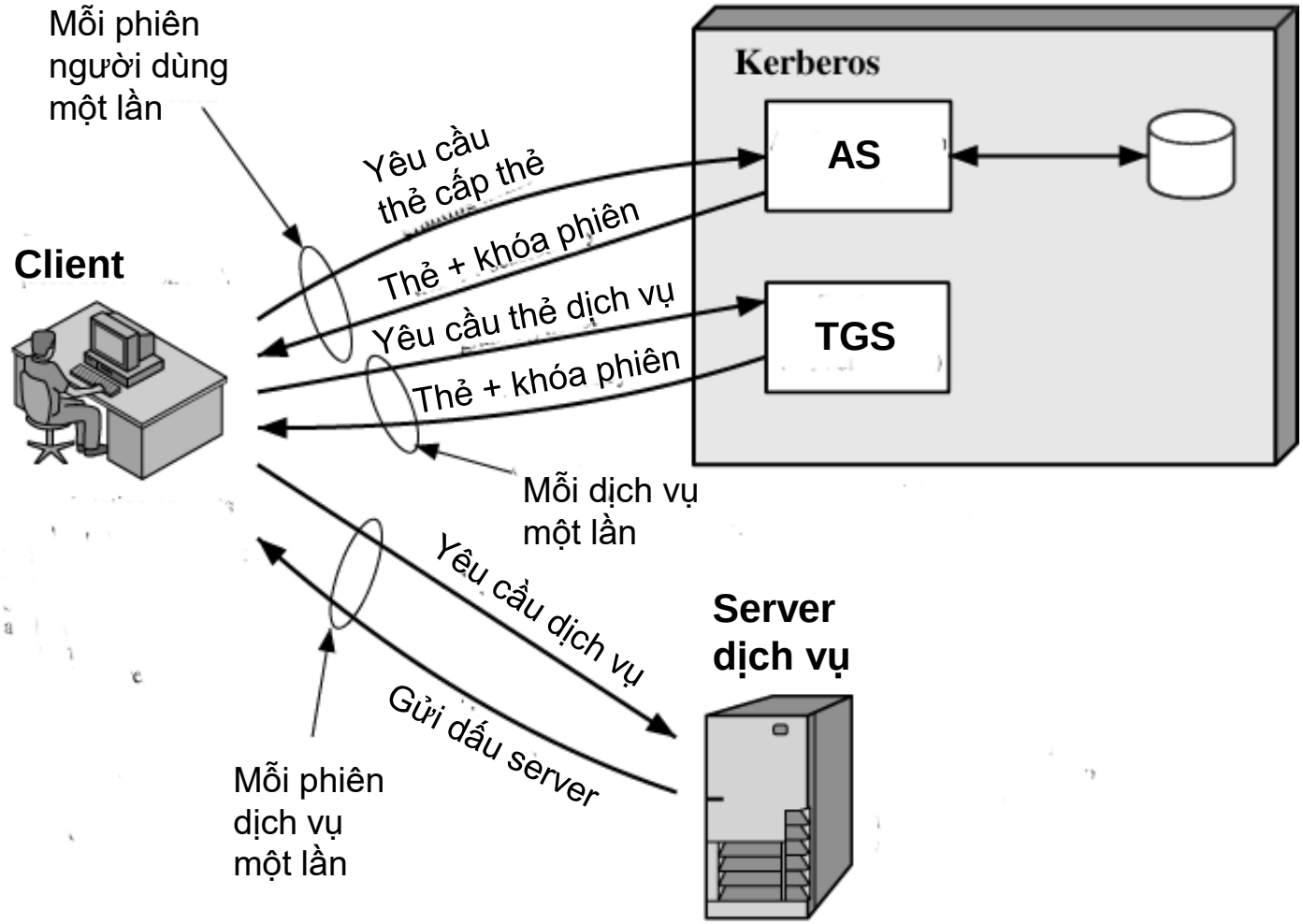
(c) Trao đổi xác thực client/server : để có dịch vụ

(5) $C \rightarrow V : \ Thẻ_V \parallel Dấu_C$

(6) $V \rightarrow C : \ E_{K_{C,V}}[TS_5 + 1]$

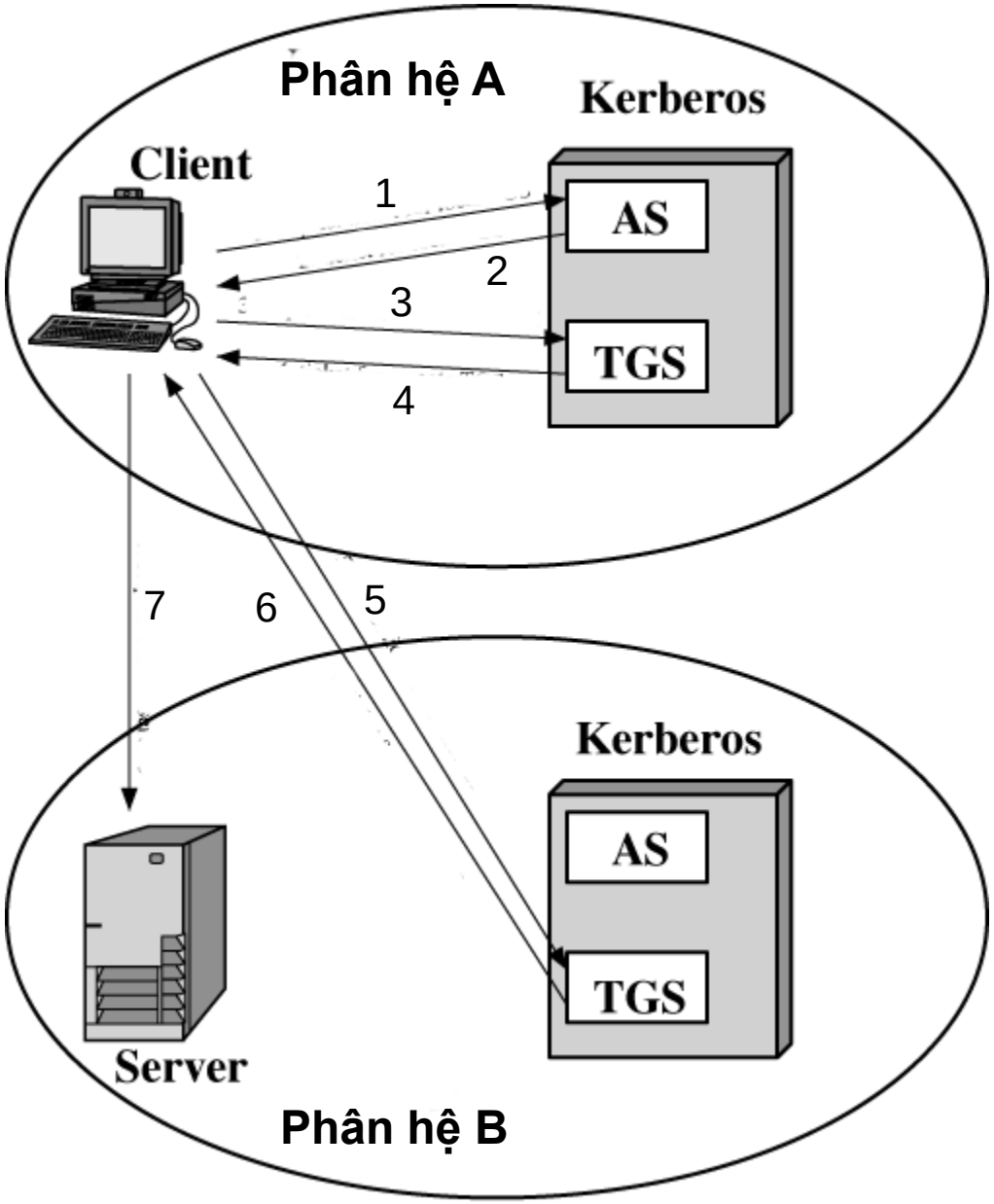
$Dấu_C = E_{K_{C,V}}[ID_C \parallel AD_C \parallel TS_5]$

Mô hình tổng quan Kerberos



Phân hệ Kerberos

- Một phân hệ Kerberos bao gồm
  - Một server Kerberos chứa trong CSDL danh tính và mật khẩu băm của các thành viên
  - Một số người dùng đăng ký làm thành viên
  - Một số server dịch vụ, mỗi server có một khóa bí mật riêng chỉ chia sẻ với server Kerberos
- Mỗi phân hệ Kerberos thường tương ứng với một phạm vi hành chính
- Hai phân hệ có thể tương tác với nhau nếu 2 server chia sẻ 1 khóa bí mật và đăng ký với nhau
  - Điều kiện là phải tin tưởng lẫn nhau



1. Yêu cầu thẻ cho TGS cục bộ
2. Thẻ cho TGS cục bộ
3. Yêu cầu thẻ cho TGS ở xa
4. Thẻ cho TGS ở xa
5. Yêu cầu thẻ cho server ở xa
6. Thẻ cho server ở xa
7. Yêu cầu dịch vụ ở xa

