

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
CƠ SỞ TP. HỒ CHÍ MINH**

**BÀI GIẢNG**

**BẢO MẬT HỆ THỐNG THÔNG TIN**

**DÀNH CHO HỆ ĐÀO TẠO TỪ XA**

**Biên soạn: Lê Phúc**

**Tháng 7/2007**

## MỞ ĐẦU

Tài liệu này được xây dựng với mục đích giúp sinh viên hệ đào tạo từ xa nghiên cứu các vấn đề về bảo mật hệ thống thông tin. Bảo mật hệ thống thông tin là tập các kỹ thuật, dịch vụ, cơ chế và ứng dụng phụ trợ giúp triển khai các hệ thống thông tin với độ an toàn cao nhất, mà cụ thể là để bảo vệ ba đặc trưng cơ bản của một hệ thống an toàn là *tính Bí mật, tính Toàn vẹn* và *tính Khả dụng* của thông tin.

Tính bảo mật của hệ thống là vấn đề được cân nhắc ngay khi thiết kế hệ thống và được thực hiện xuyên suốt trong quá trình thi công, vận hành và bảo dưỡng hệ thống. Trong thời điểm mà việc kết nối vào mạng Internet, nơi chứa rất nhiều nguy cơ tấn công tiềm ẩn, đã trở thành một nhu cầu sống còn của các hệ thống thông tin thì vấn đề bảo mật càng cần phải được quan tâm và đầu tư đúng mức.

Tài liệu này nhắm đến đối tượng sinh viên là những người vừa học vừa làm, do đó các vấn đề bảo mật thực tế trên mạng được quan tâm nhiều hơn là các cơ sở lý thuyết. Các chuyên đề về mật mã cũng được trình bày đơn giản theo cách nhìn của người sử dụng, không quá chuyên sâu về cơ sở toán học, do đó, nếu có nhu cầu tìm hiểu sâu hơn hoặc chứng minh các thuật toán, sinh viên cần phải đọc thêm các tài liệu về lý thuyết số.

Nội dung tài liệu được chia thành 3 chương:

**-Chương 1:** *Tổng quan về bảo mật hệ thống thông tin*, trình bày các vấn đề chung về bảo mật và an toàn hệ thống, các nguy cơ và các phương thức tấn công vào hệ thống thông tin, các ứng dụng bảo vệ hệ thống thông tin đang được sử dụng như Firewall và IDS...

**-Chương 2:** *Mật mã và xác thực thông tin*, trình bày các cơ chế mật mã và xác thực nhằm đảm bảo tính Bí mật và Toàn vẹn của thông tin. Phần này mô tả nguyên lý của các thuật toán mật mã thông dụng, hàm băm, chữ ký số và các vấn đề quản lý khoá.

**-Chương 3:** *Các ứng dụng bảo mật trong hệ thống thông tin*, trình bày các ứng dụng thực tế như các giao thức xác thực, bảo mật trong kết nối mạng với IPSec, bảo mật trong ứng dụng Internet với SSL và SET.

Cuối mỗi chương đều có phần tóm tắt, các câu hỏi trắc nghiệm và bài tập, giúp sinh viên hệ thống hoá lại kiến thức đã học. Đặc biệt, các bài tập thực hành và lập trình sẽ giúp sinh viên nắm rõ hơn phần lý thuyết, nên cố gắng thực hiện các bài tập này một cách chu đáo.

Hy vọng tài liệu này sẽ ít nhiều giúp ích cho việc nghiên cứu chuyên đề an toàn hệ thống thông tin của các bạn sinh viên.

Tháng 7/2007.

Tác giả.

# CHƯƠNG I

## TỔNG QUAN VỀ BẢO MẬT HỆ THỐNG THÔNG TIN

### Giới thiệu:

Chương này giúp học viên nắm được các khái niệm thường dùng trong bảo mật và an toàn hệ thống, nguyên tắc xây dựng một hệ thống thông tin bảo mật, nhận diện và phân tích các nguy cơ và rủi ro đối với hệ thống thông tin, từ đó có kế hoạch nâng cấp và bảo vệ hệ thống.

Nội dung chương này gồm các phần như sau:

- Các đặc trưng của một hệ thống bảo mật.
- Nguy cơ và rủi ro đối với hệ thống thông tin.
- Các khái niệm dùng trong bảo mật hệ thống
- Chiến lược bảo mật hệ thống AAA.
- Một số hình thức xâm nhập hệ thống.
- Kỹ thuật ngăn chặn và phát hiện xâm nhập.

### I.1 TỔNG QUAN

Vấn đề bảo đảm an toàn cho các hệ thống thông tin là một trong những vấn đề quan trọng cần cân nhắc trong suốt quá trình thiết kế, thi công, vận hành và bảo dưỡng hệ thống thông tin.

Cũng như tất cả các hoạt động khác trong đời sống xã hội, từ khi con người có nhu cầu lưu trữ và xử lý thông tin, đặc biệt là từ khi thông tin được xem như một bộ phận của tư liệu sản xuất, thì nhu cầu bảo vệ thông tin càng trở nên bức thiết. Bảo vệ thông tin là bảo vệ *tính bí mật* của thông tin và *tính toàn vẹn* của thông tin. Một số loại thông tin chỉ còn ý nghĩa khi chúng được giữ kín hoặc giới hạn trong một số các đối tượng nào đó, ví dụ như thông tin về chiến lược quân sự chẳng hạn. Đây là tính bí mật của thông tin. Hơn nữa, thông tin không phải luôn được con người ghi nhớ do sự hữu hạn của bộ óc, nên cần phải có thiết bị để lưu trữ thông tin. Nếu thiết bị lưu trữ hoạt động không an toàn, thông tin lưu trữ trên đó bị mất đi hoặc sai lệch toàn bộ hay một phần, khi đó tính toàn vẹn của thông tin không còn được bảo đảm.

Khi máy tính được sử dụng để xử lý thông tin, hiệu quả xử lý thông tin được nâng cao lên, khối lượng thông tin được xử lý càng ngày càng lớn lên, và kéo theo nó, tầm quan trọng của thông tin trong đời sống xã hội cũng tăng lên. Nếu như trước đây, việc bảo vệ thông tin chỉ chú trọng vào vấn đề dùng các cơ chế và phương tiện vật lý để bảo vệ thông tin theo đúng nghĩa đen của từ này, thì càng về sau, vấn đề bảo vệ thông tin đã trở nên đa dạng hơn và phức tạp hơn. Có thể kể ra hai điều thay đổi lớn sau đây đối với vấn đề bảo vệ thông tin:

1-Sự ứng dụng của máy tính trong việc xử lý thông tin làm thay đổi dạng lưu trữ của thông tin và phương thức xử lý thông tin. Cần thiết phải xây dựng các cơ chế bảo vệ thông tin theo đặc thù hoạt động của máy tính. Từ đây xuất hiện yêu cầu bảo vệ sự an toàn hoạt động của máy tính (**Computer Security**) tồn tại song song với yêu cầu bảo vệ sự an toàn của thông tin (**Information Security**).

2-Sự phát triển mạnh mẽ của mạng máy tính và các hệ thống phân tán làm thay đổi phạm vi tổ chức xử lý thông tin. Thông tin được trao đổi giữa các thiết bị xử lý thông tin qua một khoảng cách vật lý rất lớn, gần như không giới hạn, làm xuất hiện thêm nhiều nguy cơ hơn đối với sự an toàn của thông tin. Từ đó xuất hiện yêu cầu bảo vệ sự an toàn của hệ thống mạng (**Network**

**Security**), gồm các cơ chế và kỹ thuật phù hợp với việc bảo vệ sự an toàn của thông tin khi chúng được trao đổi giữa các thiết bị trên mạng.

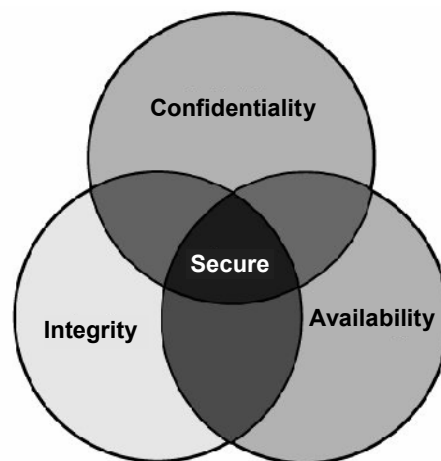
Cùng với việc nhận diện hai điều thay đổi lớn đối với vấn đề bảo đảm an toàn thông tin, hiện nay, khái niệm bảo đảm thông tin (**Information Assurance**) được đề xuất như một giải pháp toàn diện hơn cho bảo mật thông tin. Theo đó, vấn đề an toàn của thông tin không còn chỉ giới hạn trong việc đảm bảo tính bí mật và tính toàn vẹn của thông tin, phạm vi bảo vệ không còn giới hạn trong các hệ thống máy tính làm chức năng xử lý thông tin nữa, mà diễn ra trong tất cả các hệ thống tự động (automated systems). Yêu cầu bảo vệ không còn chỉ tập trung ở vấn đề an toàn động (**Security**) nữa mà bao gồm cả vấn đề an toàn tĩnh (**Safety**) và vấn đề tin cậy của hệ thống (**Reliability**).

Trong phạm vi tài liệu này, vấn đề Bảo mật hệ thống thông tin (Information System Security) là vấn đề trọng tâm nhất. Toàn bộ tài liệu sẽ tập trung vào việc mô tả, phân tích các cơ chế và kỹ thuật nhằm cung cấp sự bảo mật cho các hệ thống thông tin. Một hệ thống thông tin, theo cách hiểu ngầm định trong tài liệu này, là hệ thống xử lý thông tin bằng công cụ máy tính, được tổ chức tập trung hoặc phân tán. Do vậy, nội dung của tài liệu sẽ vừa đề cập đến vấn đề bảo mật máy tính (Computer Security) và bảo mật mạng (Network Security). Tuy vậy, các kỹ thuật bảo mật mạng chỉ được đề cập một cách giản lược, dành phần cho một tài liệu khác thuộc chuyên ngành Mạng máy tính và truyền thông, đó là tài liệu Bảo mật mạng.

## 1.2 CÁC ĐẶC TRƯNG CỦA MỘT HỆ THỐNG THÔNG TIN BẢO MẬT

Một hệ thống thông tin bảo mật (Secure Information System) là một hệ thống mà thông tin được xử lý trên nó phải đảm bảo được 3 đặc trưng sau đây:

- Tính bí mật của thông tin (**Confidentiality**)
- Tính toàn vẹn của thông tin (**Integrity**)
- Tính khả dụng của thông tin (**Availability**).



**Hình 1.1:** Mô hình CIA

Ba đặc trưng này được liên kết lại và xem như là mô hình tiêu chuẩn của các hệ thống thông tin bảo mật, hay nói cách khác, đây là 3 thành phần cốt yếu của một hệ thống thông tin Bảo mật. Mô hình này được sử dụng rộng rãi trong nhiều ngữ cảnh và nhiều tài liệu khác nhau, và

được gọi tắt là mô hình **CIA** (chú ý phân biệt với thuật ngữ CIA với ý nghĩa Confidentiality, Integrity, Authentication trong một số tài liệu khác).

Phần sau đây sẽ trình bày chi tiết về từng đặc trưng này.

### **1.2.1 Tính bí mật:**

Một số loại thông tin chỉ có giá trị đối với một đối tượng xác định khi chúng không phổ biến cho các đối tượng khác. *Tính bí mật của thông tin là tính giới hạn về đối tượng được quyền truy xuất đến thông tin.* Đối tượng truy xuất có thể là con người, là máy tính hoặc phần mềm, kể cả phần mềm phá hoại như virus, worm, spyware, ...

Tuỳ theo tính chất của thông tin mà mức độ bí mật của chúng có khác nhau. Ví dụ: các thông tin về chính trị và quân sự luôn được xem là các thông tin nhạy cảm nhất đối với các quốc gia và được xử lý ở mức bảo mật cao nhất. Các thông tin khác như thông tin về hoạt động và chiến lược kinh doanh của doanh nghiệp, thông tin cá nhân, đặc biệt của những người nổi tiếng, thông tin cấu hình hệ thống của các mạng cung cấp dịch vụ, v.v... đều có nhu cầu được giữ bí mật ở từng mức độ.

Để đảm bảo tính bí mật của thông tin, ngoài các cơ chế và phương tiện vật lý như nhà xưởng, thiết bị lưu trữ, dịch vụ bảo vệ, ... thì kỹ thuật mật mã hoá (Cryptography) được xem là công cụ bảo mật thông tin hữu hiệu nhất trong môi trường máy tính. Các kỹ thuật mật mã hoá sẽ được trình bày cụ thể ở chương II. Ngoài ra, kỹ thuật quản lý truy xuất (Access Control) cũng được thiết lập để bảo đảm chỉ có những đối tượng được cho phép mới có thể truy xuất thông tin. Access control sẽ được trình bày ở phần 3 của chương này.

Sự bí mật của thông tin phải được xem xét dưới dạng 2 yếu tố tách rời: sự *tồn tại của thông tin* và *nội dung của thông tin* đó.

Đôi khi, tiết lộ sự tồn tại của thông tin có ý nghĩa cao hơn tiết lộ nội dung của nó. Ví dụ: chiến lược kinh doanh bí mật mang tính sống còn của một công ty đã bị tiết lộ cho một công ty đối thủ khác. Việc nhận thức được rằng có điều đó tồn tại sẽ quan trọng hơn nhiều so với việc biết cụ thể về nội dung thông tin, chẳng hạn như ai đã tiết lộ, tiết lộ cho đối thủ nào và tiết lộ những thông tin gì,...

Cũng vì lý do này, trong một số hệ thống xác thực người dùng (user authentication) ví dụ như đăng nhập vào hệ điều hành Netware hay đăng nhập vào hộp thư điện tử hoặc các dịch vụ khác trên mạng, khi người sử dụng cung cấp một tên người dùng (user-name) sai, thay vì thông báo rằng user-name này không tồn tại, thì một số hệ thống sẽ thông báo rằng mật khẩu (password) sai, một số hệ thống khác chỉ thông báo chung chung là "Invalid user name/password" (người dùng hoặc mật khẩu không hợp lệ). Dụng ý đằng sau câu thông báo không rõ ràng này là việc từ chối xác nhận việc tồn tại hay không tồn tại một user-name như thế trong hệ thống. Điều này làm tăng sự khó khăn cho những người muốn đăng nhập vào hệ thống một cách bất hợp pháp bằng cách thử ngẫu nhiên.

### **1.2.2 Tính toàn vẹn:**

*Đặc trưng này đảm bảo sự tồn tại nguyên vẹn của thông tin, loại trừ mọi sự thay đổi thông tin có chủ đích hoặc hư hỏng, mất mát thông tin do sự cố thiết bị hoặc phần mềm.* Tính toàn vẹn được xét trên 2 khía cạnh:

- Tính nguyên vẹn của nội dung thông tin.
- Tính xác thực của nguồn gốc của thông tin.

Nói một cách khác, tính toàn vẹn của thông tin phải được đánh giá trên hai mặt: *toàn vẹn về nội dung* và *toàn vẹn về nguồn gốc*.

Ví dụ: một ngân hàng nhận được lệnh thanh toán của một người tự xưng là chủ tài khoản với đầy đủ những thông tin cần thiết. Nội dung thông tin được bảo toàn vì ngân hàng đã nhận được một cách chính xác yêu cầu của khách hàng (đúng như người xưng là chủ tài khoản gửi đi). Tuy nhiên, nếu lệnh thanh toán này không phải cho chính chủ tài khoản đưa ra mà do một người nào khác nhờ biết được thông tin bí mật về tài khoản đã mạo danh chủ tài khoản để đưa ra, ta nói nguồn gốc của thông tin đã không được bảo toàn.

Một ví dụ khác, một tờ báo đưa tin về một sự kiện vừa xảy ra tại một cơ quan quan trọng của chính phủ, có ghi chú rằng nguồn tin từ người phát ngôn của cơ quan đó. Tuy nhiên, nếu tin đó thật sự không phải do người phát ngôn công bố mà được lấy từ một kênh thông tin khác, không xét đến việc nội dung thông tin có đúng hay không, ta nói rằng nguồn gốc thông tin đã không được bảo toàn.

Sự toàn vẹn về nguồn gốc thông tin trong một số ngữ cảnh có ý nghĩa tương đương với sự đảm bảo tính không thể chối cãi (non-repudiation) của hệ thống thông tin.

Các cơ chế đảm bảo sự toàn vẹn của thông tin được chia thành 2 loại: các cơ chế ngăn chặn (***Prevention mechanisms***) và các cơ chế phát hiện (***Detection mechanisms***).

*Cơ chế ngăn chặn có chức năng ngăn cản các hành vi trái phép làm thay đổi nội dung và nguồn gốc của thông tin. Các hành vi này bao gồm 2 nhóm: hành vi cố gắng thay đổi thông tin khi không được phép truy xuất đến thông tin và hành vi thay đổi thông tin theo cách khác với cách đã được cho phép.*

Ví dụ: một người ngoài công ty cố gắng truy xuất đến cơ sở dữ liệu kế toán của một công ty và thay đổi dữ liệu trong đó. Đây là hành vi thuộc nhóm thứ nhất. Trường hợp một nhân viên kế toán được trao quyền quản lý cơ sở dữ liệu kế toán của công ty, và đã dùng quyền truy xuất của mình để thay đổi thông tin nhằm biển thủ ngân quỹ, đây là hành vi thuộc nhóm thứ hai.

*Nhóm các cơ chế phát hiện chỉ thực hiện chức năng giám sát và thông báo khi có các thay đổi diễn ra trên thông tin bằng cách phân tích các sự kiện diễn ra trên hệ thống mà không thực hiện chức năng ngăn chặn các hành vi truy xuất trái phép đến thông tin.*

Nếu như tính bí mật của thông tin chỉ quan tâm đến việc thông tin có bị tiết lộ hay không, thì tính toàn vẹn của thông tin vừa quan tâm tới tính chính xác của thông tin và cả mức độ tin cậy của thông tin. Các yếu tố như nguồn gốc thông tin, cách thức bảo vệ thông tin trong quá khứ cũng như trong hiện tại đều là những yếu tố quyết định độ tin cậy của thông tin và do đó ảnh hưởng đến tính toàn vẹn của thông tin. Nói chung, việc đánh giá tính toàn vẹn của một hệ thống thông tin là một công việc phức tạp.

### **1.2.3 Tính khả dụng:**

*Tính khả dụng của thông tin là tính sẵn sàng của thông tin cho các nhu cầu truy xuất hợp lệ.*

Ví dụ: các thông tin về quản lý nhân sự của một công ty được lưu trên máy tính, được bảo vệ một cách chắc chắn bằng nhiều cơ chế đảm bảo thông tin không bị tiết lộ hay thay đổi. Tuy nhiên, khi người quản lý cần những thông tin này thì lại không truy xuất được vì lỗi hệ thống. Khi đó, thông tin hoàn toàn không sử dụng được và ta nói tính khả dụng của thông tin không được đảm bảo.

Tính khả dụng là một yêu cầu rất quan trọng của hệ thống, bởi vì một hệ thống tồn tại nhưng không sẵn sàng cho sử dụng thì cũng giống như không tồn tại một hệ thống thông tin nào. Một hệ thống khả dụng là một hệ thống làm việc trôi chảy và hiệu quả, có khả năng phục hồi nhanh chóng nếu có sự cố xảy ra.

Trong thực tế, tính khả dụng được xem là nền tảng của một hệ thống bảo mật, bởi vì khi hệ thống không sẵn sàng thì việc đảm bảo 2 đặc trưng còn lại (bí mật và toàn vẹn) sẽ trở nên vô nghĩa.

*Hiện nay, các hình thức tấn công từ chối dịch vụ DoS (Denial of Service) và DDoS (Distributed Denial of Service) được đánh giá là các nguy cơ lớn nhất đối với sự an toàn của các hệ thống thông tin, gây ra những thiệt hại lớn và đặc biệt là chưa có giải pháp ngăn chặn hữu hiệu. Các hình thức tấn công này đều nhắm vào tính khả dụng của hệ thống.*

Một số hướng nghiên cứu đang đưa ra các mô hình mới cho việc mô tả các hệ thống an toàn. Theo đó, mô hình CIA không mô tả được đầy đủ các yêu cầu an toàn của hệ thống mà cần phải định nghĩa lại một mô hình khác với các đặc tính của thông tin cần được đảm bảo như:

- Tính khả dụng (Availability)
- Tính tiện ích (Utility)
- Tính toàn vẹn (Integrity)
- Tính xác thực (Authenticity)
- Tính bảo mật (Confidentiality)
- Tính sở hữu (Possession)

## **I.3 CÁC NGUY CƠ VÀ RỦI RO ĐỐI VỚI HỆ THỐNG THÔNG TIN**

### **I.3.1 Nguy cơ:**

*Nguy cơ (threat) là những sự kiện có khả năng ảnh hưởng đến an toàn của hệ thống.*

Ví dụ: tấn công từ chối dịch vụ (DoS và DDoS) là một nguy cơ đối với hệ thống các máy chủ cung cấp dịch vụ trên mạng.

Khi nói đến nguy cơ, nghĩa là sự kiện đó chưa xảy ra, nhưng *có khả năng xảy ra và có khả năng gây hại cho hệ thống*. Có những sự kiện có khả năng gây hại, nhưng không có khả năng xảy ra đối với hệ thống thì không được xem là nguy cơ.

Ví dụ: tấn công của sâu Nimda (năm 2001) có khả năng gây tê liệt toàn bộ hệ thống mạng nội bộ. Tuy nhiên, sâu Nimda chỉ khai thác được lỗi bảo mật của phần mềm IIS (Internet Information Service) trên Windows (NT và 2000) và do đó chỉ có khả năng xảy ra trên mạng có máy cài đặt hệ điều hành Windows. Nếu một mạng máy tính chỉ gồm toàn các máy cài hệ điều hành Unix hoặc Linux thì sâu Nimda hoàn toàn không có khả năng tồn tại, và do vậy, sâu Nimda không phải là một nguy cơ trong trường hợp này.

Có thể chia các nguy cơ thành 4 nhóm sau đây:

- Tiết lộ thông tin / truy xuất thông tin trái phép
- Phát thông tin sai / chấp nhận thông tin sai
- Phá hoại / ngăn chặn hoạt động của hệ thống
- Chiếm quyền điều khiển từng phần hoặc toàn bộ hệ thống

Đây là cách phân chia rất khái quát. Mỗi nhóm sẽ bao gồm nhiều nguy cơ khác nhau.

- Nghe lén, hay đọc lén (gọi chung là *snooping*) là một trong những phương thức *truy xuất thông tin trái phép*. Các hành vi thuộc phương thức này có thể đơn giản như việc nghe lén một cuộc đàm thoại, mở một tập tin trên máy của người khác, hoặc phức tạp hơn như xen vào một kết nối mạng (*wire-tapping*) để ăn cắp dữ liệu, hoặc cài các chương trình ghi bàn phím (*key-logger*) để ghi lại những thông tin quan trọng được nhập từ bàn phím.
- Nhóm nguy cơ *phát thông tin sai / chấp nhận thông tin sai* bao gồm những hành vi tương tự như nhóm ở trên nhưng mang tính chủ động, tức là có thay đổi thông tin gốc. Nếu thông tin bị thay đổi là thông tin điều khiển hệ thống thì mức độ thiệt hại sẽ nghiêm trọng hơn nhiều bởi vì khi đó, hành vi này không chỉ gây ra sai dữ liệu mà còn có thể làm thay đổi các chính sách an toàn của hệ thống hoặc ngăn chặn hoạt động bình thường của hệ thống.

Trong thực tế, hình thức tấn công xen giữa *Man-in-the-middle* (MITM) là một dạng của phương thức phát thông tin sai / chấp nhận thông tin sai. Hoạt động của hình thức tấn công này là xen vào một kết nối mạng, đọc lén thông tin và thay đổi thông tin đó trước khi gửi đến cho nơi nhận.

Giả danh (*spoofing*) cũng là một dạng hành vi thuộc nhóm nguy cơ phát thông tin sai / chấp nhận thông tin sai. Hành vi này thực hiện việc trao đổi thông tin với một đối tác bằng cách giả danh một thực thể khác.

Phủ nhận hành vi (*repudiation*) cũng là một phương thức gây sai lệch thông tin. Bằng phương thức này, một thực thể thực hiện hành vi phát ra thông tin, nhưng sau đó lại chối bỏ hành vi này, tức không công nhận nguồn gốc của thông tin, và do đó vi phạm yêu cầu về tính toàn vẹn của thông tin.

Ví dụ: một người chủ tài khoản yêu cầu ngân hàng thanh toán từ tài khoản của mình. Mọi thông tin đều chính xác và ngân hàng đã thực hiện lệnh. Tuy nhiên sau đó người chủ tài khoản lại phủ nhận việc mình đã đưa ra lệnh thanh toán. Khi đó, thông tin đã bị sai lệch do nguồn gốc của thông tin không còn xác định.

- Nhóm nguy cơ thứ 3 bao gồm các hành vi có mục đích *ngăn chặn hoạt động bình thường của hệ thống* bằng cách làm chậm hoặc gián đoạn dịch vụ của hệ thống. Tấn công từ chối dịch vụ hoặc virus là những nguy cơ thuộc nhóm này.
- *Chiếm quyền điều khiển hệ thống* gây ra nhiều mức độ thiệt hại khác nhau, từ việc lấy cắp và thay đổi dữ liệu trên hệ thống, đến việc thay đổi các chính sách bảo mật và vô hiệu hoá các cơ chế bảo mật đã được thiết lập.

Ví dụ điển hình cho nhóm nguy cơ này là các phương thức tấn công nhằm chiếm quyền root trên các máy tính chạy Unix hoặc Linux bằng cách khai thác các lỗi phần mềm hoặc lỗi cấu hình hệ thống. Tấn công tràn bộ đệm (*buffer overflow*) là cách thường dùng nhất để chiếm quyền root trên các hệ thống Linux vốn được xây dựng trên nền tảng của ngôn ngữ lập trình C.

### 1.3.2 Rủi ro và quản lý rủi ro:

*Rủi ro (risk) là xác suất xảy ra thiệt hại đối với hệ thống.*

Rủi ro bao gồm 2 yếu tố: Khả năng xảy ra rủi ro và thiệt hại do rủi ro gây ra. Có những rủi ro có khả năng xảy ra rất cao nhưng mức độ thiệt hại thì thấp và ngược lại.



Ví dụ: rủi ro mất thông tin trên hệ thống không có cơ chế bảo vệ tập tin, chẳng hạn như Windows 98. Windows 98 không có cơ chế xác thực người sử dụng nên bất cứ ai cũng có thể sử dụng máy với quyền cao nhất. Nếu trên đó chỉ có chứa các tập tin văn bản không có tính bí mật thì việc mất một tập tin thì thiệt hại gây ra chỉ là mất công sức đánh máy văn bản đó. Đây là dạng rủi ro có *xác suất xảy ra cao nhưng thiệt hại thấp*.

Một ví dụ khác: trên máy chủ cung cấp dịch vụ có một phần mềm có lỗi tràn bộ đệm, và nếu khai thác được lỗi này thì kẻ tấn công có thể chiếm được quyền điều khiển toàn bộ hệ thống. Tuy nhiên, đây là phần mềm không phổ biến và để khai thác được lỗi này, kẻ tấn công phải có những kỹ năng cao cấp. Rủi ro hệ thống bị chiếm quyền điều khiển được đánh giá là có *khả năng xảy ra thấp, nhưng nếu có xảy ra, thì thiệt hại sẽ rất cao*.

Cần chú ý phân biệt giữa nguy cơ và rủi ro. **Nguy cơ** là những hành vi, những sự kiện hoặc đối tượng có khả năng gây hại cho hệ thống. **Rủi ro** là những thiệt hại có khả năng xảy ra đối với hệ thống.

Ví dụ: Tấn công từ chối dịch vụ là một nguy cơ (threat). Đây là một sự kiện có khả năng xảy ra đối với bất kỳ hệ thống cung cấp dịch vụ nào. Thiệt hại do tấn công này gây ra là hệ thống bị gián đoạn hoạt động, đây mới là rủi ro (risk). Tuy nhiên, không phải bất kỳ tấn công từ chối dịch vụ nào xảy ra cũng đều làm cho hệ thống ngưng hoạt động, và hơn nữa, tấn công từ chối dịch vụ không phải là nguồn gốc duy nhất gây ra gián đoạn hệ thống; những nguy cơ khác như lỗi hệ thống (do vận hành sai), lỗi phần mềm (do lập trình), lỗi phần cứng (hư hỏng thiết bị, mất điện, ...) cũng đều có khả năng dẫn đến gián đoạn hệ thống.

Một ví dụ khác, xét trường hợp lưu trữ tập tin trên một máy tính chạy hệ điều hành Windows 98 đã nói ở trên. Nguy cơ đối với hệ thống là *các hành vi sửa hoặc xóa tập tin trên máy người khác*. Những người hay sử dụng máy tính của người khác cũng được xem là nguy cơ đối với hệ thống. Rủi ro đối với hệ thống trong trường hợp này là việc *tập tin bị mất hoặc bị sửa*.

*Trong thực tế, việc đề ra chính sách bảo mật cho một hệ thống thông tin phải đảm bảo được sự cân bằng giữa lợi ích của việc bảo đảm an toàn hệ thống và chi phí thiết kế, cài đặt và vận hành các cơ chế bảo vệ chính sách đó.*

Công việc quản lý rủi ro trên một hệ thống là quy trình cần thiết để nhận diện tất cả những rủi ro đối với hệ thống, những nguy cơ có thể dẫn đến rủi ro và phân tích lợi ích / chi phí của giải pháp ngăn chặn rủi ro. Quy trình phân tích rủi ro bao gồm các bước:

- Nhận dạng các rủi ro đối với hệ thống
- Chọn lựa và thực hiện các giải pháp để giảm bớt rủi ro.
- Theo dõi và đánh giá thiệt hại của những rủi ro đã xảy ra, làm cơ sở cho việc điều chỉnh lại hai bước đầu.

### **I.3.3 Vấn đề con người trong bảo mật hệ thống:**

Con người luôn là trung tâm của tất cả các hệ thống bảo mật, bởi vì tất cả các cơ chế, các kỹ thuật được áp dụng để bảo đảm an toàn hệ thống đều có thể dễ dàng bị vô hiệu hoá bởi con người trong chính hệ thống đó.

Ví dụ: hệ thống xác thực người sử dụng yêu cầu mỗi người trong hệ thống khi muốn thao tác trên hệ thống đều phải cung cấp tên người dùng và mật khẩu. Tuy nhiên, nếu người được cấp mật khẩu không bảo quản kỹ thông tin này, hoặc thậm chí đem tiết lộ cho người khác biết, thì khả năng xảy ra các vi phạm đối với chính sách an toàn là rất cao vì hệ thống xác thực đã bị vô hiệu hoá.

Những người có chủ ý muốn phá vỡ chính sách bảo mật của hệ thống được gọi chung là những người xâm nhập (*intruder* hoặc *attacker*) và theo cách nghĩ thông thường thì đây phải là những người bên ngoài hệ thống.

Tuy nhiên, thực tế đã chứng minh được rằng chính những người bên trong hệ thống, những người có điều kiện tiếp cận với hệ thống lại là những người có khả năng tấn công hệ thống cao nhất. Đó có thể là một nhân viên đang bất mãn và muốn phá hoại, hoặc chỉ là một người thích khám phá và chứng tỏ mình. Các tấn công gây ra bởi các đối tượng này thường khó phát hiện và gây thiệt hại nhiều hơn các tấn công từ bên ngoài.

Những người không được huấn luyện về an toàn hệ thống cũng là nơi tiềm ẩn các nguy cơ do những hành vi vô ý của họ như thao tác sai, bỏ qua các khâu kiểm tra an toàn, không tuân thủ chính sách bảo mật thông tin như lưu tập tin bên ngoài thư mục an toàn, ghi mật khẩu lên bàn làm việc, ...

## I.4 NGUYÊN TẮC XÂY DỰNG MỘT HỆ THỐNG BẢO MẬT

### I.4.1 Chính sách và cơ chế:

Hai khái niệm quan trọng thường được đề cập khi xây dựng một hệ thống bảo mật:

-Chính sách bảo mật (*Security policy*)

-Cơ chế bảo mật (*Security mechanism*)

*Chính sách bảo mật là hệ thống các quy định nhằm đảm bảo sự an toàn của hệ thống.*

*Cơ chế bảo mật là hệ thống các phương pháp, công cụ, thủ tục, ...dùng để thực thi các quy định của chính sách bảo mật.*

Chính sách bảo mật có thể được biểu diễn bằng ngôn ngữ tự nhiên hoặc ngôn ngữ toán học.

Ví dụ: trong một hệ thống, để bảo đảm an toàn cho một tài nguyên (resource) cụ thể, chính sách an toàn quy định rằng *chỉ có người dùng nào thuộc nhóm quản trị hệ thống (Administrators) mới có quyền truy xuất, còn những người dùng khác thì không*. Đây là cách biểu diễn bằng ngôn ngữ tự nhiên.

Có thể biểu diễn quy định này bằng ngôn ngữ toán học như sau:

Gọi:  $U$  là tập hợp các người dùng trong hệ thống.

$A$  là tập hợp các người dùng thuộc nhóm quản trị.

$O$  là tập hợp các đối tượng (tài nguyên) trong hệ thống

Thao tác  $\text{Access}(u, o)$  cho giá trị TRUE nếu người dùng  $u$  có quyền truy xuất đến đối tượng  $o$ , ngược lại, cho giá trị FALSE.

Quy định  $p$  trong chính sách an toàn được phát biểu như sau:

$$\forall u \in U, \forall o \in O: \text{Access}(u, o) = \text{TRUE} \Leftrightarrow u \in A$$

Ma trận cũng thường được dùng để biểu diễn một chính sách bảo mật.

Ví dụ: một hệ thống với các tập người dùng  $U = \{u_1, u_2, u_3, u_4\}$  và tập đối tượng  $O = \{o_1, o_2, o_3, o_4\}$ . Các thao tác mà một người dùng  $u$  có thể thực hiện được trên một đối tượng  $o$  bao gồm đọc (r), ghi (w) và thực thi (x). Quy định về khả năng truy xuất của từng người dùng đến từng đối tượng trong hệ thống được biểu diễn bằng ma trận như sau:

	$u_1$	$u_2$	$U_3$	$u_4$
$o_1$	x	x	R	
$o_2$	x	r	R	
$o_3$	w		R	
$o_4$	w		R	

Quan sát ma trận, ta biết rằng người dùng  $u_3$  được quyền đọc trên tất cả các đối tượng từ  $o_1$  đến  $o_4$ , trong khi đó người dùng  $u_4$  thì không có quyền truy xuất đến bất kỳ đối tượng nào.

*Cơ chế bảo mật thông thường là các biện pháp kỹ thuật.*

Ví dụ: xây dựng bức tường lửa (firewall), xác thực người dùng, dùng cơ chế bảo vệ tập tin của hệ thống quản lý tập tin NTFS để phân quyền truy xuất đối với từng tập tin / thư mục trên đĩa cứng, dùng kỹ thuật mật mã hoá để che giấu thông tin, v.v...

Tuy nhiên, đôi khi cơ chế chỉ là những *thủ tục (procedure)* mà khi thực hiện nó thì chính sách được bảo toàn.

Ví dụ: phòng thực hành máy tính của trường đại học quy định: sinh viên không được sao chép bài tập của sinh viên khác đã được lưu trên máy chủ. Đây là một quy định của chính sách bảo mật. Để thực hiện quy định này, các cơ chế được áp dụng bao gồm: tạo thư mục riêng trên máy chủ cho từng sinh viên, phân quyền truy xuất cho từng sinh viên đến các thư mục này và yêu cầu sinh viên phải lưu bài tập trong thư mục riêng, mỗi khi rời khỏi máy tính phải thực hiện thao tác logout khỏi hệ thống.

Trong cơ chế này, các biện pháp như tạo thư mục riêng, gán quyền truy xuất, ... là các biện pháp kỹ thuật. Biện pháp yêu cầu sinh viên thoát khỏi hệ thống (logout) khi rời khỏi máy là một biện pháp thủ tục. Nếu sinh viên ra về mà không thoát ra khỏi hệ thống, một sinh viên khác có thể sử dụng phiên làm việc đang mở của sinh viên này để sao chép bài tập. Khi đó, rõ ràng chính sách bảo mật đã bị vi phạm.

Cho trước một chính sách bảo mật, cơ chế bảo mật phải đảm bảo thực hiện được 3 yêu cầu sau đây:

- Ngăn chặn các nguy cơ gây ra vi phạm chính sách
- Phát hiện các hành vi vi phạm chính sách
- Khắc phục hậu quả của rủi ro khi có vi phạm xảy ra.

Thông thường, việc xây dựng một hệ thống bảo mật phải dựa trên 2 giả thiết sau đây:

*1-Chính sách bảo mật phân chia một cách rõ ràng các trạng thái của hệ thống thành 2 nhóm: an toàn và không an toàn.*

*2-Cơ chế bảo mật có khả năng ngăn chặn hệ thống tiến vào các trạng thái không an toàn.*

Chỉ cần một trong hai giả thiết này không đảm bảo thì hệ thống sẽ không an toàn. Từng cơ chế riêng lẻ được thiết kế để bảo vệ một hoặc một số các quy định trong chính sách. Tập hợp tất cả các cơ chế triển khai trên hệ thống phải đảm bảo thực thi tất cả các quy định trong chính sách.

Hai nguy cơ có thể xảy ra khi thiết kế hệ thống bảo mật do không đảm bảo 2 giả thiết ở trên:

*1-Chính sách không liệt kê được tất cả các trạng thái không an toàn của hệ thống, hay nói cách khác, chính sách không mô tả được một hệ thống bảo mật thật sự.*

2-Cơ chế không thực hiện được tất cả các quy định trong chính sách, có thể do giới hạn về kỹ thuật, ràng buộc về chi phí, ...

Dựa trên những nhận thức này, có thể đánh giá mức độ an toàn của một cơ chế như sau:

Gọi P là tập hợp tất cả các trạng thái của hệ thống, Q là tập hợp các trạng thái an toàn theo định nghĩa của chính sách bảo mật, giả sử cơ chế đang áp dụng có khả năng giới hạn các trạng thái của hệ thống trong tập R. Ta có các định nghĩa như sau:

-Nếu  $R \subseteq Q$ : cơ chế được đánh giá là *an toàn (secure mechanism)*.

-Nếu  $R = Q$ : cơ chế được đánh giá là *chính xác (precise mechanism)*.

-Nếu tồn tại trạng thái  $r \in R$  sao cho  $r \notin Q$ : cơ chế được đánh giá là *lỏng lẻo (broad mechanism)*.

#### 1.4.2 Các mục tiêu của bảo mật hệ thống:

Một hệ thống bảo mật, như trình bày ở phần 2 của chương này, là hệ thống thoả mãn 3 yêu cầu cơ bản là tính bí mật, tính toàn vẹn và tính khả dụng, gọi tắt là CIA.

Để thực hiện mô hình CIA, người quản trị hệ thống cần định nghĩa các trạng thái an toàn của hệ thống thông qua chính sách bảo mật, sau đó thiết lập các cơ chế bảo mật để bảo vệ chính sách đó.

Một hệ thống lý tưởng là hệ thống:

- Có chính sách xác định một cách chính xác và đầy đủ các trạng thái an toàn của hệ thống;
- Có cơ chế thực thi đầy đủ và hiệu quả các quy định trong chính sách.

Tuy nhiên trong thực tế, rất khó xây dựng những hệ thống như vậy do có những hạn chế về kỹ thuật, về con người hoặc do chi phí thiết lập cơ chế cao hơn lợi ích mà hệ thống an toàn đem lại. Do vậy, khi xây dựng một hệ thống bảo mật, thì **mục tiêu** đặt ra cho cơ chế được áp dụng phải bao gồm 3 phần như sau:

**Ngăn chặn (prevention)**: mục tiêu thiết kế là ngăn chặn các vi phạm đối với chính sách. Có nhiều sự kiện, hành vi dẫn đến vi phạm chính sách. Có những sự kiện đã được nhận diện là nguy cơ của hệ thống nhưng có những sự kiện chưa được ghi nhận là nguy cơ. Hành vi vi phạm có thể đơn giản như việc để lộ mật khẩu, quên thoát khỏi hệ thống khi rời khỏi máy tính, ... hoặc có những hành vi phức tạp và có chủ đích như cố gắng tấn công vào hệ thống từ bên ngoài.

Các cơ chế *an toàn (secure mechanism)* hoặc *cơ chế chính xác (precise mechanism)* theo định nghĩa ở trên là các cơ chế được thiết kế với mục tiêu *ngăn chặn*.

Tuy nhiên, khi việc xây dựng các cơ chế an toàn hoặc chính xác là không khả thi thì cần phải quan tâm đến 2 mục tiêu sau đây khi thiết lập các cơ chế bảo mật:

**Phát hiện (detection)**: mục tiêu thiết kế tập trung vào các sự kiện vi phạm chính sách đã và đang xảy ra trên hệ thống.

Thực hiện các cơ chế phát hiện nói chung rất phức tạp, phải dựa trên nhiều kỹ thuật và nhiều nguồn thông tin khác nhau. Về cơ bản, các cơ chế phát hiện xâm nhập chủ yếu dựa vào việc theo dõi và phân tích các thông tin trong nhật ký hệ thống (*system log*) và dữ liệu đang lưu thông trên mạng (*network traffic*) để tìm ra các dấu hiệu của vi phạm. Các dấu hiệu vi phạm này (gọi là *signature*) thường phải được nhận diện trước và mô tả trong một cơ sở dữ liệu của hệ thống (gọi là *signature database*).

Ví dụ: khi máy tính bị nhiễm virus. Đa số các trường hợp người sử dụng phát hiện ra virus khi nó đã thực hiện phá hoại trên máy tính. Tuy nhiên có nhiều virus vẫn đang ở dạng tiềm ẩn chứ

chưa thi hành, khi đó dùng chương trình quét virus sẽ có thể phát hiện ra. Để chương trình quét virus làm việc có hiệu quả thì cần thiết phải cập nhật thường xuyên danh sách virus. *Quá trình cập nhật là quá trình đưa thêm các mô tả về dấu hiệu nhận biết các loại virus mới vào cơ sở dữ liệu (virus database hoặc virus list).*

**Phục hồi (recovery):** mục tiêu thiết kế bao gồm các cơ chế nhằm chặn đứng các vi phạm đang diễn ra (*response*) hoặc khắc phục hậu quả của vi phạm một cách nhanh chóng nhất với mức độ thiệt hại thấp nhất (*recovery*).

Tùy theo mức độ nghiêm trọng của sự cố mà có các cơ chế phục hồi khác nhau. Có những sự cố đơn giản và việc phục hồi có thể hoàn toàn được thực hiện tự động mà không cần sự can thiệp của con người, ngược lại có những sự cố phức tạp và nghiêm trọng yêu cầu phải áp dụng những biện pháp bổ sung để phục hồi.

Một phần quan trọng trong các cơ chế phục hồi là việc nhận diện sơ hở của hệ thống và điều chỉnh những sơ hở đó. Nguồn gốc của sơ hở có thể do chính sách an toàn chưa chặt chẽ hoặc do lỗi kỹ thuật của cơ chế.

## **1.5 CHIẾN LƯỢC BẢO MẬT HỆ THỐNG AAA**

**AAA (Access control, Authentication, Auditing)** được xem là bước tiếp cận cơ bản nhất và là chiến lược nền tảng nhất để thực thi các chính sách bảo mật trên một hệ thống được mô tả theo mô hình CIA.

Cơ sở của chiến lược này như sau:

- 1-Quyền truy xuất đến tất cả các tài nguyên trong hệ thống được xác định một cách tường minh và gán cho các đối tượng xác định trong hệ thống.*
- 2-Mỗi khi một đối tượng muốn vào hệ thống để truy xuất các tài nguyên, nó phải được xác thực bởi hệ thống để chắc chắn rằng đây là một đối tượng có quyền truy xuất.*
- 3-Sau khi đã được xác thực, tất cả các thao tác của đối tượng đều phải được theo dõi để đảm bảo đối tượng không thực hiện quá quyền hạn của mình.*

Cần phân biệt với AAA trong ngữ cảnh quản lý *mạng truy nhập* với ý nghĩa Authentication, Authorization, Accounting – là dịch vụ trên các máy chủ truy nhập từ xa (*remote access server*) để thực hiện quản lý truy nhập mạng của người sử dụng, theo dõi lưu lượng sử dụng và tính cước truy nhập. AAA trong trường hợp này thường triển khai cùng với các dịch vụ như RADIUS, TACACS+, ...

AAA gồm 3 lĩnh vực tách rời nhưng hoạt động song song với nhau nhằm tạo ra các cơ chế bảo vệ sự an toàn của hệ thống. Phần sau đây trình bày chi tiết về 3 lĩnh vực của AAA.

### **1.5.1 Điều khiển truy xuất:**

**Điều khiển truy xuất (Access control)** được định nghĩa là *một quy trình được thực hiện bởi một thiết bị phần cứng hay một module phần mềm, có tác dụng chấp thuận hay từ chối một sự truy xuất cụ thể đến một tài nguyên cụ thể.*

Điều khiển truy xuất được thực hiện tại nhiều vị trí khác nhau của hệ thống, chẳng hạn như tại thiết bị truy nhập mạng (như remote access server-RAS hoặc wireless access point - WAP), tại hệ thống quản lý tập tin của một hệ điều hành ví dụ NTFS trên Windows hoặc trên các hệ thống Active Directory Service trong Netware 4.x hay Windows 2000 server,...

Trong thực tế, điều khiển truy xuất được thực hiện theo 3 mô hình sau đây:

**-Mô hình điều khiển truy xuất bắt buộc (Mandatory Access Control\_MAC):** là mô hình điều khiển truy xuất được áp dụng bắt buộc đối với toàn hệ thống. Trong môi trường máy tính, cơ chế điều khiển truy xuất bắt buộc được tích hợp sẵn trong hệ điều hành, và có tác dụng đối với tất cả các tài nguyên và đối tượng trong hệ thống, người sử dụng không thể thay đổi được.

Ví dụ: trong hệ thống an toàn nhiều cấp (*multilevel security*), mỗi đối tượng (*subject*) hoặc tài nguyên (*object*) được gán một mức bảo mật xác định. Trong hệ thống này, các đối tượng có mức bảo mật thấp không được đọc thông tin từ các tài nguyên có mức bảo mật cao, ngược lại các đối tượng ở mức bảo mật cao thì không được ghi thông tin vào các tài nguyên có mức bảo mật thấp. Mô hình này đặc biệt hữu dụng trong các hệ thống bảo vệ bí mật quân sự (*mô hình Bell-LaPadula, 1973*).

Những đặc điểm phân biệt của mô hình điều khiển truy xuất bắt buộc:

-Được thiết lập cố định ở mức hệ thống, người sử dụng (bao gồm cả người tạo ra tài nguyên) không thay đổi được.

-Người dùng và tài nguyên trong hệ thống được chia thành nhiều mức bảo mật khác nhau, phản ánh mức độ quan trọng của tài nguyên và người dùng.

-Khi mô hình điều khiển bắt buộc đã được thiết lập, nó có tác dụng đối với tất cả người dùng và tài nguyên trên hệ thống.

**-Mô hình điều khiển truy xuất tự do (Discretionary Access Control\_DAC):** là mô hình điều khiển truy xuất trong đó việc xác lập quyền truy xuất đối với từng tài nguyên cụ thể *do người chủ sở hữu của tài nguyên đó quyết định*. Đây là mô hình được sử dụng phổ biến nhất, xuất hiện trong hầu hết các hệ điều hành máy tính.

Ví dụ: trong hệ thống quản lý tập tin NTFS trên Windows XP, chủ sở hữu của một thư mục có toàn quyền truy xuất đối với thư mục, có quyền cho phép hoặc không cho phép người dùng khác truy xuất đến thư mục, có thể cho phép người dùng khác thay đổi các xác lập về quyền truy xuất đối với thư mục.

*Xem và thay đổi quyền truy xuất DAC trên một thư mục trong Windows XP:*

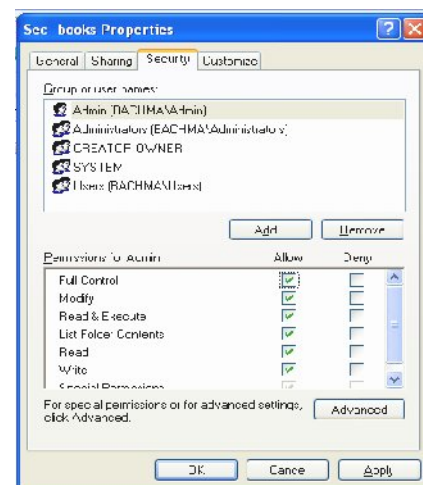
-Khởi động **Windows Explorer** bằng cách click phải vào biểu tượng **My Computer** và chọn **Explorer**.

-Mặc định, Windows XP không thể hiện các thông tin chi tiết về quyền truy xuất đối với thư mục. Muốn thể hiện các thông tin này, vào menu **Tools**, chọn **Folder Options**, click vào tab **View**, trong cửa sổ **Advanced settings**, tìm dòng **Use simple file sharing (Recommended)** ở cuối danh sách và bỏ tùy chọn này (uncheck), chọn **OK**.

-Click phải vào một thư mục tùy ý trong cửa sổ **Windows Explorer**, chọn **Properties**, click vào tab **Security** (Hình 1.2).

-Cửa sổ **Group or User names** liệt kê các người dùng và nhóm người dùng hiện có trong hệ thống. Cửa sổ **Permissions for ...** liệt kê các quyền đã được gán cho nhóm hoặc người dùng tương ứng.

-Thử cho phép hoặc xóa bỏ các quyền mặc định của một người dùng bất kỳ.



**Hình 1.2:** Điều khiển truy xuất tự do trong Windows XP

Đặc điểm phân biệt của mô hình điều khiển truy xuất tự do:

-Không được áp dụng mặc định trên hệ thống

-Người chủ sở hữu của tài nguyên (owner), thường là người tạo ra tài nguyên đó hoặc người được gán quyền sở hữu, có toàn quyền điều khiển việc truy xuất đến tài nguyên.

-Quyền điều khiển truy xuất trên một tài nguyên có thể được chuyển từ đối tượng (user) này sang đối tượng (user) khác.

***-Mô hình điều khiển truy xuất theo chức năng (Role Based Access Control\_RBAC):***  
đây là mô hình điều khiển truy xuất dựa trên vai trò của từng người dùng trong hệ thống (user' roles).

Ví dụ: một người quản lý tài chính cho công ty (financial manager) thì có quyền truy xuất đến tất cả các dữ liệu liên quan đến tài chính của công ty, được thực hiện các thao tác sửa, xóa, cập nhật trên cơ sở dữ liệu. Trong khi đó, một nhân viên kế toán bình thường thì chỉ được truy xuất đến một bộ phận nào đó của cơ sở dữ liệu tài chính và chỉ được thực hiện các thao tác có giới hạn đối với cơ sở dữ liệu.

Vấn đề quan trọng trong mô hình điều khiển truy xuất theo chức năng là định nghĩa các quyền truy xuất cho từng nhóm đối tượng tùy theo chức năng của các đối tượng đó. Việc này được định nghĩa ở mức hệ thống và áp dụng chung cho tất cả các đối tượng.

Cơ chế quản lý theo nhóm (account group) của Windows NT chính là sự mô phỏng của mô hình RBAC. Trong cơ chế này, người sử dụng được gán làm thành viên của một hoặc nhiều nhóm trong hệ thống, việc phân quyền truy xuất đến các tài nguyên được thực hiện đối với các nhóm chứ không phải đối với từng người dùng, khi đó các người dùng thành viên trong nhóm sẽ nhận được quyền truy xuất tương đương một cách mặc định. Việc thay đổi quyền truy xuất đối với từng người dùng riêng biệt được thực hiện bằng cách chuyển người dùng đó sang nhóm khác có quyền truy xuất thích hợp.

Đặc điểm phân biệt của mô hình điều khiển truy xuất theo chức năng:

-Quyền truy xuất được cấp dựa trên công việc của người dùng trong hệ thống (user's role)

-Linh động hơn mô hình điều khiển truy xuất bắt buộc, người quản trị hệ thống có thể cấu hình lại quyền truy xuất cho từng nhóm chức năng hoặc thay đổi thành viên trong các nhóm.

-Thực hiện đơn giản hơn mô hình điều khiển truy xuất tự do, không cần phải gán quyền truy xuất trực tiếp cho từng người dùng.

#### Ứng dụng các mô hình điều khiển truy xuất trong thực tế:

Trong thực tế, mô hình điều khiển truy xuất tự do (DAC) được ứng dụng rộng rãi nhất do tính đơn giản của nó đối với người dùng. Tuy nhiên, DAC không đảm bảo được các yêu cầu đặc biệt về an toàn hệ thống. Do vậy, một mô hình thích hợp nhất là phối hợp cả 3 mô hình: mô hình điều khiển truy xuất bắt buộc, mô hình điều khiển truy xuất tự do và mô hình điều khiển truy xuất theo chức năng.

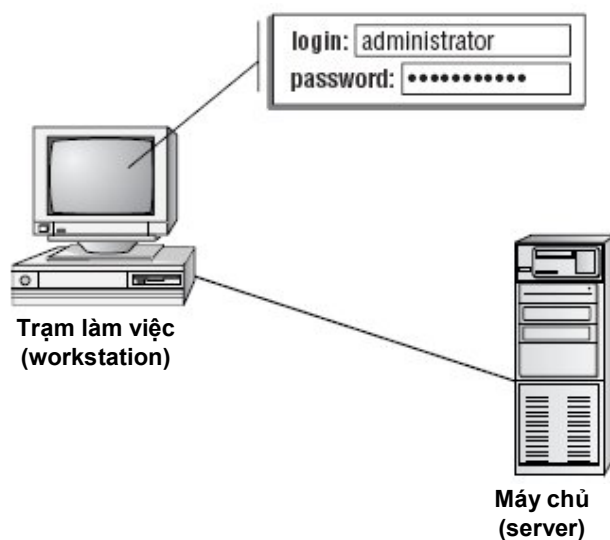
Ngoài mô hình DAC đã được tích hợp trong hầu hết các hệ điều hành; mô hình RBAC đã được ứng dụng trong dịch vụ Active Directory của Netware 4.11 và Windows 2000 trở về sau; mô hình MAC được đưa vào trong các hệ điều hành như Windows Vista (dưới dạng cơ chế Mandatory Integrity Control), SELinux (kể cả Red Hat Enterprise Linux version 4), Trusted Solaris và Apple Computer (MAC OS X version 10.5 Leopard).

### **1.5.2 Xác thực:**

**Xác thực (Authentication)** là một thủ tục có chức năng xác minh nhận dạng (identity) của một đối tượng trước khi trao quyền truy xuất cho đối tượng này đến một tài nguyên nào đó. Xác thực được thực hiện dựa trên 3 cơ sở:

- What you know (điều mà đối tượng biết), ví dụ mật khẩu.
- What you have (cái mà đối tượng có), ví dụ thẻ thông minh Smartcard.
- What you are (đặc trưng của đối tượng): các đặc điểm nhận dạng sinh trắc học như dấu vân tay, võng mạc, ...

Trong môi trường máy tính, xác thực được dùng ở nhiều ngữ cảnh khác nhau, ví dụ: xác thực tên đăng nhập và mật khẩu của người sử dụng (hình 1.3) trước khi cho phép người sử dụng thao tác trên hệ thống máy tính (xác thực của hệ điều hành), xác thực tên đăng nhập và mật khẩu trước khi cho phép người dùng kiểm tra hộp thư điện tử (xác thực của Mail server); trong giao dịch ngân hàng, thủ tục xác thực dùng để xác định người đang ra lệnh thanh toán có phải là chủ tài khoản hay không; trong trao đổi thông tin, thủ tục xác thực dùng để xác định chính xác nguồn gốc của thông tin.



**Hình 1.3:** Xác thực bằng tên đăng nhập và mật khẩu

Nhiều kỹ thuật khác nhau được áp dụng để thực thi cơ chế xác thực. Cơ chế *xác thực dùng tên đăng nhập và mật khẩu* là cơ chế truyền thống và vẫn còn được sử dụng rộng rãi hiện nay. Khi việc xác thực được thực hiện thông qua mạng, một số hệ thống thực hiện việc mã hóa tên đăng nhập và mật khẩu trước khi truyền đi để tránh bị tiết lộ, nhưng cũng có nhiều hệ thống gửi trực tiếp những thông tin nhạy cảm này trên mạng (ví dụ như các dịch vụ FTP, Telnet, ...) gọi là cleartext authentication.

Một số kỹ thuật tiên tiến hơn được dùng trong xác thực như *thẻ thông minh (Smartcard)*, *chứng thực số (digital certificate)*, *các thiết bị nhận dạng sinh trắc học (biometric devices)*,...

Để tăng độ tin cậy của cơ chế xác thực, *nhiều kỹ thuật được sử dụng phối hợp nhau* gọi là multi-factor authentication. Ví dụ: xác thực dùng thẻ thông minh kèm với mật khẩu, nghĩa là người sử dụng vừa có thẻ vừa phải biết mật khẩu thì mới đăng nhập được, tránh trường hợp lấy cắp thẻ của người khác để đăng nhập.



Trong thực tế tồn tại hai phương thức xác thực: *xác thực một chiều (one way authentication)* và *xác thực hai chiều (mutual authentication)*.

Phương thức xác thực một chiều chỉ cung cấp cơ chế để một đối tượng (thường là máy chủ) kiểm tra nhận dạng của đối tượng kia (người dùng) mà không cung cấp cơ chế kiểm tra ngược lại (tức không cho phép người dùng kiểm tra nhận dạng của máy chủ). Xét trường hợp một người sử dụng đăng nhập vào một hộp thư điện tử ở xa thông qua dịch vụ web (web mail). Người sử dụng dĩ nhiên phải cung cấp tên đăng nhập và mật khẩu đúng thì mới được phép truy xuất hộp thư. Để đánh cắp mật khẩu của người dùng, kẻ tấn công có thể xây dựng một trang web hoàn toàn giống với giao diện của máy chủ cung cấp dịch vụ thư điện tử (mail server) và đánh lừa người sử dụng kết nối đến trang web này. Do không có cơ chế xác thực máy chủ, người sử dụng không thể nhận biết đây là một máy chủ giả mạo nên yên tâm cung cấp tên đăng nhập và mật khẩu.

