PGS, TS. THÁI HỒNG NHỊ
TS. PHẠM MINH VIỆT

An toàn thông tin

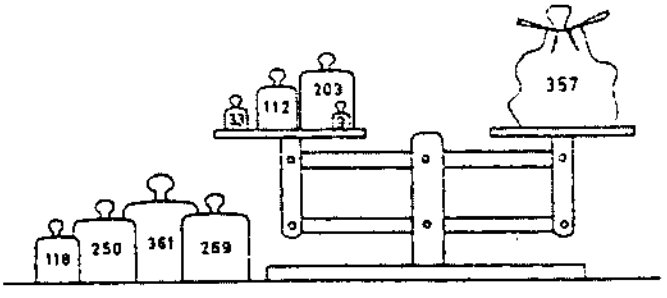
MẠNG MÁY TÍNH, TRUYỀN TIN SỐ VÀ TRUYỀN DỮ LIỆU

NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT

THÁI HỒNG NHỊ - PHẠM MINH VIỆT

AN TOÀN THÔNG TIN

MẠNG MÁY TÍNH, TRUYỀN TIN SỐ VÀ TRUYỀN DỮ LIỆU



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT

LỜI MỞ ĐẦU

Những năm gần đây công nghệ thông tin đã phát triển một cách nhanh chóng với nhiều loại hình dịch vụ phong phú đáp ứng nhu cầu ngày càng cao của con người. Các hệ thống viễn thông hiện đại, các máy tính cá nhân, các mạng máy tính quốc gia, quốc tế và Internet... đã làm cho các quốc gia và các cá nhân trên thế giới xích lại gần nhau hơn.

Thông tin là một nhu cầu không thể thiếu đối với con người và là một dạng tài nguyên đặc biệt vô cùng quí giá. Nói đến thông tin cũng đồng thời nói đến sự giao lưu trao đổi và bảo mật an toàn thông tin trong các quá trình khai thác, xử lý, lưu giữ và truyền tải.

Sách *"An toàn thông tin"* trình bày những nguyên lý cơ bản và các phương pháp thực hiện bảo vệ an toàn thông tin, đặc biệt là trong các hệ thống truyền tin, các mạng truyền dữ liệu và mạng máy tính.

Các nội dung được giới thiệu trong 9 chương:

Chương 1: **An toàn thông tin dữ liệu và các phương pháp bảo vệ**

Chương 2: **Thuật toán DES và mật mã khối**

Chương 3: **Quản lý khoá mã trong mật mã đối xứng**

Chương 4: **Mật mã có khoá công khai**

Chương 5: **Chữ ký số**

Chương 6: **Kiểm tra nhận dạng**

Chương 7: **Xác thực đồng nhất và bảo toàn dữ liệu**

Chương 8: **Bảo vệ thông tin dữ liệu trong mạng truyền tin**

Chương 9: **Internet, TCP/IP và bức tường lửa.**

Sách được biên soạn dựa trên các giáo trình về an toàn thông tin và tham khảo các tài liệu trong và ngoài nước có liên quan. Do nội dung sách bao gồm nhiều vấn đề khá phức tạp về lý thuyết và thực tế ứng dụng cho nên có thể có những thiếu sót về nội dung và thuật ngữ dùng.

Rất mong được sự góp ý của quý vị độc giả.

Xin chân thành cám ơn.

Tác giả

Chương 1.

AN TOÀN THÔNG TIN DỮ LIỆU VÀ CÁC PHƯƠNG PHÁP BẢO VỆ

1.1. TỔNG QUAN VỀ AN TOÀN THÔNG TIN DỮ LIỆU

Ngày nay, khi mà nhu cầu trao đổi thông tin dữ liệu ngày càng lớn và đa dạng, các tiến bộ về điện tử - viễn thông và công nghệ thông tin không ngừng được phát triển ứng dụng để nâng cao chất lượng và lưu lượng truyền tin thì các quan niệm ý tưởng và biện pháp bảo vệ thông tin dữ liệu cũng ngày càng được đổi mới. Bảo vệ an toàn thông tin dữ liệu là một chủ đề rộng, có liên quan đến nhiều lĩnh vực và trong thực tế có thể có rất nhiều phương pháp được thực hiện để bảo vệ an toàn thông tin dữ liệu. Các phương pháp bảo vệ an toàn thông tin dữ liệu đó có thể qui tụ vào ba nhóm sau đây:

1. Bảo vệ an toàn thông tin dữ liệu bằng các biện pháp hành chính.
2. Bảo vệ an toàn thông tin dữ liệu bằng các biện pháp kỹ thuật (cứng).
3. Bảo vệ an toàn thông tin dữ liệu bằng các biện pháp thuật toán (mềm).