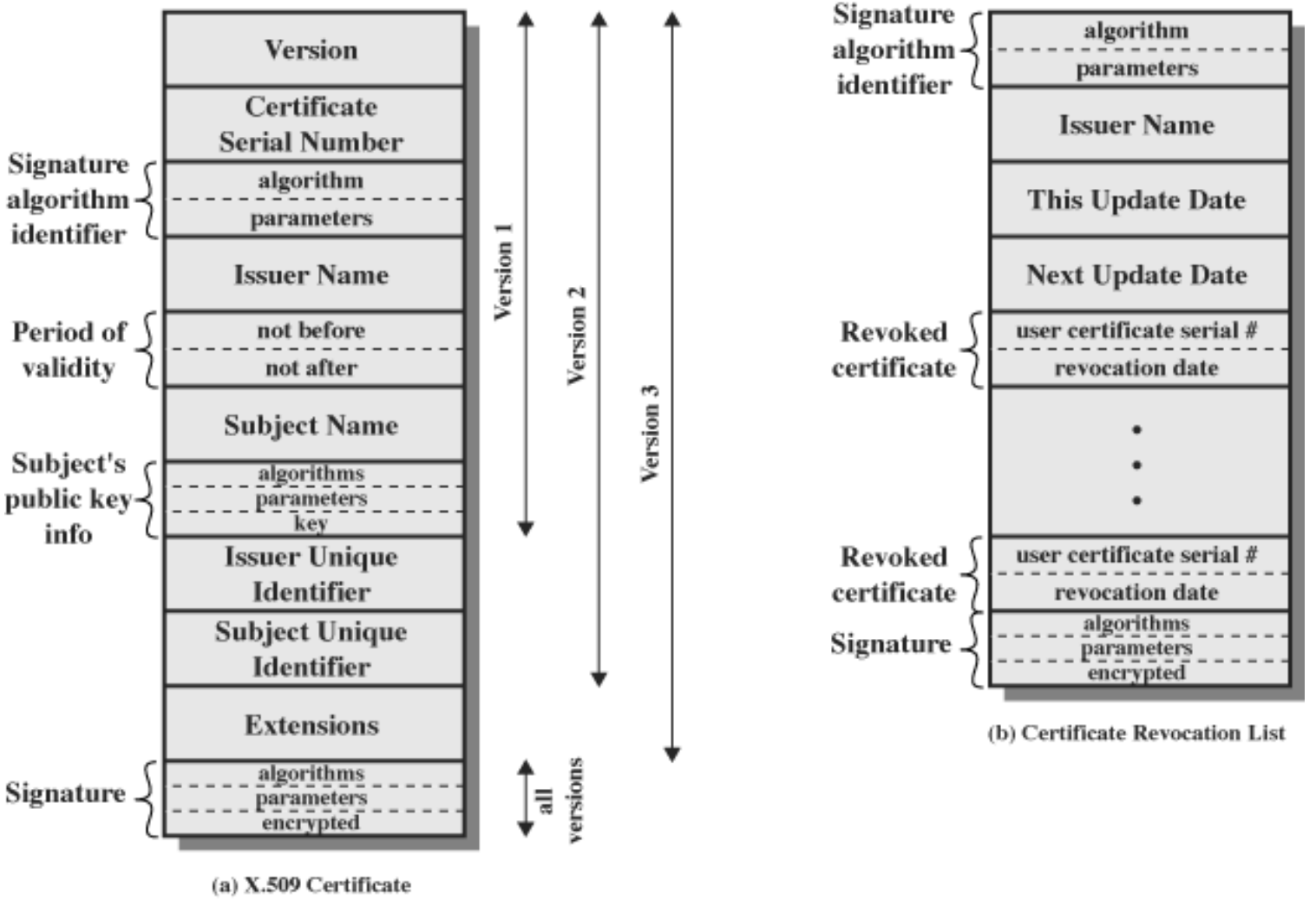
Kerberos 5

- Phát triển vào giữa những năm 1990 (sau Kerberos 4 vài năm) đặc tả trong RFC 1510
- Có một số cải tiến so với phiên bản 4
  - Khắc phục những khiếm khuyết của môi trường
    - Phụ thuộc giải thuật mã hóa, phụ thuộc giao thức mạng, trật tự byte thông báo không theo chuẩn, giá trị hạn dùng thẻ có thể quá nhỏ, không cho phép ủy nhiệm truy nhập, tương tác đa phân hệ dựa trên quá nhiều quan hệ tay đôi
  - Khắc phục những thiếu sót kỹ thuật
    - Mã hóa hai lần có một lần thừa, phương thức mã hóa PCBC để đảm bảo tính toàn vẹn không chuẩn dễ bị tấn công, khóa phiên sử dụng nhiều lần có thể bị khai thác để tấn công lặp lại, có thể bị tấn công mật khẩu

Dịch vụ xác thực X.509

- Nằm trong loạt khuyến nghị X.500 của ITU-T nhằm chuẩn hóa dịch vụ thư mục
  - Servers phân tán lưu giữ CSDL thông tin người dùng
- Định ra một cơ cấu cho dịch vụ xác thực
  - Danh bạ chứa các chứng thực khóa công khai
  - Mỗi chứng thực bao gồm khóa công khai của người dùng ký bởi một bên chuyên trách chứng thực đáng tin
- Định ra các giao thức xác thực
- Sử dụng mật mã khóa công khai và chữ ký số
  - Không chuẩn hóa giải thuật nhưng khuyến nghị RSA

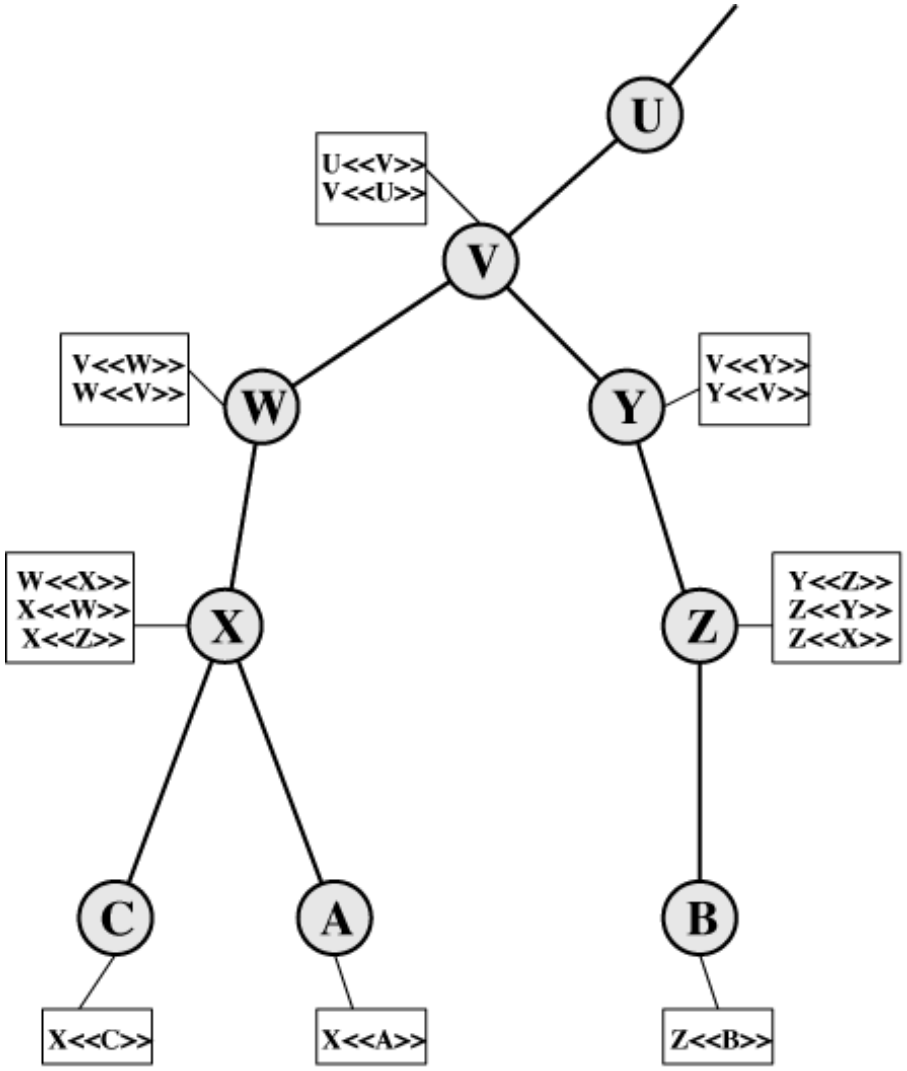
Khuôn dạng X.509



Nhận chứng thực

- Cứ có khóa công khai của CA (cơ quan chứng thực) là có thể xác minh được chứng thực
- Chỉ CA mới có thể thay đổi chứng thực
  - Chứng thực có thể đặt trong một thư mục công khai
- Cấu trúc phân cấp CA
  - Người dùng được chứng thực bởi CA đã đăng ký
  - Mỗi CA có hai loại chứng thực
    - Chứng thực thuận : Chứng thực CA hiện tại bởi CA cấp trên
    - Chứng thực nghịch : Chứng thực CA cấp trên bởi CA hiện tại
- Cấu trúc phân cấp CA cho phép người dùng xác minh chứng thực bởi bất kỳ CA nào

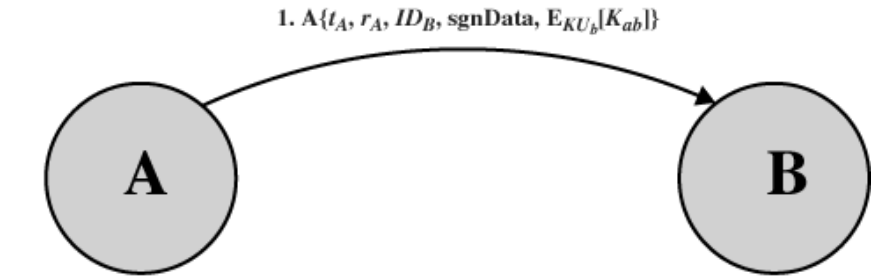
Phân cấp X.509



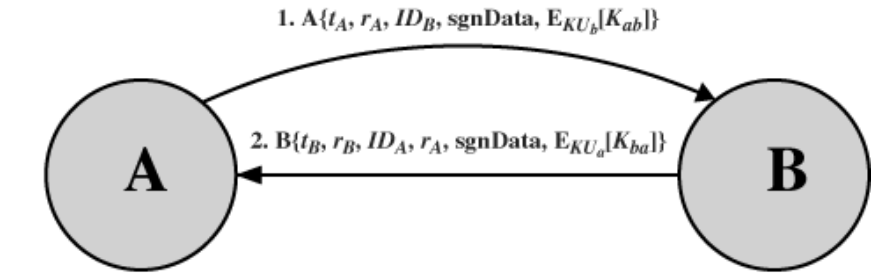
Thu hồi chứng thực

- Mỗi chứng thực có một thời hạn hợp lệ
- Có thể cần thu hồi chứng thực trước khi hết hạn
  - Khóa riêng của người dùng bị tiết lộ
  - Người dùng không còn được CA chứng thực
  - Chứng thực của CA bị xâm phạm
- Mỗi CA phải duy trì danh sách các chứng thực bị thu hồi (CRL)
- Khi nhận được chứng thực, người dùng phải kiểm tra xem nó có trong CRL không

