AN TOÀN VÀ AN NINH MẠNG

TS. Nguyễn Đại Thọ
Trường Đại học Công nghệ
Đại học Quốc gia Hà Nội

Chương 1

GIỚI THIỆU

Bối cảnh xã hội

- Thế giới đứng trước thách thức của các tấn công khủng bố với an ninh được thắt chặt
- Công nghệ thông tin cũng là nạn nhân của một số lượng lớn chưa từng có các tấn công
- An toàn thông tin là một thành phần cốt lõi của công nghệ thông tin
  - Bảo vệ thông tin điện tử có giá trị
- Nhu cầu về các chuyên gia CNTT biết bảo vệ an toàn mạng và máy tính là rất lớn

Bối cảnh công nghệ

- Hai biến đổi lớn trong yêu cầu về an toàn thông tin thời gian gần đây
  - Trước đây an toàn thông tin được đảm bảo bằng các biện pháp vật lý và hành chính
  - Sử dụng máy tính tạo yêu cầu về các công cụ tự động để bảo vệ file và các thông tin lưu trữ khác
  - Sử dụng mạng và các phương tiện truyền thông tạo yêu cầu về các biện pháp bảo vệ dữ liệu trong khi truyền

Làm rõ khái niệm an toàn thông tin

- An toàn
  - Trạng thái không bị nguy hiểm hoặc rủi ro
    - Trạng thái hay điều kiện đó tồn tại vì các biện pháp bảo vệ được thiết lập và duy trì
- An toàn thông tin
  - Mô tả nhiệm vụ bảo vệ thông tin ở khuôn dạng số
- An toàn thông tin có thể được hiểu thông qua xem xét mục tiêu và cách thức thực hiện

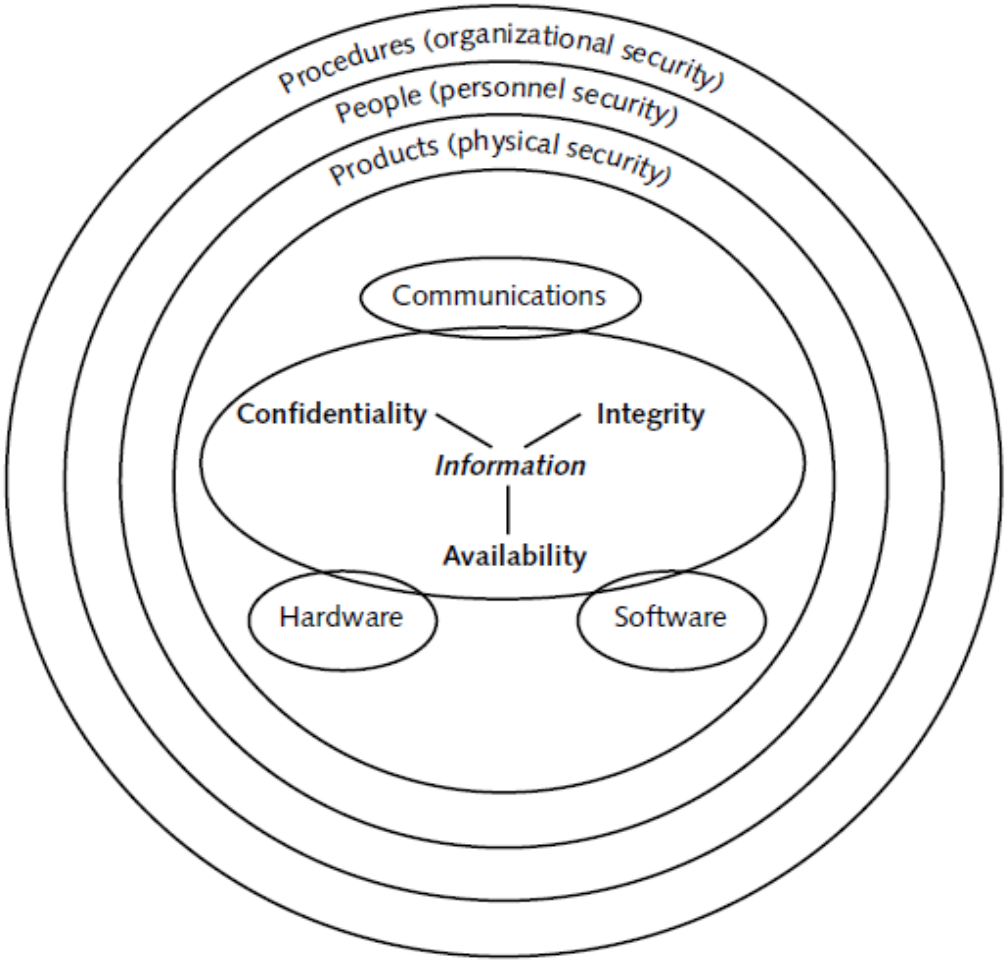
Mục tiêu của an toàn thông tin

- Đảm bảo các biện pháp bảo vệ được thực hiện một cách thích hợp
- Bảo vệ thông tin có giá trị đối với con người hoặc tổ chức
  - Giá trị ở các đặc tính **bảo mật**, **toàn vẹn**, và **khả dụng**
- Bảo vệ các đặc tính của thông tin trên các thiết bị lưu trữ, thao tác, và truyền thông tin

Cách thức thực hiện ATTT

- Thông qua kết hợp 3 thực thể
  - Phần cứng, phần mềm, và truyền thông
- Ba lớp bảo vệ
  - Sản phẩm
    - An ninh vật lý xung quanh dữ liệu
  - Con người
    - Những người cài đặt và sử dụng các sản phẩm an ninh
  - Thủ tục
    - Kế hoạch và chính sách đảm bảo sử dụng đúng đắn các sản phẩm

Các thành phần an toàn thông tin



Định nghĩa an toàn thông tin

- Một định nghĩa hoàn chỉnh hơn về an toàn thông tin
  - *Là vấn đề bảo vệ tính toàn vẹn, tính bảo mật, và tính khả dụng của thông tin trên các thiết bị lưu trữ, thao tác, và truyền dẫn thông tin thông qua các sản phẩm, con người, và các thủ tục*

Các khái niệm an toàn thông tin (1)

- Tính bảo mật
  - Bảo vệ những hạn chế cho phép về truy nhập và tiết lộ thông tin
    - Bao gồm các biện pháp bảo vệ tính riêng tư cá nhân và thông tin độc quyền
- Tính toàn vẹn
  - Bảo vệ thông tin khỏi bị sửa đổi hoặc triệt tiêu một cách không thích hợp
    - Bao gồm đảm bảo tính không thể chối bỏ và tính xác thực của thông tin

Các khái niệm an toàn thông tin (2)

- Tính khả dụng
  - Đảm bảo truy nhập và sử dụng thông tin một cách kịp thời và đáng tin cậy
- Tính xác thực
  - Tính chân thật và có thể kiểm tra và tin cậy được
- Tính trách nhiệm
  - Mục tiêu an ninh quy định các hành động của một thực thể phải được quy một cách duy nhất về thực thể đó

Các định nghĩa về an toàn

- An toàn máy tính
  - Tên chung cho tập các công cụ được thiết kế để bảo vệ dữ liệu và chống lại tin tặc
- An toàn mạng
  - Các biện pháp bảo vệ dữ liệu khi truyền dẫn
- An toàn liên mạng
  - Các biện pháp bảo vệ dữ liệu khi truyền dẫn qua một tập các mạng kết nối với nhau

Các thách thức an toàn máy tính (1)

- Không đơn giản như lầm tưởng ban đầu
- Luôn phải xem xét các tấn công tiềm tàng vào các tính năng an ninh muốn phát triển
- Các thủ tục an ninh thường trái với trực quan
- Phải quyết định triển khai các cơ chế an ninh ở đâu
- Bao hàm nhiều hơn một giải thuật hay giao thức và cần tới thông tin bí mật

Các thách thức an toàn máy tính (2)

- Cuộc đấu trí giữa kẻ tấn công và người thiết kế hay quản trị
- Không thấy là có lợi cho đến khi bị phá hoại
- Yêu cầu giám sát đều đặn thậm chí thường xuyên
- Quá thường xuyên là giải pháp tích hợp sau khi hoàn thành thiết kế
- Bị coi là trở ngại đối với việc sử dụng hiệu quả và thân thiện hệ thống hoặc thông tin

Kiến trúc an ninh OSI

- Mục tiêu
  - Ước định một cách có hiệu quả các nhu cầu an ninh
  - Đánh giá và lựa chọn các sản phẩm và chính sách an ninh thích hợp
- "Kiến trúc an ninh cho OSI" của ITU-T X.800
- Một cách thức có hệ thống định nghĩa và đáp ứng các nhu cầu an ninh
- Cung cấp một tổng quan hữu ích mặc dù trừu tượng về các khái niệm sẽ nghiên cứu

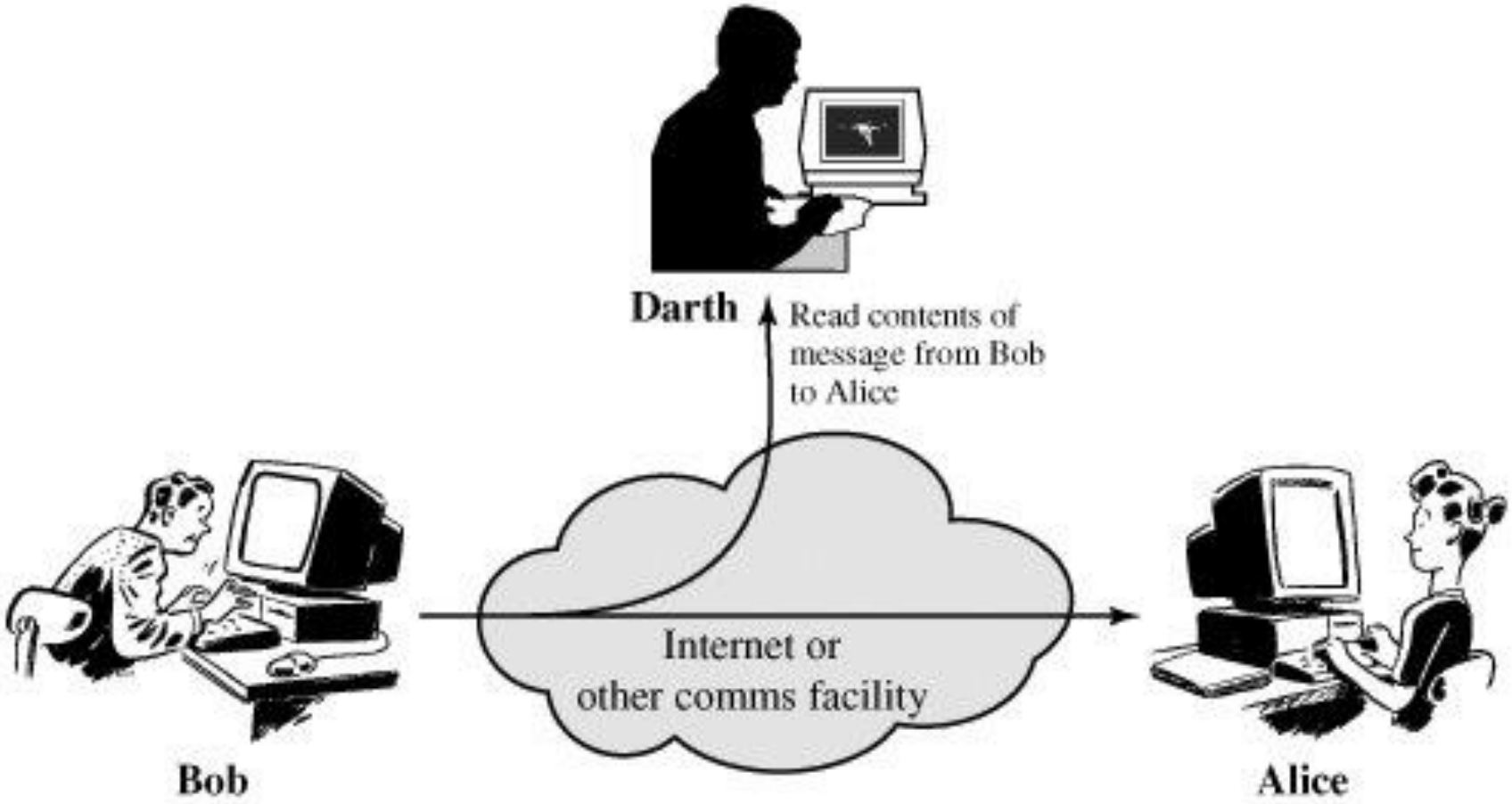
Các khía cạnh an ninh

- Tấn công an ninh
  - Hành động làm tổn hại an toàn thông tin
- Cơ chế an ninh
  - Quá trình được thiết kế để phát hiện, ngăn ngừa hoặc khôi phục từ một tấn công an ninh
- Dịch vụ an ninh
  - Dịch vụ tăng cường an ninh của các hệ thống xử lý dữ liệu và các chuyển giao thông tin

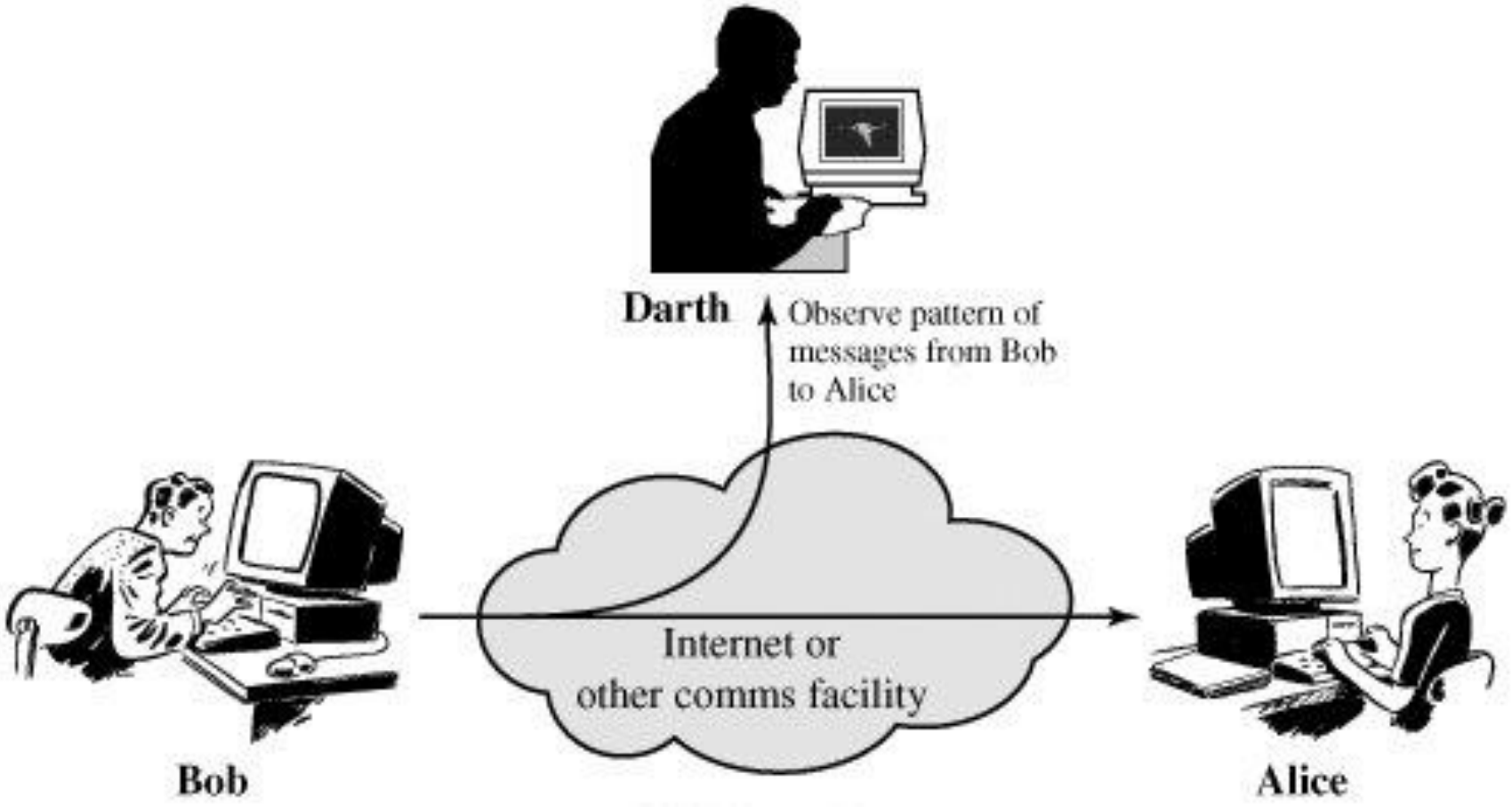
Tấn công thụ động

- Tìm cách nắm bắt và sử dụng thông tin nhưng không tác động đến tài nguyên hệ thống
  - Không bao hàm bất kỳ sửa đổi nào trên dữ liệu
- Hai kiểu
  - Làm lộ nội dung thông báo
  - Phân tích lưu lượng
- Chú trọng ngăn ngừa thay vì phát hiện
  - Thường bằng các biện pháp mã hóa

Làm lộ nội dung thông báo



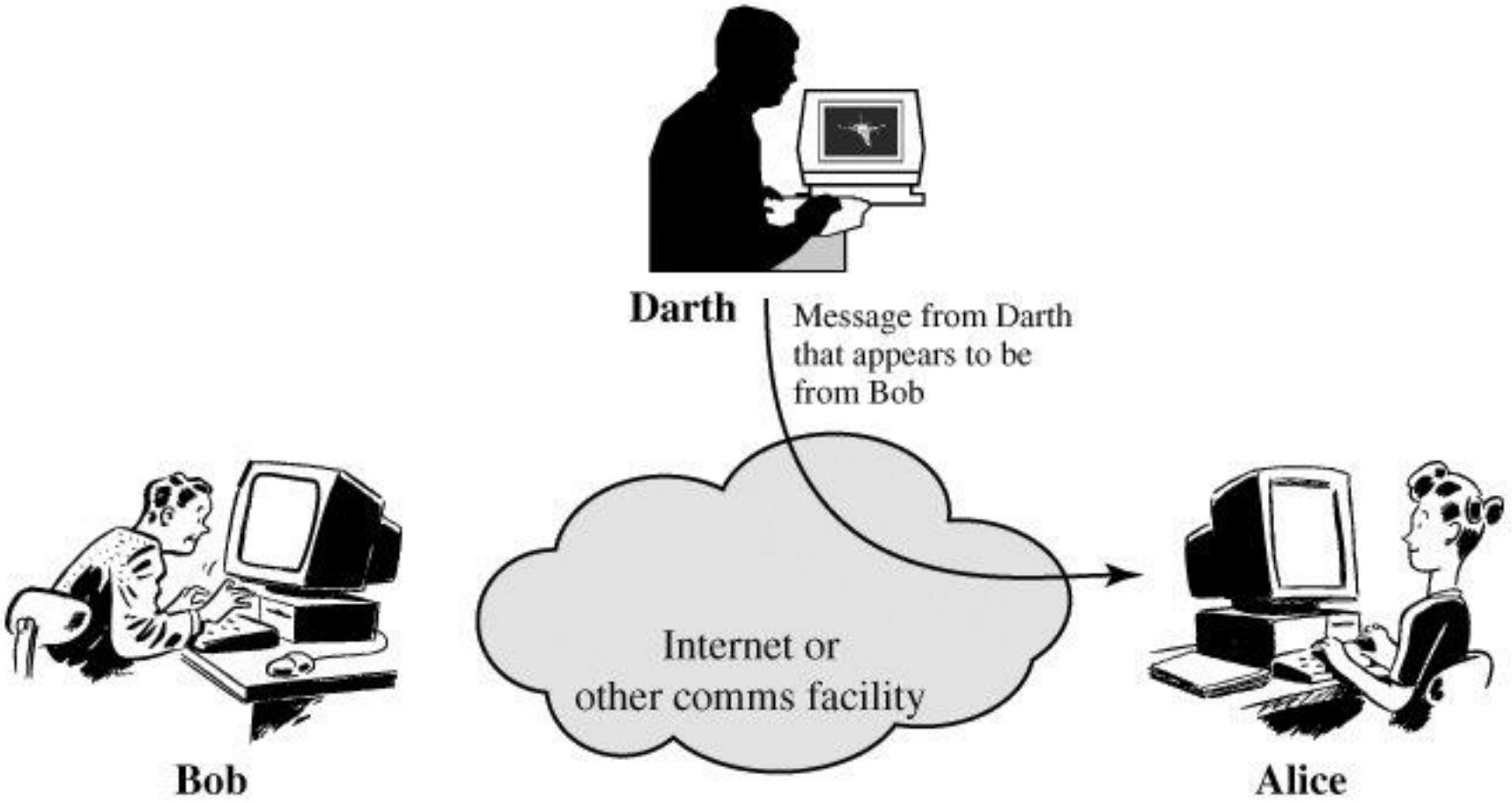
Phân tích lưu lượng



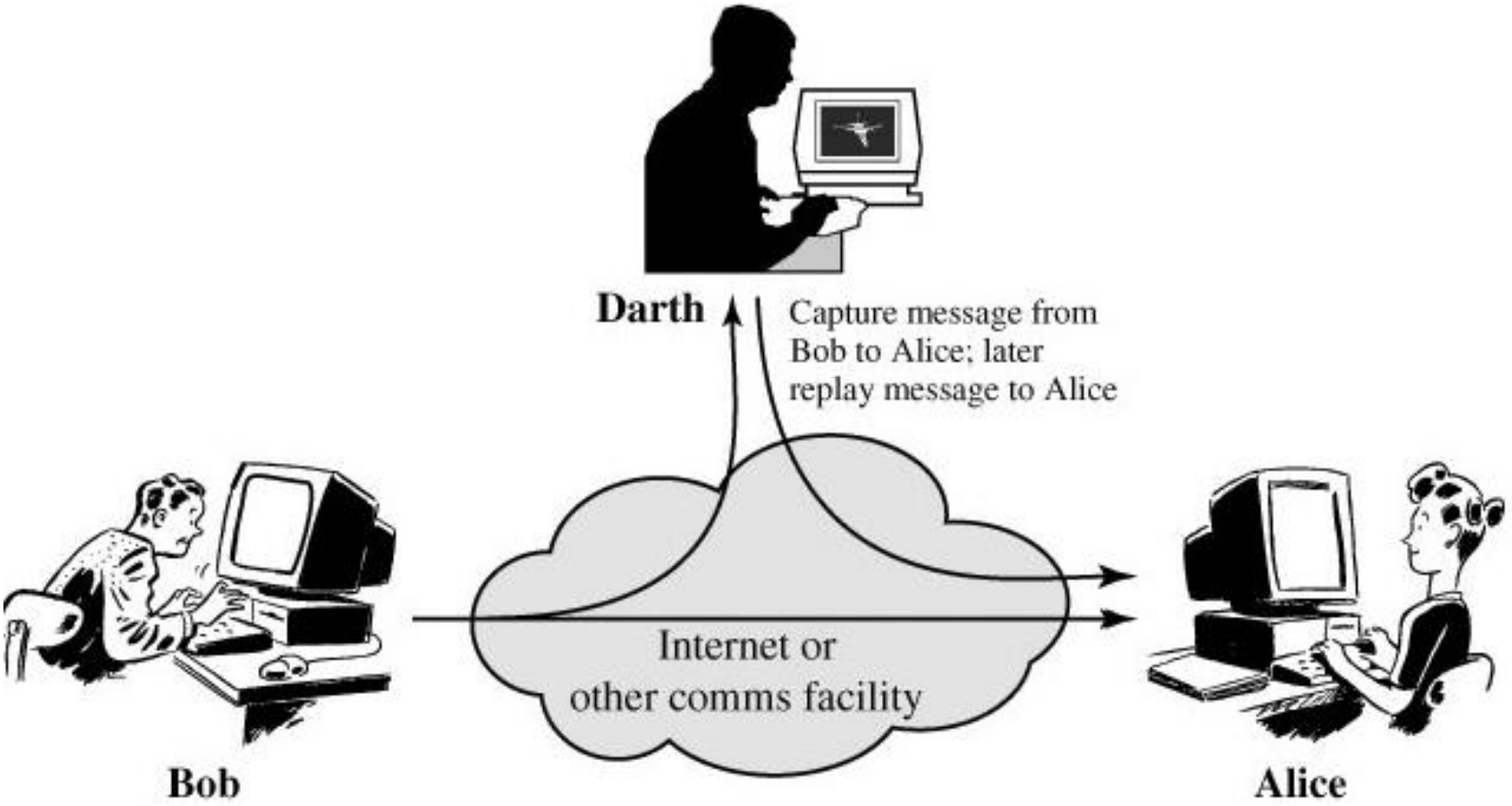
Tấn công chủ động

- Bao hàm việc sửa đổi luồng dữ liệu hoặc tạo ra luồng dữ liệu giả
- Bốn kiểu
  - Giả mạo
  - Lặp lại
  - Sửa đổi thông báo
  - Từ chối dịch vụ
- Mục tiêu là phát hiện tấn công chủ động và khôi phục khỏi ngưng trệ hay chậm trễ
  - Phát hiện có thể góp phần ngăn ngừa

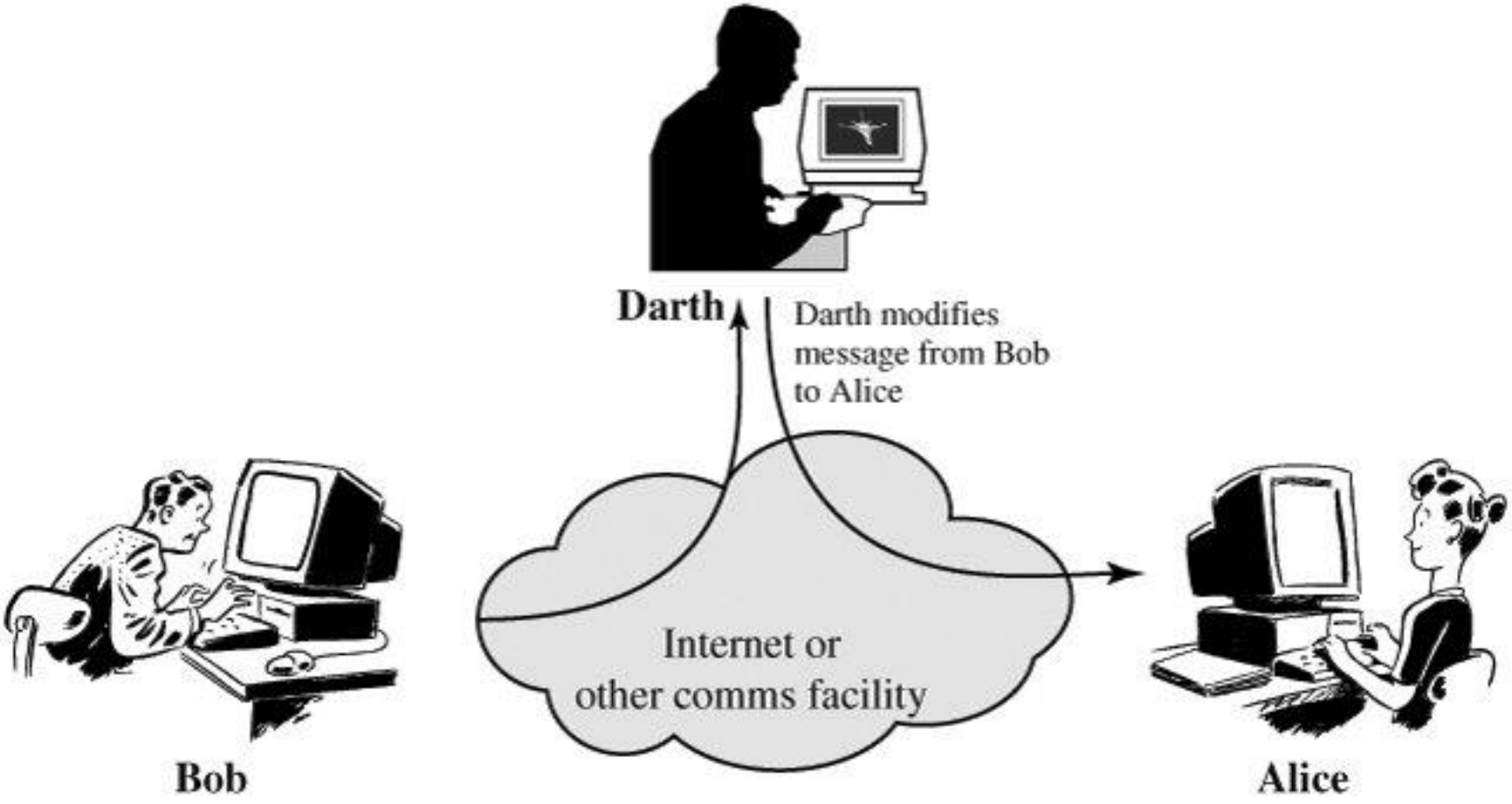
Giả mạo


Darth
Message from Darth
that appears to be
from Bob
Internet or
other comms facility
Bob
Alice

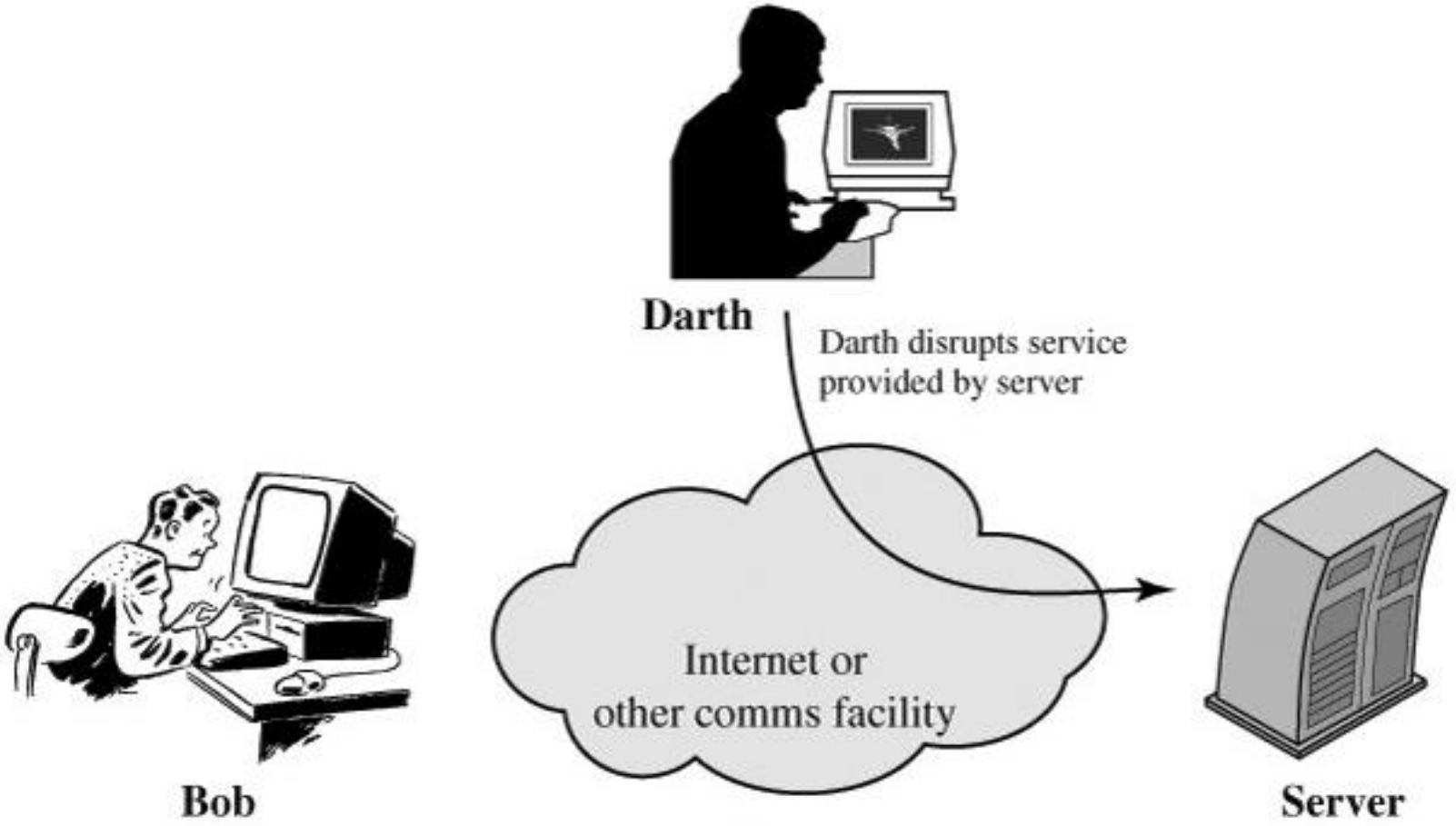
Lặp lại


Darth
Capture message from Bob to Alice; later replay message to Alice
Internet or other comms facility
Bob
Alice

Sửa đổi thông báo



Từ chối dịch vụ



Dịch vụ an ninh

- X.800
  - Dịch vụ cung cấp bởi một tầng giao thức trong các hệ thống mở truyền thông, đảm bảo an toàn thỏa đáng các hệ thống và các chuyển giao dữ liệu
- RFC 2828
  - Dịch vụ xử lý hoặc truyền thông cung cấp bởi một hệ thống để đem lại một loại bảo vệ nhất định cho các tài nguyên hệ thống
- Chủ định chống lại các tấn công an ninh

Các dịch vụ an ninh (X.800) (1)

- Xác thực
  - Đảm bảo thực thể truyền thông là cái nó khai nhận
- Điều khiển truy nhập
  - Ngăn ngừa sử dụng một cách trái phép tài nguyên
- Bảo mật dữ liệu
  - Bảo vệ dữ liệu khỏi bị tiết lộ một cách trái phép

Các dịch vụ an ninh (X.800) (2)

- Toàn vẹn dữ liệu
  - Đảm bảo dữ liệu nhận được đúng như khi gửi bởi một thực thể được phép
- Chống chối bỏ
  - Bảo vệ khỏi sự chối bỏ bởi một trong các thực thể tham gia truyền thông
- Khả dụng
  - Đảm bảo tài nguyên có thể truy nhập và sử dụng được