Ba nhóm biện pháp trên có thể ứng dụng riêng rẽ hoặc phối kết hợp. Môi trường khó bảo vệ an toàn thông tin dữ liệu nhất và cũng là môi trường đối phương dễ xâm nhập nhất là môi trường truyền tin và mạng truyền tin. Biện pháp bảo vệ an toàn thông tin dữ liệu hiệu quả nhất và kinh tế nhất trên các mạng truyền tin, mạng máy tính là biện pháp thuật toán, do đó các chương trình bày sau đây chủ yếu giới thiệu các biện pháp thuật toán.

Để bảo vệ an toàn thông tin dữ liệu trên đường truyền tin và trên mạng có hiệu quả thì điều trước tiên là phải lường trước hoặc dự đoán trước các khả năng không an toàn, các khả năng xâm phạm, các sự cố rủi ro có thể xảy ra đối với thông tin dữ liệu được lưu giữ và trao đổi trên đường truyền tin cũng như trên mạng. Xác định càng chính xác các nguy cơ nói trên thì càng quyết định được tốt các giải pháp để giảm thiểu các thiệt hại. Mọi nguy cơ đều phải quan tâm và các vi phạm nhỏ thường có xác suất xảy ra cao và các vụ việc có xác suất xảy ra bé đôi khi lại gây nên những thiệt hại khôn lường.

Về bản chất có thể phân loại các hành vi xâm phạm thông tin dữ liệu trên đường truyền tin và mạng truyền tin làm hai loại, đó là: *vi phạm chủ động và vi phạm thụ động*. Chủ động và thụ động ở

đây được hiểu là có can thiệp vào nội dung và luồng thông tin được trao đổi hay không? *Vi phạm thụ động* chỉ nhằm mục đích cuối cùng là nắm bắt được thông tin (đánh cắp thông tin); việc đánh cắp thông tin đó có khi không biết được nội dung cụ thể nhưng có thể dò ra được người gửi, người nhận nhờ các thông tin điều khiển giao thức chứa trong phần đầu các gói tin. Hơn nữa, kẻ xâm nhập có thể kiểm tra được số lượng, độ dài và tần số trao đổi để biết thêm được đặc tính của các đoạn tin. Như vậy, các vi phạm thụ động không làm sai lệch hoặc huỷ hoại nội dung thông tin dữ liệu được trao đổi. *Vi phạm chủ động* là dạng vi phạm có thể làm thay đổi nội dung, xoá bỏ, làm trễ, sắp xếp lại thứ tự hoặc làm lặp lại các gói tin tại thời điểm đó hoặc sau đó một thời gian. Vi phạm chủ động có thể thêm vào một số thông tin ngoại lai để làm sai lệch nội dung thông tin trao đổi hoặc vô hiệu hoá các chức năng trao đổi một cách tạm thời hoặc lâu dài. Vi phạm thụ động thường khó phát hiện nhưng có thể có những biện pháp ngăn chặn hiệu quả trong lúc các vi phạm chủ động dễ phát hiện nhưng biện pháp ngăn chặn có nhiều khó khăn hơn.

Trong thực tế, kẻ vi phạm có thể xâm nhập vào bất kỳ điểm nào mà thông tin đi qua hoặc được lưu giữ. Điểm đó có thể trên đường truyền dẫn, nút mạng, máy tính chủ có nhiều người sử dụng hoặc tại các giao diện kết nối liên mạng (bridge, gateway, router...). Trong quan hệ tương tác người - máy thì các thiết bị ngoại vi đặc biệt là các thiết bị đầu cuối (terminal) là các cửa ngõ thuận lợi nhất cho các xâm nhập. Ngoài ra cũng phải kể đến các loại phát xạ điện từ của các thiết bị điện tử và các máy tính. Bằng các thiết bị chuyên dụng có thể đón bắt các phát xạ này và giải mã chúng. Cũng có trường hợp có thể sử dụng các bức xạ được điều khiển từ bên ngoài để tác động gây nhiễu hoặc gây lỗi nội dung truyền tin.