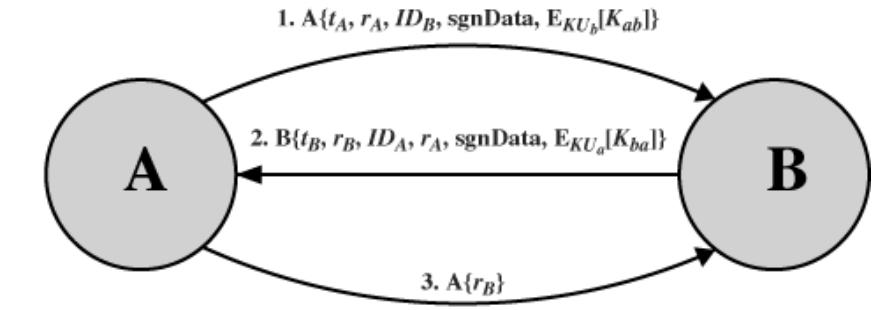
Các thủ tục xác thực



(a) One-way authentication



(b) Two-way authentication



(c) Three-way authentication

Chương 6

AN TOÀN THƯ ĐIỆN TỬ

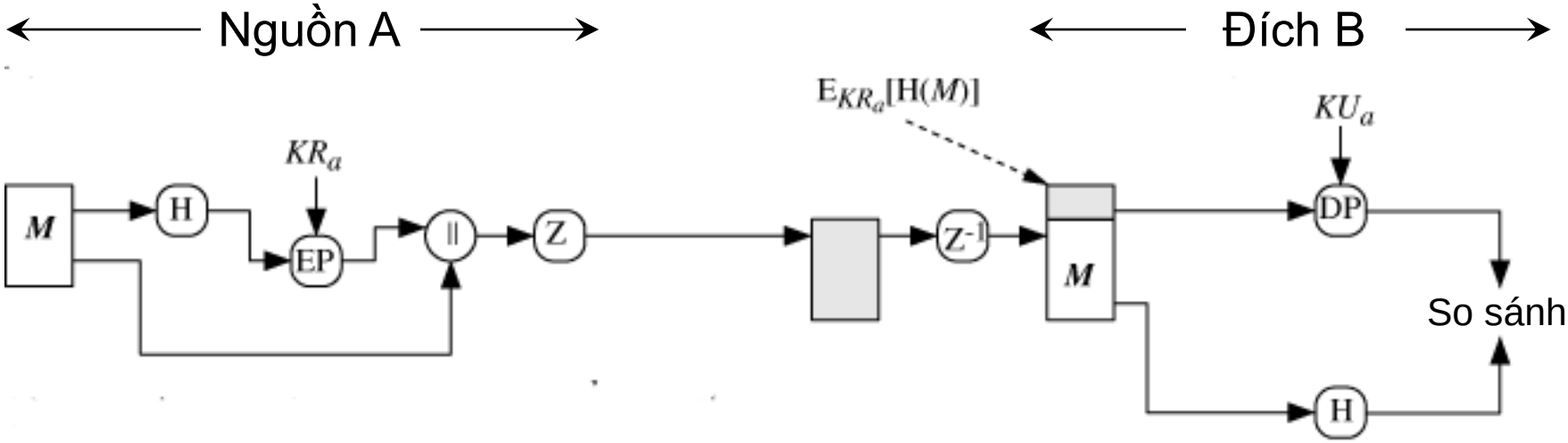
Giới thiệu

- Thư điện tử là dịch vụ mạng phổ dụng nhất
- Hiện nay các thông báo không được bảo mật
  - Có thể đọc được nội dung trong quá trình thông báo di chuyển trên mạng
  - Những người dùng có đủ quyền có thể đọc được nội dung thông báo trên máy đích
  - Thông báo dễ dàng bị giả mạo bởi một người khác
  - Tính toàn vẹn của thông báo không được đảm bảo
- Các giải pháp xác thực và bảo mật thường dùng
  - PGP (Pretty Good Privacy)
  - S/MIME (Secure/Multipurpose Internet Mail Extensions)

PGP

- Do Phil Zimmermann phát triển vào năm 1991
- Chương trình miễn phí, chạy trên nhiều môi trường khác nhau (phần cứng, hệ điều hành)
  - Có phiên bản thương mại nếu cần hỗ trợ kỹ thuật
- Dựa trên các giải thuật mật mã an toàn nhất
- Chủ yếu ứng dụng cho thư điện tử và file
- Độc lập với các tổ chức chính phủ
- Bao gồm 5 dịch vụ : xác thực, bảo mật, nén, tương thích thư điện tử, phân và ghép
  - Ba dịch vụ sau trong suốt đối với người dùng

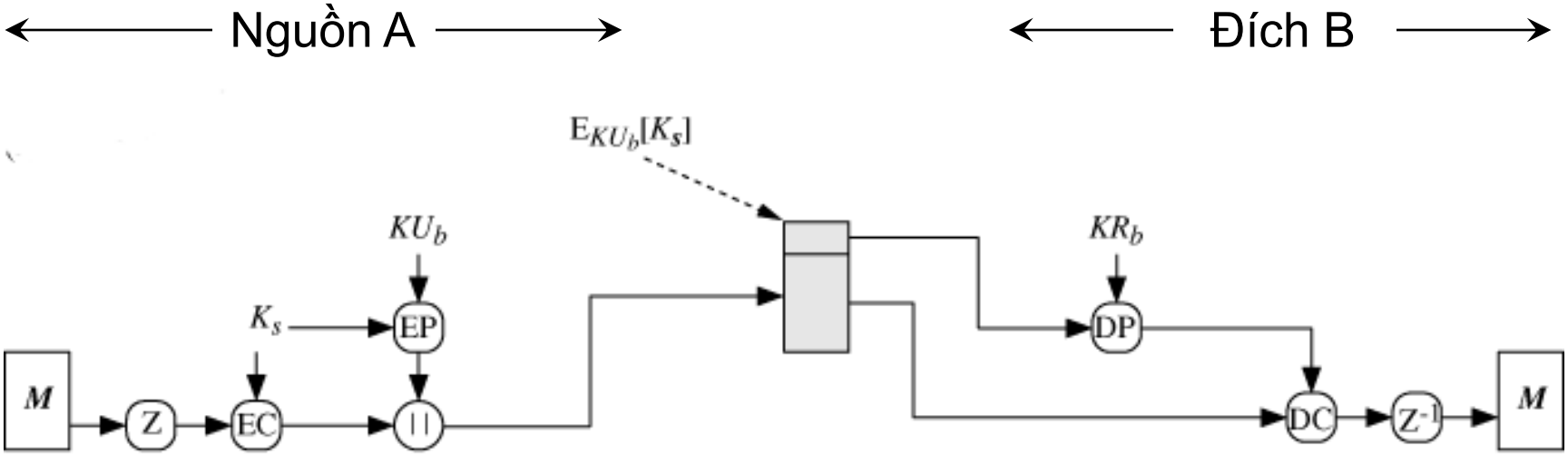
Xác thực của PGP



M = Thông báo gốc
H = Hàm băm
‖ = Ghép
Z = Nén
Z^{-1} = Cởi nén

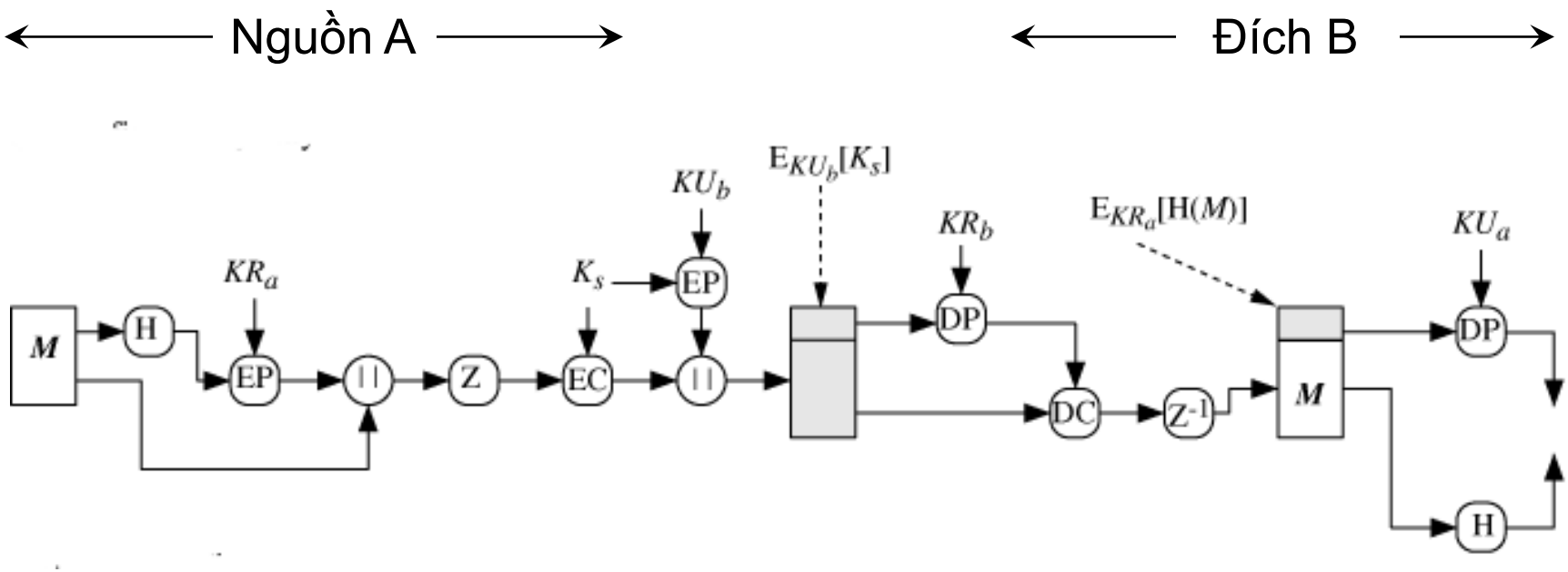
EP = Mã hóa khóa công khai
DP = Giải mã khóa công khai
KR_a = Khóa riêng của A
KU_a = Khóa công khai của A

Bảo mật của PGP



EC = Mã hóa đối xứng
DC = Giải mã đối xứng
K_s = Khóa phiên

Xác thực và bảo mật của PGP



Nén của PGP

- PGP nén thông báo sử dụng giải thuật ZIP
- Ký trước khi nén
  - Thuận tiện lưu trữ và kiểm tra, nếu ký sau khi nén thì
    - Cần lưu phiên bản nén với chữ ký, hoặc
    - Cần nén lại thông báo mỗi lần muốn kiểm tra
  - Giải thuật nén không cho kết quả duy nhất
    - Mỗi phiên bản cài đặt có tốc độ và tỷ lệ nén khác nhau
    - Nếu ký sau khi nén thì các chương trình PGP cần sử dụng cùng một phiên bản của giải thuật nén
- Mã hóa sau khi nén
  - Ít dữ liệu sẽ khiến việc mã hóa nhanh hơn
  - Thông báo nén khó phá mã hơn thông báo thô

Tương thích thư điện tử của PGP

- PGP bao giờ cũng phải gửi dữ liệu nhị phân
- Nhiều hệ thống thư điện tử chỉ chấp nhận văn bản ASCII (các ký tự đọc được)
  - Thư điện tử vốn chỉ chứa văn bản đọc được
- PGP dùng giải thuật cơ số 64 chuyển đổi dữ liệu nhị phân sang các ký tự ASCII đọc được
  - Mỗi 3 byte nhị phân chuyển thành 4 ký tự đọc được
- Hiệu ứng phụ của việc chuyển đổi là kích thước thông báo tăng lên 33%
  - Nhưng có thao tác nén bù lại