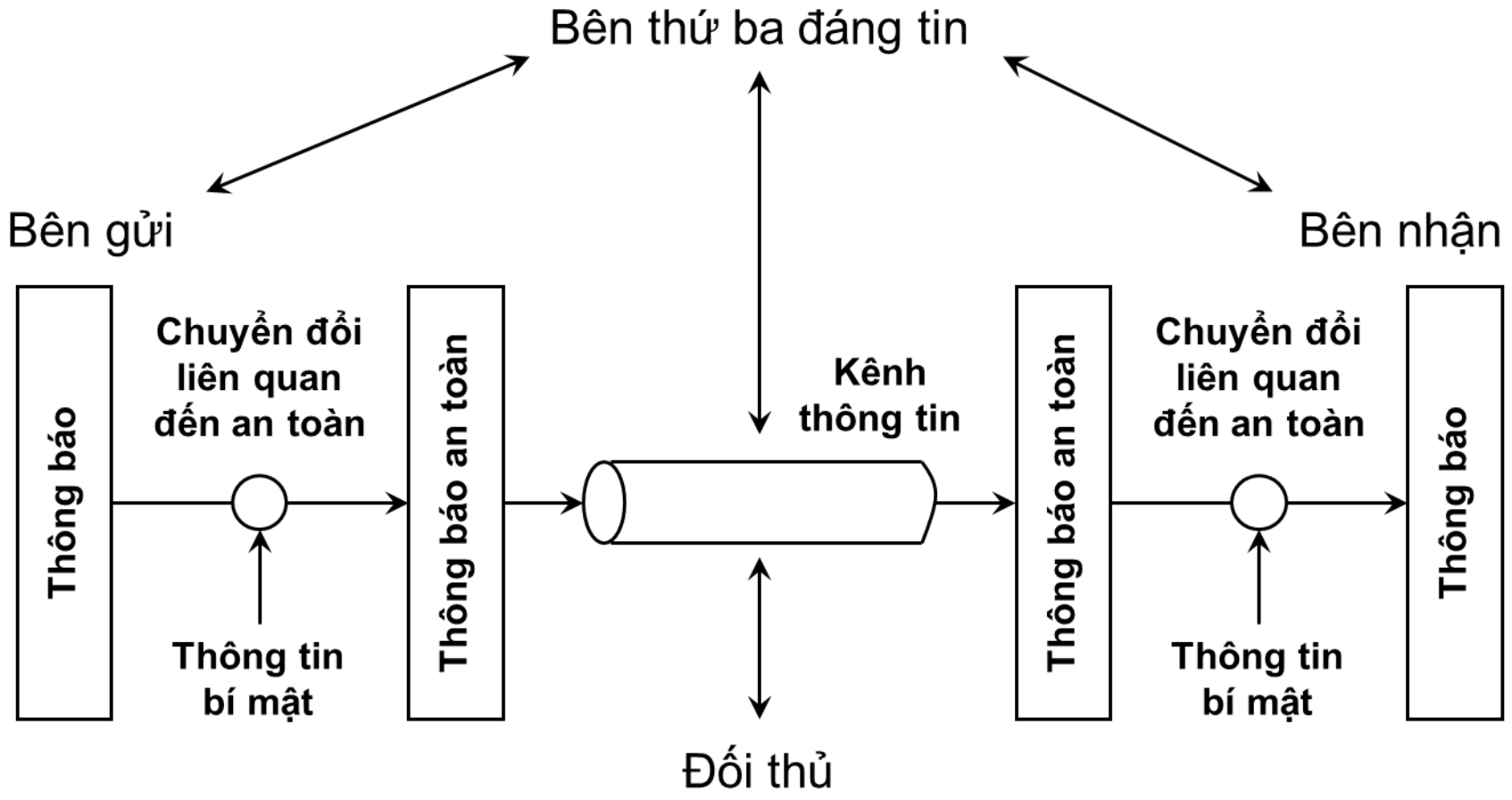
Cơ chế an ninh

- Một dịch vụ an ninh sử dụng một hoặc nhiều cơ chế an ninh
- Không có một cơ chế đơn lẻ nào hỗ trợ tất cả các dịch vụ an ninh
- Một yếu tố đặc biệt hậu thuẫn nhiều cơ chế an ninh đang được sử dụng
  - Các kỹ thuật mật mã học

Các cơ chế an ninh (X.800)

- Các cơ chế an ninh chuyên biệt
  - Được cài đặt ở một tầng giao thức chuyên biệt
  - Mã hóa, chữ ký số, điều khiển truy nhập, toàn vẹn dữ liệu, trao đổi xác thực, độn lưu lượng, điều khiển định tuyến, công chứng
- Các cơ chế an ninh phổ quát
  - Không chuyên biệt cho bất kỳ dịch vụ an ninh hay tầng giao thức đặc biệt nào
  - Tính năng đáng tin, nhãn an ninh, phát hiện sự kiện, dấu vết kiểm nghiệm an ninh, khôi phục an ninh

Mô hình an toàn mạng



Nhiệm vụ mô hình an toàn mạng

- Thiết kế giải thuật thực hiện chuyển đổi liên quan đến an toàn
- Sinh thông tin bí mật để dùng với giải thuật
- Phát triển các phương pháp phân phối và chia sẻ thông tin bí mật
- Đặc tả một giao thức cho phép các chủ thể sử dụng giải thuật an ninh và thông tin bí mật cho một dịch vụ an ninh

Định nghĩa mật mã học

- Để hiểu mật mã học cần biết
  - Nó là gì
  - Nó có thể làm gì
  - Nó có thể được sử dụng như một công cụ an ninh để bảo vệ dữ liệu như thế nào
- Định nghĩa
  - Khoa học chuyển đổi thông tin sang một dạng không thể hiểu được khi nó được truyền hay lưu trữ để những người dùng không được phép không thể truy nhập được

Mật mã học và an ninh

- Mật mã học có thể bảo vệ an toàn thông tin ở mức cơ bản
  - Có thể bảo vệ tính bảo mật của thông tin thông qua đảm bảo chỉ những bên được phép mới có thể xem được
  - Có thể bảo vệ tính toàn vẹn của thông tin
  - Giúp đảm bảo tính khả dụng của dữ liệu để những người dùng được phép (với khóa) truy nhập được
  - Có thể xác minh tính xác thực của bên gửi
  - Có thể buộc chống chối bỏ

Các giải thuật mật mã học

- Các giải thuật đối xứng
  - Sử dụng cùng một khóa để mã hóa và giải mã thông báo
- Các giải thuật bất đối xứng (khóa công khai)
  - Sử dụng hai thay vì một khóa
- Các giải thuật băm
  - Tạo ra một "chữ ký" duy nhất đại diện cho nội dung của một tập dữ liệu

Tổng kết

- Bối cảnh
- Các định nghĩa, khái niệm và thuật ngữ về an toàn thông tin
- Các thách thức đối với an toàn máy tính
- Kiến trúc an ninh X.800
  - Tấn công an ninh, dịch vụ an ninh, cơ chế an ninh
- Mô hình an toàn mạng
- Tổng quan về mật mã học

Chương 2

MÃ HÓA ĐỐI XỨNG VÀ BẢO MẬT THÔNG BÁO

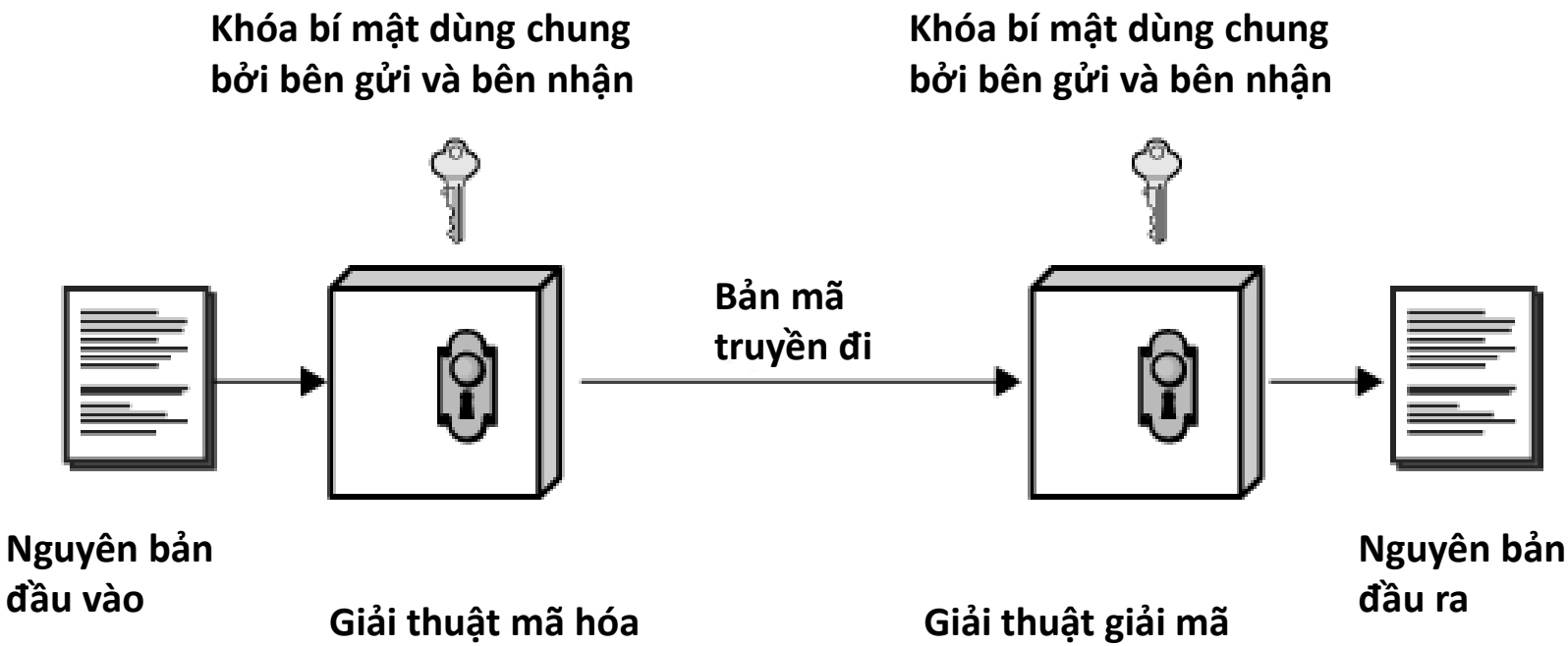
Mã hóa đối xứng

- Còn gọi là mã hóa truyền thống, khóa bí mật, khóa riêng, hay khóa đơn
- Bên gửi và bên nhận sử dụng một khóa chung
- Tất cả mã hóa từ thời cổ đại đến năm 1976 chỉ dựa trên các phương pháp đối xứng
- Được sử dụng rộng rãi nhất cho đến nay

Một số thuật ngữ cơ bản

- Nguyên bản
- Bản mã
- Phép mã hóa
  - Chuyển đổi từ nguyên bản thành bản mã
- Phép giải mã
  - Khôi phục nguyên bản từ bản mã
- Hệ mã hóa
  - Một phương thức được sử dụng cho mã hóa

Mô hình mã hóa đối xứng



Mã hóa

$Y = E_K(X)$

Giải mã

$X = D_K(Y)$

Các yêu cầu đặt ra

- Một giải thuật mã hóa mạnh
  - Không cần giữ bí mật giải thuật mã hóa
    - Có thể sử dụng rộng rãi
  - Địch thủ có thể biết một số bản mã cùng với các nguyên bản tương ứng
- Chỉ bên gửi và bên nhận biết khóa bí mật
  - An ninh chủ yếu nằm ở vấn đề duy trì tính bí mật của khóa

Phân loại mật mã học

- Phân loại theo 3 tiêu chí độc lập với nhau
  - Kiểu thao tác mã hóa được sử dụng
    - Thay thế, hoán vị, tích hợp
  - Số khóa được sử dụng
    - Đơn khóa
    - Hai khóa
  - Cách thức xử lý nguyên bản
    - Khối
    - Luồng

Thám mã

- Nỗ lực phá vỡ các hệ mật mã
- Lý do thám mã là cần thiết
  - Không tồn tại chứng minh toán học về tính an toàn cho bất kỳ hệ mật mã thực tế nào
  - Cách duy nhất đảm bảo tính an toàn của một hệ mật mã là thử phá vỡ nó (và thất bại)
- Chỉ sử dụng các hệ mật mã được biết đến một cách rộng rãi đã được thám mã trong nhiều năm bởi những nhà mật mã học giỏi

Các phương pháp thám mã

- Thám mã cổ điển
  - Khoa học phát hiện nguyên bản hoặc khóa
  - Các tấn công thám mã
    - Khai thác cấu trúc bên trong của phương pháp mã hóa
  - Các tấn công vét cạn
    - Coi giải thuật mã hóa như một hộp đen và thử tất cả các khóa có thể
- Các tấn công cài đặt
- Các tấn công kỹ nghệ xã hội

An toàn của các hệ mật mã học

- Tính an toàn tính toán
  - Chi phí phá vỡ hệ mã hóa vượt quá giá trị của thông tin được mã hóa
  - Thời gian cần thiết để phá vỡ hệ mã hóa vượt quá tuổi thọ hữu ích của thông tin
- Nếu giải thuật không có yếu điểm toán học nội tại nào thì phương pháp tìm kiếm vét cạn có thể được sử dụng để ước lượng chi phí và thời gian

Tìm kiếm vét cạn

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

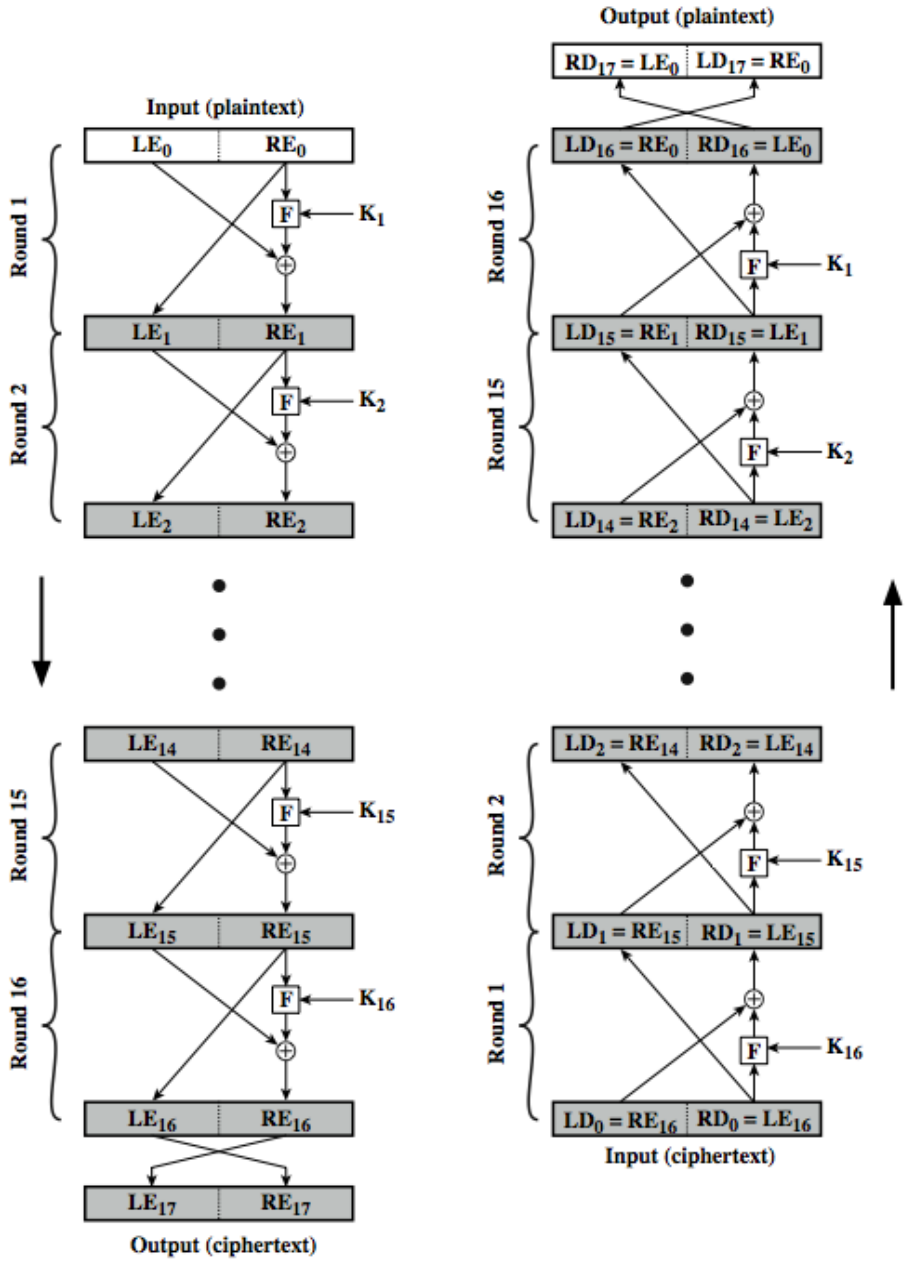
Khóa DES dài 56 bit
Khóa AES dài 128+ bit
Khóa 3DES dài 168 bit

Tuổi vũ trụ : ~ 10^{10} năm

Cấu trúc hệ mã hóa Feistel

- Được mô tả đầu tiên bởi Horst Feistel của công ty IBM vào năm 1973
- Quá trình mã hóa
  - Khối nguyên bản được chia thành 2 nửa để rồi qua nhiều vòng xử lý
    - Một phép thay thế nửa trái bằng cách áp dụng một hàm vòng vào nửa phải cùng với một khóa con rồi thực hiện XOR đầu ra với nửa trái
    - Một phép hoán vị với việc đổi hai nửa cho nhau
- Cài đặt khái niệm mạng S-P của Shannon

Mã hóa và giải mã Feistel



Các yếu tố thiết kế Feistel

- Độ dài khối
- Độ dài khóa
- Số vòng
- Giải thuật sinh khóa con
- Hàm vòng
- Mã hóa/giải mã phần mềm nhanh
- Dễ phân tích

DES (Data Encryption Standard)

- Giải thuật mã hóa được sử dụng rộng rãi nhất
- Chuẩn FIPS 46 do NIST ban hành năm 1977
- Nguyên bản 64 bit và khóa 56 bit
  - Các nguyên bản dài hơn được xử lý theo các khối 64 bit
- Một biến thể nhỏ của mạng Feistel
  - 16 vòng với 16 khóa con, 1 khóa con cho mỗi vòng
  - Phép giải mã hầu như giống hệt phép mã hóa với các khóa con được sử dụng theo thứ tự ngược lại

Độ an toàn của DES

- Hai quan ngại
  - Khả năng khai thác các đặc tính của giải thuật DES
    - Nhiều nỗ lực phá mã không thành công
  - Độ dài khóa
    - Mất hơn một nghìn năm để phá mã với một máy tính có tốc độ thực hiện 1 phép mã hóa DES /μs
    - Tháng 7/1998, EFF tuyên bố đã phá mã DES với một máy tính trị giá 250000 đô la trong chưa đến 3 ngày
    - Với khóa 128 bit, DES không thể phá được

3DES

- Chuẩn hóa lần đầu trong ANSI X9.17 năm 1985
- Một phần của DES trong FIPS 46-3 năm 1999
- Sử dụng 3 khóa và 3 lần thực hiện DES
  - $C = E(K_3, D(K_2, E(K_1, P)))$
  - Có thể sử dụng 2 khóa: $C = E(K_1, D(K_2, E(K_1, P)))$
  - Trở thành DES đơn với 1 khóa
- Vì sao không dùng 2DES?
  - Tấn công gặp nhau ở giữa với $O(2^{56})$ bước

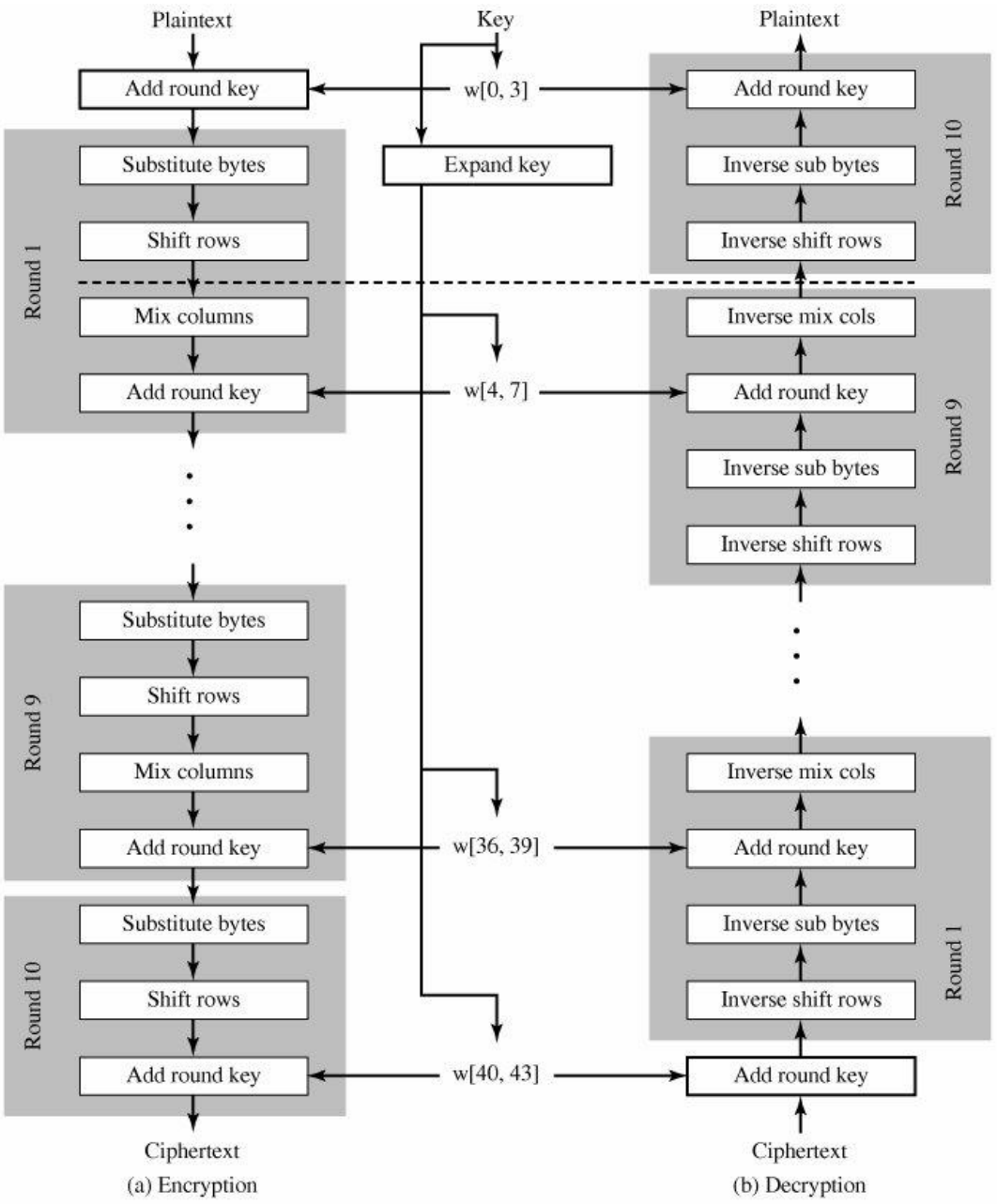
Xuất xứ của AES

- Nhược điểm của 3DES
  - Tương đối chậm về phần mềm
  - Sử dụng kích thước khối 64 bit
- Năm 1997 NIST kêu gọi đề xuất một chuẩn mới AES (Advanced Encryption Standard)
- 15 đề xuất qua vòng 1, 5 giải thuật qua vòng 2
- Rijndael được chọn làm AES tháng 10/2000
- Xuất bản trong FIPS PUB 197 tháng 11/2001

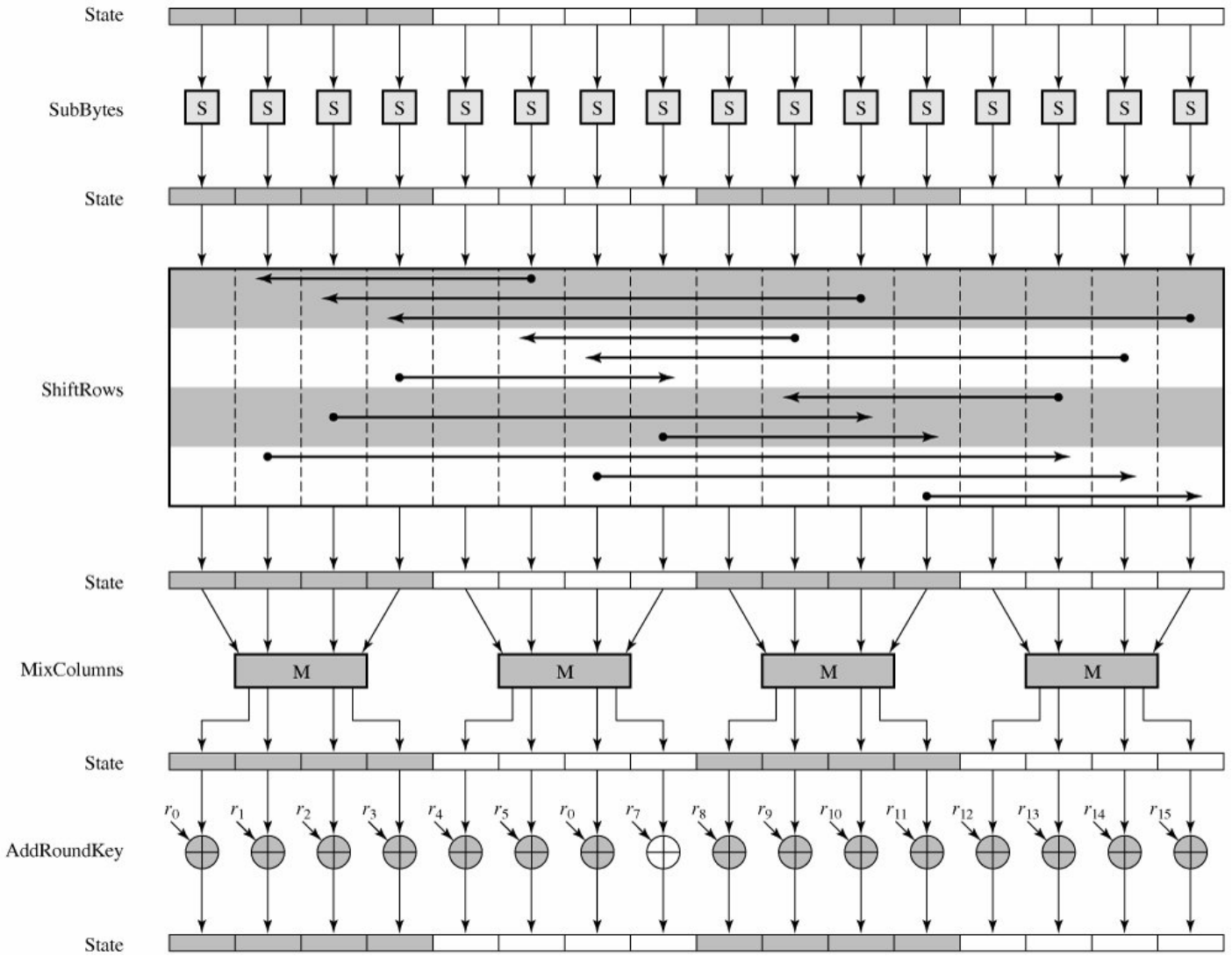
Tổng quan về AES

- Rijndael được phát triển bởi 2 nhà mật mã học người Bỉ là Rijmen và Daemen
- Sử dụng các khối 128 bit và các khóa 128/192/256 bit
- Lưu ý
  - Không theo cấu trúc Feistel
    - Xử lý toàn bộ khối dữ liệu ở mỗi vòng
    - Các khối dữ liệu và khóa được coi như các ma trận vuông các byte với thứ tự theo cột

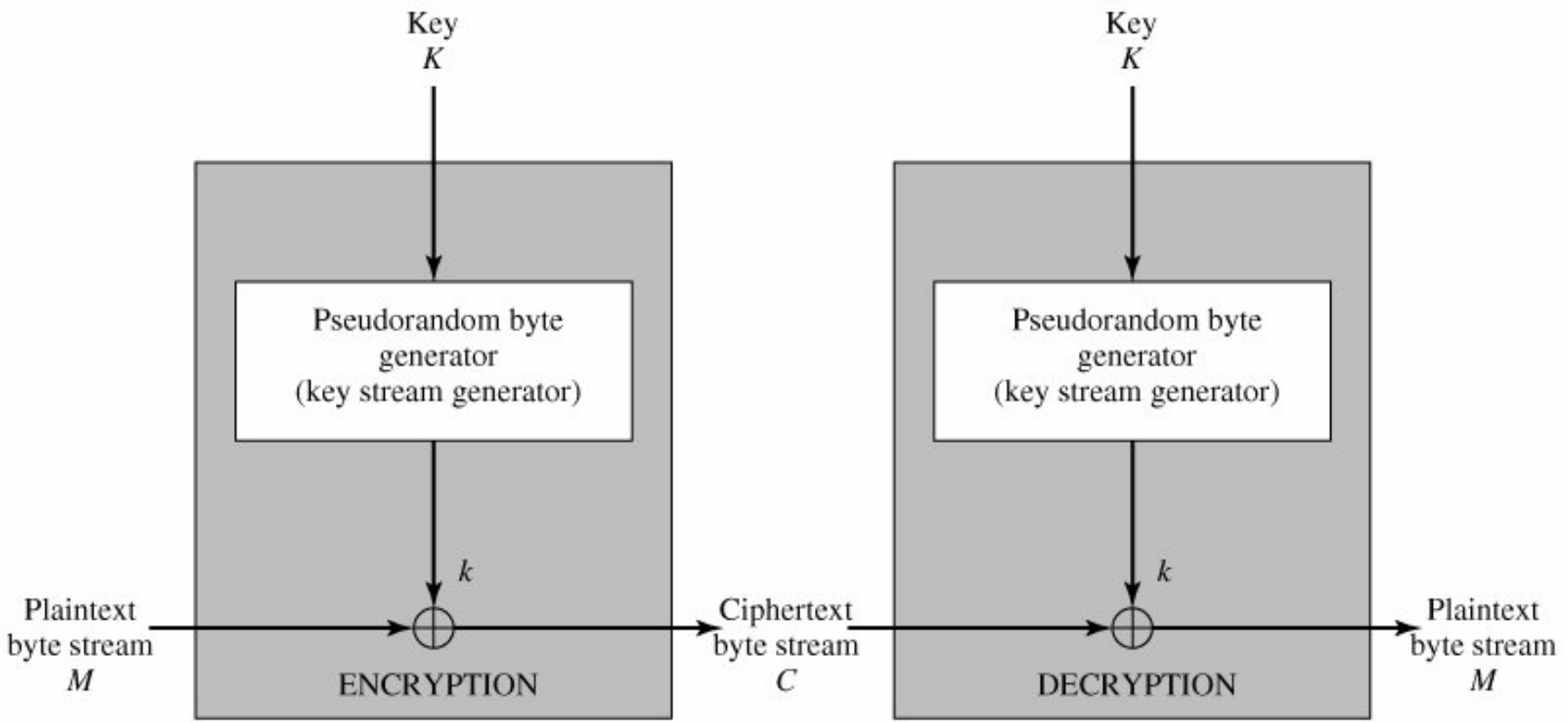
Mã hóa và giải mã AES



Vòng mã hóa AES



Cấu trúc các hệ mã hóa luồng



Tính chất các hệ mã hóa luồng

- Các tiêu chí thiết kế quan trọng
  - Chuỗi mã hóa có chu kỳ lớn
  - Luồng khóa có biểu hiện ngẫu nhiên
  - Độ dài khóa đủ lớn
    - Ít nhất là 128 bit
- Nếu được thiết kế hợp lý, có thể an toàn như hệ mã hóa khối có độ dài khóa tương tự
- Nhưng thường nhanh hơn và đơn giản hơn

RC4

- Thiết kế năm 1987 bởi Rivest cho RSA Security
- Hệ mã hóa luồng định hướng byte với kích thước khóa có thể thay đổi
- Được sử dụng trong SSL/TLS và WEP/WPA
- Rất đơn giản và tương đối dễ giải thích
- Dùng khóa khởi tạo vector trạng thái 256 byte
- Sinh byte *k* qua lựa chọn 1 trong 256 phần tử
  - Các phần tử được giao hoán ở mỗi lần sinh

Khởi tạo RC4

- Bắt đầu với một mảng S có 256 phần tử nhận các giá trị 0..255 theo thứ tự tăng dần
- Sử dụng khóa K để xáo trộn S

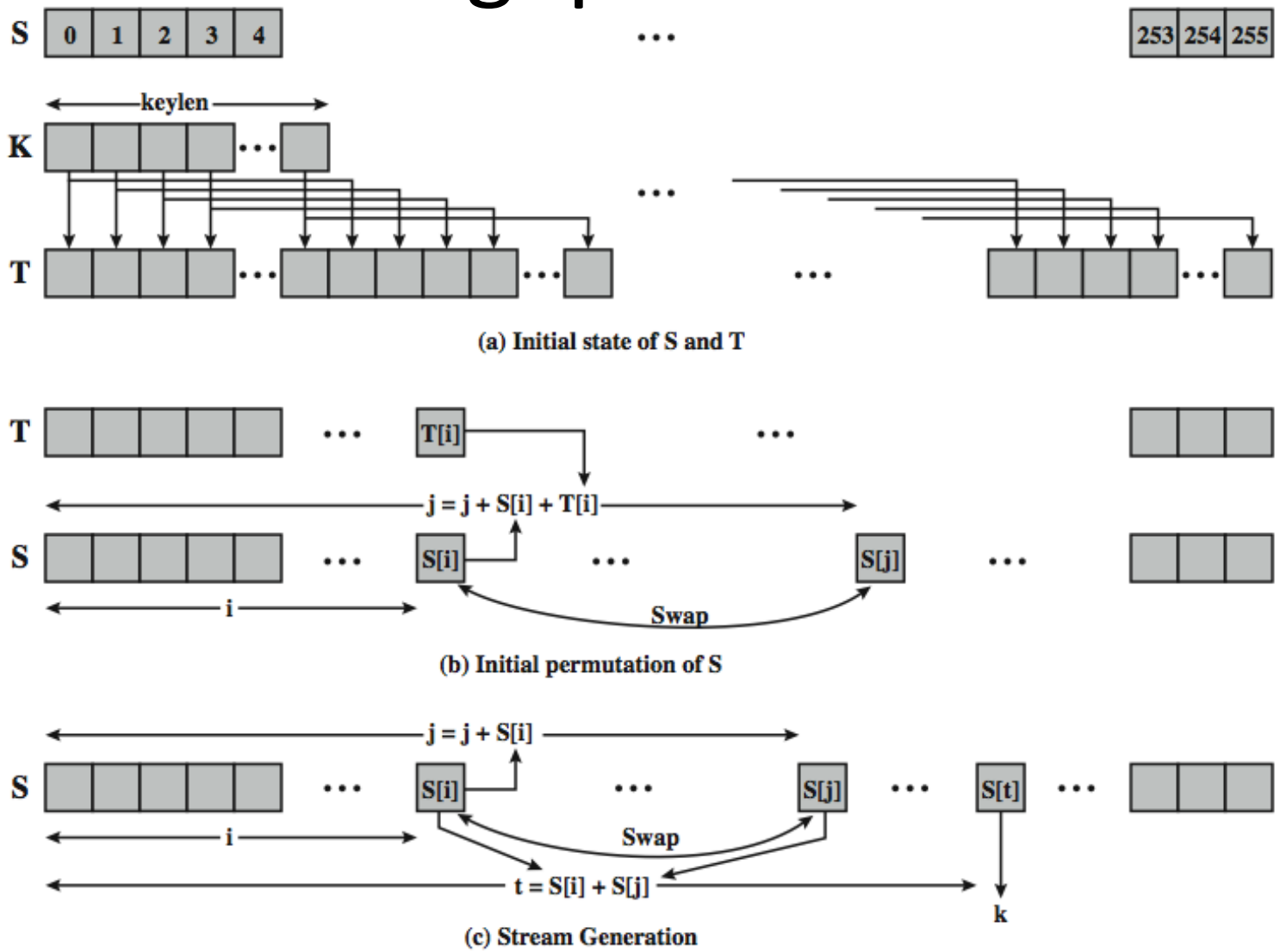
```
for i = 0 to 255 do
     S[i] = i
     T[i] = K[i mod keylen]
j = 0
for i = 0 to 255 do
     j = (j + S[i] + T[i]) mod 256
     swap(S[i], S[j])
```