Phương thức kiểm tra hai chiều cho phép hai đối tượng tham gia giao tác xác thực lẫn nhau, do đó tính chính xác của quá trình xác thực được đảm bảo. Giao thức bảo mật SSL (Secure Sockets Layer) dùng trong dịch vụ web (được trình bày ở chương III) cung cấp cơ chế xác thực hai chiều dùng chứng thực số.

*Có nhiều giải thuật xác thực khác nhau.* Giải thuật đơn giản nhất chỉ cần so sánh tên đăng nhập và mật khẩu mà người sử dụng cung cấp với tên đăng nhập và mật khẩu đã được lưu trong hệ thống, nếu giống nhau nghĩa là thủ tục xác thực thành công (PAP). Giải thuật phức tạp hơn như CHAP thì thực hiện việc mật mã hóa thông tin trên một giá trị ngẫu nhiên nào đó do máy chủ đưa ra (gọi là challenge) để tránh trường hợp mật khẩu bị đọc lên trên mạng và các hình thức tấn công phát lại (replay attack). Một giải thuật phức tạp khác là Kerberos thực hiện thủ tục xác thực theo một quá trình phức tạp gồm nhiều bước nhằm đảm bảo hạn chế tất cả các nguy cơ gây nên xác thực sai. Các giải thuật xác thực được trình bày cho tiết ở phần I của chương III.

### 1.5.3 Kiểm tra:

**Kiểm tra (Auditing)** là cơ chế *theo dõi hoạt động* của hệ thống, *ghi nhận các hành vi* diễn ra trên hệ thống và *liên kết các hành vi này với các tác nhân gây ra hành vi*.

Ví dụ: cài đặt cơ chế kiểm tra cho một thư mục trong hệ thống tập tin NTFS sẽ cho phép người quản trị theo dõi các hoạt động diễn ra trên thư mục như: thao tác nào đã được thực hiện, ngày giờ thực hiện, người sử dụng nào thực hiện, ...

Các mục tiêu của kiểm tra:

- Cung cấp các thông tin cần thiết cho việc phục hồi hệ thống khi có sự cố
- Đánh giá mức độ an toàn của hệ thống để có kế hoạch nâng cấp kịp thời
- Cung cấp các thông tin làm chứng cứ cho việc phát hiện các hành vi truy xuất trái phép trên hệ thống.

Trong một hệ thống tin cậy (reliable system) thì việc kiểm tra cũng là một yêu cầu quan trọng bởi vì nó đảm bảo rằng các hành vi của bất kỳ người dùng nào trong hệ thống (kể cả những người dùng hợp hệ đã được xác thực – authenticated user) cũng đều được theo dõi để chắc chắn rằng những hành vi đó diễn ra đúng theo các chính sách an toàn đã được định nghĩa trên hệ thống.

Nguyên tắc chung khi xây dựng các hệ thống an toàn là chia nhỏ các thủ tục thành nhiều công đoạn được thực hiện bởi nhiều tác nhân khác nhau, và do đó việc thực hiện hoàn chỉnh một thủ tục yêu cầu phải có sự tham gia của nhiều tác nhân. Đây là cơ sở để thực thi các cơ chế kiểm tra.

Ví dụ: công việc giữ kho hàng và công việc quản lý sổ sách phải được thực hiện bởi hai nhân viên khác nhau để tránh trường hợp một nhân viên vừa có thể lấy hàng ra ngoài vừa có thể thay đổi thông tin trong sổ quản lý. Nguyên tắc này được áp dụng triệt để trong cơ chế kiểm tra trên hệ thống nhằm phân biệt rõ ràng giữa chức năng kiểm tra với các hoạt động được kiểm tra. Thông thường, một đối tượng được kiểm tra sẽ không có quyền thay đổi các thông tin mà cơ chế kiểm tra ghi lại.

Các thành phần của hệ thống kiểm tra:

**-Logger:** Ghi lại thông tin giám sát trên hệ thống

**-Analyzer:** Phân tích kết quả kiểm tra

**-Notifier:** Cảnh báo về tính an toàn của hệ thống dựa trên kết quả phân tích.

Song song với cơ chế *kiểm tra thường trực* trên hệ thống (auditing), việc *kiểm tra hệ thống định kỳ* (system scanning) có chức năng kiểm tra và phát hiện các sơ hở kỹ thuật ảnh hưởng đến sự an toàn của hệ thống. Các chức năng có thể thực hiện bởi các chương trình kiểm tra hệ thống trên máy tính thường gặp:

-Kiểm tra việc tuân thủ chính sách an toàn về mật khẩu (password policy), ví dụ: người dùng có đổi mật khẩu thường xuyên không, độ dài mật khẩu, độ phức tạp của mật khẩu, ...

-Đánh giá khả năng xâm nhập hệ thống từ bên ngoài.

-Kiểm tra phản ứng của hệ thống đối với các dấu hiệu có thể dẫn đến tấn công từ chối dịch vụ hoặc sự cố hệ thống (system crash).

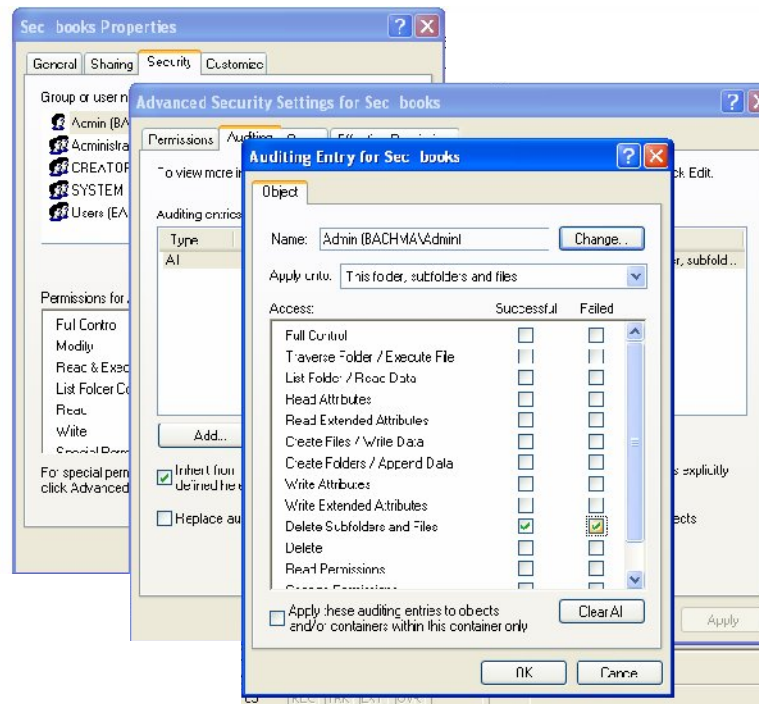
Lưu ý rằng, các công cụ kiểm tra hệ thống cũng đồng thời là các công cụ mà những *kẻ tấn công* (attacker) sử dụng để phát hiện các lỗ hổng bảo mật trên hệ thống, từ đó thực hiện các thao tác tấn công khác. Có nhiều phần mềm quét hệ thống, điển hình như SATAN (System Administrator Tool for Analyzing Network), Nessus, Nmap, ...

Cài đặt chức năng Audit của hệ điều hành Windows XP lên một thư mục trên một phân vùng NTFS:

-Mặc định, Windows XP không áp dụng cơ chế kiểm tra, do đó cần phải kích hoạt cơ chế kiểm tra của Windows XP dùng **Local Security Policy** như sau: Vào **Control Panel**, chọn **Administrative Tools**, chọn **Local Security Policy**, trong khung Security Settings ở bên trái cửa sổ, double-click vào mục **Local Policy**, sau đó click vào mục **Audit Policy**. Khi đó, khung bên phải cửa sổ liệt kê các chức năng kiểm tra của Windows XP. Để kích hoạt cơ chế kiểm tra trên thư mục, tìm dòng **Audit object access**, double-click vào dòng này và chọn cả hai mục **Success** và **Failure** trong cửa sổ mới mở. Click OK và đóng tất cả các cửa sổ lại.

-Để áp dụng cơ chế kiểm tra trên một thư mục nào đó: khởi động **Windows explorer**, tìm một thư mục muốn kiểm tra và click phải vào thư mục này, chọn **Properties**, click vào tab **Security**, click vào nút **Advanced**, sau đó click vào tab **Auditing**. Trong cửa sổ **Auditing entries** liệt kê các mục kiểm tra đã cài đặt. Để tạo một mục mới, click vào nút **Add**, chọn tên người dùng hoặc nhóm cần kiểm tra trong cửa sổ **Select User or Group** vừa xuất hiện, click **OK**. Cửa sổ **Auditing Entry for ...** xuất hiện, chọn các thao tác muốn kiểm tra, ví dụ **Delete Subfolders and Files** để theo dõi các hành vi xoá tập tin và thư mục con trong mục này. Cần chọn cả hai loại sự kiện là **Successful** và **Failed**. Click **OK** và đóng tất cả các cửa sổ lại.

-Bắt đầu từ đây, tất cả các thao tác xoá các tập tin và thư mục con trong thư mục đã chọn được thực hiện bởi người dùng hoặc nhóm đã chỉ định ở trên đều được theo dõi và ghi lại trong nhật ký hệ thống. Muốn xem các thông tin này thì vào **Control Panel**, chọn **Administrative Tools**, chọn **Event Viewer** và chọn mục **Security**.



**Hình 1.4:** Cài đặt Auditing trên thư mục NTFS

Tóm lại, AAA là phương pháp tiếp cận cơ bản nhất để thực hiện một hệ thống bảo mật theo mô hình CIA. Phương pháp này gồm 3 phần tách rời:

- Thiết lập các cơ chế điều khiển truy xuất cho từng đối tượng (Access control)
- Xác thực các đối tượng trước khi cho phép thao tác trên hệ thống (Authentication)
- Theo dõi các thao tác của đối tượng trên hệ thống (Auditing)

## I.6 CÁC HÌNH THỨC XÂM NHẬP HỆ THỐNG

Thuật ngữ *xâm nhập (intrusion)* và *tấn công (attack)* được sử dụng với ý nghĩa gần giống nhau trong ngữ cảnh bảo mật hệ thống. Xâm nhập mang ý nghĩa phổ quát hơn, chỉ bất kỳ một sự kiện nào có xâm hại đến sự an toàn của hệ thống, một cách chủ động hoặc thụ động. Tấn công thường được dùng để chỉ các hành vi xâm nhập chủ động, được thực hiện bởi con người nhằm vào một hệ thống với mục đích khai thác hoặc phá hoại.

*Mục tiêu của xâm nhập là tác động vào 3 thuộc tính CIA của hệ thống.*

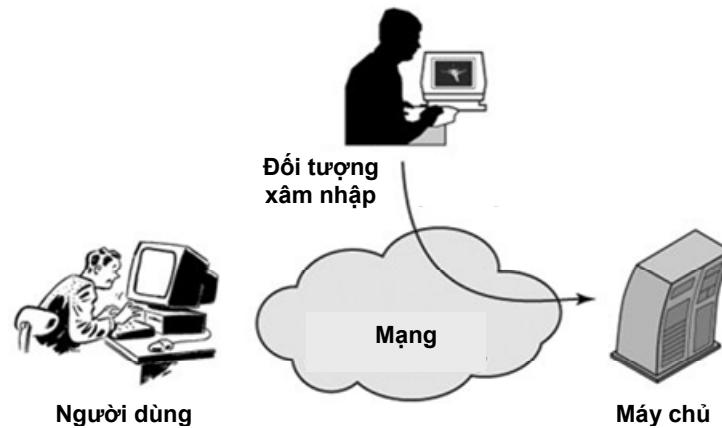
Một cách tổng quát, sự an toàn của một hệ thống thông tin có thể bị xâm phạm bằng những cách sau đây:

**-Interruption:** làm gián đoạn hoạt động của hệ thống thông tin, ví dụ như phá hoại phần cứng, ngắt kết nối, phá hoại phần mềm, ...Hình thức xâm nhập này tác động vào đặc tính Khả dụng của thông tin.

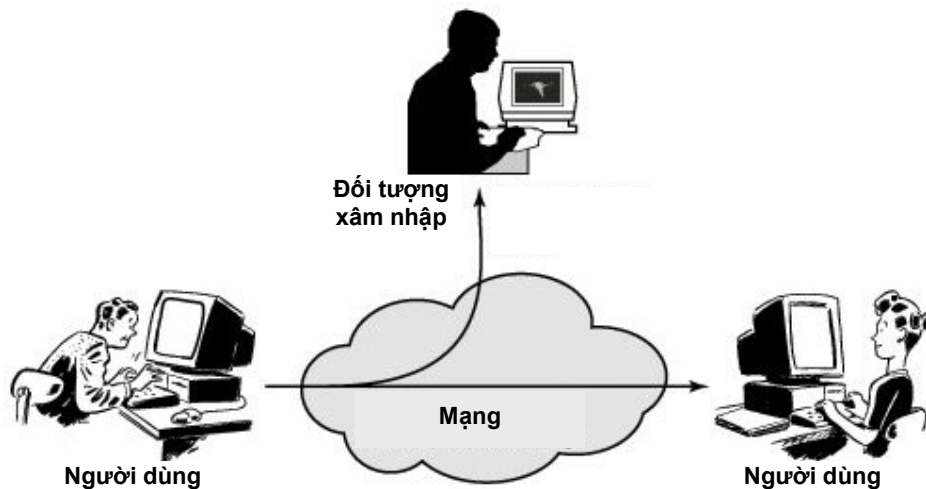
**-Interception:** truy xuất trái phép vào hệ thống thông tin. Tác nhân của các hành vi xâm nhập kiểu Interception có thể là một người, một phần mềm hay một máy tính làm việc bằng cách quan sát dòng thông tin (monitor) nhưng không làm thay đổi thông tin gốc. Hình thức xâm nhập này tác động vào đặc tính Bí mật của thông tin.

**-Modification:** truy xuất trái phép vào hệ thống thông tin, đồng thời làm thay đổi nội dung thông tin, ví dụ xâm nhập vào máy tính và làm thay đổi nội dung một tập tin, thay đổi một chương trình làm cho chương trình làm việc sai, thay đổi nội dung một thông báo đang gửi đi trên mạng, v.v... Hình thức xâm nhập này tác động vào tính Toàn vẹn của thông tin.

Ngoài ra, một hình thức xâm nhập thứ tư là hình thức xâm nhập bằng thông tin giả danh (**Farbrication**), ví dụ, giả danh một người nào đó để gửi mail đến một người khác, giả mạo địa chỉ IP của một máy nào đó để kết nối với một máy khác, ...Hình thức xâm nhập này làm thay đổi nguồn gốc thông tin, tức cũng là tác động vào đặc tính Toàn vẹn của thông tin.



Hình 1.5: Xâm nhập kiểu Interruption



Hình 1.6: Xâm nhập kiểu Interception

Trong thực tế, việc xâm nhập hệ thống được thực hiện bởi rất nhiều phương thức, công cụ và kỹ thuật khác nhau, thêm vào đó, việc phát hiện ra các phương thức xâm nhập mới là việc xảy ra rất thường xuyên, nên vấn đề nhận dạng và phân loại các xâm nhập một cách có hệ thống là khó khăn và không chính xác. Có thể phân loại xâm nhập theo các tiêu chí sau đây:

-Phân loại theo mục tiêu xâm nhập (xâm nhập mạng, xâm nhập ứng dụng, xâm nhập hỗn hợp)

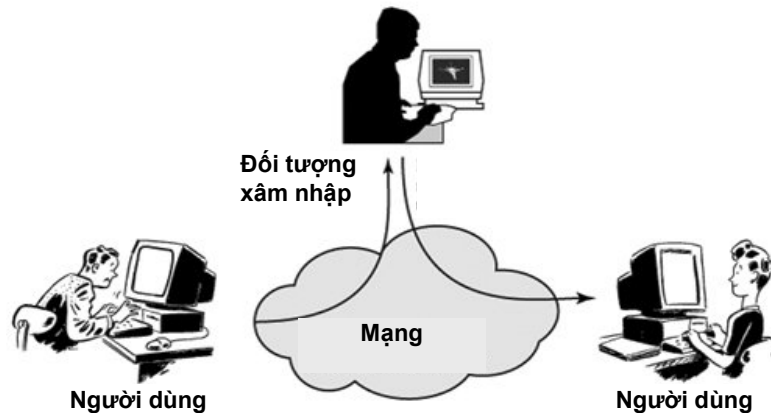
-Phân loại theo tính chất xâm nhập (xâm nhập chủ động, xâm nhập thụ động)

-Phân loại theo kỹ thuật xâm nhập (dò mật khẩu, phần mềm khai thác, ...)

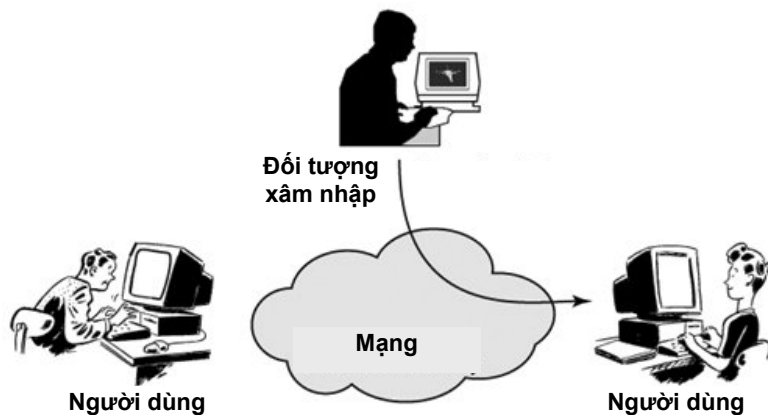
Trong tài liệu này, với mục tiêu là giúp người đọc nhận diện được những phương thức xâm nhập hệ thống cơ bản và phổ biến đã được ghi nhận và phân tích, nên các hình thức xâm nhập được trình bày theo hai nhóm như sau:

1-Các phương thức tấn công (attacks)

2-Các phương thức xâm nhập hệ thống bằng phần mềm phá hoại (malicious codes)



**Hình 1.7:** *Xâm nhập kiểu Modification*



**Hình 1.8:** *Xâm nhập kiểu Farbrication*

### **1.6.1 Các phương thức tấn công:**

**-Tấn công từ chối dịch vụ DoS (Denial of Service):**

Dạng tấn công này không xâm nhập vào hệ thống để lấy cắp hay thay đổi thông tin mà chỉ nhằm vào mục đích ngăn chặn hoạt động bình thường của hệ thống, đặc biệt đối với các hệ thống phục vụ trên mạng công cộng như Web server, Mail server, ...

Ví dụ: kẻ tấn công dùng phần mềm tự động liên tục gửi dữ liệu đến một máy chủ trên mạng, gây quá tải cho máy chủ, làm cho máy chủ không còn khả năng cung cấp dịch vụ một cách bình thường.

Các tấn công từ chối dịch vụ thường rất dễ nhận ra do tác động cụ thể của nó đối với hệ thống. Mục tiêu tấn công của từ chối dịch vụ có thể là một máy chủ hoặc một mạng con (bao gồm cả thiết bị mạng như router và kết nối mạng).

Cơ sở của tấn công từ chối dịch vụ là các sơ hở về bảo mật trong cấu hình hệ thống (cấu hình firewall), sơ hở trong giao thức kết nối mạng (TCP/IP) và các lỗ hổng bảo mật của phần mềm, hoặc đơn giản là sự hạn chế của tài nguyên như băng thông kết nối (connection bandwidth), năng lực của máy chủ (CPU, RAM, đĩa cứng, ...). Tấn công từ chối dịch vụ thường được thực hiện thông qua mạng Internet, nhưng cũng có thể xuất phát từ trong nội bộ hệ thống dưới dạng tác động của các phần mềm độc như worm hoặc trojan.

Hai kỹ thuật thường dùng để gây ra các tấn công từ chối dịch vụ truyền thống tương ứng với hai mục tiêu tấn công là *Ping of Death* và *buffer-overflow*.

- Ping of Death tấn công vào kết nối mạng (bao gồm cả router) bằng cách gửi liên tục và với số lượng lớn các gói dữ liệu ICMP (Internet Control Message Protocol) đến một mạng con nào đó, chiếm toàn bộ băng thông kết nối và do đó gây ra tắc nghẽn mạng.
- Buffer-overflow (được mô tả ở phần software exploitation attacks) tấn công vào các máy chủ bằng cách nạp dữ liệu vượt quá giới hạn của bộ đệm (buffer) trên máy chủ, gây ra lỗi hệ thống. Các tấn công từ chối dịch vụ nổi tiếng trong lịch sử bảo mật máy tính như Code Red, Slapper, Slammer,... là các tấn công sử dụng kỹ thuật buffer-overflow.

Tấn công từ chối dịch vụ thường không gây tiết lộ thông tin hay mất mát dữ liệu mà chỉ nhắm vào tính khả dụng của hệ thống. Tuy nhiên, do tính phổ biến của từ chối dịch vụ và đặc biệt là hiện nay chưa có một giải pháp hữu hiệu cho việc ngăn chặn các tấn công loại này nên từ chối dịch vụ được xem là một nguy cơ rất lớn đối với sự an toàn của các hệ thống thông tin.

#### **-Tấn công từ chối dịch vụ phân tán (Distributed DoS hay DDoS):**

Là phương thức tấn công dựa trên nguyên tắc của từ chối dịch vụ nhưng có mức độ nguy hiểm cao hơn do huy động cùng lúc nhiều máy tính cùng tấn công vào một hệ thống duy nhất.

Tấn công từ chối dịch vụ phân tán được thực hiện qua 2 giai đoạn:

1-Kẻ tấn công huy động nhiều máy tính trên mạng tham gia từ chối dịch vụ phân tán bằng cách cài đặt các phần mềm điều khiển từ xa trên các máy tính này.

Các máy tính đã được cài đặt phần mềm điều khiển này được gọi là các *zombie*. Để thực hiện bước này, kẻ tấn công dò tìm trên mạng những máy có nhiều sơ hở để tấn công và cài đặt các phần mềm điều khiển từ xa lên đó mà người quản lý không hay biết. Những phần mềm này được gọi chung là *backdoor*.

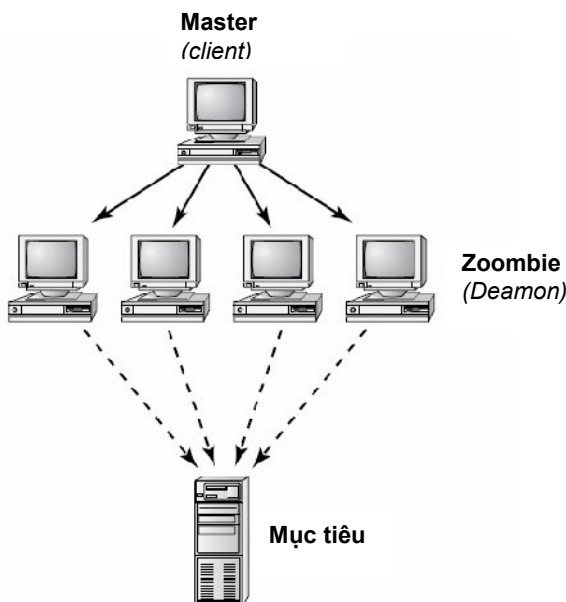
2-Kẻ tấn công điều khiển các zombie đồng loạt thực hiện tấn công vào mục tiêu.

Mô hình một chuỗi tấn công chối dịch vụ phân tán điển hình được mô tả ở hình 1.9.

Các thành phần tham gia trong chối dịch vụ phân tán bao gồm:

-*Client*: phần mềm điều khiển từ xa được kẻ tấn công sử dụng để điều khiển các máy khác tham gia tấn công. Máy tính chạy phần mềm này được gọi là master.

-*Daemon*: phần mềm chạy trên các zombie, thực hiện yêu cầu của master và là nơi trực tiếp thực hiện tấn công chối dịch vụ (DoS) đến máy nạn nhân.



**Hình 1.9:** Tấn công từ chối dịch vụ phân tán (DDoS)

#### **-Tấn công giả danh (Spoofing attack):**

Đây là dạng tấn công bằng cách giả danh một đối tượng khác (một người sử dụng, một máy tính với một địa chỉ IP xác định hoặc một phần mềm nào đó) để thực hiện một hành vi.

Ví dụ 1: một người có thể giả danh địa chỉ e-mail của một người khác để gửi thư đến một người thứ ba, đây là trường hợp đối tượng bị giả danh là một người sử dụng.

Ví dụ 2: một máy tính trên mạng có thể tạo ra các gói dữ liệu mang địa chỉ IP nguồn (source IP address) không phải là địa chỉ của mình để gửi cho máy khác (gọi là *IP spoofing*), đây là trường hợp đối tượng bị giả danh là một máy tính.

Ví dụ 3: trường hợp thứ ba là trường hợp mà đối tượng bị giả danh là một phần mềm, ví dụ chương trình xác thực người sử dụng (user login) trên hệ điều hành Windows. Bằng cách tạo ra một chương trình có giao diện giống như cửa sổ login của Windows và cho thực hiện khi Windows khởi động. Người sử dụng không phân biệt được đây là cửa sổ giả nên nhập tên đăng nhập và mật khẩu cho chương trình này và hậu quả là những thông tin này bị tiết lộ.

Tấn công giả danh như đề cập ở trên là hình thức điển hình nhất của spoofing attack, tồn tại song song với những khiếm khuyết về kỹ thuật của bộ giao thức TCP/IP. Ngày nay, Tấn công giả danh đã phát triển thêm một hướng mới dựa trên sự phổ biến của mạng Internet, đó là *Phishing*. Phishing hoạt động bằng cách giả danh các địa chỉ e-mail hoặc địa chỉ trang web để đánh lừa người sử dụng.

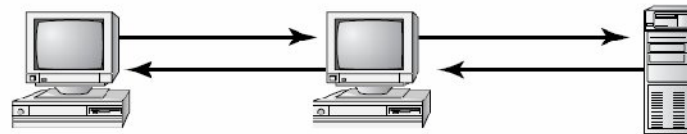
#### **-Tấn công xen giữa (Man-in-the-middle attack):**

Đây là phương thức tấn công bằng cách xen vào giữa một thủ tục đang diễn ra, thường xảy ra trên mạng IP, nhưng cũng có thể xảy ra trong nội bộ một máy tính.

Trên mạng, kẻ tấn công bằng một cách nào đó xen vào một kết nối, đặc biệt ở giai đoạn thiết lập kết nối giữa người dùng với máy chủ, và thông qua đó nhận được những thông tin quan

trọng của người dùng. Tấn công xen giữa đặc biệt phổ biến trên mạng không dây (wireless network) do đặc tính dễ xâm nhập của môi trường không dây. Do vậy, việc áp dụng các kỹ thuật mã hoá (như WEP, WPA, ...) là điều rất quan trọng để đảm bảo an toàn cho mạng không dây.

Còn trên một máy tính, tấn công dạng này có thể được thực hiện dưới dạng một chương trình thu thập thông tin ẩn (key-logger), chương trình này sẽ âm thầm chặn bắt tất cả những thông tin mà người dùng nhập vào từ bàn phím, trong đó có thể sẽ có nhiều thông tin quan trọng.

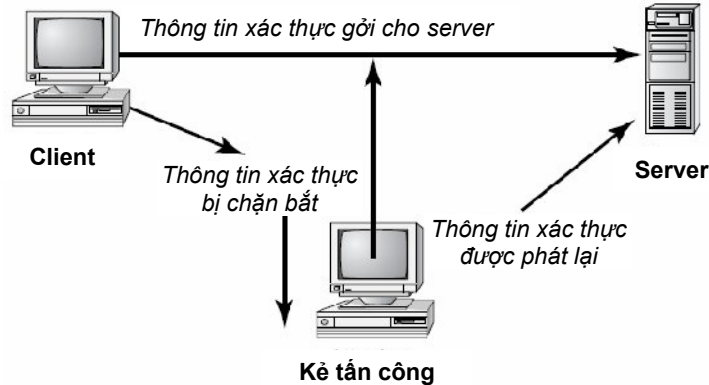


*Kẻ tấn công xen vào giữa một thủ tục bắt tay để lấy thông tin.*

**Hình 1.10:** Tấn công xen giữa (Man-in-the-middle)

#### **-Tấn công phát lại (Replay attack):**

Trong phương thức tấn công này, các gói dữ liệu lưu thông trên mạng được chặn bắt và sau đó phát lại (replay). Trong môi trường mạng, thông tin xác thực giữa người dùng và máy chủ được truyền đi trên mạng. Đây là nguồn thông tin thường bị tấn công nhất. Nếu khi phát lại, máy chủ chấp nhận thông tin này thì máy tấn công có khả năng truy xuất vào máy chủ với quyền của người dùng trước đó.



**Hình 1.11:** Tấn công phát lại (Replay)

#### **-Nghe lén (Sniffing attack):**

Đây là hình thức lấy cắp dữ liệu bằng cách đọc lén trên mạng. Hầu hết các card mạng điều có khả năng chặn bắt (capture) tất cả các gói dữ liệu lưu thông trên mạng, mặc dù gói dữ liệu đó không được gửi đến cho mình. Những card mạng có khả năng như thế được gọi là đang ở chế độ *promiscuous*.

Có rất nhiều phần mềm cho phép thực hiện chặn bắt dữ liệu từ một máy đang kết nối vào mạng, ví dụ *Ethereal*, *Common view* hoặc *Network monitor* có sẵn trên Windows server (2000



hoặc 2003 server). Bằng việc đọc và phân tích các gói dữ liệu bắt được, kẻ tấn công có thể tìm thấy nhiều thông tin quan trọng để tiến hành các hình thức tấn công khác.

**-Tấn công mật khẩu (Password attack):**

Là hình thức truy xuất trái phép vào hệ thống bằng cách dò mật khẩu. Có hai kỹ thuật dò mật khẩu phổ biến:

-*Dò tuần tự (Brute force attack)*: Dò mật khẩu bằng cách thử lần lượt các tổ hợp ký tự, thông thường việc này được thực hiện tự động bằng phần mềm. Mật khẩu càng dài thì số lần thử càng lớn và do đó khó bị phát hiện hơn. Một số hệ thống quy định chiều dài tối thiểu của mật khẩu. Ngoài ra để ngăn chặn việc thử mật khẩu nhiều lần, một số hệ thống ngắt kết nối nếu liên tiếp nhận được mật khẩu sai sau một số lần nào đó.

-*Dò theo từ điển (Dictionary attack)*: thử lần lượt các mật khẩu mà người sử dụng thường dùng. Để đơn giản, người sử dụng thường có thói quen nguy hiểm là dùng những thông tin dễ nhớ để làm mật khẩu, ví dụ như tên mình, ngày sinh, số điện thoại, ... Một số hệ thống hạn chế nguy cơ này bằng cách định ra các *chính sách về mật khẩu (password policy)*, quy định độ khó tối thiểu của mật khẩu, ví dụ mật khẩu phải khác với những thông tin liên quan đến cá nhân người sử dụng, phải bao gồm cả chữ hoa và chữ thường, chữ cái và các mẫu tự khác chữ cái,...

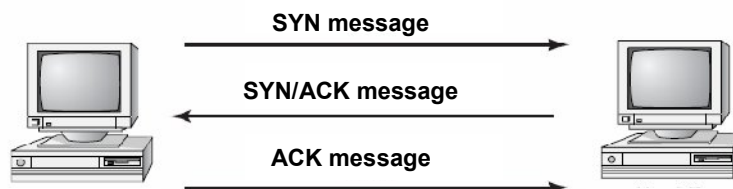
**Một số kỹ thuật tấn công dựa trên giao thức TCP/IP:**

Giao thức TCP/IP là giao thức chuẩn được sử dụng trong hầu hết các mạng máy tính, và là giao thức bắt buộc trên mạng Internet. Nhưng không may, TCP/IP chứa trong nó nhiều sơ hở về bảo mật dẫn đến những tấn công dựa trên nguyên lý của TCP/IP như sau:

**-Làm tràn kết nối TCP (TCP SYN/ACK flooding attack):**

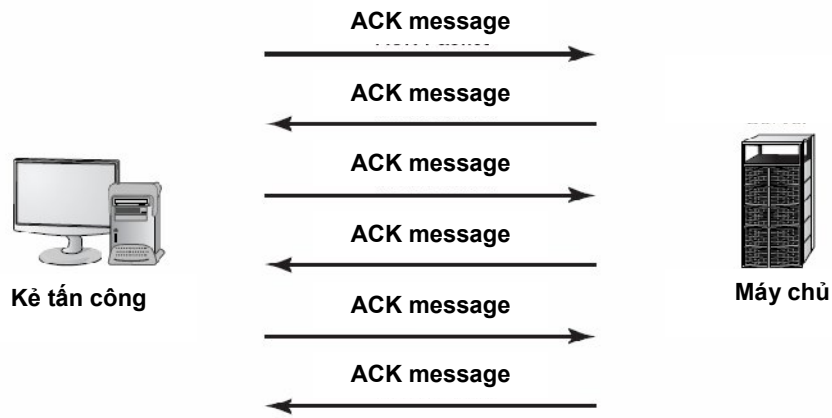
Đây là tấn công khai thác *thủ tục bắt tay ba chiều (three-way handshake)* của TCP. Mục đích của tấn công là gây ra quá tải kết nối trên máy chủ và dẫn tới từ chối dịch vụ (DoS).

Hình 1.12 mô tả thủ tục bắt tay ba chiều trong tình huống bình thường. Khi một máy (client) muốn kết nối một máy khác (server) qua một dịch vụ nào đó, nó bắt đầu bằng cách gửi bản tin *SYN* tới server trên *cổng (port)* tương ứng của dịch vụ đó. Ngay sau đó, server dành riêng một kết nối cho client này và trả lời bằng một bản tin *SYN/ACK* cho client. Để hoàn thành kết nối, client phải một lần nữa trả lời bằng một bản tin *ACK* gửi đến server. Trong trường hợp không



**Hình 1.12:** Thủ tục bắt tay ba chiều của TCP/IP

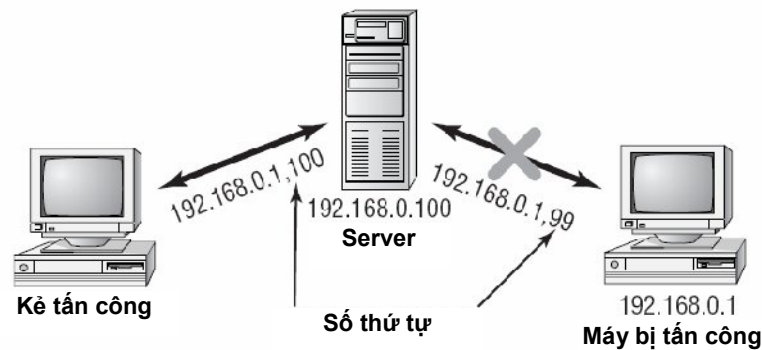
nhận được bản tin ACK trả lời từ phía client thì server phải chờ cho đến khi hết thời hiệu (timeout) rồi mới giải toả kết nối này. Với sơ hở này, nếu một kẻ tấn công cố tình tạo ra các bản ACK liên tiếp gửi đến server nhưng không hồi đáp (tức không gửi lại bản tin ACK cho server), thì đến một thời điểm nào đó, tất cả các kết nối có thể có của server đều dành hết cho việc chờ đợi này và do không có khả năng phục vụ cho các kết nối khác. Hình 1.13 trình bày phương thức tấn công dùng SYN/ACK flooding.



**Hình 1.13:** Tấn công TCP SYN/ACK flooding

**-Tấn công dựa vào số thứ tự của TCP (TCP sequence number attack):**

Trong quá trình truyền dữ liệu giữa các máy sử dụng giao thức TCP, số thứ tự (sequence number) là một thông tin quan trọng giúp xác định thứ tự các gói dữ liệu và xác nhận các gói đã được nhận thành công. Số thứ tự được đánh theo từng byte dữ liệu và được duy trì một cách đồng bộ giữa bên gửi và bên nhận. Nếu một máy thứ ba, bằng cách nào đó, chặn bắt được các gói dữ liệu đang được trao đổi và đoán được số thứ tự của quá trình truyền nhận dữ liệu, nó sẽ có khả năng xen vào kết nối, làm ngắt kết nối của một đầu và nhảy vào thay thế (hijacking). Hình 1.14 mô tả phương thức hoạt động của tấn công này.



**Hình 1.14:** Tấn công dựa vào số thứ tự TCP (TCP sequence number attack)

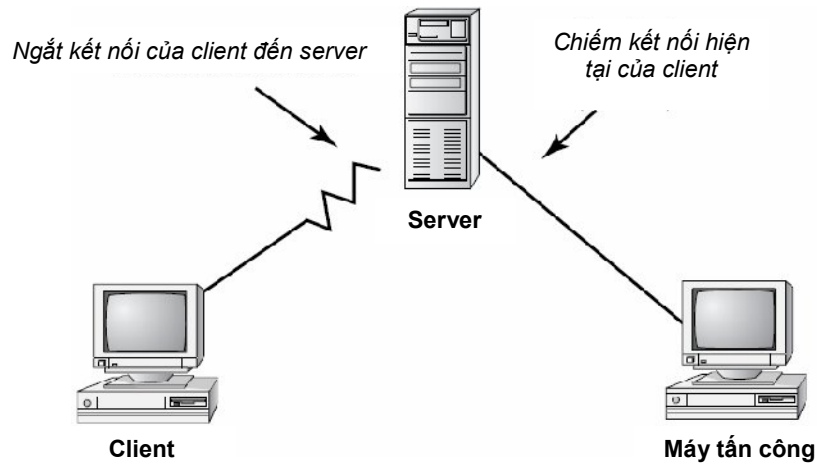
**-Chiếm kết nối TCP (TCP Hijacking):**

Giống như phương thức tấn công ở trên (sequence number attack), nhưng sau khi đoán được số thứ tự, máy tấn công sẽ cố gắng chiếm lấy một đầu của kết nối hiện hữu mà đầu kia không hay biết để tiếp tục truyền nhận dữ liệu, khi đó thông tin trao đổi giữa hai máy ban đầu bị chuyển sang một máy thứ ba. Hình 1.15 trình bày hoạt động của phương thức tấn công này.

**-Tấn công dùng giao thức ICMP (ICMP attack):**

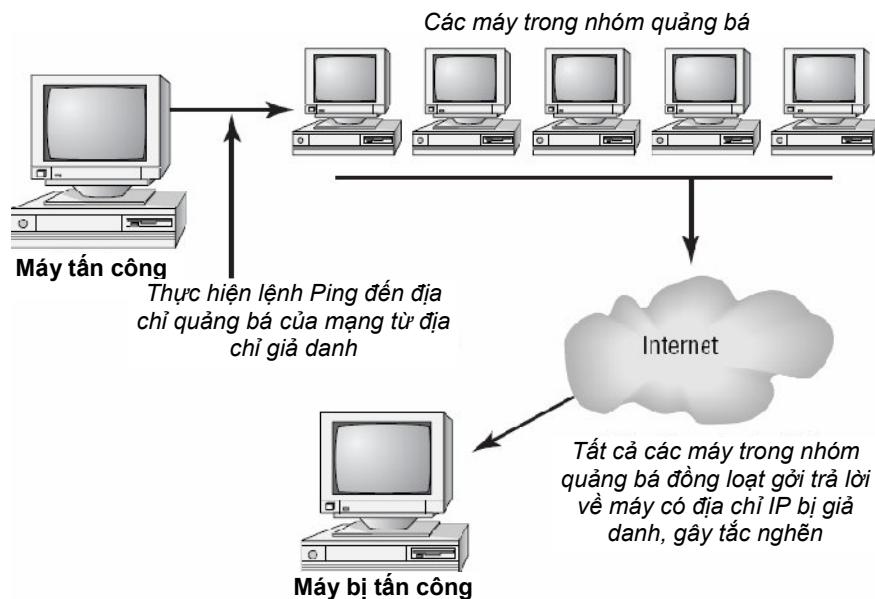
ICMP (Internet Control Message Protocol) là một giao thức điều khiển dùng trong mạng IP. Giao thức này thường được sử dụng để thực hiện các thủ tục điều khiển trên mạng IP như

kiểm tra các kết nối (ví dụ khi thực hiện các lệnh Ping, Tracert, ...). Hai phương thức tấn công phổ biến dựa trên ICMP bao gồm:



**Hình 1.15:** Chiếm kết nối TCP (TCP connection hijacking)

-*Smurf attack*: (còn được gọi là *Ping of Death*). Nguyên lý hoạt động của ICMP là hồi đáp lại (*reply*) khi nhận được các yêu cầu (*echo request*) từ các máy khác, do chức năng của ICMP là để kiểm tra các kết nối IP. Dựa vào nguyên lý này, một kẻ tấn công có thể giả danh một địa chỉ IP nào đó (*IP spoofing*) và gửi một yêu cầu (*echo request*) đến tất cả các máy trong mạng nội bộ (bằng cách sử dụng địa chỉ quảng bá *broadcast*). Ngay lập tức, tất cả các máy này đều đồng loạt trả lời cho máy có địa chỉ IP bị giả danh, dẫn đến máy này bị tắc nghẽn không còn khả năng hoạt động bình thường. Mục tiêu của tấn công *smurf* là làm tê liệt một máy nào đó bằng các gói ICMP. Hình 1.16 mô tả hoạt động của phương thức tấn công *smurf*.



**Hình 1.16:** Tấn công Ping of Death

-*ICMP tunneling*: Do gói dữ liệu ICMP thường được chấp nhận bởi nhiều máy trên mạng, nên kẻ tấn công có thể lợi dụng điều này để chuyển các thông tin không hợp lệ thông qua các gói dữ liệu ICMP. Để ngăn chặn các tấn công này, cách tốt nhất là từ chối tất cả các gói dữ liệu ICMP.

**-Tấn công khai thác phần mềm (Software exploitation):**

Đây là tên gọi chung của tất cả các hình thức tấn công nhắm vào một chương trình ứng dụng hoặc một dịch vụ nào đó ở lớp ứng dụng. Bằng cách khai thác các sơ hở và các lỗi kỹ thuật trên các phần mềm và dịch vụ này, kẻ tấn công có thể xâm nhập hệ thống hoặc làm gián đoạn hoạt động bình thường của hệ thống.

Tấn công tràn bộ đệm (*buffer overflow attack*): là phương thức tấn công vào các lỗi lập trình của số phần mềm. Lỗi này có thể do lập trình viên, do bản chất của ngôn ngữ hoặc do trình biên dịch. Ngôn ngữ C là ngôn ngữ có nhiều khả năng gây ra các lỗi tràn bộ đệm nhất, và không may, đây là ngôn ngữ vốn được dùng rộng rãi nhất trong các hệ điều hành, các chương trình hệ thống, đặc biệt trong môi trường Unix và Linux.

Đa số các trình biên dịch C không kiểm tra giới hạn vùng nhớ đã cấp phát cho các biến, do đó, khi dữ liệu lưu vào vùng nhớ vượt qua giới hạn đã cấp phát, nó sẽ ghi chồng qua những vùng nhớ kế cận và gây ra lỗi. Ví dụ: khi lập trình, mật khẩu mà người dùng nhập vào thường được xử lý dưới dạng một chuỗi (string), và được khai báo với chiều dài xác định, ví dụ 32 ký tự. Tuy nhiên, nếu trong chương trình không thực hiện việc kiểm tra chiều dài mật khẩu trước khi xử lý và trình biên dịch cũng thông tự động thực hiện việc này thì khi người sử dụng nhập mật khẩu có chiều dài lớn hơn 32 ký tự, toàn bộ chuỗi ký tự này sẽ tràn vùng nhớ đã cấp phát và có thể gây ra lỗi tràn bộ đệm.

Ngoài tấn công tràn bộ đệm, các phương thức tấn công khác nhắm vào việc khai thác các sơ hở của phần mềm và dịch vụ bao gồm: khai thác cơ sở dữ liệu (*database exploitation*), khai thác ứng dụng (*application exploitation*) ví dụ như các loại macro virus, khai thác các *phần mềm gửi thư điện tử* (e-mail exploitation), ...

**-Các kỹ thuật đánh lừa (Social engineering):**

Đây là phương thức tấn công không sử dụng kỹ thuật hay máy tính để xâm nhập hệ thống mà bằng các kỹ xảo gian lận để tìm kiếm các thông tin quan trọng, rồi thông qua đó mà xâm nhập hệ thống.

Ví dụ, một kẻ tấn công giả danh là một nhân viên hỗ trợ kỹ thuật gọi điện thoại đến một người trong hệ thống để trao đổi công việc, thông qua cuộc trao đổi này để khai thác các thông tin cần thiết để thực hiện hành vi xâm nhập hệ thống. Rõ ràng, phương thức này không sử dụng các kỹ thuật để tấn công, nên được gọi là “*social engineering*”. Đây cũng là một trong những loại tấn công phổ biến, và *đối tượng mà nó nhắm đến là vấn đề con người trong hệ thống*.

## **1.6.2 Các phương thức xâm nhập hệ thống bằng phần mềm phá hoại**

Kỹ thuật và hình thức tấn công mới thường xuyên được phát hiện và nâng cấp. Ở trên chỉ giới thiệu các hình thức tấn công phổ biến đã được phát hiện và phân tích. Ngoài các hình thức tấn công như trên, các hệ thống thông tin còn phải đối mặt với một nguy cơ xâm nhập rất lớn đó là các phần mềm virus, worm, spyware, ... gọi chung là các *phần mềm phá hoại* hay *phần mềm độc* (*malicious code*). Sau đây sẽ tập trung trình bày các hình thức xâm nhập này.

Các phần mềm độc được chia thành các nhóm sau đây: *Virus*, *worm*, *Trojan horse* và *logic bomb*.

### **-Virus:**

Là phần mềm ẩn, kích thước nhỏ và được gắn vào một tập tin chủ nào đó, thông thường là các tập tin thực thi được, nhờ đó virus mới có khả năng phá hoại và lan truyền sang các máy khác. Một số loại virus lại gắn với các tập tin tài liệu (ví dụ như word, excel, ...) và được gọi là các *virus macro*.

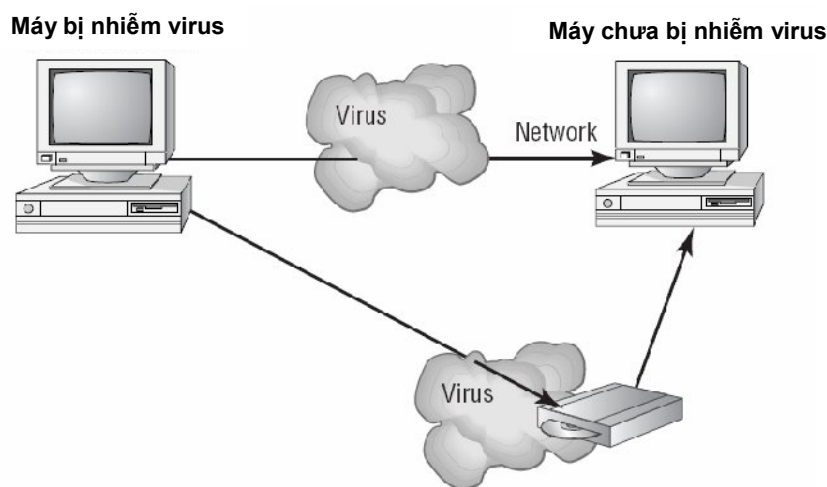
Virus lan truyền giữa các máy tính thông qua việc sao chép các tập tin có nhiễm virus từ đĩa mềm, đĩa CD, đĩa flash, hoặc thông qua các tập tin gửi kèm theo e-mail. Phạm vi phá hoại của virus là rất lớn. Thông thường nhất, các virus thường gây ra mất mát dữ liệu, hư hỏng phần mềm và hư hỏng cả hệ điều hành.

Nếu trên máy chưa cài đặt sẵn các chương trình quét virus thì dấu hiệu thông thường nhất để nhận biết có virus trên máy tính là:

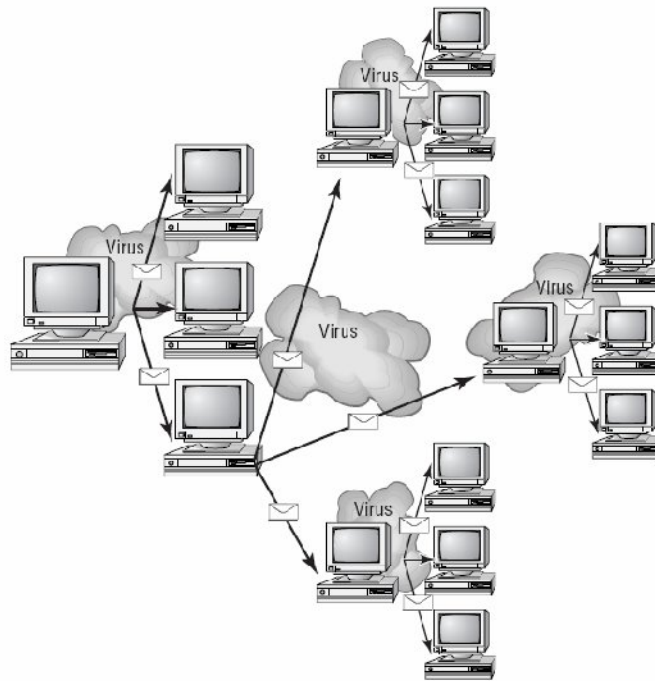
- Xuất hiện các thông báo lạ trên màn hình
- Máy tính làm việc chậm đi đáng kể, đặc biệt khi khởi động chương trình.
- Mất đột ngột một hoặc nhiều tập tin trên đĩa.
- Lỗi phần mềm không rõ lý do.
- Kích thước một số tập tin, đặc biệt là các tập tin thực thi, tăng lên bất thường.
- Máy tính tự khởi động lại khi đang làm việc
- ...

Hình 1.17 mô tả việc lây lan của virus thông qua đường sao chép tập tin (bằng đĩa hoặc qua các tập tin dùng chung trên mạng). Hình 1.18 mô tả quá trình phát tán virus thông qua email.

Có thể thấy mức độ phát tán của virus thông qua e-mail nghiêm trọng hơn nhiều, bởi vì đối với hình thức lây lan qua đường sao chép tập tin thì chỉ có các máy tính *chủ động sao chép tập tin* mới bị nhiễm virus; ngược lại trong phương thức phát tán bằng e-mail, những máy *không chủ động sao chép tập tin* cũng có khả năng bị lây nhiễm nếu vô ý mở những tập tin nhiễm virus được gửi kèm theo e-mail.



**Hình 1.17:** Virus lây lan từ máy này sang máy khác qua phương tiện lưu trữ (đĩa) hoặc qua thư mục dùng chung trên mạng



**Hình 1.18:** Virus phát tán qua e-mail

#### **-Worm:**

Là loại phần mềm độc có cơ chế hoạt động và tầm phá hoại gần giống như virus. Điểm khác nhau cơ bản giữa worm và virus là *worm có khả năng tự sao chép thông qua mạng* (trong khi virus phải nhờ vào thao tác sao chép của người sử dụng) và *tự tồn tại như một chương trình độc lập* (trong khi virus phải gắn vào một tập tin khác).

Đặc trưng cơ bản nhất của worm là tính phát tán nhanh trên phạm vi rộng bằng nhiều phương tiện khác nhau, như sử dụng trực tiếp giao thức TCP/IP, sử dụng các dịch vụ mạng ở lớp ứng dụng, phát tán qua e-mail và nhiều phương tiện khác. Worm Nimda xuất hiện năm 2001 là một worm điển hình với tốc độ phát tán cực nhanh và mức độ nguy hiểm lớn, có thể gây tê liệt các hệ thống mạng lớn sử dụng hệ điều hành Windows trong nhiều giờ.

#### **-Trojan horse:**

Một dạng phần mềm độc hoạt động núp dưới danh nghĩa một phần mềm hữu ích khác, và sẽ thực hiện các hành vi phá hoại hệ thống khi chương trình giả danh được kích hoạt bởi người sử dụng.

*Trojan không có khả năng tự sao chép* như worm (mà phải giả dạng thành một phần mềm có ích hoặc được gắn vào một phần mềm thực thi khác để được cài đặt vào máy), *không có khả năng tự thực thi* như virus (mà chỉ thực hiện khi người sử dụng khởi động chương trình).

Mức độ phá hoại của Trojan cũng rất đa dạng, trong đó quan trọng nhất là thực thi như một phần mềm gián điệp (back-door) giúp cho những kẻ tấn công từ xa có thể dễ dàng xâm nhập hệ thống. *Spyware* là một ví dụ của Trojan, đây là các phần mềm được tự động cài vào máy khi người sử dụng tải các phần mềm trên Internet về cài trên máy của mình. Spyware có thể tự động gửi e-mail, tự động mở các trang web hoặc thực hiện các hành vi khác gây ảnh hưởng đến hoạt động bình thường của máy tính bị nhiễm.

### **-Logic bomb:**

Là các phần mềm nằm ẩn trên máy tính và chỉ thực hiện khi có một sự kiện nào đó xảy ra, ví dụ khi người quản trị mạng đăng nhập vào hệ thống, khi một ứng dụng nào đó được chạy hoặc đến một ngày giờ định trước nào đó.

Thông thường, khi được thực hiện, logic bomb gửi một thông báo về một máy trung tâm định trước nào đó để thông báo sự kiện xảy ra. Nhận được thông báo này, kẻ tấn công từ máy tính trung tâm đó sẽ thực hiện tiếp các thủ thuật tấn công vào hệ thống, ví dụ khởi động một cuộc tấn công từ chối dịch vụ (DoS hoặc DDoS).

Trên đây là các phương thức xâm nhập vào hệ thống sử dụng các phần mềm phá hoại. Mặc dù sự xâm nhập vào một hệ thống cụ thể nào đó của các phần mềm này có thể không do chủ đích của một cá nhân nào, nhưng thiệt hại do các hình thức xâm nhập này gây ra là rất lớn, do tính phổ biến của nó. Bất kỳ máy nào cũng có thể bị nhiễm phần mềm độc, đặc biệt khi kết nối đến mạng Internet. Các nguyên tắc chung để tránh sự xâm nhập của các phần mềm độc vào máy tính nói riêng và vào một hệ thống thông tin nói chung bao gồm:

- Không sao chép dữ liệu từ các nguồn không tin cậy (từ đĩa hay qua mạng).

- Không cài đặt các phần mềm không rõ nguồn gốc, đặc biệt là các phần mềm download từ Internet.

- Thường xuyên cập nhật các bản sửa lỗi (*Hotfixes hoặc service pack*) cho hệ thống (cả hệ điều hành và chương trình ứng dụng).

- Cài đặt các chương trình *Antivirus*, *Antispyware* và cập nhật thường xuyên cho các chương trình này.

- Theo dõi các thông tin về các loại virus mới, phương thức hoạt động và cách thức ngăn chặn trên các trang web chuyên về bảo mật (ví dụ trang CERT tại địa chỉ <http://www.cert.org>).

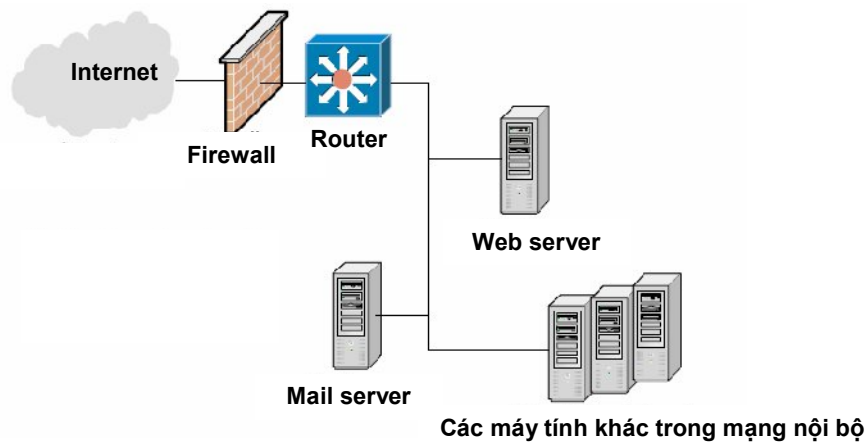
## **1.7 KỸ THUẬT NGĂN CHẶN VÀ PHÁT HIỆN XÂM NHẬP**

Sau khi nhận diện các nguy cơ và rủi ro đối với hệ thống, phân tích các phương thức và kỹ thuật tấn công có khả năng ảnh hưởng đến sự an toàn của hệ thống, các hệ thống thông tin thường triển khai các biện pháp kỹ thuật cần thiết để **ngăn chặn** và **phát hiện** xâm nhập. Phần này giới thiệu về tường lửa (*Firewall*) và hệ thống phát hiện xâm nhập (*IDS*), là hai ứng dụng bảo mật điển hình nhất hiện nay.

### **1.7.1 Tường lửa:**

*Tường lửa hay firewall* là kỹ thuật *ngăn chặn các tấn công xâm nhập từ bên ngoài (mạng Internet) vào hệ thống bên trong (mạng LAN và server)*. Hình 1.19 mô tả một cấu trúc mạng điển hình trong đó firewall được lắp đặt trước router, với vai trò bảo vệ cho toàn bộ hệ thống mạng bên trong.

Nguyên tắc chung của các bức tường lửa là điều khiển truy xuất mạng bằng cách giám sát tất cả các gói dữ liệu được gửi thông qua tường lửa, và tùy vào các cài đặt trong chính sách bảo mật mà cho phép hoặc không cho phép chuyển tiếp các gói này đến đích. Hình 1.20 mô tả hoạt động điển hình của một bức tường lửa, trong đó, lưu lượng *HTTP (TCP port 80)* được phép đi qua tường lửa, còn lưu lượng *NetBIOS (TCP port 445)* thì bị chặn lại.