Bảng chuyển đổi cơ số 64

6-bit value	character encoding	6-bit value	character encoding	6-bit value	character encoding	6-bit value	character encoding
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						(pad)	=

Phân và ghép của PGP

- Các giao thức thư điện tử thường hạn chế độ dài tối đa của thông báo
  - Ví dụ thường là 50 KB
- PGP phân thông báo quá lớn thành nhiều thông báo đủ nhỏ
- Việc phân đoạn thông báo thực hiện sau tất cả các công đoạn khác
- Bên nhận sẽ ghép các thông báo nhỏ trước khi thực hiện các công đoạn khác

Danh tính khóa PGP

- Với một thông báo nhất định cần xác định sử dụng khóa nào trong nhiều khóa công khai / khóa riêng
  - Có thể gửi khóa công khai cùng với thông báo nhưng lãng phí đường truyền không cần thiết
- Gán cho mỗi khóa một danh tính riêng
  - Gồm 64 bit bên phải của khóa
  - Xác suất cao là mỗi khóa có một danh tính duy nhất
- Sử dụng danh tính khóa trong chữ ký

Quản lý khóa PGP

- Thay vì dựa trên các CA (cơ quan chứng thực), đối vởi PGP mỗi người dùng là một CA
  - Có thể chứng thực cho những người dùng quen biết
- Tạo nên một mạng lưới tin cậy
  - Tin các khóa đã được chứng thực
- Mỗi khóa có một chỉ số tin cậy
- Người dùng có thể thu hồi khóa của bản thân

S/MIME

- Nâng cấp từ chuẩn khuôn dạng thư điện tử MIME có thêm tính năng an toàn thông tin
- MIME khắc phục những hạn chế của SMTP (Simple Mail Transfer Protocol)
  - Không truyền được file nhị phân (chương trình, ảnh,...)
  - Chỉ gửi được các ký tự ASCII 7 bit
  - Không nhận thông báo vượt quá kích thước cho phép
  - ...
- S/MIME có xu hướng trở thành chuẩn công nghiệp sử dụng trong thương mại và hành chính
  - PGP dùng cho cá nhân

Các chức năng của S/MIME

- Bao bọc dữ liệu
  - Mã hóa nội dung thông báo và các khóa liên quan
- Ký dữ liệu
  - Chữ ký số tạo thành nhờ mã hóa thông tin tổng hợp thông báo sử dụng khóa riêng của người ký
  - Thông báo và chữ ký số được chuyển đổi cơ số 64
- Ký và để nguyên dữ liệu
  - Chỉ chữ ký số được chuyển đổi cơ số 64
- Ký và bao bọc dữ liệu
  - Kết hợp ký và bao bọc dữ liệu

Xử lý chứng thực S/MIME

- S/MIME sử dụng các chứng thực khóa công khai theo X.509 v3
- Phương thức quản lý khóa lai ghép giữa cấu trúc phân cấp CA theo đúng X.509 và mạng lưới tin cậy của PGP
- Mỗi người dùng có một danh sách các khóa của bản thân, danh sách các khóa tin cậy và danh sách thu hồi chứng thực
- Chứng thực phải được ký bởi CA tin cậy

Chương 7

AN TOÀN IP

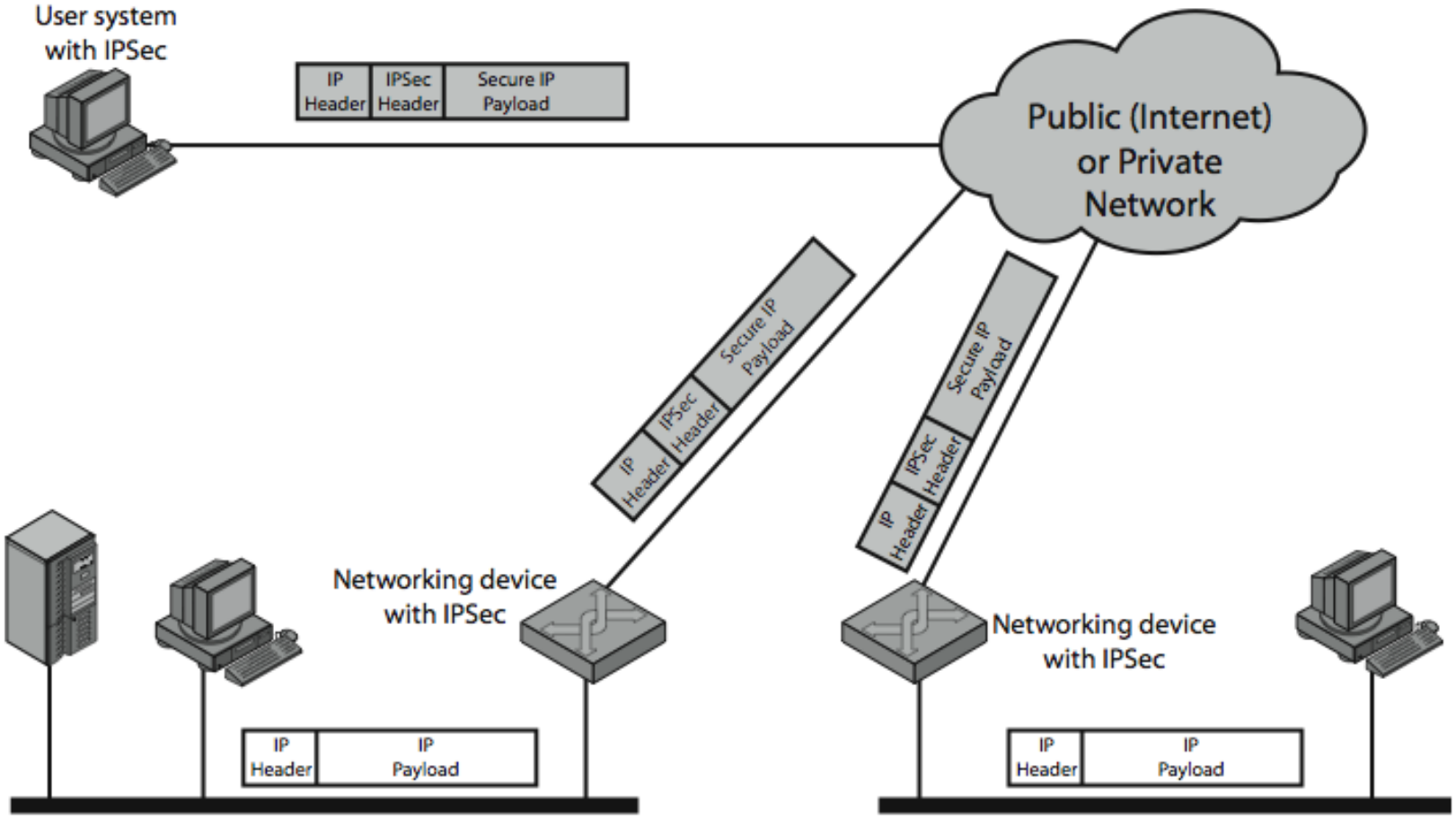
Giới thiệu

- Lý do cần IPSec
  - Có những vấn đề an toàn cần giải quyết ở mức thấp hơn tầng ứng dụng
    - Đặc biệt các hình thức tấn công ở tầng IP rất phổ biến như giả mạo IP, xem trộm gói tin
  - An toàn ở mức IP sẽ đảm bảo an toàn cho tất cả các ứng dụng
    - Bao gồm nhiều ứng dụng chưa có tính năng an toàn
- Các cơ chế an toàn của IPSec
  - Xác thực
  - Bảo mật
  - Quản lý khóa

Các ứng dụng của IPSec

- Xây dựng mạng riêng ảo an toàn trên Internet
  - Tiết kiệm chi phí thiết lập và quản lý mạng riêng
- Truy nhập từ xa an toàn thông qua Internet
  - Tiết kiệm chi phí đi lại
- Giao tiếp an toàn với các đối tác
  - Đảm bảo xác thực, bảo mật và cung cấp cơ chế trao đổi khóa
- Tăng cường an toàn thương mại điện tử
  - Hỗ trợ thêm cho các giao thức an toàn có sẵn của các ứng dụng Web và thương mại điện tử

Minh họa ứng dụng IPSec



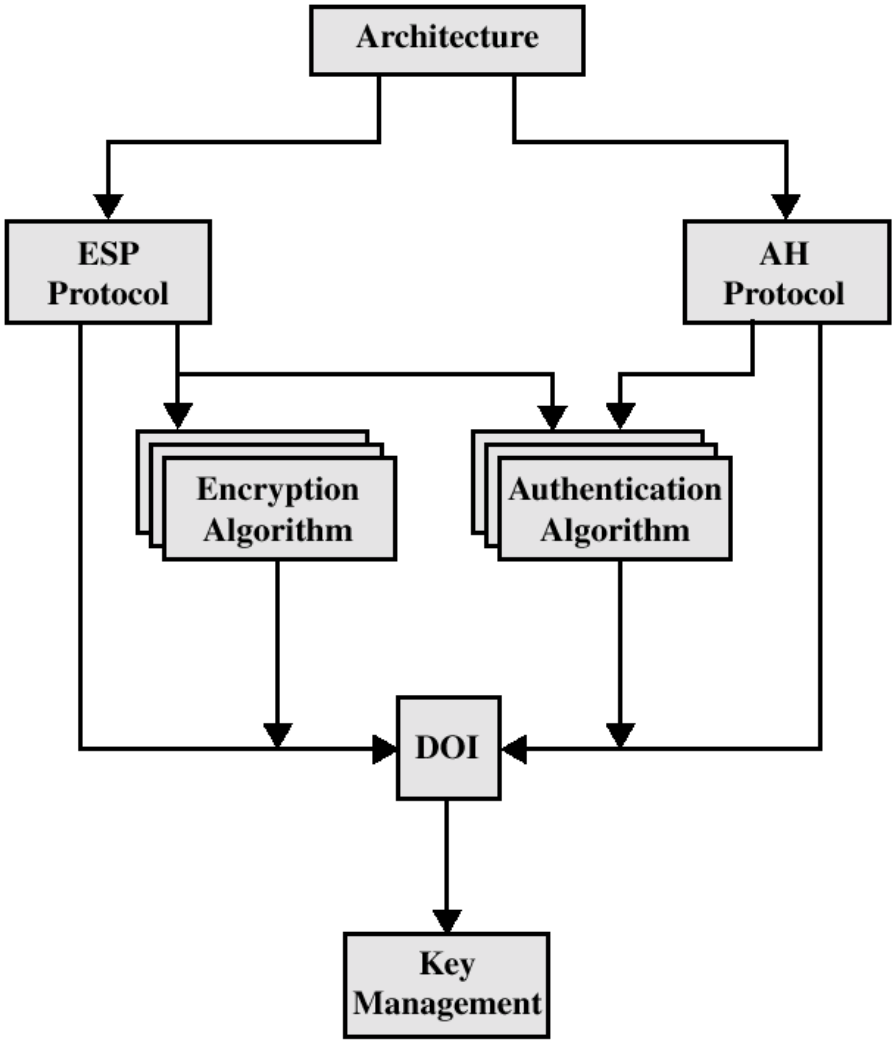
Ích lợi của IPSec

- Tại tường lửa hoặc bộ định tuyến, IPSec đảm bảo an toàn cho mọi luồng thông tin vượt biên
- Tại tường lửa, IPSec ngăn chặn thâm nhập trái phép từ Internet vào
- IPSec nằm dưới tầng giao vận, do vậy trong suốt với các ứng dụng
- IPSec có thể trong suốt với người dùng cuối
- IPSec có thể áp dụng cho người dùng đơn lẻ
- IPSec bảo vệ an toàn kiến trúc định tuyến