Trước khi đi vào các giải pháp bảo vệ an toàn thông tin dữ liệu cụ thể, ở đây cần nhấn mạnh một thực tế là: *không có gì là an toàn tuyệt đối cả*. Một hệ thống dù có được bảo vệ chắc chắn đến đâu cũng chưa thể đảm bảo là an toàn tuyệt đối và công việc bảo vệ an toàn thông tin dữ liệu có thể nói là một cuộc chạy tiếp sức không ngừng mà không ai dám khẳng định có đích cuối cùng hay không.

1.2. ĐÁNH GIÁ ĐỘ AN TOÀN BẢO VỆ THÔNG TIN DỮ LIỆU

1.2.1. Tổng quan

Bảo vệ an toàn thông tin dữ liệu là một chủ đề khó mà đánh giá được như thế nào là tối ưu. Vấn đề dẫn đến sự lựa chọn hoặc căn cứ vào tiêu chí tham số nào để đánh giá, ví dụ độ bảo mật, tính hiệu quả, tính kinh tế hoặc độ phức tạp của hệ thống... Một hệ thống được chấp nhận là đảm bảo an toàn thông tin nếu như các nhu cầu về an toàn thông tin dữ liệu về phía gửi tin cũng như phía nhận tin được thoả mãn. Các kết quả tranh chấp được giải quyết theo phương án lựa chọn của cả hai bên. Lý thuyết toán học cũng giải quyết các dạng bài toán trên theo "lý thuyết trò chơi" và ở đây là trò chơi giữa hai người, tổng khác không và thông tin không đầy đủ. Tổng khác không vì giả thiết có xâm nhập và hai người là để đơn giản việc xem xét. Nếu xem xét cùng một lúc nhiều dạng xâm nhập thì bài toán trò chơi trở nên quá phức tạp và hầu như không thể giải được.

Công việc trước tiên của bài toán đánh giá là phải dự kiến hoặc dự đoán những hoạt động và ý đồ của xâm nhập. Thông tin đánh cắp có thể phục vụ cho các mục đích gian trá, tình báo, lợi dụng buôn bán, tống tiền hoặc các mục đích mờ ám khác.

Như vậy, việc đánh giá hiệu năng hoặc độ bảo mật an toàn của một hệ bảo vệ thông tin dữ liệu nói chung là một công việc không đơn giản và không phải lúc nào cũng thực hiện được chính xác như mong muốn vì nó phụ thuộc rất nhiều yếu tố khách quan ngoài hệ thống.

Có thể có rất nhiều phương pháp mà đối phương có thể sử dụng để xâm nhập vào các đường truyền và các mạng truyền tin. Xâm nhập trên các mạng kết nối Internet là một ví dụ. Ngay các mạng an ninh quốc phòng của các nước công nghiệp tiên tiến cũng có những lúc bị xâm nhập.

Cũng chính vì những lý do trên mà việc đánh giá các yếu tố nguy hiểm rủi ro của một hệ truyền tin cũng như hiệu năng của một hệ bảo vệ an toàn thông tin dữ liệu phụ thuộc vào yêu cầu cụ thể của chính chủ sở hữu thông tin dữ liệu đó và cũng chỉ có tính chất tương đối. Cũng có một số hệ bảo mật áp dụng lý thuyết trò chơi nói trên hoặc dùng các biện pháp giả đột nhập để đánh giá độ an toàn của hệ, nhưng cũng chỉ nhận được kết quả tương đối cho phép chấp nhận của người sử dụng.

Cũng không nên có một ảo tưởng về một phương pháp bảo vệ an toàn hoàn chỉnh tuyệt đối. Phương án chỉ được xây dựng cho một hệ thống đường truyền cụ thể hoặc một mạng cụ thể cùng với các yêu cầu cụ thể của nhà khai thác và người sử dụng. Về triết lý, dưới một góc độ nào đó, thông tin dữ liệu được xem là an toàn khi phía gửi tin thừa nhận là an toàn và phía nhận tin không có nghi vấn gì; tuy vậy trong nhiều trường hợp điều đó chưa đảm bảo đủ độ tin cậy.

Thông tin dữ liệu cũng có thể xem là đảm bảo độ an toàn nếu như khi đối phương khai thác được thì đã mất hiệu lực sử dụng nó. Trong các mạng truyền tin lớn, khi có sự tham gia của nhiều người sử dụng, nhiều thành phần cứng, mềm thì việc khẳng định được độ tin cậy là một việc làm khó khăn, tốn kém.