**Hình 1.19:** Bức tường lửa đặt trước Router để bảo vệ toàn bộ mạng bên trong

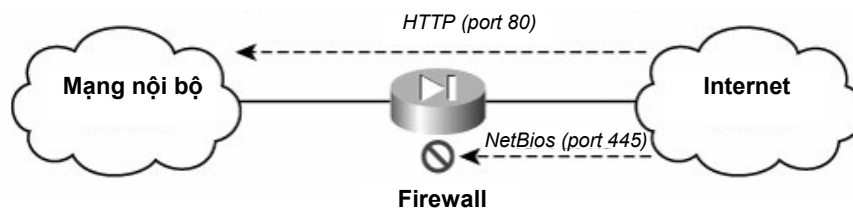
Chức năng của tường lửa trên mạng là quản lý lưu lượng vào/ra trên kết nối Internet và ghi lại các sự kiện diễn ra trên kết nối này phục vụ cho các mục đích an toàn mạng. Tuy nhiên, do bản chất của tường lửa là giám sát lưu lượng luân chuyển thông qua một kết nối giữa mạng nội bộ và mạng công cộng bên ngoài, cho nên *tường lửa không có khả năng giám sát và ngăn chặn các tấn công xuất phát từ bên trong mạng nội bộ*. Có thể tóm tắt chức năng chủ yếu của tường lửa như sau:

-*Separator*: Tách rời giữa mạng nội bộ và mạng công cộng, ràng buộc tất cả các kết nối từ trong ra ngoài hoặc từ ngoài vào trong phải đi qua tường lửa như một đường đi duy nhất.

-*Restrictor*: Chỉ cho phép một số lượng giới hạn các loại lưu lượng được phép xuyên qua tường lửa, nhờ đó người quản trị có thể thực thi chính sách bảo mật bằng cách thiết lập các quy tắc lọc gói tương ứng gọi là các *access rules*.

-*Analyzer*: Theo dõi (tracking) lưu lượng luân chuyển qua tường lửa, ghi lại các thông tin này lại (logging) theo yêu cầu của người quản trị để phục vụ cho các phân tích để đánh giá mức độ an toàn của hệ thống.

Ngoài các chức năng cơ bản trên, một số bức tường lửa còn có chức năng xác thực (authentication) đối với người sử dụng trước khi chấp nhận kết nối.



**Hình 1.20:** Hoạt động cơ bản của bức tường lửa

**\*-Phân loại tường lửa theo đặc tính kỹ thuật:**

Tường lửa có thể là một phần mềm chạy trên một máy tính nào đó với ít nhất là hai giao tiếp mạng (*dual-home host*), khi đó nó được gọi là **firewall mềm**. Các firewall mềm thông dụng hiện nay gồm: *SunScreen, ISA server, Check point, Gauntlet, IPTables,...*



Ngược lại, chức năng tường lửa cũng có thể được thực hiện trong một khối phần cứng riêng biệt và được gọi là **firewall cứng**. Các sản phẩm firewall cứng điển hình hiện nay bao gồm: *Cisco PIX, NetScreen firewalls, SonicWall appliances, WatchGuard Fireboxes, Nokia firewalls, ...*

**\*-Phân loại firewall theo phạm vi bảo vệ:**

Căn cứ vào phạm vi mà tường lửa bảo vệ, có thể chia tường lửa thành 2 nhóm riêng biệt: tường lửa dành cho máy tính cá nhân (*personal firewalls*) và tường lửa dành cho mạng (*network firewalls*).

**-Personal firewall** thông thường là các firewall mềm, được cài đặt trên máy cá nhân để bảo vệ cho máy cá nhân. Hệ điều hành Windows (2000 và XP) đã có tích hợp sẵn personal firewall. Ngoài ra, các phần mềm antivirus chuyên nghiệp cũng có chức năng của personal firewall như *Norton Antivirus, McAfee, ...*

**-Network firewall** có thể là firewall mềm hoặc firewall cứng, thường được lắp đặt trước hoặc sau bộ định tuyến (router) nhằm mục đích bảo vệ cho toàn hệ thống mạng.

**\*-Phân loại firewall theo cơ chế làm việc:**

Dựa trên cơ chế làm việc, firewall được chia thành 3 loại như sau:

**-Tường lửa lọc gói (packet filtering firewall hay stateless firewall)**

Nguyên lý của các bức tường lửa lọc gói là đọc tất cả các thông tin trong tiêu đề của các gói dữ liệu IP luân chuyển qua bức tường lửa, và dựa trên các thông tin này để quyết định chấp nhận (*accept*) hay loại bỏ gói dữ liệu (*drop*). Như vậy, khi thiết lập các quy tắc lọc gói của tường lửa, người quản trị mạng phải căn cứ trên các thông tin sau đây:

-Địa chỉ IP, bao gồm địa chỉ IP của máy gửi và địa chỉ IP của máy nhận (source IP address và destination IP address).

-Số cổng kết nối (port number), bao gồm cả cổng của máy gửi và cổng của máy nhận (source port và destination port)

-Giao thức kết nối (protocol), ví dụ TCP, UDP hay ICMP.

*Packet filtering firewall chỉ phân tích tiêu đề của gói IP, không phân tích nội dung gói và do đó không có khả năng ngăn chặn truy xuất theo nội dung dữ liệu.*

Packet filtering firewall hữu ích trong các trường hợp muốn ngăn chặn một hoặc một số cổng xác định nào đó, từ chối một hoặc một số địa chỉ IP xác định hoặc một giao thức xác định nào đó (ví dụ ICMP). Trong thực tế, các tấn công xâm nhập thường được thực hiện thông qua các cổng khác với các cổng dịch vụ phổ biến. Bảng 1.1 liệt kê danh sách một số dịch vụ thông dụng trên Internet và số cổng tương ứng.

**-Tường lửa lớp ứng dụng (Application Layer gateway):**

Hoạt động của tường lửa lớp ứng dụng tương tự như tường lửa lọc gói, tức là cũng dựa trên việc phân tích các gói dữ liệu IP để quyết định có cho phép đi xuyên qua bức tường lửa hay không. Điểm khác của tường lửa lớp ứng dụng là nó *có khả năng phân tích cả nội dung của gói dữ liệu IP (phần data payload)*, và do đó cho phép thiết lập các quy tắc lọc gói phức tạp hơn. Ví dụ, có thể chấp nhận lưu lượng HTTP đi qua bức tường lửa, tuy nhiên với những gói nào có chứa nội dung trùng với mẫu định trước thì chặn lại.

Do đặc tính của tường lửa lớp ứng dụng can thiệp trực tiếp vào tất cả các gói dữ liệu đi qua nó, nên nhìn dưới góc độ truy xuất mạng, bức tường lửa lớp ứng dụng trực tiếp thực hiện các

giao dịch với mạng bên ngoài thay cho các máy tính bên trong. Do vậy, tường lửa lớp ứng dụng cũng còn được gọi là các phần mềm Proxy.

Kỹ thuật này có ích trong các trường hợp cần quản lý nội dung truy cập của người sử dụng hoặc để nhận dạng dấu hiệu của một số loại phần mềm độc (virus, worm, trojan, ...), ví dụ ngăn chặn người sử dụng tải các tập tin hình ảnh hoặc phim với kích thước lớn.

Do phải phân tích toàn bộ cấu trúc gói dữ liệu để lấy thông tin nên nhược điểm của tường lửa lớp ứng dụng là yêu cầu năng lực xử lý mạnh, và là nơi có thể xảy ra tắc nghẽn tiềm năng của mạng.

***-Tường lửa kiểm soát trạng thái (stateful inspection firewall):***

Là loại tường lửa kết hợp cả hai nguyên lý làm việc của tường lửa lọc gói và tường lửa lớp ứng dụng.

Tường lửa kiểm soát trạng thái cho phép thiết lập các quy tắc lọc gói phức tạp hơn so với tường lửa lọc gói, tuy nhiên không mất quá nhiều thời gian cho việc phân tích nội dung của tất cả các gói dữ liệu như trường hợp tường lửa lớp ứng dụng. Tường lửa kiểm soát trạng thái theo dõi trạng thái của tất cả các kết nối đi qua nó và các gói dữ liệu liên quan đến từng kết nối. Theo đó, chỉ các gói dữ liệu thuộc về các kết nối hợp lệ mới được chấp nhận chuyển tiếp qua tường lửa, các gói khác đều bị loại bỏ tại đây.

Tường lửa kiểm soát trạng thái phức tạp hơn do phải tích hợp chức năng của cả 2 loại tường lửa ở trên. Tuy nhiên, cơ chế thực hiện của tường lửa này đã chứng tỏ được tính hiệu quả của nó và trong thực tế, các sản phẩm tường lửa mới đều hỗ trợ kỹ thuật này.

**Bảng 1.1:** Một số dịch vụ phổ biến trên TCP

<b>Cổng</b>	<b>Dịch vụ</b>
20	FTP, kênh điều khiển (Control port)
21	FTP, kênh dữ liệu (Data port)
22	Secure Shell (SSH)
23	Telnet
25	Simple Mail Transfer Protocol (SMTP)
80	HyperText Transfer Protocol (HTTP)
110	Post Office Protocol, version 3 (POP3)
143	Internet Message Access Protocol
443	Secure Sockets Layer (SSL)

### **1.7.2 Hệ thống phát hiện xâm nhập:**

*Hệ thống phát hiện xâm nhập IDS (Intrusion Detection System) là hệ thống phát hiện các dấu hiệu của tấn công xâm nhập. Khác với bức tường lửa, IDS không thực hiện các thao tác ngăn chặn truy xuất mà chỉ theo dõi các hoạt động trên mạng để tìm ra các dấu hiệu của tấn công và cảnh báo cho người quản trị mạng.*

IDS không thực hiện chức năng phân tách giữa mạng nội bộ và mạng công cộng như bức tường lửa nên không gánh toàn bộ lưu lượng qua nó và do đó không có nguy cơ làm tắc nghẽn mạng.

*Intrusion (xâm nhập) được định nghĩa là bất kỳ một sự kiện hay hành vi nào tác động vào 3 thành phần cơ bản của một hệ thống an toàn là tính Bảo mật, tính Toàn vẹn và tính Khả dụng.*

IDS phát hiện dấu vết của tấn công bằng cách phân tích hai nguồn thông tin chủ yếu sau đây:

1-Thông tin về các thao tác thực hiện trên máy chủ được lưu trong nhật ký hệ thống (system log)

2-Lưu lượng đang lưu thông trên mạng.

Chức năng ban đầu của IDS chỉ là phát hiện các dấu hiệu xâm nhập, do đó IDS chỉ có thể tạo ra các cảnh báo tấn công khi tấn công đang diễn ra hoặc thậm chí sau khi tấn công đã hoàn tất. Càng về sau, nhiều kỹ thuật mới được tích hợp vào IDS, giúp nó có khả năng dự đoán được tấn công (*prediction*) và thậm chí phản ứng lại các tấn công đang diễn ra (*Active response*).

Hai thành phần quan trọng nhất cấu tạo nên hệ thống IDS là *sensor* (bộ cảm nhận) có chức năng chặn bắt và phân tích lưu lượng trên mạng và các nguồn thông tin khác để phát hiện dấu hiệu xâm nhập; *signature database* là cơ sở dữ liệu chứa dấu hiệu (signature) của các tấn công đã được phát hiện và phân tích. Cơ chế làm việc của signature database giống như virus database trong các chương trình antivirus, do vậy, việc duy trì một hệ thống IDS hiệu quả phải bao gồm việc cập nhật thường xuyên cơ sở dữ liệu này.

\*-Phân loại IDS theo phạm vi giám sát:

Dựa trên phạm vi giám sát, IDS được chia thành 2 loại:

**-Networ- based IDS (NIDS):**

Là những IDS giám sát trên toàn bộ mạng. Nguồn thông tin chủ yếu của NIDS là các gói dữ liệu đang lưu thông trên mạng. NIDS thường được lắp đặt tại ngõ vào của mạng, có thể đứng trước hoặc sau bức tường lửa. Hình 1.21 mô tả một NIDS điển hình.

**-Host-based IDS (HIDS):**

Là những IDS giám sát hoạt động của từng máy tính riêng biệt. Do vậy, nguồn thông tin chủ yếu của HIDS ngoài lưu lượng dữ liệu đến và đi từ máy chủ còn có hệ thống dữ liệu nhật ký hệ thống (system log) và kiểm tra hệ thống (system audit).

Hình 1.22 trình bày cấu trúc của HIDS. IDS được thiết kế để phối hợp với hệ điều hành để xử lý các thông tin giám sát hệ thống. Dịch vụ nhật ký hệ thống (logging) ghi lại các sự kiện và trạng thái của hệ thống vào một cơ sở dữ liệu (Event database). Ngoài ra, kết quả giám sát trên mạng của IDS cũng được ghi vào Event Database. Để phát hiện xâm nhập, IDS duy trì một cơ sở dữ liệu (IDS database) chứa các mô tả về từng loại tấn công.

\*-Phân loại IDS theo kỹ thuật thực hiện:

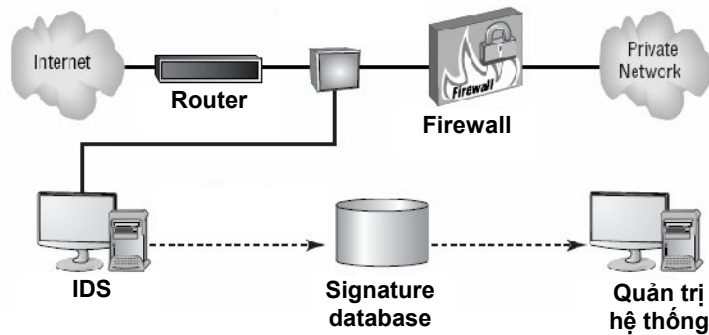
Dựa trên kỹ thuật thực hiện, IDS cũng được chia thành 2 loại:

**-Signature-based IDS:**

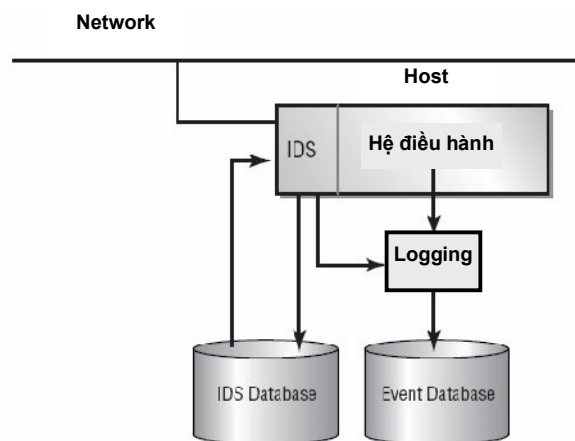
Signature-based IDS phát hiện xâm nhập dựa trên dấu hiệu của hành vi xâm nhập, thông qua phân tích lưu lượng mạng và nhật ký hệ thống. Kỹ thuật này đòi hỏi phải duy trì một cơ sở dữ liệu về các dấu hiệu xâm nhập (signature database), và cơ sở dữ liệu này phải được cập nhật thường xuyên mỗi khi có một hình thức hoặc kỹ thuật xâm nhập mới.

**-Anomaly-based IDS:** phát hiện xâm nhập bằng cách so sánh (mang tính thống kê) các hành vi hiện tại với hoạt động bình thường của hệ thống để phát hiện các bất thường (anomaly) có thể là dấu hiệu của xâm nhập. Ví dụ, trong điều kiện bình thường, lưu lượng trên một giao tiếp

mạng của server là vào khoảng 25% băng thông cực đại của giao tiếp. Nếu tại một thời điểm nào đó, lưu lượng này đột ngột tăng lên đến 50% hoặc hơn nữa, thì có thể giả định rằng server đang bị tấn công DoS.



**Hình 1.21:** *Network-based IDS (NIDS)*



**Hình 1.22:** *Host-based IDS (HIDS)*

Để hoạt động chính xác, các IDS loại này phải thực hiện một quá trình “*học*”, tức là giám sát hoạt động của hệ thống trong điều kiện bình thường để ghi nhận các thông số hoạt động, đây là cơ sở để phát hiện các bất thường về sau.

Trong thực tế, IDS là một kỹ thuật mới so với firewall, tuy nhiên, cho đến thời điểm này, với sự phát triển khá mạnh mẽ của kỹ thuật tấn công thì IDS vẫn chưa thật sự chứng tỏ được tính hiệu quả của nó trong việc đảm bảo an toàn cho các hệ thống mạng. Một trong những phần mềm IDS phổ biến hiện nay là Snort. Đây là một sản phẩm NIDS mã nguồn mở với hệ thống signature database (được gọi là *rule database*) được cập nhật thường xuyên bởi nhiều thành viên trong cộng đồng Internet.

Tóm tắt chương:

-Một hệ thống thông tin an toàn là hệ thống đảm bảo được 3 đặc trưng cơ bản:

-Tính Bảo mật (Confidentiality)

-Tính Toàn vẹn (Integrity)

-Tính Khả dụng (Availability)

Ba đặc trưng này được gọi tắt là CIA.

-Chiến lược cơ bản nhất để đảm bảo tính bảo mật của một hệ thống thông tin:

-Access Control

-Authentication

-Auditing

Kỹ thuật này gọi tắt là AAA.

-Nguy cơ (threat) của một hệ thống thông tin là các sự kiện, hành vi có khả năng ảnh hưởng đến 3 đặc trưng CIA của hệ thống. Rủi ro đối với hệ thống thông tin là xác suất xảy ra các thiệt hại đối với hệ thống.

-Chính sách bảo mật (security policy) định nghĩa các trạng thái an toàn của hệ thống, các hành vi mà người sử dụng được phép hoặc không được phép thực thi. Cơ chế bảo mật (security mechanism) là các biện pháp kỹ thuật (technical) hoặc thủ tục (procedure) nhằm đảm bảo chính sách. Nguyên tắc xây dựng một hệ thống thông tin an toàn bao gồm xây dựng chính sách bảo mật để định nghĩa một cách chính xác và đầy đủ các trạng thái an toàn của hệ thống, sau đó thiết lập các cơ chế để đảm bảo thực thi chính sách.

-Có nhiều hình thức xâm nhập / tấn công khác nhau trên hệ thống. Các tấn công này dựa trên các sơ hở về an toàn của giao thức (TCP/IP), của hệ điều hành (Windows, Linux, ...) hoặc của các chương trình ứng dụng chạy trên các hệ điều hành đó. Kỹ thuật tấn công luôn luôn được phát triển và hoàn thiện, do đó công nghệ an toàn mạng cũng phải được phát triển tương xứng.

-Hai giải pháp kỹ thuật giúp phát hiện và ngăn chặn các tấn công trên một hệ thống thông tin là IDS và Firewall. IDS giám sát hệ thống để phát hiện các dấu hiệu tấn công và tạo ra cảnh báo. Firewall ngăn chặn hoặc cho phép các truy xuất thông qua Firewall theo các quy luật định trước (access rules).

## **CÂU HỎI VÀ BÀI TẬP.**

### **A- Câu hỏi trắc nghiệm**

Câu 1. Thế nào là tính bảo mật của hệ thống thông tin?

- a- Là đặc tính của hệ thống trong đó thông tin được giữ bí mật không cho ai truy xuất.
- b- Là đặc tính của hệ thống trong đó tất cả thông tin được lưu trữ dưới dạng mật mã.
- c- Là đặc tính của hệ thống trong đó chỉ có những người dùng được cho phép mới có thể truy xuất được thông tin
- d- Tất cả đều đúng

Câu 2. Chọn câu đúng khi nói về tính bảo mật của hệ thống thông tin:

- a- Một hệ thống đảm bảo tính bí mật (confidential) là một hệ thống an toàn (secure).
- b- Tính bí mật của thông tin bao gồm tính bí mật về sự tồn tại của thông tin và tính

bí mật nội dung thông tin.

- c- Tính bí mật của thông tin bao gồm tính bí mật về nội dung thông tin và tính bí mật về nguồn gốc thông tin.
- d- Tất cả đều sai.

Câu 3. Thế nào là tính toàn vẹn của hệ thống thông tin?

- a- Là đặc tính của hệ thống trong đó thông tin không bị sửa đổi hoặc xoá bỏ bởi người sử dụng.
- b- Là đặc tính của hệ thống trong đó thông tin không bị thay đổi theo thời gian
- c- Là đặc tính của hệ thống trong đó thông tin không bị truy xuất bởi những người không được phép.
- d- Là đặc tính của hệ thống trong đó thông tin không bị thay đổi, hư hỏng hay mất mát.

Câu 4. Chọn câu đúng khi nói về tính toàn vẹn của thông tin:

- a- Một hệ thống an toàn là một hệ thống đảm bảo tính toàn vẹn của thông tin.
- b- Tính toàn vẹn của thông tin bao gồm toàn vẹn về nội dung và toàn vẹn về nguồn gốc thông tin.
- c- Tính toàn vẹn của thông tin bao gồm toàn vẹn về nội dung và sự tồn tại của thông tin.
- d- Câu a và b.

Câu 5. Các cơ chế đảm bảo tính toàn vẹn của thông tin:

- a- Gồm các cơ chế ngăn chặn và cơ chế phát hiện các vi phạm về toàn vẹn thông tin.
- b- Mật mã hoá toàn bộ thông tin trong hệ thống.
- c- Lưu toàn bộ thông tin trong hệ thống dưới dạng nén.
- d- Tất cả các cơ chế trên.

Câu 6. Hành vi nào sau đây ảnh hưởng đến tính toàn vẹn của hệ thống thông tin:

- a- Một sinh viên sao chép bài tập của một sinh viên khác.
- b- Virus xóa mất các tập tin trên đĩa cứng.
- c- Mất điện thường xuyên làm hệ thống máy tính làm việc gián đoạn.
- d- Tất cả các hành vi trên.

Câu 7. Hành vi nào sau đây ảnh hưởng đến tính khả dụng của hệ thống thông tin:

- a- Một sinh viên sao chép bài tập của một sinh viên khác.
- b- Virus xóa mất các tập tin trên đĩa cứng.
- c- Mất điện thường xuyên làm hệ thống máy tính làm việc gián đoạn.
- d- Tất cả các hành vi trên.

Câu 8. Hành vi nào sau đây ảnh hưởng đến tính bí mật của hệ thống thông tin:

- a- Một sinh viên sao chép bài tập của một sinh viên khác.
- b- Virus xóa mất các tập tin trên đĩa cứng.
- c- Mất điện thường xuyên làm hệ thống máy tính làm việc gián đoạn.
- d- Tất cả các hành vi trên.

Câu 9. Các cơ chế bảo vệ tính bí mật của thông tin:

- a- Mật mã hoá toàn bộ thông tin trong hệ thống.
- b- Xây dựng các cơ chế điều khiển truy xuất (access control) phù hợp.
- c- Lắp đặt các phương tiện bảo vệ hệ thống thông tin ở mức vật lý.
- d- Tất cả các cơ chế trên.

Câu 10. Thế nào là tính khả dụng của hệ thống thông tin?

- a- Là tính sẵn sàng của thông tin trong hệ thống cho mọi nhu cầu truy xuất.
- b- Là tính sẵn sàng của thông tin trong hệ thống cho các nhu cầu truy xuất hợp lệ.
- c- Là tính dễ sử dụng của thông tin trong hệ thống.
- d- Tất cả đều sai.

Câu 11. Thế nào là nguy cơ đối với hệ thống thông tin?

- a- Là các sự kiện, hành vi ảnh hưởng đến sự an toàn của hệ thống thông tin.
- b- Là các thiệt hại xảy ra đối với hệ thống thông tin
- c- Là các hành vi vô ý của người sử dụng làm ảnh hưởng đến tính khả dụng của hệ thống thông tin.
- d- Tất cả đều đúng.

Câu 12. Các nguy cơ nào sau đây có thể ảnh hưởng đến tính khả dụng của hệ thống thông tin:

- a- Thiết bị không an toàn.
- b- Các tấn công từ chối dịch vụ (DoS và DDoS).
- c- Virus và các loại phần mềm phá hoại khác trên máy tính.
- d- Tất cả các nguy cơ trên.

Câu 13. Chọn câu sai khi nói về các nguy cơ đối với sự an toàn của hệ thống thông tin:

- a- Những kẻ tấn công hệ thống (attacker) có thể là con người bên trong hệ thống.
- b- Người sử dụng không được huấn luyện về an toàn hệ thống cũng là một nguy cơ đối với hệ thống.
- c- Một hệ thống không kết nối vào mạng Internet thì không có các nguy cơ tấn công.
- d- Xâm nhập hệ thống (intrusion) có thể là hành vi xuất phát từ bên ngoài hoặc từ bên trong hệ thống.

Câu 14. Chọn câu đúng khi nói về các nguy cơ và rủi ro đối với hệ thống thông tin:

- a- Tất cả các rủi ro đều có ít nhất một nguy cơ đi kèm với nó.
- b- Có thể ngăn chặn rủi ro bằng cách ngăn chặn các nguy cơ tương ứng.
- c- Mục tiêu của an toàn hệ thống là ngăn chặn tất cả các rủi ro xảy ra trên hệ thống.
- d- Tất cả các câu trên.

Câu 15. Nguyên tắc xây dựng một hệ thống bảo mật:

- a- Áp dụng các cơ chế an toàn phù hợp với hệ thống.
- b- Xây dựng các chính sách an toàn chặt chẽ.
- c- Xây dựng chính sách bảo mật và triển khai các cơ chế để đảm bảo chính sách đó.
- d- Tất cả đều đúng.

Câu 16. Mục tiêu của chính sách bảo mật hệ thống:

- a- Xác định các trạng thái an toàn mà hệ thống cần đảm bảo.
- b- Ngăn chặn các nguy cơ đối với hệ thống.
- c- Hạn chế các rủi ro đối với hệ thống.
- d- Tất cả các câu trên.

Câu 17. Mục tiêu của an toàn hệ thống theo thứ tự ưu tiên giảm dần:

- a- Ngăn chặn, phát hiện, phục hồi.
- b- Phát hiện, ngăn chặn, phục hồi.
- c- Phát hiện và ngăn chặn.
- d- Phát hiện và phục hồi.

Câu 18. Chọn câu đúng khi nói về các mô hình điều khiển truy xuất (access control):

- a- MAC là cơ chế điều khiển bắt buộc được áp dụng cho toàn hệ thống
- b- Cơ chế quản lý theo nhóm trên Windows 2000 là một dạng thực thi tương đương với cơ chế RBAC.
- c- Đa số các hệ điều hành đều có thực hiện mô hình DAC.
- d- Tất cả đều đúng.

Câu 19. Các cơ chế xác thực thông dụng trong hệ thống thông tin:

- a- Dùng các cơ chế quản lý truy xuất tập tin trên đĩa cứng.
- b- Dùng cơ chế phân quyền cho người sử dụng.
- c- Dùng user-name/password.
- d- Tất cả đều sai.

Câu 20. Các giao thức xác thực thông dụng trong hệ thống thông tin:

- a- Kerberos
- b- CHAP
- c- Cả hai đều sai
- d- Cả hai đều đúng..

Câu 21. Chức năng của cơ chế kiểm tra (auditing) trên hệ thống:

- a- Ghi lại (Logger), phân tích (Analyzer) và thông báo (Notifier).
- b- Theo dõi và ghi nhận các sự kiện và hành vi diễn ra trên hệ thống.
- c- Cung cấp thông tin để phục hồi hệ thống khi có sự cố.
- d- Cung cấp thông tin làm chứng cứ cho các hành vi vi phạm chính sách an toàn hệ thống.

Câu 22. Chọn câu đúng:

- a- Tấn công kiểu Interception tác động vào đặc tính toàn vẹn của hệ thống thông tin.
- b- Modification là kiểu tấn công vào đặc tính bí mật của hệ thống thông tin.
- c- Tấn công bằng hình thức giả danh (farbrication) tác động đến đặc tính toàn vẹn của thông tin.
- d- Vấn đề phủ nhận hành vi (repudiation) là một hình thức tấn công hệ thống kiểu Interruption.

Câu 23. Phương thức tấn công nào ngăn chặn các user hợp lệ truy xuất các tài nguyên hệ thống?



- a- Sniffing
- b- Spoofing
- c- DoS
- d- Man-In-The-Middle.

Câu 24. Chọn câu đúng:

- a- Có thể ngăn chặn các tấn công tràn bộ đệm (buffer overflow) bằng các phần mềm antivirus.
- b- Có thể ngăn chặn các tấn công tràn bộ đệm bằng cách cài đặt firewall.
- c- Tất cả các phần mềm viết bằng ngôn ngữ C đều có chứa lỗi tràn bộ đệm.
- d- Lỗi tràn bộ đệm chỉ xảy ra trên các phần mềm có nhập liệu từ người dùng.

Câu 25. Một máy tính nghe lén thông tin trên mạng và dùng các thông tin này để xâm nhập trái phép vào một hệ thống thông tin, đây là phương thức tấn công nào?

- a- Spoofing
- b- Replay
- c- Man-In-The-Middle
- d- Sniffing

Câu 26. Phương thức tấn công nào sau đây không dựa trên bản chất của giao thức TCP/IP:

- a- SYN/ACK flooding
- b- TCP sequence number attack
- c- ICMP attack
- d- Software exploitation

Câu 27. Chọn câu đúng khi nói về các phương thức tấn công bằng phần mềm độc (malicious code):

- a- Virus có thể tự sao chép và lan truyền thông qua mạng máy tính.
- b- Worm là loại phần mềm độc hoạt động dựa vào một phần mềm khác.
- c- Trojan horse là một loại phần mềm độc nhưng có tên giống như các tập tin bình thường.
- d- Logic bomb không thể phá hoại hệ thống nếu đồng hồ hệ thống luôn chậm hơn thời gian hiện hành.

Câu 28. Chọn câu đúng khi nói về firewall:

- a- Firewall chỉ có thể ngăn chặn các tấn công từ bên ngoài hệ thống.
- b- Tất cả các gói dữ liệu đi qua firewall đều bị đọc toàn bộ nội dung, nhờ đó firewall mới có cơ sở để phân biệt các tấn công với các loại lưu lượng khác.
- c- Nếu mở tất cả các cổng (port) trên firewall thì firewall sẽ hoàn toàn bị vô hiệu hoá.
- d- Tất cả đều đúng.

Câu 29. Ứng dụng nào sau đây có chức năng thay đổi địa chỉ IP của tất cả các gói dữ liệu đi qua nó:

- a- IDS
- b- Proxy

- c- NAT
- d- Không có ứng dụng nào như vậy

Câu 30. Nguyên lý hoạt động của IDS:

- a- Phân tích các gói dữ liệu lưu thông trên mạng để tìm dấu hiệu của tấn công.
- b- Phân tích các dữ liệu trong nhật ký hệ thống (system log) để phát hiện dấu hiệu của tấn công.
- c- Duy trì một cơ sở dữ liệu về các dấu hiệu tấn công (signature database).
- d- Tất cả các điều trên.

Câu 31. Chọn câu đúng khi nói về IDS:

- a- IDS là một ứng dụng có chức năng phát hiện và ngăn chặn các tấn công vào hệ thống thông tin.
- b- IDS chỉ có thể phát hiện được các tấn công từ bên ngoài vào hệ thống.
- c- Network-based IDS không có khả năng phát hiện tấn công vào một máy chủ cụ thể.
- d- Signature-based IDS không có khả năng phát hiện các tấn công hoàn toàn mới, chưa từng được mô tả trong cơ sở dữ liệu.

## **B- Bài tập**

Câu 32. Liệt kê và sắp xếp các phương thức tấn công theo hai loại: tấn công chủ động (active attacks) và tấn công thụ động (passive attacks).

Câu 33. Liệt kê và sắp xếp các phương thức tấn công theo hai loại: tấn công vào giao thức TCP/IP và tấn công vào phần mềm (chương trình ứng dụng và hệ điều hành).

Câu 34. Cài đặt và cấu hình phần mềm IDS Snort trên Hệ điều hành Linux.

Câu 35. Cài đặt và cấu hình ISA server 2004 trên Windows.



## CHƯƠNG II

# MẬT MÃ VÀ XÁC THỰC THÔNG TIN

### Giới thiệu:

Chương này trình bày cơ chế mật mã và các vấn đề liên quan như hàm băm, chữ ký số, chứng thực và cơ sở hạ tầng khoá công khai PKI. Mật mã là cơ chế cơ bản nhất nhằm đảm bảo tính Bí mật của thông tin. Các cơ chế xác thực như hàm băm và chữ ký số có chức năng bảo vệ tính Toàn vẹn của thông tin. Các nội dung đề cập trong chương này bao gồm:

- Tổng quan về kỹ thuật mật mã.
- Kỹ thuật mật mã đối xứng
- Kỹ thuật mật mã bất đối xứng
- Các hàm băm bảo mật
- Chữ ký số
- Vấn đề quản lý khoá và cơ sở hạ tầng khoá công khai

## II.1 TỔNG QUAN VỀ MẬT MÃ:

### II.1.1 Giới thiệu:

**Mật mã (Encryption)** là một kỹ thuật cơ sở quan trọng trong bảo mật thông tin. Nguyên tắc của mật mã là biến đổi thông tin gốc thành dạng thông tin bí mật mà chỉ có những thực thể tham gia xử lý thông tin một cách hợp lệ mới hiểu được.

Một thực thể hợp lệ có thể là một người, một máy tính hay một phần mềm nào đó được phép nhận thông tin. Để có thể giải mã được thông tin mật, thực thể đó cần phải biết cách giải mã (tức là biết được thuật toán giải mã) và các thông tin cộng thêm (khóa bí mật).

Quá trình chuyển thông tin gốc thành thông tin mật theo một thuật toán nào đó được gọi là *quá trình mã hoá (encryption)*. Quá trình biến đổi thông tin mật về dạng thông tin gốc ban đầu gọi là *quá trình giải mã (decryption)*. Đây là hai quá trình không thể tách rời của một kỹ thuật mật mã bởi vì mật mã (giấu thông tin) chỉ có ý nghĩa khi ta có thể giải mã (phục hồi lại) được thông tin đó. Do vậy, khi chỉ dùng thuật ngữ *mật mã* thì nó có nghĩa bao hàm cả *mã hóa* và *giải mã*.

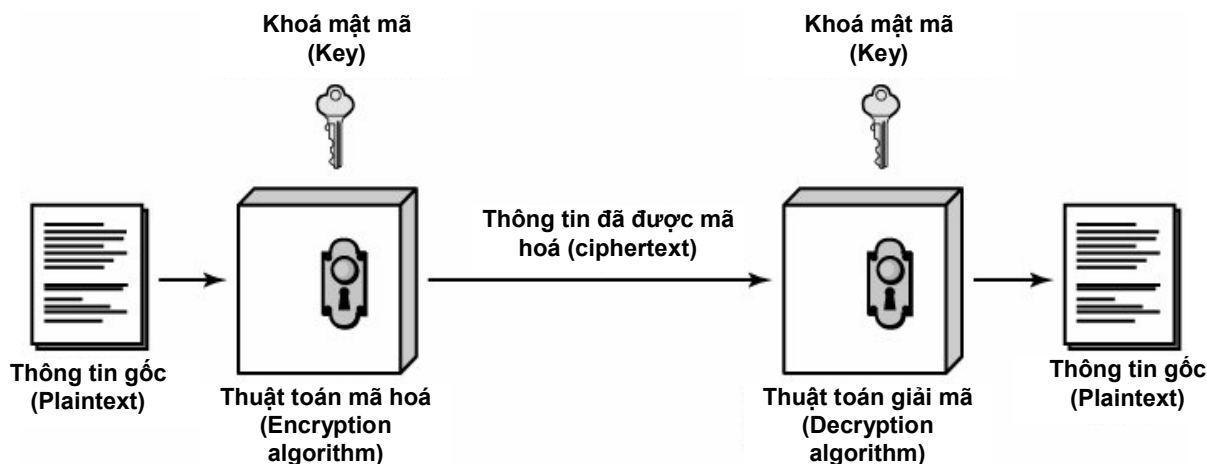
Kỹ thuật mã hoá được chia thành hai loại: mã hoá dùng khoá đối xứng (symmetric key encryption) và mã hoá dùng khoá bất đối xứng (asymmetric key encryption) như sẽ trình bày trong các phần tiếp theo.

### II.1.2 Các thành phần của một hệ thống mã hoá:

Hình 2.1 mô tả nguyên tắc chung của một hệ thống mật mã quy ước. Các thành phần trong một hệ thống mật mã điển hình bao gồm:

- Plaintext:** là *thông tin gốc* cần truyền đi giữa các hệ thống thông tin
- Encryption algorithm:** *thuật toán mã hóa*, đây là cách thức tạo ra *thông tin mật* từ thông tin gốc.
- Key:** *khóa mật mã*, gọi tắt là *khóa*. Đây là thông tin cộng thêm mà thuật toán mã hóa sử dụng để trộn với thông tin gốc tạo thành *thông tin mật*.
- Ciphertext:** *thông tin đã mã hóa (thông tin mật)*. Đây là kết quả của thuật toán mã hóa.

**-Decryption algorithm:** Thuật toán giải mã. Đầu vào của thuật toán này là thông tin đã mã hóa (ciphertext) cùng với khóa mật mã. Đầu ra của thuật toán là thông tin gốc (plaintext) ban đầu.



Hình 2.1: Cấu trúc một hệ thống mật mã quy ước

### II.1.3 Các tiêu chí đặc trưng của một hệ thống mã hoá:

Một hệ thống mã hóa bất kỳ được đặc trưng bởi 3 tiêu chí sau đây:

**-Phương pháp mã (operation):** có hai phương pháp mật mã bao gồm *thay thế (substitution)* và *chuyển vị (transposition)*. Trong phương pháp mã thay thế, các đơn vị thông tin (bit, ký tự, byte hoặc khối) trong thông tin gốc được *thay thế* bằng các đơn vị thông tin khác theo một quan hệ nào đó. Trong phương pháp mã chuyển vị, các đơn vị thông tin trong thông gốc được *đổi chỗ* cho nhau để tạo thành thông tin mã hóa. Các hệ thống mã hoá hiện đại thường kết hợp cả hai phương pháp thay thế và chuyển vị.

**-Số khóa sử dụng (number of keys):** nếu phía mã hóa (phía gửi) và phía giải mã (phía nhận) sử dụng chung một khóa, ta có hệ thống mã dùng khoá đối xứng (*symmetric key*) - gọi tắt là *mã đối xứng* hay còn có các tên gọi khác như *mã một khóa (single-key)*, *mã khóa bí mật (secret key)* hoặc *mã quy ước (conventional cryptosystem)*. Nếu phía mã hóa và phía giải mã dùng 2 khóa khác nhau, hệ thống này được gọi là *mã bất đối xứng (asymmetric key)*, *mã hai khóa (two key)* hoặc *mã khóa công khai (public key)*.

**-Cách xử lý thông tin gốc (mode of cipher):** thông tin gốc có thể được xử lý liên tục theo từng phần tử, khi đó ta có hệ thống mã dòng (*stream cipher*). Ngược lại, nếu thông tin gốc được xử lý theo từng khối, ta có hệ thống mã khối (*block cipher*). Các hệ thống mã dòng thường phức tạp và không được phổ biến công khai, do đó chỉ được dùng trong một số ứng dụng nhất định (ví dụ trong thông tin di động GSM). Các thuật toán mật mã được giới thiệu trong tài liệu này chỉ tập trung vào cơ chế mã khối.

### II.1.4 Tấn công một hệ thống mật mã:

*Tấn công (attack)* hay *bẻ khoá (crack)* một hệ thống mật mã là quá trình thực hiện việc giải mã thông tin mật một cách trái phép. Thuật ngữ *cryptanalysis* được dùng để chỉ hành vi bẻ khoá và người thực hiện bẻ khoá được gọi là *cryptanalyst*.

Thông thường, đây là hành vi của một kẻ tấn công khi muốn xâm nhập vào một hệ thống đã được bảo vệ bằng mật mã. Theo nguyên tắc mật mã, để lấy được thông tin gốc, thì tác nhân

giải mã phải có được 3 thành phần: *thông tin mật (ciphertext)*, *khóa (secret key)* và *thuật toán giải mã (decryption algorithm)*. Kẻ tấn công thường không có đầy đủ 3 thông tin này, do đó, thường cố gắng để giải mã thông tin bằng hai phương pháp sau:

**-Phương pháp phân tích mã (cryptanalysis):** dựa vào bản chất của thuật toán mã hóa, cùng với một đoạn thông tin gốc hoặc thông tin mật có được, kẻ tấn công tìm cách phân tích để tìm ra toàn bộ thông tin gốc hoặc tìm ra khóa, rồi sau đó thực hiện việc giải mã toàn bộ thông tin mật.

**-Phương pháp thử tuần tự (brute-force):** bằng cách thử tất cả các khóa có thể, kẻ tấn công có khả năng tìm được khóa đúng và do đó giải mã được thông tin mật.

Thông thường, để tìm được khóa đúng thì cần phải thử một số lượng khóa bằng khoảng một nửa số khóa có thể có của hệ thống mã. Ví dụ, nếu khóa có chiều dài là 8 bit thì sẽ có tất cả  $2^8 = 256$  khóa khác nhau. Để chọn được khóa đúng thì kẻ tấn công phải thử trung bình khoảng  $256 / 2 = 128$  lần. Việc thử này thường được trợ giúp bởi các máy tính và phần mềm chuyên nghiệp.

Hai thành phần đảm bảo sự an toàn của một hệ thống mật mã là **thuật toán mã** (bao gồm thuật toán mã hoá và thuật toán giải mã) và **khóa**.

Trong thực tế, thuật toán mã không được xem như một thông tin bí mật, bởi vì mục đích xây dựng một thuật toán mã là để phổ biến cho nhiều người dùng và cho nhiều ứng dụng khác nhau, hơn nữa việc che giấu chi tiết của một thuật toán chỉ có thể tồn tại trong một thời gian ngắn, sẽ có một lúc nào đó, thuật toán này sẽ được tiết lộ ra, khi đó toàn bộ hệ thống mã hóa trở nên vô dụng. Do vậy, tất cả các tình huống đều giả thiết rằng kẻ tấn công đã biết trước thuật toán mã.

Như vậy, thành phần quan trọng cuối cùng của một hệ thống mã là khóa của hệ thống, khóa này phải được giữ bí mật giữa các thực thể tham gia nên được gọi là *khóa bí mật*.

Một cách tổng quát, chiều dài khóa càng lớn thì thời gian cần thiết để dò ra khóa bằng cách thử càng lớn, do vậy khả năng phát hiện khóa càng thấp. Bảng sau đây liệt kê một số khóa với độ dài khác nhau và thời gian cần thiết để dò ra khóa.

**Bảng 2.1:** Quan hệ giữa độ dài khóa và thời gian dò khóa.

<i>Chiều dài khóa (bit)</i>	<i>Số khóa tối đa</i>	<i>Thời gian dò khóa với tốc độ thử 1 khóa /ms</i>	<i>Thời gian dò khóa với tốc độ thử <math>10^6</math> khóa /ms</i>
32	$2^{32} = 4,3 * 10^9$	$2^{31} \text{ ms} = 35,8 \text{ phút}$	2,15 milli giây
56	$2^{56} = 7,2 * 10^{16}$	$2^{55} \text{ ms} = 1.142 \text{ năm}$	10,01 giờ
128	$2^{128} = 3,4 * 10^{38}$	$2^{127} \text{ ms} = 5,4 * 10^{24} \text{ năm}$	$5,4 * 10^{18} \text{ năm}$
168	$2^{168} = 3,7 * 10^{50}$	$2^{167} \text{ ms} = 5,9 * 10^{36} \text{ năm}$	$5,9 * 10^{30} \text{ năm}$
26 ký tự (hoán vị)	$26! = 4 * 10^{26}$	$2 * 10^{26} \text{ ms} = 6,4 * 10^{12} \text{ năm}$	$6,4 * 10^6 \text{ năm}$

## II.2 KỸ THUẬT MẬT MÃ ĐỐI XỨNG:

Kỹ thuật mật mã đối xứng được đặc trưng bởi việc *sử dụng một khóa duy nhất cho cả quá trình mã hóa và giải mã thông tin*. Bằng một cách an toàn nào đó, khóa chung này phải được trao

đổi thống nhất giữa bên gửi và bên nhận (tức bên mã hóa và bên giải mã), đồng thời được giữ bí mật trong suốt thời gian sử dụng.

Kỹ thuật mật mã đối xứng còn được gọi là mật mã quy ước (conventional encryption) hoặc mật mã dùng khóa bí mật (secret key encryption).

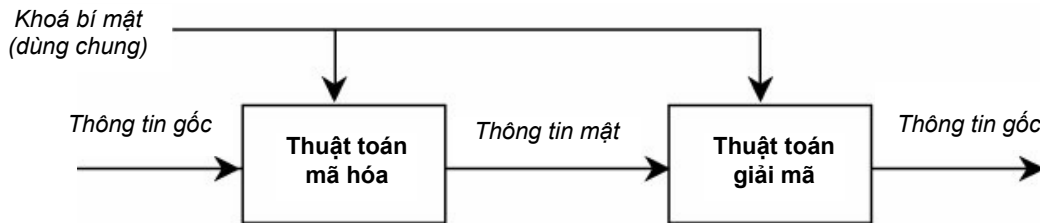
Cấu trúc chung của một hệ thống mật mã hóa quy ước như trình bày ở hình 2.2, trong đó, kênh thông tin dùng để trao đổi khóa bí mật phải là một kênh an toàn. Có thể thực hiện việc trao đổi khóa bí mật giữa hai thực thể A và B theo những cách sau đây:

1-A chọn ra một khóa bí mật và chuyển trực tiếp cho B (chuyển bằng phương tiện vật lý như ghi lên đĩa, nói trực tiếp, ghi ra giấy, ...)

2-Một thực thể thứ 3 chọn ra khóa bí mật và thông báo khóa này cho cả A và B (bằng phương tiện vật lý như trên)

3-Nếu A và B trước đó đã dùng một khóa nào đó để thông tin với nhau, thì một trong hai thực thể sẽ tiếp tục dùng khóa cũ để gửi thông báo về khóa mới cho thực thể kia.

4-Nếu A và B có các kết nối an toàn đến một thực thể thứ 3 là C, thì C có thể gửi thông báo về khóa cho cả hai thực thể A và B thông qua kết nối an toàn này.



**Hình 2.2:** Trao đổi khoá trong mật mã đối xứng

Mã hóa đối xứng dựa chủ yếu trên hai thao tác: thay thế và chuyển vị.

Thao tác thay thế sẽ thay từng từ mã bởi một từ mã khác theo một quy ước nào đó, và quy ước này chính là khóa của hệ thống mã. Ví dụ: thay thế từng ký tự trong một thông điệp bằng một ký tự đứng cách nó 3 vị trí trong bảng chữ cái la tinh, thông điệp “HELLO WORLD” sẽ được mã hóa thành “KHOOR ZRUOG”.

Thao tác chuyển vị thực hiện việc thay thế vị trí của các từ mã trong thông tin gốc theo một quy ước nào đó và quy ước này cũng trở thành khóa của hệ thống. Ví dụ: dịch từng ký tự trong một thông điệp qua phải một vị trí có xoay vòng, thông điệp “HELLO WORLD” sẽ được mã hóa thành “DHELLO WORL”.

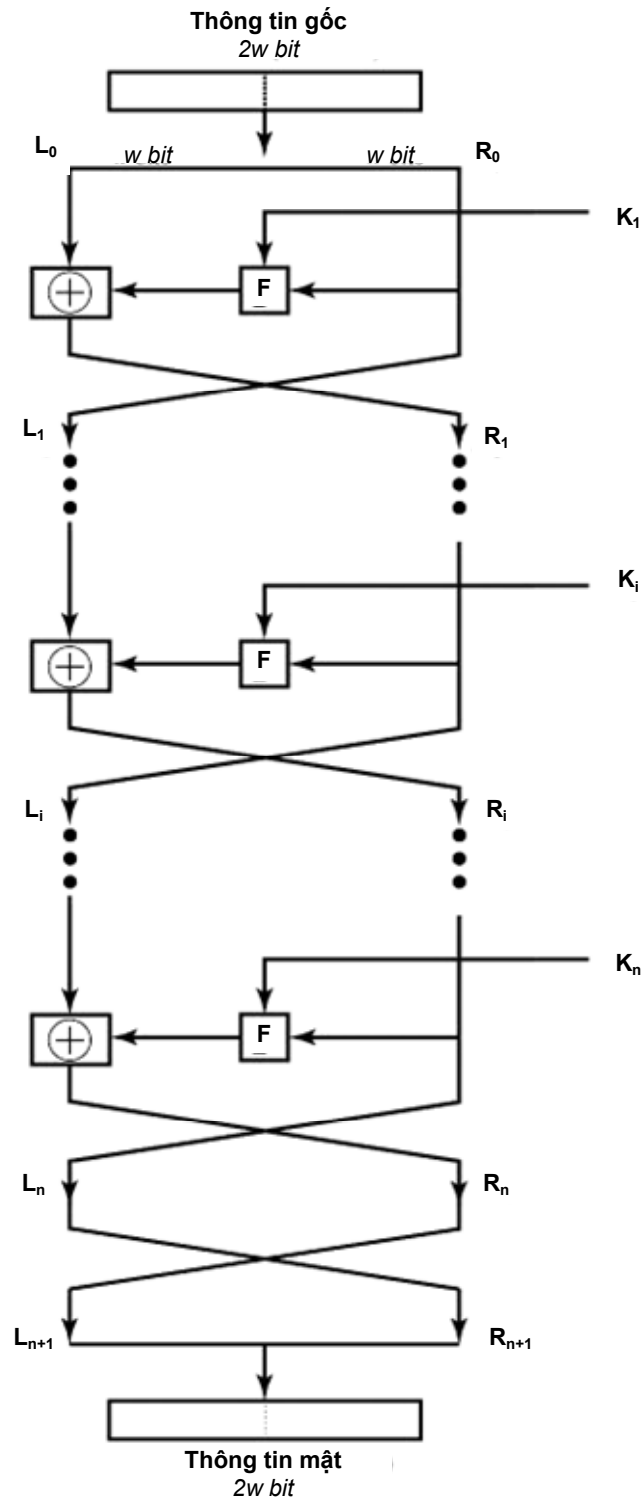
## II.2.1 Cấu trúc mã khối cơ bản Feistel:

Cấu trúc mã khối cơ bản Feistel (Feistel Cipher Structure) được IBM đưa ra vào năm 1973, được xem như là cấu trúc mật mã cơ bản nhất và được áp dụng trong nhiều thuật toán mật mã phổ biến hiện nay như DES, Blowfish, IDEA, ... Cần chú ý rằng Feistel chưa phải là một thuật toán mật mã, mà chỉ là một mô hình được xây dựng phù hợp cho việc thiết kế các thiết bị mật mã bằng phần cứng. Các thuật toán mật mã phải thực hiện hoàn chỉnh mô hình Feistel theo yêu cầu của mình, bao gồm việc định nghĩa các hàm F, S-Box và thuật toán tạo khóa phụ (subkey generation algorithm). Cấu trúc Feistel được trình bày ở hình 2.3.

Nguyên lý hoạt động của Feistel dựa trên việc *hoán vị và thay thế nhiều lần trên khối dữ liệu gốc*, cụ thể như sau:

-Thông tin gốc được cắt thành từng khối có kích thước  $2w$  bit (tức là một số bit chẵn). Mỗi khối bit được xử lý thành 2 phần bằng nhau:  $w$  bit bên trái (L) và  $w$  bit bên phải (R).

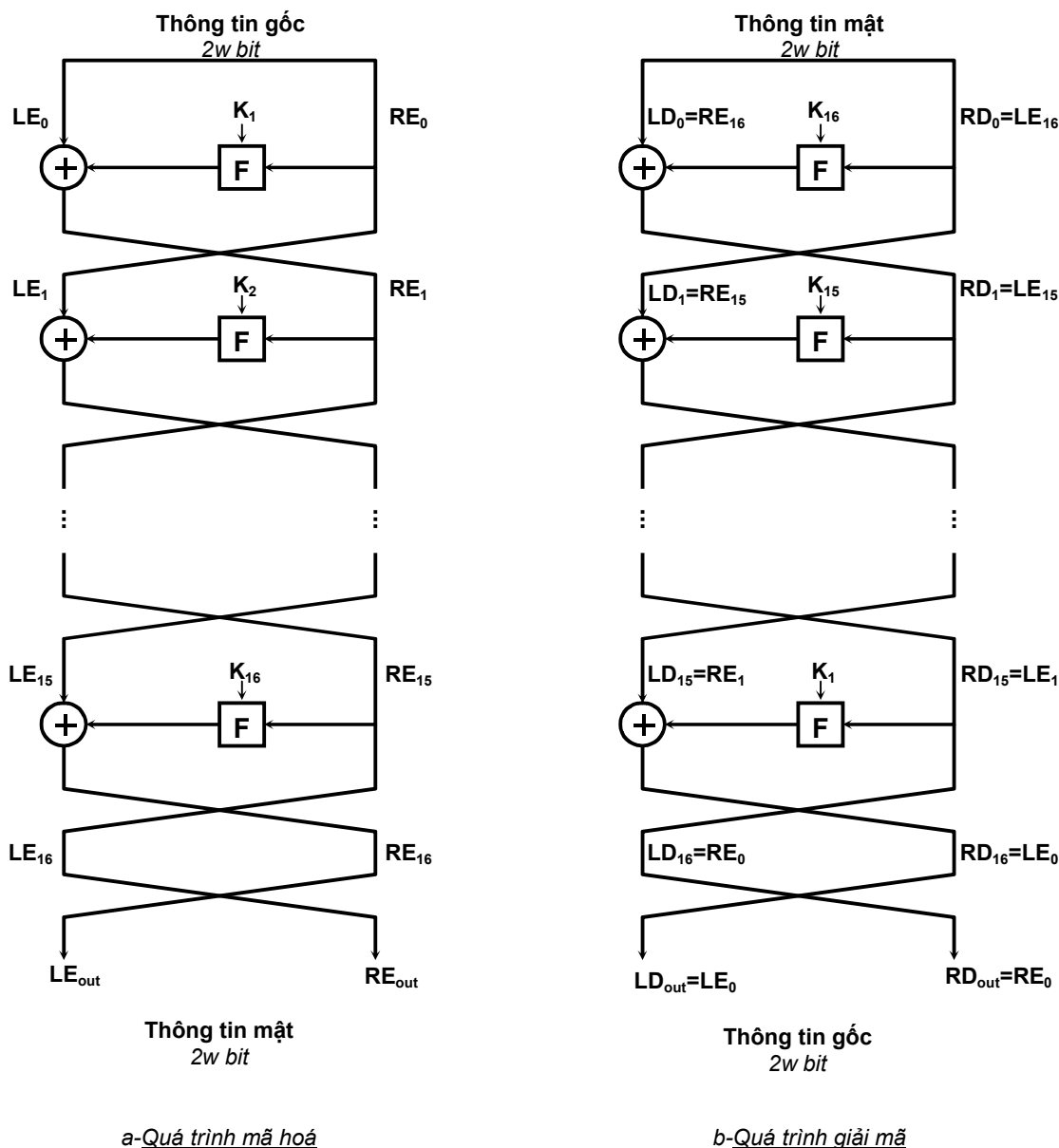
-Cả hai phần bên trái và bên phải được đưa lần lượt vào khối mã hoá gồm  $n$  vòng liên tiếp và giống nhau. Các thao tác thực hiện tại mỗi vòng bao gồm: hoán vị phần bên trái và phần bên phải, đưa phần bên phải vào một hàm xử lý  $F$  cùng với khoá con  $K_i$ , ngõ ra sẽ được XOR với phần bên trái. Kết quả cuối cùng được hoán vị một lần nữa trước khi xuất ra.



**Hình 2.3:** Cấu trúc mã khối Feistel

Quá trình giải mã của Feistel tương tự như quá trình mã hoá, chỉ khác ở chỗ thứ tự các khoá phụ đưa vào tại mỗi vòng bị đảo ngược so với quá trình mã hoá, nghĩa là khoá  $K_n$  sẽ đưa vào vòng thứ nhất, khoá  $K_1$  đưa vào vòng cuối cùng. Cũng vì lý do này, *tất cả các thao tác trong cấu trúc Feistel, kể cả hàm  $F$ , đều không cần phải có thao tác ngược*.

Quá trình giải mã được minh hoạ ở hình 2.4, cụ thể cho trường hợp Feistel sử dụng 16 vòng. Ta sẽ chứng minh được rằng ngõ ra của thuật toán giải mã chính là thông tin gốc ban đầu. Từ kết quả chứng minh này, ta có thể áp dụng tương tự cho thuật toán Feistel bất kỳ với  $n$  vòng.



**Hình 2.4:** Mã hoá và giải mã dùng cấu trúc Feistel

Để phân biệt giữa quá trình mã hoá và quá trình giải mã, ta ký hiệu các khối thông tin tại từng vòng như sau:

- $LE_i$  và  $RE_i$ : ngõ vào bên trái và bên phải của thuật toán mã hóa ở vòng thứ  $i$ .



- $LD_i$  và  $RD_i$ : ngõ vào bên trái và bên phải của thuật toán giải mã ở vòng thứ  $i$ .
- $F(RE_i, K_i)$ : áp dụng hàm  $F$  lên khối thông tin  $RE_i$  và khoá  $K_i$ .