Sinh luồng RC4

- Duyệt qua tất cả các phần tử của S
  - S tiếp tục được xáo trộn
  - Tổng của cặp xáo trộn chọn nên giá trị khóa luồng

```
i = j = 0
while (true)
        i = (i + 1) mod 256
        j = (j + S[i]) mod 256
        swap(S[i], S[j])
        t = (S[i] + S[j]) mod 256
        k = S[t]
```

Tổng quan RC4



Chế độ hoạt động

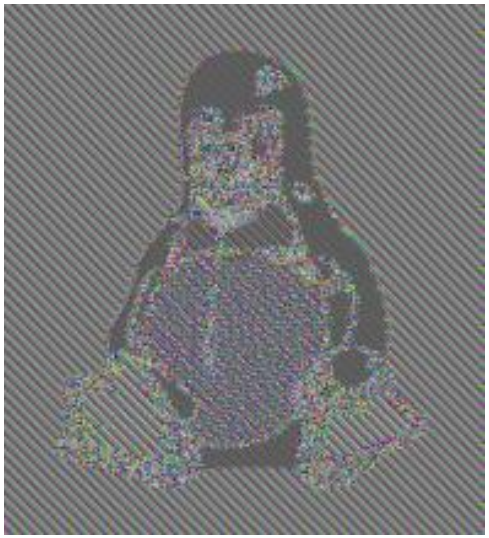
- Các hệ mã hóa khối xử lý các khối có kích thước cố định
  - Với DES và 3DES là 64 bit, AES là 128 bit
- Cần phân nguyên bản dài hơn thành các khối
  - Độn khối cuối cùng nếu cần thiết
- SP 800-38A của NIST định ra 5 chế độ
  - Đủ hỗ trợ gần như tất cả các ứng dụng có thể
  - Có thể dùng với bất kỳ hệ mã hóa khối nào
  - Có các chế độ mã hóa theo khối và theo luồng

ECB (Electronic Codebook)

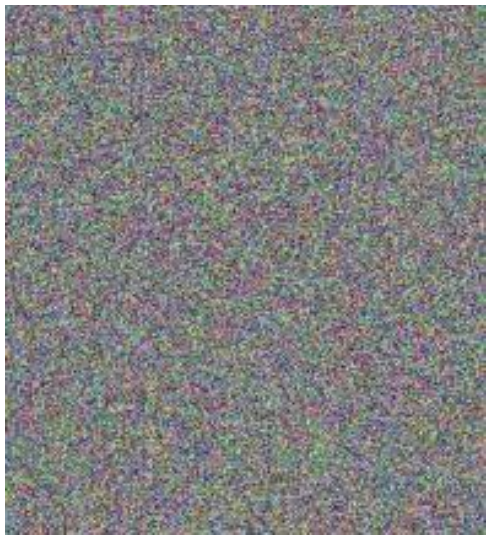
- Nguyên bản được xử lý theo từng khối một
- Mỗi khối được mã hóa một cách độc lập sử dụng cùng một khóa
  - Như một bảng mã trong đó mỗi khối nguyên bản ánh xạ tới đúng một khối bản mã
- Mỗi khi xuất hiện cùng một khối nguyên bản luôn tạo ra cùng một bản mã
  - Có thể không an toàn với các thông báo dài
    - Có tính cấu trúc cao hoặc trùng lặp theo độ dài khối

Ví dụ tính không an toàn của ECB

Nguyên bản



Mã hóa theo chế độ ECB

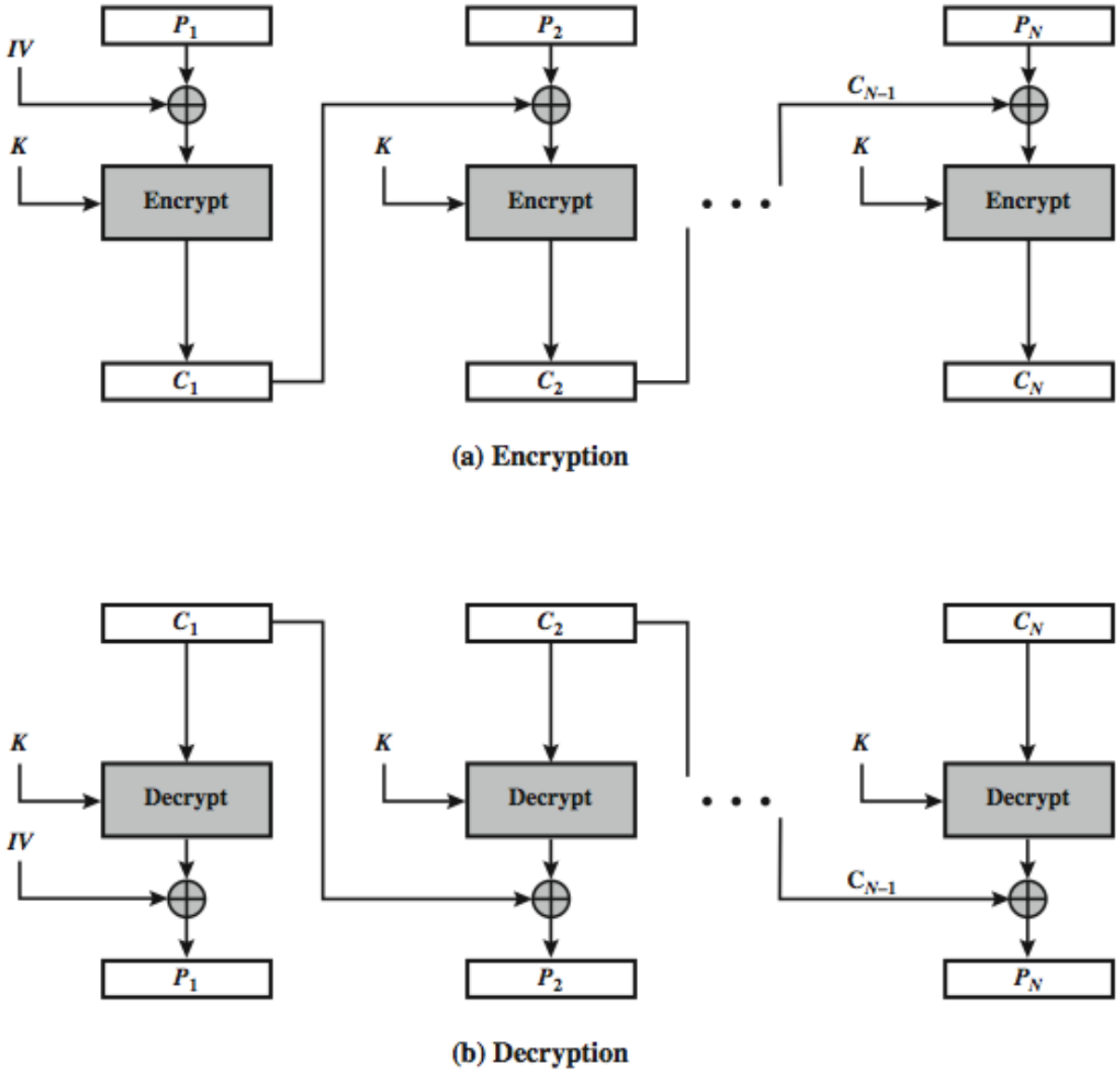


Các chế độ khác ECB

CBC (Cipher Block Chaining)

- Thông báo được phân thành các khối
- Đầu vào cho phép mã hóa là kết quả XOR khối nguyên bản hiện thời và khối bản mã trước đó
  - Mỗi khối bản mã trước được xâu chuỗi với khối nguyên bản hiện thời
  - Sử dụng IV (Initial Vector) để khởi tạo quá trình
- Không làm bộc lộ các mẫu khối trùng lặp
- IV cần được bảo vệ như khóa

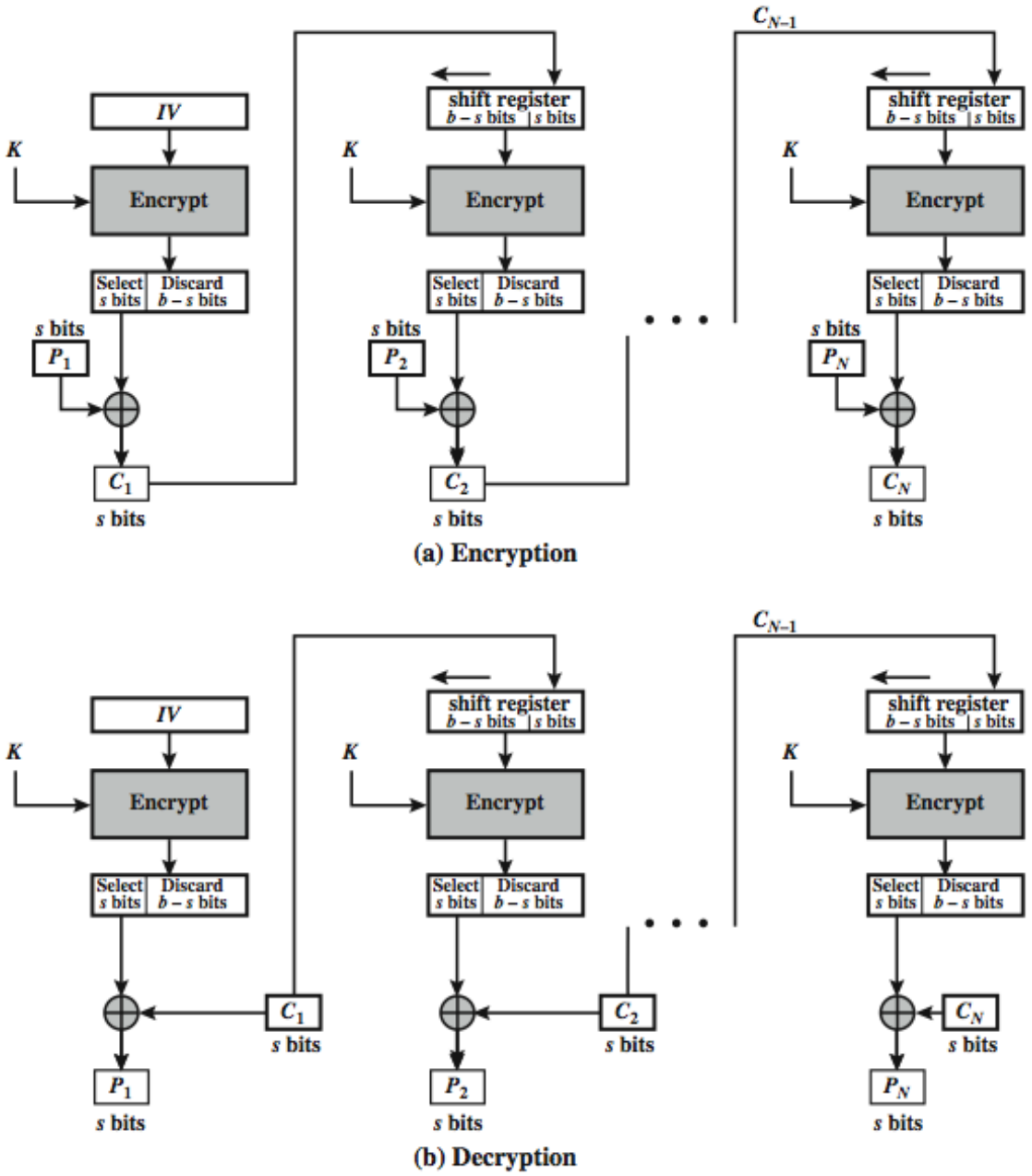
Mã hóa và giải mã CBC



CFB (Cipher Feedback)

- Chuyển đổi một hệ mã hóa khối thành luồng
  - Không cần độn thông báo
  - Có thể hoạt động theo thời gian thực
  - Bản mã có cùng độ dài như nguyên bản
- Đơn vị bên trái nhất ở đầu ra của phép mã hóa được XOR với đơn vị nguyên bản hiện thời tạo ra đơn vị bản mã hiện thời
  - Đơn vị bản mã được phản hồi vào giai đoạn sau
  - Đầu vào của phép mã hóa được khởi tạo bằng IV

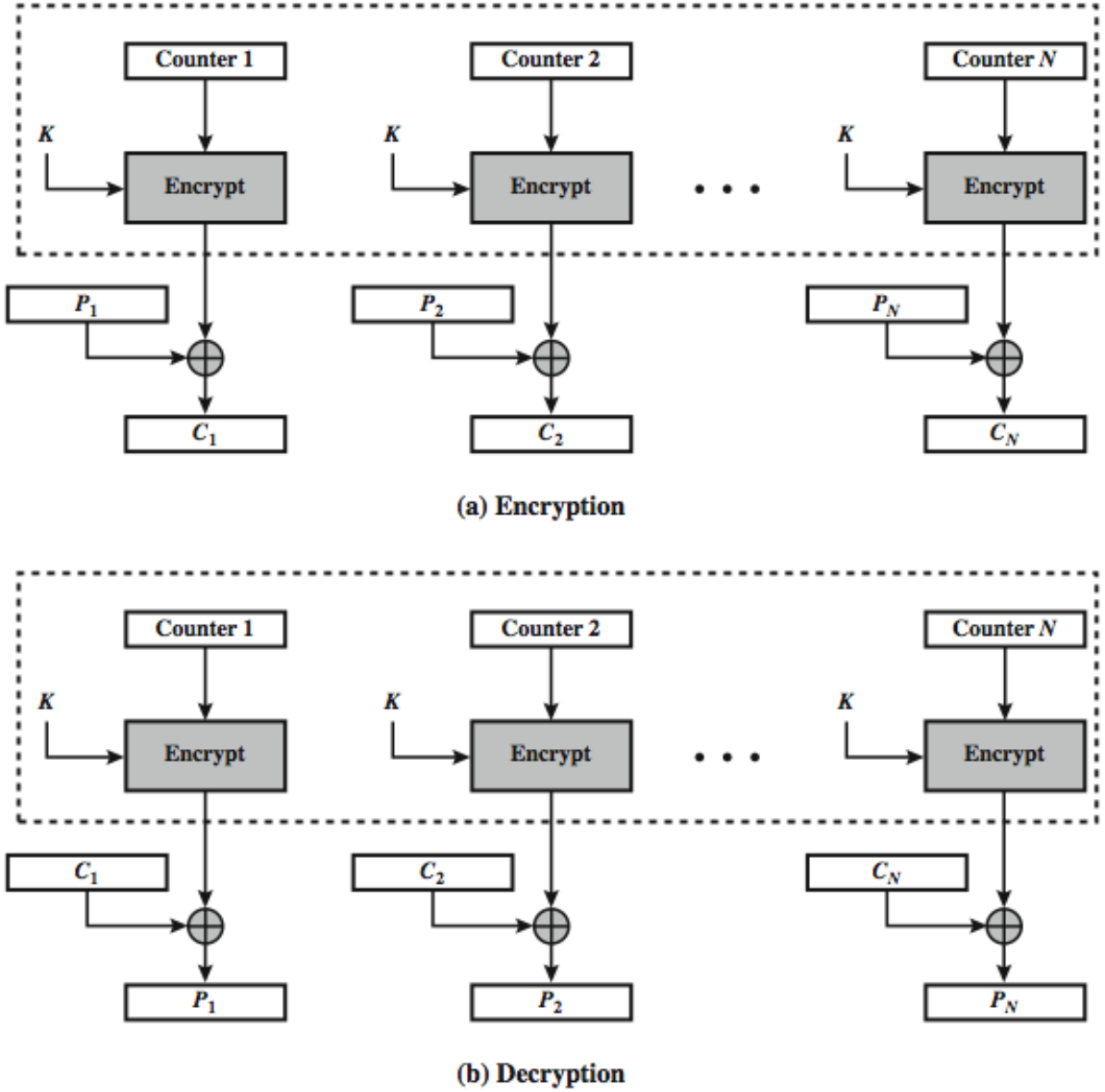
Mã hóa và giải mã CFB



CTR (Counter)

- Một chế độ mã hóa khối mới được quan tâm gần đây, mặc dù đã được đề xuất khá lâu
- Sử dụng một biến đếm có kích thước bằng khối nguyên bản
- Giá trị biến đếm phải khác nhau với mỗi khối nguyên bản
  - Thường được khởi tạo sau đó tăng 1 mỗi lần sau
- Biến đếm được mã hóa sau đó XOR với khối nguyên bản để tạo khối bản mã

Mã hóa và giải mã CTR



Ưu điểm của CTR

- Tính hiệu quả
  - Có thể mã hóa song song bằng phần cứng hoặc phần mềm
  - Có thể tiền xử lý trước khi cần
- Truy nhập ngẫu nhiên vào các khối dữ liệu
- Tính an toàn đã được chứng minh
  - Tốt như các chế độ khác
- Chỉ cần cài đặt phép mã hóa

Tổng kết

- Các nguyên tắc mã hóa đối xứng
  - Cấu trúc hệ mã hóa Feistel
- Các giải thuật mã hóa khối đối xứng
  - DES, 3DES, AES
- Mã hóa luồng và hệ mã hóa luồng RC4
- Các chế độ hoạt động của các giải thuật mã hóa khối
  - ECB, CBC, CFB, CTR

Chương 3

MẬT MÃ KHÓA CÔNG KHAI VÀ XÁC THỰC THÔNG BÁO

Xác thực thông báo

- Các yêu cầu đối với xác thực thông báo
  - Cho phép kiểm tra tính xác thực của thông báo
    - Có nguồn gốc đúng như tự nhận
    - Không bị thay đổi
  - Có thể cho phép kiểm tra trình tự và thời điểm
- Các hàm xác thực thông báo
  - Hàm băm
  - Mã hóa thông báo
  - Mã xác thực thông báo (MAC)

Mã hóa truyền thống

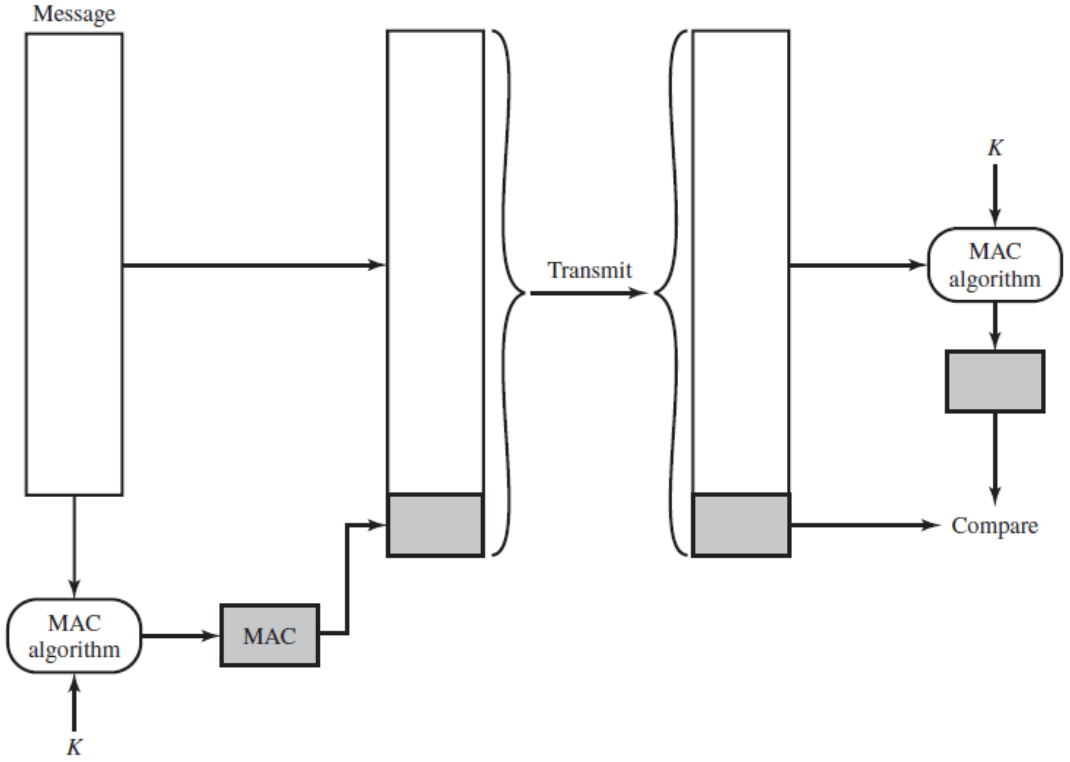
- Điều kiện
  - Bên nhận có thể nhận biết thông báo hợp lệ hoặc thông báo có chứa mã phát hiện lỗi
- Có thể kiểm tra tính xác thực của thông báo
  - Bên nhận biết chỉ có thể là bên gửi đã tạo ra nó
    - Chỉ bên gửi và bên nhận biết khóa được sử dụng
  - Bên nhận yên tâm không có thay đổi nào
- Có thể kiểm tra trình tự và thời điểm
  - Nếu thông báo chứa số thứ tự và nhãn thời gian

Không mã hóa thông báo

- Tạo thẻ xác thực gắn với mỗi thông báo
  - Bản thân thông báo không được mã hóa
- Những tình huống mong muốn xác thực thông cần bảo mật
  - Phát tỏa một thông báo đến nhiều đích với chỉ một đích chịu trách nhiệm giám sát tính xác thực
  - Xác thực một cách ngẫu nhiên để làm nhẹ tải
  - Xác thực chương trình ở dạng nguyên bản

Mã xác thực thông báo

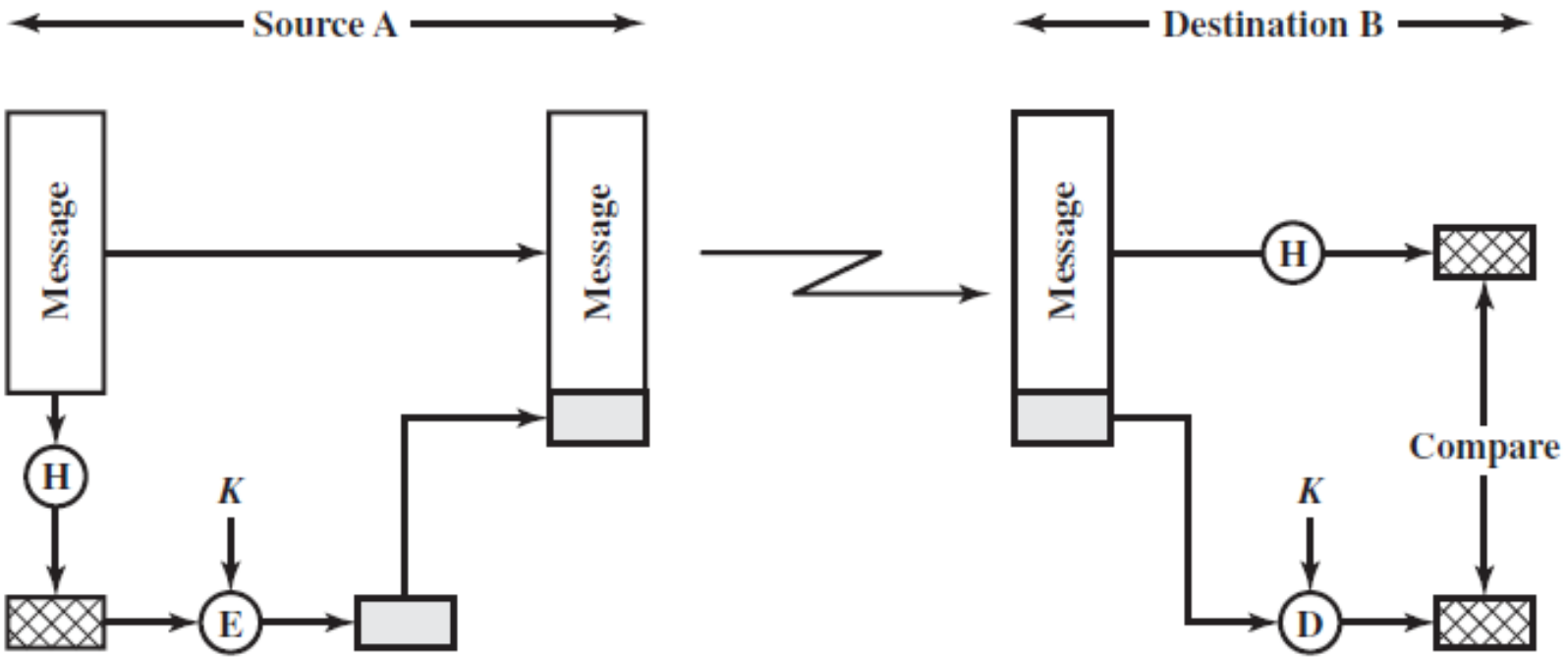
- Sử dụng một khóa bí mật để tạo một khối nhỏ kích thước cố định (MAC) gắn với thông báo
- Cung cấp chức năng xác thực thông báo



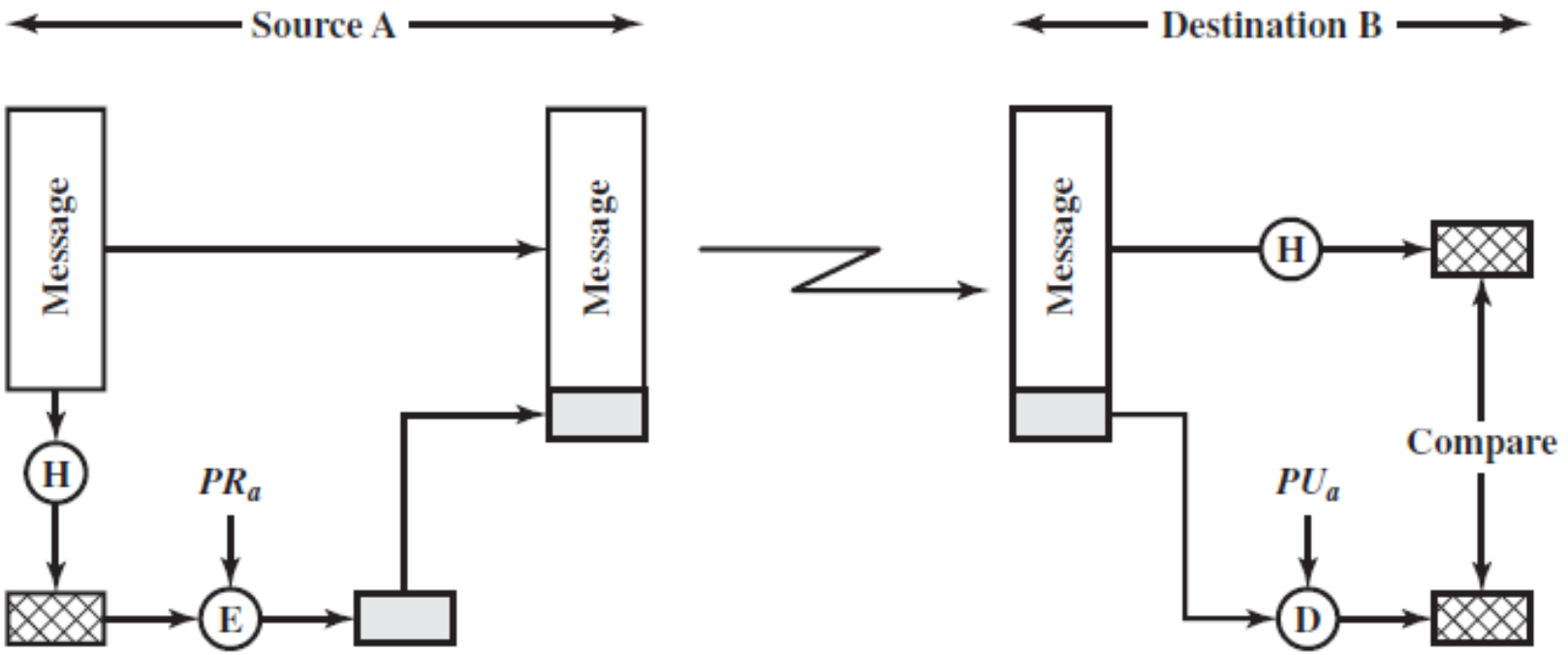
Hàm băm

- Cô đọng một thông báo có kích thước tùy ý thành một bản tóm lược có kích thước cố định
- Cung cấp chức năng xác thực thông báo nếu bản tóm lược được đảm bảo là xác thực
- Các phương thức xác thực thông báo
  - Sử dụng mã hóa đối xứng
  - Sử dụng mã hóa khóa công khai
  - Sử dụng giá trị bí mật

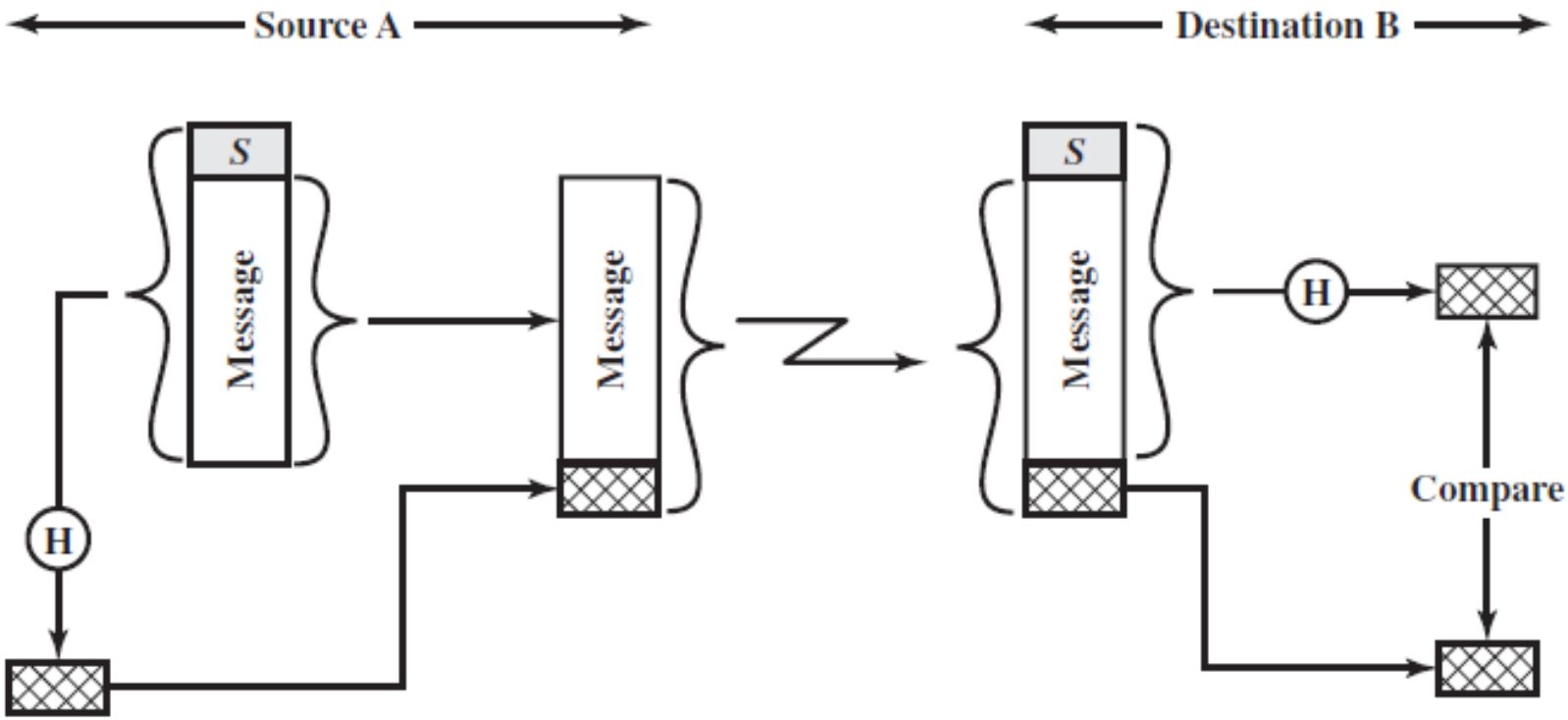
Sử dụng mã hóa đối xứng



Sử dụng mã hóa khóa công khai



Sử dụng giá trị bí mật



Các yêu cầu đối với hàm băm

- Các yêu cầu về an ninh đối với hàm băm H
  - Không thể tìm được *x* bằng tính toán sao cho H(*x*) = *h* với bất kỳ giá trị băm *h* cho trước nào
    - Tính một chiều hay chống tiền ảnh
  - Không thể tìm được $y \neq x$ bằng tính toán sao cho H(*y*) = H(*x*) với bất kỳ dữ liệu *x* cho trước nào
    - Tính chống tiền ảnh thứ hai hay chống xung đột yếu
  - Không thể tìm được cặp (*x*, *y*) bằng tính toán sao cho H(*x*) = H(*y*)
    - Tính chống xung đột hay chống xung đột mạnh