Kiến trúc an toàn IP

- Đặc tả IPSec khá phức tạp
- Định nghĩa trong nhiều tài liệu
  - Bao gồm RFC 2401 (tổng quan kiến trúc), RFC 2402 (mô tả mở rộng xác thực), RFC 2406 (mô tả mở rộng mã hóa), RFC 2408 (đặc tả khả năng trao đổi khóa)
  - Các tài liệu khác được chia thành 7 nhóm
- Việc hỗ trợ IPSec là bắt buộc đối với IPv6, tùy chọn đối với IPv4
- IPSec được cài đặt như các phần đầu mở rộng sau phần đầu IP
  - Phần đầu mở rộng cho xác thực là AH
  - Phần đầu mở rộng cho mã hóa là ESP

Tổng quan tài liệu IPSec



Các dịch vụ IPSec

- Bao gồm
  - Điều khiển truy nhập
  - Toàn vẹn phi kết nối
  - Xác thực nguồn gốc dữ liệu
  - Từ chối các gói tin lặp
    - Một hình thức của toàn vẹn thứ tự bộ phận
  - Bảo mật (mã hóa)
  - Bảo mật luồng tin hữu hạn
- Sử dụng một trong hai giao thức
  - Giao thức xác thực (ứng với AH)
  - Giao thức xác thực/mã hóa (ứng với ESP)

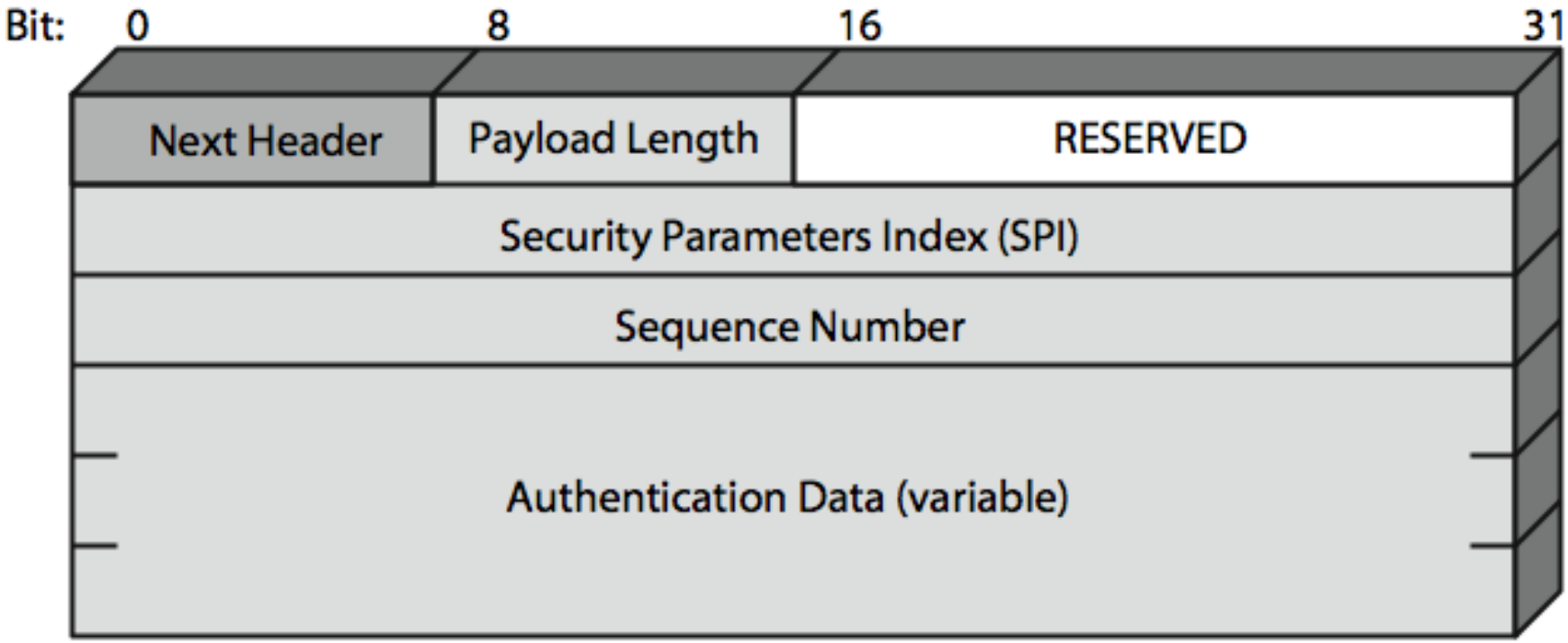
Các liên kết an toàn

- Khái niệm liên kết an toàn (SA)
  - Là quan hệ một chiều giữa bên gửi và bên nhận, cho biết các dịch vụ an toàn đối với luồng tin lưu chuyển
- Mỗi SA được xác định duy nhất bởi 3 tham số
  - Chỉ mục các tham số an toàn (SPI)
  - Địa chỉ IP đích
  - Định danh giao thức an toàn
- Các tham số khác lưu trong CSDL SA (SAD)
  - Số thứ tự, các thông tin AH và ESP, thời hạn,...
- CSDL chính sách an toàn (SPD) cho phép điều chỉnh mức độ áp dụng IPSec

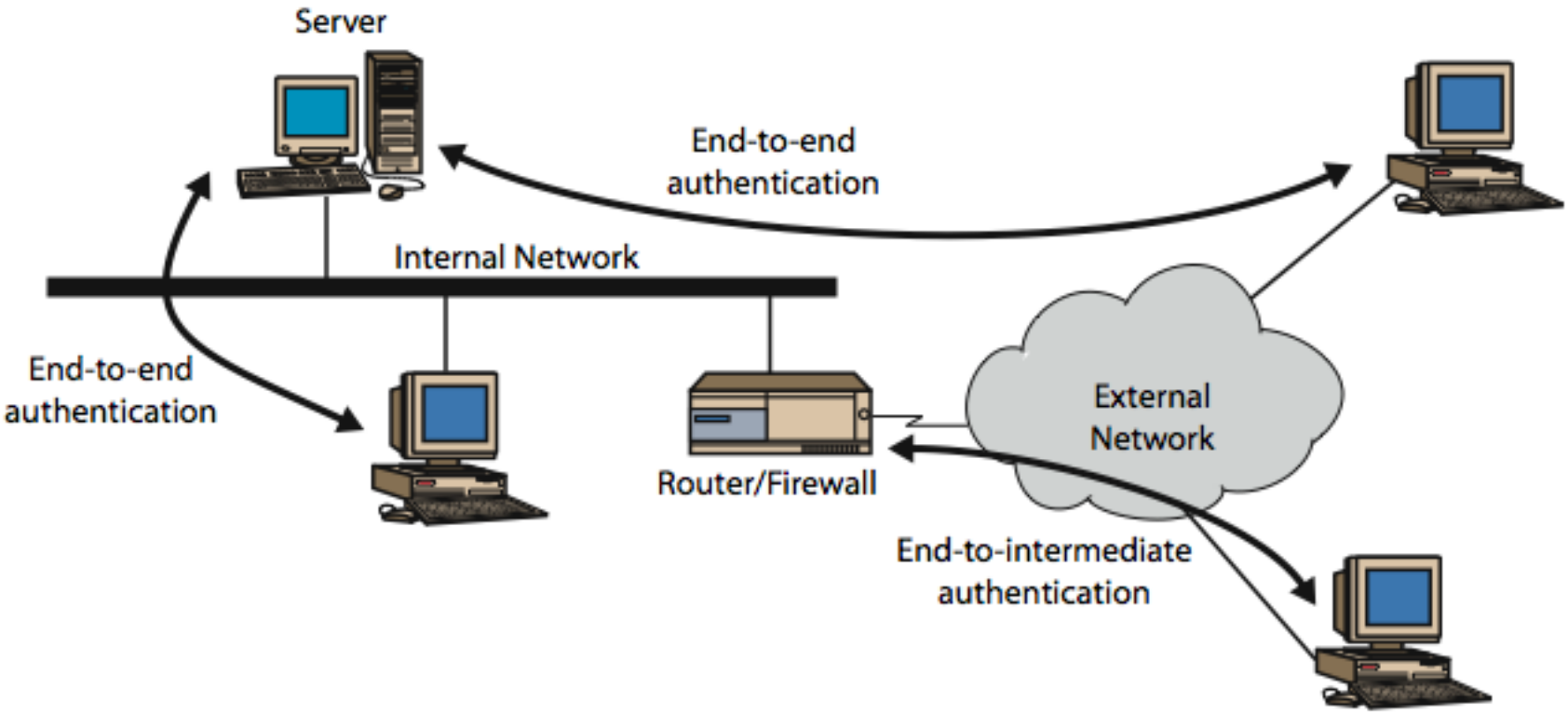
Phần đầu xác thực

- Đảm bảo toàn vẹn và xác thực các gói IP
  - Cho phép một hệ thống đầu cuối hay một thiết bị mạng xác thực người dùng hoặc ứng dụng
  - Tránh giả mạo địa chỉ nhờ xem xét số thứ tự
  - Chống lại hình thức tấn công lặp lại
- Sử dụng mã xác thực thông báo
- Bên gửi và bên nhận phải có một khóa bí mật dùng chung