Xét vòng cuối cùng (vòng 16) của quá trình mã hoá:

$$\begin{aligned} LE_{16} &= RE_{15} \\ RE_{16} &= LE_{15} \oplus F(RE_{15}, K_{16}) \end{aligned} \quad (1)$$

Khi đưa ngõ ra của quá trình mã hoá vào ngõ vào của quá trình giải mã, chú ý lần hoán vị sau cùng của quá trình mã hoá, ta có:

$$\begin{aligned} LD_0 &= LE_{out} = RE_{16} \\ RD_0 &= RE_{out} = LE_{16} \end{aligned} \quad (2)$$

Xét vòng thứ nhất của quá trình giải mã, ta có:

$$\begin{aligned} LD_1 &= RD_0 \\ RD_1 &= LD_0 \oplus F(RD_0, K_{16}) \end{aligned} \quad (3)$$

Kết hợp (1), (2) và (3) ta có:

$$\begin{aligned} LD_1 &= RE_{15} \\ RD_1 &= RE_{16} \oplus F(LE_{16}, K_{16}) = [LE_{15} \oplus F(RE_{15}, K_{16})] \oplus F(RE_{15}, K_{16}) = LE_{15} \end{aligned}$$

Do với phép XOR, ta luôn có:

$$\begin{aligned} A \oplus A &= 0 \\ (A \oplus B) \oplus C &= A \oplus (B \oplus C). \end{aligned}$$

Một cách tổng quát, tại vòng thứ  $i$  của quá trình mã hoá:

$$\begin{aligned} LE_i &= RE_{i-1} \\ RE_i &= LE_{i-1} \oplus F(RE_{i-1}, K_i) \end{aligned}$$

Hay có thể viết:

$$\begin{aligned} RE_{i-1} &= LE_i \\ LE_{i-1} &= RE_i \oplus F(RE_{i-1}, K_i) = RE_i \oplus F(LE_i, K_i) \end{aligned} \quad (4)$$

Với (4), ta hoàn toàn có thể kiểm chứng được kết quả của từng vòng giải mã như ở hình 2.4b. Ví dụ ở vòng thứ 2:

$$\begin{aligned} LD_2 &= RD_1 = LE_{15} = RE_{14} \\ RD_2 &= LD_1 \oplus F(RD_1, K_{15}) = RE_{15} \oplus F(LE_{15}, K_{15}) = LE_{14} \end{aligned}$$

Ở vòng thứ 16, ta có:

$$\begin{aligned} LD_{16} &= RD_{15} = LE_1 = RE_0 \\ RD_{16} &= LD_{15} \oplus F(RD_{15}, K_1) = RE_1 \oplus F(LE_1, K_1) = LE_0 \end{aligned}$$

Lần hoán vị sau cùng cho ra:

$$LD_{out} = LE_0 \text{ và } RD_{out} = RD_0, \text{ đây chính là thông tin gốc ban đầu.}$$

Các thuật toán mật mã dựa trên cấu trúc Feistel phân biệt với nhau bởi các thông số sau đây:

- 1-Kích thước khối dữ liệu đầu vào (block size)
- 2-Chiều dài khoá (key size)
- 3-Số vòng lặp (number of rounds)

4-Thuật toán sinh khoá phụ (subkey generation algorithm)

5-Hàm F thực hiện tại mỗi vòng (round function)

Đây là những thông số chưa được xác định trong cấu trúc Feistel.

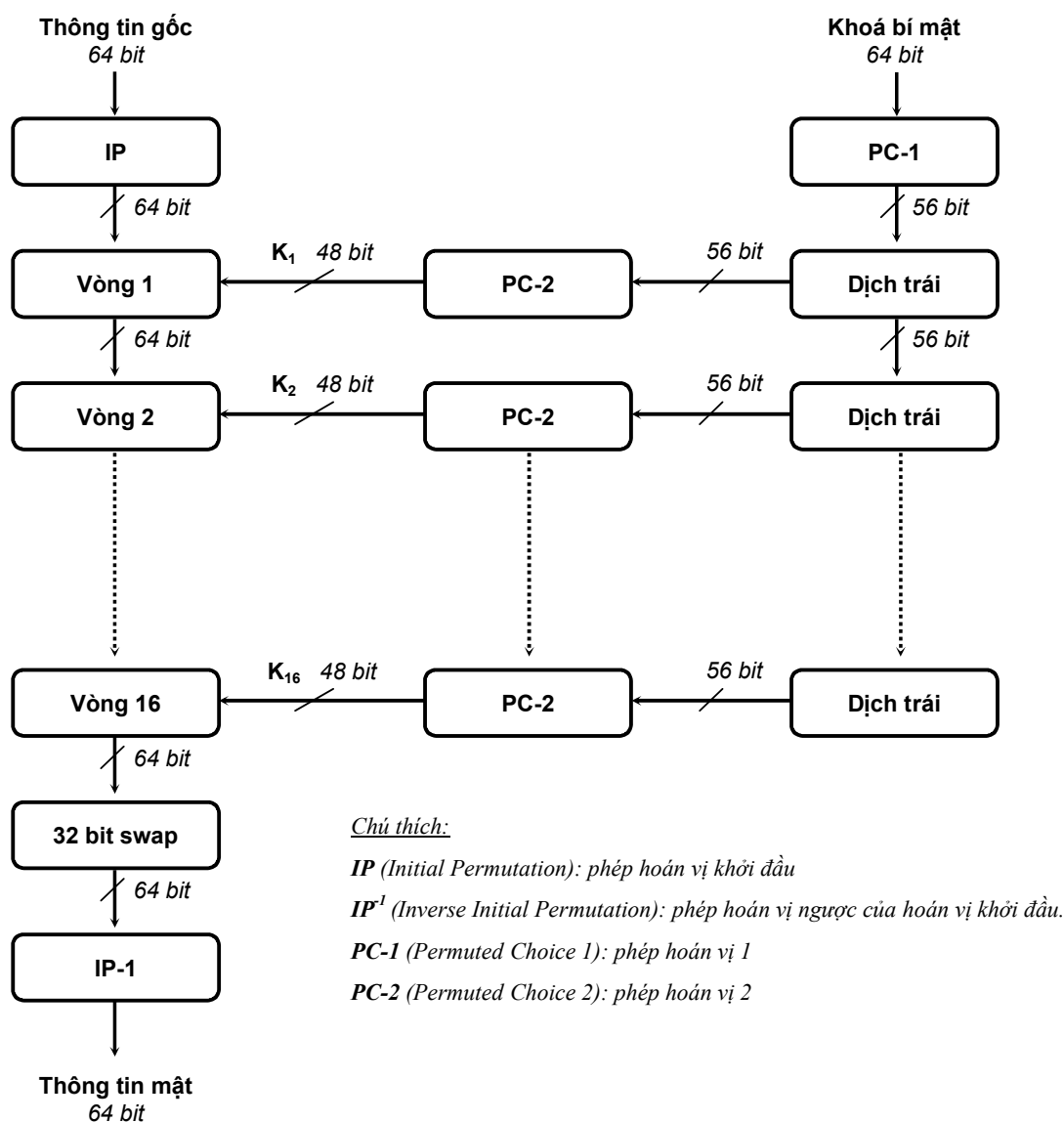
Ngoài ra, hai tiêu chí khác cần quan tâm khi thiết kế thuật toán mã dựa trên Feistel:

- Đạt tốc độ tối đa khi cài đặt bằng phần mềm.
- Dễ phân tích và thực hiện.

## II.2.2 Thuật toán mật mã DES:

DES (Data Encryption Standard) là một thuật toán mã dựa trên cấu trúc Feistel được chuẩn hóa năm 1977 bởi cơ quan chuẩn hóa Hoa kỳ (NIST – National Institute of Standards and Technology).

Cơ chế thực hiện mã hóa DES được mô tả ở hình 2.5.



Hình 2.5: Thuật toán mật mã DES

DES xác định các thông số của cấu trúc Feistel như sau:

- Kích thước khối: 64 bit
- Chiều dài khoá: 64 bit, thực ra là 56 bit như sẽ trình bày sau đây
- Số vòng lặp: 16 vòng
- Thuật toán sinh khoá phụ: kết hợp phép dịch trái và hoán vị
- Hàm F: kết hợp các phép XOR, hoán vị và thay thế (S-box).

Chi tiết thực hiện các thông số của DES được trình bày sau đây:

**-Phép hoán vị khởi đầu (IP):** có chức năng làm thay đổi vị trí các bit trong khối thông tin gốc. Đây là phần thực hiện không có trong cấu trúc Feistel. Ở phần cuối của quá trình mã hoá, phép hoán vị ngược sẽ trả lại các bit về vị trí ban đầu của nó. Phép hoán vị IP và  $IP^{-1}$  thực hiện dựa trên hai ma trận như sau, với các giá trị trong ma trận cho biết số thứ tự của bit trong khối thông tin (từ 1 đến 64):

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

**Hình 2.6:** Ma trận hoán vị khởi đầu (IP)

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

**Hình 2.7:** Ma trận ngược của ma trận hoán vị khởi đầu ( $IP^{-1}$ )

64 bit trong khối thông tin ( $M_1, M_2, \dots, M_{64}$ ) được ánh xạ vào các vị trí tương ứng trong ma trận IP và  $IP^{-1}$ , sau đó được đọc ra tuần tự theo từng dòng từ trên xuống.

Ví dụ: đối với phép hoán vị IP, bit  $M_1$  được ghi vào vị trí cột 8 dòng 5, bit  $M_2$  được ghi vào cột 8 dòng 1 và tiếp tục như thế đến bit  $M_{64}$  được ghi vào cột 1 dòng 4. Sau đó, khối thông tin này được đọc ra lần lượt từng dòng, khi đó 8 bit đầu tiên tương ứng với dòng đầu tiên sẽ là các bit có thứ tự là: 58, 50, 42, 34, 26, 18, 10, 2. Hay nói cách khác, chuỗi bit:

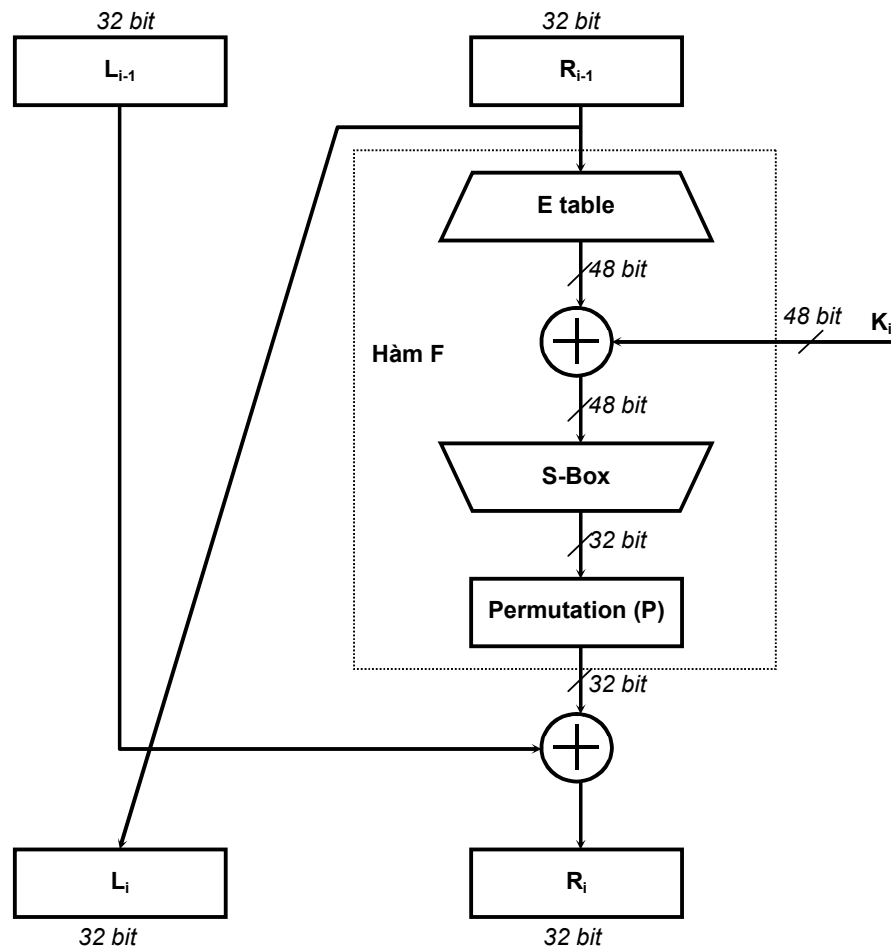
$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$	$M_7$	$M_8$
-------	-------	-------	-------	-------	-------	-------	-------

được hoán vị thành chuỗi bit:

$M_{58}$	$M_{50}$	$M_{42}$	$M_{34}$	$M_{26}$	$M_{18}$	$M_{10}$	$M_2$
----------	----------	----------	----------	----------	----------	----------	-------

**-Hàm F:** có chức năng trộn giữa khoá phụ  $K_i$  với khối thông tin tại từng vòng. Hàm F trong DES gồm có thao tác: hoán vị mở rộng (E table) chuyển từ 32 bit thành 48 bit, hàm XOR cộng 48 bit vừa tạo ra với 48 bit của khoá phụ  $K_i$ , khối thay thế S-Box chuyển 48 bit thành 32 bit, cuối cùng là khối hoán vị P.

Hoạt động của hàm F tại từng vòng được mô tả ở hình 2.8.



**Hình 2.8:** Cấu trúc từng vòng của DES

E table (Expansion/Permutation) thực hiện chức năng hoán vị các bit trong khối thông tin, đồng thời chuyển từ 32 bit thành 48 bit bằng cách sử dụng ma trận E table (hình 2.9). 32 bit thông tin theo thứ tự được đọc vào 48 vị trí (tương ứng với 6 cột và 8 dòng) của E table. Như vậy, sẽ có một số bit được lặp lại trong ma trận.

S-Box (Substitution Box) thực hiện thao tác thay thế chuỗi bit thành một chuỗi bit khác, đồng thời thực hiện thao tác ngược lại với E table là chuyển khối thông tin từ 48 bit thành 32 bit. S-Box cũng được thực hiện thông qua các ma trận S-Box (hình 2.10).

Nguyên tắc hoạt động của S-box như sau:

- 48 bit ngõ ra của phép XOR được chia thành 8 phần, mỗi phần 6 bit.
- Từng phần 6 bit được xử lý riêng biệt bằng một ma trận S-Box khác nhau (có 8 S-Box khác nhau).
- Tại mỗi S-Box, bit đầu và bit cuối của phần 6 bit thông tin được dùng để chọn 1 trong 4 hàng của ma trận, 4 bit còn lại được dùng để chọn 1 trong 16 giá trị của hàng tương ứng, giá trị được chọn sẽ chuyển thành 4 bit nhị phân.

Ví dụ, xét ma trận  $S_1$ , với chuỗi bit là 101100:

- bit đầu và bit cuối là 10, có giá trị thập phân là 2, do đó hàng được chọn là hàng số 2.
- 4 bit còn lại là 0110 nhị phân, giá trị thập phân tương ứng là 6, do đó giá trị tại cột 6 được chọn.
- Giá trị tại hàng 2 cột 6 trong ma trận  $S_1$  là 2, giá trị xuất ra là 0010.

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

**Hình 2.9:** Ma trận E table

Phép hoán vị P (Permuatation) có chức năng chuyển đổi vị trí các bit trong khối thông tin 32 bit xuất ra từ S-Box. Thao tác hoán vị P cũng được thực hiện dựa trên ma trận P gồm 8 cột và 4 dòng (hình 2.11).

$S_1$	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

$S_2$	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

$S_3$	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

$S_4$	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

$S_5$	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

$S_6$	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

$S_7$	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

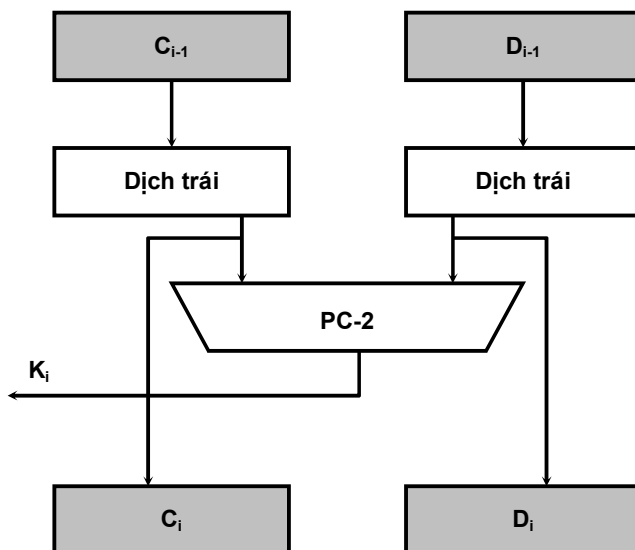
  

$S_8$	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Hình 2.10: Ma trận S-Box

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

**Hình 2.11:** Ma trận hoán vị  $P$



**Hình 2.12:** Thuật toán sinh khoá phụ của DES

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

**Hình 2.13:** Ma trận hoán vị  $PC-1$

**-Thuật toán sinh khoá phụ:** Khoá đưa vào cho thuật toán DES là 64 bit, tuy nhiên trong quá trình thực hiện, chỉ có 56 bit được sử dụng. Tất cả các bit cuối cùng của byte (tứ bit 8, 16, 24, 32, 40, 48, 56 và 64) bị loại bỏ ngay từ vòng xử lý đầu tiên.

Hình 1.12 mô tả thuật toán sinh khoá phụ của DES. 64 bit khoá ban đầu được chọn lấy 56 bit theo quy tắc đã nói ở trên, sau đó được đưa vào khối hoán vị  $PC-1$ . Mục đích của khối hoán vị

PC-1 là thay đổi vị trí các bit của 56 bit khoá vừa tạo ra. Chú ý rằng PC-1 chỉ được thực hiện 1 lần duy nhất trước khi bắt đầu vòng đầu tiên. Trong tất cả các vòng mã hoá, phép hoán vị thực hiện trên các khoá phụ là phép hoán vị PC-2. PC-1 và PC-2 được thực hiện thông qua các ma trận PC-1 và PC-2 ở hình 2.13 và 2.14. Ngõ ra của khối hoán vị PC-1 được chia thành 2 phần, mỗi phần 28 bit (C và D). Tại mỗi vòng mã hoá, hai phần này được dịch trái 1 hoặc 2 bit trước khi đi qua khối hoán vị PC-2 để thành 48 bit khoá phụ đưa vào hàm XOR cùng với khối thông tin của vòng tương ứng. Số bit dịch trái tương ứng với mỗi vòng như sau:

Vòng	1	2	3	4	5	6	7	8	9	0	11	12	13	14	15	16
Số bit dịch	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

**Hình 2.14:** Ma trận hoán vị PC-2

#### Nhận xét:

-Thuật toán mật mã DES là một thuật toán dựa trên cấu trúc Feistel nhưng có cách thực hiện phức tạp, được thiết kế dựa trên các thao tác xử lý bit (bitwise operations) như phép XOR, phép dịch, hoán vị, ... do đó thích hợp với các thiết bị mã hoá bằng phần cứng. Thuật toán DES không dễ phân tích, và trong một thời gian dài đã được giữ bí mật.

-Hai thông tin liên quan đến mức độ an toàn của thuật toán mã DES là tính phức tạp của giải thuật và chiều dài khoá. Đến thời điểm hiện nay, tức 30 năm kể từ khi DES được chấp nhận như một thuật toán mật mã tiêu chuẩn, chưa có một phát hiện nào về điểm yếu trong bản thân thuật toán. Tuy nhiên, với chiều dài khoá là 56 bit, việc dò khoá bằng phương pháp thử lần lượt là có thể thực hiện được với các máy tính đa dụng hiện nay với thời gian tìm kiếm khoảng 10 giờ.

Do vậy, nguy cơ tấn công mật mã đối với các hệ thống sử dụng DES là khá cao trong thời điểm hiện nay. Điều đó yêu cầu phải xây dựng một tiêu chuẩn mật mã khác hoặc cải tiến DES để tăng mức độ an toàn. Phần tiếp theo sẽ trình bày cả hai giải pháp này.

### **II.2.3 Thuật toán mật mã Triple DES:**

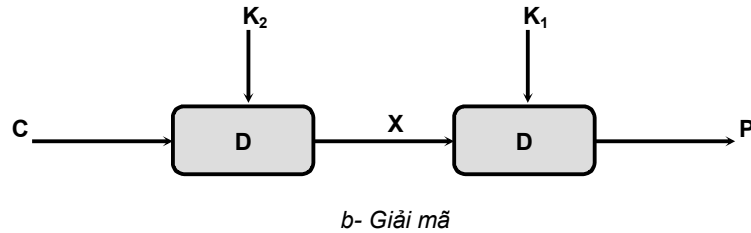
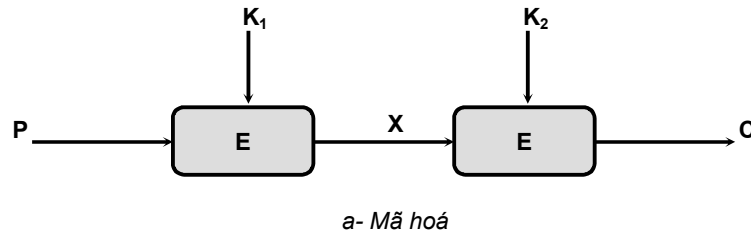
Tripple DES hay DES bội ba (viết tắt là 3DES hoặc TDES) là một phiên bản cải tiến của DES. Nguyên tắc của Triple DES là tăng chiều dài khoá của DES để tăng độ an toàn, nhưng vẫn giữ tính tương thích với thuật toán DES cũ.

Gọi P là thông tin gốc, K là khoá và C là thông tin đã mật mã hóa; E là thuật toán mã hóa và D là thuật toán giải mã, quá trình mã hóa và giải mã dùng thuật toán DES đơn giản được biểu diễn như sau:



$$C = E(P, K)$$

$$P = D(C, K)$$



**Hình 2.15:** DES bội hai (double DES)

Để tăng độ an toàn của giải thuật mật mã DES, ý tưởng cơ bản là thực hiện DES nhiều lần đối với cùng một khối thông tin gốc. Nếu thực hiện DES hai lần, ta có DES bội hai (double DES) (hình 2.15) với công thức biểu diễn như sau:

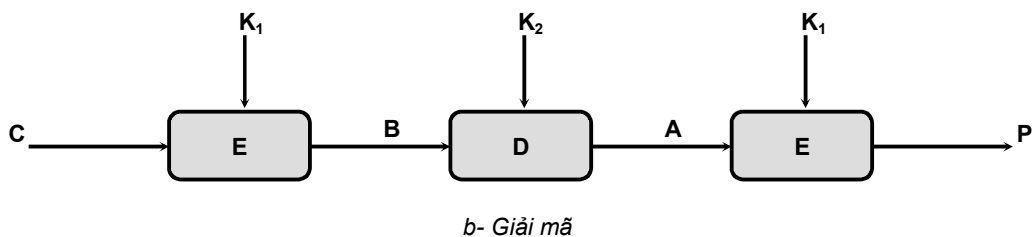
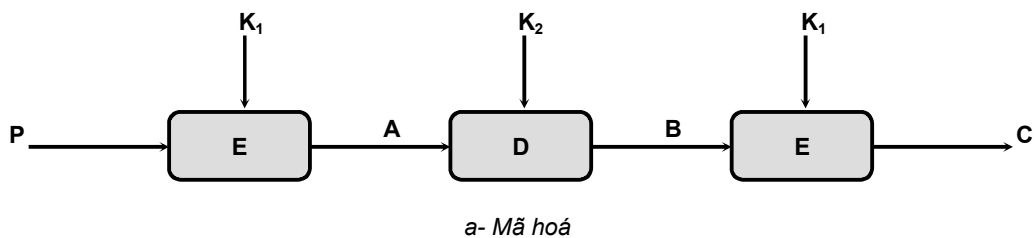
$$C = E(E(P, K_1), K_2)$$

$$P = D(D(C, K_2), K_1)$$

Tuy nhiên, với 112 bit khoá, DES bội hai vẫn chưa chứng tỏ được tính an toàn cao của nó, các hệ thống dùng DES bội hai vẫn có thể bị tấn công bằng phương thức xen giữa (Man-In-The-Middle). Bằng cách thực hiện DES ba lần trên cùng một khối thông tin, trong đó có hai lần mã hoá và một lần giải mã (hình 2.16), ta được Triple DES hay DES bội ba:

$$C = E(D(E(P, K_1), K_2), K_1)$$

$$P = D(E(D(C, K_1), K_2), K_1)$$



**Hình 2.16:** DES bội ba (triple DES) dùng 2 khoá

Khi đó, chiều dài khoá của thuật toán này vẫn là  $K_1 + K_2 = 112$  bit.

Việc xen vào một lần giải mã ở giữa trong thuật toán Triple DES không nhằm mục đích tăng thêm độ an toàn cho thuật toán mà chỉ giúp tạo ra sự tương thích giữa Triple DES và thuật toán DES cũ. Khi đó, thiết bị giải mã Triple DES có thể giải mã được thông tin mật được mã hoá bằng DES:

$$C = E(D(E(P, K_1), K_2), K_1) = E(P, K_1).$$

Triple DES với hai khoá là một thuật toán mật mã an toàn, tránh được các tấn công xen giữa và đã được sử dụng thay thế DES trong nhiều ứng dụng (ANS X9.17, ISO 8732, ...).

Một phiên bản khác của Triple DES là sử dụng cả 3 khoá khác nhau  $K_1, K_2, K_3$  với cùng cấu trúc như trên. Khi đó chiều dài khoá của thuật toán là  $K_1 + K_2 + K_3 = 168$  bit. Khi cần thiết phải đảm bảo tính tương thích với các ứng dụng DES cũ thì đặt  $K_1 = K_2$  hoặc  $K_3 = K_2$ . Triple DES 3 khoá cũng đã được ứng dụng trong nhiều dịch vụ, đặc biệt là PGP, S/MIME.

## II.2.4 Thuật toán mật mã AES:

Triple DES đã khắc phục được các điểm yếu của DES và hoạt động ổn định trong nhiều ứng dụng trên mạng Internet. Tuy nhiên, Triple DES vẫn còn chứa những nhược điểm của DES như tính khó phân tích, chỉ thích hợp với thực thi bằng phần cứng chứ không thích hợp cho thực thi bằng phần mềm, kích thước khối cố định 64 bit, ... Do đó, cần thiết phải xây dựng một chuẩn mật mã mới, dựa trên một cơ sở toán học vững chắc, có tính linh động để có thể điều chỉnh cho phù hợp với ứng dụng và đặc biệt là phải thích hợp với việc thực thi cả bằng phần mềm và phần cứng. Đó là những yêu cầu cơ bản đối với chuẩn mật mã cao cấp AES (Advanced Encryption Standard).

Thuật toán mật mã Rijndael được chọn để chuẩn hoá thành AES năm 2002. Hiện nay, AES vẫn còn trong giai đoạn thử nghiệm, các ứng dụng dựa trên AES chưa nhiều, nhưng trong thời gian ngắn sắp tới, các ứng dụng mật mã dùng khoá đối xứng sẽ chuyển dần sang AES.

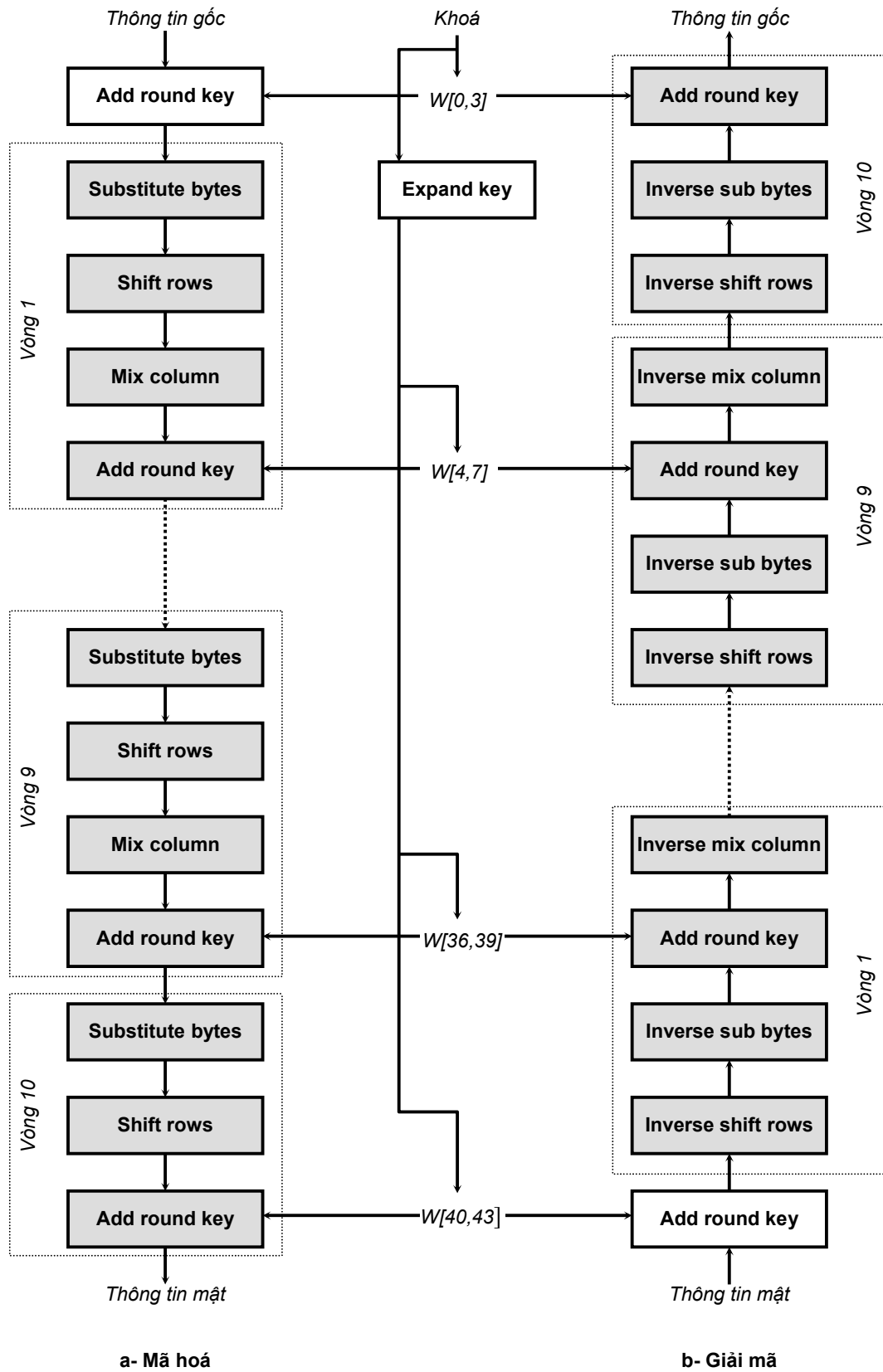
Các thông số chính của AES được tóm tắt như sau:

Chiều dài khoá (bit)	128	192	256
Kích thước khối (bit)	128	128	128
Số vòng mã (vòng)	10	12	14
Chiều dài khoá phụ (bit)	128	128	128
Chiều dài khoá mở rộng (byte)	176	208	240

Chiều dài khoá của AES có thể là 128, 192 hoặc 256 bit. Ứng với mỗi trường hợp, các thông số còn lại được cho tương ứng ở bảng trên, trong đó, kích thước khối thông tin luôn cố định là 128 bit.

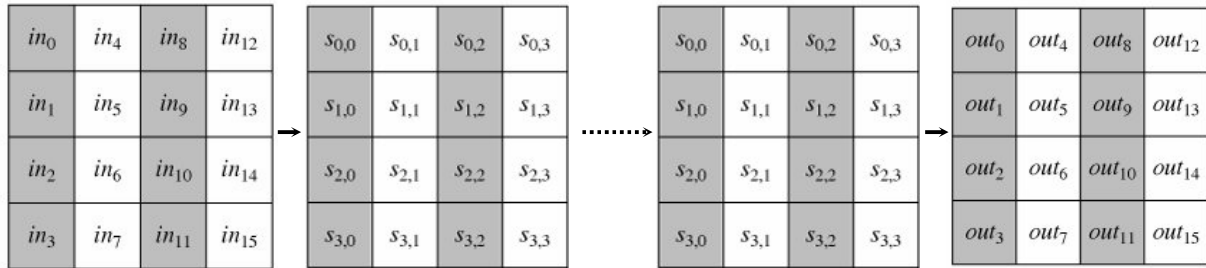
Một lưu ý quan trọng là AES không dựa trên cấu trúc Feistel. Tất cả các thao tác trong thuật toán đều có thể được mô tả bằng công cụ toán học, do đó AES có thể thực hiện bằng phần cứng hoặc phần mềm với tốc độ tối đa. AES sử dụng hai thuật toán khác nhau cho mã hoá và giải mã, do vậy *tất cả các thao tác trong thuật toán bắt buộc phải có thao tác ngược* (ngoại trừ phép XOR).

Hình 2.17 mô tả thuật toán AES trong trường hợp đơn giản nhất (128 bit khoá).



Hình 2.17: Thuật toán mã AES

Khối thông tin gốc (128 bit) được xử lý như một mảng 2 chiều kích thước 4 x 4 gọi là mảng trạng thái (State array), mỗi phần tử của mảng tương đương với 8 bit của khối thông tin. Mỗi vòng mã hoá sẽ làm thay đổi giá trị của mảng trạng thái, và ngõ ra của thuật toán mật mã chính là giá trị cuối cùng của mảng trạng thái (hình 2.18).



**Hình 2.18:** Quá trình biến đổi mảng trạng thái trong thuật toán AES

Thuật toán mã AES thực hiện dựa trên 4 thao tác sau đây:

- Thay thế byte (Byte Substitution)
- Dịch dòng (ShiftRows)
- Trộn cột (MixColumns)
- Cộng khoá (AddRoundKey)

Thuật toán mật mã AES dùng khoá 128 bit (cả mã hoá và giải mã) bắt đầu bằng một thao tác cộng khoá, sau đó là 9 vòng liên tiếp, mỗi vòng gồm đủ 4 bước như trên, và một vòng cuối cùng gồm 3 bước (không có thao tác trộn cột).

**-Thao tác thay thế byte:** thao tác này có chức năng thay thế từng byte trong mảng trạng thái thành một byte khác sử dụng một ma trận kích thước 16 x 16 (được gọi là S-Box). Nguyên tắc thay thế dùng ma trận S-Box như sau: ứng với mỗi byte trong mảng trạng thái hiện hành, 4 bit bên trái được dùng để chọn một trong 16 dòng, 4 bit bên phải được dùng để chọn một trong 16 cột. Giá trị của ô tương ứng với dòng và cột được chọn sẽ là giá trị thay thế cho byte hiện hành. Ở quá trình giải mã, thao tác này cũng được thực hiện tương tự nhưng sử dụng một ma trận khác, gọi là ma trận S-Box ngược (hình 2.19).

Ví dụ: mảng trạng thái hiện hành có giá trị (Hex) như sau:

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

Sau khi qua thao tác thay thế byte sử dụng ma trận S-Box ở hình 2.19 sẽ trở thành:

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

a- Ma trận S-Box

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

b- Ma trận S-Box ngược

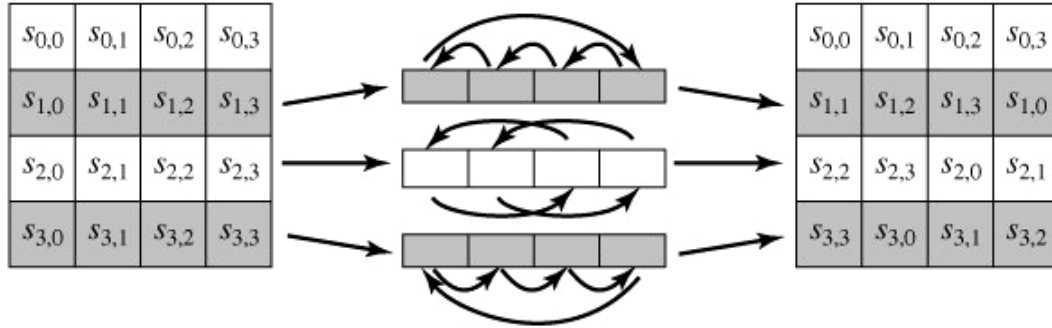
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Hình 2.19: Ma trận thay thế byte (S-Box)

**-Thao tác dịch dòng:** Thao tác này có mục đích hoán vị các byte trong mảng trạng thái. Nguyên tắc dịch như sau: dòng đầu tiên của mảng được giữ nguyên, dòng thứ hai được dịch trái 1 byte, dòng thứ ba được dịch trái 2 byte và dòng thứ tư được dịch trái 3 byte (hình 2.20).

Thao tác ngược được thực tương tự nhưng với phép dịch phải được dùng thay cho phép dịch trái, nghĩa là dòng đầu tiên cũng được giữ nguyên, dòng thứ hai được dịch phải 1 byte, dòng thứ ba được dịch phải 2 byte và dòng thứ tư được dịch phải 3 byte.

**-Thao tác trộn cột:** Thao tác này được thực hiện trên từng cột, có tác dụng thay thế từng



**Hình 2.20:** Thao tác dịch dòng

byte trong cột bằng một giá trị được tạo ra từ giá trị của tất cả các byte trong cùng cột đó. Thao tác này được biểu diễn bằng phép nhân ma trận như sau:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Với phép nhân này, ta có:

$$s'_{0,j} = 2s_{0,j} \oplus 3s_{1,j} \oplus s_{2,j} \oplus s_{3,j}$$

$$s'_{1,j} = s_{0,j} \oplus 2s_{1,j} \oplus 3s_{2,j} \oplus s_{3,j}$$

$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus 2s_{2,j} \oplus 3s_{3,j}$$

$$s'_{3,j} = 3s_{0,j} \oplus s_{1,j} \oplus s_{2,j} \oplus 2s_{3,j}$$

Phép nhân ma trận được thực hiện trong trường  $GF(2^8)$  (\*).

Thao tác ngược của thao tác trộn cột được thực hiện tương tự nhưng với phép nhân ma trận sau:

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Khi đó, giá trị của mạng trạng thái được xác định như sau:

$$s'_{0,j} = 14s_{0,j} \oplus 11s_{1,j} \oplus 13s_{2,j} \oplus 9s_{3,j}$$

$$s'_{1,j} = 9s_{0,j} \oplus 14s_{1,j} \oplus 11s_{2,j} \oplus 13s_{3,j}$$

$$s'_{2,j} = 13s_{0,j} \oplus 9s_{1,j} \oplus 14s_{2,j} \oplus 11s_{3,j}$$

$$s'_{3,j} = 11s_{0,j} \oplus 13s_{1,j} \oplus 9s_{2,j} \oplus 14s_{3,j}$$

Ví dụ: mạng trạng thái hiện hành có giá trị (Hex) như sau:

(\*) Xem thêm tài liệu về các phép toán trong trường Galois, đặc biệt là dạng  $GF(2^n)$

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

Sau khi qua thao tác trộn cột bằng phép nhân ma trận ở trên sẽ trở thành:

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

**-Thao tác cộng khoá:** là thao tác đơn giản nhất của thuật toán, có tác dụng trộn giá trị của mảng trạng thái hiện hành với khoá phụ của vòng tương ứng. Thao tác trộn được thực hiện bằng phép XOR giữa 128 bit của mảng trạng thái hiện hành với 128 bit của khoá phụ. Thao tác cộng khoá không có thao tác ngược, hay nói đúng hơn là thao tác ngược cũng chính là phép XOR.

**-Thuật toán sinh khoá phụ:** 128 bit khoá ban đầu được mở rộng (expand key) thành 176 byte, được tổ chức thành 44 từ (word), mỗi từ 4 byte, vừa đủ để tạo thành 10 khoá phụ cho 10 vòng mã hoá của thuật toán (mỗi khoá phụ gồm 4 từ) cộng với 1 khoá phụ cho thao tác cộng khoá ban đầu. Như vậy, thuật toán sinh khoá phụ của AES thực chất là thuật toán mở rộng bốn từ khoá (128 bit) ban đầu thành 44 từ khoá. Thao tác mở rộng khoá được thực hiện như sau:

-Bốn từ khoá gốc được đưa trực tiếp vào phép cộng khoá ban đầu, tức  $w[0,3] = \text{key}$ .

-Các từ khoá mở rộng tiếp theo (có thứ tự không là bội số của 4) được tạo ra bằng cách XOR giữa từ khoá liền trước nó với từ khoá cách nó 4 vị trí, tức  $w[i] = w[i-1] \oplus w[i-4]$ .

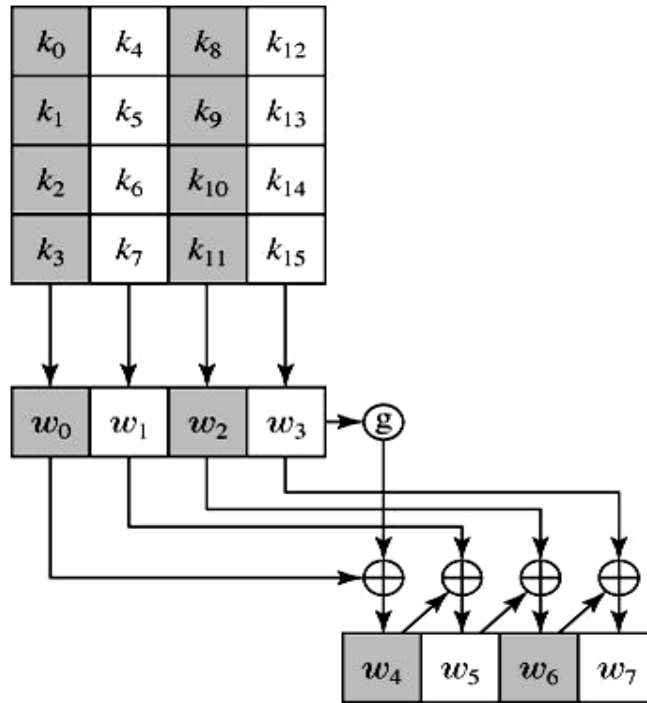
-Đối với các từ khoá mở rộng có thứ tự là bội số của 4 thì cách tạo ra gồm các bước:

- ✓ Thực hiện dịch từ khoá liền trước nó sang trái 1 byte,  $\text{temp} = \text{leftshift}(w[i-1], 8 \text{ bit})$
- ✓ Thay thế các byte trong từ khoá vừa tạo ra bằng các giá trị khác sử dụng ma trận S-Box ở hình 2.19,  $\text{temp} = \text{S-Box}(\text{temp})$
- ✓ Giá trị tạo ra được XOR với một hằng số xác định cho từng vòng mã hoá gọi là Round Constant hay  $\text{RC}[j]$ . Giá trị  $\text{RC}[j]$  được định nghĩa riêng biệt cho từng vòng như sau:

Vòng	1	2	3	4	5	6	7	8	9	10
$\text{RC}[j]$	01	02	04	08	10	20	40	80	1B	36

- ✓ Giá trị sau khi XOR với  $\text{RC}[j]$  được XOR một lần nữa với từ khoá cách từ khoá hiện hành 4 vị trí để tạo thành từ khoá mới.

Hình 2.21 trình bày thuật toán mở rộng khoá của AES, trong đó hàm  $g$  biểu diễn một phép toán phức tạp gồm 4 thao tác vừa trình bày, áp dụng cho các từ khoá có vị trí là bội số của 4.



Hình 2.21: Thuật toán mở rộng khoá của AES

## II.2.5 Các thuật toán mật mã đối xứng khác:

Ngoài 2 thuật toán mật mã hóa tiêu chuẩn ở trên (Triple DES được xem như là một phiên bản nâng cấp của DES chứ không phải một thuật toán độc lập), có nhiều thuật toán khác cũng đã chứng minh được tính hiệu quả của nó và được sử dụng trong một số ứng dụng khác nhau:

**-IDEA (International Data Encryption Algorithm)** là một thuật toán mật mã đối xứng được phát triển ở Thụy điển năm 1991. IDEA dựa trên cấu trúc Feistel, sử dụng khóa 128 bit và có nhiều điểm khác biệt so với DES. IDEA không sử dụng S-box mà dựa vào 3 phép toán là XOR, phép cộng nhị phân và phép nhân nhị phân trên các thanh ghi 16 bit. IDEA sử dụng thuật toán mã hóa gồm 8 vòng, khóa phụ tại mỗi vòng được sinh ra từ các phép dịch phức tạp. IDEA được sử dụng trong các ứng dụng bảo mật thư điện tử (PGP).

**-Blowfish** được phát triển năm 1993, bởi một người nghiên cứu mật mã hóa độc lập (Bruce Schneier) và cũng đã nhanh chóng được sử dụng song song với giải thuật mã hóa DES. Blowfish được thiết kế đơn giản và tốc độ thực thi nhanh. Giải thuật này sử dụng khóa có chiều dài thay đổi (có thể lên đến 448 bit) nhưng thường sử dụng nhất là khóa 128 bit. Blowfish cũng dùng cấu trúc mã khối Feistel, thực hiện 16 vòng mã, sử dụng các phép toán S-box, XOR và phép cộng nhị phân.

**-RC4 và RC5** là giải thuật mã hóa đối xứng được thiết kế bởi Ron Rivest (một trong những người phát minh ra giải thuật mã hóa bất đối xứng RSA) vào năm 1988 và 1994. RC4 là một thuật toán mã dòng (Stream cipher), có cấu trúc đơn giản, được ứng dụng trong bảo mật Web (SSL/TSL) và trong mạng không dây (WEP). RC5 là thuật toán mã khối, được thiết kế với các đặc tính như: phù hợp với việc thực thi bằng cả phần cứng và phần mềm, tốc độ cao, đơn giản, dùng khóa có chiều dài thay đổi và số vòng mã hóa cũng có thể thay đổi.



**-CAST-128** là một thuật toán khác được thiết kế năm 1997 bởi Carlisle Adams và Stafford Tavares. CAST-128 dùng khóa có độ dài thay đổi, cũng sử dụng S-box nhưng với kích thước lớn hơn so với DES, và điều đặc biệt là các vòng mã hóa không hoàn toàn giống nhau.

Bảng 2.1 tóm tắt các thuật toán mật mã khối dùng khoá đối xứng hiện có trong thực tế và các ứng dụng của chúng.

**Bảng 2.1:** Các thuật toán mật mã đối xứng

Thuật toán	Chiều dài khóa	Số vòng mã hóa	Phép toán sử dụng	Ứng dụng
DES	56 bit	16	XOR, S-box	SET, Kerberos
3DES	112 hoặc 168 bit	48	XOR, S-box	PGP, S/MIME, các ứng dụng quản lý khóa
AES	128, 192 hoặc 256 bit	10, 12 hoặc 14	XOR, dịch, S-box	SSL
IDEA	128 bit	8	XOR, cộng nhị phân, nhân nhị phân	PGP
Blowfish	Thay đổi, tối đa 448 bit	16	XOR, S-box, cộng nhị phân	Các công cụ mật mã.
RC5	Thay đổi, tối đa 2048 bit	Thay đổi, tối đa 255 vòng	Cộng nhị phân, trừ nhị phân, XOR, phép quay	Các công cụ mật mã.
CAST-128	40 đến 128 bit	16	Cộng, trừ nhị phân, XOR, quay, S-box	PGP

## II.3 KỸ THUẬT MẬT MÃ BẮT ĐỐI XỨNG

### II.3.1 Cấu trúc hệ thống mật mã bắt đối xứng:

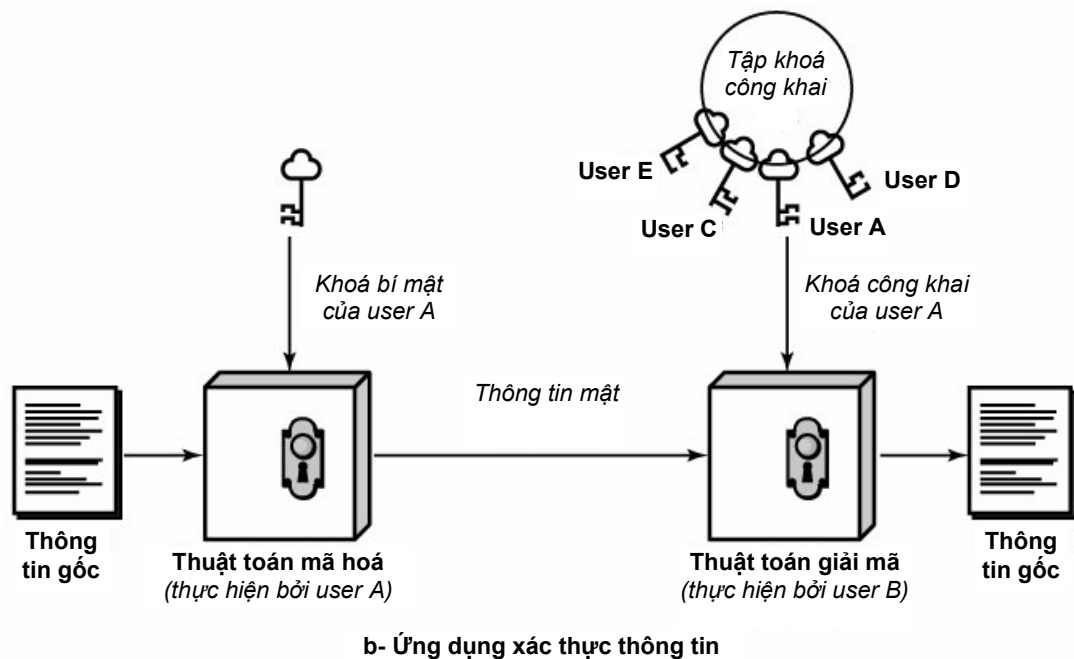
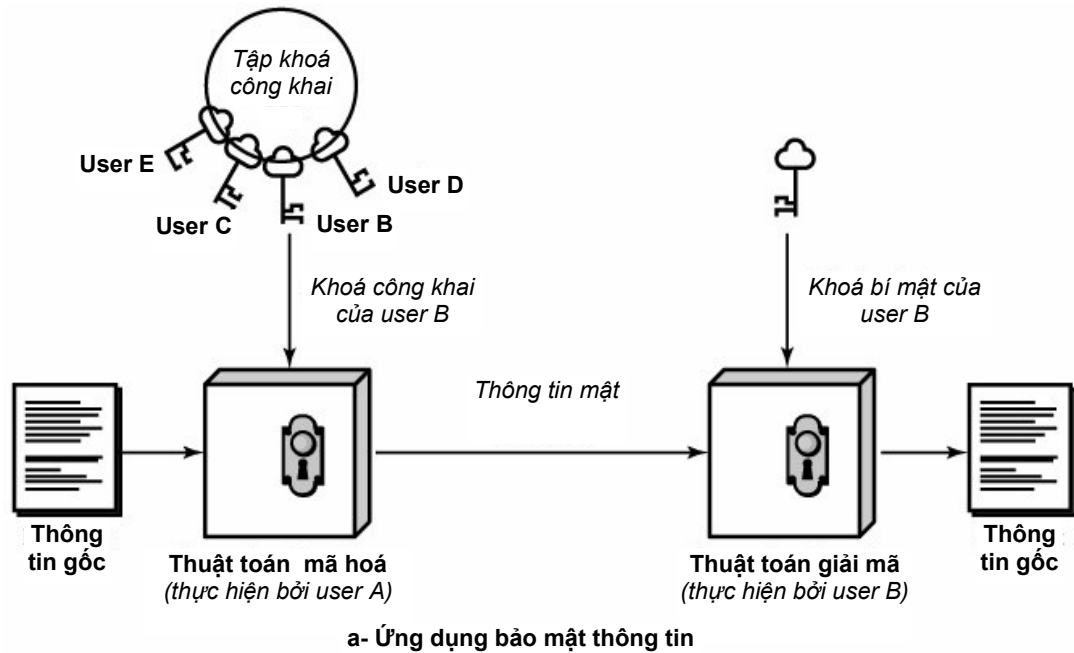
Đặc trưng của kỹ thuật mật mã bắt đối xứng là dùng 2 khóa riêng biệt cho hai quá trình mã hóa và giải mã, trong đó có một khóa được phổ biến công khai (public key hay PU) và khóa còn lại được giữ bí mật (private key hay PR). Cả hai khóa đều có thể được dùng để mã hoá hoặc giải mã. Việc chọn khoá công khai hay khoá bí mật cho quá trình mã hoá sẽ tạo ra hai ứng dụng khác nhau của kỹ thuật mật mã bắt đối xứng:

- Nếu dùng khoá công khai để mã hoá và khoá bí mật để giải mã, ta có ứng dụng bảo mật trên thông tin (confidentiality).
- Nếu dùng khoá bí mật để mã hoá và khoá công khai để giải mã, ta có ứng dụng xác thực nội dung và nguồn gốc thông tin (authentication).

Thuật toán mật mã bắt đối xứng dựa chủ yếu trên các hàm toán học hơn là dựa vào các thao tác trên chuỗi bit. Mật mã hóa bắt đối xứng còn được gọi bằng một tên thông dụng hơn là mật mã hóa dùng khóa công khai (public key encryption).

Nói chung, mật mã hóa bắt đối xứng không phải là một kỹ thuật mật mã an toàn hơn so với mật mã đối xứng, mà độ an toàn của một thuật toán mã nói chung phụ thuộc vào 2 yếu tố: Độ dài của khóa và mức độ phức tạp khi thực hiện thuật toán (trên máy tính). Hơn nữa, mặc dù được

ra đời sau nhưng không có nghĩa rằng mật mã bất đối xứng hoàn toàn ưu việt hơn và sẽ được sử dụng thay thế cho mật mã đối xứng. Mỗi kỹ thuật mã có một thế mạnh riêng và mật mã đối xứng vẫn rất thích hợp cho các hệ thống nhỏ và đơn giản. Ngoài ra, *vấn đề phân phối khóa* trong mật mã bất đối xứng cũng được đánh giá là một trong những vấn đề phức tạp khi triển khai kỹ thuật mật mã này trong thực tế.



**Hình 2.22:** Cấu trúc hệ thống mật mã bất đối xứng

Cấu trúc một hệ thống mật mã bất đối xứng được trình bày trong hình 2.22. Các bước cơ bản của một hệ thống mật mã dùng khóa công khai bao gồm:

- Mỗi thực thể thông tin (user) tạo ra một cặp khóa (public/private) để dùng cho việc mã hóa và giải mã.
- Mỗi user thông báo một trong hai khóa của mình cho các user khác biết, khóa này được gọi là khóa công khai (public key). Khóa còn lại được giữ bí mật, và gọi là khóa riêng (private key).
- Nếu một user A muốn gửi thông tin cho user B, user A sẽ thực hiện mã hóa thông tin cần gửi bằng khóa công khai của user B.
- Khi nhận được thông tin đã mã hóa từ user A, user B thực hiện giải mã thông tin đó bằng khóa riêng của mình. Do khóa riêng không phổ biến công khai nên chỉ có một mình user B có khả năng giải mã được.

Mật mã hóa bất đối xứng được sử dụng trong các ứng dụng: che giấu thông tin, tạo chữ ký số (digital signature) và trao đổi khóa trong các thuật toán mật mã đối xứng (key exchange).

### II.3.2 Thuật toán mật mã RSA:

RSA là thuật toán mật mã bất đối xứng được xây dựng bởi Ron Rivest, Adi Shamir và Len Adleman tại viện công nghệ Massachusetts (MIT), do đó được đặt tên là Rivest – Shamir – Adleman hay RSA. Thuật toán này ra đời năm 1977 và cho đến nay đã được ứng dụng trong nhiều lĩnh vực. Cũng như các thuật toán mật mã bất đối xứng khác, nguyên lý của RSA dựa chủ yếu trên lý thuyết số chứ không dựa trên các thao tác xử lý bit.

Trong phạm vi tài liệu này, thuật toán mã RSA được mô tả khái quát giúp người đọc nắm được nguyên lý của thuật toán mã chứ không chú trọng đến vấn đề phân tích và chứng minh các cơ sở lý thuyết của thuật toán.

RSA là một thuật toán mật mã khối, kích thước khối thông thường là 1024 hoặc 2048 bit. Thông tin gốc của RSA được xử lý như các số nguyên. Ví dụ, khi chọn kích thước khối của thuật toán là 1024 bit thì số nguyên này có giá trị từ 0 đến  $2^{1024} - 1$ , tương đương với số thập phân có 309 chữ số. Chú ý rằng đây là những số nguyên cực lớn, không thể xử lý được bằng cách sử dụng các cấu trúc dữ liệu có sẵn của các ngôn ngữ lập trình phổ biến.

Thuật toán RSA được mô tả như sau:

1-Để tạo ra một cặp khóa RSA, trước hết, chọn hai số nguyên tố đủ lớn  $p$  và  $q$ . Gọi  $N$  là tích của  $p$  và  $q$  ( $N = pq$ ).

2-Tiếp theo, chọn một số  $e$  sao cho  $e$  và  $(p-1)(q-1)$  là hai số nguyên tố cùng nhau. Sau đó tìm số  $d$  sao cho  $ed = 1 \bmod (p-1)(q-1)$ . Ký hiệu  $\bmod m$  biểu diễn phép modulo trên cơ số  $m$ .

3-Bây giờ, bỏ qua vai trò của  $p$  và  $q$ . Với 3 thành phần còn lại là  $N$ ,  $e$  và  $d$ , ta có:

-Khóa công khai (public key) là tổ hợp  $(N, e)$

-Khóa bí mật (private) là tổ hợp  $(N, d)$ .

4-Việc mã hóa một khối thông tin gốc  $M$  được thực hiện theo công thức:

$$C = M^e \bmod N \quad (\text{với } M \text{ là số nguyên nhỏ hơn } N)$$

5-Và quá trình giải mã  $C$  được thực hiện theo công thức:

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

Cơ sở lý thuyết của thuật toán RSA dựa trên lý thuyết về số nguyên tố, phép toán modulo và định lý Euler như sau:

**Hàm Euler:** Cho một số nguyên dương  $n$ , định nghĩa  $\phi(n)$  là số các số nguyên dương nhỏ hơn  $n$  và là số nguyên tố cùng nhau với  $n$ . Ví dụ: cho  $n = 8$ , các số nguyên dương nhỏ hơn 8 và là số nguyên tố cùng nhau với 8 là các số 1, 3, 5, 7, do đó  $\phi(8) = 4$ .  $\phi(n)$  được gọi là hàm Euler của  $n$ .

-Quy ước  $\phi(1) = 1$ .

-Nếu  $n$  là số nguyên tố thì tất cả các số nguyên dương nhỏ hơn  $n$  đều là số nguyên tố cùng nhau với  $n$ , khi đó  $\phi(n) = n - 1$ .

-Nếu  $p$  và  $q$  là hai số nguyên tố và  $N = pq$ . Khi đó  $\phi(N) = \phi(p) \cdot \phi(q)$ . Thật vậy, trong  $N-1$  hay  $(pq-1)$  số nguyên dương nhỏ hơn  $N$ : các số  $p, 2p, \dots, (q-1)p$  và các số  $q, 2q, \dots, (p-1)q$  là các số không phải nguyên tố cùng nhau với  $N$ . Như vậy:

$$\begin{aligned}\phi(N) &= (pq - 1) - [(p - 1) + (q - 1)] \\ &= pq - (p + q) + 1 \\ &= (p - 1)(q - 1) \\ &= \phi(p) \cdot \phi(q)\end{aligned}$$

**Định lý Euler:** cho  $a$  và  $n$  là hai số nguyên tố cùng nhau, ta có  $a^{\phi(n)} = 1 \pmod n$

Ta chấp nhận định lý này mà không phải chứng minh.