Độ an toàn của hàm băm

- Hai kiểu tấn công
  - Thám mã phân tích hoặc vét cạn
- Độ an toàn đối với các tấn công vét cạn chỉ phụ thuộc vào độ dài *n* của mã băm
  - Chống tiền ảnh: 2^n
  - Chống tiền ảnh thứ hai: 2^n
  - Chống xung đột: $2^{n/2}$
- Giá trị $2^{n/2}$ xác định sức mạnh của hàm băm
  - 128 bit là không đủ, 160 bit là không chắc chắn

SHA (Secure Hash Algorithm)

- Hàm băm được sử dụng rộng rãi nhất
- Phát triển bởi NIST và công bố thành chuẩn FIPS 180 (SHA-0) vào năm 1993
- Hiệu chỉnh thành SHA-1 năm 1995, ban hành thành FIPS 180-1 (SHS), có đặc tả trong RFC 3174
- SHA dựa trên thiết kế MD4
- SHA-1 sinh các giá trị băm 160 bit
- Phát hiện vấn đề về an toàn của SHA-1 năm 2005

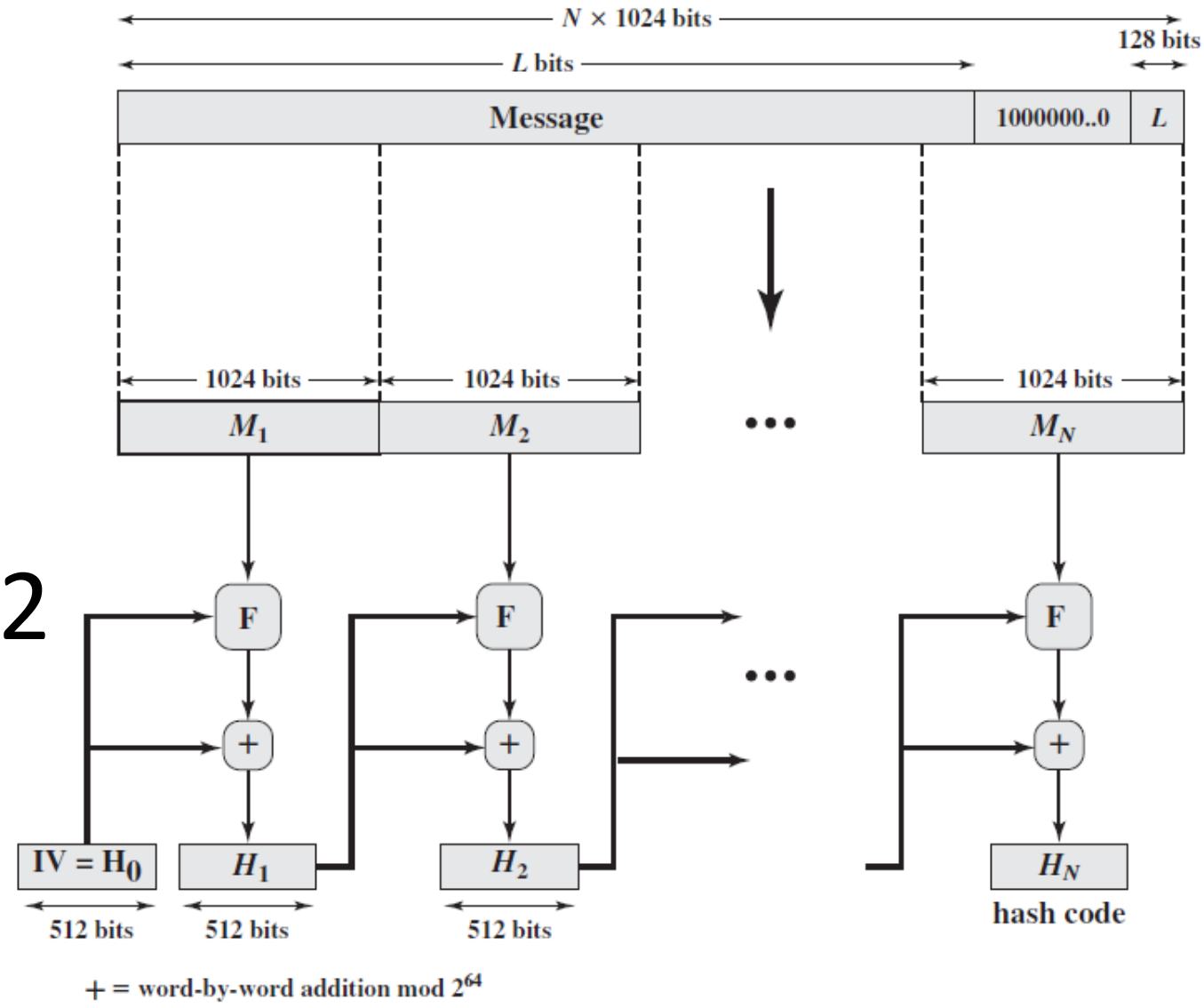
SHS hiệu chỉnh

- NIST đưa ra 1 bản hiệu chỉnh của SHS trong FIPS 180-2 vào năm 2002 với 3 phiên bản mới của SHA (gọi chung là SHA-2)
  - SHA-256, SHA-384, SHA-512
- Có cùng cấu trúc và kiểu các phép thao tác như SHA-1
- Một bản hiệu chỉnh nữa được ban hành thành FIPS 180-3 vào năm 2008 với một phiên bản 224 bit, cũng có đặc tả trong RFC 4634

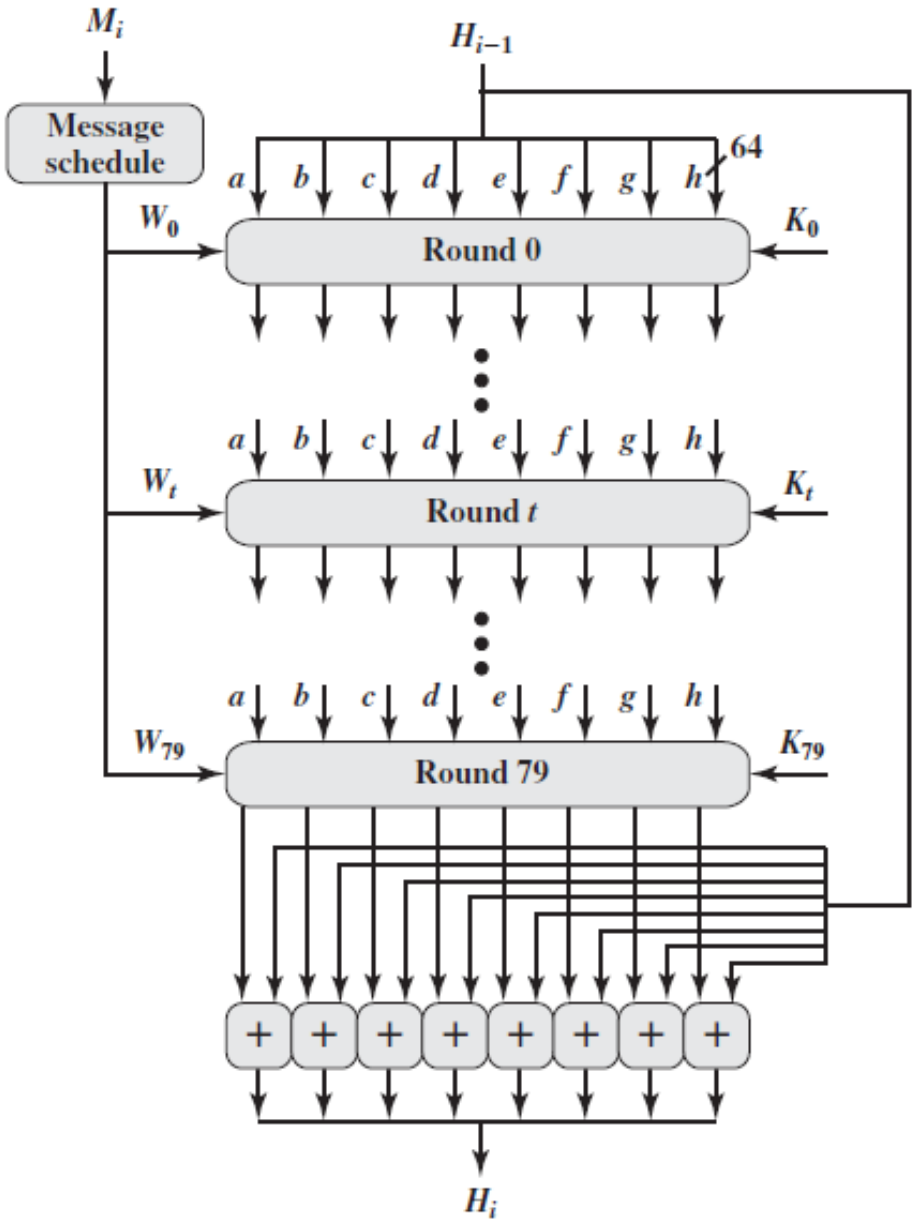
So sánh các thông số SHA

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message Digest Size	160	224	256	384	512
Message Size	$< 2^{64}$	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block Size	512	512	512	1024	1024
Word Size	32	32	32	64	64
Number of Steps	80	64	64	80	80
Security	80	112	128	192	256

Tổng quan SHA-512



Xử lý một khối 1024 bit của SHA-512



HMAC

- Ưu điểm của MAC dựa trên hàm băm
  - Hàm băm cài đặt bằng phần mềm nói chung thực hiện nhanh hơn mã hóa truyền thống
  - Mã thư viện cho hàm băm có sẵn phổ biến
- Nguyên tắc tích hợp khóa bí mật vào một giải thuật băm có sẵn
- Được hỗ trợ nhiều nhất, thành RFC 2104
- MAC phải cài đặt cho IPSec, được dùng trong các giao thức Internet khác (TLS, SET,…)

Mục tiêu thiết kế HMAC

- Sử dụng hàm băm không cần thay đổi
- Cho phép thay thế dễ dàng hàm băm đã được sử dụng
- Bảo toàn mà không làm suy biến đáng kể hiệu năng ban đầu của hàm băm
- Sử dụng và xử lý khóa một cách đơn giản
- Có phân tích mật mã học dễ hiểu về độ mạnh của cơ chế xác thực

Giải thuật HMAC

- Có thể biểu thị như sau

  $\text{HMAC}(K, M) = \text{H}[(K^+ \oplus \text{opad}) \,\|\, \text{H}[(K^+ \oplus \text{ipad}) \,\|\, M)]]$

  – H = hàm băm được sử dụng

  – M = thông báo đầu vào của HMAC

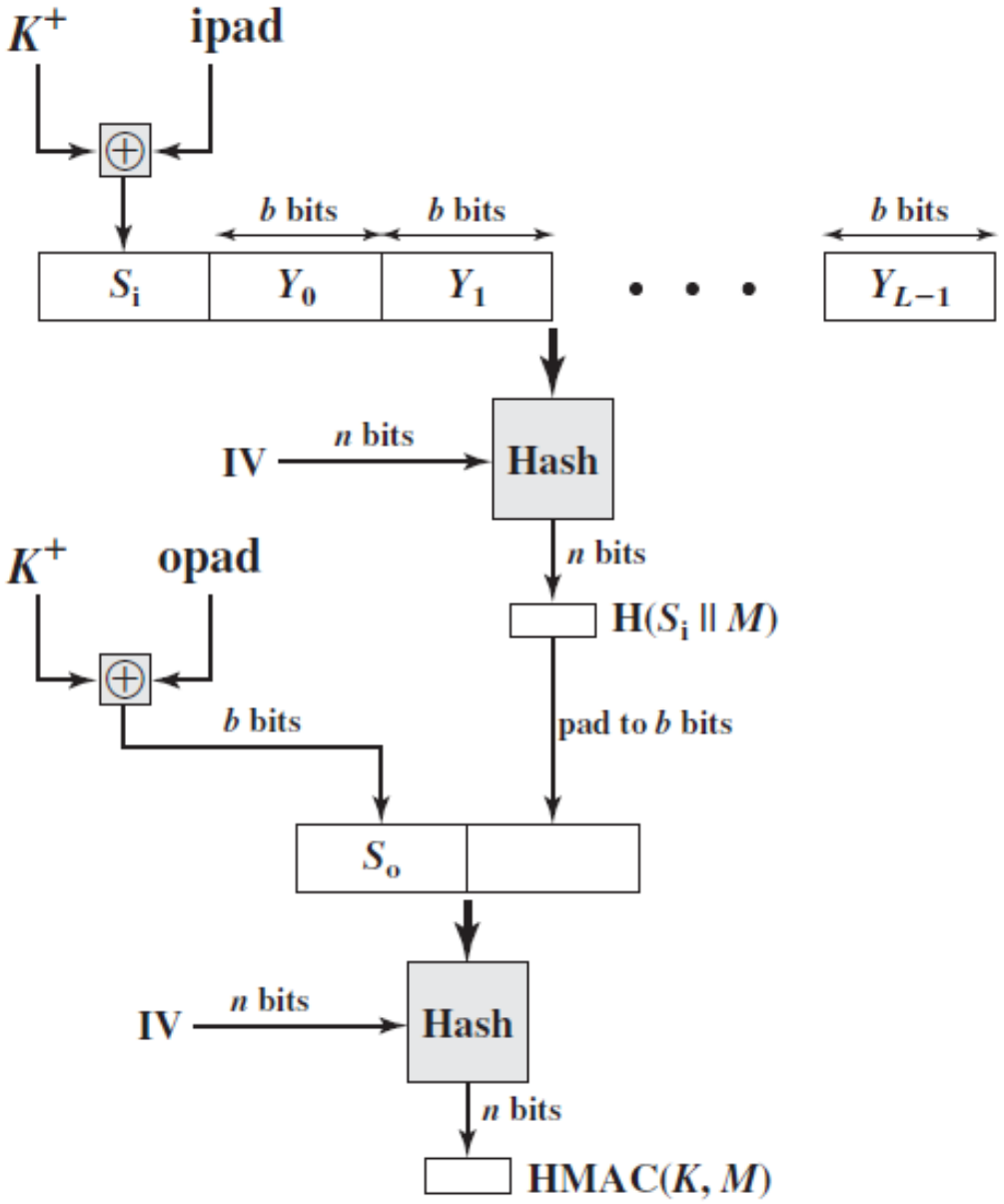
  – K = khóa bí mật

  – K^+ = K sau khi độn để tròn độ dài

  – opad, ipad = các hằng số độn định trước

- Thêm 3 lần thực hiện hàm băm cơ bản

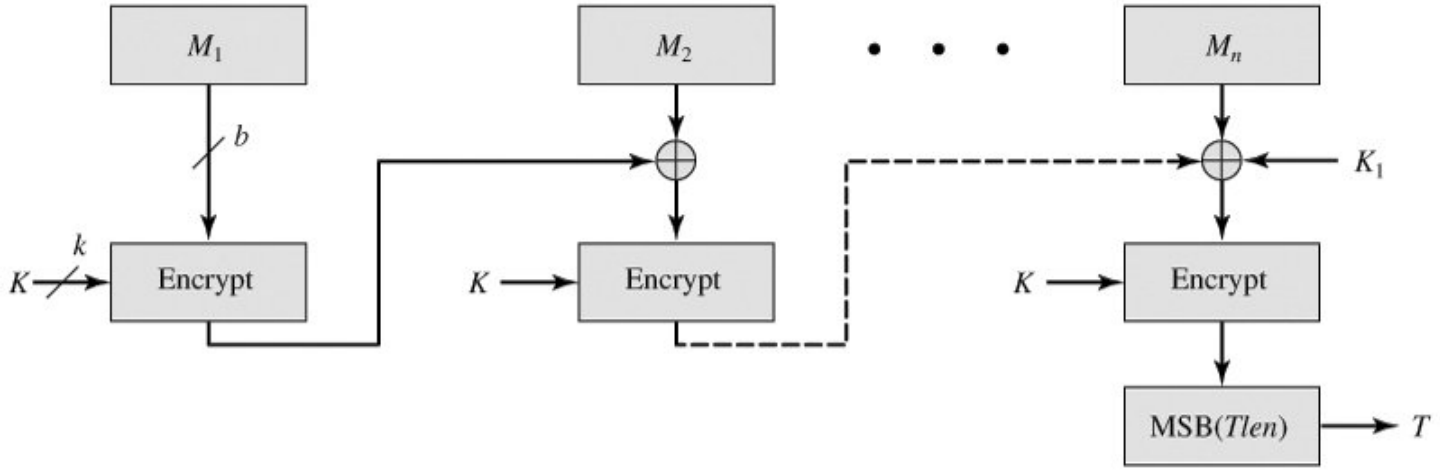
Cấu trúc HMAC



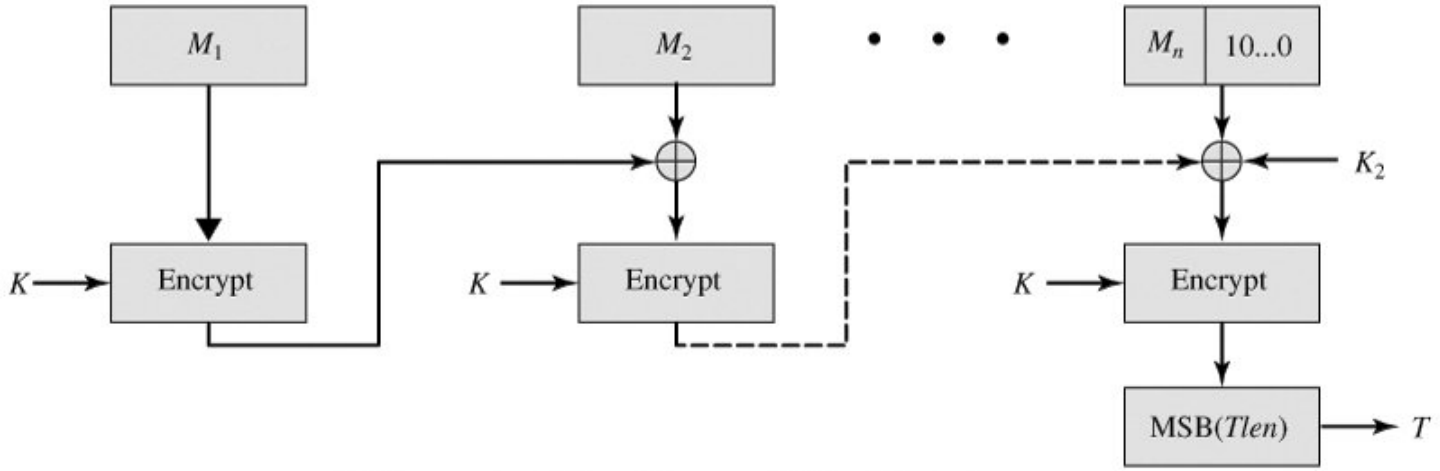
CMAC

- Cipher-based Message Authentication Code
- Để sử dụng với AES hoặc 3DES
- Đặc tả trong SP 800-38B của NIST
- Sử dụng chế độ hoạt động CBC với $IV = 0$
- Sử dụng 3 khóa
  - Một khóa K có độ dài k tại mỗi bước của CBC
  - Hai khóa K_1 và K_2 có độ dài n (độ dài khối bản mã) suy ra từ khóa mã hóa K

Cấu trúc CMAC



(a) Message length is integer multiple of block size

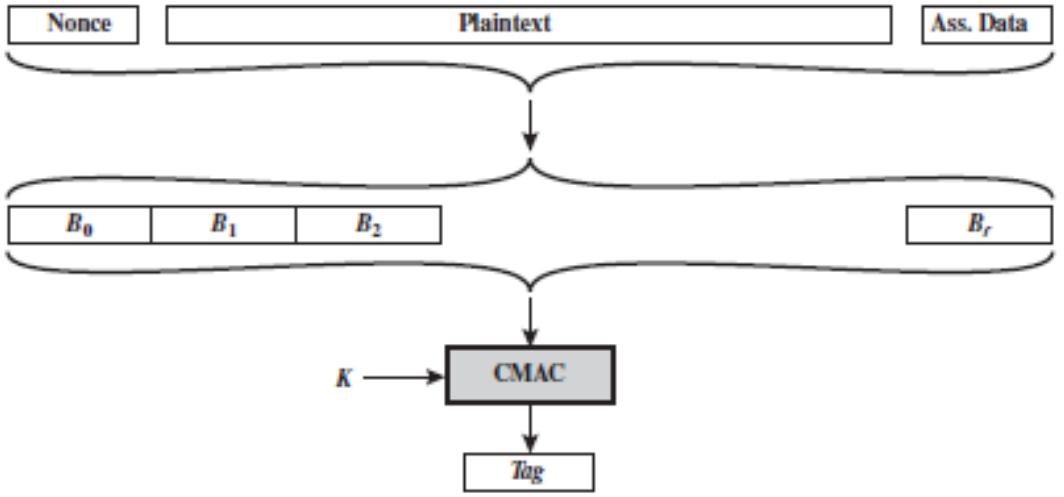


(b) Message length is not integer multiple of block size

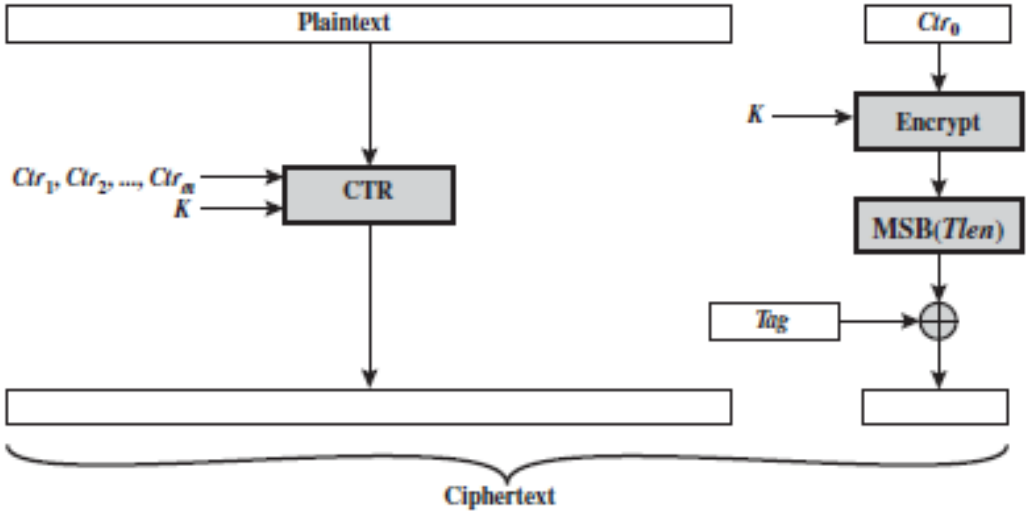
CCM

- Counter with CBC-MAC, SP 800-38C của NIST
- Một chế độ mã hóa được xác thực
  - Bảo vệ tính bảo mật và tính xác thực (toàn vẹn)
- Các thành phần giải thuật chủ chốt
  - Giải thuật mã hóa AES
  - Chế độ hoạt động CTR
  - Giải thuật xác thực CMAC
- Chỉ một khóa cho cả mã hóa và MAC

Hoạt động của CCM



(a) Authentication



(b) Encryption

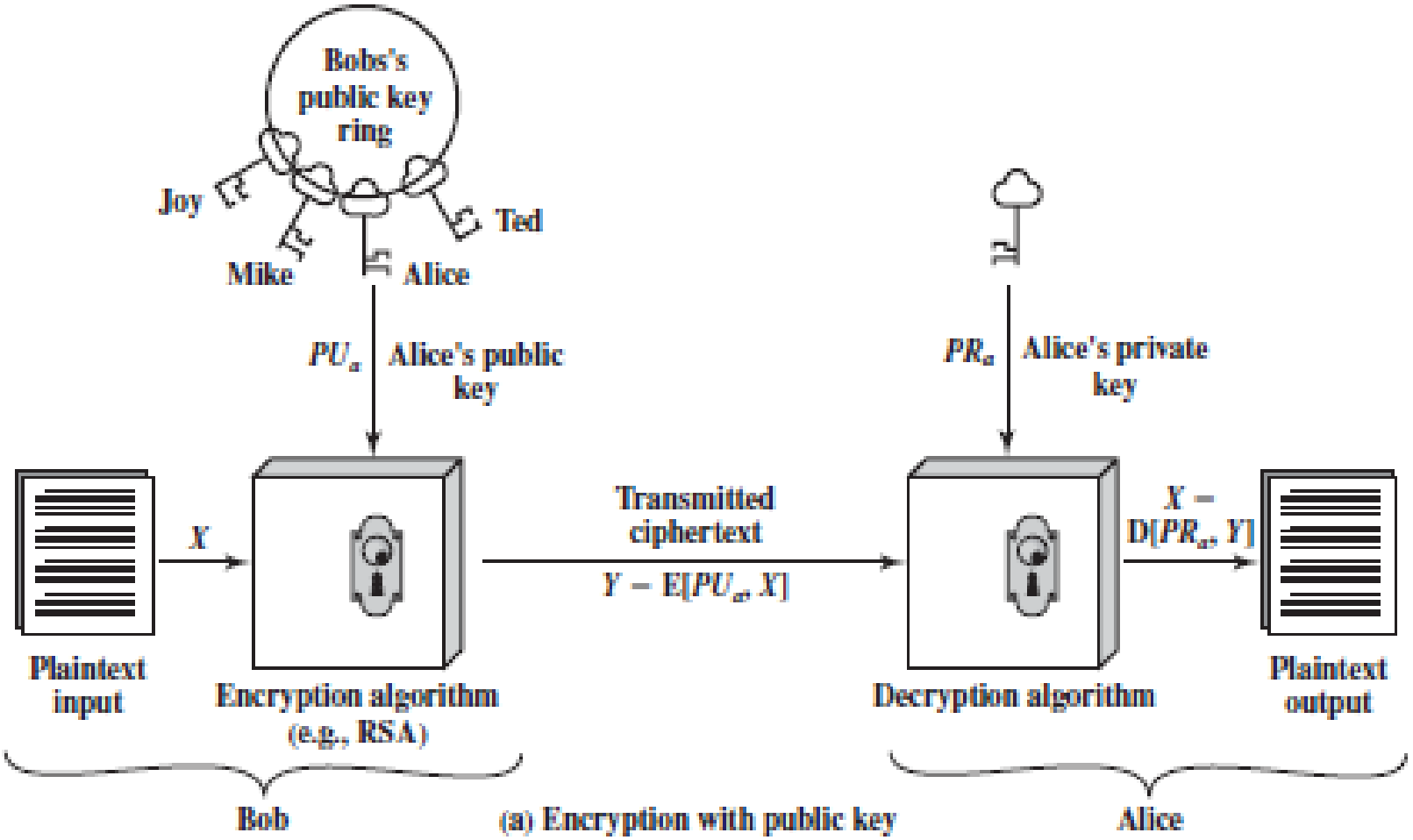
Mật mã khóa công khai

- Được đề xuất một cách công khai đầu tiên bởi Diffie và Hellman vào năm 1976
- Là cách mạng thực sự duy nhất trong lịch sử mật mã học
- Dựa trên các hàm toán học thay vì các phép thay thế và hoán vị
- Có tính bất đối xứng, sử dụng 2 khóa
  - Ảnh hưởng sâu sắc đến tính bảo mật, chức năng phân phối khóa, và chức năng xác thực

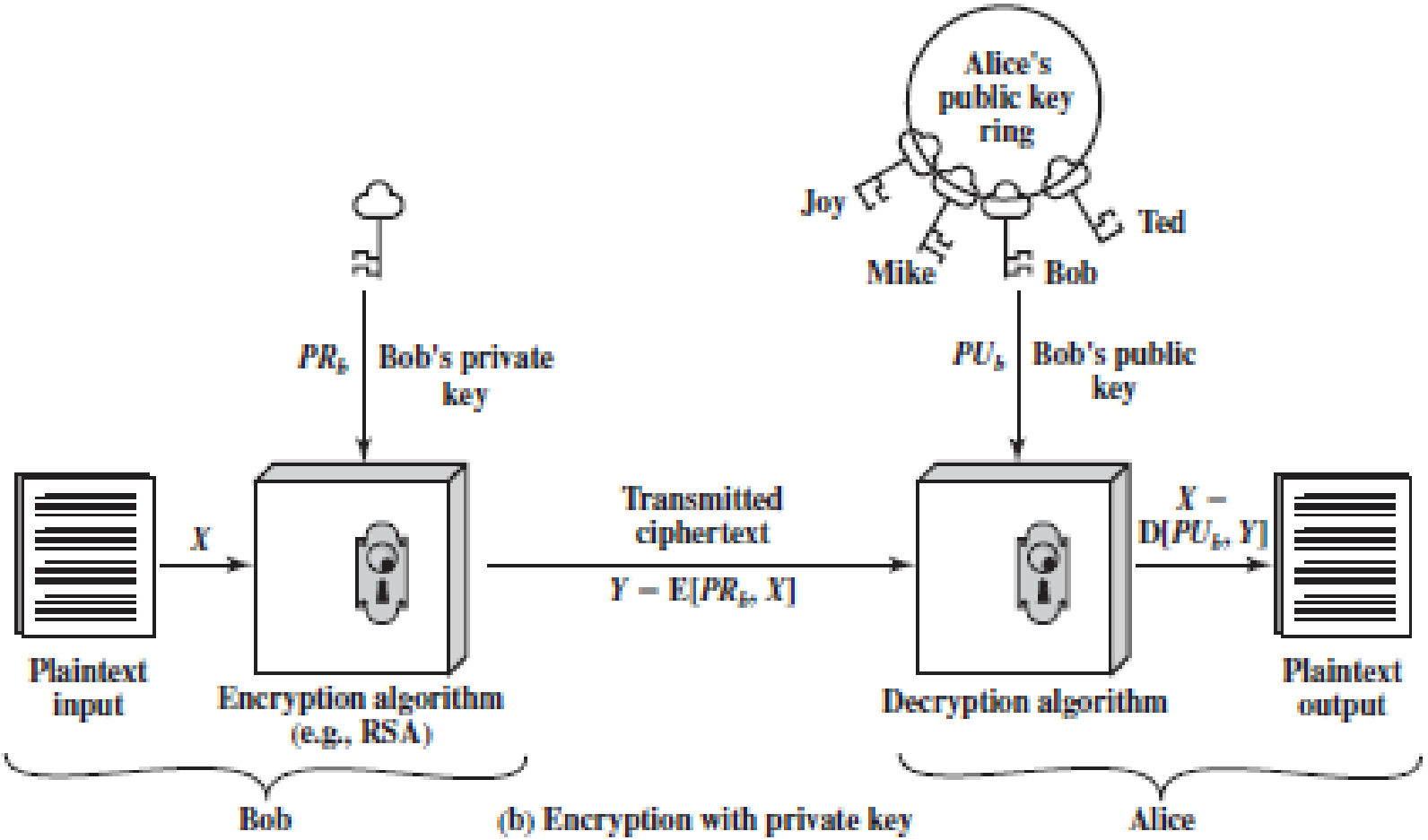
Những quan niệm sai phổ biến

- Mã hóa khóa công khai an toàn hơn
  - Độ an toàn phụ thuộc vào độ dài khóa và khối lượng tính toán cần cho thám mã
- Mã hóa khóa công khai có tính đa dụng
  - Do phụ tải tính toán lớn, nó không thể thay thế hoàn toàn mã hóa truyền thống
- Phân khối khóa công khai không là vấn đề gì
  - Không đơn giản hơn hay hiệu quả hơn phân khối khóa bí mật

Cấu trúc mã hóa khóa công khai (1)



Cấu trúc mã hóa khóa công khai (2)



Các ứng dụng khóa công khai

- Sử dụng 2 khóa, khóa riêng được giữ riêng và khóa công khai được công bố công khai
- Phân thành 3 loại
  - Mã hóa/giải mã
    - Bên gửi mã hóa thông báo với khóa công khai của bên nhận
  - Chữ ký số
    - Bên gửi "ký" thông báo với khóa riêng của mình
  - Trao đổi khóa
    - Hai bên hợp tác trao đổi một khóa phiên

Các yêu cầu khóa công khai

- Đặt ra bởi Diffie và Hellman
- Các yêu cầu thực tế
  - Dễ tính toán sinh ra cặp khóa, sinh ra bản mã từ khóa công khai và nguyên bản, khôi phục nguyên bản từ bản mã và khóa riêng
- Các yêu cầu an ninh
  - Không thể tính toán xác định khóa riêng từ khóa công khai, khôi phục nguyên bản từ bản mã và khóa công khai

Giải thuật RSA

- Phát triển năm 1977 bởi Rivest, Shamir và Adleman tại MIT, công bố lần đầu năm 1978
- Phương pháp mã hóa khóa công khai được chấp nhận và cài đặt rộng rãi nhất
- Hệ mã hóa khối với nguyên bản và bản mã là các số tự nhiên từ 0 đến n – 1 với n nào đó
  - Kích thước tiêu biểu của n là 1024 bit hay 309 chữ số thập phân

Mã hóa và giải mã RSA

- Bên gửi mã hóa thông báo M sử dụng khóa công khai của bên nhận $KU = \{e, n\}$

$$C = M^e \bmod n$$

- Bên nhận giải mã bản mã C sử dụng khóa riêng của bản thân $KR = \{d, n\}$

$$M = C^d \bmod n$$

- Các yêu cầu
  - $M^{ed} \bmod n = M$ với mọi $M < n$
  - Không thể xác định được d mặc dù biết e và n

Sinh khóa RSA

- Chọn 2 số nguyên tố lớn $p \neq q$
- Tính mô đun hệ thống $n = p \times q$
- Tính hàm số Euler của n $\phi(n) = (p - 1)(q - 1)$
- Chọn số mũ mã hóa e:

  $\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
- Tính số mũ giải mã d:

  $de \bmod \phi(n) = 1$
- Công bố khóa công khai $KU = \{e, n\}$
- Giữ bí mật khóa riêng $KR = \{d, n\}$

Ví dụ RSA – Sinh khóa

- Chọn 2 số nguyên tố: $p = 17$ and $q = 11$
- Tính $n = p \times q = 17 \times 11 = 187$
- Tính $\phi(n) = (p - 1)(q - 1) = 16 \times 10 = 160$
- Chọn e: gcd(e, 160) = 1 và $1 < e < 160$; chọn $e = 7$
- Tính d: de mod 160 = 1 và $d < 160$; giá trị đúng là $d = 23$, vì $23 \times 7 = 161 = 1 \times 160 + 1$
- Công bố khóa công khai $KU = \{7, 187\}$
- Giữ bí mật khóa riêng $KR = \{23, 187\}$