Khuôn dạng AH



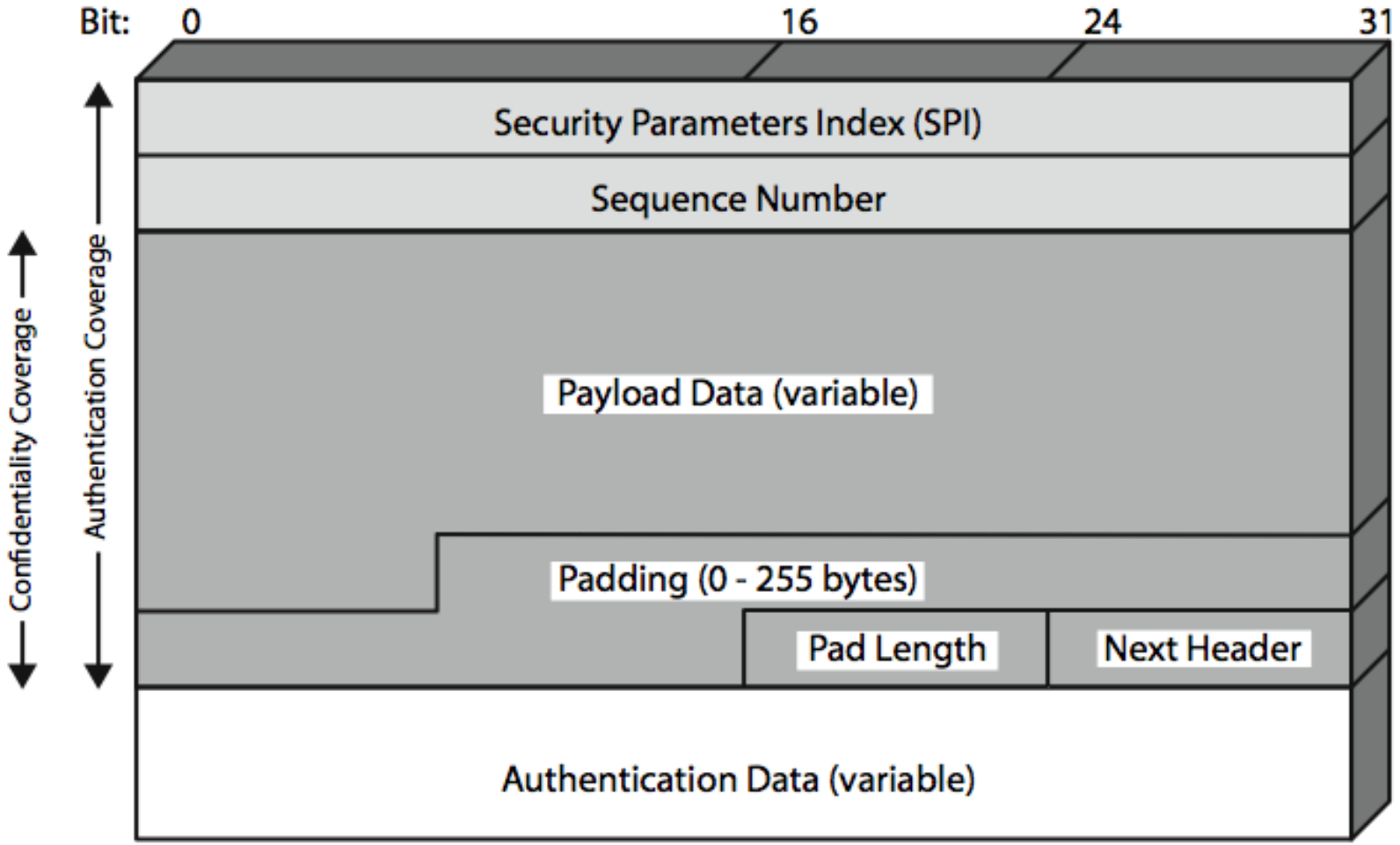
Chế độ giao vận và đường hầm



Phần đầu ESP

- Đảm bảo bảo mật nội dung và bảo mật luồng tin hữu hạn
- Có thể cung cấp các dịch vụ xác thực giống như với AH
- Cho phép sử dụng nhiều giải thuật mã hóa, phương thức mã hóa, và cách độn khác nhau
  - DES, 3DES, RC5, IDEA, CAST,...
  - CBC,...
  - Độn cho tròn kích thước khối, kích thước trường, che dấu lưu lượng luồng tin

Khuôn dạng ESP



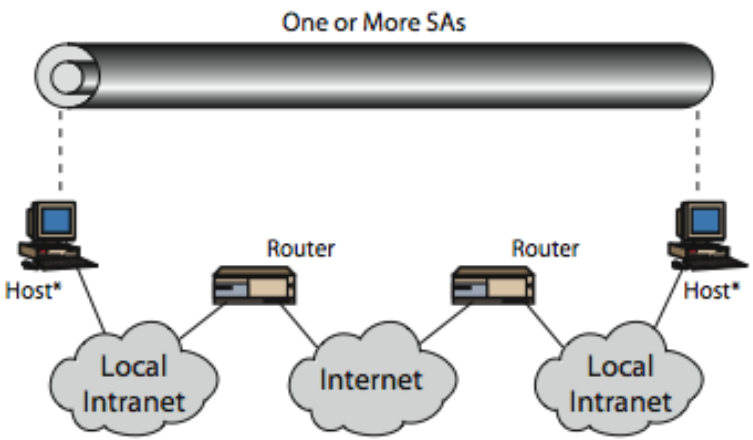
Giao vận và đường hầm ESP

- Chế độ giao vận ESP dùng để mã hóa và có thể có thêm chức năng xác thực dữ liệu IP
  - Chỉ mã hóa dữ liệu không mã hóa phần đầu
  - Dễ bị phân tích lưu lượng nhưng hiệu quả
  - Áp dụng cho truyền tải giữa hai điểm cuối
- Chế độ đường hầm mã hóa toàn bộ gói tin IP
  - Phải bổ xung phần đầu mới cho mỗi bước chuyển
  - Áp dụng cho các mạng riêng ảo, truyền tải thông qua cầu nối

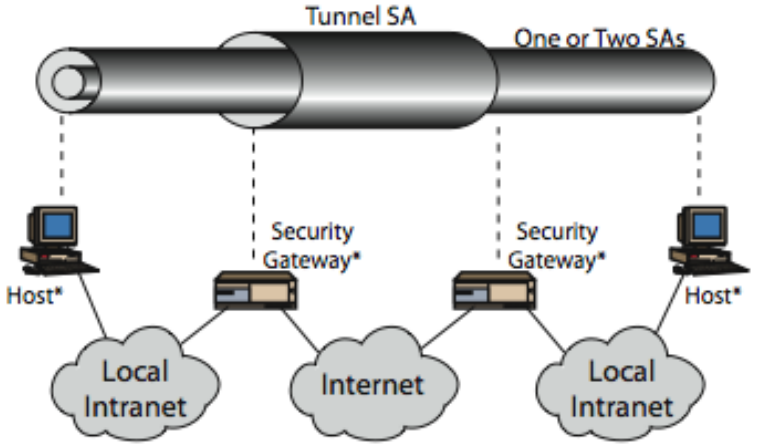
Kết hợp các liên kết an toàn

- Mỗi SA chỉ có thể cài đặt một trong hai giao thức AH và ESP
- Để cài đặt cả hai cần kết hợp các SA với nhau
  - Tạo thành một gói liên kết an toàn
  - Có thể kết thúc tại các điểm cuối khác nhau hoặc giống nhau
- Kết hợp theo 2 cách
  - Gần với giao vận
  - Tạo đường hầm theo nhiều bước
- Cần xem xét thứ tự xác thực và mã hóa

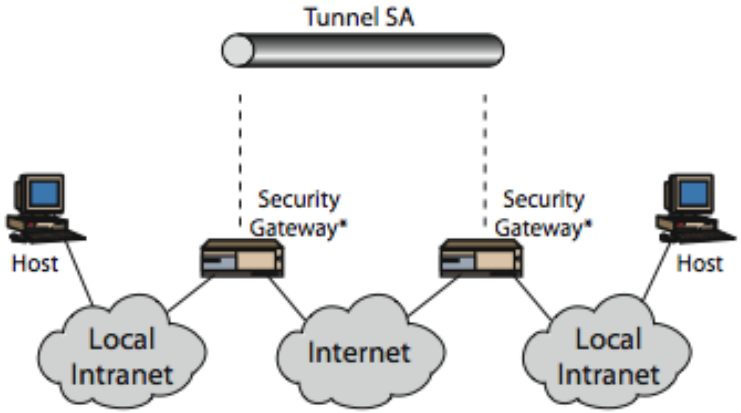
Ví dụ kết hợp các SA



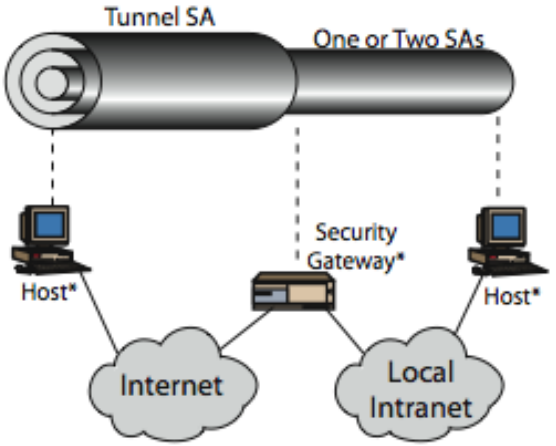
(a) Case 1



(c) Case 3



(b) Case 2



(d) Case 4

Quản lý khóa

- Có chức năng sản sinh và phân phối khóa
- Hai bên giao tiếp với nhau nói chung cần 4 khóa
  - Mỗi chiều cần 2 khóa: 1 cho AH, 1 cho ESP
- Hai chế độ quản lý khóa
  - Thủ công
    - Quản trị hệ thống khai báo các khóa khi thiết lập cấu hình
    - Thích hợp với các môi trường nhỏ và tương đối tĩnh
  - Tự động
    - Cho phép tạo khóa theo yêu cầu cho các SA
    - Thích hợp với các hệ phân tán lớn có cấu hình luôn thay đổi
    - Gồm các thành phần Oakley và ISAKMP