Với những cơ sở này, ta có thể kiểm chứng thuật toán RSA như sau:

Cho trước khối thông tin mật  $C = M^e \pmod N$ , cần kiểm chứng rằng  $M = C^d \pmod N$ .

Ta có:

$$C^d \pmod N = (M^e)^d \pmod N = M^{ed} \pmod N$$

Xét quá trình tạo cặp khoá của RSA, ta có:

$$ed = 1 \pmod{(p-1)(q-1)}$$

Hơn nữa,  $N = pq$  nên  $\phi(N) = (p-1)(q-1)$  với  $p, q$  là các số nguyên tố.

Như vậy:

$$ed - 1 = k \phi(N) \text{ với một số nguyên } k \text{ nào đó.}$$

Và:

$$\begin{aligned}C^d \pmod N &= M^{ed} \pmod N \\ &= M^{(ed-1)+1} \pmod N \\ &= M \cdot M^{ed-1} \pmod N \\ &= M \cdot M^{k\phi(N)} \pmod N \\ &= M \cdot 1^k \pmod N \\ &= M.\end{aligned}$$

**Ví dụ:** Cặp số nguyên tố  $p = 11$  và  $q = 3$  được chọn để tạo ra cặp khoá RSA cho user A.

Khi đó,  $N = pq = 3 \cdot 11 = 33$

$$(p-1)(q-1) = (11-1)(3-1) = 20$$

Tiếp theo, chọn  $e = 3$  thoả điều kiện 3 và 20 là cặp số nguyên tố cùng nhau.

Với  $e = 3$ , ta xác định được  $d = 7$  vì  $ed = 3 \cdot 7 = 1 \pmod{20}$ . Thật ra, có nhiều giá trị  $d$  thoả mãn yêu cầu này, nhưng để cho đơn giản, ta chọn giá trị nhỏ nhất.

Khi đó, ta xác định được cặp khoá như sau:

Khóa công khai:  $(N, e) = (33, 3)$

Khóa bí mật:  $(N, d) = (33, 7)$

Giả sử, user B muốn gửi đoạn thông tin  $M = 15$  cho user A, dựa trên khóa công khai của A, B thực hiện như sau:

$$C = M^e \pmod N = 15^3 \pmod{33} = 3375 \pmod{33} = 9 \pmod{33}.$$

Khi đó, thông tin mật gửi cho A là  $C = 9$ .

Khi nhận được thông tin này, A giải mã bằng khóa riêng của mình ( $d = 7$ ) như sau:

$$M = C^d \bmod N = 9^7 \bmod 33 = 4.782.969 \bmod 33 = 15 \bmod 33.$$

Như vậy, thông tin giải mã được là  $M = 15$ , đúng với thông tin gốc ban đầu.

Tóm lại, thuật toán mật mã RSA được thực hiện gồm 3 quá trình tách rời: tạo khoá, mã hoá và giải mã được tóm tắt như sau:

**1-Tạo khoá:**

- **Chọn  $p, q$**  ( $p$  và  $q$  là số nguyên tố,  $p \neq q$ )
- **Tính  $N = p.q$**
- **Tính  $\phi(N) = (p - 1)(q - 1)$**
- **Chọn  $e$  sao ước số chung lớn nhất của  $e$  và  $\phi(N)$  là 1**
- **Chọn  $d$  sao cho  $e.d \bmod \phi(N) = 1$**
- **Cặp khoá RSA được tạo ra là  $PU = (N, e)$ ,  $PR = (N, d)$**

**2- Mã hoá:**

- **$C = M^e \bmod N$**  ( $M$  là số nguyên nhỏ hơn  $N$ )

**3- Giải mã:**

- **$M = C^d \bmod N$**

Trong thực tế, để đạt được độ an toàn cao, cặp khoá phải được chọn trên các số  $p$  và  $q$  đủ lớn ( $N$  nhỏ nhất phải là 1024 bit), do vậy, vấn đề thực thi RSA bao gồm các phép toán lũy thừa trên các số rất lớn. Vấn đề giảm chi phí tính toán và tăng tốc độ thực hiện thuật toán RSA là một trong những vấn đề quan trọng cần phải giải quyết. Trên các hệ thống máy tính hiện nay, hiệu suất thực hiện giải thuật RSA là chấp nhận được.

**-Độ an toàn của RSA:**

Theo lý thuyết, hệ thống RSA có thể bị tấn công bằng những phương thức sau đây:

- Brute-force attack: tìm lần lượt khoá riêng PR
- Mathematical attack: xác định  $p$  và  $q$  bằng cách phân tích  $N$  thành tích của các thừa số nguyên tố rồi từ đó xác định  $e$  và  $d$ .
- Timing attack: dựa trên thời gian thực thi của thuật toán giải mã.
- Chosen ciphertext attack: sử dụng các đoạn thông tin mật (ciphertext) đặc biệt để khôi phục thông tin gốc.

Tuy nhiên trong thực tế, nguy cơ tấn công các hệ thống mật mã RSA là rất thấp, do RSA là một thuật toán linh động, kích thước khối dữ liệu gốc và chiều dài khoá dễ dàng được thay đổi mà không ảnh hưởng đến thuật toán mã.

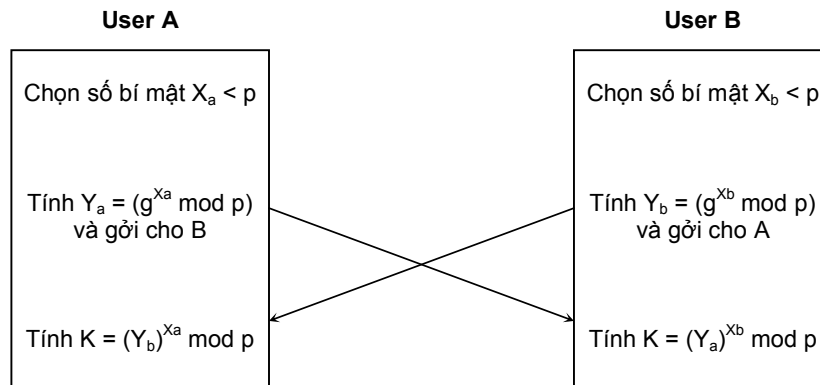
### II.3.3 Thuật toán trao đổi khoá Diffie-Hellman:

Diffie-Hellman là một thuật toán dùng để trao đổi khoá (key exchange) chứ không dùng để mật mã hóa (che giấu) dữ liệu. Tuy nhiên, Diffie-Hellman lại có ích trong giai đoạn trao đổi khoá bí mật của các thuật toán mật mã đối xứng. Như trong phần đầu của chương này đã trình

bây, một trong những vấn đề quan trọng liên quan trực tiếp đến tính an toàn của các thuật toán mật mã đối xứng là vấn đề thống nhất khoá bí mật giữa các thực thể thông tin.

Thuật toán trao đổi khoá Diffie-Hellman dựa trên phép logarit rời rạc (discrete log). Cho trước một số  $g$  và  $x = g^k$ , để tìm  $k$ , ta đơn giản thực hiện phép logarit:  $k = \log_g(x)$ . Tuy nhiên, nếu cho trước  $g$ ,  $p$  và  $(g^k \bmod p)$ , thì quá trình xác định  $k$  được thực hiện theo cách khác với cách ở trên và được gọi là logarit rời rạc. Việc tính logarit rời rạc nói chung rất phức tạp nhưng vẫn có thể thực hiện được.

Thuật toán Diffie-Hellman khá đơn giản như sau:



**Hình 2.23:** Thuật toán trao đổi khoá Diffie-Hellman

-Gọi  $p$  là một số nguyên tố và  $g$  là một cơ sở sinh (generator) thoả điều kiện với mọi  $x \in \{1, 2, \dots, p-1\}$ , ta luôn tìm được số  $n$  sao cho  $x = g^n \bmod p$ .

-Giá trị  $p$  và  $g$  được phổ biến công khai giữa các thực thể trao đổi khoá. Sau đó user A tạo ra một số bí mật  $X_a < p$ , tính giá trị  $Y_a = (g^{X_a} \bmod p)$  và gửi cho B. Tương tự, user B cũng tạo ra một số bí mật  $X_b < p$ , tính giá trị  $Y_b = (g^{X_b} \bmod p)$  và gửi lại cho A.

-Dựa trên thông tin nhận được từ A, user B xác định được khoá bí mật dùng cho phiên làm việc bằng cách tính giá trị  $(g^{X_a} \bmod p)^{X_b} = (g^{X_a X_b} \bmod p)$ . Bằng cách tương tự, user A cũng xác định được khoá bí mật này bằng cách tính giá trị  $(g^{X_b} \bmod p)^{X_a} = (g^{X_a X_b} \bmod p)$ .

-Giả sử trong quá trình trao đổi các giá trị  $(g^{X_a} \bmod p)$  và  $(g^{X_b} \bmod p)$ , một người thứ 3 nào đó bắt được thông tin này thì cũng rất khó xác định được  $a$  và  $b$  vì độ phức tạp của phép toán logarit rời rạc là rất cao.

Ví dụ:

Cho  $p = 353$  và  $g = 3$ . Có thể kiểm chứng được rằng với một số nguyên  $n$  bất kỳ sao cho  $0 < n < 353$ , ta luôn xác định được một số nguyên  $i$  thoả  $3^i = n$ .

Giả sử, user A chọn giá trị bí mật  $X_a = 97$  và user B chọn giá trị bí mật  $X_b = 233$ .

User A tính được  $Y_a = (3^{97} \bmod 353) = 40$  và gửi cho B.

User B tính được  $Y_b = (3^{233} \bmod 353) = 248$  và gửi cho A.

User A tính được khoá bí mật  $K = (Y_b)^{X_a} \bmod 353 = 248^{97} \bmod 353 = 160$

User B tính được khoá bí mật  $K = (Y_a)^{X_b} \bmod 353 = 40^{233} \bmod 353 = 160$

**-Mức độ an toàn của thuật toán trao đổi khoá Diffie-Hellman:**

Tính an toàn của Diffie-Hellman dựa trên độ phức tạp của phép toán logarit rời rạc. Nói chung, việc xác định các giá trị  $X_a$ ,  $X_b$  từ các giá trị  $p$ ,  $g$ ,  $Y_a$  và  $Y_b$  là không thể thực hiện được

trên các số nguyên đủ lớn. Tuy nhiên, thuật toán này không ngăn chặn được các tấn công theo phương thức xen giữa Man-In-The-Middle (MITM) như sau:

- Để thực hiện tấn công MITM trên kết nối giữa user A và user B, user C cũng chọn cho mình hai số nguyên  $X_{C1}$  và  $X_{C2}$  thỏa điều kiện  $X_{C1} < p$  và  $X_{C2} < p$ , sau đó cũng tính hai giá trị tương ứng  $Y_{C1} = (g^{X_{C1}} \bmod p)$  và  $Y_{C2} = (g^{X_{C2}} \bmod p)$ .
- Khi user A gửi  $Y_a$  cho user B, user C sẽ chặn lấy thông tin này, đồng thời mạo danh A để gửi cho B giá trị  $Y_{C1}$ . User B xác định khoá  $K_1$  dựa trên  $Y_{C1}$ , và gửi lại cho A giá trị  $Y_b$ . User C lại chặn lấy giá trị này và mạo danh B để gửi cho A giá trị  $Y_{C2}$ .
- User A xác định khoá  $K_2$  dựa trên  $Y_{C2}$ . Bắt đầu từ đây, các thông tin trao đổi giữa A và B đều được C chặn bắt và thay đổi bằng cách sử dụng cặp khoá  $K_1$  và  $K_2$ .

*Thuật toán Diffie-Hellman không giải quyết được vấn đề này do không có cơ chế xác thực giữa các thực thể trao đổi khoá. Điểm yếu này được khắc phục bằng cách sử dụng kết hợp với các thuật toán xác thực như sẽ trình bày ở phần kế tiếp.*

Ngoài hai thuật toán RSA và Diffie-Hellman, một số thuật toán khác cũng được phát triển dựa trên nguyên lý sử dụng một cặp khoá công khai và bí mật. Elliptic-Curve Cryptography (ECC) là một giải thuật mới đang được thử nghiệm và hứa hẹn nhiều ưu điểm so với RSA như độ phức tạp tính toán giảm trong khi tính an toàn vẫn được đảm bảo. ECC thích hợp với các ứng dụng chạy trên các thiết bị có năng lực xử lý hạn chế như các thiết bị nhúng (embedded devices).

### II.3.4 Đánh giá kỹ thuật mật mã bất đối xứng:

Kỹ thuật mật mã bất đối xứng hoàn toàn có thể đáp ứng được những yêu cầu về bảo mật hệ thống như trong kỹ thuật mật mã đối xứng, mặc dù tốc độ thực thi của mã bất đối xứng thường thấp hơn do bản chất thuật toán dựa trên các thao tác số học chứ không dựa trên các thao tác xử lý bit. Hơn nữa, mã bất đối xứng chỉ phù hợp với việc thực thi bằng phần mềm. *Mật mã bất đối xứng đảm bảo được 2 yêu cầu cơ bản của thông tin là tính bí mật và tính toàn vẹn.*

Kỹ thuật mật mã bất đối xứng có 2 ưu điểm so với mã đối xứng:

1-Hai thực thể thông tin không cần thực hiện thủ tục trao đổi khóa trước khi bắt đầu làm việc.

2-Bên cạnh công dụng đảm bảo tính toàn vẹn của dữ liệu, mật mã bất đối xứng (khi được sử dụng cho mục đích xác thực) còn đảm bảo được tính không thể phủ nhận (non-repudiation) của thông tin.

## II.4 CÁC HÀM BĂM

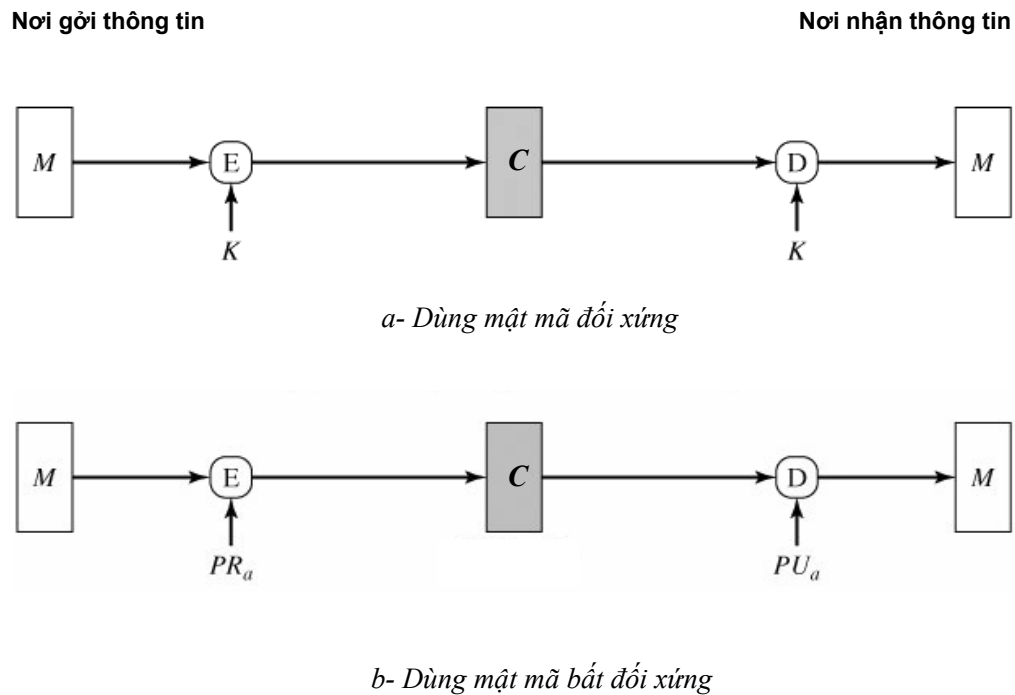
### II.4.1 Xác thực thông tin:

Xác thực thông tin (message authentication) là một cơ chế được ứng dụng trong xử lý thông tin với mục đích:

- Đảm bảo nội dung thông tin trao đổi giữa các thực thể là chính xác, không bị thêm, sửa, xóa hay phát lại (đảm bảo tính toàn vẹn về nội dung).
- Đảm bảo đối tượng tạo ra thông tin (nguồn gốc thông tin) đúng là đối tượng hợp lệ đã được khai báo (đảm bảo tính toàn vẹn về nguồn gốc thông tin).

Để thực hiện xác thực thông tin, về nguyên tắc có 3 phương pháp sau đây:

1-Dùng các thuật toán mật mã (đối xứng và bất đối xứng) để xác thực thông tin. Nguyên tắc của mật mã là chỉ có những đối tượng hợp lệ mới khôi phục được thông tin gốc từ thông tin mật. Ta có thể sử dụng nguyên tắc này để xác thực thông tin như sau (hình 2.24):



M: thông tin gốc	E: thuật toán mật mã	D: Thuật toán giải mã
C: Thông tin mật	K: Khóa bí mật dùng chung giữa bên gửi và bên nhận	
PR <sub>a</sub> : Khóa bí mật của bên gửi.	PU <sub>a</sub> : Khóa công khai của bên gửi	

**Hình 2.24:** Xác thực thông tin dùng mật mã

Trường hợp thứ nhất: dùng mật mã đối xứng. Theo quy ước, chỉ có nơi gửi thông tin và nơi nhận thông tin hợp lệ mới có khóa bí mật K, do đó chỉ có thực thể gửi thông tin hợp lệ mới có khả năng tạo ra khối thông tin mật hợp lệ từ khối thông tin gốc M. Tương tự, chỉ có thực thể nhận thông tin hợp lệ mới có khả năng giải mã được thông tin mật để khôi phục đúng thông tin gốc M. Tất cả các cố gắng khác đều cho ra kết quả sai.

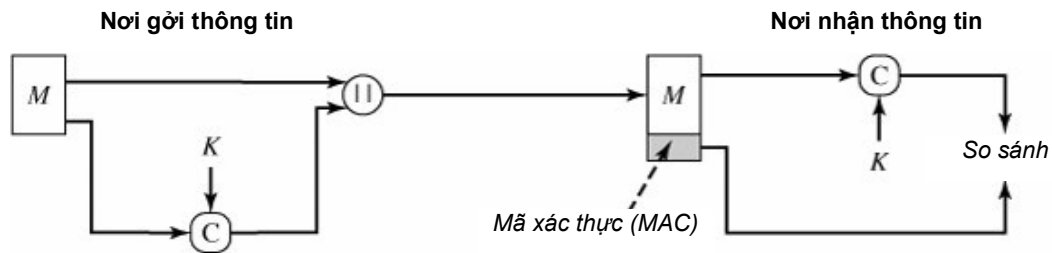
Trường hợp thứ hai: dùng mật mã bất đối xứng. Thực thể gửi thông tin thực hiện mã hóa dùng khóa bí mật (PR) thay vì dùng khóa công khai. Khối thông tin mật tạo ra có thể được giải mã bởi bất kỳ đối tượng nào biết khóa công khai của thực thể gửi. Tuy nhiên, nếu quá trình giải mã thành công, đối tượng nhận thông tin có thể chắc chắn rằng thông tin nhận được là đúng và chính đối tượng gửi hợp lệ đã gửi thông tin này, bởi vì chỉ có đối tượng đó mới có khóa riêng PR.

Phương pháp xác thực dùng mật mã dựa hoàn toàn vào độ tin cậy của khóa bí mật.

2-Dùng mã xác thực MAC (Message Authentication Code). Mã xác thực MAC được sinh ra từ tổ hợp gồm một khối thông tin gốc có độ dài bất kỳ và một khóa bí mật. Kích thước của MAC là cố định, không phụ thuộc vào kích thước của khối dữ liệu gốc và thường nhỏ hơn dữ liệu gốc. Đối tượng gửi sẽ gửi kèm giá trị MAC đi cùng với thông tin gốc. Phía nhận sau khi nhận



được thông tin gốc cùng với giá trị MAC gửi kèm sẽ thực hiện thao tác tạo ra giá trị MAC mới từ thông tin gốc cùng với khóa bí mật đã thống nhất giữa hai bên. Nếu giá trị MAC vừa tạo ra giống với giá trị MAC nhận được từ phía gửi, phía nhận có thể chắc chắn rằng thông tin gốc không bị thay đổi trong quá trình truyền (hình 2.25).



M: thông tin gốc  
C: Hàm tạo mã xác thực  
K: Khóa bí mật dùng chung giữa bên gửi và bên nhận  
||: Nối mã xác thực vào thông tin gốc

**Hình 2.25:** Xác thực thông tin dùng MAC

Việc dùng MAC để xác thực thông tin dựa vào hai cơ sở:

- Ứng với một khối thông tin gốc  $M$  và một khóa bí mật  $K$ , hàm  $C$  chỉ tạo ra duy nhất một mã xác thực MAC.
- Chỉ có phía gửi và phía nhận hợp lệ mới được biết khóa  $K$ .

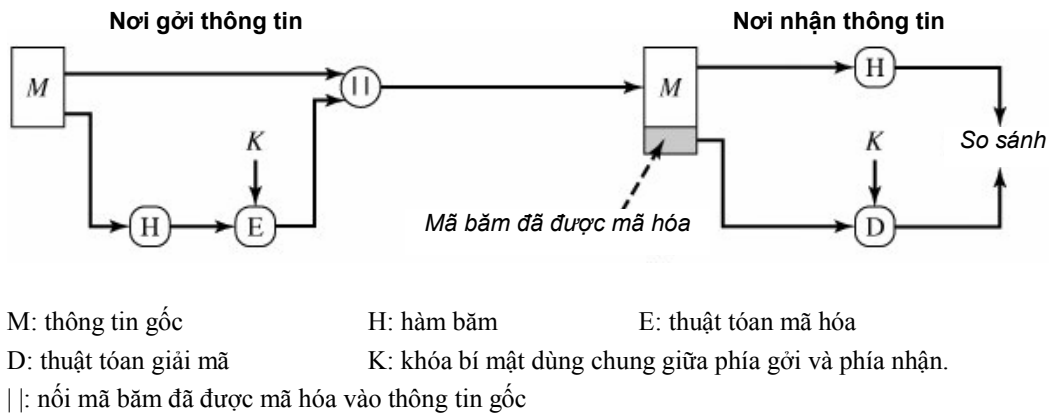
Có hai kỹ thuật tạo ra mã xác thực MAC: kỹ thuật thứ nhất dùng cơ chế mật mã khối (Cipher Block Chaining) và được gọi là CMAC hay CBC-MAC. Kỹ thuật thứ hai dựa trên các hàm băm bảo mật và được gọi là HMAC.

Mã xác thực MAC được ứng dụng trong các trường hợp thông tin chỉ yêu cầu đảm bảo tính xác thực mà không cần đảm bảo tính bí mật.

3-Dùng các hàm băm bảo mật (secure hash function). Giống như mã xác thực MAC, hàm băm cũng tạo ra một khối thông tin ngắn có độ dài xác định gọi là mã băm (hash code) từ một khối thông tin gốc có độ dài bất kỳ. Tuy nhiên, khác với MAC, hàm băm chỉ dựa vào thông tin gốc để tạo ra mã băm mà không dùng thêm bất kỳ khóa bí mật nào. Do vậy, để có thể sử dụng như một cơ chế xác thực thông tin, hàm băm phải được dùng kèm với một thuật toán mật mã nào đó (đối xứng hoặc bất đối xứng).

Hình 2.26 trình bày một ứng dụng điển hình của hàm băm trong xác thực thông tin. Theo cơ chế này, mã băm sau khi được tạo ra sẽ được mã hóa bằng một thuật toán mật mã đối xứng với khóa bí mật  $K$  chỉ có bên gửi và bên nhận biết. Đoạn mã băm đã được mật mã hóa được gửi đi kèm với thông tin gốc và quá trình kiểm tra ở phía nhận cũng được tiến hành theo trình tự ngược lại, tức là giải mã đoạn mã băm bằng khóa bí mật, sau đó tạo ra mã băm mới từ thông tin gốc và so sánh hai đoạn mã băm.

Có nhiều cách áp dụng các thuật toán mật mã vào hàm băm để xác thực thông tin: dùng mã đối xứng hoặc bất đối xứng, chỉ mã hóa mã băm hoặc mã hóa cả thông tin gốc và mã băm, thậm chí có thể tổ hợp nhiều cách trên lại với nhau.



**Hình 2.26:** Xác thực thông tin dùng hàm băm

Ngoài ứng dụng xác thực thông tin, hàm băm còn được dùng trong nhiều ứng dụng khác. Phần tiếp theo trình bày chi tiết hơn về các hàm băm bảo mật.

#### II.4.2 Các hàm băm bảo mật:

Các hàm băm bảo mật (secure hash functions) hay gọi tắt là hàm băm là một trong những kỹ thuật cơ bản để thực hiện các cơ chế xác thực thông tin (message authentication). Ngoài ra, hàm băm cũng còn được sử dụng trong nhiều thuật toán mật mã, trong chữ ký số (digital signature) và nhiều ứng dụng khác.

Nguyên tắc của hàm băm là biến đổi khối thông tin gốc có độ dài bất kỳ thành một đoạn thông tin ngắn hơn có độ dài cố định gọi là mã băm (*hash code* hay *message digest*). Mã băm được dùng để kiểm tra tính chính xác của thông tin nhận được. Thông thường, mã băm được gửi kèm với thông tin gốc. Ở phía nhận, hàm băm lại được áp dụng đối với thông tin gốc để tìm ra mã băm mới, giá trị này được so sánh với mã băm đi kèm với thông tin gốc. Nếu hai mã băm giống nhau, nghĩa là thông tin gửi đi không bị thay đổi.

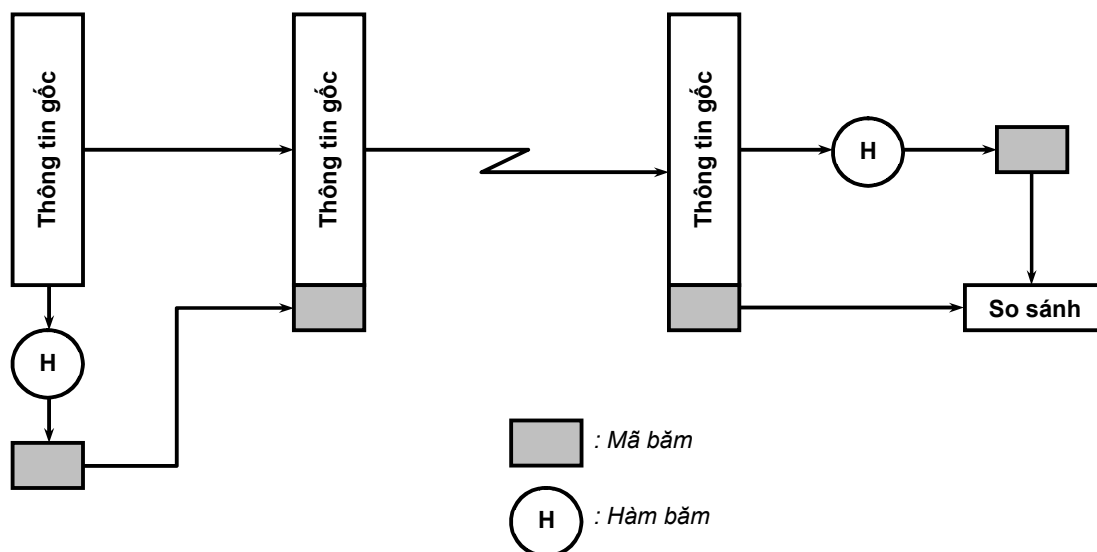
Chỉ có thể dùng hàm băm để tính mã băm từ thông tin gốc chứ không thể tính được thông tin gốc từ mã băm. Do đặc tính này, các hàm băm bảo mật cũng còn được gọi là hàm băm một chiều (one way hash function).

Hình 2.27 mô tả nguyên lý hoạt động của một giải thuật xác thực thông tin sử dụng hàm băm đơn giản.

Các yêu cầu của một hàm băm bảo mật  $H$ :

- $H$  có thể được áp dụng cho khối thông tin với chiều dài bất kỳ.
- Kết quả của hàm  $H$  luôn có chiều dài cố định.
- Việc tính giá trị của  $H(x)$  với một giá trị  $x$  cho trước phải đơn giản, có thể thực hiện được bằng cả phần cứng hoặc phần mềm.

- Cho trước một giá trị  $h$ , không thể tìm được một giá trị  $x$  sao cho  $H(x) = h$ , đây được gọi là thuộc tính một chiều của hàm băm (one-way property).
- Cho trước khối thông tin  $x$ , không thể tìm được một khối thông tin  $y$  khác  $x$  sao cho  $H(y) = H(x)$ . Thuộc tính này được gọi là **weak collision resistance**.
- Không thể tìm được hai khối thông tin  $x$  và  $y$  khác nhau sao cho  $H(x) = H(y)$ . Thuộc tính này được gọi là **strong collision resistance**.



**Hình 2.27:** Một ứng dụng điển hình của hàm băm

**-Tấn công trên các hàm băm:** Nguyên lý làm việc của hàm băm là biểu diễn một khối thông tin có kích thước lớn bởi một đoạn thông tin có kích thước nhỏ hơn nhiều gọi là mã băm, và trong trường hợp lý tưởng nhất thì các biểu diễn này là các ánh xạ 1:1, tức sẽ không xảy ra tình huống 2 khối thông tin khác nhau cùng cho ra một mã băm. Trường hợp có 2 khối thông tin khác nhau cùng cho ra một mã băm, ta nói thuật toán băm bị đụng độ (collision). Mục tiêu tấn công vào một hàm băm bảo mật là tạo ra các tình huống đụng độ này.

Xác suất để hai khối thông tin có cùng mã băm phụ thuộc vào kích thước của mã băm, tức phụ thuộc vào số lượng mã băm có thể có. Kích thước này càng nhỏ thì khả năng xảy ra càng lớn, và do đó xác suất tấn công thành công càng lớn. Bài toán ngày sinh (Birthday problem)<sup>(\*)</sup> chỉ ra rằng: với kích thước mã băm là  $n$  bit, để xác suất xảy ra đụng độ là 50% thì cần có khoảng  $2^{n/2}$  khối thông tin được xử lý. Người ta thường dùng nguyên lý này để tấn công vào các ứng dụng có sử dụng hàm băm, các tấn công này được gọi là **Birthday attack**.

Nói chung, độ an toàn của một hàm băm phụ thuộc vào kích thước ngõ ra của nó.

Phần sau đây sẽ giới thiệu một số thuật toán băm thông dụng thường được sử dụng trong xác thực thông tin.

### II.4.3 Thuật toán băm SHA:

<sup>(\*)</sup> Bài toán ngày sinh: trong một căn phòng có  $n$  người,  $n$  tối thiểu phải bằng bao nhiêu để có ít nhất 2 người có cùng ngày sinh (trong năm). Lý thuyết xác suất xác định  $n = 23$ . Bài toán này được mở rộng cho hàm băm.

SHA (Secure Hash Function) được chuẩn hoá năm 1993, sau đó được chỉnh sửa năm 1995 và đặt tên là SHA-1, từ đó phiên bản cũ được gọi là SHA-0.

SHA-1 tạo ra mã băm có chiều dài cố định là 160 bit. Về sau, có nhiều nâng cấp đối với SHA, chủ yếu là tăng chiều dài mã băm, từ đó xuất hiện các phiên bản khác nhau của SHA, bao gồm: SHA-256 (mã băm dài 256 bit), SHA-384 (mã băm dài 384 bit) và SHA-512 (mã băm dài 512 bit).

Bảng 2.2 tóm tắt các thông số của các phiên bản SHA.

**Bảng 2.2:** Các phiên bản SHA

Thông số	SHA-1	SHA-256	SHA-384	SHA-512
Kích thước mã băm (bit)	160	256	384	512
Kích thước thông tin gốc (bit)	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Kích thước khối (bit)	512	512	1024	1024
Độ dài từ (bit)	32	32	64	64
Số bước thực hiện (bước)	80	64	80	80

Phần này chỉ mô tả thuật toán băm SHA-1, các phiên bản khác của SHA cũng được thiết kế theo nguyên lý tương tự.

SHA-1 chấp nhận các khối thông tin có kích thước tối đa là  $2^{64}$  bit để tạo ra mã băm với độ dài cố định 160 bit. Toàn bộ khối thông tin được xử lý theo từng khối 512 bit, qua 5 công đoạn như sau:

**1- Gắn bit đệm – Append padding bit:** thông tin gốc được gắn thêm các bit thừa để có chiều dài (448 modulo 512) bit, tức là tất cả các khối trước có chiều dài bằng nhau là 512 bit, riêng khối cuối cùng là 448 bit. Chú ý rằng việc chèn thêm bit vào khối thông tin được thực hiện đối với tất cả các khối thông tin gốc, kể cả khi khối thông tin gốc có số bit chính xác bằng 448 mod 512 (khi đó chuỗi bit chèn vào sẽ có chiều dài là 512 bit).

**2- Gắn chiều dài – Append length:** một chuỗi 64 bit được gắn thêm vào khối thông tin. 64 bit này được xử lý như một số nguyên không dấu, cho biết chiều dài của khối thông tin gốc (tức chiều dài thật sự khi chưa thực hiện công đoạn 1).

Sau công đoạn này, khối thông tin nhận được có chiều dài là bội số của 512 bit, được chia thành các nhóm, mỗi nhóm tương đương với 16 thanh ghi 32 bit ( $16 \times 32 = 512$  bit).

**3- Khởi tạo bộ đệm MD – Initialize MD buffer:** bộ đệm MD (message digest) là bộ nhớ có dung lượng 160 bit dùng để chứa các kết quả trung gian và kết quả cuối cùng của mã băm. Bộ nhớ này được tổ chức thành 5 thanh ghi 32 bit và được khởi tạo các giá trị ban đầu như sau (Hex):

A = 67452301

B = EFCDAB89

C = 98BADCFE

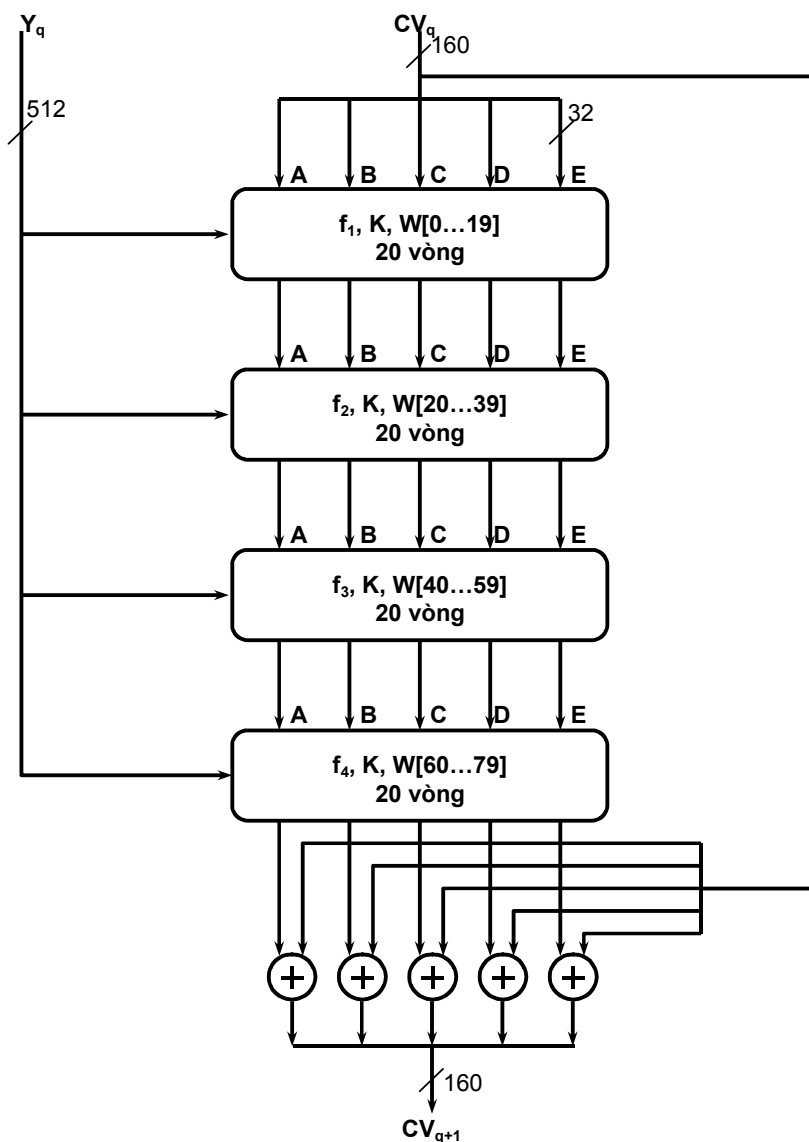
D = 10325476

E = C3D2E1F0

**4- Xử lý thông tin theo từng khối 512 bit – Process message:** đây là công đoạn trung tâm của hàm băm, còn được gọi là hàm nén (compress function), bao gồm 4 vòng, mỗi vòng 20 bước. Hình 2.28 trình bày sơ đồ khối của bước 4.

Cả 4 vòng có cấu trúc tương tự nhau, nhưng mỗi vòng sử dụng một hàm luận lý khác nhau là  $f_1$ ,  $f_2$ ,  $f_3$  và  $f_4$ .

Ngõ vào của mỗi vòng là khối bit  $Y$  (512 bit) đang xử lý cùng với giá trị của bộ đệm MD. Mỗi vòng sử dụng một biến cộng  $K_t$  khác nhau, với  $0 \leq t \leq 79$  biểu diễn cho 80 bước của 4 vòng. Tuy nhiên, thực tế chỉ có 4 giá trị  $K$  khác nhau như sau:



**Hình 2.28:** Xử lý thông tin trong SHA-1

Bước	Giá trị K (Hexa)
$0 \leq t \leq 19$	$K_t = 5A827999$
$20 \leq t \leq 39$	$K_t = 6ED9EBA11$

$$40 \leq t \leq 59 \quad K_t = 8F1BBCDC$$

$$60 \leq t \leq 79 \quad K_t = CA62C1D6$$

Ngõ ra của vòng thứ tư (tức bước 80) được cộng với ngõ vào của vòng đầu tiên để tạo ra  $CV_{q+1}$ . Thao tác cộng được thực hiện một cách độc lập, ứng với từng thanh ghi trong bộ đệm MD với một từ tương ứng trong  $CV_q$ , sử dụng phép cộng modulo  $2^{32}$ .

**5- Xuất kết quả - Output:** Sau khi tất cả các khối 512 bit đã được xử lý, ngõ ra của bước cuối cùng chính là giá trị của mã băm.

Một thuộc tính quan trọng của giải thuật băm SHA-1 là mỗi bit trong mã băm đều có quan hệ với tất cả các bit trong thông tin gốc. Việc lặp lại các hàm  $f$  một cách phức tạp như vậy nhằm mục đích đảm bảo rằng dữ liệu đã được trộn một cách kỹ lưỡng và do đó rất khó tìm được 2 khối thông tin gốc khác nhau có thể tạo ra cùng một mã băm.

#### II.4.4 Thuật toán băm MD5:

MD5 là một giải thuật xác thực thông tin được sử dụng phổ biến trong thời gian qua trong cộng đồng Internet, đặc biệt dùng để kiểm tra tính chính xác của các phần mềm mã nguồn mở phát hành trên mạng. Giải thuật này được xây dựng bởi Ron Rivest, và được chuẩn hóa bằng RFC 1321. MD5 có thể xử lý các khối thông tin có độ dài không giới hạn để tạo ra mã băm dài 128 bit. Thông tin gốc cũng được xử lý theo từng đoạn 512 bit. Bảng 2.3 so sánh các thông số giữa SHA-1 và MD5.

**Bảng 2.3: So sánh MD5 và SHA-1**

Thông số so sánh	MD5	SHA-1
Kích thước mã băm (bit)	128	160
Kích thước khối (bit)	512	512
Số bước	64	80
Kích thước thông tin gốc (bit)	Không giới hạn	$< 2^{64}$
Số lượng hàm luận lý	4	4

Với 128 bit mã băm, việc tìm ra hai khối thông tin để có cùng một giá mã băm không còn là điều bất khả thi đối với năng lực của các bộ xử lý hiện nay. Do đó, độ an toàn của MD5 đang bị đe dọa nghiêm trọng, và trong thời gian ngắn sắp tới, mức độ phổ biến của MD5 có thể sẽ giảm đi và được thay thế bằng một giải thuật xác thực khác.

## II.5 CHỮ KÝ SỐ

### II.5.1 Nguyên lý hoạt động của chữ ký số:

Chữ ký số là một cơ chế xác thực cho phép người tạo ra thông tin (message creator) gắn thêm một đoạn mã đặc biệt vào thông tin có tác dụng như một chữ ký. Chữ ký được tạo ra bằng cách áp dụng một hàm băm lên thông gốc, sau đó mã hóa thông tin gốc dùng khóa riêng của người gửi. Chữ ký số có mục đích đảm bảo tính toàn vẹn về nguồn gốc và nội dung của thông tin.

*Tại sao phải dùng chữ ký số trong khi các cơ chế xác thực thông tin (message authentication) đã thực hiện chức năng xác thực nguồn gốc thông tin?* Các cơ chế xác thực thông tin sử dụng các hàm băm một chiều có tác dụng bảo vệ thông tin trao đổi giữa hai thực thể thông

tin khỏi sự xâm phạm của một thực thể thứ 3, tuy nhiên nó không có tác dụng ngăn chặn được sự xâm phạm của chính hai thực thể. Ví dụ:

Thực thể A gửi một bản tin X cho thực thể B sử dụng một cơ chế xác thực nào đó, cơ chế này đảm bảo chỉ có A và B dùng chung một khoá bí mật K để tạo ra các mã xác thực từ thông tin gốc. Tuy nhiên, thực thể B có thể đổi bản tin X thành một bản tin Y, và với khóa bí mật K, thực thể B hoàn toàn có thể tạo ra thông tin xác thực mới để gắn vào Y, làm cho nó trở thành một bản tin hợp lệ mặc dù thực chất đây không phải là bản tin do thực thể A tạo ra.

Một ví dụ khác, thực thể A có thể từ chối xác nhận việc mình đã gửi bản tin X cho thực thể B, vì với các cơ chế xác thực như trên, thực thể B hoàn toàn có khả năng giả mạo thông tin đưa ra từ thực thể A.

Giống như một chữ ký thông thường (chữ ký bằng tay), một chữ ký số phải có đầy đủ các thuộc tính sau đây:

- *Phải xác nhận chính xác người ký và ngày giờ phát sinh chữ ký.*
- *Phải xác thực nội dung thông tin ngay tại thời điểm phát sinh chữ ký.*
- *Phải có khả năng cho phép kiểm chứng bởi một người thứ 3 để giải quyết các tranh chấp nếu có.*

Như vậy, chức năng của chữ ký số bao gồm chức năng của xác thực thông tin.

Các yêu cầu đối với chữ ký số:

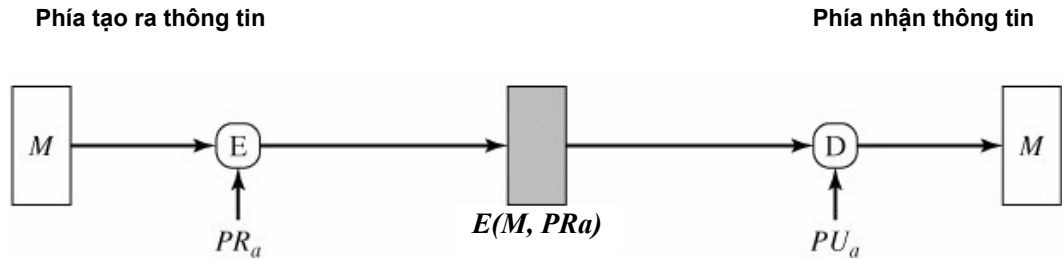
- Là một chuỗi bit phát sinh từ khối thông tin cần được xác nhận (thông tin gốc).
- Chữ ký phải chứa thông tin nhận dạng riêng của người ký để tránh giả mạo và tránh phủ nhận.
- Quy trình tạo ra chữ ký cũng như xác minh chữ ký phải đơn giản, nhanh chóng
- Chữ ký thông thể bị giả mạo bằng bất cứ cách nào.
- Có thể sao chép một bản sao của chữ ký dành cho mục đích lưu trữ.

**-Phân loại chữ ký số:** Có nhiều thuật toán phát sinh chữ ký số khác nhau. Có thể phân loại các thuật toán này theo các cách như sau:

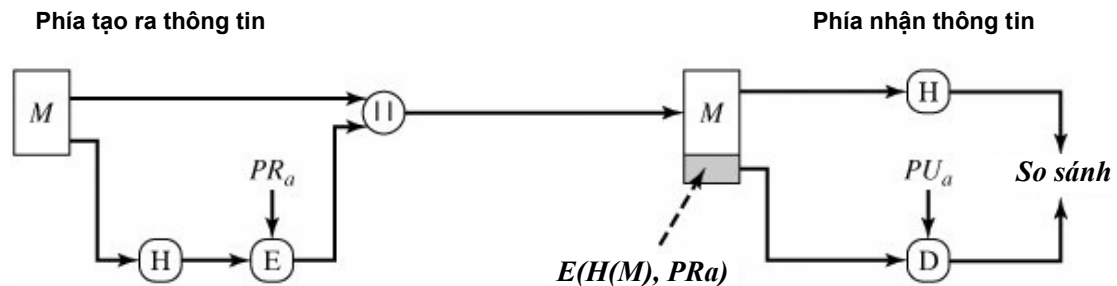
- Chữ ký cố định và chữ ký ngẫu nhiên: thuật toán tạo chữ ký cố định (*deterministic*) tạo ra một chữ ký duy nhất ứng với một khối thông tin gốc xác định, nghĩa là nếu thực hiện nhiều lần thuật toán tạo chữ ký trên một bản tin thì vẫn cho ra một kết quả duy nhất. Ngược lại, chữ ký ngẫu nhiên (*probabilistic*) tạo ra những chữ ký khác nhau đối với cùng một bản tin.
- Chữ ký phục hồi được và chữ ký không phục hồi được: cơ chế tạo chữ ký phục hồi được (*reversible signature*) cho phép người nhận phục hồi lại thông tin gốc từ chữ ký, điều này cũng có nghĩa là chữ ký phải có chứa thông tin gốc trong nó dưới một dạng mã hoá nào đó, và kết quả là chữ ký số sẽ có kích thước lớn hơn thông tin gốc. Khi đó, người gởi chỉ cần gởi đi chữ ký là đủ. Do vậy, cơ chế tạo chữ ký này cũng còn được gọi là chữ ký khôi phục bản tin (*signature with message recovery*). Ngược lại, cơ chế tạo chữ ký không phục hồi được (*non-reversible signature*) không cho phép phục hồi thông tin gốc từ chữ ký, do vậy, chữ ký chỉ là một khối thông tin cộng thêm có kích thước nhỏ hơn thông tin gốc. Người gởi cần phải gởi chữ ký đi kèm với thông tin gốc như một dạng phụ lục, do đó cơ chế tạo chữ ký này cũng còn được gọi là chữ ký với phụ lục (*signature with appendix*).

**-Các phương pháp thực hiện chữ ký số:** Có hai phương pháp thực hiện chữ ký số là ký trực tiếp (*direct signature*) và ký thông qua trọng tài (*arbitrated signature*).

- Ký trực tiếp (*direct signature*): Ở phương pháp này, giả thiết rằng phía nhận biết được khóa công khai của phía gửi. Do đó, chữ ký có thể được tạo ra bằng cách mã hóa toàn bộ bản tin bằng khóa riêng của người tạo ra thông tin, hoặc là chỉ mã hóa phần mã băm (kết quả tạo ra từ hàm băm đối với thông tin gốc) dùng khóa riêng của người tạo ra thông tin.



a- Tạo chữ ký trực tiếp bằng cách mã hóa toàn bộ thông tin gốc



b- Tạo chữ ký trực tiếp bằng cách mã hóa phần mã băm của thông tin gốc

M: thông tin gốc

E: Thuật toán mã hóa

D: Thuật toán giải mã

H: Hàm băm

$||$ : Nối mã băm vào thông tin gốc

$PR_a$ : Khóa bí mật của người ký

$PU_a$ : Khóa công khai của người ký

**Hình 2.29:** Chữ ký trực tiếp

Để đạt được tính bảo mật của thông tin thì thông tin gốc cùng với chữ ký vừa được tạo ra sẽ được mã hóa sử dụng khóa công khai của thực thể nhận chữ ký (trong trường hợp dùng mật mã bất đối xứng) hoặc dùng khóa bí mật (trong trường hợp dùng mật mã đối xứng).

Một nhược điểm rất dễ thấy của phương thức ký trực tiếp đó là độ an toàn của chữ ký phụ thuộc cao độ vào khóa riêng của người tạo ra chữ ký. Do vậy, nếu khóa riêng này bị mất hoặc bị tiết lộ thì ý nghĩa của chữ ký số sẽ không còn.



- Ký thông qua trọng tài (arbitrated signature): đây là một giải pháp được xây dựng để khắc phục nhược điểm của chữ ký trực tiếp. Khi thực thể A muốn gửi một bản tin cho thực thể B, quá trình tạo ra một chữ ký được thực hiện bình thường như đối với chữ ký trực tiếp. Tuy nhiên, trước khi bản tin này được gửi đến B, nó phải được gửi đến một thực thể thứ 3 gọi là trọng tài (arbiter). Trọng tài thực hiện việc kiểm tra, xác nhận tính chính xác của thông tin và chữ ký, sau đó ghi lại ngày giờ rồi mới gửi cho thực thể B, kèm theo thông tin xác nhận của trọng tài. Sự xuất hiện của trọng tài trong quy trình đảm bảo được thực thể A sẽ không phủ nhận được thông tin mình đã gửi.

Nếu gọi X là thực thể tạo ra thông tin, Y là thực thể nhận thông tin, A là trọng tài, H là hàm băm bảo mật và E là thuật toán mật mã, quá trình tạo chữ ký thông qua trọng tài được thực hiện như sau:

-Trường hợp thứ nhất: sử dụng kỹ thuật mật mã đối xứng và trọng tài có thể đọc nội dung thông tin mà X gửi cho Y:

Bước 1:  $X \rightarrow A: M + E([ID_X + H(M)], K_{xa})$

Bước 2:  $A \rightarrow Y: E([ID_X + M + E([ID_X + H(M)], K_{xa}) + T], K_{ay})$

Với M là thông tin gốc mà X gửi cho Y,  $K_{xa}$  là khóa bí mật dùng chung giữa X và A,  $K_{ay}$  là khóa bí mật dùng chung giữa Y và A,  $ID_X$  là thông tin nhận dạng của thực thể X và T là thời điểm chữ ký được tạo ra.

-Trường hợp thứ 2: sử dụng kỹ thuật mật mã đối xứng và trọng tài không đọc được nội dung thông tin X gửi cho Y:

Bước 1:  $X \rightarrow A: ID_X + E(M, K_{xy}) + E([ID_X + H(E(M, K_{xy}))], K_{xa})$

Bước 2:  $A \rightarrow Y: E([ID_X + E(M, K_{xy})], K_{ay}) + E([ID_X + H(E(M, K_{xy})) + T], K_{xa})$

Với  $K_{xy}$  là khóa bí mật dùng chung giữa X và Y.

-Trường hợp thứ 3: sử dụng kỹ thuật mật mã bất đối xứng, trọng tài không đọc được nội dung thông tin X gửi cho Y:

Bước 1:  $X \rightarrow A: ID_X + E([ID_X + E(E(M, PR_x), PU_y)], PR_x)$

Bước 2:  $A \rightarrow Y: E([ID_X + E(E(M, PR_x), PU_y) + T], PR_a)$

Với  $PR_x$  là khóa riêng của X,  $PU_y$  là khóa công khai của Y,  $PR_a$  là khóa riêng của A

## II.5.2 Chuẩn chữ ký DSS:

DSS (Digital Signature Standard) là một chuẩn về chữ ký số, được chuẩn hóa năm 1991, sửa đổi năm 1993 và 1996, sau đó mở rộng vào năm 2000. DSS sử dụng hàm băm SHA và thuật toán tạo chữ ký DSA (Digital Signature Algorithm). DSS thuộc loại chữ ký ngẫu nhiên và không phục hồi được.

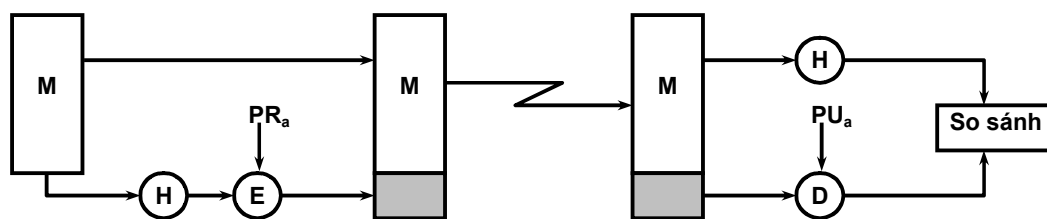
Hình 2.30 so sánh cấu trúc DSS so với phương thức xác thực thông tin sử dụng mật mã bất đối xứng RSA.

Trong thuật toán xác thực thông tin dùng mật mã RSA, thông tin gốc được đưa vào hàm băm SHA để tạo ra mã băm (tức message digest) có kích thước cố định. Mã băm này sau đó được mã hóa (bằng thuật toán RSA) dùng khóa riêng của thực thể tạo thông tin (phía gửi). Kết quả của phép mã hóa được gắn vào thông tin gốc và gửi đi. Phía thu nhận được thông tin, tách phần mã băm ra khỏi thông tin gốc và giải mã nó bằng khóa công khai của phía gửi. Chú ý rằng khóa công

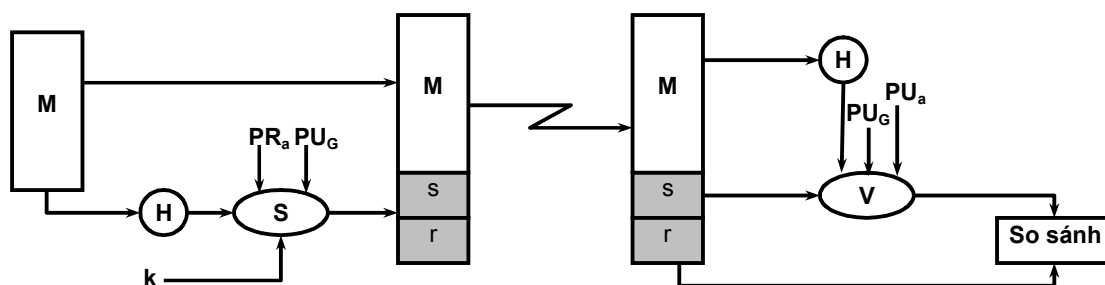
khai là thông tin được công bố rộng rãi cho bất kỳ thực thể nào có quan tâm. Đồng thời, thông tin gốc cũng được đưa vào hàm băm để tính mã băm, sau đó đem so sánh với mã băm vừa nhận được. Nếu hai mã này giống nhau thì thông tin vừa nhận được chấp nhận như là thông tin hợp lệ.

Hoạt động của DSS cũng bao gồm việc đưa thông tin gốc vào hàm băm để tạo ra mã băm có kích thước cố định. Tuy nhiên, mã băm này sẽ không được mã hóa trực tiếp bằng một giải thuật mã hóa mà được sử dụng làm ngõ vào của một hàm tạo chữ ký  $S$  (Signature function). Các thông tin đưa vào hàm tạo chữ ký bao gồm:

- Mã băm của thông tin gốc
- Một số ngẫu nhiên  $k$
- Khóa riêng của người ký ( $PR_a$ )
- Khóa công khai của nhóm các thực thể liên quan đến giao dịch chữ ký ( $PU_G$ )



a- Xác thực thông tin dùng mật mã RSA



a- Dùng chữ ký số DSS

**Hình 2.30:** Xác thực thông tin dùng mật mã RSA và dùng chữ ký số DSS

Kết quả của hàm sinh chữ ký gồm hai thành phần, đặt tên là  $r$  và  $s$ . Cả hai được gửi kèm với thông tin gốc.

Ở phía nhận thu, thông tin gốc được tách riêng để đưa vào hàm băm. Sau đó, mã băm được đưa vào hàm kiểm chứng  $V$  (Verification function) cùng với khóa công khai của nhóm ( $PU_G$ ) và khóa công khai của phía gửi ( $PR_A$ ). Nếu kết quả của hàm kiểm chứng bằng với thành phần  $r$  của chữ ký thì thông tin được xem là xác thực.

Hình 2.31 mô tả quá trình tạo chữ ký và kiểm chứng chữ ký dùng DSS.

Chú ý rằng thành phần  $r$  của chữ ký không phụ thuộc vào thông tin gốc mà chỉ phụ thuộc vào số ngẫu nhiên  $k$  và 3 thành phần của khóa công khai của nhóm ( $PU_G$ ) là  $p$ ,  $q$  và  $g$ . Do vậy, để

a- Tạo chữ ký

b- Kiểm chứng chữ ký

$$w = (s')^{-1} \bmod q$$

$$u_1 = [H(M')w] \bmod q$$

$$u_2 = (r')w \bmod q$$

$$v = [(g^{u_1} y^{u_2}) \bmod p] \bmod q$$

Lưu ý:  $s', r', M'$  tương ứng với các phần  $s, r$  và  $M$  tại phía thu.

Với độ phức tạp của phép toán logarit rời rạc, rất khó có thể xác định được  $k$  khi biết  $r$  hoặc xác định được  $x$  khi biết  $s$ .

## II.6 QUẢN LÝ KHOÁ

### II.6.1 Quản lý khoá công khai trong mật mã bất đối xứng:

Trong kỹ thuật mật mã bất đối xứng, khoá riêng của mỗi thực thể được chính thực thể đó quản lý mà không cần phải chia sẻ cho ai, tuy nhiên cơ chế nào được dùng để phổ biến khoá công khai một cách an toàn và hiệu quả?