Ví dụ RSA – Mã hóa/giải mã

- Cho thông báo $M = 88 < 187$
- Mã hóa

  $C = 88^7 \bmod 187 = 11$

  $88^7 \bmod 187 = [(88^1 \bmod 187) \times (88^2 \bmod 187) \times (88^4 \bmod 187)] \bmod 187 = [88 \times (7744 \bmod 187) \times (7744^2 \bmod 187)] \bmod 187 = [88 \times 77 \times (77^2 \bmod 187)] \bmod 187 = (88 \times 77 \times 132) \bmod 187 = 894432 \bmod 187 = 11$
- Giải mã

  $M = 11^{23} \bmod 187 = 88$

Trao đổi khóa Diffie-Hellman

- Giải thuật mã hóa khóa công khai đầu tiên được công bố
- Đề xuất bởi Diffie và Hellman trong bài báo tiên phong định ra mật mã khóa công khai
- Một phương pháp thực tế trao đổi một cách công khai khóa bí mật để sử dụng trong mã hóa các thông báo về sau
- Chỉ dùng cho trao đổi khóa
  - Không thể dùng để trao đổi một thông báo bất kỳ

Sinh khóa Diffie-Hellman

- Hai số công khai
  - *q:* một số nguyên tố lớn
  - α: một nguyên căn của *q*
- Bên A chọn một số ngẫu nhiên $X_A < q$ và tính $Y_A = \alpha^{X_A} \bmod q$
- Bên B chọn một số ngẫu nhiên $X_B < q$ và tính $Y_B = \alpha^{X_B} \bmod q$
- Mỗi bên giữ riêng giá trị *X* và công bố công khai giá trị Y cho bên kia

Sinh khóa bí mật Diffie-Hellman

- Bên A tính khóa $K = Y_B^{X_A} \bmod q$
- Bên B tính khóa $K = Y_A^{X_B} \bmod q$
- Hai phép tính cho kết quả giống nhau

  $K = Y_B^{X_A} \bmod q = (\alpha^{X_B} \bmod q)^{X_A} \bmod q = \alpha^{X_B X_A} \bmod q = (\alpha^{X_A} \bmod q)^{X_B} \bmod q = Y_A^{X_B} \bmod q$
- *K* được dùng làm khóa phiên trong mã hóa đối xứng giữa A và B
- Không thể tính toán tìm ra log rời rạc mô đun một số nguyên tố lớn

Ví dụ Diffie-Hellman

- Đồng ý số nguyên tố $q = 353$ và nguyên căn $\alpha = 3$
- Chọn các khóa riêng ngẫu nhiên
  - A chọn $X_A = 97$, B chọn $X_B = 233$
- Tính các khóa công khai tương ứng
  - A tính $Y_A = \alpha^{X_A} \bmod q = 3^{97} \bmod 353 = 40$
  - B tính $Y_B = \alpha^{X_B} \bmod q = 3^{233} \bmod 353 = 248$
- Tính khóa bí mật chung
  - A computes $K = Y_B^{X_A} \bmod q = 248^{97} \bmod 353 = 160$
  - B computes $K = Y_A^{X_B} \bmod q = 40^{233} \bmod 353 = 160$

Tấn công người ở giữa

Các thành phần công khai tổng thể: q và α

Alice	Darth	Bob
Chọn X_A, Tính Y_A	Chọn X_{D1}, Tính Y_{D1} Chọn X_{D2}, Tính Y_{D2}	Chọn X_B, Tính Y_B
Truyền Y_A cho Bob →	Chặn bắt Y_A	Truyền Y_{D1} cho Bob →
	Tính $K_2 = (Y_A)^{X_{D2}} \bmod q$	
← Truyền Y_{D2} cho Alice	Chặn bắt Y_B	← Truyền Y_B cho Alice
	Tính $K_1 = (Y_B)^{X_{D1}} \bmod q$	
Tính $K_2 = (Y_{D2})^{X_A} \bmod q$		Tính $K_1 = = (Y_{D1})^{X_B} \bmod q$
Gửi E(K_2, M) cho Bob →	Chặn bắt E(K_2, M)	
	Giải mã lấy M	Gửi E(K_1, M) hay E(K_1, M') cho Bob →

Các giải thuật khóa công khai khác

- DSS (Digital Signature Standard)
  - Sử dụng SHA-1
  - Chỉ cung cấp tính năng chữ ký số
    - Không thể dùng mã hóa hoặc trao đổi khóa như RSA
- ECC (Elliptic-Curve Cryptography)
  - Có tiềm năng thay thế RSA
    - Độ an toàn dường như tương đương với kích thước bit nhỏ hơn làm giảm phụ tải tính toán
    - Lý thuyết đã có từ lâu nhưng mới được ứng dụng gần đây nên mức độ tin cậy vào giải thuật không cao

Chữ ký số

- Mã hóa thông báo với khóa riêng của bên gửi tạo nên chữ ký số
  - Không ai khác có khóa riêng của bên gửi để tạo bản mã có thể giải mã với khóa công khai
  - Không có khóa riêng không thể sửa thông báo
- Nhược điểm của mã hóa thông báo
  - Phải lưu trữ bản mã kèm với thông báo
- Mã hóa một khối nhỏ đại diện cho thông báo sẽ hiệu quả hơn
  - Có thể sử dụng một hàm băm như SHA-1

Chương 4

PHÂN PHỐI KHÓA VÀ XÁC THỰC NGƯỜI DÙNG

Phân phối bằng mã hóa đối xứng

- Các yêu cầu đối với khóa đối xứng
  - Được bảo vệ khỏi bị truy nhập bởi các bên thứ ba
  - Thường mong muốn thay đối khóa thường xuyên
    - Để giảm thiểu lượng dữ liệu bị xâm hại nếu địch thủ biết được khóa
- Sức mạnh của bất kỳ hệ mật mã học nào đều dựa vào kỹ thuật phân phối khóa
- Phân phối khóa là biện pháp chuyển giao khóa cho 2 bên giao thiệp không cho các bên khác thấy

Các tùy chọn cho phân phối khóa

- A chọn khóa và vận thân chuyển giao nó cho B
- Một bên thứ ba chọn khóa và vận thân chuyển giao nó cho A và B
- Nếu A và B trước đây đã có một khóa chung thì có thể dùng khóa cũ để mã hóa khóa mới
- Nếu mỗi bên A và B đều có 1 kết nối được mã hóa với 1 bên thứ ba C thì C có thể chuyển giao khóa trên các kết nối đó cho A và B

Thảo luận về các tùy chọn

- Hai tùy chọn đầu tiên
  - Yêu cầu hợp lý đối với mã hóa liên kết
  - Bất tiện đối với mã hóa đầu cuối tới đầu cuối
- Tùy chọn thứ ba
  - Có thể sử dụng với mã hóa liên kết hay đầu cuối tới đầu cuối
  - Nếu một khóa bị xâm hại, tất cả các khóa sau bị lộ
- Tùy chọn thứ tư
  - Được ưa chuộng với mã hóa đầu cuối tới đầu cuối

Kerberos

- Một dịch vụ phân phối khóa và xác thực người dùng được phát triển tại MIT
- Vấn đề được giải quyết bởi Kerberos
  - Người dùng tại các trạm làm việc muốn truy nhập dịch vụ trên các server phân tán khắp mạng
  - Các server có thể chỉ cho những người được phép truy nhập và xác thực các truy vấn tới các dịch vụ
  - Các trạm làm việc không đáng tin cậy để định danh đúng đắn người dùng cho các dịch vụ mạng

Các hiểm họa đối với Kerberos

- Truy nhập trái phép dịch vụ
  - Một người dùng vào một trạm làm việc nào đó và giả bộ là một người dùng khác đang hoạt động từ trạm làm việc đó
  - Thay đổi địa chỉ mạng của một trạm làm việc để làm như truy vấn được gửi từ trạm làm việc bị giả mạo
  - Nghe trộm các trao đổi và sử dụng tấn công lặp lại để giành quyền truy nhập vào một server hay làm ngưng trệ hoạt động

Các đặc tính của Kerberos

- Cung cấp một server xác thực tập trung để xác thực người dùng với server và server với người dùng
  - Thay vì xây dựng các giao thức xác thực phức tạp tại từng server
- Dựa một cách duy nhất vào mã hóa đối xứng
- Hai phiên bản đang được sử dụng: 4 và 5
  - Phiên bản 5 hiệu chỉnh một số khiếm khuyết an ninh của phiên bản 4 (đang dần ngừng sử dụng)

Một hội thoại xác thực đơn giản

- Sử dụng một server xác thực (AS)
  - Biết mật khẩu của tất cả người dùng
    - Biết cả quyền sử dụng dịch vụ
  - Chia sẻ một khóa bí mật duy nhất với mỗi server
    - Phân phối vận thân hay bằng một cách an toàn khác
- Hội thoại giả định

(1) C $\rightarrow$ AS: $ID_c \parallel P_c \parallel ID_v$

(2) AS $\rightarrow$ C: *Thẻ*

(3) C $\rightarrow$ V: $ID_c \parallel$ *Thẻ*

Thẻ = E(K_v, [$ID_c \parallel AD_c \parallel ID_v$])

Các vấn đề với hội thoại thứ nhất

- Cần giảm thiểu số lần nhập mật khẩu
  - Nếu mỗi thẻ chỉ có thể sử dụng được một lần, thì mỗi lần truy nhập đều yêu cầu nhập lại mật khẩu
  - Nếu các thẻ có thể tái sử dụng thì mỗi lần truy nhập một server mới đều yêu cầu nhập lại mật khẩu
- Mật khẩu được truyền nguyên bản
  - Có thể nghe trộm lấy mật khẩu và sử dụng bất kỳ dịch vụ nào có thể truy nhập được bởi nạn nhân

Một hội thoại an toàn hơn

- **Một lần mỗi phiên người dùng đăng nhập**
  - (1) C $\rightarrow$ AS: $ID_c \parallel ID_{tgs}$
  - (2) AS $\rightarrow$ C: E(K_c, $Thẻ_{tgs}$)
- **Một lần với mỗi kiểu dịch vụ**
  - (3) C $\rightarrow$ TGS: $ID_c \parallel ID_v \parallel Thẻ_{tgs}$
  - (4) TGS $\rightarrow$ C: $Thẻ_v$
- **Một lần với mỗi phiên dịch vụ**
  - (5) C $\rightarrow$ V: $ID_c \parallel Thẻ_v$

$Thẻ_{tgs}$ = E(K_{tgs}, [$ID_c \parallel AD_c \parallel ID_{tgs} \parallel TS_1 \parallel Hạn_1$])

$Thẻ_v$ = E(K_v, [$ID_c \parallel AD_c \parallel ID_v \parallel TS_2 \parallel Hạn_2$])

Các vấn đề với hội thoại thứ hai

- Phải chứng minh người sử dụng thẻ cũng chính là người được phát thẻ
  - Nếu thời hạn rất ngắn thì người dùng sẽ bị hỏi mật khẩu liên tục
  - Nếu thời hạn dài thì địch thủ có cơ hội lớn để tấn công lặp lại
- Yêu cầu các server phải tự xác thực đối với các người dùng
  - Một server giả sẽ lấy thông tin từ người dùng và không cung cấp dịch vụ thật

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 : $E_{K_{C,tgs}}[K_{C,V} \parallel ID_V \parallel TS_4 \parallel Thẻ_V]$

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

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

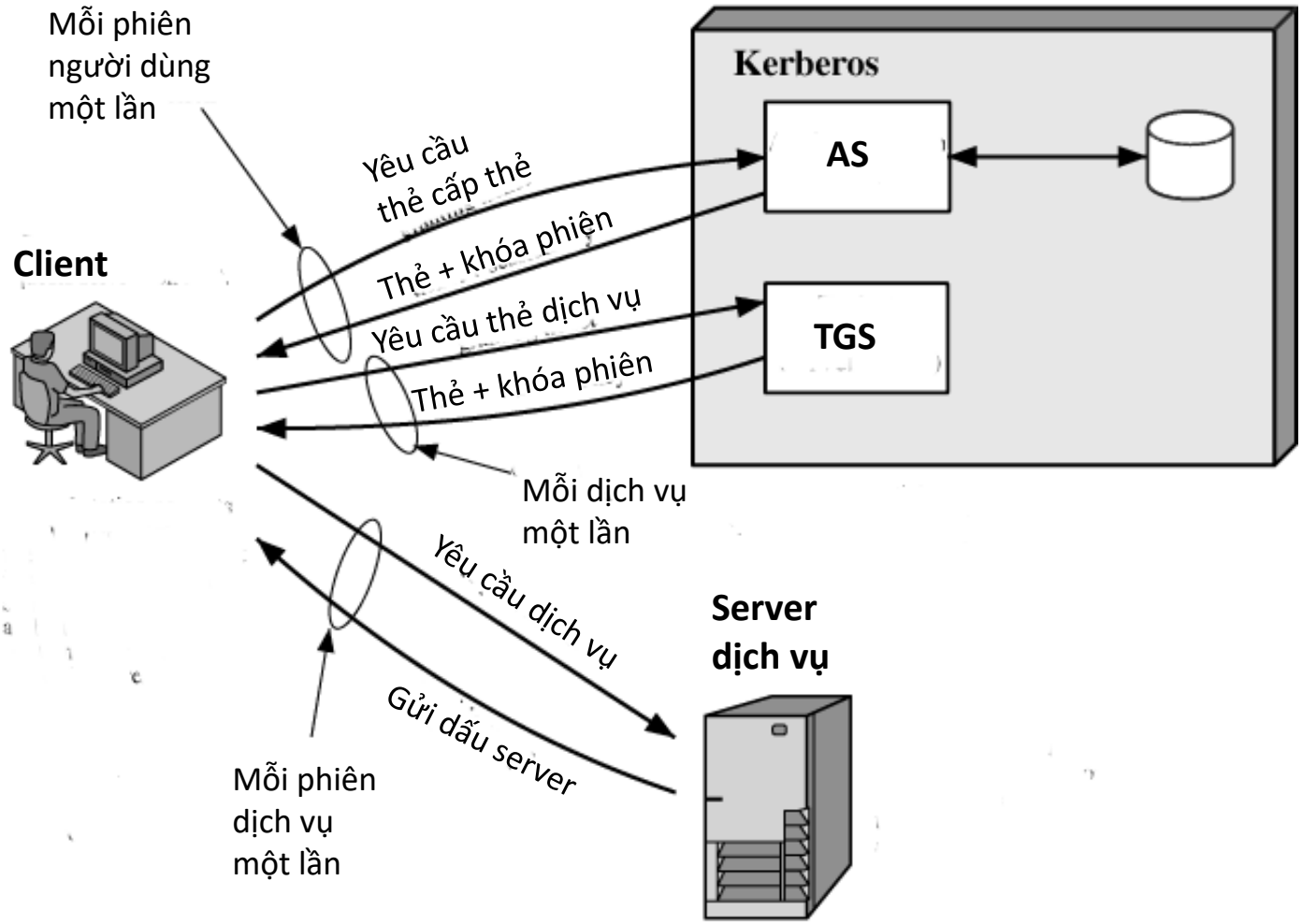
(c) Trao đổi xác thực client/server : để có dịch vụ

(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

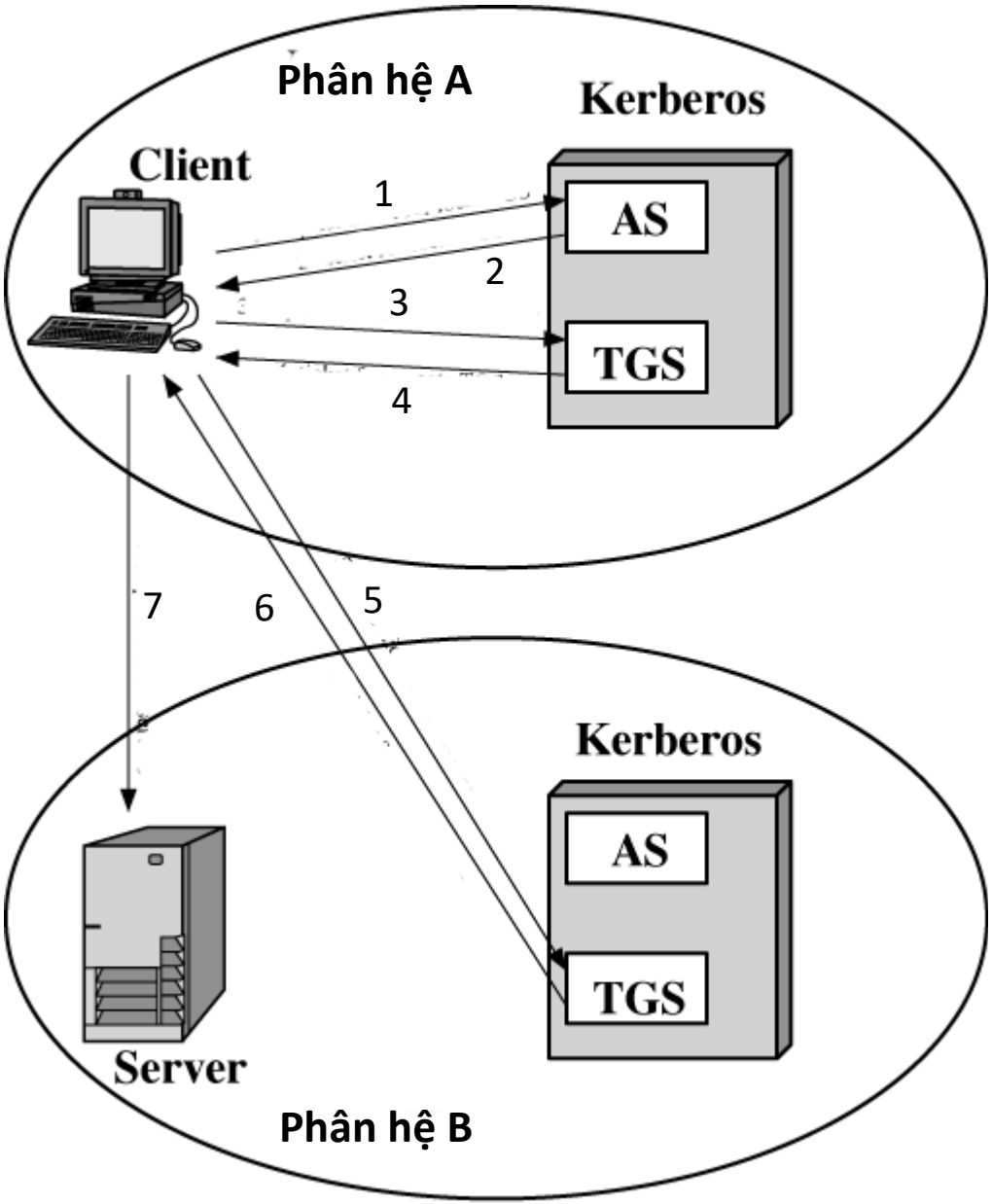


Phân hệ Kerberos

- Một phân hệ Kerberos bao gồm một server Kerberos, các client và các server ứng dụng
- Các yêu cầu đối với phân hệ Kerberos
  - Server Kerberos có định danh và mật khẩu băm của tất cả các người dùng trong CSDL của nó
    - Và cả các quyền truy nhập
  - Server Kerberos có 1 khóa chung với mỗi server
- Một phân hệ Kerberos thường tương ứng với một đơn vị hành chính

Xác thực liên phân hệ

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

- Được đặc tả trong RFC 4120
- Khắc phục một số hạn chế của phiên bản 4
  - Những khiếm khuyết ngoại cảnh
    - Phụ thuộc hệ mã hóa, phụ thuộc giao thức Internet, trật tự byte thông báo, hạn sử dụng thẻ, chuyển tiếp xác thực, xác thực liên phân hệ
  - Những thiếu sót kỹ thuật
    - Mã hóa kép, chế độ mã hóa PCBC, khóa phiên, tấn công mật khẩu

Hội thoại Kerberos 5

(1) C → AS $Options \| ID_c \| Realm_c \| ID_{tgs} \| Times \| Nonce_1$

(2) AS → C $Realm_c \| ID_C \| Ticket_{tgs} \| \mathrm{E}(K_c, [K_{c,tgs} \| Times \| Nonce_1 \| Realm_{tgs} \| ID_{tgs}])$

$Ticket_{tgs} = \mathrm{E}(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) C → TGS $Options \| ID_v \| Times \| Nonce_2 \| Ticket_{tgs} \| Authenticator_c$

(4) TGS → C $Realm_c \| ID_C \| Ticket_v \| \mathrm{E}(K_{c,tgs}, [K_{c,v} \| Times \| Nonce_2 \| Realm_v \| ID_v])$

$Ticket_{tgs} = \mathrm{E}(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$

$Ticket_v = \mathrm{E}(K_v, [Flags \| K_{c,v} \| Realm_c \| ID_C \| \mathrm{AD}_C \| Times])$

$Authenticator_c = \mathrm{E}(K_{c,tgs}, [ID_C \| Realm_c \| TS_1])$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

(5) C → V $Options \| Ticket_v \| Authenticator_c$

(6) V → C $\mathrm{E}_{Kc,v} [TS_2 \| Subkey \| Seq\#]$

$\mathrm{Ticket}_v = \mathrm{E}(K_v, [\mathrm{Flags} \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times])$

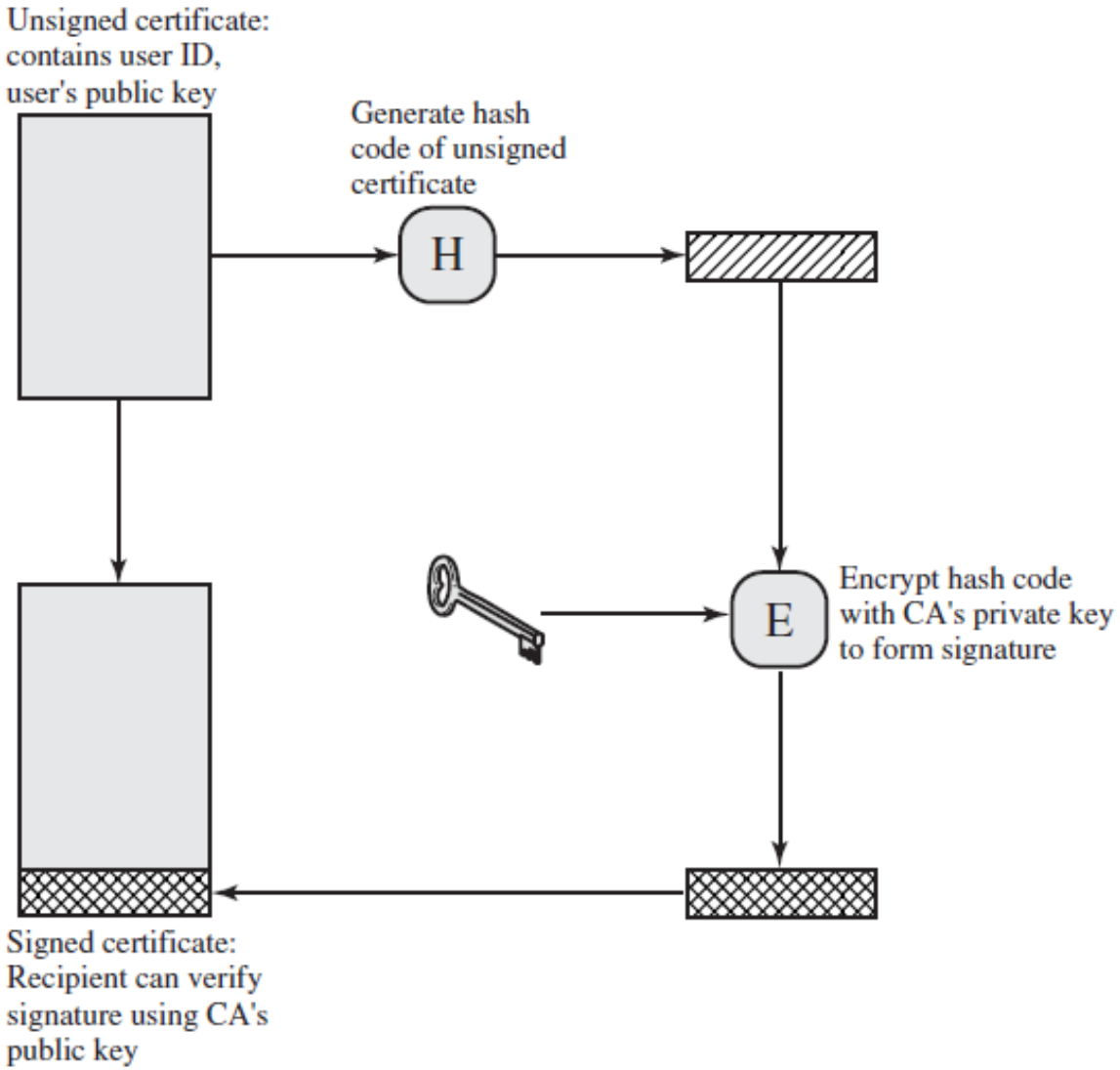
$Authenticator_c = \mathrm{E}(K_{c,v}, [ID_C \| Realm_c \| TS_2 \| Subkey \| Seq\#])$

(c) Client/Server Authentication Exchange to obtain service

Phân khóa với mã hóa bất đối xứng

- Phân phối khóa công khai
  - Thông qua các thông báo công khai
    - Một người có thể giả danh là một người A
    - Kẻ giả danh có thể đọc tất cả các thông báo mã hóa gửi cho A và sử dụng các khóa giả mạo để xác thực
  - Thông qua các chứng thực khóa công khai
    - Chuẩn X.509
- Phân khối khóa bí mật
  - Sử dụng trao đổi khóa Diffie-Hellman
  - Sử dụng các chứng thực khóa công khai

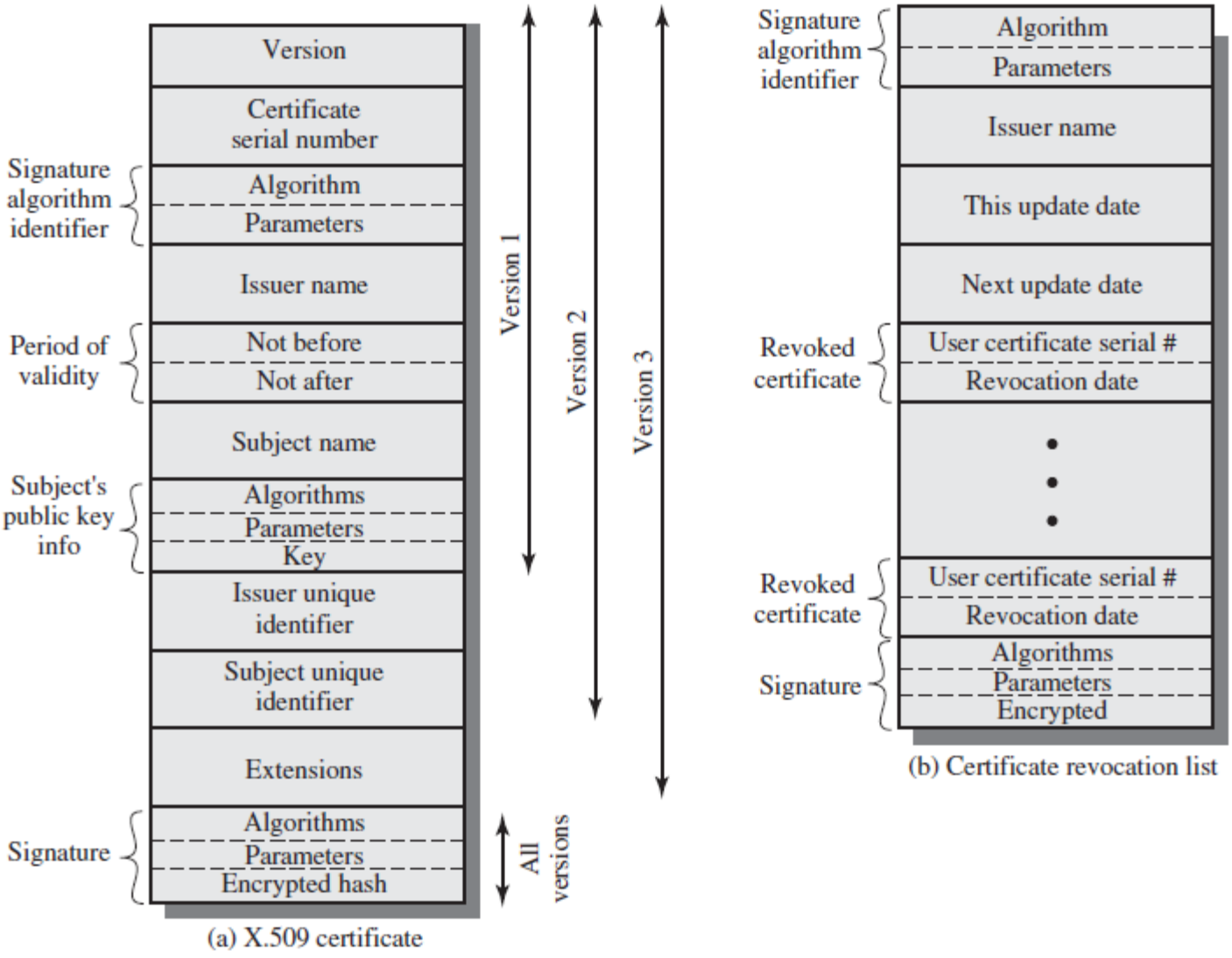
Sử dụng chứng thực



X.509

- Nằm trong loạt khuyến nghị X.500 của ITU-T nhằm chuẩn hóa dịch vụ thư mục
- Định ra một cơ cấu để cung cấp các dịch vụ xác thực thông qua thư mục X.500
- Định ra các giao thức xác thực tùy chọn dựa trên sử dụng các chứng thực khóa công khai
- Được sử dụng trong S/MIME, IPSec, SSL/TLS
- Ban hành lần đầu 1988, hiệu chỉnh 1993; phiên bản 3 ban hành 1995, hiệu chỉnh 2000

Khuôn dạng X.509



(a) X.509 certificate

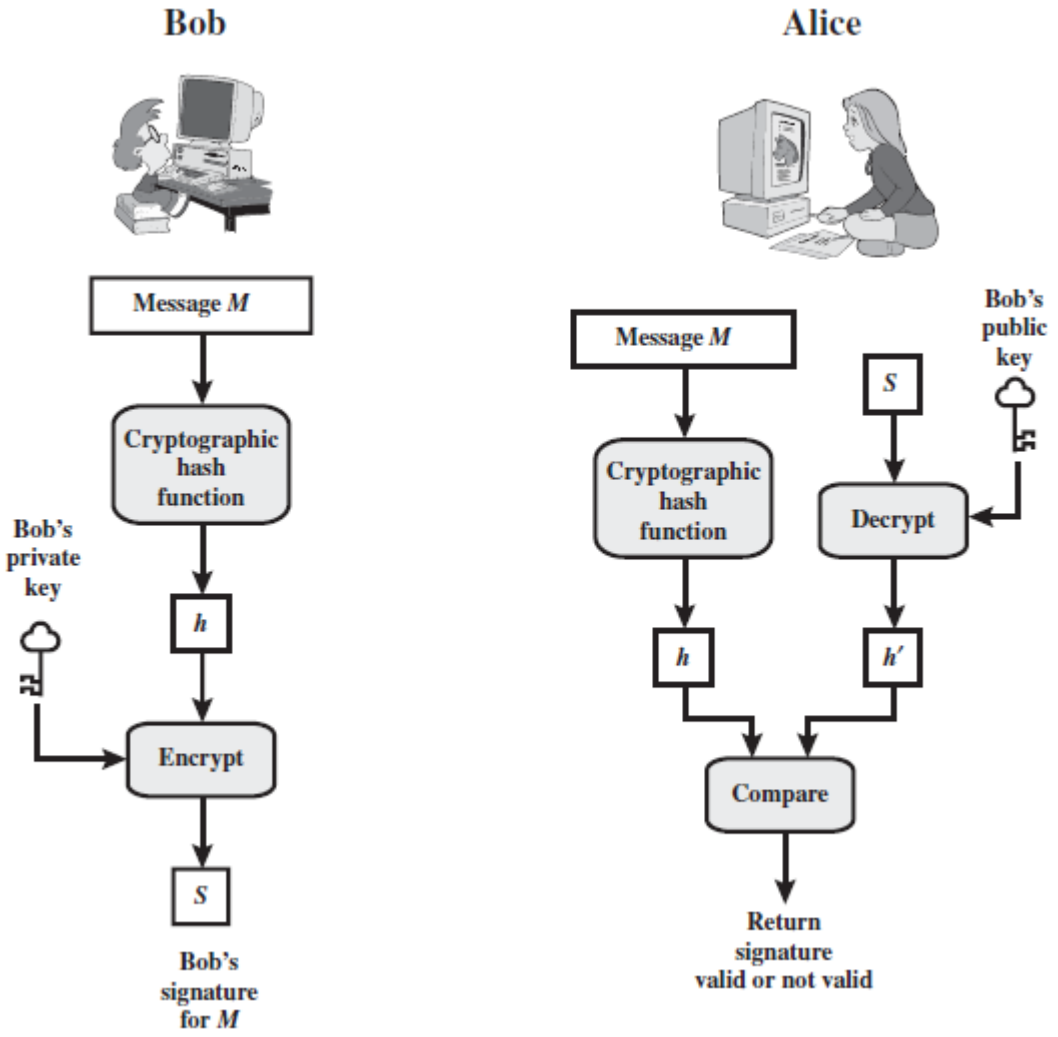
(b) Certificate revocation list

Ký hiệu X.509

CA<<A>> = CA{V, SN, AI, CA, UCA, A, UA, Ap, T^A}

- Y<<X>> = chứng thực của X phát hành bởi Y
- Y{I} = nội dung I được ký bởi Y; bao gồm I đính kèm mã băm được mã hóa với khóa riêng của Y
- Các ký hiệu khác
  - V = phiên bản của chứng thực
  - SN = số thứ tự của chứng thực
  - AI = định danh của giải thuật dùng để ký chứng thực
  - CA, A = tên của cơ quan chứng thực và người dùng A
  - UCA, UA = định danh duy nhất tùy chọn của CA và A
  - AP = khóa công khai của người dùng A
  - T^A = thời hạn hợp lệ của chứng thực

Quy trình chữ ký số



Lấy chứng thực người dùng

- Các đặc tính của chứng thực sinh ra bởi 1 CA
  - Bất kỳ người dùng nào có khóa công khai của CA cũng có thể xác minh xem khóa công khai người dùng có được chứng thực hay không
  - Chỉ CA có thể sửa đổi chứng thực mà không bị phát hiện
- Vì không thể bị giả mạo, chứng thực có thể được đặt ở một thư mục công khai