Ở các hệ thống truyền tin số hoặc truyền dữ liệu có sự tham gia của máy tính, một khi tất cả các yếu tố nguy hiểm đã được liệt kê và có phương án đề phòng thì vấn đề còn lại cho việc đảm bảo an toàn thông tin dữ liệu là an toàn phần mềm và con người.

1.2.2. An toàn phần mềm

Sự phức tạp của một hệ thống truyền tin hoặc truyền dữ liệu có sự tham gia của máy tính chủ yếu nằm trong phần mềm của nó. Chính phần mềm đó cũng quyết định phần cứng là đơn giản, phức tạp hoặc tuân theo các chuẩn nào đó. Hiệu lực của phần mềm có thể thay đổi hoặc sửa đổi. Sự mềm dẻo đó là có lợi dưới góc độ khai thác hệ thống nhưng chính đó cũng là điều bất lợi dưới góc độ bảo vệ an toàn thông tin dữ liệu. Một dạng phần mềm xâm nhập bất hợp pháp là các virut tin học. Nhiều loại virut nguy hiểm có thể phá hoại các dữ liệu, phần mềm lưu trữ trong các máy tính hoặc làm tê liệt hoạt động của cả hệ thống.

1.2.3. An toàn về con người

Ngoài những vấn đề về phần mềm như đề cập ở trên, trong bảo vệ an toàn thông tin dữ liệu, có một vấn đề không thể không đề cập đến, đó là con người. Những người tham gia trong bản thân hệ thống, có thể nắm bắt được tất cả các yếu tố bảo vệ an toàn thông tin dữ liệu, các khoá mã đặc biệt là các khoá chủ (ở đây không đi sâu vào chủ đề này).

1.2.4. Sự phát triển của công nghệ và các ảnh hưởng của nó

Ngày nay, công nghệ thông tin được phát triển một cách nhanh chóng thì các nguy cơ xâm nhập thông tin dữ liệu vào các hệ thống truyền tin, các mạng dữ liệu cũng gia tăng. Sự gia tăng xâm nhập và sự phức tạp đó bao gồm cả việc chống lại sự xâm nhập can thiệp của một hệ thống khác. Có thể nhìn thấy điều đó một cách đơn giản là, sự khai thác thông tin phức tạp hơn xuất phát từ một phương thức kết nối truyền tin hoặc sự thay đổi dữ liệu trong quá trình truyền dẫn. Việc gia tăng các phức tạp truyền dẫn, ghép kênh đó kéo theo việc gia tăng các chuẩn mà các chuẩn đó có thể tuỳ ý sử dụng trong các phần mềm và phần cứng. Một ví dụ thấy rõ là, ngày nay người ta có thể mua được một thiết bị đo để dịch và phân tích các chuỗi bit nhị phân trên đường truyền dữ liệu. Việc phân tích đó cho phép một chuyên gia có thể xác định được giao thức (protocol) của lớp kết nối truyền tin và khai thác rút ra nội dung của các đường truyền cũng như dịch các gói của giao thức X25 và các lớp chuyển vận, tập hợp và tài liệu của thủ tục teletex. Với các thiết bị đo như vậy có thể vượt qua nhiều mức giao thức truyền dẫn để đi đến các phán đoán cuối cùng. Kẻ xâm nhập cũng có thông dịch những gì được truyền qua đường truyền và sau đó có thể cài đặt phần mềm phụ, lấy cắp thông tin, làm sai lạc hoặc thêm vào thông tin với khuôn dạng và giao thức đúng. Những công việc đó không phải là dễ dàng nhưng không phải không làm được với các chuyên viên có trình độ.

Như vậy, việc luôn thay đổi phát triển công nghệ mới cũng kéo theo việc gia tăng các xâm nhập. Nếu sử dụng một mạng cục bộ theo kiểu vòng hoặc theo kiểu Ethernet, thì tất cả các dữ liệu được truyền giữa các đầu cuối và các nguồn đều có thể đến bởi các rẽ nhánh đơn giản. Đường Ethernet phân phối tất cả các đoạn tin cho tất cả các đầu cuối. Trong phần lớn các mạng vòng thì tất cả các dữ liệu phải thực hiện một vòng hoàn chỉnh trước khi quay về nơi phát. Một sự rẽ nhánh trên đường dây trong một mạng vòng hoặc một mạng Ethernet cho phép truy nhập tất cả thông tin truyền trên mạng. Ví dụ, có một số tổ chức sử dụng các mạng Ethernet kết nối với nhau để trao đổi chuyển vận các thông tin có độ nhạy cảm khác nhau. Phần mềm có liên quan đến đường Ethernet cách biệt lẫn nhau giữa những người nhận, nhưng một việc thay đổi phần mềm đó có thể nhận được thông tin của một người khác nào đó. Ở đây nếu không có sự bảo vệ phụ trợ, ví dụ mã hoá dữ liệu, thì môi trường mạng như nói trên là không đảm bảo an toàn.