Các cơ chế khác nhau có thể dùng để phổ biến khoá công khai bao gồm:

-*Phổ biến công khai trên các diễn đàn công cộng*: người sử dụng thực hiện việc này bằng cách gửi các thông báo kèm theo khoá công khai của mình đến các website hoặc diễn đàn công cộng trên mạng Internet. Phương pháp này đơn giản nhưng có nhược điểm là khoá dễ bị giả mạo. Một người A có thể đưa khoá công khai của mình lên mạng nhưng thông báo rằng đó là khoá của người B, bằng cách đó, A có thể đọc được những thông tin bí mật mà người khác gửi cho B.

-*Sử dụng danh bạ khoá công khai (public key directory)*: với danh bạ này, những người dùng nào muốn phổ biến khoá của mình thì phải đăng ký với nhà xuất bản, và để tránh việc giả mạo, nhà xuất bản phải áp dụng một cơ chế kiểm duyệt an toàn nào đó đối với người đăng ký.

Phương pháp này an toàn hơn cách mà mỗi cá nhân tự phổ biến khoá của mình. Tuy nhiên, nó cũng có khả năng bị giả mạo khi khoá bí mật của nhà xuất bản bị lộ, kẻ tấn công có thể thay đổi các thông tin mà người sử dụng đã đăng ký lên đó.

-*Chứng thực khoá công khai (public-key certificate)*: Phương pháp sử dụng danh bạ công cộng có một điểm yếu khác đó là mọi người dùng muốn liên lạc với một người khác cần đến khoá công khai thì phải liên lạc với nhà xuất bản để được cung cấp, điều này đặt nhà xuất bản vào trạng thái có nguy cơ quá tải bất cứ lúc nào, hơn nữa đây chính là điểm thất bại của các giao dịch trên mạng.

Khái niệm **chứng thực khoá công khai** (public key certificate hay gọi tắt là certificate hay chứng thực khoá) là một cơ chế phổ biến khoá công khai trong đó mỗi thực thể tự phổ biến khoá của mình bằng bất cứ phương tiện gì những vẫn đảm bảo được tính xác thực của khoá.

Chứng thực khoá công khai là một tổ hợp gồm có *khóa công khai của một thực thể, nhận dạng của thực thể đó và chữ ký số (digital signature) xác nhận của một thực thể thứ 3*, thực thể thứ 3 này là một tổ chức được tin tưởng trong cộng đồng (ví dụ như cơ quan nhà nước hoặc các tổ chức tài chính). Các đặc trưng của cơ chế này bao gồm:

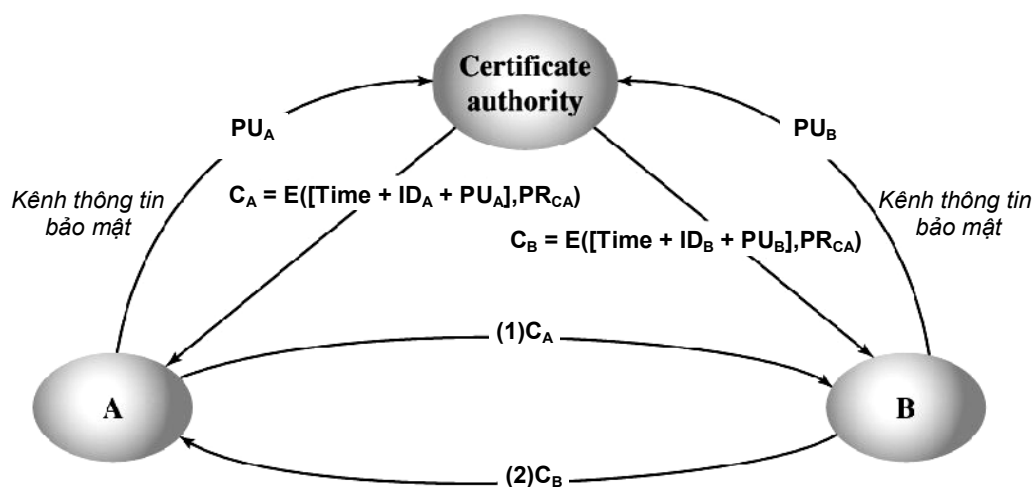
- Mỗi thực thể đều có thể đọc các chứng thực khoá để biết được khoá công khai cũng như nhận diện chủ sở hữu của khoá đó.
- Mỗi thực thể đều có thể xác thực thông tin trong chứng thực khoá là chính xác nhờ vào chữ ký của một thực thể được tin cậy thứ 3.

- Chỉ có người chứng thực (Certificate Authority hay CA) mới có quyền tạo ra và cấp nhật các chứng thực khóa.

Quá trình tạo ra và phân phối chứng thực khóa diễn ra như sau (hình 2.32):

-Để tạo chứng thực khóa cho mình, thực thể A gửi yêu cầu đến cơ quan chứng thực CA (Certificate Authority), trong yêu cầu có chứa khoá công khai của A ( $PU_A$ ). Để tránh các tình huống giả mạo CA, yêu cầu cung cấp chứng thực gửi từ các thực thể đầu cuối phải được gửi đến CA bằng một kênh bảo mật, trên đó có áp dụng các cơ chế xác thực chặt chẽ.

-CA tạo ra chứng thực khóa cho A bằng cách mã hoá khối thông tin bao gồm: nhận dạng của thực thể A ( $ID_A$ ), khoá công khai của A ( $PU_A$ ) và thời điểm thực hiện việc cấp chứng thực, bằng khoá riêng của CA ( $PR_{CA}$ ).



$C_A$ : Chứng thực khóa của thực thể A	$ID_A$ : Thông tin nhận dạng của thực thể A
$C_B$ : Chứng thực khóa của thực thể B	$ID_B$ : Thông tin nhận dạng của thực thể B
$PU_A$ : Khoá công khai của thực thể A	$PR_{CA}$ : Khoá riêng của CA
$PU_B$ : Khoá công khai của thực thể B	Time: Thời điểm tạo ra chứng thực khóa

**Hình 2.32:** Quản lý khoá công khai dùng chứng thực khóa (Certificate)

Như vậy, thực thể A đã tạo được chứng thực khóa cho mình ( $C_A$ ).

Tương tự như vậy, thực thể B cũng yêu cầu CA cung cấp chứng thực khóa cho nó ( $C_B$ ).

Để bắt đầu trao đổi thông tin với nhau sử dụng mật mã bất đối xứng, hai thực thể A và B trao đổi chứng thực khóa cho nhau để thực thể này nhận được khoá công khai của thực thể kia.

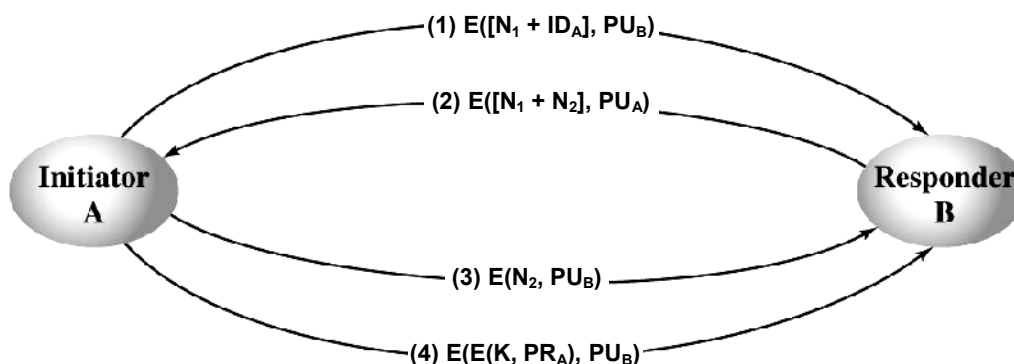
Với việc nhờ một thực thể tin cậy thứ 3 làm trung gian để tạo ra chứng thực khóa, khoá công khai có thể được phân phối một cách an toàn mà không bị giả mạo.

Một trong những cơ chế được sử dụng rộng rãi để tạo ra các chứng thực khóa công khai là chuẩn X.509. Chuẩn này được dùng trong nhiều dịch vụ và giao thức bảo mật như IPSec, SSL, S/MIME, SET, ...

## II.6.2 Sử dụng mật mã bất đối xứng để trao đổi khóa bí mật:

Trong kỹ thuật mật mã đối xứng, cả hai thực thể thông tin phải dùng chung một khóa bí mật. Vấn đề là làm thế nào để trao đổi khóa bí mật giữa hai thực thể này?

Thuật toán trao đổi khóa Diffie-Hellman được trình bày trong phần mã hóa bất đối xứng là một thuật toán an toàn, cho phép hai thực thể trao đổi khóa bí mật mà một thực thể thứ 3 không lấy cắp được. Tuy nhiên, hạn chế của Diffie-Hellman là không có tính xác thực, nghĩa là một thực thể sẽ không thể biết chắc chắn rằng khóa mình nhận được đúng là khóa của thực thể mà mình đang muốn trao đổi thông tin hay không. Do vậy, trong thực tế, Diffie-Hellman thường được dùng phối hợp với một cơ chế xác thực đầu cuối (peer authentication).



**Hình 2.33:** Dùng mật mã bất đối xứng để trao đổi khoá

Dùng khóa công khai để trao đổi khóa bí mật của mã hóa đối xứng là một cách hiệu quả có thể giải quyết được vấn đề trên đây. Một thực thể A (*thực thể khởi tạo – Initiator*) muốn trao đổi khóa bí mật với một thể B (*thực thể đáp ứng - responder*) có thể thực hiện thủ tục trao đổi khoá như sau:

(1)-A dùng khoá công khai của B ( $PU_B$ ) để mã hoá một bản tin, bản tin này chứa nhận dạng của A ( $ID_A$ ) và một giá trị ngẫu nhiên  $N_1$  (nonce) để nhận diện giao tác đang thực hiện.

$$A \rightarrow B: E([N_1 + ID_A], PU_B)$$

(2)-B gửi lại cho A một bản tin chứa giá trị ngẫu nhiên  $N_2$  do B tạo ra, cùng với số  $N_1$  nhận được từ A. Toàn bộ bản tin được mã hoá sử dụng khoá công khai của A ( $PU_A$ ).

$$B \rightarrow A: E([N_1 + N_2], PU_A)$$

(3)-Một lần nữa, A gửi lại cho B một bản tin chứa giá trị  $N_2$  được mã hoá bằng khoá công khai của A ( $PU_A$ ).

$$A \rightarrow B: E(N_2, PU_B)$$

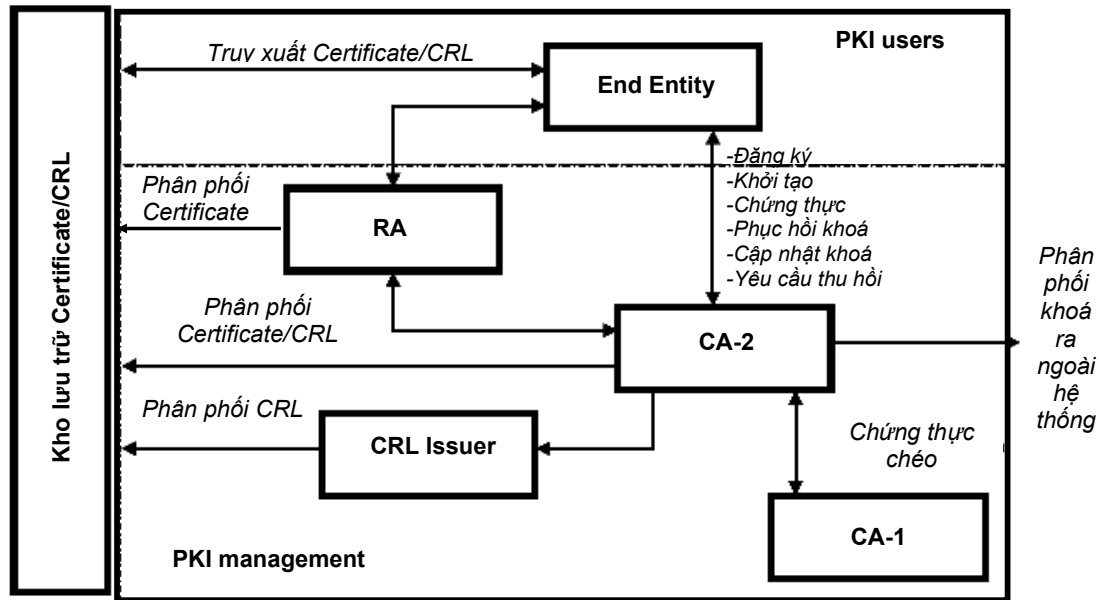
(4)-A chọn khoá bí mật K cho thuật toán mã hoá đối xứng sắp diễn ra, sau đó mã hoá nó bằng chính khoá riêng của A ( $PR_A$ ), rồi mã hoá một lần nữa bằng khoá công khai của B ( $PU_B$ ) rồi gửi cho B. Đến bước này, B đã nhận được khoá bí mật mà A tạo ra một cách an toàn.

$$A \rightarrow B: E(E(K, PR_A), PU_B)$$

### II.6.3 Cơ sở hạ tầng khóa công khai:

Cơ sở hạ tầng khóa công khai PKI (Public Key Infrastructure) là một hệ thống hạ tầng bao gồm các thiết bị phần cứng, chương trình phần mềm, các chính sách, thủ tục và con người cần thiết để tạo ra, quản lý, lưu trữ và phân phối các chứng thực khóa phục vụ cho mục đích phổ biến khóa công khai của các thực thể thông tin.

Vai trò của PKI trong hệ thống là quản lý các chứng thực khóa một cách an toàn và cung



Hình 2.34: Cấu trúc PKI

cấp nó cho user một cách hiệu quả nhất.

Mục tiêu của PKI là cung cấp một môi trường làm việc phối hợp, trong đó, thiết bị, phần mềm của nhiều nhà sản xuất khác nhau có thể cùng sử dụng chung một cấu trúc chứng thực khóa.

#### -Các thành phần của PKI:

- *End Entity (thực thể đầu cuối)*: là người sử dụng, một phần mềm hoặc một thiết bị tham gia vào quá trình trao đổi thông tin sử dụng mã hóa khóa công khai. Các thực thể có một cặp khóa của mình, trong đó khóa công khai được phổ biến bởi PKI dưới dạng các chứng thực khóa, còn khóa bí mật do chính thực thể quản lý.
- *Certificate Authority (CA)*: là thực thể tạo ra các chứng thực khóa. CA tạo ra chứng thực khóa từ các khóa công khai mà các thực thể đầu cuối ủy quyền cho nó phổ biến cộng với chữ ký số của chính CA đó. Do vậy, CA phải là một thực thể được tin cậy, nếu không, chữ ký của CA sẽ không có ý nghĩa gì.
- *Registration Authority (RA)*: là một thành phần tùy chọn của PKI, có chức năng xử lý một số công việc quản lý nhằm giảm tải cho CA, chẳng hạn như đăng ký thực thể đầu cuối, kiểm chứng các thực thể đầu cuối, tạo ra các cặp khóa public-private, ...
- *Repository*: Kho lưu trữ chứng thực khóa và cung cấp chứng thực khóa cho các thực thể đầu cuối khi có yêu cầu. Có nhiều cách để thực thể đầu cuối truy xuất các

chứng thực khóa tại PKI: thông qua dịch vụ thư mục LDAP (X.500), thông qua FTP hoặc HTTP, ...

- *Certificate revocation list (CRL) Issuer*: Một chứng thực khóa khi đã được tạo ra và phổ biến thì không có nghĩa là nó sẽ được tồn tại vĩnh viễn. Sau một khoảng thời gian nhất định hoặc theo yêu cầu của thực thể đầu cuối, chứng thực khóa có thể bị thu hồi. CRL là danh sách các chứng thực khóa bị thu hồi, được tạo ra bởi CA hoặc ủy quyền cho CRL issuer. Như vậy, CRL issuer cũng là một thành phần tùy chọn của PKI.

**-Các chức năng quản lý của PKI:**

- *Đăng ký (Registration)*: là thủ tục mà thực thể đầu cuối phải thực hiện để tham gia vào PKI lần đầu tiên.
- *Khởi tạo (Initialization)*: Khởi tạo các thông tin của thực thể đầu cuối tại CA, tạo ra cặp khóa public-private cho thực thể đầu cuối.
- *Chứng thực (Certification)*: CA tạo ra chứng thực khóa cho thực thể đầu cuối, ứng với khóa công khai vừa được tạo ra ở giai đoạn khởi tạo hoặc do thực thể đầu cuối cung cấp.
- *Phục hồi khóa (Key-pair recovery)*: cho phép phục hồi một khóa cũ trước đó. Thủ tục này thường được dùng trong trường hợp khóa mật mã vì một lý do nào đó không truy xuất được. Để khôi phục dữ liệu đã bị mật mã hoá, cần phải có thủ tục này để lấy lại khoá.
- *Cập nhật khóa (Key-pair update)*: Mỗi chứng thực khóa được tạo ra với một khoảng thời gian tồn tại nhất định, sau khoảng thời gian này, chứng thực khóa sẽ bị thu hồi (revoke). Thủ tục key-pair update có tác dụng gia hạn tồn tại của chứng thực khóa, cho phép một chứng thực khóa tiếp tục tồn tại sau khi đã hết thời gian hiệu lực.
- *Yêu cầu thu hồi chứng thực khóa (Revocation request)*: Yêu cầu thu hồi một chứng thực khóa vì một lý do nào đó, như khóa riêng bị lộ chẳng hạn. Thủ tục này cho phép một thực thể đầu cuối yêu cầu thu hồi một chứng thực khóa chưa hết hiệu lực.

Có thể tóm tắt các bước diễn hình của một quy trình khi một thực thể A muốn gửi một bản tin đến một thực thể B trong môi trường PKI như sau:

-Thực thể A thực hiện hàm băm trên bản tin để tạo ra mã băm. Sau đó mã băm được mã hóa bằng khóa riêng  $PR_A$  của thực thể A để tạo ra một chữ ký của thực thể A.

-Sử dụng khóa công khai của CA, thực thể A yêu cầu CA cung cấp khóa công khai của thực thể B ( $PU_B$ ).

-Thực thể A mã hóa bản tin bằng khóa công khai  $PU_B$  của thực thể B vừa nhận được từ CA, sau đó gắn chữ ký của mình vào bản tin đã mã hóa và gửi cho B.

-Thực thể B giải mã thông tin nhận được bằng khóa riêng của chính nó ( $PR_B$ ), sau đó áp dụng hàm băm lên bản tin này để tạo ra mã băm.

-Thực thể B giải mã chữ ký của thực thể A bằng khóa công khai của thực thể A ( $PU_A$ ), sau đó so sánh với mã băm vừa tạo ra ở bước trên. Nếu hai thông tin này giống nhau, thì bản tin nhận được xem như hợp lệ.



### Tóm tắt chương:

-Mật mã là một cơ chế cơ bản nhất được dùng để bảo đảm an toàn cho thông tin khi trao đổi giữa các hệ thống thông tin (thường thông qua mạng máy tính). Kỹ thuật mật mã bảo vệ được 2 đặc trưng của mô hình CIA là tính Bí mật và tính Toàn vẹn của thông tin.

-Kỹ thuật mật mã hiện đại được chia thành 2 loại: Mật mã đối xứng (symmetric key encryption) và mật mã bất đối xứng (asymmetric key encryption).

-Mật mã đối xứng (hay còn gọi là mật mã quy ước) sử dụng 1 khóa duy nhất cho việc mã hóa và giải mã, khóa này được giữ bí mật, chỉ có các thực thể tham gia việc truyền nhận thông tin mới biết được. Kỹ thuật mật mã quy ước dựa chủ yếu trên các thao tác xử lý bit (như dịch, xoay vòng, XOR, ...) do đó thích hợp với phần việc thực thi bằng phần cứng, tốc độ mã hoá cao. Các thuật toán mật mã đối xứng thông dụng bao gồm DES, Blowfish, IDEA, AES,...

-Mật mã bất đối xứng (hay còn gọi mật mã dùng khóa công khai) sử dụng 2 khóa khác nhau cho quá trình mã hóa và giải mã. Một trong hai khóa là khóa công cộng (public key), được phổ biến công khai cho bất kỳ một thực thể nào cũng có thể truy xuất được; và khóa còn lại là khóa riêng (private key) được giữ bí mật, chỉ có chủ thể của khóa đó biết. Mã hóa khóa công khai dựa chủ yếu trên các hàm toán học, do đó thích hợp với thực thi bằng phần mềm và tốc độ mã hoá thấp. RSA là thuật toán mật mã bất đối xứng phổ biến nhất hiện nay.

-Mật mã dùng khóa công khai có nhiều ứng dụng khác nhau như: mật mã dữ liệu, tạo chữ ký số, trao đổi khóa bí mật của mật mã đối xứng, ...

-Hàm băm bảo mật (secure hash function) là cơ chế dùng trong xác thực thông tin (message authentication). Nguyên lý của hàm băm là biến đổi khối thông tin gốc thành một giá trị kiểm tra có kích thước cố định gọi là mã băm, giá trị này được gửi đi kèm với thông tin gốc. Ở đầu thu, thông tin nhận được cũng được đưa vào hàm băm để tạo ra giá trị kiểm tra. Nếu giá trị kiểm tra vừa được tạo ra bằng với giá trị gửi kèm của phía gửi thì thông tin được xem là xác thực. Hàm băm được dùng trong các thuật toán xác thực thông tin (message authentication), tạo chữ ký số và là thành phần của một số thuật toán mật mã.

-Chữ ký số (digital signature) là kỹ thuật dùng để nhận dạng một thực thể thông tin cùng với thông tin do thực thể này tạo ra. Ứng dụng cơ bản nhất của chữ ký số là để chứng thực và đảm bảo tính không thể phủ nhận (non-repudiation) của thông tin. Có nhiều cách phân loại các thuật toán tạo chữ ký: chữ ký phục hồi được và không phục hồi được, chữ ký cố định và chữ ký ngẫu nhiên. Hai phương pháp thực hiện chữ ký số là ký trực tiếp và ký thông qua trọng tài. Chuẩn chữ ký số DSS sử dụng thuật toán DSA, tạo ra chữ ký số dựa trên các hàm băm bảo mật (SHA) và kỹ thuật mật mã khóa công khai (RSA).

-Mật mã hóa dùng khóa công khai chỉ có ưu điểm khi nó có một cơ chế phân phối khóa công khai một cách an toàn và hiệu quả cho các thực thể trong hệ thống. Chứng thực khóa công khai (Certificate) mà một cơ chế hiệu quả để thực hiện vấn đề này. Mỗi chứng thực khóa bao gồm nhận dạng thực thể đầu cuối, khóa công khai của thực thể đầu cuối và xác nhận (bằng chữ ký số) của một thực thể thứ 3. Một hệ thống cung cấp cơ chế tạo ra và quản lý chứng thực khóa được gọi là cơ sở hạ tầng khóa công khai PKI.

## **CÂU HỎI VÀ BÀI TẬP.**

### **A- Câu hỏi trắc nghiệm.**

Câu 1. Chức năng của mật mã thông tin:

- a- Bảo vệ tính toàn vẹn của thông tin.
- b- Bảo vệ tính bí mật của thông tin.
- c- Bảo vệ tính khả dụng của thông tin .
- d- Bảo vệ tính không thể phủ nhận của thông tin.

Câu 2. Các nguy cơ của một hệ thống mật mã:

- a- Tấn công bằng cách dò khoá bí mật (brute force attack).
- b- Tấn công bằng phương pháp phân tích mã (crypanalysis).
- c- Tấn công từ chối dịch vụ
- d- Câu a và b.

Câu 3. Một hệ thống mã hoá quy ước dùng khoá dài 128 bit. Nếu dùng phương pháp tấn công brute force thì phải thử trung bình bao nhiêu lần và thời gian cần thiết để thực hiện nếu tốc độ xử lý là một tỉ lần trong một giây?

- a- Phải thử  $2^{128}$  lần, thời gian thử là  $5,4 * 10^{18}$  năm.
- b- Phải thử  $2^{64}$  lần, thời gian thử là  $5,4 * 10^{18}$  năm.
- c- Phải thử  $2^{127}$  lần, thời gian thử là  $5,4 * 10^{18}$  năm.
- d- Phải thử  $2^{128}$  lần, thời gian thử là 18 năm.

Câu 4. Cơ chế trao đổi khoá bí mật trong mã hoá đối xứng?

- a- A và B trao đổi khoá với nhau bằng e-mail.
- b- A và B trao đổi khoá với nhau bằng một phương tiện vật lý.
- c- A gửi khoá bí mật cho một thực thể thứ 3 bằng e-mail, thực thể thứ 3 này sẽ gửi lại khoá này cho B cũng bằng e-mail.
- d- Một trong 3 cách trên đều được.

Câu 5. Chọn câu đúng khi nói về cấu trúc mật mã Feistel:

- a- Tất cả các thao tác trong cấu trúc đều có thao tác ngược tương ứng.
- b- Cấu trúc Feistel dựa trên các thao tác xử lý bit với các phép hoán vị và thay thế lặp lại nhiều lần.
- c- Kích thước khối dữ liệu (block size) là bất kỳ.
- d- Mạch mã hoá và mạch giải mã có cấu trúc khác nhau.

Câu 6. Chọn câu đúng khi nói về thuật toán mật mã DES:

- a- Mạch mã hoá và mạch giải mã là giống nhau.
- b- S-box là một hàm hoán vị, cho kết quả ngược lại với phép hoán vị IP.
- c- Thứ tự sinh khoá phụ ở quá trình mã hoá và giải mã là giống nhau.
- d- Thao tác hoán vị PC-1 được thực hiện 16 lần trong quá trình mã hoá và giải mã.

Câu 7. Chọn câu đúng về độ an toàn của DES:

- a- Chỉ có thể tấn công hệ thống mật mã DES bằng phương pháp brute force.
- b- Khoá đưa vào thuật toán là 64 bit, nhưng thực chất chỉ sử dụng 56 bit.

- c- Mạch giải mã TDES không thể giải mã được thông tin mã hoá bởi DES.
- d- Tất cả đều đúng.

Câu 8. Chọn câu đúng khi nói về chuẩn mật mã AES:

- a- Là chuẩn mật mã được thiết kế để làm việc song song với DES.
- b- Kích thước khối và chiều dài khoá có thể thay đổi được.
- c- Mạch mã hoá và mạch giải mã hoàn toàn giống nhau.
- d- Tất cả đều đúng.

Câu 9. Chọn phát biểu sai khi nói về chuẩn mật mã AES:

- a- Thứ tự sinh khoá phụ trong quá trình mã hoá và giải mã hoàn toàn giống nhau.
- b- Tất cả các thao tác trong thuật toán đều có thao tác ngược.
- c- AES không dựa trên cấu trúc mã khối Feistel.
- d- Thuật toán mã Rijndael chính là AES.

Câu 10. Một hệ thống gồm 10 thiết bị đầu cuối liên lạc với nhau sử dụng mật mã đối xứng. Mỗi đầu cuối sử dụng các khoá bí mật khác nhau khi kết nối với mỗi đầu cuối khác. Có bao nhiêu khoá bí mật trong toàn bộ hệ thống?

- a- 10 khoá
- b- 20 khoá
- c- 45 khoá
- d- 90 khoá

Câu 11. Thuật toán mật mã nào được dùng trong giao thức xác thực Kerberos 4?

- a- Blowfish
- b- CAST-128
- c- TDES
- d- DES

Câu 12. Ứng dụng của mật mã bất đối xứng:

- a- Bảo mật thông tin
- b- Xác thực thông tin
- c- Bảo vệ tính khả dụng của hệ thống
- d- Câu a và b

Câu 13. Chọn câu đúng về thuật toán mã RSA:

- a- Thuật toán RSA thích hợp cho thực thi bằng phần cứng.
- b- RSA dùng khoá và khối dữ liệu có kích thước cố định.
- c- Mỗi khối thông tin  $n$  bit đưa vào thuật toán RSA được xử lý như một số nguyên có giá trị từ 0 đến  $2^n - 1$ .
- d- Chỉ có thể dùng khoá công khai để mã hoá thông tin, không thể mã hoá thông tin bằng khoá bí mật.

Câu 14. So sánh RSA và DES:

- a- RSA có tốc độ thực thi bằng phần mềm cao hơn DES.
- b- RSA an toàn hơn DES.

- c- RSA dựa trên các hàm toán học, còn DES dựa trên các thao tác xử lý bit.
- d- Bằng cách phân tích khoá công khai thì có thể tìm ra khoá bí mật của RSA, trong khi đối với DES, cách duy nhất để tìm khoá là thử lần lượt.

Câu 15. Thuật toán trao đổi khoá Diffie-Hellman:

- a- Là một dạng của mật mã hoá bất đối xứng.
- b- Diffie-Hellman không có chức năng bảo mật dữ liệu mà chỉ dùng để trao đổi khoá bí mật.
- c- Diffie-Hellman có thể bị tấn công Man-In-The-Middle.
- d- Tất cả đều đúng.

Câu 16. Chức năng của các hàm băm (hash function)?

- a- Tạo ra một khối thông tin ngắn cố định từ một khối thông tin gốc lớn hơn.
- b- Mật mã hoá thông tin.
- c- Xác thực nguồn gốc thông tin
- d- Ngăn chặn việc phủ nhận hành vi của chủ thể thông tin

Câu 17. Các thuộc tính của một giải thuật chữ ký số:

- a- Phải xác nhận chính xác người ký và ngày giờ phát sinh chữ ký.
- b- Phải xác thực nội dung thông tin ngay tại thời điểm phát sinh chữ ký
- c- Phải có khả năng cho phép kiểm chứng bởi một người thứ 3 để giải quyết các tranh chấp nếu có.
- d- Tất cả các câu trên.

Câu 18. Các thông tin cần thiết để tạo ra chữ lý số:

- a- Mã băm của thông tin cần chứng thực.
- b- Khoá bí mật của người ký.
- c- Khoá công khai của nhóm
- d- Tất cả các thông tin trên

Câu 19. Chứng thực khóa (certificate) là gì?

- a- Là một sự chứng thực của một thực thể được tin cậy về sự ràng buộc giữa một thực thể thông tin và khoá công khai của thực thể đó.
- b- Chứng thực khóa là một dạng của chữ ký số.
- c- Chứng thực khóa là một ứng dụng để phân phối khoá bí mật trong mật mã đối xứng.
- d- Tất cả đều đúng.

Câu 20. Chọn câu đúng khi nói về PKI:

- a- PKI tạo ra và quản lý các chứng thực khóa.
- b- Chứng thực khóa sau khi đã tạo ra có thể bị huỷ bỏ theo yêu cầu của chủ sở hữu.
- c- CA là thành phần của PKI có chức năng tạo ra chứng thực khóa theo yêu cầu của người sử dụng.
- d- Tất cả đều đúng.

**B- Bài tập.**

- Câu 21. Xác định các bit 1, 16, 33 và 48 tại ngõ ra vòng thứ nhất của thuật toán giải mã DES, biết rằng ciphertext chứa toàn bit 1 và khoá ban đầu cũng là chuỗi bit 1.
- Câu 22. Chứng minh rằng mạch mã hoá của DES cũng đồng thời là mạch giải mã của DES.
- Câu 23. So sánh ma trận IP và ma trận PC-1 của DES. Rút ra kết luận gì từ sự so sánh này?
- Câu 24. Tính 8 word đầu tiên của khoá mở rộng trong thuật toán mã AES nếu biết khoá ban đầu là chuỗi 128 bit 0.
- Câu 25. Thực hiện mã hoá và giải mã dùng thuật toán RSA với các dữ liệu sau đây:
- a-  $p = 3; q = 11, e = 7; M = 5$
  - b-  $p = 5; q = 11, e = 3; M = 9$
  - c-  $p = 7; q = 11, e = 17; M = 8$
  - d-  $p = 11; q = 13, e = 11; M = 7$
  - e-  $p = 17; q = 31, e = 7; M = 2$
- Câu 26. User A và B trao đổi khoá dùng Diffie-Hellman với  $p = 71$  và  $g = 7$ .
- a- Nếu A chọn  $X_a = 5$ , tính  $Y_a$ ?
  - b- Nếu B chọn  $X_b = 12$ , tính  $Y_b$ ?
  - c- Khoá bí mật dùng chung giữa A và B?
- Câu 27. Cài đặt thuật toán mật mã DES bằng C.
- Câu 28. Cài đặt thuật toán mật mã RSA bằng C.

-----❖-----

## CHƯƠNG III

# CÁC ỨNG DỤNG BẢO MẬT TRONG HỆ THỐNG THÔNG TIN

### Giới thiệu:

Các cơ chế mật mã và xác thực thông tin là cơ sở cho việc xây dựng các ứng dụng bảo mật trong hệ thống, đặc biệt trong môi trường mạng. Chương này sẽ trình bày một số ứng dụng của kỹ thuật mật mã và xác thực thông tin trong việc xây dựng các giao thức (protocol) và dịch vụ (service) trên mạng nhằm đảm bảo an toàn hệ thống. Các ứng dụng được trình bày trong chương này bao gồm:

- Giao thức xác thực (Authentication protocol)
- IPSec (Internet Protocol Security)
- SSL (Secure Sockets Layer)
- SET (Secure Electronic Transaction)

## III.1 GIAO THỨC XÁC THỰC

### III.1.1 Mật khẩu:

Trong số các cơ chế xác thực, cơ chế xác thực dựa trên thông tin mà thực thể truy xuất biết (what you know) là cơ chế đơn giản nhất và được sử dụng nhiều nhất. Thông tin này thường là mật khẩu (password), được liên kết với một thực thể dùng để xác thực thực thể đó.

Mật khẩu thường là một chuỗi ký tự. Không gian mật khẩu (password space) là tập hợp tất cả các chuỗi ký tự có thể xuất hiện trong mật khẩu. Mỗi hệ thống xác thực có một không gian mật khẩu khác nhau. Không gian mật khẩu càng lớn thì khả năng bị tấn công mật khẩu theo phương thức brute force càng thấp.

Mật khẩu được gọi là phức tạp nếu nó khó bị phát hiện bằng phương pháp dò mật khẩu theo từ điển (dictionary attack).

Theo khảo sát, những loại mật khẩu được dùng phổ biến nhất hiện nay bao gồm:

-Dùng tên của người sử dụng (user-name hoặc account-name) làm mật khẩu hoặc thêm một vài chữ số (ví dụ ngày sinh, số điện thoại, ...) để làm mật khẩu.

-Dùng tên đăng nhập (logon-name) làm mật khẩu.

-Dùng tên máy tính (computer name) làm mật khẩu.

-Mật khẩu chỉ bao gồm các ký tự số (lấy từ số điện thoại, ngày sinh, ...).

-Mật khẩu là những từ khóa đặc biệt như *computer*, *hacker*, ...

-Lấy một từ có nghĩa trong từ điển làm mật khẩu.

-Lấy tên một người khác làm mật khẩu (thường là người có quan hệ mật thiết)

Những mật khẩu như trên đều có độ phức tạp rất thấp và do đó dễ dàng bị tiết lộ. Các hệ thống xác thực thường đưa ra các chính sách về mật khẩu (password policy) đối với người sử dụng. Các chính sách này thường quy định những ràng buộc sau đây:

-Chiều dài tối thiểu và độ khó của mật khẩu, mật khẩu không được chứa user-name hoặc logon-name (password complexity).

-Thời gian sử dụng tối đa của mật khẩu (password age).

-Không được phép dùng lại mật khẩu cũ (password history).

Về phía người sử dụng, những nguyên tắc chung để tăng độ an toàn cho việc xác thực dùng mật khẩu bao gồm:

-Sử dụng nhiều loại ký tự khác nhau để làm mật khẩu, mục đích là mở rộng không gian mật khẩu (dùng chữ cái, chữ số, các ký hiệu đặc biệt, dùng phối hợp giữa chữ hoa và chữ thường, ...)

-Không sử dụng các mật khẩu quá ngắn.

-Không sử dụng những từ khóa hoặc từ có nghĩa trong mật khẩu.

-Thường xuyên thay đổi mật khẩu.

-Không ghi chép mật khẩu lên bất kỳ vị trí nào.

-Không tiết lộ mật khẩu cho người khác, ngay cả những tình huống an toàn nhất.

Trên các máy chủ xác thực, mật khẩu của người sử dụng thường không được lưu trữ một cách trực tiếp dưới dạng ký tự gốc (cleartext) mà phải được mã hoá dưới một dạng nào đó để đảm bảo an toàn. Ngoài ra, để mật khẩu không bị đánh cắp khi truyền đi trên mạng, nhiều thủ tục xác thực phức tạp được xây dựng để đảm bảo rằng mật khẩu không được truyền đi trực tiếp (cleartext) trên mạng.

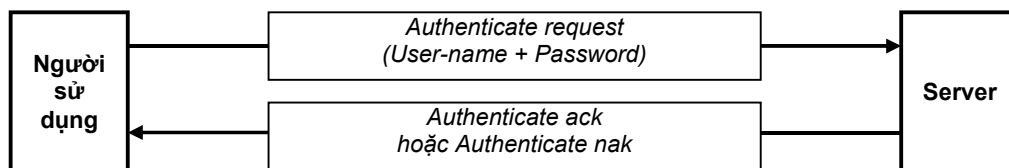
### III.1.2 Xác thực trong mô hình điểm-điểm:

Một thực thể bên ngoài hệ thống thông tin muốn truy xuất hệ thống như một chủ thể của hệ thống thì phải cung cấp các thông tin để hệ thống xác thực nhận dạng của chủ thể. Các thông tin này thường là mật khẩu, thẻ xác thực, dấu vân tay, ... Quá trình xác thực một thực thể bao gồm việc lấy thông tin mà thực thể cung cấp, phân tích và xác định xem thông tin có liên kết với thực thể đó hay không.

Hai mô hình thực tế của một hệ thống xác thực là *xác thực tại chỗ (local authentication)* hoặc *xác thực từ xa (remote authentication)* thông qua môi trường mạng. Mô hình thứ nhất được sử dụng khi người sử dụng đăng nhập trực tiếp vào một thống nội bộ (local login), thông tin xác thực (tên người dùng và mật khẩu) được cung cấp trực tiếp cho hệ thống xác thực (server). Trong mô hình thứ hai, người sử dụng đăng nhập vào một hệ thống ở xa. Tình huống này bắt buộc các thông tin xác thực phải được gửi đi trên mạng và do đó, nguy cơ bị nghe lén thông tin là rất cao. Các giao thức xác thực được thiết kế để giảm thiểu các nguy cơ này.

Trong các hệ thống cổ điển, kết nối từ xa thường được thực hiện thông qua các giao thức điểm – điểm như SLIP (Serial Line Internet Protocol) hoặc PPP (Point to Point Protocol). Các thủ tục xác thực đều là một chiều, tức là chỉ có máy chủ xác thực người sử dụng chứ không có thủ tục ngược lại. Hai giao thức xác thực thường được dùng trong các hệ thống này là PAP (Password Authentication Protocol) và CHAP (Challenge-Handshake Authentication Protocol).

-PAP là giao thức xác thực đơn giản nhất và do đó kém an toàn nhất. Để xác thực với một hệ thống server ở xa, người sử dụng chỉ cần gửi tên đăng nhập và mật khẩu của mình một cách trực tiếp (clear text) cho server dưới dạng một gói yêu cầu xác thực (authenticate request packet). Server sẽ kiểm tra thông tin xác thực chứa trong gói dữ liệu này, nếu trùng với thông tin đã lưu trữ trước đó thì sẽ trả lời bằng một gói xác nhận (authenticate ack packet) và quá trình xác thực xem như thành công. Ngược lại, nếu thông tin xác thực không đúng, server sẽ trả lời bằng gói từ chối (Authenticate nak packet).

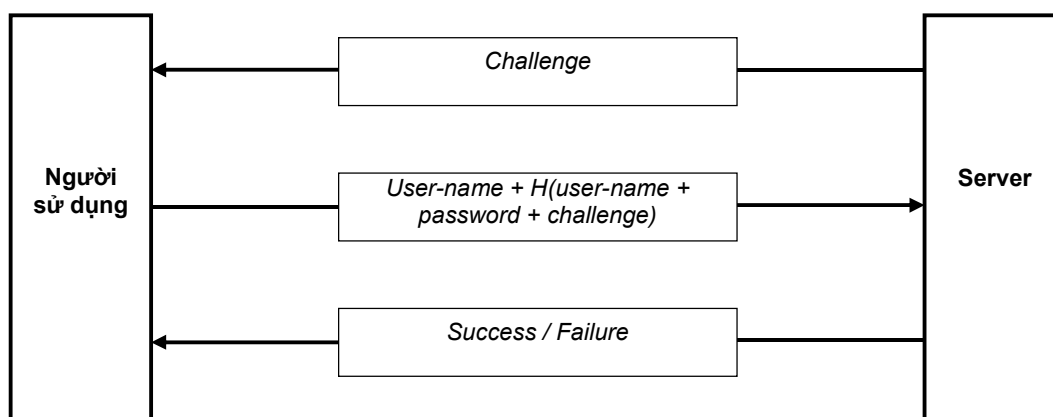


**Hình 3.1:** Giao thức xác thực PAP

-CHAP là giao thức xác thực phức tạp hơn, được dùng trong giao thức kết nối PPP và một số hệ thống khác. CHAP có ưu điểm hơn PAP về phương diện bảo mật là có dùng các hàm băm một chiều và *thông tin xác thực không được gửi đi trực tiếp trên mạng*. Quá trình xác thực bằng giao thức CHAP gồm các bước sau đây (gọi là quá trình challenge-response):

- Sau khi thiết lập xong kết nối PPP, để xác định xem người sử dụng có quyền truy xuất hay không, server sẽ gửi cho người sử dụng một khối dữ liệu *thách thức (challenge)*, trong đó có chứa một giá trị ngẫu nhiên do server tạo ra.
- Người sử dụng sau khi nhận được khối challenge sẽ gắn thêm tên đăng nhập và mật khẩu của mình vào đó, sau đó thực hiện một hàm băm một chiều (ví dụ MD5) lên khối thông tin đó và gửi mã băm lại cho server.
- Phía server cũng thực hiện một quá trình tương tự và so sánh với kết quả nhận được từ người sử dụng để xác định quá trình xác thực có thành công hay không.

Một đặc điểm nữa của giao thức này làm tăng tính an toàn của kết nối là quá trình challenge-response được lặp lại nhiều lần trong suốt thời gian duy trì kết nối. Nếu gói tin trả lời của người dùng không hợp lệ, kết nối sẽ bị giải tỏa tức thời.



**Hình 3.2:** Giao thức xác thực CHAP

### III.1.3 Xác thực trong các hệ thống phân tán:



Trong các hệ thống phân tán, nhiều máy chủ cung cấp dịch vụ được quản lý bởi một trung tâm xác thực duy nhất. Giao thức xác thực trong các hệ thống này phải đảm bảo được 2 yêu cầu cơ bản:

- Đảm bảo an toàn đối với thông tin xác thực (tên đăng nhập và mật khẩu không được truyền đi trực tiếp trên mạng).
- Người dùng chỉ cần đăng nhập một lần cho phiên làm việc nhưng có khả năng sử dụng tất cả các dịch vụ có trong hệ thống.

Ví dụ: Một hệ thống mạng Windows được cấu hình theo mô hình Domain. Trong Domain này có nhiều máy chủ cung cấp dịch vụ khác nhau, gồm dịch vụ in ấn (print server), dịch vụ lưu trữ dữ liệu (file server), dùng chung kết nối Internet (qua proxy server), ... Khi người sử dụng đăng nhập vào hệ thống từ một máy thành viên của Domain, người sử dụng này phải có khả năng truy xuất đến tất cả các dịch vụ trong Domain (tùy theo quyền được cấp) mà không phải nhập lại tên đăng nhập và mật khẩu cho từng dịch vụ. Cơ chế quản lý tập trung này cung cấp sự tiện lợi cho cả người sử dụng lẫn hệ thống.

Một thủ tục xác thực điển hình trong các hệ thống này bao gồm các bước như sau:

- Một người dùng đăng nhập từ một máy con (C) trong hệ thống và yêu cầu truy xuất đến máy chủ V.
- Máy con C yêu cầu người dùng cung cấp tên đăng nhập và mật khẩu rồi sau đó chuyển thông tin này cho trung tâm xác thực AS (Authentication Server).
- Máy chủ AS kiểm tra xem tên đăng nhập và mật khẩu có hợp lệ hay không, đồng thời kiểm tra xem người dùng này có được phép truy xuất các dịch vụ trên máy chủ V hay không.
- Nếu cả hai việc kiểm tra trên đều thành công thì người dùng được phép truy xuất dịch vụ trên máy chủ V. Để làm được việc đó, AS tạo ra một thẻ truy xuất (ticket) chứa các thông tin bao gồm nhận dạng của người dùng, địa chỉ mạng của máy con và nhận dạng của máy chủ V. Thẻ truy xuất này được mã hóa bằng khóa bí mật dùng chung giữa AS và V. Thẻ truy xuất cũng được gửi cho C.
- Bắt đầu từ đó, C có thể yêu cầu các dịch vụ của V bằng cách gửi các bản tin có gắn kèm thẻ truy xuất vừa tạo ra cho V. Máy chủ V sẽ giải mã thẻ truy xuất và chấp nhận cho C truy xuất các dịch vụ của mình.

Quá trình được biểu diễn như sau:

$C \rightarrow AS: ID_C + P_C + ID_V$   
 $AS \rightarrow C: Ticket$   
 $C \rightarrow V: ID_C + Ticket$   
 $Ticket = E([ID_C + AD_C + ID_V], K_V)$

Trong đó:

C: Máy con

AS: máy chủ xác thực (Authentication server).

V: máy chủ cung cấp dịch vụ.

ID<sub>C</sub>: Nhận dạng (tên đăng nhập) của người dùng.

ID<sub>V</sub>: Nhận dạng của máy chủ V.

$P_C$ : Mật khẩu của người dùng.

$AD_C$ : Địa chỉ mạng của máy con.

$K_V$ : Khóa bí mật của máy chủ cung cấp dịch vụ V.

Thủ tục xác thực như trên giải quyết được vấn đề bảo mật bằng cách đưa ra khái niệm thẻ truy xuất (ticket), trong đó các thông tin bí mật được mã hóa trong một bản tin đặc biệt trước khi luân chuyển trên mạng. Tuy nhiên, vẫn còn hai vấn đề chưa được giải quyết:

*1-Nếu người dùng có nhu cầu sử dụng dịch vụ nhiều lần, hoặc sử dụng nhiều dịch vụ khác nhau trên các máy chủ khác nhau, vậy người dùng phải thực hiện thủ tục xác thực nhiều lần, tức là phải nhập mật khẩu nhiều lần?*

*2-Thủ tục xác thực vẫn còn một bước (bước đầu tiên) trong đó thông tin xác thực (mật khẩu) được gửi đi trực tiếp trên mạng mà không mã hóa.*

Thủ tục sau đây sẽ giải quyết hai vấn đề trên:

- Khi người dùng đăng nhập hệ thống:
  - (1)  $C \rightarrow AS: ID_C + ID_{tgs}$
  - (2)  $AS \rightarrow C: E(Ticket_{tgs}, K_C)$
- Khi người dùng truy xuất một loại dịch vụ (per service type):
  - (3)  $C \rightarrow TGS: ID_C + ID_V + Ticket_{tgs}$
  - (4)  $TGS \rightarrow C: Ticket_V$
- Khi người dùng truy xuất một phiên giao dịch cụ thể (per service session):
  - (5)  $C \rightarrow V: ID_C + Ticket_V$

Trong đó:

$$Ticket_{tgs} = E([ID_C + AD_C + ID_{tgs} + TS_1 + Lifetime_1], K_{tgs})$$

$$Ticket_V = E([ID_C + AD_C + ID_V + TS_2 + Lifetime_2], K_V)$$

Trong thủ tục trên, một thành phần mới được thêm vào hệ thống xác thực là **máy chủ cấp thẻ TGS** (Ticket-Granting server).

Khi người dùng xác thực thành công với AS, thay vì cấp thẻ sử dụng dịch vụ trực tiếp cho người dùng, AS chỉ cấp cho người dùng thẻ truy xuất của TGS, có tác dụng như một xác nhận đây là một người dùng hợp lệ. Kể từ đó về sau, mỗi khi người dùng cần truy xuất dịch vụ nào thì chỉ cần gửi thẻ truy xuất và yêu cầu của mình đến TGS để được cấp thẻ truy xuất dịch vụ.

Như vậy, AS chỉ cần cấp thẻ cho người dùng một lần, hay nói cách khác, thẻ có thể dùng lại, cả trong trường hợp người dùng sử dụng dịch vụ nhiều lần hoặc sử dụng nhiều dịch vụ khác nhau mà không cần phải nhập lại mật khẩu.

Thủ tục này được mô tả chi tiết như sau:

- Máy con C yêu cầu một thẻ xác nhận người dùng hợp lệ (Ticket Granting Ticket) bằng cách gửi nhận dạng của người dùng cho AS, trong đó có nhận dạng của TGS.
- AS gửi lại thẻ xác nhận người dùng hợp lệ cho máy con nhưng được mã hóa với khóa là mật khẩu của người dùng ( $K_C$ ). Do đó, nếu người dùng cung cấp đúng mật khẩu thì thẻ này được giải mã thành công, ngược lại, việc xác thực xem như kết thúc không thành công.

Như vậy, mật khẩu của người dùng đã không được gửi đi trực tiếp trên mạng.

Do thẻ này có khả năng dùng lại, nên để quản lý việc tồn tại của nó, trong thẻ được gắn thêm một nhãn thời gian quy định thời gian tồn tại hợp lệ của thẻ.

Để tránh trường hợp thay đổi và giả mạo thẻ, thẻ được mã hóa một lần nữa bằng khóa bí mật của AS và TGS.

- Sau khi đã có thẻ xác nhận người dùng hợp lệ, máy con có thể yêu cầu dịch vụ trên máy chủ V bằng cách yêu cầu thẻ sử dụng dịch vụ (Service-granting Ticket) từ TGS. Thông tin gửi đến cho TGS bao gồm nhận dạng của máy chủ V, thẻ xác nhận người dùng hợp lệ và tên đăng nhập của người dùng.
- TGS giải mã thẻ xác nhận người dùng hợp lệ để kiểm tra, nếu hợp lệ thì cấp thẻ truy xuất dịch vụ cho người dùng. Thẻ này được mã hóa bằng khóa bí mật của V và TGS.
- Sau khi có thẻ truy xuất dịch vụ, người dùng có thể sử dụng dịch vụ trên máy chủ V.

Như vậy, thủ tục trên giải quyết được 2 vấn đề: dùng lại thẻ và không gửi mật khẩu trực tiếp trên mạng.

Tuy nhiên, lại thêm 2 vấn đề khác nảy sinh:

*-Thứ nhất, nếu thời gian tồn tại của các ticket quá ngắn, người dùng có thể phải nhập lại mật khẩu để tạo thẻ mới. Nếu thời gian này quá dài, nguy cơ bị lấy cắp thẻ tăng lên. Do đó, khi xác nhận một thẻ, máy chủ (TGS hoặc V) phải biết chắc rằng mình đang làm việc với đúng người dùng có tên đăng nhập chứa trong thẻ.*

*-Thứ hai, song song với việc người dùng xác thực với máy chủ, thì cũng cần phải có thao tác xác thực ngược lại từ máy chủ đến người dùng để loại trừ trường hợp chính máy chủ bị giả mạo.*

Đây chính là tồn tại được giải quyết bởi giao thức xác thực Kerberos.

### III.1.4 Giao thức xác thực Kerberos:

Kerberos là một thủ tục được xây dựng để nâng cao độ an toàn khi xác thực trong môi trường mạng phân tán. Kerberos dựa trên kỹ thuật mật mã đối xứng (DES).

Có thể tóm lược thủ tục xác thực của Kerberos version 4 như sau (hình 3.3):

1- Máy con yêu cầu AS cung cấp thẻ xác nhận người dùng:

$$C \rightarrow AS: ID_c + ID_{tgs} + TS_1$$

2- AS cung cấp thẻ xác nhận người dùng cho máy con:

$$AS \rightarrow C: E([K_{c,tgs} + ID_{tgs} + TS_2 + Lifetime_2 + Ticket_{tgs}], K_c)$$

$$\text{Với } Ticket_{tgs} = E([K_{c,tgs} + ID_c + AD_c + ID_{tgs} + TS_2 + Lifetime_2], K_{tgs})$$

3- Máy con yêu cầu TS cung cấp thẻ truy xuất dịch vụ:

$$C \rightarrow TGS: ID_v + Ticket_{tgs} + Authenticator_c$$

4- TGS cung cấp thẻ truy xuất dịch vụ cho máy con:

$$TGS \rightarrow C: E([K_{c,v} + ID_v + TS_4 + Ticket_v], K_{c,tgs})$$

$$\text{Với } Ticket_{tgs} = E([K_{c,tgs} + ID_c + AD_c + ID_{tgs} + TS_2 + Lifetime_2], K_{tgs})$$

$$Ticket_v = E([K_{c,v} + ID_c + AD_c + ID_v + TS_4 + Lifetime_4], K_v)$$

$$Authenticator_c = E([ID_C + AD_C + TS_3], K_{c, tgs})$$

5- Máy con yêu cầu dịch vụ:

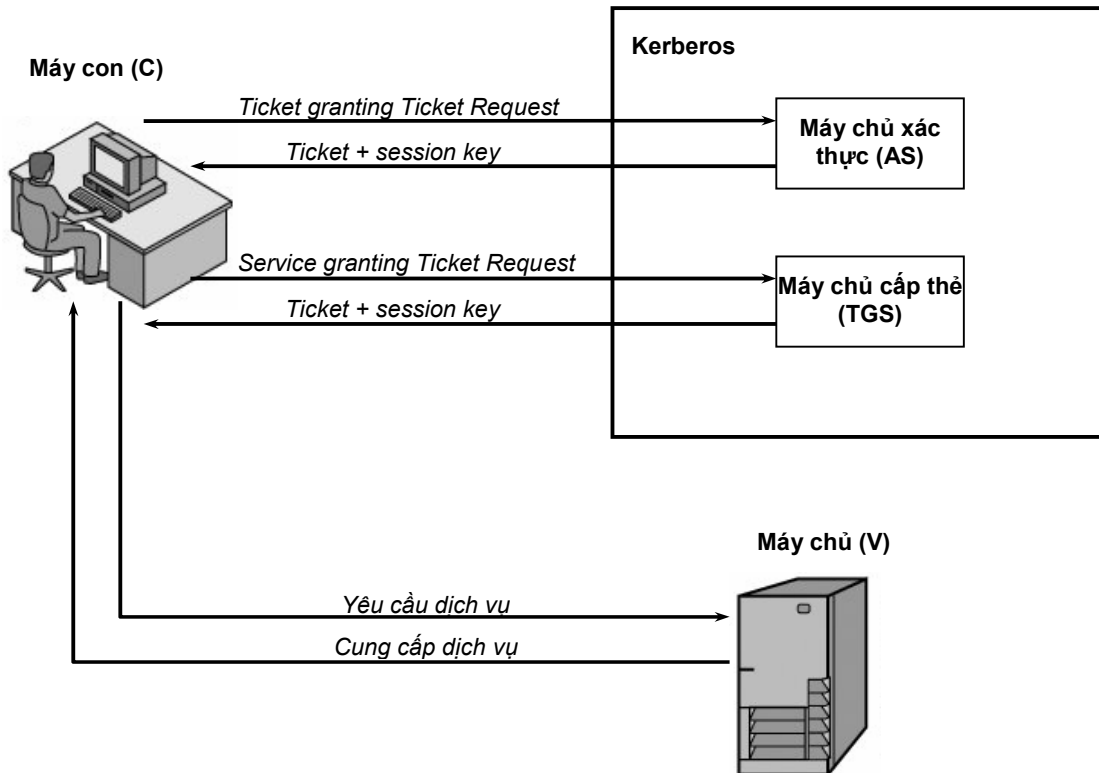
$$C \rightarrow V: Ticket_v + Authenticator_c$$

$$Với \quad Authenticator_c = E([ID_c + AD_c + TS_5], K_{c,v})$$

6- Server xác thực với máy con (không bắt buộc):

$$V \rightarrow C: E([TS_5 + 1], K_{c,v})$$

$$Với: \quad Ticket_v = E([K_{c,v} + ID_c + AD_c + ID_v + TS_4 + Lifetime_4], K_v)$$



**Hình 3.3:** Thủ tục xác thực Kerberos 4

Các thành phần trong các bản tin của Kerberos:

**-Bản tin (1):** Máy con yêu cầu cấp thẻ xác nhận người dùng (Ticket-granting-Ticket):

- $ID_C$ : Nhận diện của người dùng (do máy con gửi đến cho AS, dựa trên thông tin đăng nhập của người dùng).
- $ID_{tgs}$ : Nhận diện của TGS, mục đích cho AS biết rằng máy con đang muốn truy xuất đến TGS.
- $TS_1$ : Nhãn thời gian, mục đích để đồng bộ thời gian giữa AS và máy con.

**-Bản tin (2):** AS cung cấp thẻ xác nhận người dùng cho máy con:

- $K_c$ : Dùng chính mật khẩu của người dùng làm khoá mật mã, vừa có mục đích bảo vệ thông tin vừa cho phép AS xác thực mật khẩu của người dùng. Nếu máy con không có mật khẩu đúng thì sẽ không giải mã được bản tin này.

- $K_{c, tgs}$ : khoá bí mật được dùng giữa máy con và TGS do AS tạo ra. Khóa này chỉ có tác dụng trong một phiên làm việc (session key).
- $ID_{tgs}$ : Nhận diện của TGS, dùng để xác nhận rằng thẻ này có tác dụng cho phép máy con truy xuất đến TGS.
- $TS_2$ : Nhãn thời gian, cho biết thời điểm thẻ được tạo ra.
- $Lifetime_2$ : Cho máy con biết thời gian tồn tại của thẻ.
- $Ticket_{tgs}$ : Máy con dùng thẻ này để truy xuất TGS.

**-Bản tin (3): Máy con yêu cầu thẻ truy xuất dịch vụ (Service Granting Ticket):**

- $ID_V$ : Nhận dạng của máy chủ V, dùng để thông báo cho TGS là máy con muốn truy xuất đến dịch vụ của máy chủ V.
- $Ticket_{tgs}$ : Thẻ được cấp cho máy con bởi AS
- $Authenticator_c$ : một giá trị được tạo ra bởi máy con để xác minh thẻ.