Vấn đề phân cấp CA

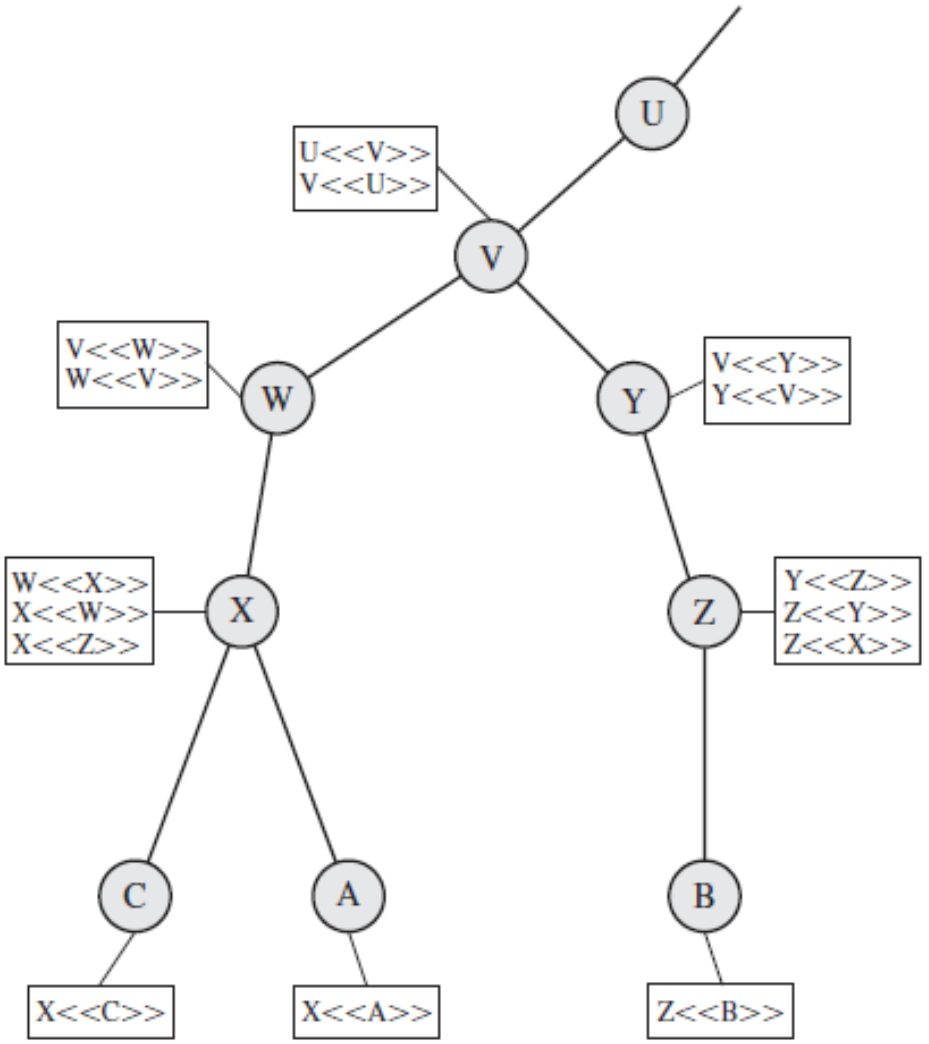
- Nếu tất cả người dùng đăng ký với cùng 1 CA
  - Tất cả mọi người đều tin tưởng vào cùng CA
  - Tất cả chứng thực có thể được đặt vào cùng 1 thư mục cho tất cả mọi người truy nhập
- Với 1 cộng đồng rộng lớn, khó có thể yêu cầu tất cả người dùng đăng ký với cùng 1 CA
  - Cần có một số CA
    - Khóa công khai của CA cần được cung cấp tới mỗi người dùng một cách tuyệt đối an toàn
    - Mỗi CA chỉ cấp khóa công khai đến một số người dùng

Tổ chức phân cấp CA

- Các CA tạo thành một tổ chức phân cấp
  - Một thư mục với mỗi đề mục tương ứng với 1 CA
  - Đề mục cho mỗi CA X chứa 2 kiểu chứng thực
    - Các chứng thực thuận: Chứng thực của X sinh bởi các CA khác
    - Các chứng thực nghịch: Chứng thực sinh bởi X cho các CA khác
- Cho phép xác minh chứng thực từ bất kỳ CA nào bởi tất cả người dùng của các CA khác trong tổ chức phân cấp

Sử dụng phân cấp CA

- A có chứng thực từ X muốn xác minh chứng thực của B do Z cấp
  - A lấy chứng thực của W do X cấp,…, chứng thực của Z do Y cấp
  - A xác minh chuỗi X<<W>>W<<V>>V<<Y>>Y<<Z>>Z<<B>>



Thu hồi chứng thực

- Mỗi chứng thực có một thời hạn hợp lệ
- Có thể cần thu hồi chứng thực trước hạn
  - Khóa riêng của người dùng bị lộ
  - Người dùng không còn được chứng thực bởi CA
  - Chứng thực của CA không còn an toàn
- Mỗi CA phải duy trì một danh sách các chứng thực bị thu hồi (CRL), đặt nó trong thư mục
- Người dùng khi nhận một chứng thực phải xác định xem nó đã bị thu hồi hay chưa

Những hạn chế của X.509 v2

- Trường Subject không thể hiện được định danh người dùng
- Trường Subject không thích hợp với nhiều ứng dụng
- Không có thông tin về chính sách an ninh
- Không có ràng buộc về phạm vi sử dụng
- Không cho phép định danh các khóa khác nhau thuộc về cùng một chủ thể

X.509 phiên bản 3

- Một phương pháp linh động hơn thay vì thêm các trường vào một khuôn dạng cố định
  - Bao gồm một số phần mở rộng tùy chọn
  - Mỗi phần mở rộng bao gồm định danh, chỉ số mức quan trọng, và giá trị
    - Mức quan trọng chỉ ra phần mở rộng có thể bỏ qua hay nếu không có thì chứng thực bị coi là không hợp lệ
- Bao gồm 3 kiểu mở rộng
  - Thông tin khóa và chính sách, các thuộc tính chủ thể và CA, các ràng buộc chuỗi chứng thực

Thông tin khóa và chính sách

- Định danh khóa của CA
- Định danh khóa của người dùng
- Phạm vi sử dụng khóa
- Thời hạn sử dụng khóa riêng
- Các chính sách chứng thực
  - Mỗi chính sách là 1 tập các luật chỉ ra phạm vi áp dụng của chứng thực cho 1 cộng đồng và/hay 1 lớp ứng dụng với các yêu cầu an ninh chung
- Ánh xạ chính sách

Các thuộc tính chủ thể và CA

- Tên thay thế của chủ thể
  - Chứa một hoặc nhiều tên thay thế, sử dụng một trong hàng loạt các mẫu
    - Một số ứng dụng như thư điện tử, EDI, và IPSec sử dụng mẫu tên riêng
- Tên thay thế của cơ quan phát hành
  - Tương tự tên thay thế của chủ thể
- Các thuộc tính thư mục của chủ thể
  - Truyền tải thông tin bổ sung như địa chỉ, chức vụ, hay hình ảnh của chủ thể

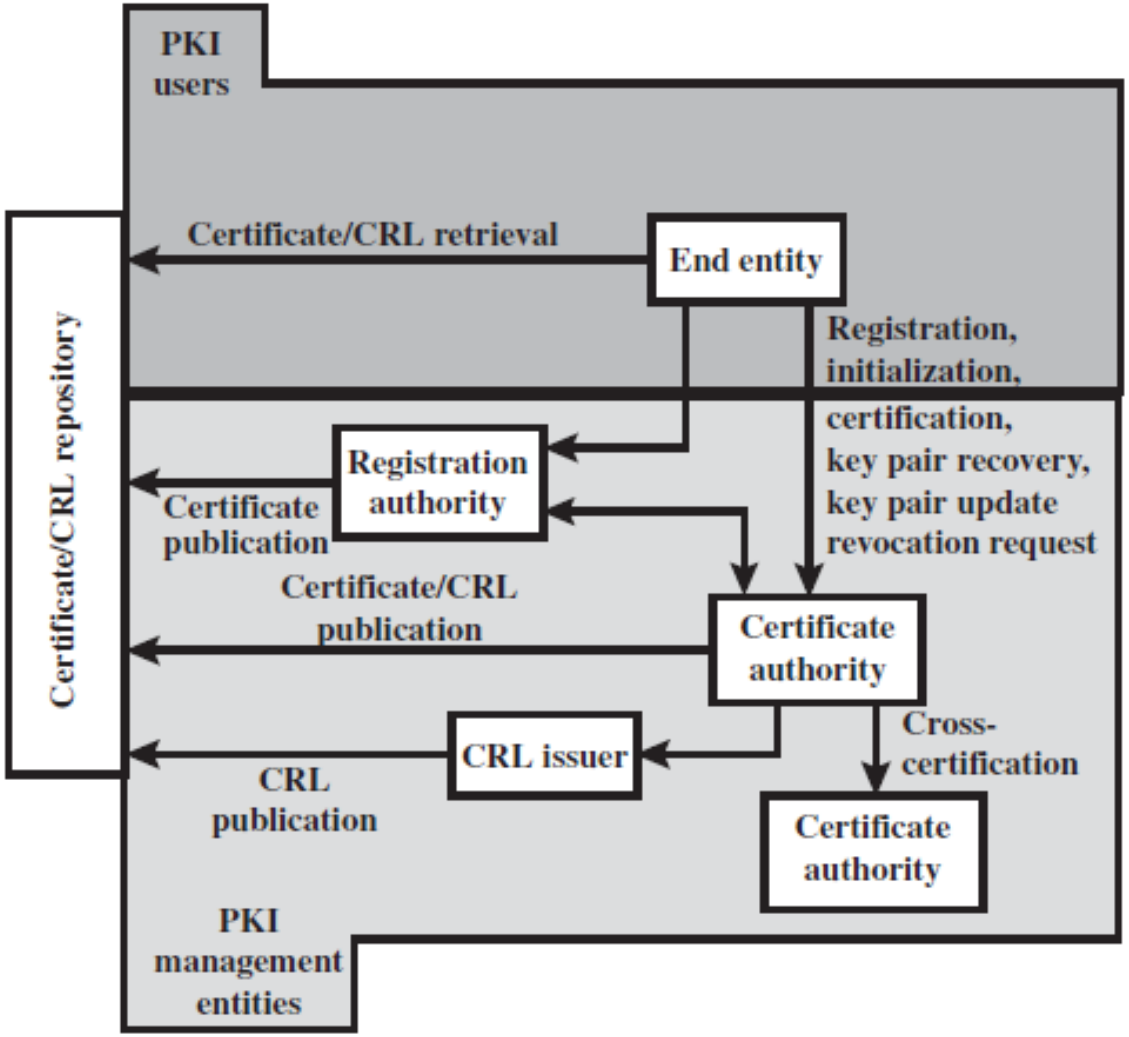
Các ràng buộc chuỗi chứng thực

- Các ràng buộc cơ bản
  - Chủ thể có thể đóng vai trò như 1 CA hay không
    - Nếu có thì có thể chỉ định hạn chế về độ dài chuỗi
- Các ràng buộc tên
  - Miền tên bắc buộc với các chủ thể sau trong chuỗi
- Các ràng buộc chính sách
  - Yêu cầu định danh tường minh chính sách chứng thực hay cấm ánh xạ chứng thực với phần còn lại của chuỗi

Cơ sở hạ tầng khóa công khai

- Định nghĩa của RFC 2822
  - Tập các phần cứng, phần mềm, con người, chính sách, và thủ tục cần thiết để tạo lập, quản lý, lưu trữ, phân phối và thu hồi các chứng thực số
    - Cho phép tiếp nhận các khóa công khai một cách an toàn, thuận tiện, và hiệu quả
- Mô hình PKIX (Public Key Infrastructure X.509)
  - Do IETF thiết lập dựa trên X.509
  - Thích hợp cho triển khai một kiến trúc dựa trên chứng thực trên mạng Internet

Mô hình kiến trúc PKIX



Các thành phần của PKIX

- Thực thể đầu cuối
- Cơ quan chứng thực (CA)
- Cơ quan đăng ký (RA)
  - Tùy chọn, thường gắn với quá trình đăng ký
- Bộ phận phát hành CRL
  - Tùy chọn, được CA ủy nhiệm công bố CRL
- Kho lưu trữ
  - Lưu trữ công khai các chứng thực và CRL

Các chức năng quản lý của PKIX

- Đăng ký
- Tạo lập
- Chứng thực
- Khôi phục cặp khóa
- Cập nhật cặp khóa
- Yêu cầu thu hồi
- Chứng thực chéo

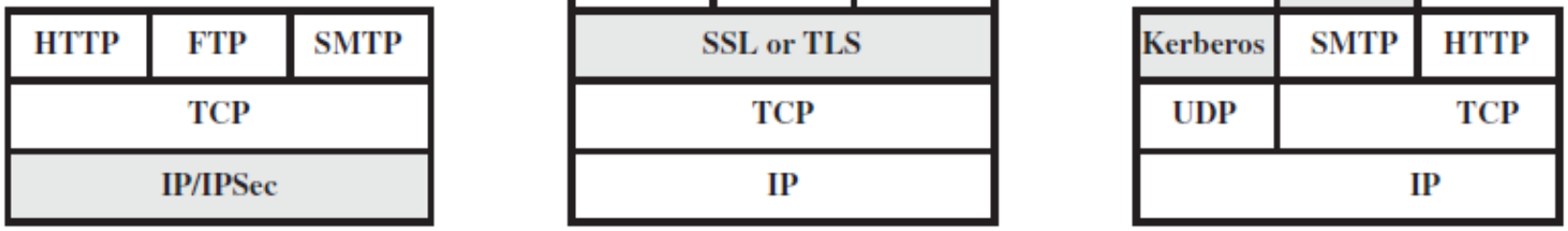
Chương 5

AN NINH TẦNG GIAO VẬN

Đặt vấn đề

- Web được sử dụng rộng rãi bởi các doanh nghiệp, cơ quan chính phủ và cá nhân
- Thực tế Internet và Web rất dễ bị tấn công
- Các hiểm họa an ninh Web
  - Tấn công thụ động và tấn công chủ động
  - Tấn công vào server Web, trình duyệt Web và thông tin truyền giữa trình duyệt và server
- Chương này tập trung vào 3 chuẩn an ninh truyền thông Web SSL/TLS, HTTPS và SSH

An ninh truyền thông Web

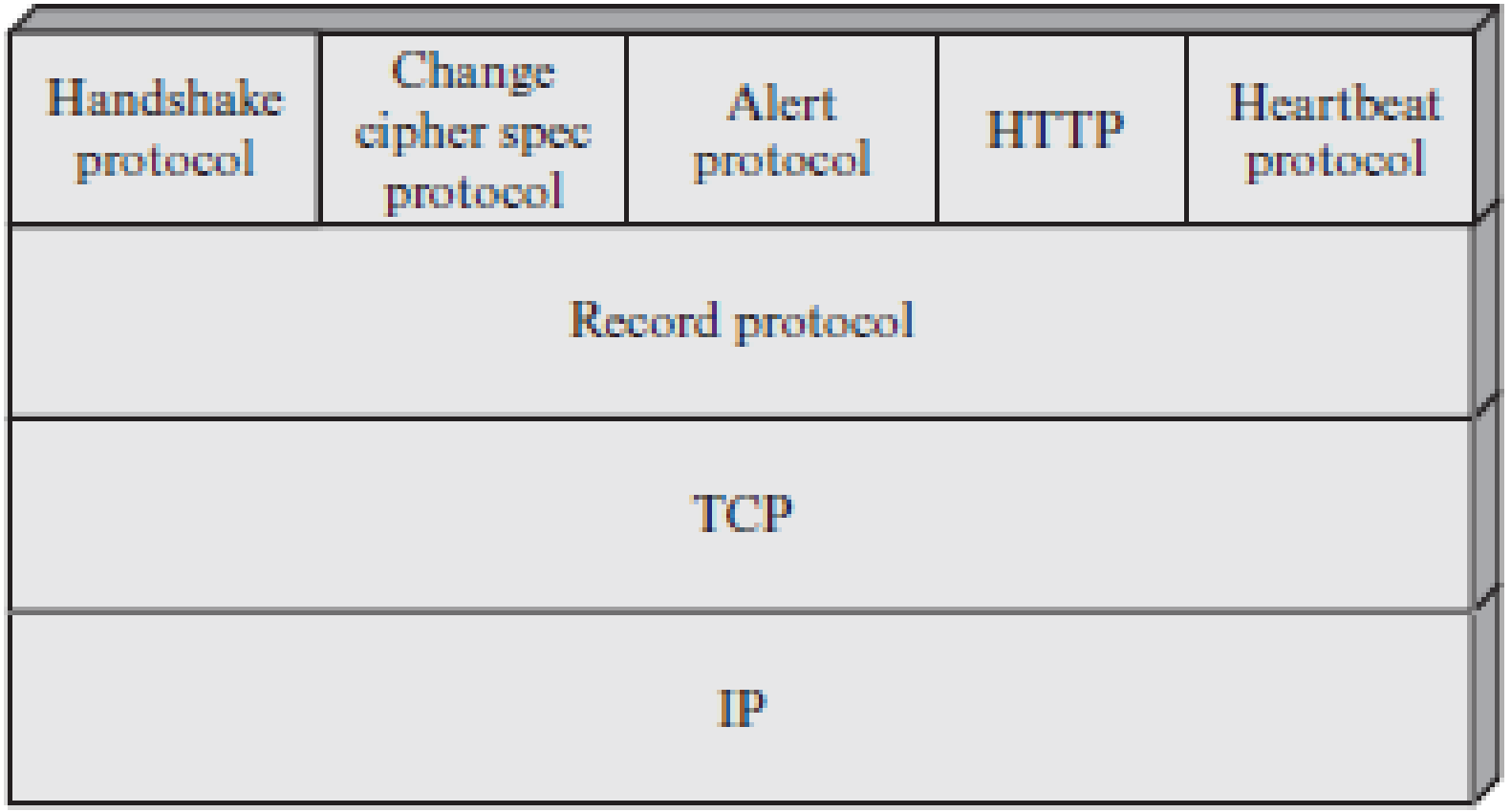


(a) Network level (b) Transport level (c) Application level

TLS (Transport Layer Security)

- Phiên bản hiện tại là 1.2, đặc tả trong RFC 5246
- Phát triển từ giao thức SSL (Secure Sockets Layer)
- Là một dịch vụ an ninh được cài đặt như một tập các giao thức trên nền TCP
- Có hai lựa chọn cài đặt
  - Như một bộ giao thức ở tầng dưới, trong suốt với các ứng dụng
  - Nhúng vào các gói phần mềm chuyên biệt
- Hầu hết các trình duyệt Web và Web server cài đặt sẵn TLS

Kiến trúc TLS



Các khái niệm TLS quan trọng

- Kết nối
  - Một giao vận cung cấp một kiểu dịch vụ phù hợp
    - Quan hệ ngang hàng và có tính nhất thời
  - Gắn với một phiên
- Phiên
  - Một liên kết giữa 1 client và 1 server
  - Được khởi tạo bởi giao thức Handshake
  - Định ra một tập các thông số an ninh
  - Có thể chung cho nhiều kết nối

Trạng thái phiên

- Mỗi phiên có 1 trạng thái hoạt động hiện thời cho cả 2 thao tác đọc và viết (nhận và gửi)
- Trong giao thức Handshake, các trạng thái đọc và viết treo được tạo ra
  - Khi Handshake thành công, các trạng thái treo trở thành các trạng thái hiện thời
- Các thông số xác định một trạng thái phiên
  - Định danh phiên, chứng thực đối tác, phương pháp nén, đặc tả mật mã, bí mật chủ, cờ nối tiếp

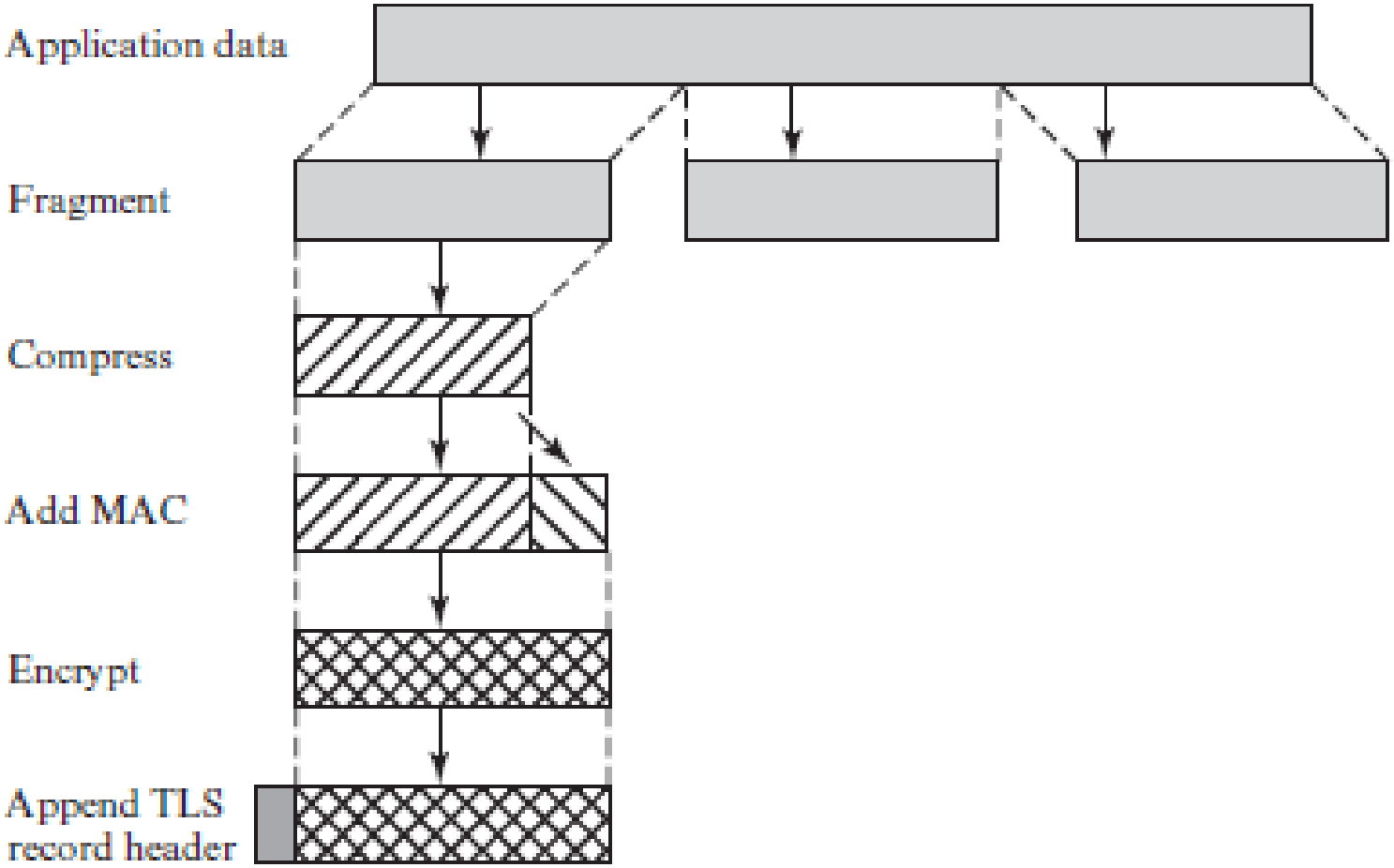
Trạng thái kết nối

- Các thông số xác định một trạng thái kết nối
  - Số ngẫu nhiên của server và của client
  - Bí mật viết MAC của server
  - Bí mật viết MAC của client
  - Khóa viết của server
  - Khóa viết của client
  - Các IV
  - Các số thứ tự

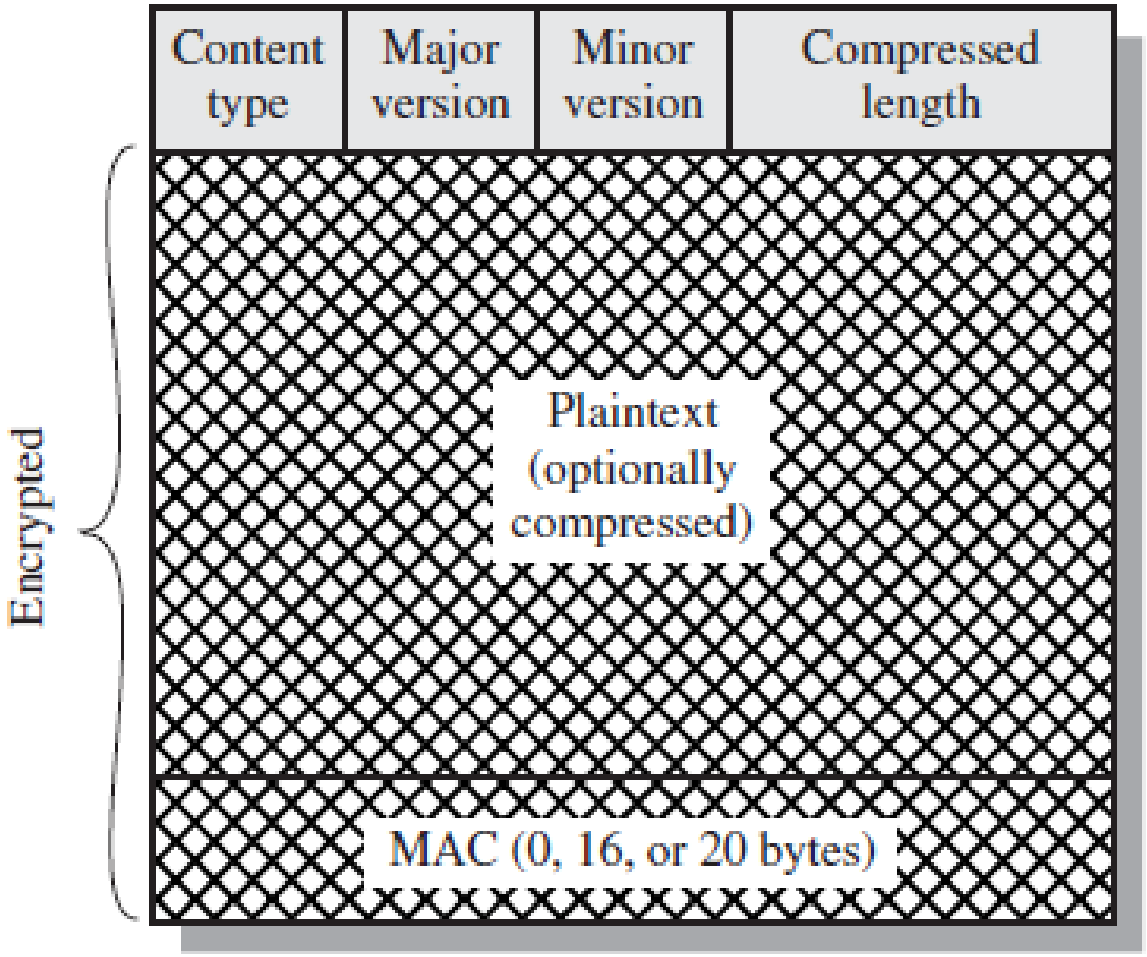
Giao thức Record của TLS

- Cung cấp 2 dịch vụ cho các kết nối TLS
  - Bảo mật
    - Sử dụng khóa bí mật chung xác định bởi giao thức Handshake để mã hóa truyền thống các nội dung TLS
  - Toàn vẹn thông báo
    - Sử dụng khóa bí mật chung xác định bởi giao thức Handshake để tạo mã xác thực thông báo (MAC)
    - HMAC_hash(MAC_write_secret, seq_num || TLSCompressed.type || TLSCompressed.length || TLSCompressed.fragment)

Hoạt động của giao thức Record

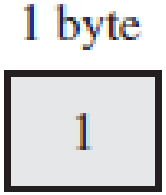


Khuôn dạng bản ghi



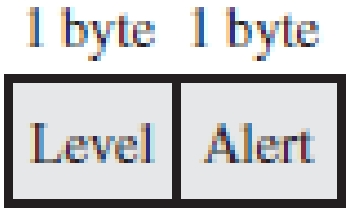
Giao thức Change Cipher Spec

- Một trong 4 giao thức riêng của TLS sử dụng giao thức Record của TLS
- Bao gồm một thông báo với 1 byte có giá trị 1
- Khiến trạng thái treo được sao chép vào trạng thái hiện thời
  - Cập nhật tập mật mã dùng cho kết nối hiện thời



Giao thức Alert

- Dùng để truyền tải các báo động liên quan TLS đến các thực thể truyền thông
- Được nén và mã hóa theo đặc tả trạng thái hiện thời, như các thông báo ứng dụng khác
- Mỗi thông báo bao gồm 2 byte
  - Byte đầu tiên chỉ mức độ tai hại hay cảnh báo
    - Nếu mức độ tai hại thì SSL lập tức kết thúc kết nối, không cho phép thiết lập các kết nối mới
  - Byte thứ hai chứa mã báo động



Các thông báo Alert

- Báo động tai hại
  - unexpected_message, bad_record_mac, decompression_failure, handshake_failure, illegal_parameter,…
- Cảnh báo
  - close_notify, bad_certificate, unsupported_certificate, certificate_revoked, certificate_expired, certificate_unknown,…

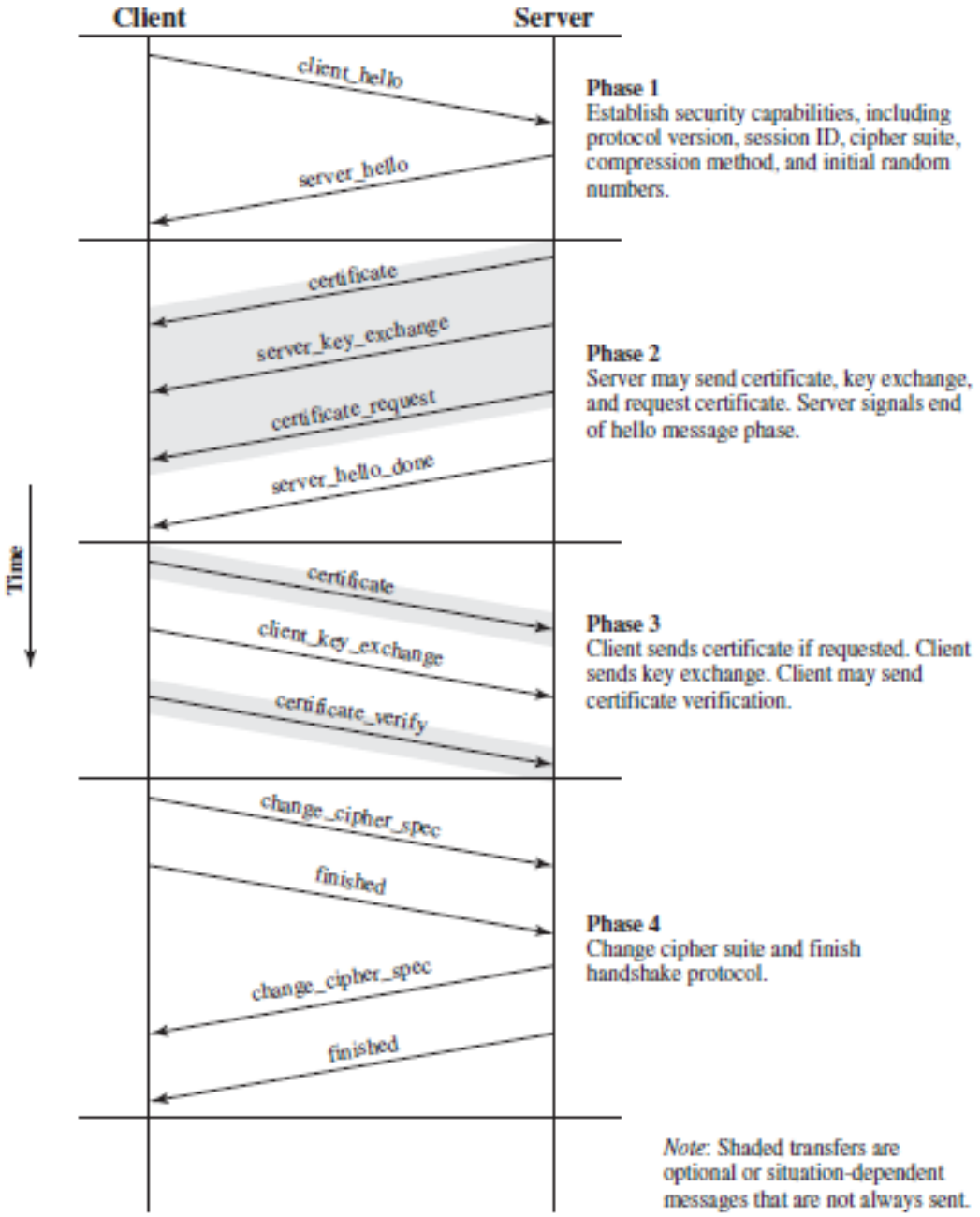
Giao thức Handshake (1)

- 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
- Bao gồm một chuỗi các thông báo trao đổi
  - Mỗi thông báo gồm có 3 trường
    - **Kiểu (1 byte):** Chỉ ra 1 trong 10 thông báo
    - **Độ dài (3 byte):** Độ dài thông báo tính bằng byte
    - **Nội dung (**≥ 0 **byte):** Các thông số gắn với thông báo

Giao thức Handshake (2)

- Trao đổi thông báo qua 4 giai đoạn
  - Thiết lập các năng lực an ninh
  - Xác thực server và trao đổi khóa
  - Xác thực client và trao đổi khóa
  - Kết thúc

Lược đồ giao thức Handshake



Thông báo client_hello (1)

- Phiên bản
  - Phiên bản TLS cao nhất client hiểu được
- Số ngẫu nhiên
  - Nhãn thời gian 32 bit + số ngẫu nhiên 28 byte
  - Dùng để ngăn ngừa tấn công lặp lại
- Định danh phiên
  - Giá trị khác 0 nếu client muốn cập nhật kết nối hiện tại hay tạo kết nối mới cùng phiên
  - Giá trị 0 nếu client muốn tạo kết nối và phiên mới

Thông báo client_hello (2)

- Bộ mật mã
  - Danh sách các tổ hợp giải thuật mật mã được client hỗ trợ
    - Theo thứ tự ưa thích giảm dần
  - Mỗi bộ mật mã chỉ định cả giải thuật trao đổi khóa và đặc tả mật mã
- Phương pháp nén
  - Danh sách các phương pháp nén được client hỗ trợ