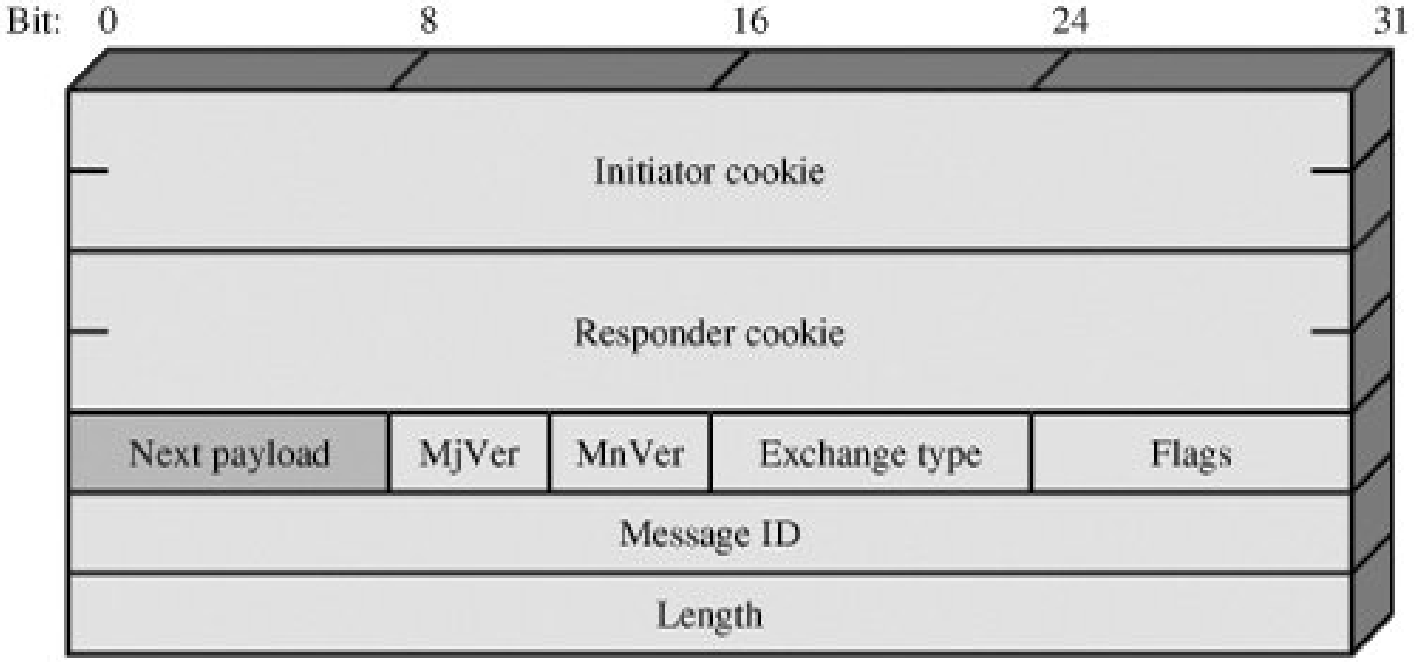
Oakley

- Là một giao thức trao đổi khóa dựa trên giải thuật Diffie-Hellman
- Bao gồm một số cải tiến quan trọng
  - Sử dụng cookie để ngăn tấn công gây quá tải
    - Cookie cần phụ thuộc vào các bên giao tiếp, không thể sinh ra bởi một bên khác với bên sinh cookie, có thể sinh và kiểm tra một cách nhanh chóng
  - Hỗ trợ việc sử dụng các nhóm với các tham số Diffie-Hellman khác nhau
  - Sử dụng các giá trị nonce để chống tấn công lặp lại
  - Xác thực các trao đổi Diffie-Hellman để chống tấn công người ở giữa

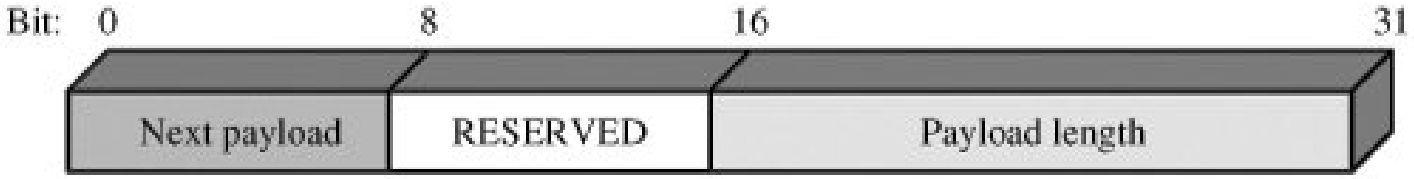
ISAKMP

- Viết tắt của Internet Security Association and Key Management Protocol
- Cung cấp một cơ cấu cho việc quản lý khóa
- Định nghĩa các thủ tục và các khuôn dạng thông báo cho việc thiết lập, thỏa thuận, sửa đổi, và hủy bỏ các liên kết an toàn
- Độc lập với giao thức trao đổi khóa, giải thuật mã hõa, và phương pháp xác thực

Các khuôn dạng ISAKMP



(a) ISAKMP header



(b) Generic payload header

Chương 8

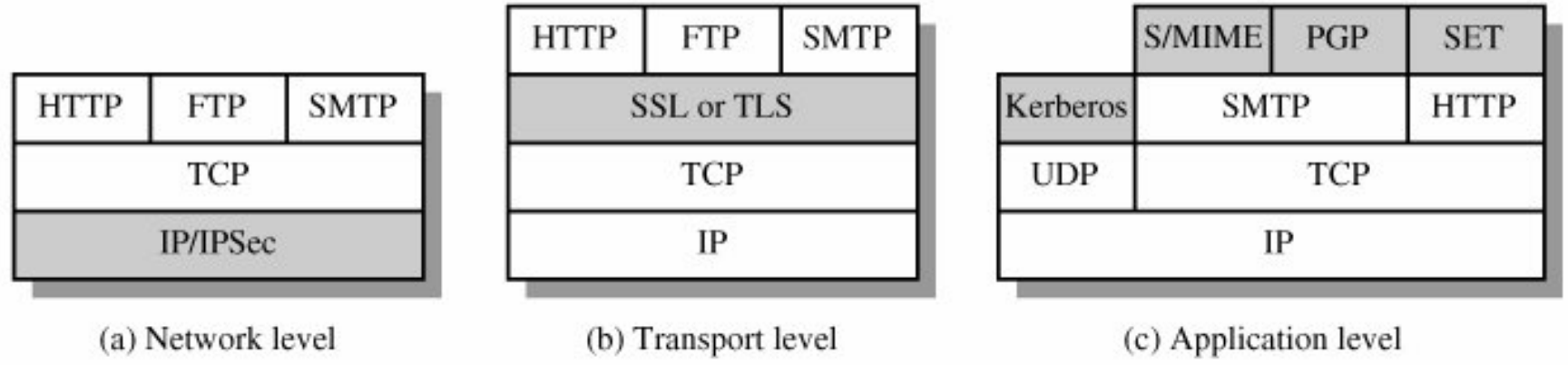
AN TOÀN WEB

Vấn đề an toàn Web (1)

- Web được sử dụng rộng rãi bởi các công ty, tổ chức, và các cá nhân
- Các vấn đề đặc trưng đối với an toàn Web
  - Web dễ bị tấn công theo cả hai chiều
  - Tấn công Web server sẽ gây tổn hại đến danh tiếng và tiền bạc của công ty
  - Các phần mềm Web thường chứa nhiều lỗi an toàn
  - Web server có thể bị khai thác làm căn cứ để tấn công vào hệ thống máy tính của một tổ chức
  - Người dùng thiếu công cụ và kiến thức để đối phó với các hiểm họa an toàn

Vấn đề an toàn Web (2)

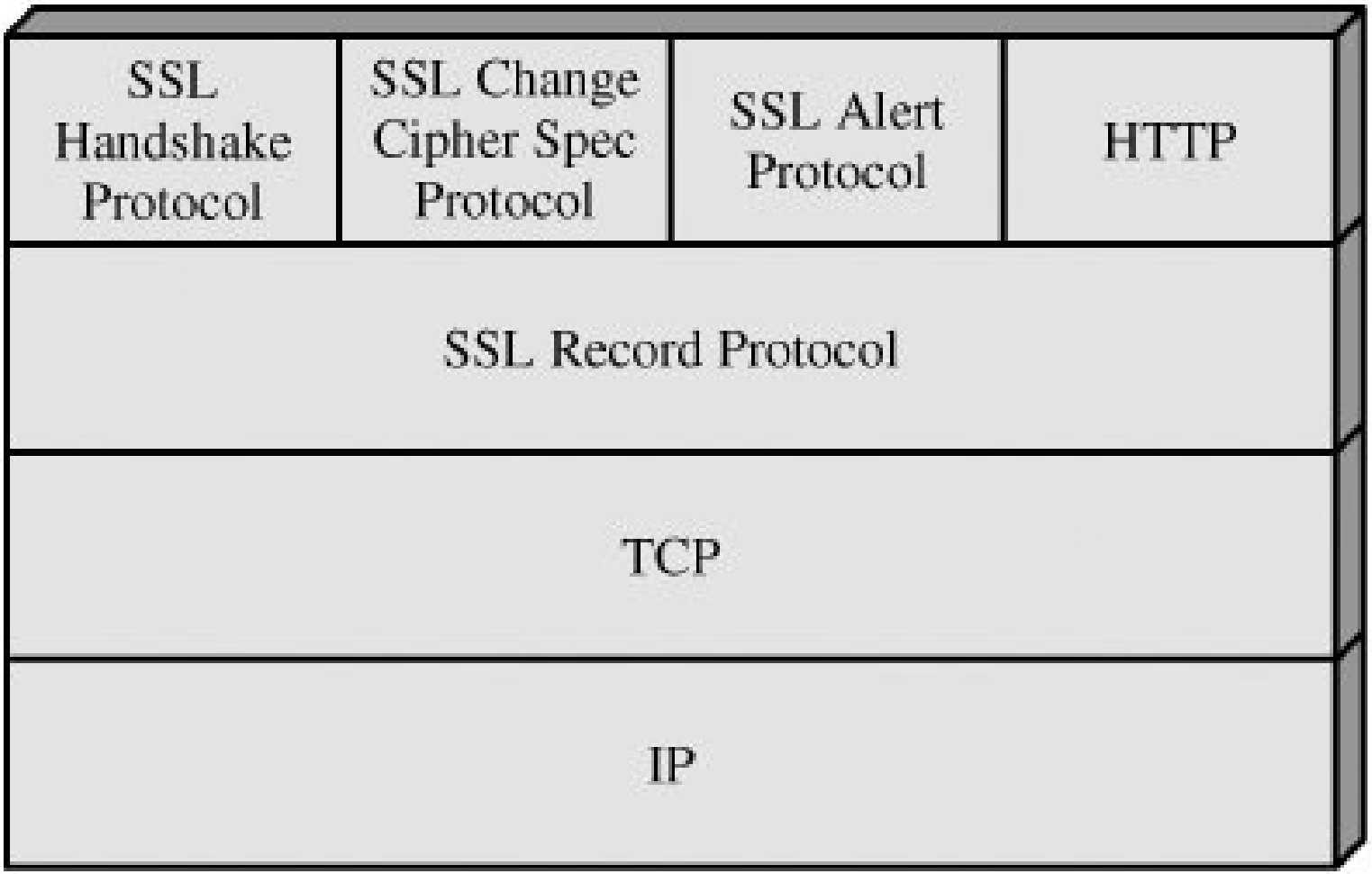
- Các hiểm họa đối với an toàn Web
  - Tính toàn vẹn
  - Tính bảo mật
  - Từ chối dịch vụ
  - Xác thực
- Các biện pháp an toàn Web