**-Bản tin (4): TGS cung cấp thẻ truy xuất dịch vụ cho máy con::**

- $K_{c, tgs}$ : Khoá bí mật dùng chung giữa máy con và TGS để bảo vệ nội dung của bản tin (4)
- $K_{c, v}$ : Khoá bí mật được dùng giữa máy con và máy chủ V do TGS tạo ra. Khóa này chỉ có giá trị trong từng phiên làm việc (session key).
- $ID_V$ : nhận diện của máy chủ V, có chức năng xác nhận thẻ của máy chủ V
- $TS_4$ : nhãn thời gian cho biết thời điểm thẻ được tạo ra.
- $Ticket_v$ : Thẻ được máy con dùng để truy xuất máy chủ V.
- $Ticket_{tgs}$ : Thẻ này được dùng lại để người dùng không phải nhập lại mật khẩu khi muốn truy xuất dịch vụ khác.
- $K_{tgs}$ : Khóa bí mật dùng chung giữa AS và TGS.
- $K_{c, tgs}$ : Session key được TGS dùng để giải mã authenticator. Khóa này được dùng chung giữa máy con và TGS.
- $ID_C$ : Nhận diện máy con, cho biết đây là chủ sở hữu của thẻ.
- $AD_C$ : Địa chỉ mạng của máy con, dùng để ngăn chặn trường hợp một máy khác lấy cắp thẻ để yêu cầu dịch vụ.
- $ID_{tgs}$ : nhận diện của TGS, để xác nhận thẻ đã được giải mã thành công.
- $TS_2$ : nhãn thời gian cho biết thời điểm tạo ra thẻ.
- $Lifetime_2$ : Thời gian tồn tại của thẻ, nhằm ngăn chặn việc sử dụng lại thẻ (replay).
- $Authenticator_c$ : Thông tin xác thực của máy con.
- $K_{c, tgs}$ : Khoá bí mật dùng chung giữa máy con và TGS, dùng để mã hoá thông tin xác thực của máy con.
- $ID_c$ : nhận dạng máy con, phải trùng với ID trong thẻ.
- $AD_c$ : địa chỉ mạng của máy con, phải trùng với địa chỉ trong thẻ.
- $TS_3$ : Nhãn thời gian, cho biết thời điểm mà authenticator được tạo ra.

**-Bản tin (5): Máy con yêu cầu truy xuất dịch vụ:**

- $Ticket_v$ : Thẻ cho biết máycon đã được xác thực bởi AS.

- Authenticator<sub>c</sub>: Thông tin xác thực thẻ của máy con.

**-Bản tin (6): Máy chủ V xác thực với máy con:**

- K<sub>c, v</sub>: Khoá bí mật dùng chung giữa máy con và máy chủ V.
- TS<sub>5</sub> + 1: Nhãn thời gian, dùng để tránh trường hợp thông tin xác thực cũ được dùng lại.
- Ticket<sub>v</sub>: Thẻ truy xuất máy chủ V. Thẻ này có thể dùng lại khi máy con truy xuất dịch vụ đến chính máy chủ V mà không cần yêu cầu cấp thẻ mới.
- K<sub>v</sub>: Khoá bí mật dùng chung giữa TGS và máy chủ V.
- K<sub>c, v</sub>: Khoá bí mật dùng chung giữa máy con và máy chủ V, dùng để giải mã thông tin xác thực.
- ID<sub>c</sub>: Nhận dạng của máy con.
- AD<sub>c</sub>: Địa chỉ mạng của máy con.
- ID<sub>v</sub>: Nhận dạng của máy chủ V.
- TS<sub>4</sub>: Nhãn thời gian cho biết thời điểm thẻ được tạo.
- Lifetime<sub>4</sub>: Thời gian tồn tại của thẻ.
- Authenticator<sub>c</sub>: Thông tin xác thực của máy con.
- K<sub>c, v</sub>: Khoá bí mật, dùng chung giữa máy con và máy chủ V để mã hoá thông tin xác thực.
- ID<sub>c</sub>: Nhận diện máy con, phải giống với ID<sub>c</sub> trong thẻ
- AD<sub>c</sub>: Địa chỉ mạng của máy con, phải giống với địa chỉ trong thẻ.
- TS<sub>5</sub>: Thời điểm thông tin xác thực được tạo ra.

**-Kết hợp giữa nhiều hệ thống Kerberos:**

Một môi trường sử dụng hệ thống xác thực Kerberos đầy đủ bao gồm máy chủ Kerberos (Kerberos server), các máy chủ dịch vụ (application server) và các máy con sử dụng dịch vụ (client), trong đó:

*-Máy chủ Kerberos phải có danh sách tất cả các tên đăng nhập và mật khẩu đã được mã hóa của các người dùng này. Tất cả các máy con đều phải đăng ký với máy chủ Kerberos.*

*-Máy chủ Kerberos sử dụng một khóa bí mật chung với các máy chủ còn lại. Tất cả các máy chủ đều phải đăng ký với máy chủ Kerberos.*

Một môi trường thỏa mãn các điều kiện như vậy được gọi là một lãnh địa Kerberos (Kerberos realm).

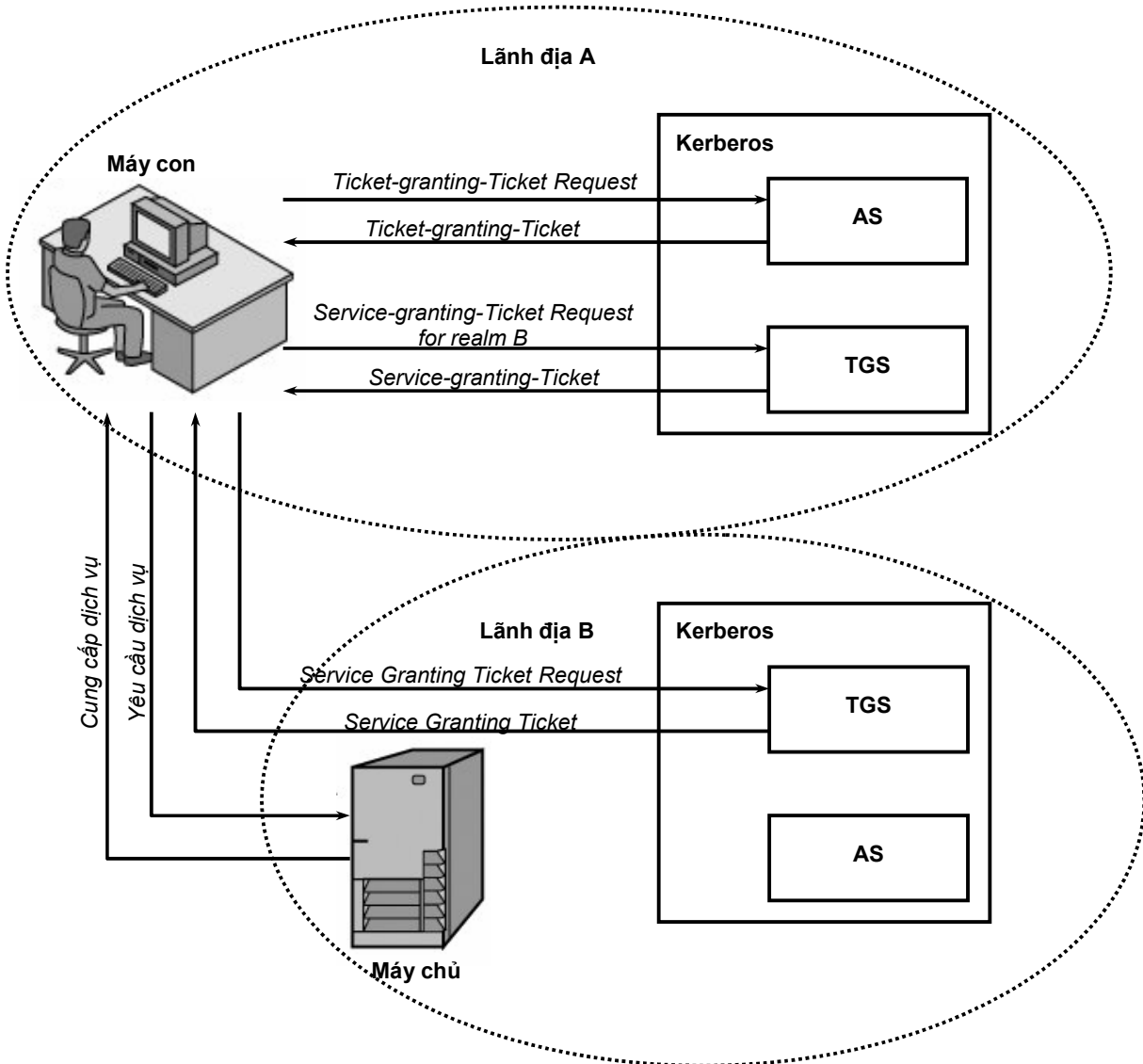
Như vậy, các máy chủ và máy con thuộc các đơn vị quản lý khác nhau sẽ thuộc về các lãnh địa Kerberos khác nhau. Giao thức xác thực Kerberos cũng bao gồm các thủ tục cho phép kết hợp các lãnh địa Kerberos lại để cung cấp dịch vụ một cách đồng nhất. Hình 3.4 mô tả hoạt động của thủ tục này.

Trình tự của thủ tục kết hợp lãnh địa Kerberos được tóm tắt như sau:

(1) C → AS: ID<sub>c</sub> + ID<sub>tgs</sub> + TS<sub>1</sub>

(2) AS → C: E([K<sub>c,tgs</sub> + ID<sub>tgs</sub> + TS<sub>2</sub> + Lifetime<sub>2</sub> + Ticket<sub>tgs</sub>], K<sub>c</sub>)

- (3)  $C \rightarrow TGS: ID_{tgsrem} + Ticket_{tgs} + Authenticator_c$
- (4)  $TGS \rightarrow C: E([K_{c,tgsrem} + ID_{tgsrem} + TS_4 + Ticket_{tgsrem}], K_{c,tgs})$
- (5)  $C \rightarrow TGS_{rem}: ID_{vrem} + Ticket_{tgsrem} + Authenticator_c$
- (6)  $TGS_{rem} \rightarrow C: E([K_{c,vrem} + ID_{vrem} + TS_6 + Ticket_{vrem}], K_{c,tgsrem})$
- (7)  $C \rightarrow V_{rem}: Ticket_{vrem} + Authenticator_c$



**Hình 3.4:** Xác thực giữa hai lãnh địa Kerberos

**-Kerberos 5:** là một phiên bản nâng cấp của Kerberos 4 với những điểm khác biệt như sau:

- Kerberos 4 phụ thuộc chặt chẽ vào giải thuật mã hóa đối xứng DES, trong khi Kerberos 5 thì tương thích với bất kỳ một giải thuật mã hóa nào.

- Kerberos 4 phụ thuộc vào địa chỉ IP để xác thực người dùng, Kerberos 5 có thể sử dụng bất kỳ địa chỉ nào (ví dụ MAC address).
- Kerberos 4 sử dụng thêm 1 byte trong các bản tin để biết thứ tự byte trong bản tin. Kerberos 5 dùng cú pháp ANS.1 (Abstract Syntax Notation One) và luật mã hóa cơ bản BER (Basic Coding Rules) để tạo ra cơ chế xếp thứ tự byte trong bản tin một cách rõ ràng.
- Thời gian tồn tại của thẻ trong Kerberos 4 được chứa trong một trường dài 8 bit, tính theo đơn vị 5 phút, như vậy, thời gian sống tối đa của thẻ là  $5 * 2^8 = 1280$  phút (khoảng 21 giờ). Trong Kerberos 5, thời gian tồn tại được biểu thị bằng thời điểm bắt đầu và thời điểm kết thúc, cho phép thời gian này được biến thiên không giới hạn.
- Kerberos 4 không cho phép cơ chế chuyển tiếp xác thực, tức là cơ chế một máy con truy xuất đến một máy chủ, và yêu cầu máy chủ này truy xuất đến dịch vụ của một máy chủ khác thông qua nhận dạng của máy con. Kerberos 5 cung cấp khả năng này.
- Kerberos 4 yêu cầu  $N^2$  quan hệ giữa các lãnh địa Kerberos trong trường hợp liên kết hoạt động giữa N lãnh địa. Kerberos 5 yêu cầu số quan hệ ít hơn nhiều.

Thủ tục xác thực dùng Kerberos 5 được tóm tắt như sau:

- (1)  $C \rightarrow AS: Options + ID_c + Realm_c + ID_{tgs} + Times + Nonce_I$
- (2)  $AS \rightarrow C: Realm_c + ID_C + Ticket_{tgs} + E([K_{c,tgs} + Times + Nonce_I + Realm_{tgs} + ID_{tgs}], K_c)$   
 Với  $Ticket_{tgs} = E([Flags + K_{c,tgs} + Realm_c + ID_c + AD_c + Times], K_{tgs})$
- (3)  $C \rightarrow TGS: Options + ID_v + Times + Nonce_2 + Ticket_{tgs} + Authenticator_c$
- (4)  $TGS \rightarrow C: Realm_c + ID_c + Ticket_v + E([K_{c,v} + Times + Nonce_2 + Realm_v + ID_v], K_{c,tgs})$   
 Với  $Ticket_{tgs} = E([Flags + K_{c,tgs} + Realm_c + ID_C + AD_C + Times], K_{tgs})$   
 $Ticket_v = E([Flags + K_{c,v} + Realm_c + ID_C + AD_c + Times], K_v)$   
 $Authenticator_c = E([ID_C + Realm_c + TS_1], K_{c,tgs})$
- (5)  $C \rightarrow V: Options + Ticket_v + Authenticator_c$
- (6)  $V \rightarrow C: E([TS_2 + Subkey + Seq\#], K_{c,v})$   
 Với  $Ticket_v = E([Flags + K_{c,v} + Realm_c + ID_C + AD_C + Times], K_v)$   
 $Authenticator_c = E([ID_C + Realm_c + TS_2 + Subkey + Seq\#], K_{c,v})$

Trong thủ tục trên, ngoài những thành phần đã xuất hiện trong Kerberos 4 còn có thêm các thành phần mới sau đây:

-*Realm*: Biểu thị lãnh địa của người dùng.

-*Options*: Các tùy chọn, dùng để yêu cầu các thông tin cộng thêm xuất hiện trong thẻ.

-*Times*: Dùng để máy con yêu cầu các thông số thời gian trong thẻ như from, till, rtime.

-*Nonce*: Giá trị ngẫu nhiên được tạo ra trong bản tin để đảm bảo rằng bản tin trả lời là bản tin hợp lệ chứ không phải bản tin cũ dùng lại.



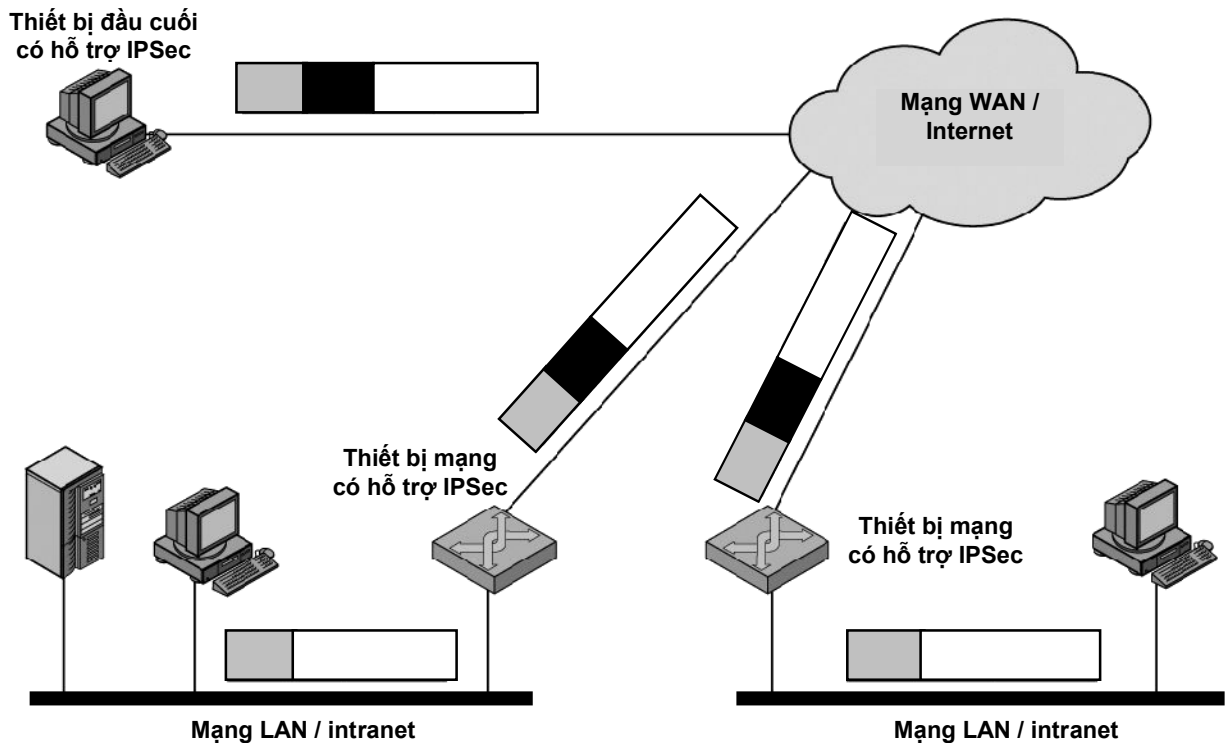
## III.2 IP SECURITY

### III.2.1 Các ứng dụng và đặc điểm của IPSec:


IP security (IPSec) cung cấp một phương tiện truyền thông an toàn trên mạng LAN, giữa các mạng LAN nối với nhau thông qua mạng WAN và giữa các mạng khác nhau trên mạng Internet. IPSec là phần mở rộng của giao thức IP, được thực hiện thống nhất trong cả hai phiên bản của IP và IPv4 và IPv6.


**-Các ứng dụng điển hình của IPSec bao gồm:**


- *Kết nối giữa các chi nhánh của một tổ chức thông qua mạng Internet:* bằng cách xây dựng các mạng riêng ảo VPN (Virtual Private Network) trên nền của mạng WAN công cộng hoặc mạng Internet. Các tổ chức có thể kết nối các mạng con ở các chi nhánh của mình lại thành một mạng riêng với chi phí thấp nhưng vẫn đảm bảo được độ an toàn.
- *Truy xuất từ xa thông qua mạng Internet:* Để truy xuất từ xa đến một dịch vụ nào đó, thông thường người dùng phải thực hiện kết nối bằng đường dây điện thoại (dial-up) đến máy chủ cung cấp dịch vụ. Với IPSec, người dùng chỉ cần kết nối đến một nhà cung cấp dịch vụ Internet gần nhất (ISP) và sau đó thực hiện kết nối đến máy chủ ở xa thông qua IPSec một cách an toàn mà không phải tốn chi phí điện thoại đường dài.
- *Nâng cao tính an toàn của các giao dịch thương mại trên mạng Internet,* áp dụng cho các website bán hàng qua mạng hoặc các dịch vụ thanh toán qua Internet.



Các thành phần của gói dữ liệu:

 Tiêu đề IP  
(IP header)

 Tiêu đề IPSec  
(IPSec header)

 Dữ liệu của gói IP  
(IP Payload)

Hình 3.5: Ứng dụng của IPSec

### ***-Các ưu điểm của IPSec:***

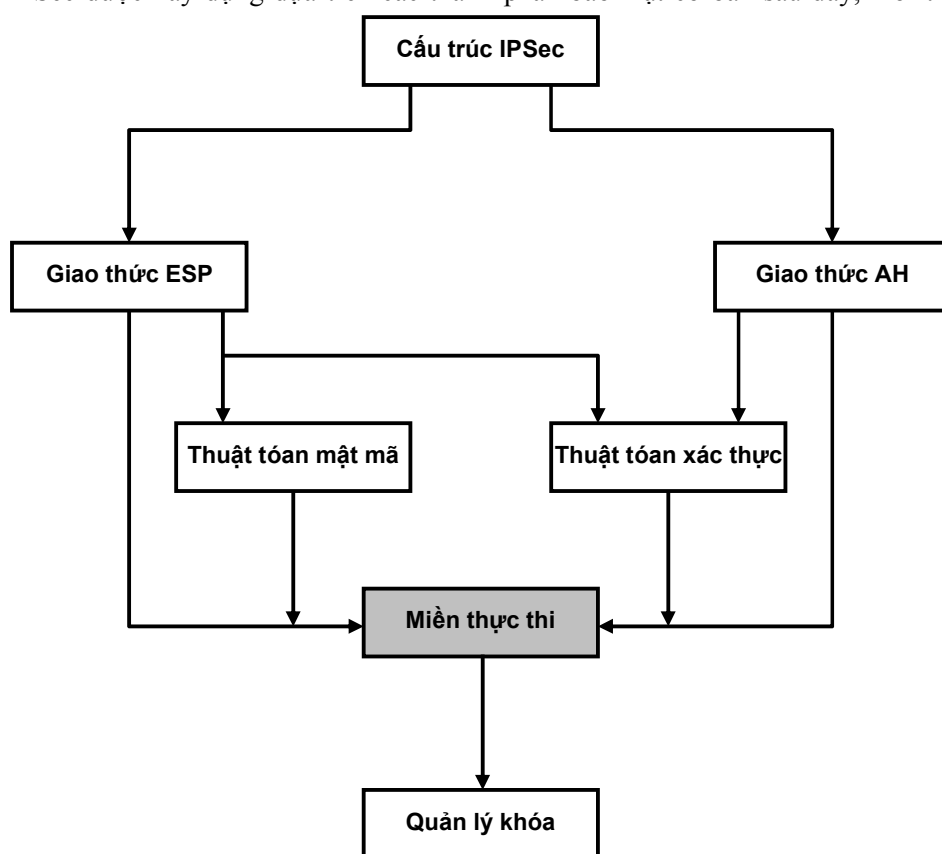
-Khi IPSec được triển khai trên bức tường lửa hoặc bộ định tuyến của một mạng riêng, thì tính năng an toàn của IPSec có thể áp dụng cho toàn bộ lưu lượng vào ra mạng riêng đó mà các thành phần khác không cần phải xử lý thêm các công việc liên quan đến bảo mật.

-IPSec được thực hiện bên dưới của lớp TCP và UDP, đồng thời nó hoạt động một cách trong suốt với các lớp này. Do vậy, không cần phải thay đổi phần mềm hay cấu hình lại các dịch vụ khi IPSec được triển khai.

-IPSec có thể được cấu hình để hoạt động một cách trong suốt đối với các ứng dụng đầu cuối, điều này giúp che giấu những chi tiết cấu hình phức tạp mà người dùng phải thực hiện khi kết nối đến mạng nội bộ từ xa thông qua mạng Internet.

### **III.2.2 Cấu trúc IPSec:**

IPSec được xây dựng dựa trên các thành phần bảo mật cơ bản sau đây, mỗi thành phần



**Hình 3.6:** *Cấu trúc IPSec*

được định nghĩa trong một tài liệu riêng tương ứng (hình 3.6):

*-Cấu trúc (Architecture):* Quy định cấu trúc, các khái niệm và yêu cầu của IPSec.

*-Giao thức ESP:* Mô tả giao thức ESP, là một giao thức mật mã và xác thực thông tin trong IPSec.

*-Giao thức AH:* Định nghĩa một giao thức khác với chức năng gần giống ESP. Nhưng vậy, khi triển khai IPSec, người sử dụng có thể chọn dùng ESP hoặc AH. Mỗi giao thức có ưu và nhược điểm riêng, sẽ được trình bày trong phần này.

-*Thuật toán mật mã*: Định nghĩa các thuật toán mã hóa và giải mã sử dụng trong IPSec. IPSec dựa chủ yếu vào các giải thuật mã hóa đối xứng.

-*Thuật toán xác thực*: Định nghĩa các thuật toán xác thực thông tin sử dụng trong AH và ESP.

-*Quản lý khóa*: Mô tả các cơ chế quản lý và phân phối khóa trong IPSec.

-*Miền thực thi (Domain of Interpretation\_DOI)*: Định nghĩa môi trường thực thi IPSec. Như đã trình bày, IPSec không phải là một công nghệ riêng biệt mà sự tổ hợp của nhiều cơ chế, giao thức và kỹ thuật khác nhau, trong đó mỗi cơ chế, giao thức đều có nhiều chế độ hoạt động khác nhau. Việc xác định một tập các chế độ cần thiết để triển khai IPSec trong một tình huống cụ thể là chức năng của miền thực thi.

### III.2.3 Quan hệ bảo mật:

Mục tiêu của IPSec là cung cấp một cơ chế truyền an toàn đảm bảo tính toàn vẹn và xác thực của dữ liệu. Trong môi trường IPSec, một khái niệm quan trọng được dùng để diễn tả *một quan hệ truyền thông bảo mật giữa một đầu gửi và một đầu nhận* đó là **quan hệ bảo mật** (SA\_Security Association). Mỗi SA được xem như một liên kết một chiều giữa hai thực thể, do đó, một kết nối hai chiều thường thấy sẽ bao gồm 2 SA. Mỗi SA sử dụng một giao thức xác thực nhất định (AH hoặc ESP) chứ không thể sử dụng đồng thời cả hai.

Mỗi SA được nhận dạng bởi 3 thông số sau đây:

-*Security Parameters Index (SPI)*: là một chuỗi bit được gán cho SA, chỉ có giá trị nội bộ. SPI được đặt trong tiêu đề của AH và ESP, cho phép phía nhận (receiving system) chọn một SA cụ thể để xử lý các gói dữ liệu nhận được.

-*IP Destination Address*: Đây là địa chỉ đầu cuối của SA, địa chỉ này là địa chỉ của thiết bị mà SA kết thúc tại đó, có thể là địa chỉ của một hệ thống đầu cuối hoặc của một thiết bị mạng (router, firewall)

-*Security Protocol Identifier*: Cho biết SA sử dụng giao thức xác thực nào (AH hay ESP).

Như vậy, trong mỗi gói IP của IPSec, SA được nhận dạng bằng tổ hợp gồm địa chỉ đích (destination address) và SPI trong tiêu đề mở rộng (AH hoặc ESP).

### III.2.4 Chế độ vận chuyển và chế độ đường hầm:

IPSec (cả AH và ESP) cung cấp hai chế độ làm việc khác nhau:

-*Chế độ vận chuyển (transport mode)*: cung cấp cơ chế bảo vệ cho dữ liệu của các lớp cao hơn (TCP, UDP hoặc ICMP). Ở cơ chế này, phần dữ liệu (payload) của gói IP được áp dụng các cơ chế bảo vệ (mật mã hoặc xác thực). Chế độ này thường dùng cho các kết nối từ đầu cuối đến đầu cuối, ví dụ từ trạm làm việc đến máy chủ hoặc giữa hai trạm làm việc với nhau.

-*Chế độ đường hầm (tunnel mode)*: cung cấp cơ chế bảo vệ ở lớp IP, nghĩa là gói IP cùng với các tiêu đề của AH hoặc ESP được gói thêm một lần nữa bằng các tiêu đề mới. Khi đó, các gói IP gốc được xem như di chuyển trong một đường hầm (tunnel) từ đầu này đến đầu kia của mạng mà các nút trung gian không xen vào được. Chế độ này thường được dùng trong các SA nối giữa hai gateway của hai mạng.

Chế độ vận chuyển và chế độ đường hầm sẽ được trình bày riêng trong từng giao thức AH và ESP.

Các thuật toán mã hóa / giải mã và các thuật toán xác thực thông tin đã được trình bày ở chương 2, nên trong phần này chỉ tập trung mô tả hoạt động của hai giao thức AH và ESP, sau đó giới thiệu các cơ chế quản lý khóa của IPSec.

### III.2.5 AH:

AH (Authentication Header) là một giao thức xác thực dùng trong IPSec, có chức năng đảm bảo tính toàn vẹn của dữ liệu chuyển đi trên mạng IP. AH cho phép xác thực người dùng, xác thực ứng dụng và thực hiện các cơ chế lọc gói tương ứng. Ngoài ra, AH còn có khả năng hạn chế các tấn công giả danh (spoofing) và tấn công phát lại (replay).

Cơ chế hoạt động của AH dựa trên mã xác thực MAC (Message Authentication Code), do đó, để thực thi AH thì hai đầu cuối của SA phải dùng chung một khóa bí mật.

Cấu trúc tiêu đề của gói AH (hình 3.7) bao gồm các phần sau:

Bit 0	8	16	31
Tiêu đề kế tiếp	Kích thước dữ liệu	Dành riêng	
Security Parameters Index (SPI)			
Số thứ tự gói			
Mã xác thực (Kích thước thay đổi)			

**Hình 3.7:** Cấu trúc gói AH

-*Tiêu đề kế tiếp (Next Header - 8 bits):* Nhận dạng kiểu tiêu đề đi liền sau tiêu đề của AH.

-*Kích thước dữ liệu (Payload Length - 8 bits):* Chiều dài của gói AH, tính bằng đơn vị 32 bit trừ đi 2. Ví dụ, chiều dài phần dữ liệu xác thực là 96 bit ( $= 3 * 32$  bit), cộng với chiều dài phần tiêu đề AH (cố định) là  $3 * 32$  bit nữa thành  $6 * 32$  bit, khi đó giá trị của trường kích thước dữ liệu là 4.

-*Dành riêng (Reserved - 16 bits):* Phần dành riêng, chưa dùng.

-*Security Parameters Index (SPI - 32 bits):* Nhận dạng SA như đã trình bày ở trên.

-*Số thứ tự gói (Sequence Number - 32 bits):* Số thứ tự.

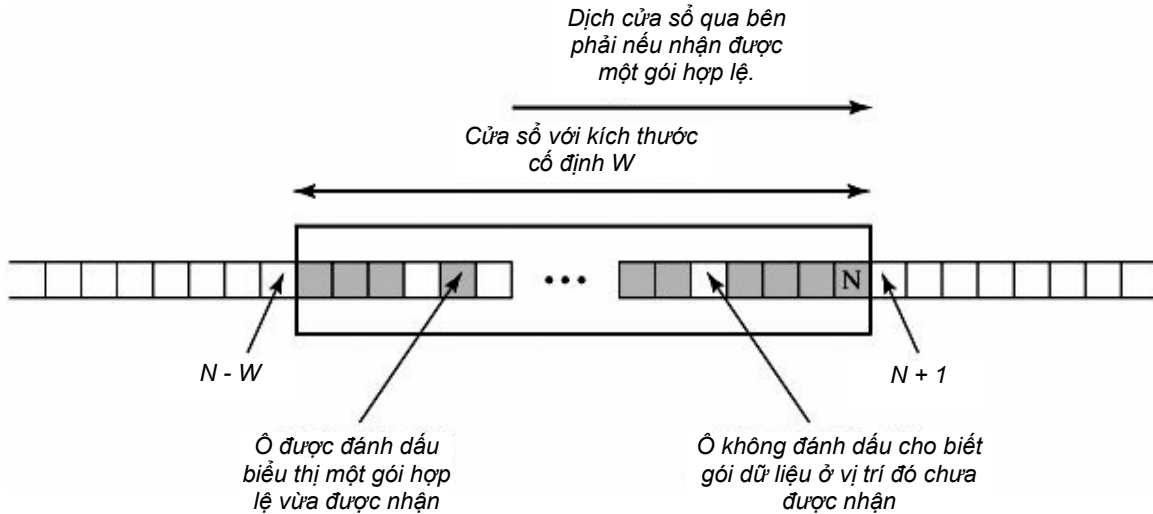
-*Mã xác thực (Authentication Data):* dữ liệu xác thực, có chiều dài thay đổi nhưng phải là bội số của 32 bit. Trường này chứa giá trị kiểm tra ICV (Integrity Check Value) hoặc MAC (Message Authentication Code) cho toàn bộ gói

\*-*Anti-replay service:* dịch vụ cho phép ngăn chặn các hành vi tấn công phát lại (replay) như đã trình bày ở chương 1. Trường số thứ tự (Sequence number) trong tiêu đề AH được dùng để đánh dấu thứ tự các gói được gửi đi trên một SA. Ban đầu, giá trị này được khởi tạo bằng 0 và tăng dần sau mỗi gói được gửi. Để đảm bảo không có gói lặp lại, khi số thứ tự đạt giá trị cực đại

( $2^{32}-1$ ), nó sẽ không được quay lại giá trị 0 mà thay vào đó, một SA mới được thiết lập để tiếp tục việc truyền dữ liệu.

Ở phía nhận, quá trình xử lý các gói nhận được thực hiện theo cơ chế dịch cửa sổ như mô tả ở hình 3.8. Kích thước mặc định của cửa sổ là 64. Cơ chế thực hiện như sau:

- Nếu gói nhận được nằm trong vùng hợp lệ của cửa sổ và là một gói mới chứ không phải gói truyền lại thì giá trị MAC của gói đó sẽ được kiểm tra. Nếu chính



**Hình 3.8:** Cơ chế dịch cửa sổ trong AH

xác (tức gói đã được xác thực) thì khe tương ứng trong cửa sổ được đánh dấu.

- Nếu gói nhận được nằm bên phải của cửa sổ và là gói mới, giá trị MAC của gói được kiểm tra. Nếu đúng thì cửa sổ được dịch một khe sang bên phải, đồng thời khe tương ứng trong cửa sổ được đánh dấu.
- Nếu gói nhận được nằm bên trái cửa sổ hoặc giá trị MAC không hợp lệ thì bị hủy bỏ.

#### **-Xác thực thông tin:**

Mã xác thực (trường Authentication Data) được tạo ra dùng một trong 2 cách:

-HMAC-MD5-96: dùng phương pháp HMAC, thuật toán MD5, cắt lấy 96 bit đầu tiên.

-HMAC-SHA-1-96: dùng phương pháp HMAC, thuật toán SHA-1, cắt lấy 96 bit đầu tiên.

Thuật toán MAC được áp dụng trên các phần thông tin sau đây:

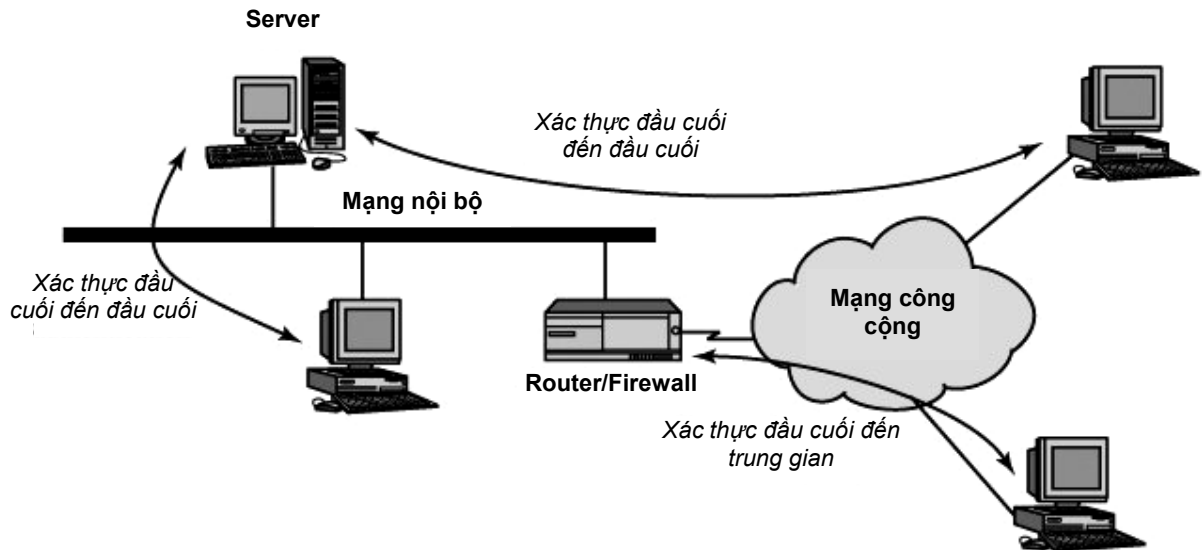
- Các trường không bị thay đổi trong tiêu đề gói IP khi được chuyển tiếp trên mạng hoặc có thể dự đoán được tại đầu cuối của SA. Những trường còn lại trong tiêu đề gói IP được thay bằng các bit 0 khi tính toán.
- Các trường trong tiêu đề AH ngoại trừ trường Authentication Data. Trường này được thay bằng các bit 0 khi tính.
- Toàn bộ gói dữ liệu của lớp trên (tức phần payload của gói IP).

#### **-Chế độ vận chuyển và chế độ đường hầm:**

Hình 3.9 mô tả hai trường hợp xác thực khác nhau:

-*Xác thực từ đầu cuối đến đầu cuối (End-to-End Authentication)*: là trường hợp xác thực trực tiếp giữa hai hệ thống đầu cuối (giữa máy chủ với trạm làm việc hoặc giữa hai trạm làm việc), việc xác thực này có thể diễn ra trên cùng mạng nội bộ hoặc giữa hai mạng khác nhau, chỉ cần 2 đầu cuối biết được khóa bí mật của nhau. Trường hợp này sử dụng chế độ transport của AH.

-*Xác thực từ đầu cuối đến trung gian (End-to-Intermediate Authentication)*: là trường hợp xác thực giữa một hệ thống đầu cuối với một thiết bị trung gian (router hoặc firewall). Trường



**Hình 3.9:** Hai chế độ xác thực của AH

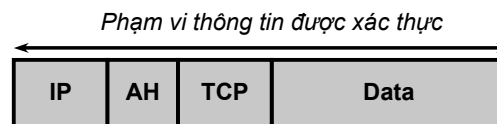
hợp này sử dụng chế độ tunnel của AH.

Hình 3.10 mô tả phạm vi áp dụng cơ chế bảo vệ của AH lên gói dữ liệu trong hai chế độ khác nhau.

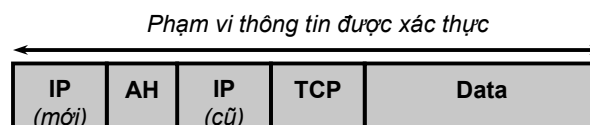
a- Gói IP gốc



b- Gói IP ở chế độ transport



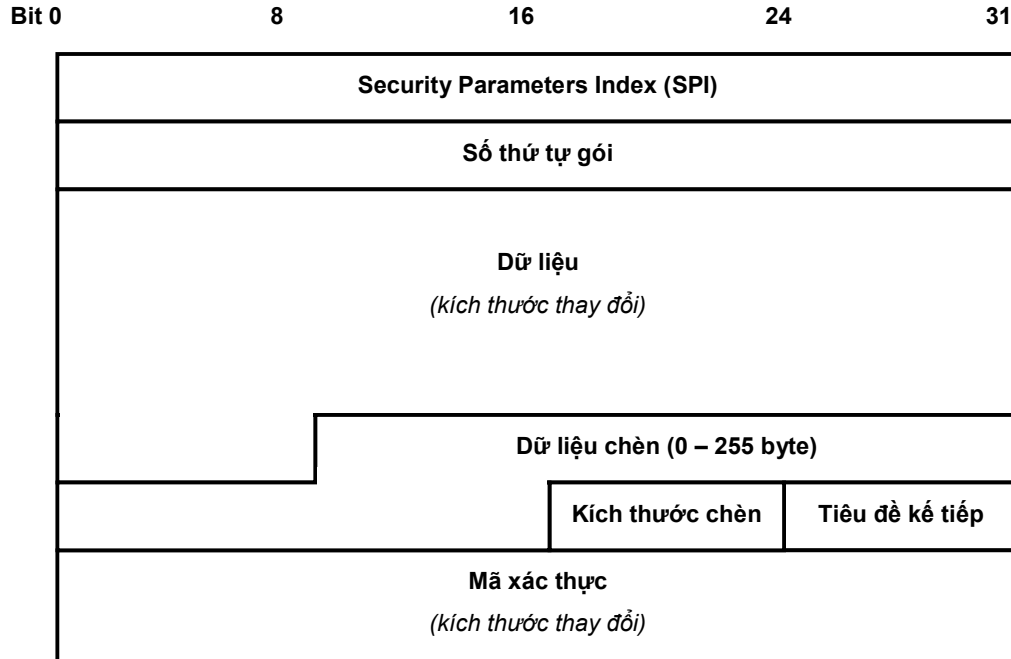
b- Gói IP ở chế độ tunnel



**Hình 3.10:** Phạm vi áp dụng của AH lên gói dữ liệu ở hai chế độ transport và tunnel

### III.2.6 ESP:

ESP (*Encapsulating Security Payload*) là một lựa chọn khác để thực thi IPsec bên cạnh giao thức xác thực thông tin AH. Chức năng chính của ESP là cung cấp tính bảo mật cho dữ liệu truyền trên mạng IP bằng các kỹ thuật mật mã. Tuy nhiên, ESP cũng còn có một tùy chọn khác là cung cấp cả dịch vụ bảo đảm tính toàn vẹn của dữ liệu thông qua cơ chế xác thực. Như vậy, khi



**Hình 3. 11:** Cấu trúc gói ESP

dùng ESP, người dùng có thể chọn hoặc không chọn chức năng xác thực, còn chức năng mã hóa là chức năng mặc định của ESP.

Gói dữ liệu ESP gồm các thành phần sau (hình 3.11):

-*Security Parameters Index (SPI - 32 bits)*: Nhận dạng SA như trong giao thức AH.

-*Số thứ tự gói (Sequence Number - 32 bits)*: Số thứ tự, có chức năng như số thứ tự trong AH.

-*Dữ liệu (Payload Data)*: Đây là phần dữ liệu được bảo vệ bằng mật mã. Trường này có độ dài thay đổi. Trong chế độ vận chuyển, đây là toàn bộ gói dữ liệu của lớp 4 (TCP hoặc UDP). Còn trong chế độ đường hầm, đây là toàn bộ gói IP. ESP chuẩn sử dụng thuật toán mật mã đối xứng DES, tuy nhiên, có thể dùng các thuật toán mật mã khác như 3DES (3 khóa), RC5, IDEA, triple IDEA (3 khóa), CAST, Blowfish.

-*Dữ liệu chèn (Padding 0-255 bytes)*: Một số thuật toán mật mã yêu cầu kích thước dữ liệu gốc phải cố định. Các byte dữ liệu giả được thêm vào để đảm bảo độ dài vùng dữ liệu. Tuy nhiên, theo quy định của ESP, chiều dài trường pad-length và trường next-header phải cố định là 32 bit tính từ bên phải, do vậy, phần padding phải có kích thước sao cho toàn bộ phần thông tin cần mã hóa là bội số của 32 bit.

- *Kích thước chèn (Pad Length - 8 bits)*: Cho biết số byte của vùng dữ liệu chèn (padding).

- *Tiêu đề kế tiếp (Next Header - 8 bits)*: Nhận dạng kiểu dữ liệu chứa trong phần payload data.

- *Mã xác thực (Authentication Data)*: Chứa thông tin xác thực, có chiều dài thay đổi nhưng phải là bội số của 32 bit. Thông tin xác thực được tính trên toàn gói ESP ngoại trừ phần Authentication Data.

**-Chế độ vận chuyển và chế độ đường hầm:**

*Chế độ vận chuyển*: chức năng mã hóa và xác thực thông tin được thực hiện trên phần dữ liệu (payload data) của gói IP (tức toàn bộ đơn vị dữ liệu của lớp trên IP).

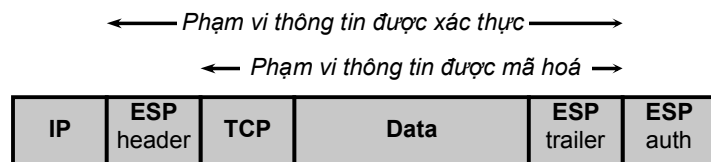
*Chế độ đường hầm*: toàn bộ gói IP được mã hóa và xác thực.

Sự khác nhau giữa hai chế độ hoạt động được mô tả ở hình 3.12.

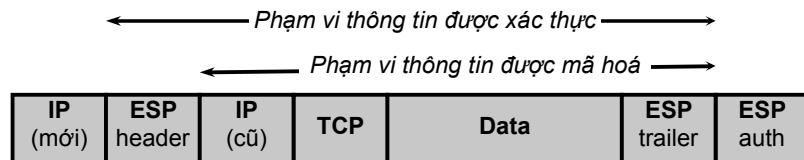
**a- Gói IP gốc**



**b- Gói IP ở chế độ transport**



**b- Gói IP ở chế độ tunnel**



**Hình 3.12:** Tác dụng của ESP lên gói IP ở hai chế độ transport và tunnel

### III.2.7 Quản lý khóa trong IPSec:

IPSec dựa trên kỹ thuật xác thực HMAC (hashed based MAC) và các phương pháp mật mã đối xứng mà cơ bản là DES. Do vậy, vấn đề quản lý và phân phối các khóa bí mật giữa các đầu cuối SA là vấn đề quan trọng trong triển khai IPSec. Có hai cơ chế để quản lý khóa:

-*Quản lý khóa bằng tay (manual)*: người quản trị mạng tạo ra khóa và cài đặt cho các hệ thống đầu cuối. Cơ chế này chỉ phù hợp với các hệ thống có quy mô nhỏ.

-*Quản lý khóa tự động (automated)*: một hệ thống tự động tạo ra và phân phối khóa cho các hệ thống đầu cuối.

IPSec sử dụng hai hệ thống quản lý khóa tự động là Oakley và ISAKMP.

-*Oakley Key Determination Protocol*: Đây là giao thức trao đổi khóa dựa trên giải thuật Diffie-Hellman, có bổ sung thêm các chức năng bảo mật.

-*Internet Security Association and Key Management Protocol (ISAKMP)*: cung cấp một mô hình chung cho việc quản lý khóa trên Internet, định nghĩa các thủ tục và khuôn dạng riêng.



### III.3 SECURE SOCKETS LAYER

Secure Sockets layer hay SSL là một giao thức bảo mật được Netscape thiết kế nhằm cung cấp các kết nối bảo mật cho các ứng dụng trên nền giao thức TCP/IP. SSL đã được chuẩn hóa và sử dụng rộng rãi trong nhiều ứng dụng trên mạng Internet như web, mail, ... Phiên bản hiện tại của SSL là 3.0. Phiên bản SSL được IEEE chuẩn hóa là được gọi là TLS (Transport Layer Security), và được xem như là SSL phiên bản 3.1.

#### III.3.1 Cấu trúc SSL:

SSL thực ra bao gồm hai lớp giao thức nằm phía trên TCP. Lớp thứ nhất là giao thức truyền dữ liệu SSL (SSL record protocol) và lớp thứ hai gồm một tập các giao thức phụ trợ (hình 3.13). Phần này giới thiệu khái quát các thành phần của SSL.



Hình 3.13: Cấu trúc SSL

Hai khái niệm cơ bản thường được dùng trong SSL là **kết nối (connection)** và **phiên giao dịch (session)**.

-**Kết nối** là một kết nối (tạm thời) giữa một đầu cuối này với một đầu cuối kia để cung cấp một loại dịch vụ thích hợp. Mỗi kết nối liên kết với một phiên giao dịch (session).

-**Phiên giao dịch** là một liên kết giữa một máy con và một máy chủ, được tạo ra bởi giao thức *SSL Handshake protocol*. Phiên giao dịch định nghĩa các tham số bảo mật dùng chung cho nhiều kết nối.

Trạng thái của phiên giao dịch được định nghĩa bởi các thông số sau đây:

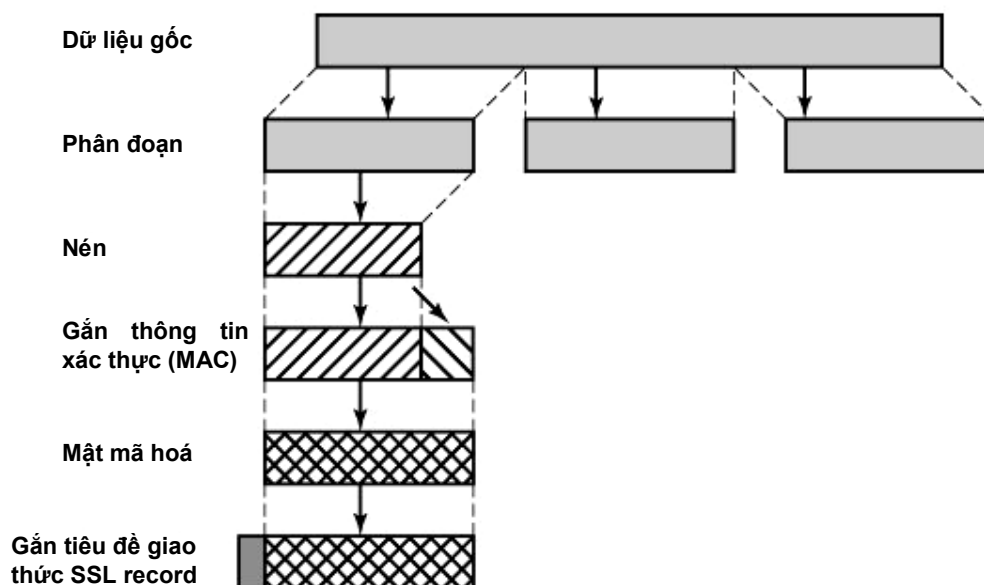
- *Nhận dạng phiên (Session identifier)*: Một chuỗi byte ngẫu nhiên được server chọn để nhận dạng một trạng thái của phiên giao dịch.
- *Chứng thực khóa đối phương (Peer certificate)*: Chứng thực khóa công khai (X509.v3) của thực thể đối phương. Thành phần này có thể có hoặc không.
- *Phương pháp nén (Compression method)*: Giải thuật nén dữ liệu trước khi mã hóa.
- *Thuật toán mã (Cipher spec)*: Xác định thuật toán mã hóa và hàm băm được sử dụng cho phiên giao dịch.
- *Khóa (Master secret)*: Khóa bí mật (48-byte) dùng chung giữa máy con và server.

- *Khả năng phục hồi (Is resumable)*: Cho biết phiên giao dịch này có thể khởi tạo một kết nối mới hay không.

Tương tự, các thông số định nghĩa trạng thái của một kết nối bao gồm:

- *Số nhận dạng ngẫu nhiên (Server and client random)*: Chuỗi byte chọn ngẫu nhiên bởi server và client, có chức năng phân biệt các kết nối với nhau.
- *Khóa xác thực của máy chủ (Server write MAC secret)*: Khóa bí mật dùng để tính giá trị xác thực MAC trên dữ liệu gửi đi từ server.
- *Khóa xác thực của máy con (Client write MAC secret)*: Khóa bí mật dùng để tính giá trị xác thực MAC trên dữ liệu gửi đi từ máy con.
- *Khóa mật mã của máy chủ (Server write key)*: Khóa bí mật dùng để mật mã hóa dữ liệu gửi đi từ server.
- *Khóa mật mã của máy con (Client write key)*: Khóa bí mật dùng để mật mã hóa dữ liệu gửi đi từ client.
- *Véc – tơ khởi tạo (Initialization vectors)*: vec-tơ khởi tạo (IV) dùng trong chế độ mã hóa CBC (Chaining Block Cipher). Giá trị này được khởi tạo bởi giao thức SSL record.
- *Số thứ tự gói (Sequence numbers)*: Số thứ tự của các bản tin được gửi đi và nhận về trên kết nối.

### III.3.2 Giao thức truyền dữ liệu SSL:

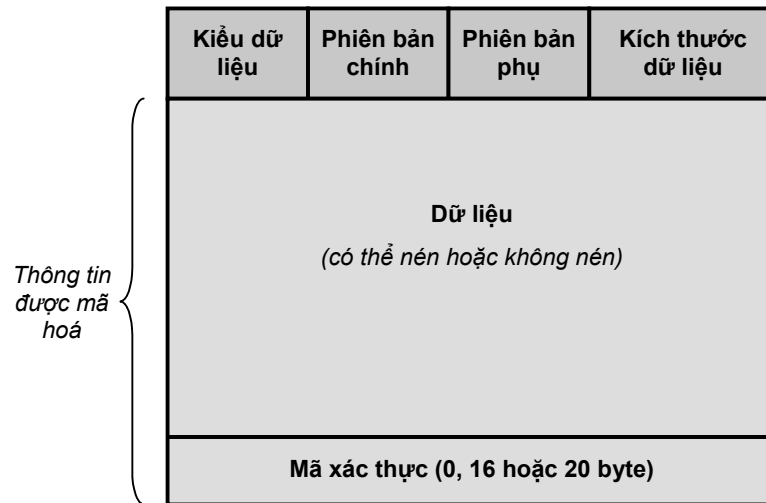


**Hình 3.14:** Hoạt động của giao thức truyền dữ liệu SSL

Giao thức truyền dữ liệu SSL (SSL record protocol) cung cấp 2 dịch vụ cơ bản cho các kết nối SSL là *dịch vụ bảo mật* và *dịch vụ toàn vẹn dữ liệu*.

Hình 3.14 mô tả hoạt động của giao thức truyền dữ liệu SSL. Theo đó, các thao tác mà SSL thực hiện trên dữ liệu bao gồm: phân đoạn dữ liệu (fragmentation), nén dữ liệu

(compression), xác thực dữ liệu (MAC), mã hóa, thêm các tiêu đề cần thiết và cuối cùng gói toàn bộ đoạn thông tin trên trong một segment TCP. Ở phía nhận, quá trình được thực hiện ngược lại.



**Hình 3.15:** Cấu trúc gói SSL record

Cấu trúc gói dữ liệu SSL record gồm các thành phần sau (hình 3.15):

- *Kiểu dữ liệu (Content Type - 8 bits)*: Giao thức lớp trên. Giao thức này sẽ xử lý thông tin trong gói dữ liệu SSL.

- *Phiên bản chính (Major Version - 8 bits)*: Phiên bản chính của SSL. Đối với SSL v3, giá trị này là 3.

- *Phiên bản phụ (Minor Version - 8 bits)*: Phiên bản phụ của SSL. Ví dụ: đối với SSLv3 thì giá trị trường này là 0.

- *Kích thước dữ liệu (Compressed Length - 16 bits)*: Chiều dài của phần dữ liệu (plaintext), tính theo byte.

- *Dữ liệu (Plaintext)*: Dữ liệu của lớp trên được chuyển đi trong gói SSL record. Dữ liệu này có thể được nén hoặc không.

- *Mã xác thực (MAC)*: Mã xác thực, có kích thước = 0 byte nếu không dùng chức năng xác thực.

### III.3.3 Giao thức thay đổi thông số mã:

Giao thức thay đổi thông số mã (Change cipher spec protocol) là giao thức đơn giản nhất trong cấu trúc SSL, dùng để thay đổi các thông số mã hóa trên kết nối SSL. Giao thức này chỉ gồm có một bản tin có kích thước 1 byte, mang giá trị 1. Chức năng của bản tin này là yêu cầu cập nhật các thông số mã hóa cho kết nối hiện hành.

### III.3.4 Giao thức cảnh báo:

Giao thức cảnh báo (Alert protocol) dùng để trao đổi các bản tin cảnh báo giữa hai đầu của kết nối SSL. Có hai mức độ cảnh báo: warning (1) và fatal (2). Mức warning chỉ đơn giản dùng để thông báo cho đầu kia các sự kiện bất thường đang diễn ra. Mức fatal yêu cầu kết thúc kết nối SSL hiện hành, các kết nối khác trong cùng phiên giao dịch có thể vẫn được duy trì nhưng phiên giao dịch không được thiết lập thêm kết nối mới.

Các bản tin cảnh báo của SSL bao gồm:

- unexpected\_message*: Nhận được một bản tin không phù hợp.
- bad\_record\_mac*: Bản tin vừa nhận có giá trị MAC không hợp lệ.
- decompression\_failure*: Thao tác giải nén thực hiện không thành công..
- handshake\_failure*: Phía gửi không thương lượng các thông số bảo mật.
- illegal\_parameter*: Một trường nào đó trong bản tin bắt tay (handshake message) không hợp lệ.
- close\_notify*: Thông báo kết thúc kết nối.
- no\_certificate*: Khi nhận được yêu cầu cung cấp chứng thực khóa (certificate), nhưng nếu không có chứng thực khóa nào thích hợp thì gửi cảnh báo này.
- bad\_certificate*: Chứng thực khóa không hợp lệ (chữ ký sai)
- unsupported\_certificate*: Kiểu chứng thực không được hỗ trợ.
- certificate\_revoked*: Chứng thực khóa đã bị thu hồi.
- certificate\_expired*: Chứng thực khóa đã hết hạn sử dụng.
- certificate\_unknown*: Không xử lý được chứng thực khóa vì các lý do khác với các lý do trên.

### III.3.5 Giao thức bắt tay:

*Giao thức bắt tay (handshake protocol)* là giao thức phức tạp nhất của SSL, được hai phía sử dụng để xác thực lẫn nhau và thương lượng để thống nhất các thuật toán xác thực MAC và mã hóa. Thủ tục này cũng được để trao đổi các khóa bí mật dùng cho mã hóa và MAC. Thủ tục phải được thực hiện trước khi dữ liệu được truyền.