Thông báo server_hello (1)

- Phiên bản
  - Phiên bản thấp hơn trong phiên bản gợi ý bởi client và phiên bản cao nhất hỗ trợ bởi server
- Số ngẫu nhiên
  - Sinh bởi server độc lập với số của client
- Định danh phiên
  - Cũng là định danh phiên của client nếu khác 0
  - Nếu không thì chứa giá trị cho một phiên mới

Thông báo server_hello (2)

- Bộ mật mã
  - Chứa chỉ một bộ mật mã được server lựa chọn từ danh sách đề xuất bởi client
- Phương pháp nén
  - Chứa phương pháp nén được server lựa chọn từ danh sách đề xuất bởi client

Phương pháp trao đổi khóa (1)

- RSA
  - Khóa bí mật được mã hóa với khóa công khai RSA của bên nhận
    - Phải có chứng thực khóa công khai của bên nhận
- Diffie-Hellman cố định
  - Chứng thực của server chứa các thông số công khai Diffie-Hellman
  - Client cung cấp các thông số khóa công khai D-H trong chứng thực
  - Tạo khóa bí mật cố định

Phương pháp trao đổi khóa (2)

- Diffie-Hellman tức thời
  - Tạo ra các khóa bí mật tạm thời, dùng 1 lần
  - Các khóa công khai Diffie-Hellman được ký với khóa riêng RSA hay DSS của bên gửi
  - Chứng thực dùng để xác thực các khóa công khai
  - Phương pháp an toàn nhất trong 3 tùy chọn D-H
- Diffie-Hellman vô danh
  - Không xác thực, có thể bị tấn công người ở giữa

Đặc tả mật mã (1)

- Giải thuật mã hóa
  - RC4, RC2, DES, 3DES, DES40, IDEA, hoặc Fortezza
- Giải thuật MAC
  - MD5 hoặc SHA-1
- Kiểu mã hóa
  - Luồng hay khối
- Có thể xuất khẩu
  - Đúng hay sai

Đặc tả mật mã (2)

- Kích thước giá trị băm
  - 0, 16 (với MD5), hoặc 20 (với SHA-1) byte
- Chất liệu khóa
  - Chuỗi các byte dùng để sinh các khóa viết
- Kích thước IV
  - Kích thước Initialization Vector cho mã hóa CBC (Cipher Block Chaining)

Thông báo certificate

- Chứa một hay một chuỗi các chứng thực X.509
- Server gửi chứng thực của nó nếu nó cần được xác thực
  - Yêu cầu với bất kỳ phương pháp trao đổi khóa nào ngoại trừ Diffie-Hellman vô danh
  - Thay cho thông báo server_key_exchange trong Diffie-Hellman cố định
- Client gửi certificate nếu server yêu cầu

Thông báo server_key_exchange

- Cần thiết trong các trường hợp sau
  - Diffie-Hellman vô danh
    - Chứa 2 giá trị Diffie-Hellman tổng thể và khóa công khai Diffie-Hellman của server
  - Diffie-Hellman tức thời
    - Chứa 3 thông số Diffie-Hellman như Diffie-Hellman vô danh cùng với một chữ ký cho các thông số
  - RSA với khóa RSA của server chỉ dùng để ký
    - Chứa khóa công khai RSA tạm thời được ký của server
  - Fortezza

Xác thực server

- Chữ ký được tạo ra bằng cách mã hóa giá trị băm với khóa riêng của bên gửi
  - hash(ClientHello.random ‖ ServerHello.random ‖ ServerParams)
    - Bao trùm không chỉ các tham số Diffie-Hellman hay RSA mà cả các giá trị nonce trong các thông báo hello để chống tấn công lặp lại
    - Sử dụng SHA-1 với chữ ký DSS
    - Ghép 2 giá trị băm MD5 và SHA-1 với chữ ký RSA

Thông báo certificate_request

- Có thể được gửi bởi 1 server không vô danh
  - Không sử dụng Diffie-Hellman vô danh
- Bao gồm 2 tham số
  - certificate_type
    - Chỉ ra giải thuật khóa công khai và chế độ sử dụng
  - certificate_authorities
    - Danh sách tên các cơ quan chứng thực có thể chấp nhận được

Thông báo client_key_exchange

- RSA
  - Client sinh ra một bí mật pre-master và mã hóa nó với khóa công khai RSA
    - Bí mật pre-master dùng để tính bí mật master
- Diffie-Hellman tức thời hoặc vô danh
  - Chứa các thông số Diffie-Hellman của client
- Diffie-Hellman cố định
  - Thông báo rỗng

Thông báo certificate_verify

- Để xác minh client có thật sự sở hữu khóa riêng tương ứng với chứng thực client không
  - Chỉ được gửi khi chứng thực client có khả năng ký
    - Tất cả các chứng thực ngoại trừ chứng thực chứa các tham số Diffie-Hellman cố định
- Ký mã băm dựa trên các thông báo trước đó (handshake_messages) và master_secret
  - Sử dụng SHA-1 với chữ ký DSS
  - Ghép 2 giá trị băm MD5 và SHA-1 với chữ ký RSA

Giai đoạn kết thúc

- Thông báo change_cipher_spec khiến bên nhận sao chép CipherSpec treo vào CipherSpec hiện thời
  - Được gửi với giao thức Change Cipher Spec
- Các thông báo finished là ghép 2 giá trị băm MD5 và SHA-1 dựa trên handshake_messages, master_secret và Sender (client hay server)
  - Áp dụng các giải thuật, khóa và bí mật mới

Tạo master_secret

- Trao đổi pre_master_secret
  - RSA
  - Diffie-Hellman
- Tính toán master_secret
  - Theo cùng công thức trên client và server dựa trên
    - pre_master_secret
    - ClientHello.random
    - ServerHello.random

Sinh các tham số mật mã

- Từ master_secret
  - Theo thứ tự
    - Bí mật MAC viết của client, bí mật MAC viết của server, khóa viết của client, khóa viết của server, IV viết của client, IV viết của server
  - Bằng cách băm master_secret thành chuỗi các byte đủ độ dài cho tất cả các tham số
    - key_block = MD5(master_secret ‖ SHA('A' ‖ master_secret ‖ ServerHello.random ‖ ClientHello.random)) ‖ MD5(master_secret ‖ SHA('BB' ‖ master_secret ‖ ServerHello.random ‖ ClientHello.random)) ‖ …

Chương 6

AN TOÀN THƯ ĐIỆN TỬ

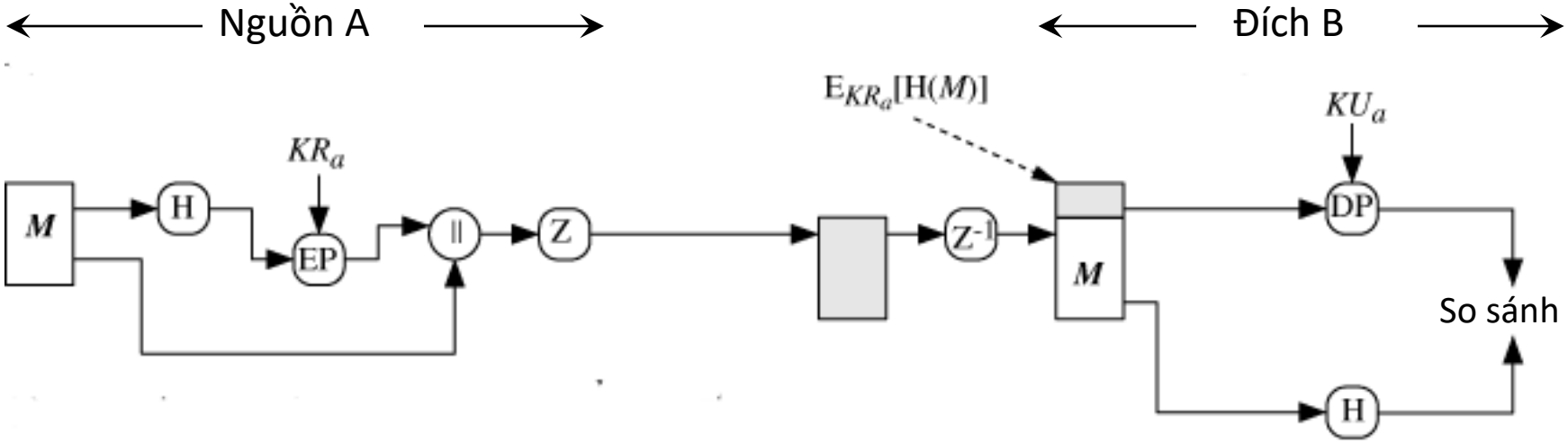
Giới thiệu

- 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

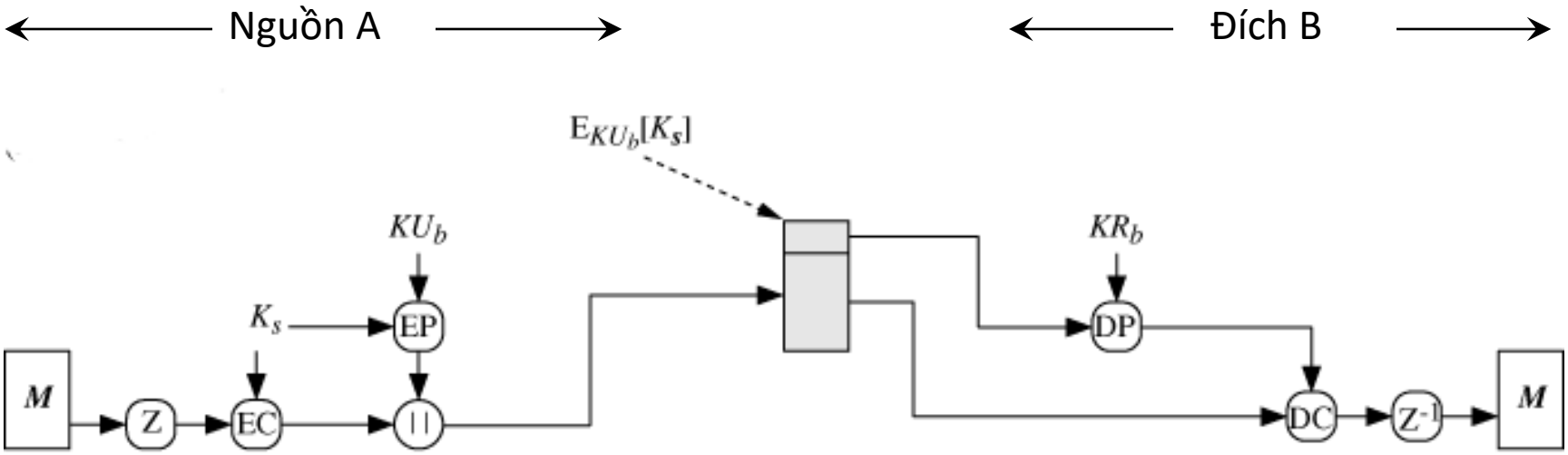
- Do Phil Zimmermann phát triển vào năm 1991
- Chương trình miễn phí, chạy trên nhiều môi trường khác nhau (phần cứng, hệ điều hành)
  - Có phiên bản thương mại nếu cần hỗ trợ kỹ thuật
- Dựa trên các giải thuật mật mã an ninh nhất
- Chủ yếu ứng dụng cho thư điện tử và file
- Độc lập với các tổ chức chính phủ
- Bao gồm 4 dịch vụ : xác thực, bảo mật, nén, tương thích thư điện tử
  - Hai dịch vụ sau trong suốt đối với người dùng

Xác thực của PGP



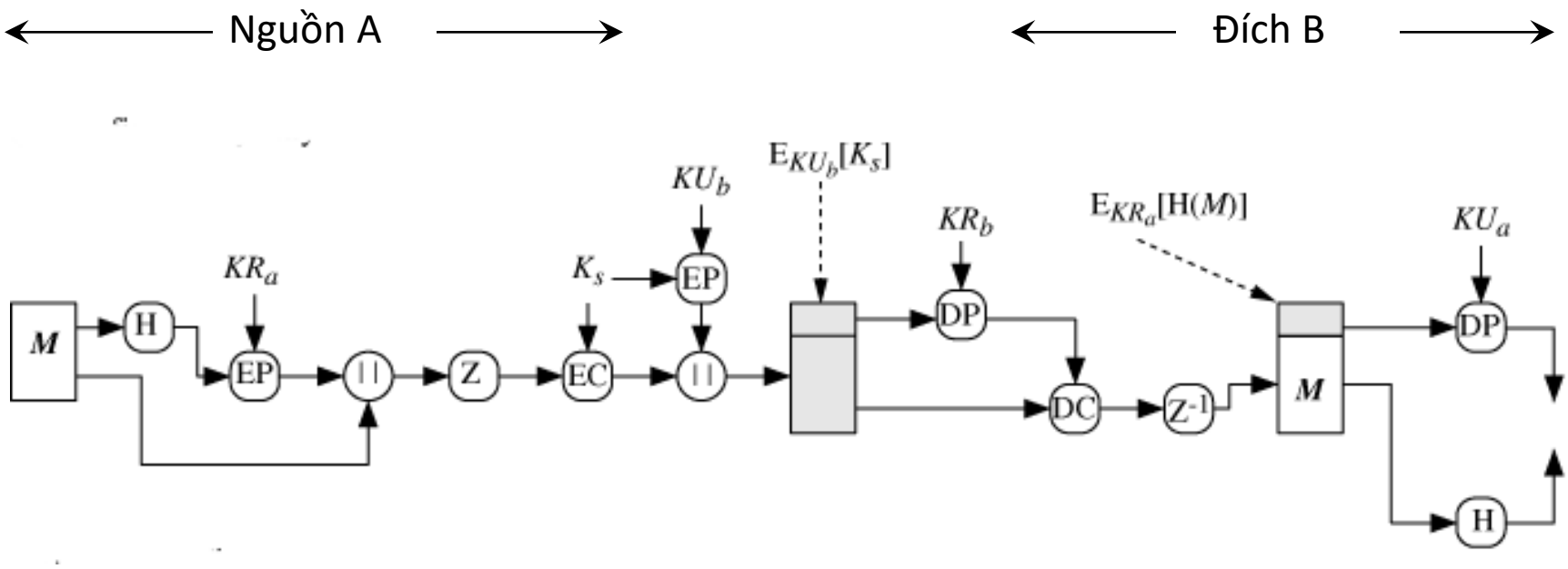
M = Thông báo gốc EP = Mã hóa khóa công khai
H = Hàm băm DP = Giải mã khóa công khai
‖ = Ghép KR_a = Khóa riêng của A
Z = Nén KU_a = Khóa công khai của A
Z^{-1} = Cởi nén

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

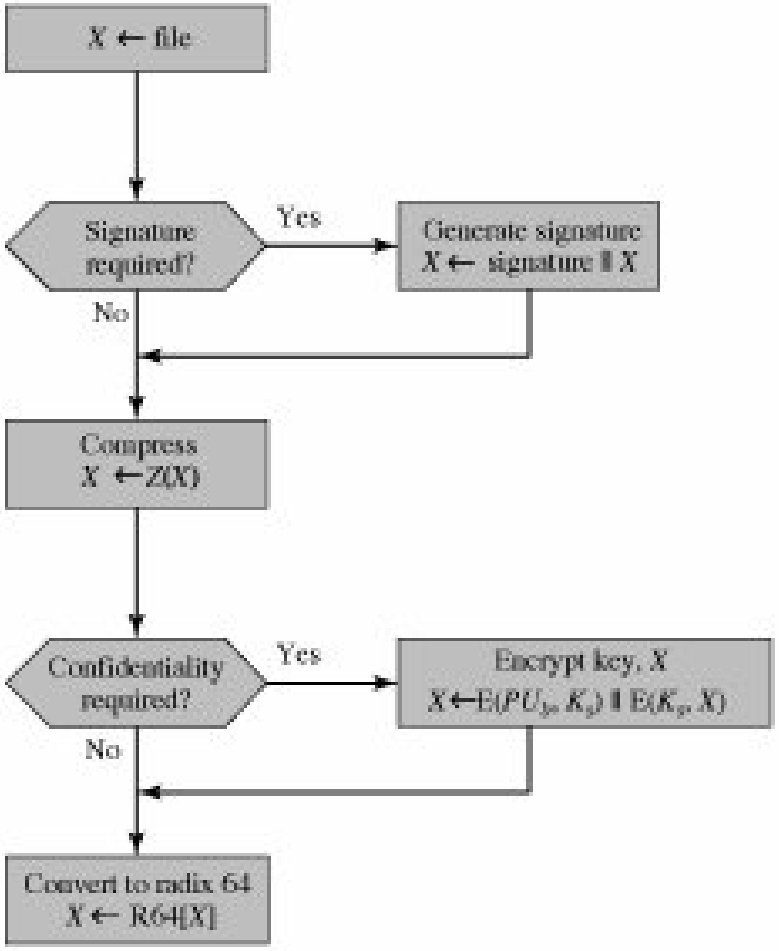
Tương thích thư điện tử của PGP

- 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

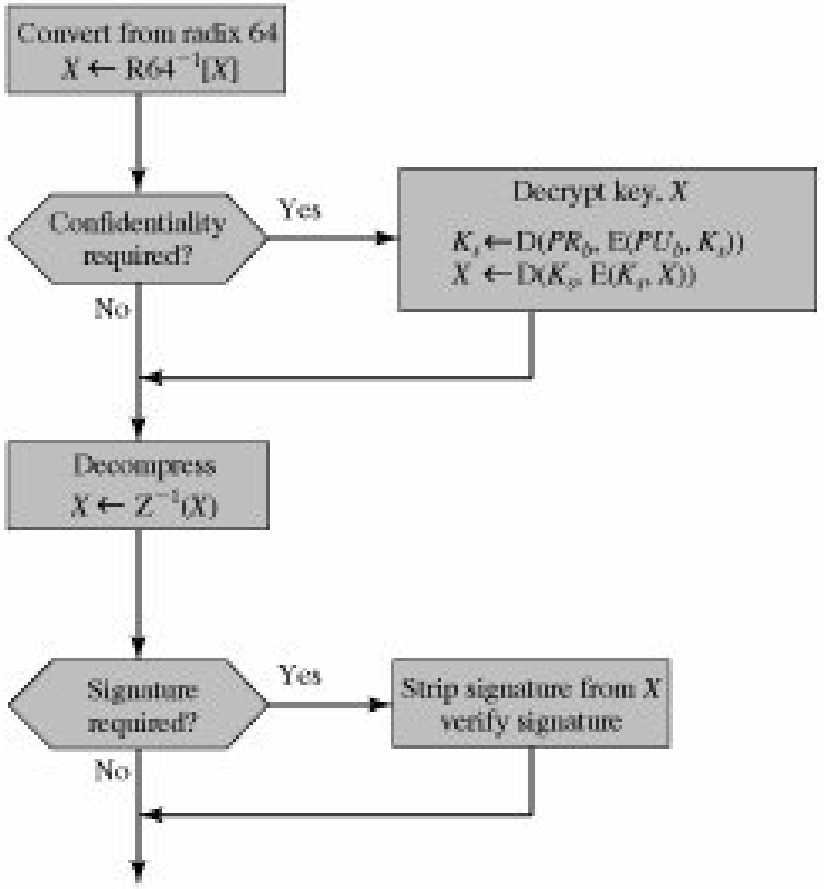
Bảng chuyển đổi cơ số 64

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)	=

Sơ đồ xử lý PGP



(a) Generic transmission diagram (from A)



(b) Generic reception diagram (to B)

Khóa phiên PGP

- Cần sử dụng một khóa phiên cho mỗi thông báo
  - Độ dài 56 bit với DES, 128 bit với CAST-128 và IDEA, 168 bit với 3DES
- Cách thức sinh khóa phiên cho CAST-128
  - Sử dụng chính CAST-128 theo phương thức CFB
  - Từ một khóa 128 bit và 2 khối nguyên bản 64 bit sinh ra 2 khối bản mã 64 bit tạo thành khóa phiên 128 bit
  - Hai khối nguyên bản đầu vào được sinh ngẫu nhiên dựa vào chuỗi các phím gõ từ người dùng
  - Khóa đầu vào được sinh từ các khối nguyên bản đầu vào và khóa phiên đầu ra trước đó

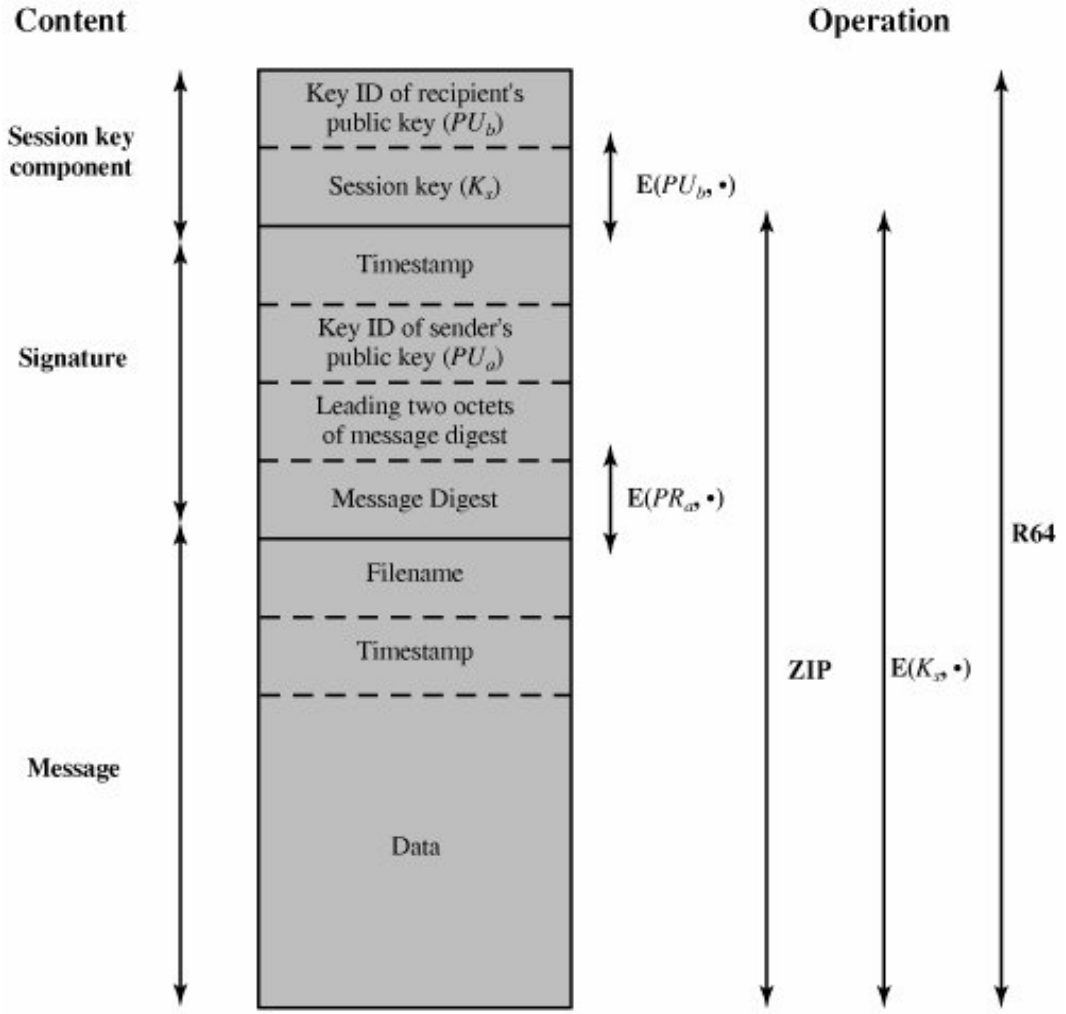
Khóa công khai/khóa riêng PGP

- Người dùng có thể có nhiều cặp khóa công khai/khóa riêng
  - Nhu cầu thay đổi cặp khóa hiện thời
  - Giao tiếp với nhiều nhóm đối tác khác nhau
  - Hạn chế lượng thông tin mã hóa với mỗi khóa để nâng cao độ an toàn
- Cần chỉ ra khóa công khai nào được sử dụng để mã hóa khóa phiên
- Cần chỉ ra chữ ký của bên gửi tương ứng với khóa công khai nào

Định danh khóa công khai PGP

- Để chỉ ra mã công khai nào được sử dụng có thể truyền khóa công khai cùng với thông báo
  - Không hiệu quả
    - Khóa công khai RSA có thể dài hàng trăm chữ số thập phân
- Định danh gắn với mỗi khóa công khai là 64 bit trọng số nhỏ nhất của nó
  - ID của KU_a = KU_a mod 2^{64}
  - Xác suất cao là mỗi khóa công khai có một định danh duy nhất

Khuôn dạng thông báo PGP



Vòng khóa PGP

- Mỗi người dùng PGP có hai vòng khóa
  - Vòng khóa riêng chứa các cặp khóa công khai/khóa riêng của người dùng hiện thời
    - Có thể được chỉ mục bởi định danh khóa công khai (**Key ID**) hoặc định danh người dùng (**User ID**)
    - Khóa riêng được mã hóa sử dụng khóa là giá trị băm của mật khẩu nhập trực tiếp từ người dùng
  - Vòng khóa công khai chứa các khóa công khai của những người dùng quen biết với người dùng hiện thời
    - Có thể được chỉ mục bởi định danh khóa công khai hoặc định danh người dùng

Cấu trúc các vòng khóa PGP

Private-Key Ring

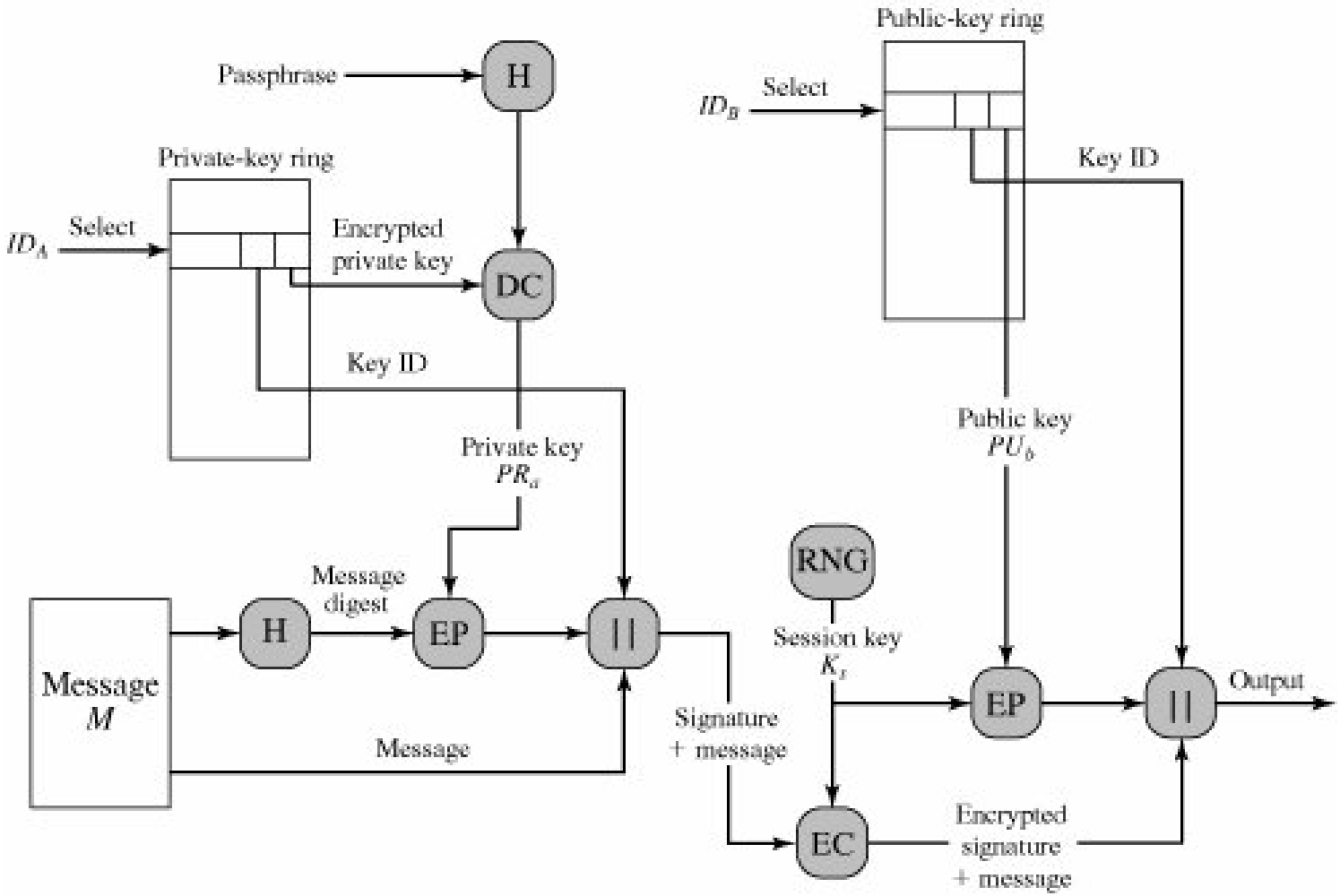
Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
. . .	. . .	. . .	. . .	. . .
T_i	$PU_i \bmod 2^{64}$	PU_i	$E(H(P_i), PR_i)$	User i
. . .	. . .	. . .	. . .	. . .

Public-Key Ring

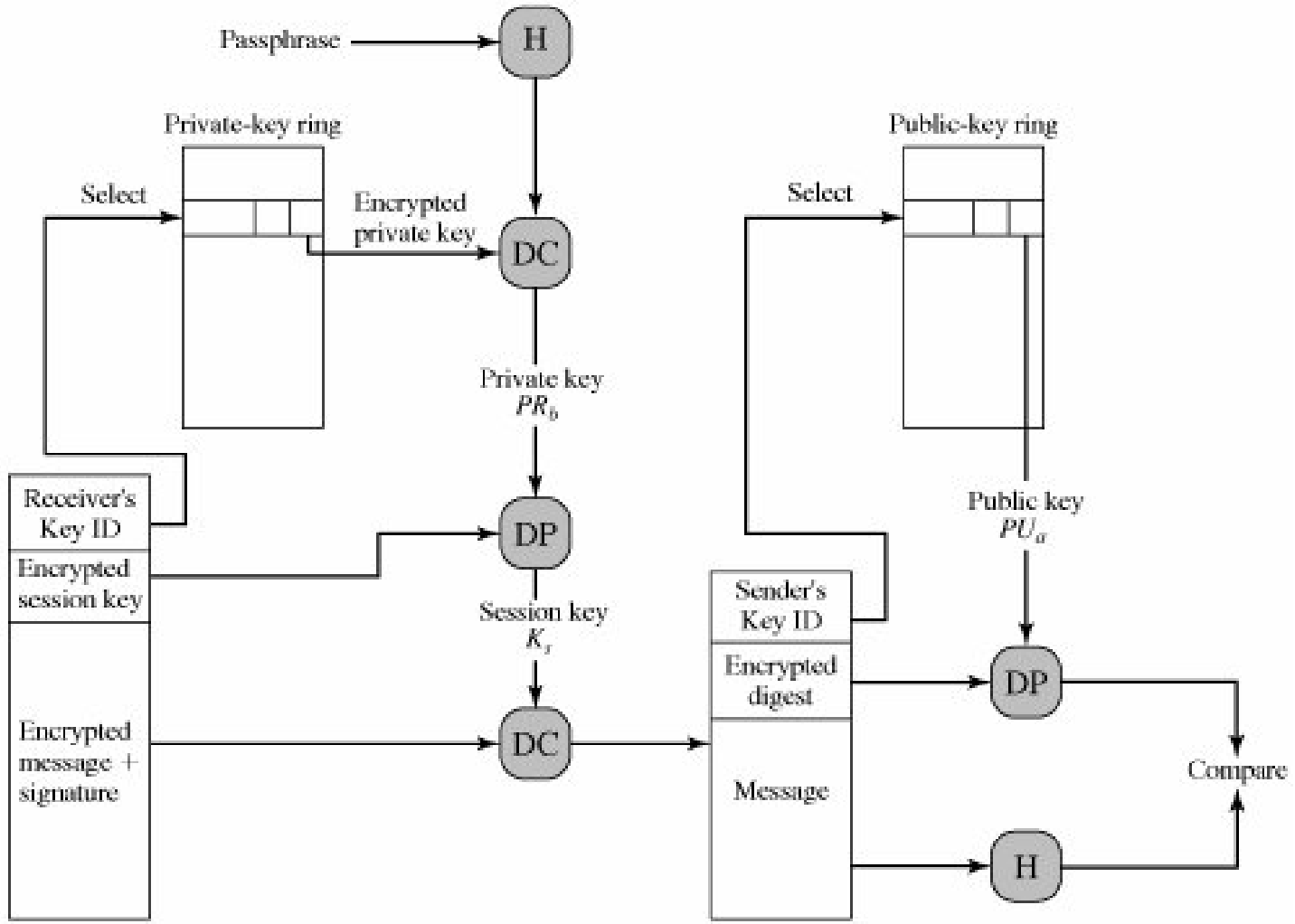
Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
T_i	$PU_i \bmod 2^{64}$	PU_i	$trust\_flag_i$	User i	$trust\_flag_i$		
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .

* = field used to index table

Sơ đồ tạo thông báo PGP



Sơ đồ nhận thông báo PGP



Quản lý khóa PGP

- Thay vì dựa trên các CA (cơ quan chứng thực), đối với PGP mỗi người dùng là một CA
  - Có thể ký cho những người dùng quen biết trực tiếp
- Tạo nên một mạng lưới tin cậy
  - Tin các khóa đã được chính bản thân ký
  - Có thể tin các khóa những người dùng khác ký nếu có một chuỗi các chữ ký tới chúng
- Mỗi khóa có một chỉ số tin cậy
- Các người dùng có thể thu hồi khóa của họ

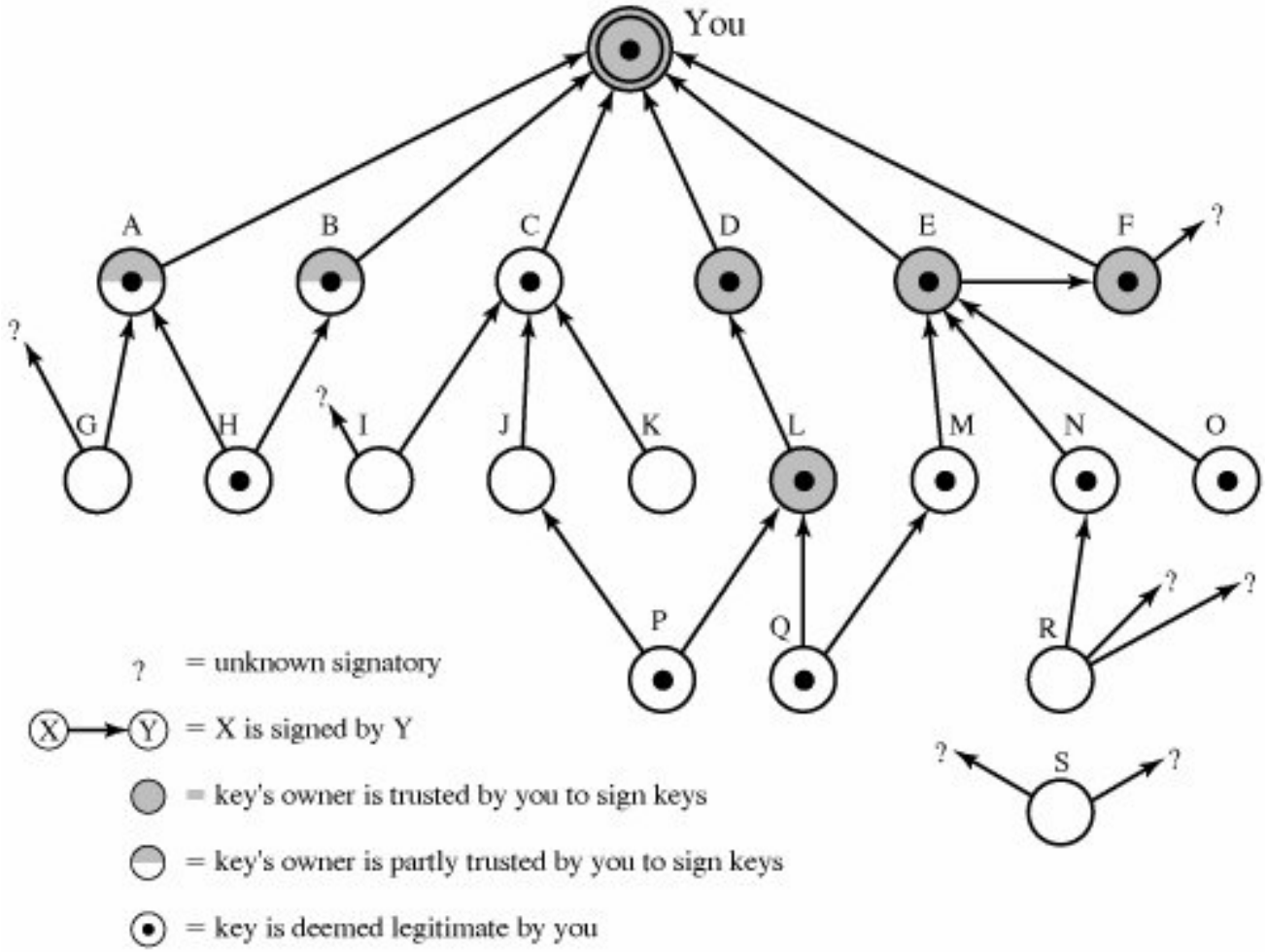
Mô hình tin cậy PGP (1)

- Với mỗi khóa công khai người dùng ấn định độ tin cậy vào chủ nhân của nó trong trường **Owner trust**
  - Giá trị *ultimate trust* được tự động gán nếu khóa công khai có trong vòng khóa riêng
  - Giá trị người dùng có thể gán là *unknown*, *untrusted*, *marginally trusted*, hay *completely trusted*
- Giá trị các trường **Signature trust** được sao chép từ các trường **Owner trust** tương ứng
  - Với điều kiện khóa công khai của người dùng tương ứng có **Key legitimacy** là *complete*
  - Nếu không có thì được gán giá trị *unknown user*

Mô hình tin cậy PGP (2)

- Xác định giá trị của trường **Key legitimacy**
  - Nếu khóa công khai có ít nhất một chữ ký với giá trị **Signature trust** là *ultimate* thì **Key legitimacy** là *complete*
  - Nếu không, **Key legitimacy** được tính bằng tổng có trọng số các giá trị **Signature trust**
    - Các chữ ký *completely trusted* có trọng số là 1/*X*
    - Các chữ ký *marginally trusted* có trọng số là 1/*Y*
    - *X* và *Y* là các tham số do người dùng xác định
    - Nếu tổng số đạt hoặc vượt ngưỡng 1 thì **Key legitimacy** được gán giá trị *complete*

Ví dụ mô hình tin cậy PGP



Thu hồi khóa công khai

- Lý do thu hồi khóa công khai
  - Địch thủ biết nguyên bản khóa riêng
  - Địch thủ biết bản mã khóa riêng và mật khẩu
  - Tránh sử dụng cùng một khóa trong một thời gian dài
- Quy trình thu hồi khóa công khai
  - Chủ sở hữu phát hành chứng thực thu hồi khóa
    - Cùng khuôn dạng như chứng thực bình thường nhưng bao gồm chỉ dấu thu hồi khóa công khai
    - Chứng thực được ký với khóa riêng tương ứng khóa công khai cần thu hồi
  - Mau chóng phát tán chứng thực một cách rộng rãi để các đối tác kịp thời cập nhật vòng khóa công khai

Chương 7

AN TOÀN IP

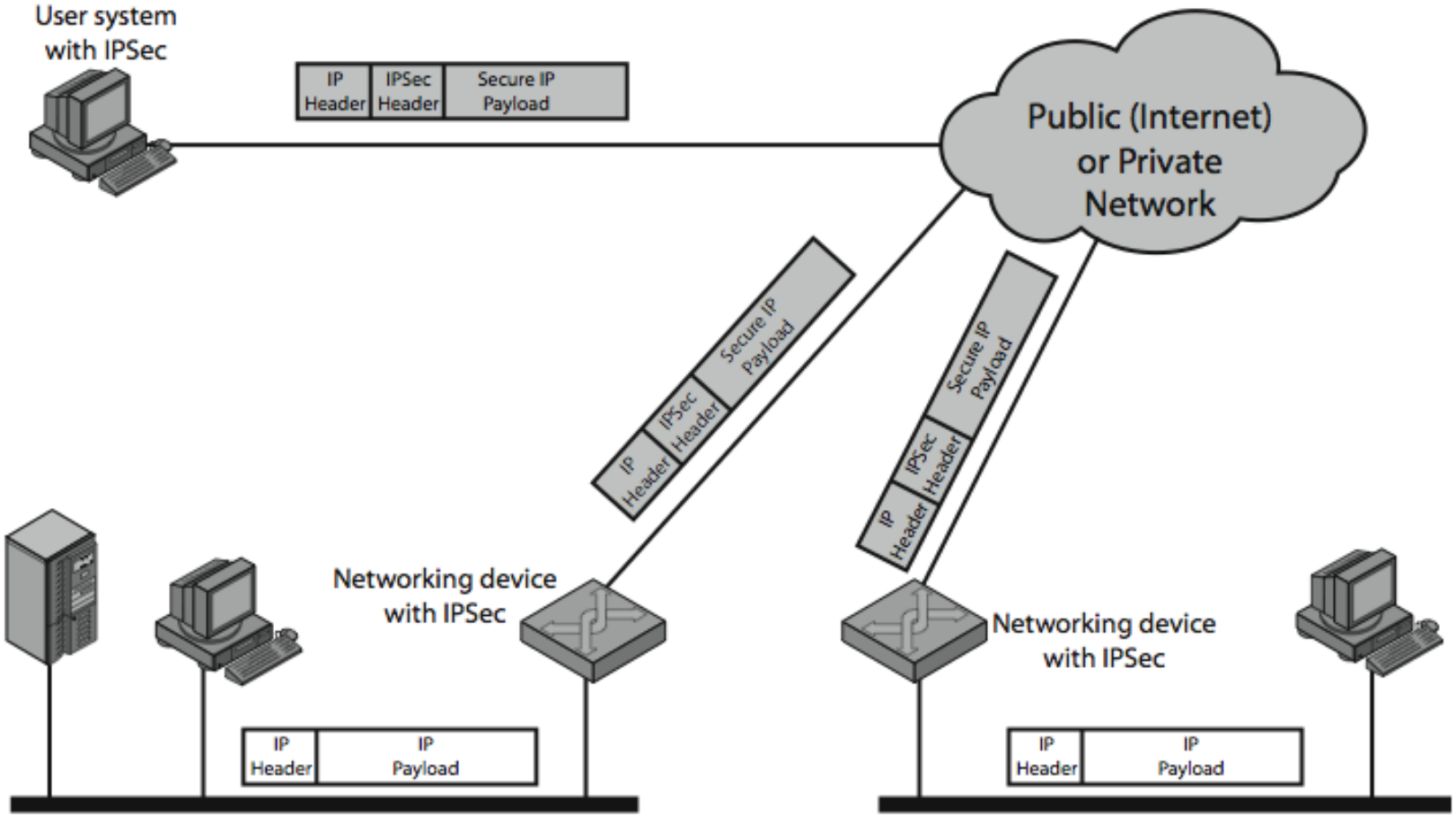
Giới thiệu

- Lý do cần IPSec
  - Có những vấn đề an ninh 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 ninh ở mức IP sẽ đảm bảo an ninh cho tất cả các ứng dụng
    - Bao gồm nhiều ứng dụng chưa có tính năng an ninh
- Các cơ chế an ninh 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 ninh thương mại điện tử
  - Hỗ trợ thêm cho các giao thức an ninh 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 ninh 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 ninh kiến trúc định tuyến

Kiến trúc an ninh IP

- Đặc tả IPSec khá phức tạp
- Định nghĩa trong nhiều tài liệu
  - Kiến trúc (RFC 4301), Authentication Header (RFC 4302), Encapsulating Security Payload (RFC 4303), Internet Key Exchange (RFC 4306)
    - AH không còn được sử dụng trong các ứng dụng mới
  - Các tài liệu mô tả các giải thuật mật mã
    - Mã hóa, xác thực thông báo, hàm giả ngẫu nhiên, trao đổi khóa
  - Các tài liệu khác
    - Chính sách an ninh và cơ sở thông tin quản lý (MIB)
- Việc hỗ trợ IPSec là bắt buộc đối với IPv6, tùy chọn đối với IPv4

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 ninh

- Khái niệm liên kết an ninh (SA)
  - Quan hệ một chiều từ bên gửi đến bên nhận, cho biết các dịch vụ an ninh áp dụng cho thông tin lưu chuyển
- Mỗi SA được xác định duy nhất bởi 3 tham số
  - Chỉ mục các tham số an ninh (SPI)
    - Chỉ có ý nghĩa cục bộ
    - Cho phép bên nhận chọn SA tương ứng xử lý gói tin
  - Địa chỉ IP đích
    - Hiện chỉ cho phép địa chỉ đơn phát
  - Định danh giao thức an ninh
    - Chỉ ra đây là AH hay ESP

Các tham số SA

- IPSec được cài đặt kèm theo CSDL liên kết an ninh (SAD)
- Các tham số liên kết an ninh
  - Bộ đếm số thứ tự
  - Cờ tràn số thứ tự
  - Cửa sổ chống lặp lại
  - Thông tin AH
  - Thông tin ESP
  - Tuổi thọ liên kết an ninh
  - Chế độ giao thức IPSec
  - MTU đường dẫn

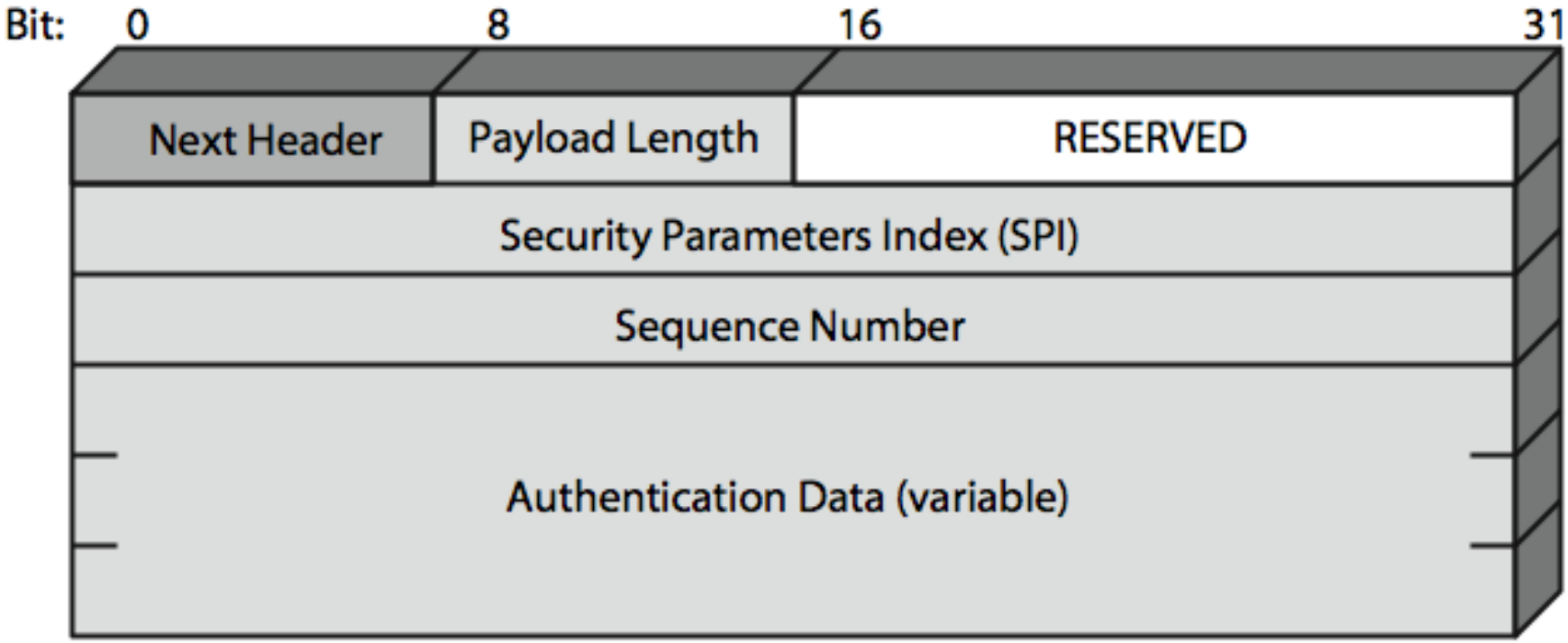
Các chọn lọc SA

- IPSec cho phép một luồng IP gửi đi phải áp dụng nhiều SA kết hợp hoặc miễn trừ
  - Cách thức gắn kết luồng IP với các SA được định nghĩa trong CSDL chính sách an ninh (SPD)
    - Nhiều đề mục SPD có thể gắn với cùng một SA
    - Một đề mục SPD có thể gắn với nhiều SA
  - Mỗi đề mục SPD có các chọn lọc luồng IP như sau
    - Địa chỉ IP đích
    - Địa chỉ IP nguồn
    - Định danh người dùng
    - Mức độ nhạy cảm dữ liệu
    - Giao thức tầng giao vận
    - Các cổng nguồn và đích

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
  - 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

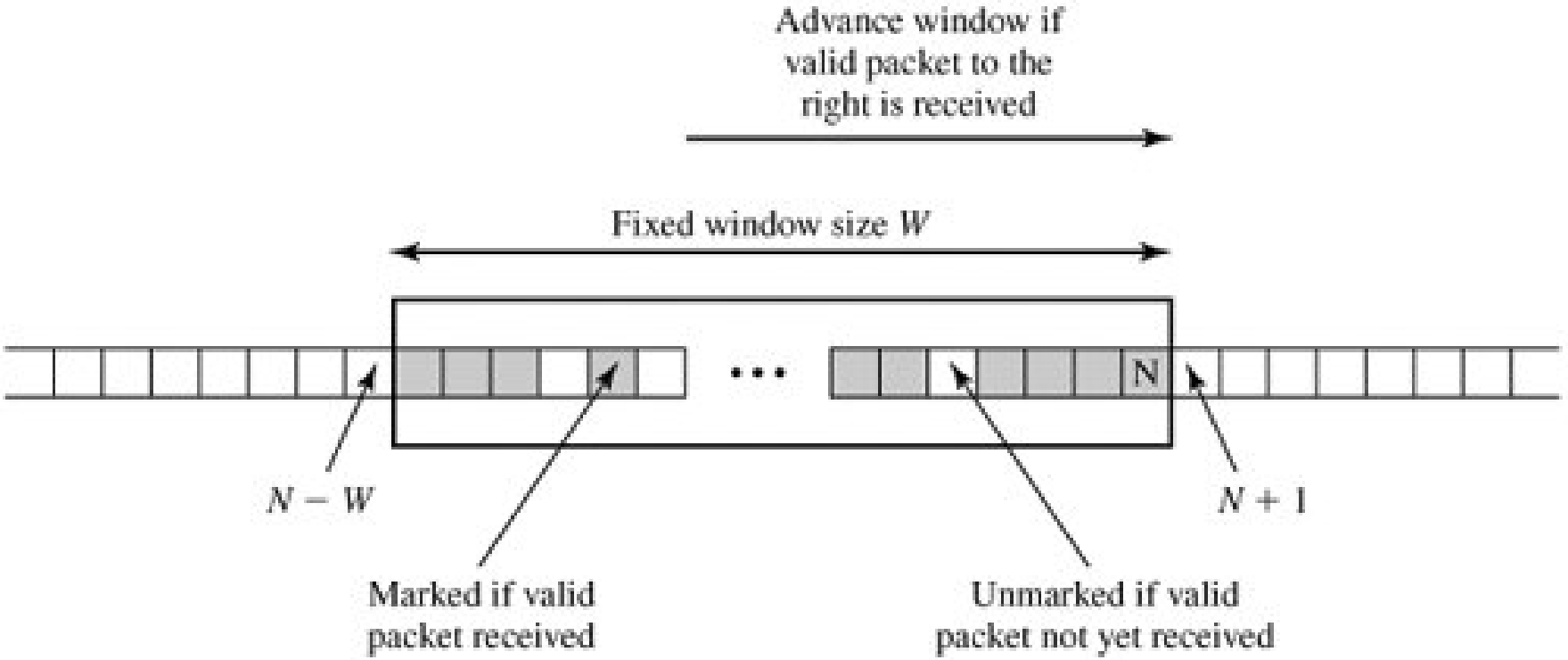
Khuôn dạng AH



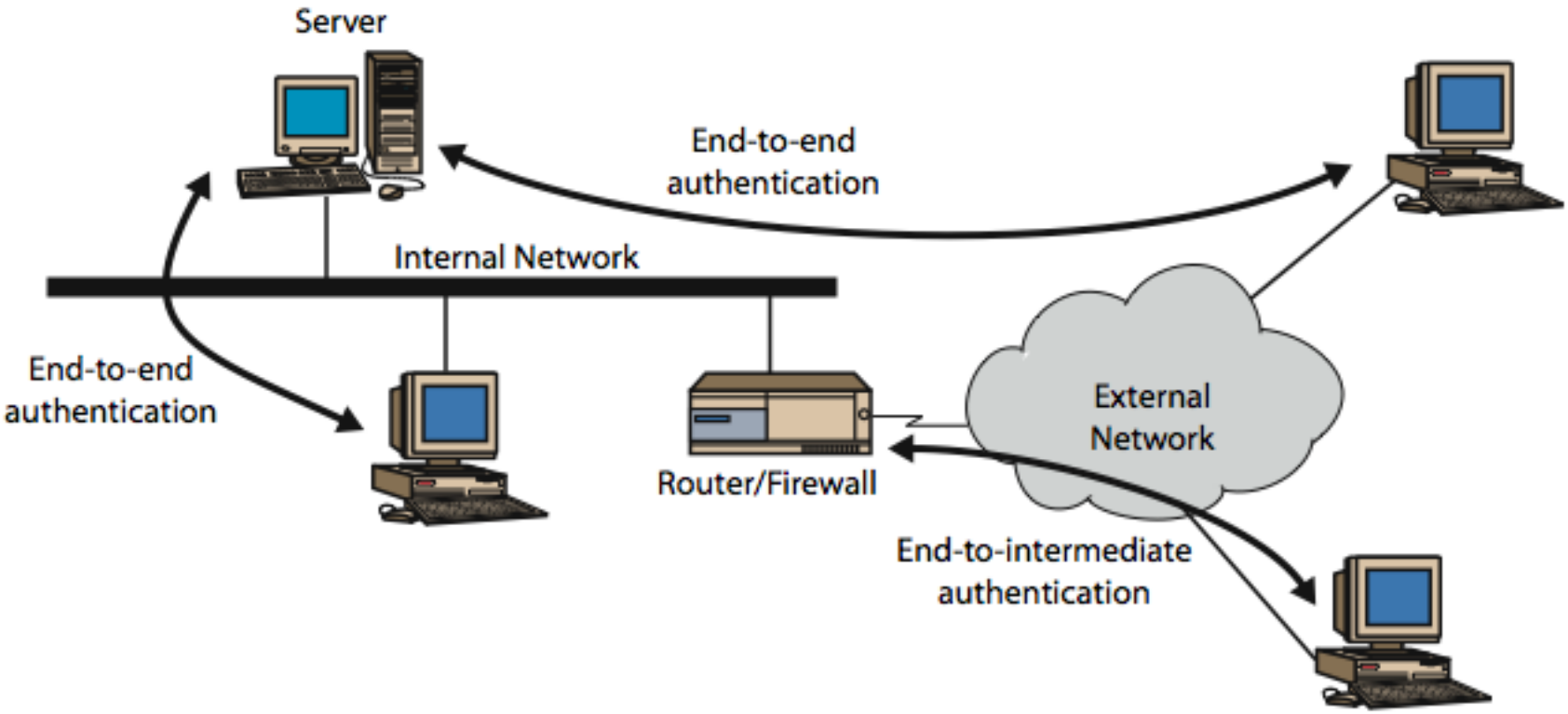
Dịch vụ chống lặp lại

- Bên gửi
  - Khởi tạo bộ đếm số thứ tự ở giá trị 0
  - Mỗi lần gửi một gói tin mới tăng bộ đếm lên 1 và đặt giá trị bộ đếm vào trường số thứ tự của gói tin
  - Nếu bộ đếm đạt tới $2^{32} – 1$ thì kết thúc SA hiện thời
- Bên nhận
  - Cài đặt một cửa sổ chống lặp lại độ dài *W*, với ô tận cùng bên phải là số thứ tự *N* lớn nhất nhận được
    - Nếu gói tin nằm trong cửa sổ, mới, và MAC hợp lệ thì ô tương ứng được đánh dấu
    - Nếu gói tin nằm bên phải cửa sổ và MAC hợp lệ thì dịch cửa sổ về tận cùng bên phải mới và đánh dấu ô đó

Minh họa chống lặp lại



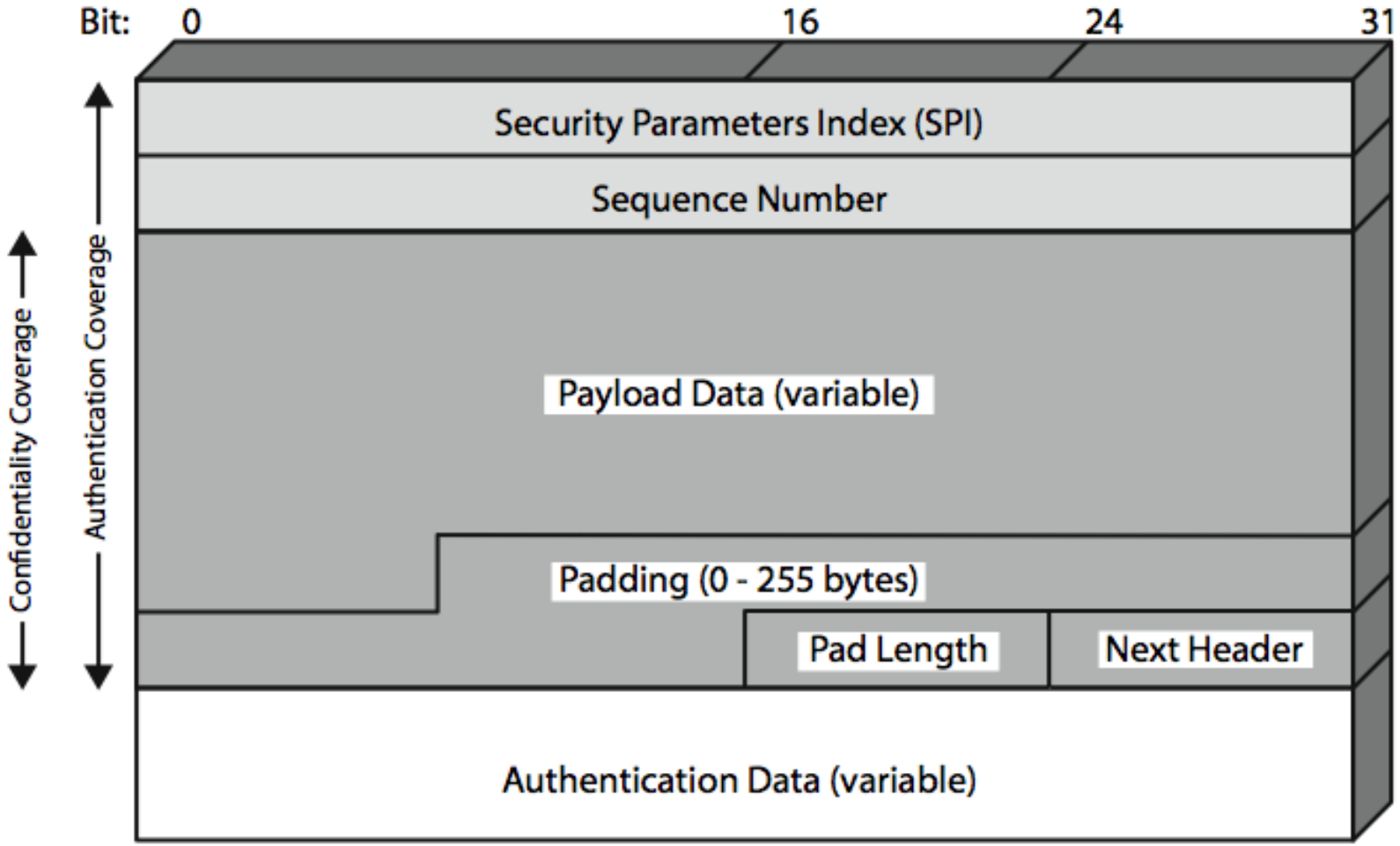
Chế độ giao vận và đường hầm



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



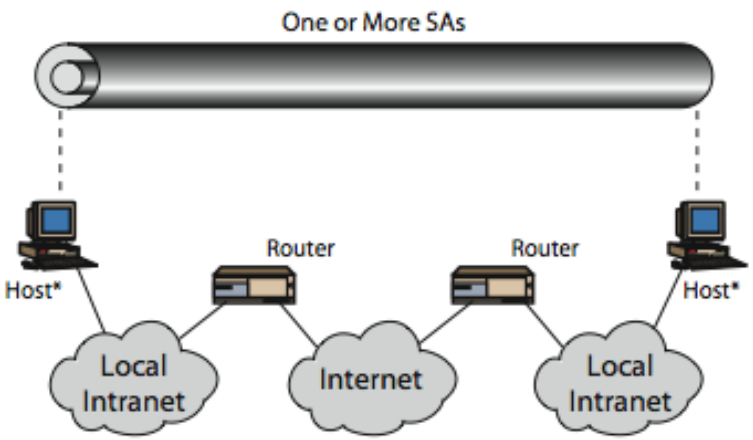
Giao vận và đường hầm ESP

- 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
  - 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

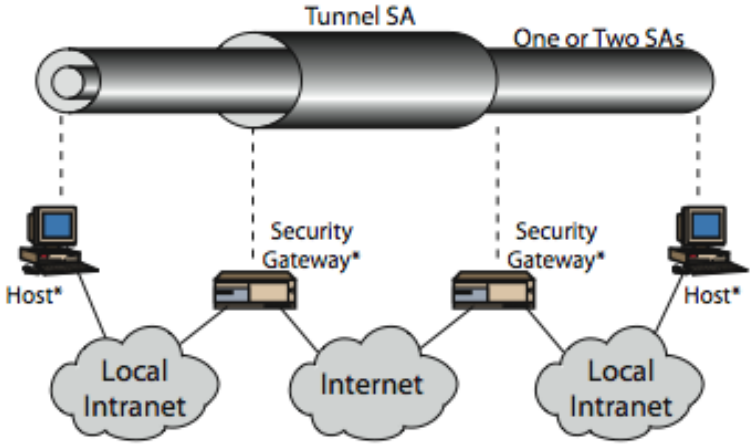
Kết hợp các liên kết an ninh

- 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 ninh
  - 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

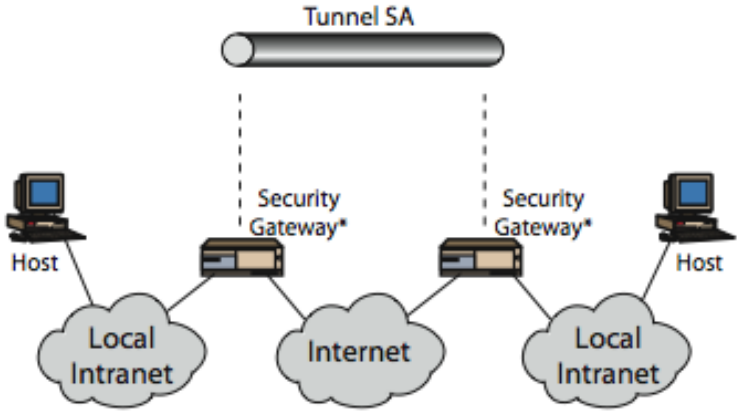
Ví dụ kết hợp các SA



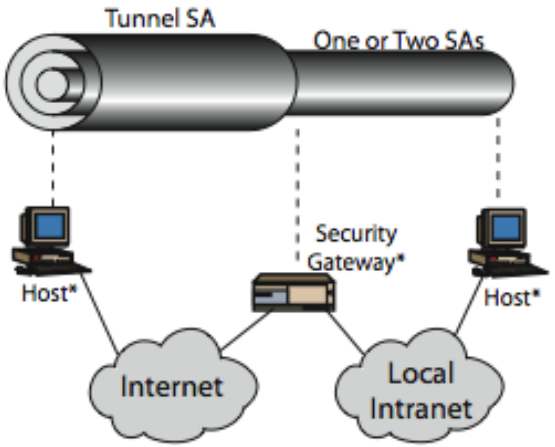
(a) Case 1



(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

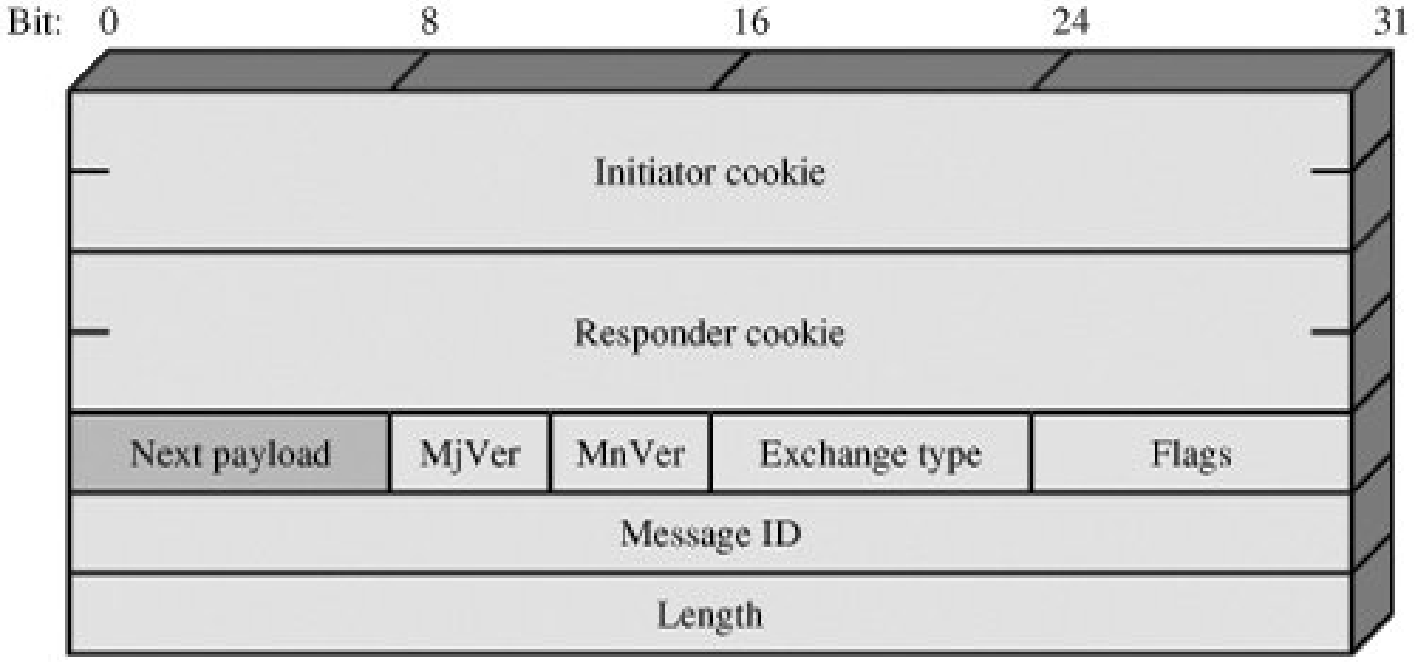
Oakley

- 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

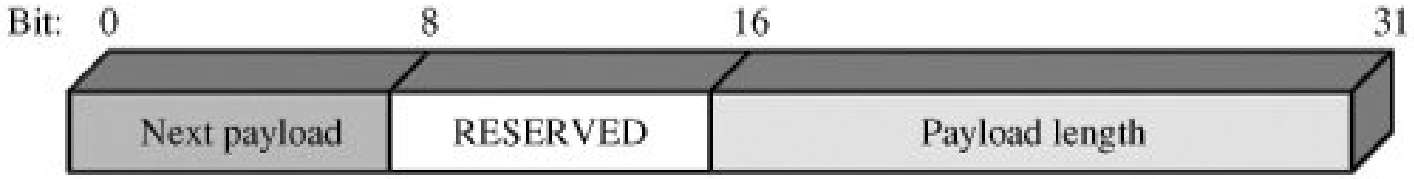
ISAKMP

- 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 ninh
- 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