SSL

- Là một dịch vụ an toàn ở tầng giao vận
- Do Netscape khởi xướng
- Phiên bản 3 được công bố dưới dạng bản thảo Internet
- Trở thành chuẩn TLS
  - Phiên bản đầu tiên của TLS ≈ SSLv3.1 tương thích ngược với SSLv3
- Sử dùng TCP để cung cấp dịch vụ an toàn từ đầu cuối tới đầu cuối
- Gồm 2 tầng giao thức

Mô hình phân tầng SSL



Kiến trúc SSL (1)

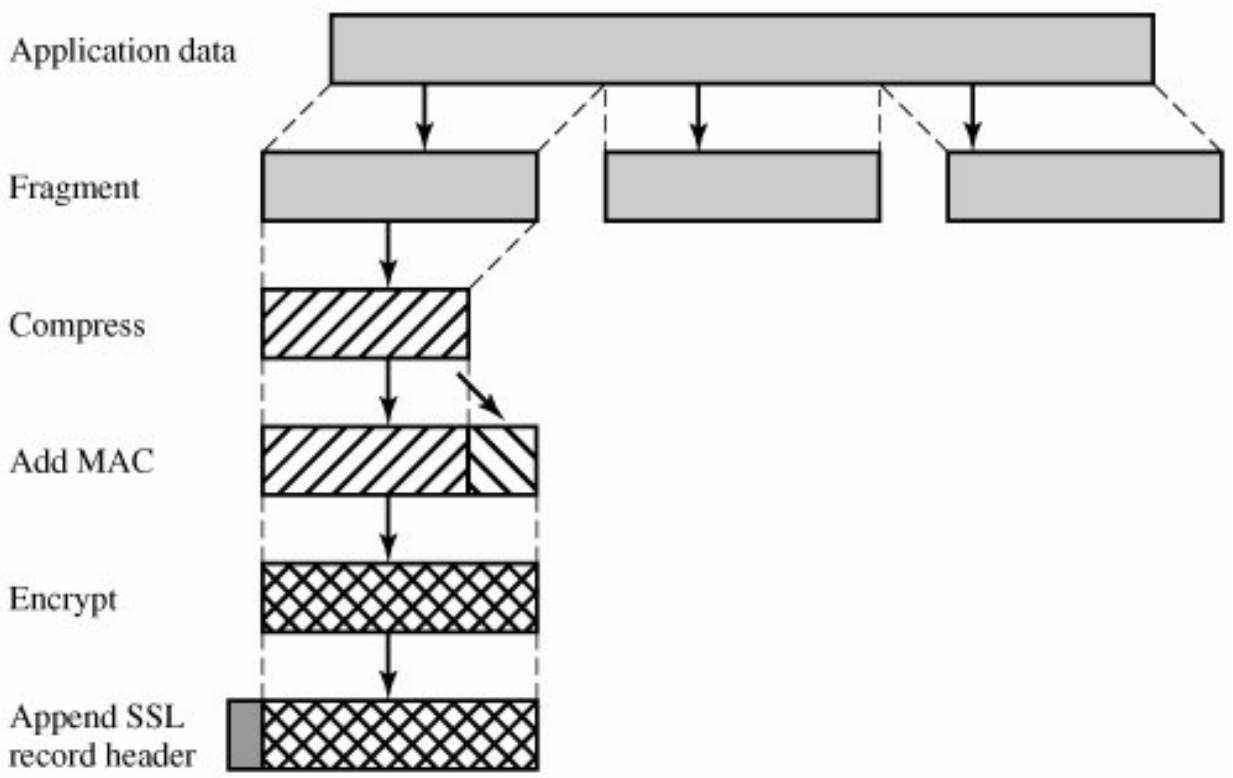
- Kết nối SSL
  - Liên kết giao tiếp từ điểm nút tới điểm nút
  - Mang tính nhất thời
  - Gắn với một phiên giao tác
  - Các tham số xác định trạng thái kết nối
    - Các số ngẫu nhiên chọn bởi server và client
    - Khóa MAC của server
    - Khóa MAC của client
    - Khóa mã hóa của server
    - Khóa mã hóa client
    - Các vector khởi tạo
    - Các số thứ tự

Kiến trúc SSL (2)

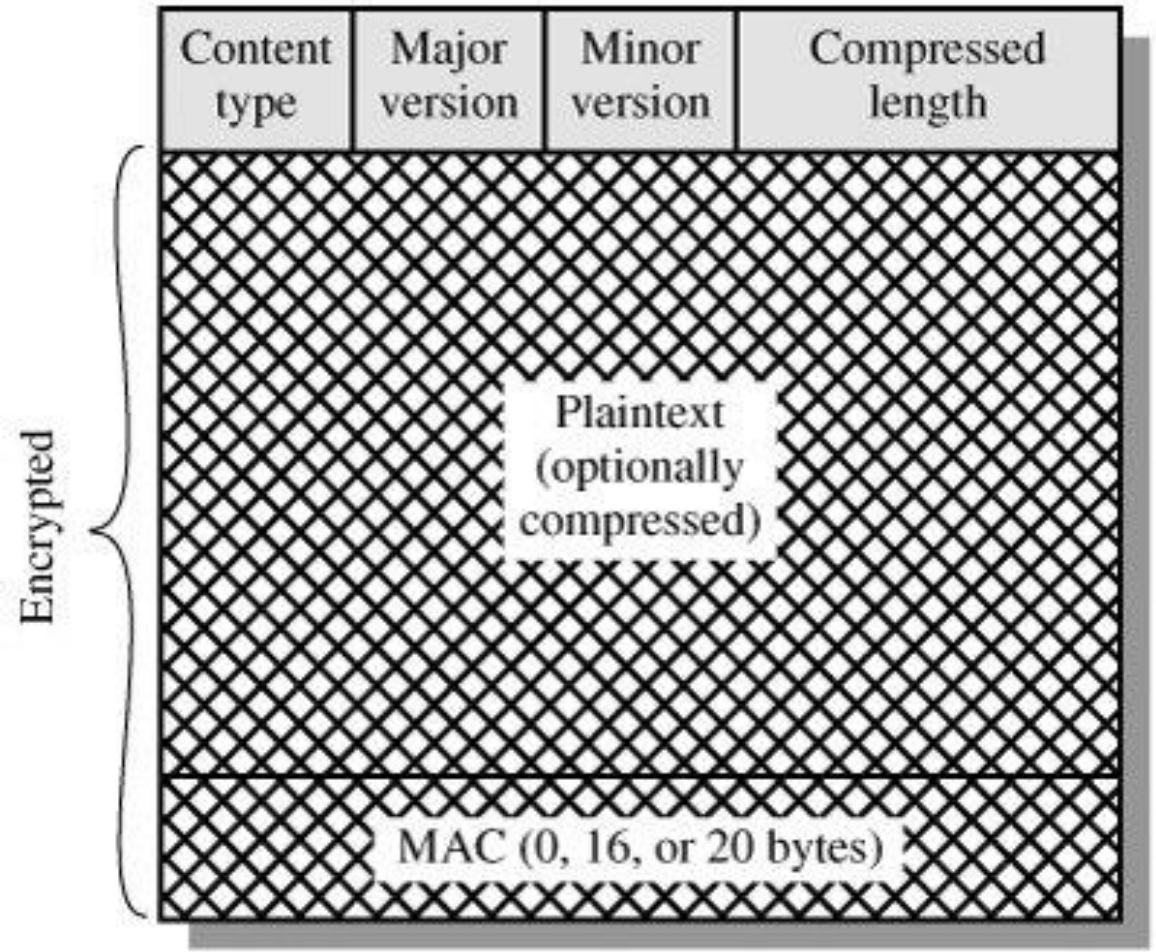
- Phiên SSL
  - Liên kết giữa client và server
  - Tạo lập nhờ giao thức bắt tay
  - Có thể bao gồm nhiều kết nối
  - Xác lập một tập các tham số an toàn sử dụng bởi tất cả các kết nối trong phiên giao tác
    - Định danh phiên
    - Chứng thực điểm nút
    - Phương pháp nén
    - Đặc tả mã hóa
    - Khóa bí mật chủ
    - Cờ có thể tiếp tục hay không

Giao thức bản ghi SSL

- Cung cấp các dịch vụ bảo mật và xác thực
  - Khóa bí mật chung do giao thức bắt tay xác lập



Khuôn dạng bản ghi SSL



Giao thức đổi đặc tả mã hóa SSL

- Một trong ba giao thức chuyên dụng SSL sử dụng giao thức bản ghi SSL
- Chỉ gồm một thông báo chứa một byte dữ liệu có giá trị là 1
- Khiến cho trạng thái treo trở thành trạng thái hiện thời
  - Cập nhật đặc tả mã hóa cho kết nối

Giao thức báo động SSL

- Dùng chuyển tải các báo động liên quan đến SSL tới các thực thể điểm nút
- Mỗi thông báo gồm 2 byte
  - Byte thứ nhất chỉ mức độ nghiêm trọng
    - Cảnh báo : có giá trị là 1
    - Tai họa : có giá trị là 2
  - Byte thứ hai chỉ nội dung báo động
    - Tai họa : unexpected_message, bad_record_mac, decompression_failure, handshake_failure, illegal_parameter
    - Cảnh báo : close_notify, no_certificate, bad_certificate, unsupported_certificate, certificate_revoked, certificate_expired, certificate_unknown

Giao thức bắt tay SSL

- Cho phép server và client
  - Xác thực lẫn nhau
  - Thỏa thuận các giải thuật mã hóa và MAC
  - Thỏa thuận các khóa mật mã sẽ được sử dụng
- Gồm một chuỗi các thông báo trao đổi giữa client và server
- Mỗi thông báo gồm 3 trường
  - Kiểu (1 byte)
  - Độ dài (3 byte)
  - Nội dung ($\geq$ 0 byte)

TLS

- Là phiên bản chuẩn Internet của SSL
  - Mô tả trong RFC 2246 rất giống với SSLv3
  - Một số khác biệt nhỏ so với SSLv3
    - Số phiên bản trong khuôn dạng bản ghi SSL
    - Sử dụng HMAC để tính MAC
    - Sử dụng hàm giả ngẫu nhiên để khai triển các giá trị bí mật
    - Có thêm một số mã báo động
    - Không hỗ trợ Fortezza
    - Thay đổi trong trao đổi chứng thực
    - Thay đổi trong việc sử dụng dữ liệu đệm