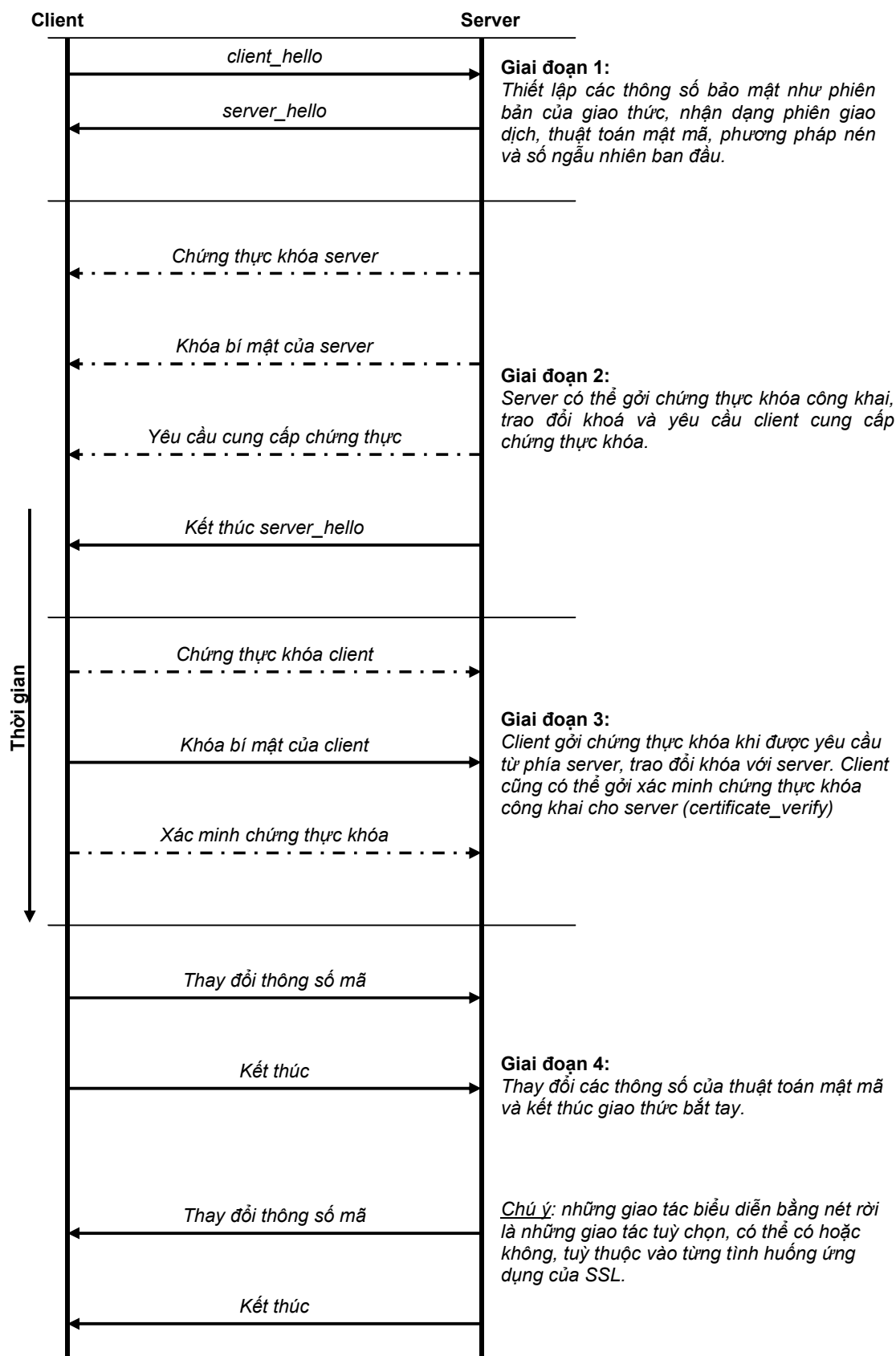
Thủ tục bắt tay gồm 4 giai đoạn được mô tả ở hình 3.16.

### III.3.6 So sánh SSL và IPSec:

SSL và IPSec là hai giao thức tương đồng với nhau về chức năng. Cả hai đều được thiết kế để bảo vệ dữ liệu truyền trên các kết nối bằng các cơ chế xác thực và mã hóa. Tuy nhiên, hai kỹ thuật này có những điểm khác biệt nhau như sau:

- SSL hoạt động ở lớp socket (hình 3.13), do đó nó được gắn kết ở *phần người sử dụng (user space)* trong các hệ thống đầu cuối. IPSec hoạt động ở lớp mạng (network layer), nên được tích hợp vào trong *chức năng của hệ điều hành*. Đây chính là sự khác nhau cơ bản nhất giữa SSL và IPSec.
- Cả SSL và IPSec đều cung cấp chức năng mã hóa (Encryption), bảo vệ dữ liệu (Integrity) và xác thực thông tin (Authentication), tuy nhiên SSL đơn giản hóa các kỹ thuật này để áp dụng trong mô hình của nó, trong khi IPSec bao gồm một cách đầy đủ các chi tiết thiết kế của tất cả các kỹ thuật tạo thành, và do đó, khi tổ hợp lại sẽ xuất hiện nhiều lỗi tương thích trong nội bộ IPSec.
- IPSec là thành phần của hệ điều hành, do đó, để triển khai IPSec thì phải thay đổi cấu hình hệ điều hành mà không cần thay đổi cấu hình chương trình ứng dụng. Ngược lại, SSL nằm ở mức người dùng nên phải cài đặt với từng ứng dụng cụ thể (ví dụ mail, web, ...) mà không cần khai báo với hệ điều hành,

Vì những khác biệt trên đây, SSL thường được sử dụng để bảo vệ kết nối cho từng ứng dụng cụ thể, đặc biệt là Web, E-mail. Trong khi đó, IPSec thường được dùng để xây dựng các mạng riêng ảo (VPN) rồi trên cơ sở đó mới triển khai các dịch vụ ứng dụng.



**Hình 3.16:** Thủ tục bắt tay SSL

### III.4 SECURE ELECTRONIC TRANSACTION

#### III.4.1 Tổng quan về SET:

Secure Electronic Transaction hay SET là một kỹ thuật được thiết kế để bảo vệ các thông tin quan trọng trao đổi trên mạng (ví dụ số thẻ tín dụng) dùng trong các giao dịch thanh toán qua mạng Internet.

SET phiên bản 1 được đề xuất năm 1996 (*MasterCard* và *Visa* chủ trì), sau đó được nhiều nhà sản xuất khác tham gia phát triển (như *Microsoft*, *IBM*, *Netscape*, *RSA*, *Terisa* và *Verisign*).

SET không phải là một hệ thống thanh toán, mà chỉ là một giao thức an toàn cho phép các đầu cuối trao đổi các thông tin bí mật, đặc biệt là các thông tin về tài khoản ngân hàng, thông qua các môi trường công cộng ví dụ như Internet.

##### **-Các tính năng của SET:**

- *Bảo mật thông tin:* đặc biệt là thông tin về tài khoản ngân hàng khi những thông tin này được trao đổi qua mạng. SET còn có chức năng ngăn chặn người bán hàng biết số thẻ tín dụng (credit card) của người mua hàng. Kỹ thuật mã hóa quy ước với thuật toán DES được dùng để cung cấp chức năng này.
- *Bảo toàn dữ liệu:* các thông tin về việc đặt hàng, thanh toán, thông tin cá nhân khi gửi từ một người mua hàng đến người bán hàng được đảm bảo toàn vẹn, không bị thay đổi. Kỹ thuật chữ ký số DSA với hàm băm SHA-1 được dùng để bảo đảm tính năng này (trong một số bản tin của SET, HMAC được dùng thay cho DSA).
- *Xác thực tài khoản của người sử dụng thẻ:* cho phép người bán hàng xác minh người dùng thẻ là chủ nhân hợp lệ của tài khoản đang đề cập. Để thực hiện chức năng này, SET dùng chuẩn xác thực X.509 version 3.
- *Xác thực người bán hàng:* SET cho phép người sử dụng thẻ xác thực rằng người bán hàng có quan hệ với một tổ chức tài chính có chấp nhận thanh toán qua thẻ. Chức năng này cũng được thực hiện dùng X.509 version 3.

Một điều cần chú ý là SET hoạt động bằng cách truy xuất trực tiếp đến lớp TCP/IP mà không dùng các giao thức ở lớp ứng dụng khác. Tuy vậy hoạt động của SET cũng không ảnh hưởng đến các cơ chế bảo mật khác như IPsec hoặc SSL.

##### **-Các thành phần của SET:**

-*Người dùng thẻ (Cardholder):* Người dùng thẻ tín dụng để thực hiện các giao dịch thanh toán trên Internet (người mua hàng).

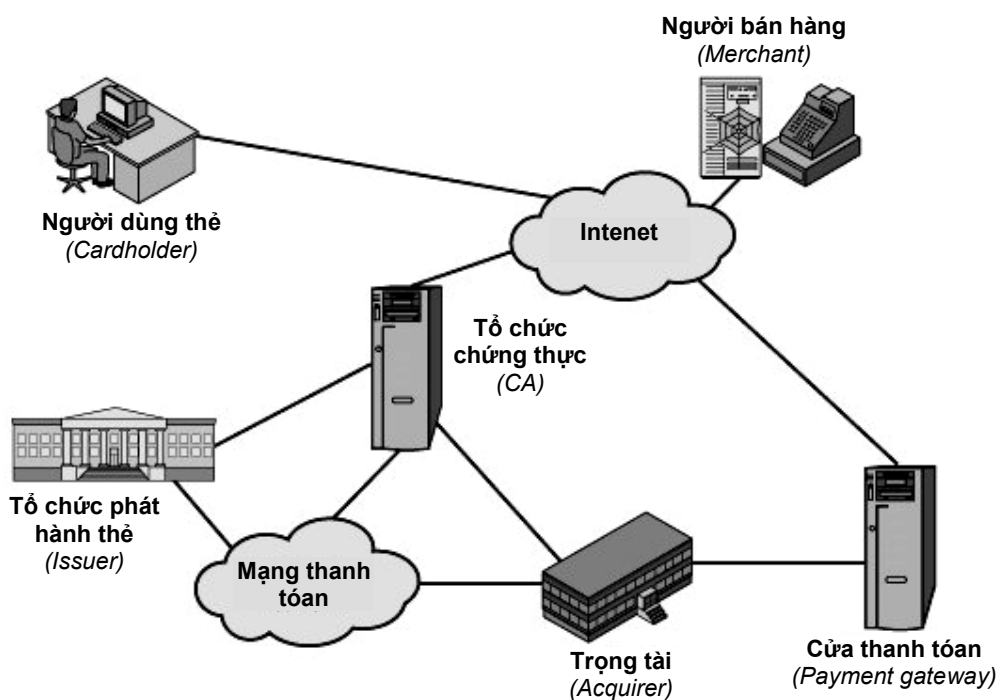
-*Người bán hàng (Merchant):* Một cá nhân hay tổ chức bán hàng hoặc dịch vụ trên mạng (thông qua web hoặc email). Người bán hàng phải có khả năng chấp nhận thanh toán bằng thẻ, và phải có quan hệ với một tổ chức tài chính nào đó (Acquirer).

-*Tổ chức phát hành thẻ (Issuer):* Đây là tổ chức tài chính (thường là ngân hàng) phát hành thẻ tín dụng. Tổ chức này có trách nhiệm thanh toán theo yêu cầu của người sử dụng thẻ.

-*Trọng tài (Acquirer):* Một tổ chức tài chính khác có quan hệ với người bán hàng, thực hiện việc xác thực tài khoản của người mua hàng và thanh toán. Trọng tài sẽ kiểm tra tài khoản của người mua hàng để thông báo cho người bán hàng biết số dư trong tài khoản của người mua có đủ để thực hiện giao dịch hay không. Sau khi giao dịch mua hàng được thực hiện, trọng tài thực hiện việc chuyển tiền từ tài khoản của người mua hàng sang tài toàn khoản của người bán hàng, đồng thời ra yêu cầu thanh toán đối với ngân hàng phát hành thẻ (Issuer).

-*Cửa thanh toán (Payment gateway)*: Đây là thành phần chịu trách nhiệm xử lý các bản tin thanh toán (payment message) được điều hành bởi trọng tài hoặc một tổ chức thứ 3 được chỉ định. Payment gateway giao tiếp giữa SET và hệ thống thanh toán của ngân hàng để thực hiện các thao tác xác thực và thanh toán. Như vậy, người bán hàng thật ra trao đổi các thông báo với cửa ngõ thanh toán thông qua mạng Internet, sau đó, Payment gateway mới liên kết đến hệ thống xử lý tài chính của Acquirer.

-*Tổ chức chứng thực (Certification authority \_ CA)*: Là thành phần có chức năng tạo ra các chứng thực (certificate) theo chuẩn X.509v3 và phân phối cho Cardholder, Merchant và Payment Gateway. Sự thành công của SET phụ thuộc vào sự tồn tại của CA. Thông thường, CA được tổ chức theo một mô hình phân cấp với nhiều CA liên hệ với nhau.



**Hình 3.17:** Các thành phần của SET

#### **-Thực hiện giao dịch với SET:**

Một giao dịch SET điển hình gồm các bước sau đây:

1. Khách hàng mở tài khoản tại một ngân hàng có dịch vụ thanh toán qua mạng (ví dụ MasterCard, Visa card, ...) và trở thành Cardholder.
2. Khách hàng nhận được một chứng thực X.509v3, được ký bởi ngân hàng bằng chữ ký số (digital signature), trong đó chứa khóa công khai RSA của khách hàng và ngày hết hạn.
3. Người bán hàng nhận chứng thực: Người bán hàng phải có 2 chứng thực khác nhau chứa khóa công khai cho hai mục đích: ký nhận các thông báo (message signing) và trao đổi khóa (key exchange). Ngoài ra, người bán hàng cũng có một bản sao chứng thực của Payment gateway.



4. Khách hàng đặt hàng: thao tác này được thực hiện thông qua website của người bán hàng hoặc qua email.
5. Xác nhận người bán hàng: người bán hàng gửi chứng thực của mình cho người mua hàng để chứng minh tính sở hữu của mình đối với một kho hàng nào đó.
6. Lệnh đặt hàng và thanh toán được thực hiện: người mua hàng gửi lệnh đặt hàng và lệnh thanh toán cho người bán hàng cùng với chứng thực của mình. Thông tin thanh toán (số thẻ tín dụng) được mã hoá sao cho người bán hàng không thể thấy được nhưng có thể kiểm tra tính hợp lệ của nó.
7. Người bán hàng yêu cầu xác thực việc thanh toán thông qua Payment gateway.
8. Người bán hàng xác nhận đơn đặt hàng bằng cách gửi thông báo cho người mua hàng.
9. Người bán hàng giao hàng (hoặc bắt đầu cung cấp dịch vụ) cho người mua hàng.
10. Người bán hàng yêu cầu thanh toán thông qua Payment gateway.

#### III.4.2 Chữ ký song song:

Chữ ký song song (dual signature) là một thuật ngữ được dùng trong SET để diễn đạt một liên kết giữa *hai bản tin được gửi đi bởi cùng một người gửi nhưng cho hai người nhận khác nhau*.

Khi mua hàng qua mạng, khách hàng gửi thông tin đặt hàng OI (Order information) với chữ ký của mình cho người bán hàng, đồng thời gửi thông tin thanh toán PI (Payment information) cho ngân hàng cũng với chữ ký của mình. Về nguyên tắc, ngân hàng không cần biết các chi tiết về đặt hàng, và người bán hàng cũng không cần biết các chi tiết về thanh toán. Chữ ký song song được sử dụng trong trường hợp này để tránh các tranh chấp xảy ra khi thông tin đặt hàng và thông tin thanh toán không khớp nhau. Hình 3.18 mô tả hoạt động của chữ ký song song.

-Người mua hàng áp dụng hàm băm lên PI và OI (dùng SHA-1), sau đó hai giá trị băm được nối với nhau và áp dụng hàm băm một lần nữa với khóa riêng của chính người mua hàng để tạo thành chữ ký song song:

$$DS = E([H(H(PI) + H(OI)), PR_c])$$

Trong đó  $PR_c$  là khóa riêng của người mua hàng.

-Người bán hàng xác nhận chữ ký của người mua hàng bằng cách tính hai giá trị:

$$H(PIMS + H[OI]) \text{ và } D(DS, PU_c)$$

Trong đó PIMS là mã băm của PI,  $PU_c$  là khóa công khai của khách hàng, DS là chữ ký song song nhận được trên đơn đặt hàng.

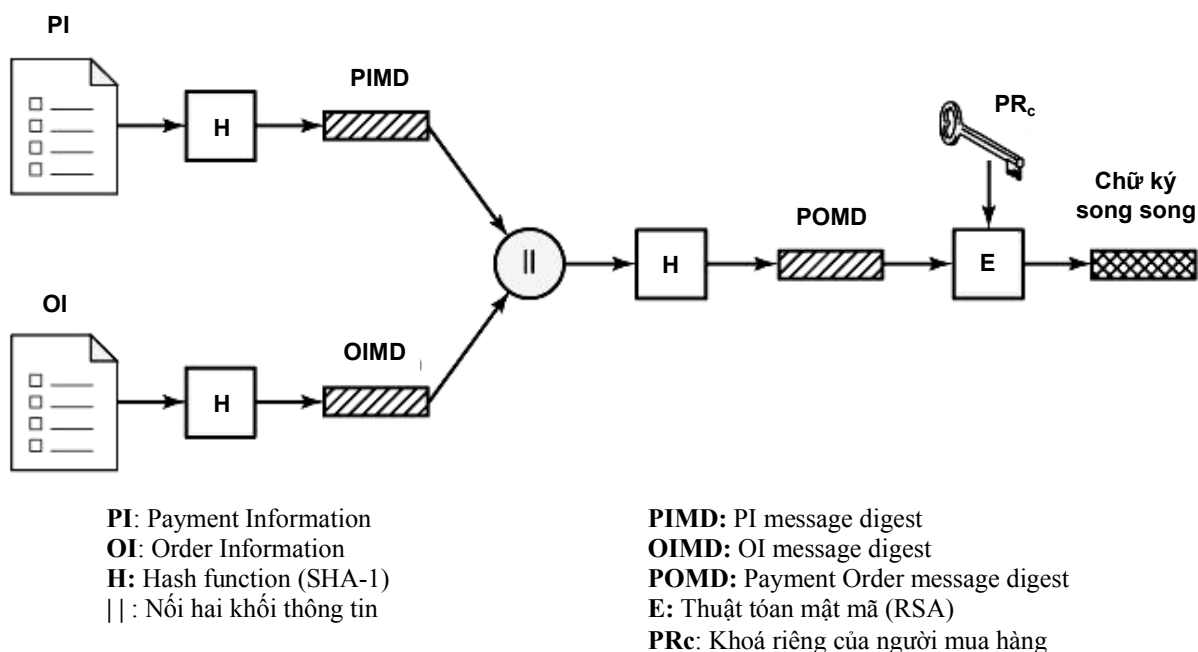
Nếu hai giá trị trên bằng nhau, thì chữ ký xem như chính xác và đơn đặt hàng được chấp nhận.

-Song song đó, ngân hàng cũng xác thực chữ ký song song bằng cách so sánh hai giá trị sau đây:

$$H(H[OI] + OIMD) \text{ và } D(DS, PU_c)$$

Trong đó OIMD là message digest của OI.

Nếu hai giá trị vừa tính được là bằng nhau thì xem như chữ ký là chính xác và lệnh thanh toán được chấp nhận.



**Hình 3.18:** Chữ ký song song (dual signature)

### III.4.3 Thực hiện thanh toán trong SET:

Xử lý thanh toán (Payment processing) là công đoạn quan trọng nhất trong giao dịch SET. Quá trình xử lý thanh toán gồm 3 công việc như sau:

- Yêu cầu mua hàng (Purchase Request).
- Xác thực thanh toán (Payment Authorization).
- Thực hiện thanh toán (Payment Capture).

**Bảng 3.1:** Các giao tác của SET

Tên giao tác	Ý nghĩa
Cardholder registration	Người mua hàng đăng ký với CA trước khi thực hiện các giao dịch SET khác với người bán hàng.
Merchant registration	Người bán hàng đăng ký với CA trước khi gửi các thông báo SET với khách hàng và với Payment gateway.
Purchase request	Thông báo được người mua hàng gửi đi, trong đó chứa lệnh đặt hàng (OI) cho người bán hàng và lệnh thanh toán (PI) cho ngân hàng.
Payment authorization	Trao đổi giữa người bán hàng và Payment gateway để kiểm tra số dư trong tài khoản của người mua hàng.
Payment capture	Người bán hàng gửi yêu cầu thanh toán đến Payment gateway.
Certificate inquiry and status	Trong trường hợp CA không xử lý được yêu cầu cung cấp chứng thực tức thời, nó sẽ trả lời cho người mua hàng và người bán hàng về việc trì hoãn này. Sau đó, người mua hàng hoặc người bán hàng có thể dùng giao dịch này để kiểm tra trạng thái của chứng thực. Nếu chứng thực đã được xử lý

	xong thì khách hàng hoặc người bán hàng sẽ được nhận.
Purchase inquiry	Người mua hàng kiểm tra trạng thái của đơn đặt hàng sau khi đã xác nhận đơn đặt hàng với người bán hàng.
Authorization reversal	Người bán hàng hiệu chỉnh yêu cầu xác thực trước đó. Nếu đơn đặt hàng không thực hiện được thì toàn bộ việc xác thực trước đó được hồi lại (reverse). Nếu đơn đặt hàng chỉ được thực hiện một phần (người mua hàng hồi lại một phần) thì người bán hàng chỉ hồi lại phần đã xác thực tương ứng.
Capture reversal	Người bán hàng hiệu chỉnh các thông tin yêu cầu thanh toán đã gửi cho Payment gateway.
Credit	Người bán hàng trả lại tiền vào tài khoản của người mua hàng khi hàng được trả lại vì lý do nào đó (hư hỏng, sai quy cách, ...).
Credit reversal	Người bán hàng hiệu chỉnh lại yêu cầu trả lại tiền vào tài khoản của người mua hàng (giao tác Credit) vừa rồi.
Payment gateway certificate request	Người bán hàng yêu cầu bản sao chứng thực của Payment gateway.
Batch administration	Người bán hàng thông báo cho Payment gateway về các đợt giao hàng.
Error message	Thông báo lỗi xảy ra trong giao dịch.

### **-Yêu cầu mua hàng:**

Sau khi người mua hàng hoàn tất các công việc chọn hàng và đặt mua trên mạng, thủ tục yêu cầu mua hàng mới được bắt đầu. Chú ý rằng thao tác chọn hàng và đặt mua được thực hiện trên các kết nối bình thường (như e-mail hay web) mà không cần có sự tham gia của SET.

Quá trình yêu cầu mua hàng bao gồm 4 giao tác: *Initiate Request*, *Initiate Response*, *Purchase Request*, và *Purchase Response*.

Để gửi được các bản tin SET đến người bán hàng, người mua hàng cần có một bản sao các chứng thực của Merchant và Payment gateway. Bản tin *Initiate Request* được sử dụng để yêu cầu người bán hàng cung cấp các chứng thực cần thiết cho người mua hàng.

Người bán hàng sẽ trả lời bản tin *Initiate Request* bằng một bản tin hồi đáp *Initiate Response* trong đó có chứa giá trị ngẫu nhiên (nonce) đã được tạo ra trước đó bởi người mua hàng, một giá trị ngẫu nhiên khác do người bán hàng tạo ra, nhận diện của giao tác hiện hành, cùng với các chứng thực của chính người bán hàng và Payment gateway. Tất cả các thông tin này được xác thực bởi chữ ký của người bán hàng.

Người mua hàng xác minh các chứng thực nhận được, sau đó tạo ra thông tin đặt hàng (OI) và thông tin thanh toán (PI), trong đó có chứa nhận diện giao tác mà người bán hàng vừa tạo ra trước đó. Người mua hàng chuẩn bị bản tin *Purchase Request*. Bản tin này chứa các thông tin sau đây:

- Các thông tin liên quan đến việc thanh toán bao gồm: PI, chữ ký song song, OIMD và một phong bì số (digital envelope). Các thông tin này được mã hoá bằng khoá bí mật  $K_s$  do người mua hàng tạo ra cho từng phiên giao dịch.
- Các thông tin liên quan đến đơn đặt hàng bao gồm OI, chữ ký song song, PIMD. Chú ý rằng OI được gửi đi trực tiếp mà không cần mã hoá.

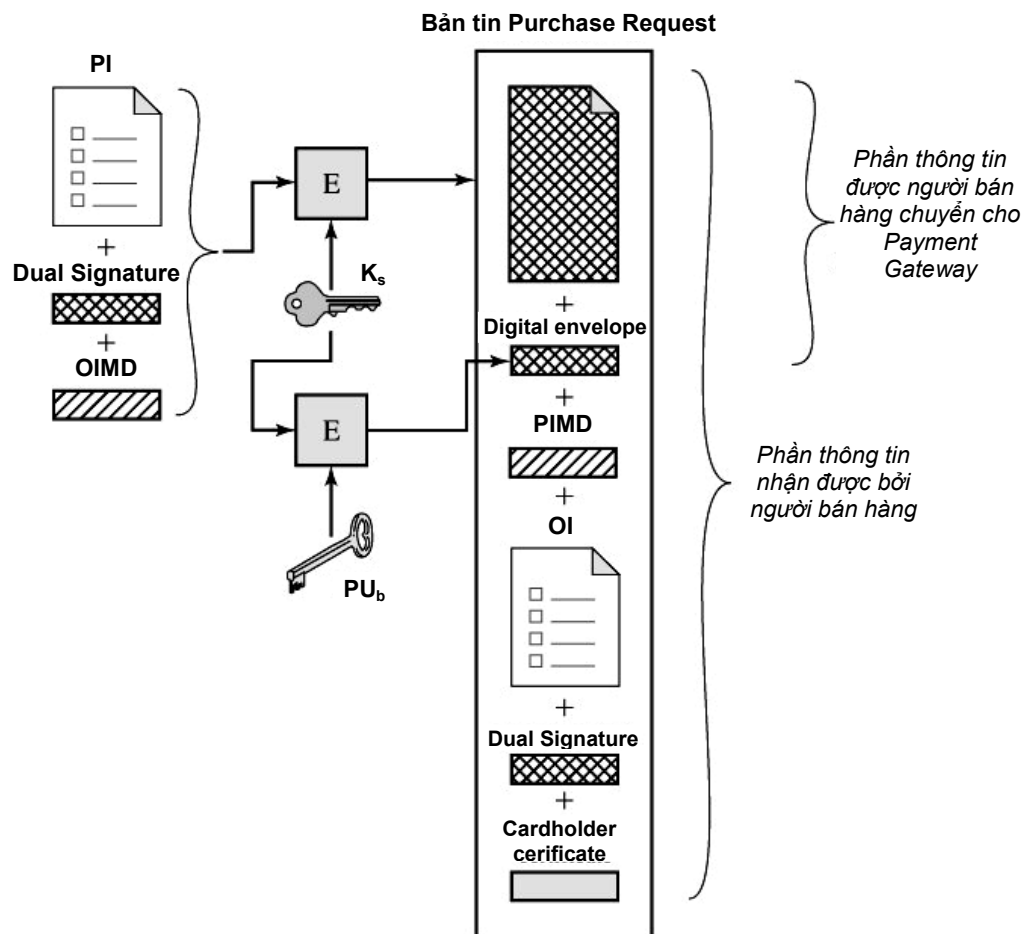
- Chứng thực của người mua hàng.

Khi người bán hàng nhận được *Purchase Request*, họ sẽ thực hiện các thao tác sau đây:

- Xác minh chứng thực của người mua hàng.
- Kiểm chứng chữ ký song song của người mua hàng.
- Xử lý đơn đặt hàng và chuyển thông tin thanh toán cho Payment Gateway để kiểm tra.
- Gửi bản tin *Purchase Response* cho người mua hàng.

Bản tin *Purchase Response* chứa các thông tin để chấp nhận đơn đặt hàng và các tham chiếu đến số nhận diện giao tác tương ứng. Thông tin này được ký bởi người bán hàng và gửi cho người mua hàng cùng với chứng thực của người bán.

Người mua hàng khi nhận được bản tin *Purchase Response* sẽ tiến hành kiểm tra chữ ký và chứng thực của người bán hàng.

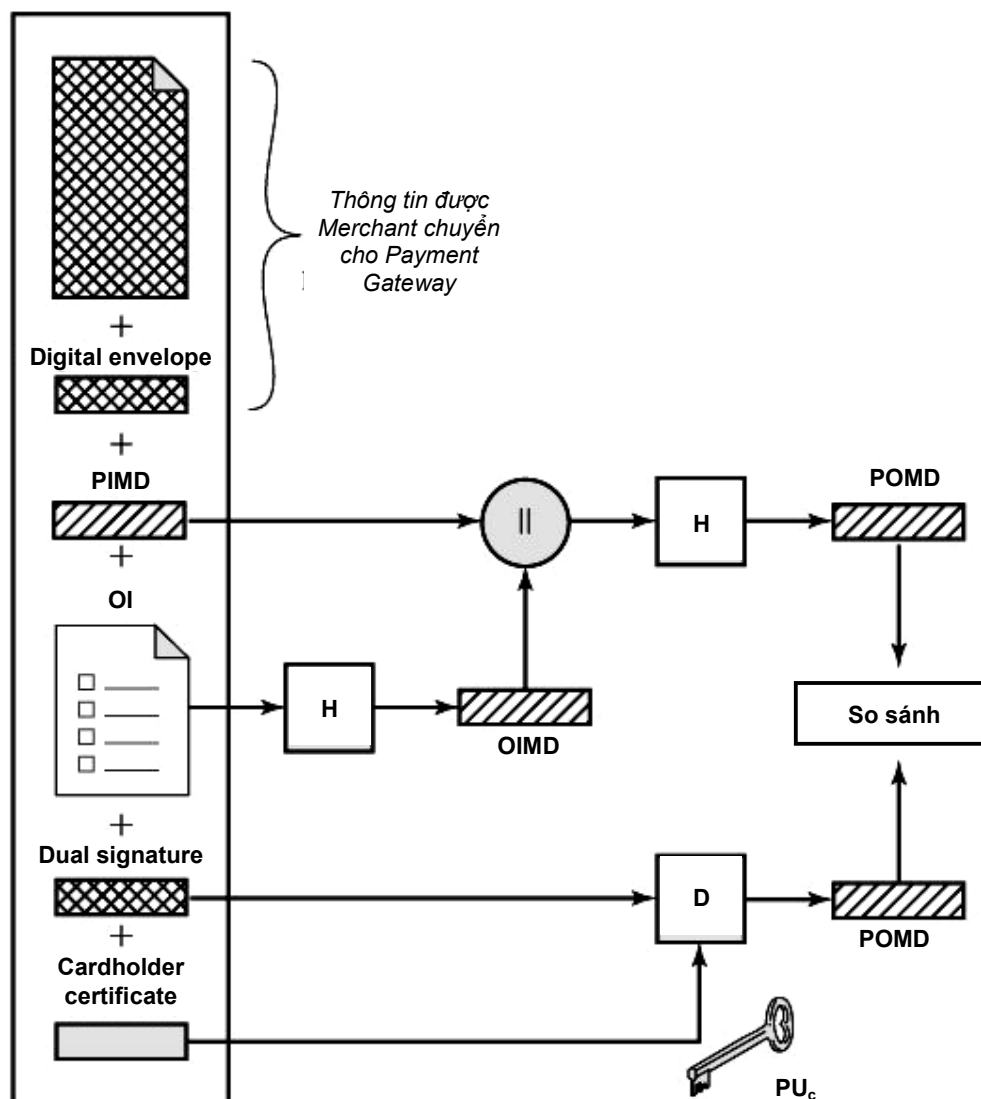


**Hình 3.19:** Quá trình tạo bản tin *Purchase request* của người mua hàng

#### **-Xác thực thanh toán:**

Đây là thủ tục mà người bán hàng xác thực tính hợp lệ của người mua hàng thông qua Payment Gateway. Quá trình xác thực nhằm bảo đảm rằng giao dịch này được chấp thuận bởi ngân hàng phát hành thẻ (Issuer), và do đó người bán hàng sẽ được đảm bảo thanh toán. Quá trình này được thực hiện thông qua hai bản tin: *Authorization Request* và *Authorization response*.

### Bản tin Purchase Request



**Hình 3.18:** Quá trình xác minh yêu cầu mua hàng (Purchase Request) tại Merchant

Bản tin *Authorization Request* được người bán hàng gửi đến Payment Gateway bao gồm các thông tin sau:

- Thông tin liên quan đến việc mua hàng, bao gồm: PI, chữ ký song song, OIMD và phong bì số (digital envelope).
- Thông tin liên quan đến xác thực bao gồm: nhận diện giao tác, được mã hoá bằng khoá bí mật do người bán hàng tạo ra và phong bì số, được mã hoá bằng khoá công khai của Payment gateway.
- Các chứng thực của người mua hàng và người bán hàng.

Khi nhận được *Authorization Request*, Payment Gateway thực hiện các thao tác sau:

- Xác minh tất cả các chứng thực.
- Giải mã phong bì số của khối thông tin mua hàng.

- Xác minh chữ ký của người bán hàng.
- Giải mã phong bì số của khối thông tin xác thực.
- Xác minh chữ ký song song.
- Xác minh nhận diện giao tác (transaction ID).
- Yêu cầu xác thực từ ngân hàng phát hành thẻ.

Nếu nhận được thông tin xác thực thành công từ ngân hàng phát hành thẻ, Payment Gateway sẽ hồi đáp bằng bản tin *Authorization Response* trong đó chứa các thông tin sau:

- Thông tin liên quan đến xác thực bao gồm: khối thông tin xác thực được ký bởi Payment Gateway và mã hoá bằng khoá bí mật do Payment Gateway tạo ra, ngoài ra còn có phong bì số.
- Thông tin liên quan đến thực hiện thanh toán.
- Chứng thực của Payment gateway.

Với thông tin xác thực này, người bán hàng đã có thể bắt đầu giao hàng hoặc cung cấp dịch vụ cho người mua hàng.

#### ***-Thực hiện thanh toán:***

Để thực hiện thanh toán, người bán hàng thực hiện một giao tác với Payment Gateway gọi là Capture transaction, giao tác này được thực hiện qua hai bản tin: *Capture Request* và *Capture Response*.

Trong bản tin *Capture Request*, người bán hàng tạo ra thông tin yêu cầu thanh toán, trong đó có khối lượng thanh toán và nhận diện giao tác (transaction ID), cùng với thông tin xác thực nhận được trước đó từ Payment Gateway, chữ ký và chứng thực của người bán hàng.

Payment Gateway nhận được bản tin này, giải mã và thực hiện các bước kiểm tra cần thiết trước khi yêu cầu ngân hàng phát hành thẻ chuyển tiền cho người bán hàng. Cuối cùng, Payment Gateway sẽ thông báo cho người bán hàng bằng bản tin *Capture Response*.

#### **Tóm tắt chương:**

-Các ứng dụng bảo mật (security application) được xây dựng dựa trên các kỹ thuật cơ sở trình bày ở chương 2 bao gồm: mật mã đối xứng, mật mã bất đối xứng, hàm băm, chữ ký số, chứng thực khóa công khai, ...

-Kỹ thuật xác thực được xem là kỹ thuật cơ bản nhất để quản lý truy xuất. Mật khẩu là phương tiện xác thực đơn giản nhất và hiệu quả nhất từ trước đến nay. Tuy nhiên, mật khẩu được quản lý và sử dụng bởi con người, nên cần phải có các chính sách hợp lý để đảm bảo mật khẩu không bị tiết lộ.

-Trong mô hình thông tin điểm – điểm, hai giao thức xác thực thường được dùng là PAP (Password Authentication Protocol) và CHAP (Challenge Handshake Authentication Protocol) trong đó, giao thức CHAP có nhiều ưu điểm hơn và an toàn hơn do không gửi mật khẩu đi trực tiếp trên mạng.

-Trong mô hình phân tán, giao thức xác thực cần phải đáp ứng được hai yêu cầu: đảm bảo thông tin xác thực không bị đánh cắp và người sử dụng chỉ cần xác thực một lần cho tất cả các dịch vụ trong hệ thống phân tán. Kerberos là một giao thức xác thực đáp ứng được 2 yêu cầu này.

-Giao thức bảo mật IP Security (IPSec) là một sự mở rộng của giao thức IP, cho phép lớp mạng thực hiện các chức năng bảo mật và toàn vẹn cho dữ liệu truyền đi trên mạng. IPSec là một chuẩn phức tạp, bao gồm đặc tả của nhiều chuẩn khác, được triển khai dựa trên hai giao thức đóng gói cơ bản là ESP và AH. IPSec hoạt động ở hai chế độ là chế độ vận chuyển (transport) và chế độ đường hầm (tunnel). Hoạt động của IPSec là trong suốt đối với các giao thức ở lớp ứng dụng.

-Giao thức bảo mật SSL (Secure Sockets Layer) là một giao thức cộng thêm hoạt động bên trên giao thức TCP. SSL cung cấp hai dịch vụ cơ bản là mã hóa và xác thực dữ liệu / xác thực đầu cuối cho các ứng dụng Internet như web, e-mail, .... SSL được sử dụng rất phổ biến hiện nay trên mạng Internet, đặt biệt trong các thủ tục trao đổi thông tin bí mật giữa client và server như đăng nhập vào hộp thư điện tử, nhập số thẻ tín dụng khi mua hàng, ...

-SET (Secure Electronic Transaction) là một ứng dụng bảo mật trong các hệ thống thanh toán qua mạng. SET là một ứng dụng truy xuất trực tiếp đến lớp TCP (tức không thông qua các giao thức ứng dụng như mail hay web, ...). SET định nghĩa một mô hình phức tạp bao gồm nhiều thực thể như người mua hàng, người bán hàng, ngân hàng phát hành thẻ, trọng tài, cửa thanh toán, ... SET được phát triển bởi các tổ chức tài chính có uy tín như MasterCard, VISA, các tổ chức công nghệ như Microsoft, IBM, RSA, Verisign, ...

## **CÂU HỎI VÀ BÀI TẬP.**

### **A- Câu hỏi trắc nghiệm.**

Câu 1. Nguyên tắc đảm bảo an toàn cho mật khẩu đối với người sử dụng:

- a- Quy định thời gian sử dụng tối đa của mật khẩu.
- b- Không dùng mật khẩu quá ngắn, mật khẩu có chứa tên người dùng, mật khẩu là những từ có nghĩa trong tự nhiên.
- c- Mã hoá mật khẩu khi lưu trữ.
- d- Tất cả đều đúng.

Câu 2. Trong thủ tục xác thực mạng đơn giản, cơ chế nào đảm bảo mỗi thẻ (ticket) chỉ được sử dụng bởi một máy duy nhất?

- a- Máy con phải được xác thực bởi Authentication Server (AS).
- b- Trong thẻ cấp cho máy con có chứa địa chỉ mạng của máy máy con (AD<sub>C</sub>).
- c- Trong thẻ có chứa nhận dạng của máy chủ cung cấp dịch vụ (ID<sub>V</sub>).
- d- Tất cả đều đúng.

Câu 3. Mục đích của TGS (Ticket Granting Server) trong thủ tục xác thực qua mạng?

- a- Cho phép người dùng chỉ đăng nhập một lần nhưng sử dụng được nhiều dịch vụ.
- b- Giảm tải xử lý cho AS
- c- Để hạn chế việc gửi mật khẩu trực tiếp trên mạng.
- d- Tất cả đều sai.

Câu 4. Chọn câu đúng về giao thức xác thực Kerberos 4:

- a- Sử dụng thuật toán mã hoá DES
- b- Để sử dụng một dịch vụ nào đó, client phải thực hiện 2 thao tác: xác thực với AS để được cấp thẻ Ticket-granting-Ticket, sau đó xác thực với TGS để nhận được

- thẻ Service-granting-Ticket trước khi có thẻ yêu cầu máy chủ cung cấp dịch vụ.
- c- Người dùng chỉ cần nhập mật khẩu một lần trong suốt phiên làm việc.
  - d- Tất cả đều đúng.
- Câu 5. Trong Kerberos 4, bản tin yêu cầu xác thực gửi từ máy con đến AS chứa các thông tin nào?
- a- Nhận diện của người dùng ( $ID_C$ ), nhận diện của TGS ( $ID_{TGS}$ ) và nhãn thời gian đồng bộ  $TS_1$ .
  - b- Tên đăng nhập và mật khẩu.
  - c- Tên đăng nhập và địa chỉ mạng của máy con.
  - d- Mật khẩu đã mã hoá và địa chỉ mạng của máy con.
- Câu 6. Trong Kerberos 4, bản tin yêu cầu dịch vụ gửi từ máy con đến máy chủ dịch vụ chứa các thông tin nào?
- a- Chứa thẻ truy xuất dịch vụ được cấp bởi TGS.
  - b- Chứa thẻ truy xuất dịch vụ được cấp bởi TGS và tên đăng nhập.
  - c- Chứa thẻ truy xuất dịch vụ cùng với Authenticator gồm ( $ID_C + AD_C + TS_5$ ) gửi trực tiếp.
  - d- Chứa thẻ truy xuất dịch vụ cùng với Authenticator gồm ( $ID_C + AD_C + TS_5$ ) được mã hoá bằng khoá bí mật dùng chung giữa máy con và máy chủ cung cấp dịch vụ.
- Câu 7. Thế nào là một lãnh địa Kerberos (Kerberos Realm)?
- a- Là hệ thống bao gồm Kerberos server, các máy chủ cung cấp dịch vụ và nhiều máy con.
  - b- Là phạm vi mạng được quản lý bởi một AS.
  - c- Là phạm vi mạng được quản lý bởi một TGS.
  - d- Tất cả đều sai.
- Câu 8. Điểm khác nhau giữa Krberos 4 và Kerberos 5:
- a- Kerberos 5 không giới hạn thời gian tồn tại của thẻ, Kerberos 4 giới hạn thời gian tồn tại của thẻ là khoảng 21 giờ.
  - b- Kerberos 5 sử dụng mật mã bất đối xứng, Kerberos 4 sử dụng mật mã đối xứng.
  - c- Kerberos 5 dùng tên đăng nhập và mật khẩu để xác thực người dùng, Kerberos 4 dùng địa chỉ IP để xác thực.
  - d- Tất cả đều đúng.
- Câu 9. Ứng dụng của IPSec:
- a- Xây dựng các website an toàn cho các ứng dụng thương mại điện tử.
  - b- Xây dựng các mạng riêng ảo VPN trên nền mạng Internet công cộng.
  - c- Cho phép truy xuất từ xa một cách an toàn.
  - d- Tất cả các ứng dụng trên.
- Câu 10. Chọn câu đúng về IPSec:
- a- Khi sử dụng IPSec, kích thước gói dữ liệu IP tăng lên, do đó hiệu suất truyền giảm xuống.
  - b- Khi cài đặt IPSec trên một hệ thống thì IPSec sẽ có tác dụng bảo vệ cho tất cả các



dịch vụ ứng dụng chạy trên hệ thống đó.

- c- IPSec có thể được thực hiện như một phần mềm ứng dụng.
- d- Câu a và b.

Câu 11. SA là gì?

- a- Là một kết nối dùng IPSec giữa hai máy tính bất kỳ.
- b- Là một quan hệ truyền thông một chiều giữa hai thực thể IPSec.
- c- Là một ứng dụng có chức năng phân tích và đánh giá mức độ an toàn của hệ thống.
- d- Tất cả đều sai.

Câu 12. Đặc điểm của AH:

- a- Có khả năng mật mã toàn bộ dữ liệu trao đổi giữa các thực thể IPSec.
- b- Dùng chữ ký số để xác thực thông tin.
- c- Chế độ vận chuyển chỉ cho phép xác thực dữ liệu giữa hai thiết bị mạng (router) có hỗ trợ IPSec.
- d- Tất cả đều sai.

Câu 13. Giao thức ESP:

- a- Cung cấp cơ chế mật mã và xác thực dữ liệu.
- b- Tiêu đề của ESP gồm hai phần, nằm trước và nằm sau gói IP gốc.
- c- Sử dụng kỹ thuật mật mã đối xứng để bảo vệ dữ liệu.
- d- Tất cả đều đúng.

Câu 14. Quản lý khoá trong IPSec:

- a- Có chức năng tạo ra và phân phối khoá công khai của các đầu cuối IPSec.
- b- Có thể sử dụng PKI cho mục đích quản lý khoá trong IPSec.
- c- Dùng giao thức *ISAKMP* để tạo và phân phối khoá bí mật giữa các đầu cuối IPSec.
- d- Tất cả đều sai.

Câu 15. Đặc điểm của SSL:

- a- Là thành phần của Hệ điều hành.
- b- Cung cấp kết nối an toàn cho tất cả các dịch vụ ứng dụng trên cùng một hệ thống.
- c- Sử dụng mật mã đối xứng để mã hoá dữ liệu.
- d- Tất cả các đặc điểm trên.

Câu 16. Chức năng của giao thức SSL record:

- a- Phân đoạn dữ liệu, nén, tạo mã xác thực, mật mã hoá dữ liệu.
- b- Cung cấp cơ chế đảm bảo tính toàn vẹn và tính bảo mật cho dữ liệu.
- c- Nén dữ liệu để tăng hiệu suất truyền
- d- Tất cả đều sai.

Câu 17. Thủ tục bắt tay (handshake protocol) trong SSL thực hiện các chức năng nào sau đây:

- a- Thiết lập các thông số kết nối giữa client và server.
- b- Trao đổi chứng thực để client nhận được khoá công khai của server và ngược lại,

các khoá này dùng để mật mã dữ liệu trao đổi giữa client và server.

- c- Thay đổi các thông số về thuật toán mật mã.
- d- Câu a và c.

Câu 18. Secure Electronic Transaction (SET):

- a- Là một ứng dụng thương mại điện tử trên nền của IPSec.
- b- Là một giao thức an toàn cho các ứng dụng toán qua mạng.
- c- Dùng mật mã bất đối xứng (RSA) để mật mã hóa thông tin.
- d- Tất cả đều đúng.

Câu 19. Trong một giao dịch trên SET:

- a- Người mua hàng (cardholder) phải có thẻ tín dụng do một ngân hàng có hỗ trợ dịch vụ thanh toán qua mạng phát hành.
- b- Người bán hàng (merchant) phải có quan hệ với ngân hàng phát hành thẻ.
- c- Việc chuyển tiền từ tài khoản của người mua hàng sang tài khoản của người bán hàng được thực hiện theo yêu cầu của người bán hàng mà không cần một thành phần thứ 3 nào.
- d- Việc chọn lựa hàng và quyết định mua hàng phải được thực hiện thông qua giao dịch SET thì mới có ý nghĩa.

Câu 20. Thế nào là chữ ký song song (dual signature)?

- a- Là một chữ ký duy nhất nhưng gồm hai bản sao gửi cho hai đối tác cùng lúc.
- b- Là một chữ ký nhưng gồm hai thành phần, có chức năng chứng thực hai nội dung khác nhau với hai đối tác khác nhau.
- c- Gồm hai chữ ký khác nhau nhưng được ghép chung trong một bản tin để tiết kiệm chi phí truyền trên mạng.
- d- Là một chữ ký nhưng được tạo ra bằng việc áp dụng hàm tạo chữ ký hai lần lên cùng một khối thông tin gốc nhằm đảm bảo tính an toàn của chữ ký.

Câu 21. Thứ tự thực hiện các giao tác trong SET:

- a- Xác thực thanh toán, yêu cầu mua hàng, thực hiện thanh toán.
- b- Yêu cầu mua hàng, xác thực thanh toán, thực hiện thanh toán.
- c- Yêu cầu mua hàng, thực hiện thanh toán, xác thực thanh toán.
- d- Tùy từng trường hợp mà thứ tự thực thi có thể khác nhau.

## **B- Bài tập**

Câu 22. Trong giao thức AH của IPSec, thao tác tạo ra mã xác thực (MAC) không được thực hiện trên toàn bộ gói dữ liệu IP mà chỉ thực hiện trên các phần không thay đổi trong quá trình truyền (immutable) hoặc những phần có thay đổi nhưng có thể đoán được. Hãy chỉ ra trong gói IP (version 4), những phần nào có thay đổi, không thay đổi hoặc thay đổi nhưng đoán trước được trong quá trình truyền?

Câu 23. Ở chế độ vận chuyển của IPSec, một lớp tiêu đề (header) khác của gói IP được tạo ra song song với tiêu đề cũ. Những thành phần nào của tiêu đề mới giống với tiêu đề cũ?

Câu 24. Thực hiện cấu hình IPSec trên Windows 2003 server.

Câu 25. Cài đặt và cấu SSL cho Website trên Windows 2003 server.

-----C3 ✧ 80-----

## HƯỚNG DẪN TRẢ LỜI CÂU HỎI VÀ BÀI TẬP.

### Chương I:

Câu 1.	c	Câu 9.	d	Câu 17.	a	Câu 25.	d
Câu 2.	b	Câu 10.	b	Câu 18.	d	Câu 26.	d
Câu 3.	d	Câu 11.	a	Câu 19.	c	Câu 27.	c
Câu 4.	d	Câu 12.	d	Câu 20.	d	Câu 28.	a
Câu 5.	a	Câu 13.	c	Câu 21.	b	Câu 29.	c
Câu 6.	b	Câu 14.	d	Câu 22.	c	Câu 30.	d
Câu 7.	c	Câu 15.	c	Câu 23.	c	Câu 31.	d
Câu 8.	a	Câu 16.	a	Câu 24.	d		

### Chương II:

Câu 1.	b	Câu 6.	a	Câu 11.	d	Câu 16.	a
Câu 2.	d	Câu 7.	b	Câu 12.	d	Câu 17.	d
Câu 3.	c	Câu 8.	b	Câu 13.	c	Câu 18.	d
Câu 4.	b	Câu 9.	a	Câu 14.	c	Câu 19.	a
Câu 5.	b	Câu 10.	c	Câu 15.	d	Câu 20.	D

Câu 21. Thực hiện thuật toán DES bằng tay. Chú ý khoá phụ là  $K_{16}$ .

Câu 22. Chứng minh tương tự đối với cấu trúc Feistel.

Câu 23.

Câu 24. Thực hiện bằng tay thao tác mở rộng khoá (expand key) của AES.

Câu 25. Thực hiện thuật toán RSA với các thông số tương ứng.

Câu 26. Thực hiện thuật toán Diffie-Hellman.

### Chương III:

Câu 1.	b	Câu 7.	a	Câu 13.	d	Câu 19.	a
Câu 2.	b	Câu 8.	a	Câu 14.	c	Câu 20.	b
Câu 3.	a	Câu 9.	d	Câu 15.	c	Câu 21.	b
Câu 4.	d	Câu 10.	b	Câu 16.	b		
Câu 5.	a	Câu 11.	d	Câu 17.	d		
Câu 6.	d	Câu 12.	d	Câu 18.	b		

## THUẬT NGỮ VIẾT TẮT.

3DES	Triple Data Encryption Standard
AAA	Access Control, Authentication, Auditing
AES	Advanced Encryption Standard
AH	Authentication Header
ANSI	American National Standards Institute
AS	Authentication Server
CBC	Cipher Block Chaining
CC	Common Criteria
CESG	Communications-Electronics Security Group
CFB	Cipher Feedback
CHAP	Challenge Handshake Authentication Protocol
CIA	Confidentiality, Integrity, Availability
CMAC	Cipher-Based Message Authentication Code
CRT	Chinese Remainder Theorem
DAC	Discretionary Access Control
DDoS	Distributed Denial of Service
DES	Data Encryption Standard
DoS	Denial of Service
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
ECB	Electronic Codebook
ESP	Encapsulating Security Payload
FIPS	Federal Information Processing Standard
HMAC	Hash-based Message Authentication Code
IAB	Internet Architecture Board
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IDEA	International Data Encryption Algorithm
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSec	IP Security
ISAKMP	Internet Security Association and Key Management Protocol
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardization Sector
IV	Initialization Vector

KDC	Key Distribution Center
LAN	Local Area Network
MAC	Message Authentication Code
MAC	Mandatory Access Control
MD5	Message Digest, Version 5
MIC	Message Integrity Code
MIME	Multipurpose Internet Mail Extension
MITM	Man-in-the-middle attack
MTU	Maximum Transmission Unit
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NTFS	NT File System
OFB	Output Feedback
PAP	Password Authentication Protocol
PCBC	Propagating Cipher Block Chaining
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PRNG	Pseudorandom Number Generator
RBAC	Role-based Access Control
RFC	Request for Comments
RNG	Random Number Generator
SATAN	System Administrator Tool for Analyzing Network
RSA	Rivest-Shamir-Adelman
SET	Secure Electronic Transaction
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
S/MIME	Secure MIME
SNMP	Simple Network Management Protocol
SNMPv3	Simple Network Management Protocol Version 3
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TGS	Ticket-Granting Server
TLS	Transport Layer Security
UDP	User Datagram Protocol
WAN	Wide Area Network.

### **TÀI LIỆU THAM KHẢO:**

- [1] William Stallings, *Cryptography and Network Security: Principles and Practices*, 4<sup>th</sup> edition, Prentice Hall, 2005.
- [2] Matt Bishop, *Introduction to Computer Security*, Prentice Hall PTR, 2004.
- [3] Mark Stamp, *Information Security: Principles and Practices*, John Wiley & Sons, 2006
- [4] Wenbo Mao, *Modern Cryptography: Theory and Practice*, Prentice Hall PTR, 2003
- [5] Vesna Hasler, *Security Fundamentals for E-Commerce*, Artech House, 2001
- [6] Will Schmied, *Security + Study guide*, Syngress, 2003.

## MỤC LỤC

<b>CHƯƠNG I</b>	<b>TỔNG QUAN VỀ BẢO MẬT HỆ THỐNG THÔNG TIN .....</b>	<b>2</b>
I.1	TỔNG QUAN.....	2
I.2	CÁC ĐẶC TRƯNG CỦA MỘT HỆ THỐNG THÔNG TIN BẢO MẬT .....	3
I.2.1	Tính bí mật: .....	4
I.2.2	Tính toàn vẹn:.....	4
I.2.3	Tính khả dụng:.....	5
I.3	CÁC NGUY CƠ VÀ RỦI RO ĐỐI VỚI HỆ THỐNG THÔNG TIN.....	6
I.3.1	Nguy cơ: .....	6
I.3.2	Rủi ro và quản lý rủi ro:.....	7
I.3.3	Vấn đề con người trong bảo mật hệ thống: .....	8
I.4	NGUYÊN TẮC XÂY DỰNG MỘT HỆ THỐNG BẢO MẬT.....	9
I.4.1	Chính sách và cơ chế: .....	9
I.4.2	Các mục tiêu của bảo mật hệ thống:.....	11
I.5	CHIẾN LƯỢC BẢO MẬT HỆ THỐNG AAA.....	12
I.5.1	Điều khiển truy xuất: .....	12
I.5.2	Xác thực:.....	14
I.5.3	Kiểm tra: .....	16
I.6	CÁC HÌNH THỨC XÂM NHẬP HỆ THỐNG.....	18
I.6.1	Các phương thức tấn công:.....	20
I.6.2	Các phương thức xâm nhập hệ thống bằng phần mềm phá hoại .....	27
I.7	KỸ THUẬT NGĂN CHẶN VÀ PHÁT HIỆN XÂM NHẬP .....	30
I.7.1	Tường lửa: .....	30
I.7.2	Hệ thống phát hiện xâm nhập:.....	33
<b>CHƯƠNG II</b>	<b>MẬT MÃ VÀ XÁC THỰC THÔNG TIN.....</b>	<b>42</b>
II.1	TỔNG QUAN VỀ MẬT MÃ: .....	42
II.1.1	Giới thiệu:.....	42
II.1.2	Các thành phần của một hệ thống mã hoá: .....	42
II.1.3	Các tiêu chí đặc trưng của một hệ thống mã hoá: .....	43
II.1.4	Tấn công một hệ thống mật mã: .....	43
II.2	KỸ THUẬT MẬT MÃ ĐỐI XỨNG:.....	44
II.2.1	Cấu trúc mã khối cơ bản Feistel: .....	45
II.2.2	Thuật toán mật mã DES: .....	49
II.2.3	Thuật toán mật mã Triple DES:.....	55
II.2.4	Thuật toán mật mã AES: .....	57
II.2.5	Các thuật toán mật mã đối xứng khác: .....	63

II.3	KỸ THUẬT MẬT MÃ BẤT ĐỐI XỨNG.....	64
II.3.1	Cấu trúc hệ thống mật mã bất đối xứng: .....	64
II.3.2	Thuật toán mật mã RSA: .....	66
II.3.3	Thuật toán trao đổi khoá Diffie-Hellman: .....	68
II.3.4	Đánh giá kỹ thuật mật mã bất đối xứng: .....	70
II.4	CÁC HÀM BẮM.....	70
II.4.1	Xác thực thông tin: .....	70
II.4.2	Các hàm băm bảo mật: .....	73
II.4.3	Thuật toán băm SHA: .....	74
II.4.4	Thuật toán băm MD5: .....	77
II.5	CHỮ KÝ SỐ.....	77
II.5.1	Nguyên lý hoạt động của chữ ký số: .....	77
II.5.2	Chuẩn chữ ký DSS: .....	80
II.6	QUẢN LÝ KHOÁ.....	83
II.6.1	Quản lý khoá công khai trong mật mã bất đối xứng: .....	83
II.6.2	Sử dụng mật mã bất đối xứng để trao đổi khóa bí mật:.....	84
II.6.3	Cơ sở hạ tầng khóa công khai: .....	85
<b>CHƯƠNG III CÁC ỨNG DỤNG BẢO MẬT TRONG HỆ THỐNG THÔNG TIN .....</b>		<b>93</b>
III.1	GIAO THỨC XÁC THỰC.....	93
III.1.1	Mật khẩu:.....	93
III.1.2	Xác thực trong mô hình điểm-điểm: .....	94
III.1.3	Xác thực trong các hệ thống phân tán: .....	95
III.1.4	Giao thức xác thực Kerberos:.....	98
III.2	IP SECURITY .....	104
III.2.1	Các ứng dụng và đặc điểm của IPSec: .....	104
III.2.2	Cấu trúc IPSec:.....	105
III.2.3	Quan hệ bảo mật:.....	106
III.2.4	Chế độ vận chuyển và chế độ đường hầm:.....	106
III.2.5	AH: .....	107
III.2.6	ESP: .....	110
III.2.7	Quản lý khóa trong IPSec:.....	111
III.3	SECURE SOCKETS LAYER.....	112
III.3.1	Cấu trúc SSL: .....	112
III.3.2	Giao thức truyền dữ liệu SSL:.....	113
III.3.3	Giao thức thay đổi thông số mã:.....	114
III.3.4	Giao thức cảnh báo:.....	114
III.3.5	Giao thức bắt tay: .....	115
III.3.6	So sánh SSL và IPSec: .....	115



III.4	SECURE ELECTRONIC TRANSACTION .....	118
III.4.1	Tổng quan về SET:.....	118
III.4.2	Chữ ký song song:.....	120
III.4.3	Thực hiện thanh toán trong SET: .....	121
HƯỚNG DẪN TRẢ LỜI CÁC CÂU HỎI VÀ BÀI TẬP .....		130
THUẬT NGỮ VIẾT TẮT.....		131
TÀI LIỆU THAM KHẢO.....		133

-----❖-----