Các đường truyền tin, truyền dữ liệu vô tuyến hoặc thông tin vệ tinh cho phép khả năng truy nhập và truyền thông tin lớn nhưng khả năng thu trộm thông tin cũng rất dễ.

Cũng đã trong một thời gian, người ta chỉ cần bảo vệ phần thông tin dữ liệu mật trong một khối lượng lớn thông tin dữ liệu truyền, bởi vì với khối lượng lớn dữ liệu đó, kẻ xâm nhập cũng đã ngập chìm trong đó.

Ngày nay, dữ liệu trao đổi giữa bên gửi và bên nhận được chuyển mạch gói. Các gói và các đoạn tin có tiêu đề của chúng, do đó càng khó hơn trước khi muốn xâm nhập đến thông tin lựa chọn. Ở mức địa chỉ thì đích đó có thể là một đầu cuối, một con người, một tiến trình tin học hoặc có thể là một điểm truy nhập. Ở các mạng lớn, phần tiêu đề thường là thông tin rõ để nút mạng có thể nhận biết định tuyến. Kẻ xâm nhập có thể dựa vào tiêu đề để tìm cách xâm nhập trường thông tin. Những hệ có độ bảo mật cao, không những trường thông tin được mã hoá mà cả phần tiêu đề cũng được mã hoá. Việc ứng dụng các hệ mật mã phức tạp có quan hệ với tốc độ truyền tin, đặc biệt là ở những hệ thống truyền tin tốc độ cao, cần được tính toán kỹ để đảm bảo hiệu năng của hệ thống.

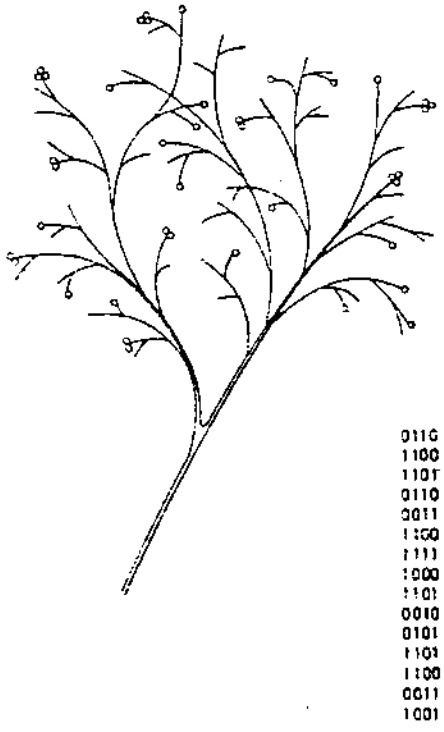
Tóm lại, cùng với sự phát triển công nghệ thông tin hiện đại thì các nguy cơ xâm nhập cũng gia tăng theo.

1.3. MÃ HIỆU MẬT VÀ CÁC ĐẶC TÍNH CỦA MÃ HIỆU MẬT

1.3.1. Lịch sử phát triển mã hiệu mật

Mã hiệu và mã hiệu mật đã được con người sử dụng từ lâu đời. Khoảng 4000 năm về trước, để tỏ lòng tôn kính những người đã chết, người Ai Cập đã khắc những mã hiệu tượng hình lên các ngôi mộ. Các mã hiệu mật được khắc trong các ngôi mộ cổ cho đến nay vẫn đang được các nhà khảo cổ tìm hiểu khám phá. Qua nhiều thế kỷ, các phương pháp mã hiệu và mã hiệu mật cũng đã có nhiều biến đổi.

Ngày nay đã có nhiều phương pháp mã hoá và phương tiện mã hoá hiện đại, tuy vậy một số nguyên lý cơ bản về mã hiệu mật vẫn có giá trị. Người xưa thường che dấu thông tin mã hiệu dưới hai dạng, hoặc bằng phương pháp nào đó che dấu thông tin mà đối phương không phát hiện được hoặc bằng cách biến đổi mã hiệu thành một dạng công khai nào đó nhưng khó nhận biết được nội dung. Thời thượng cổ Hy Lạp các chủ nô lệ đã cạo trọc đầu nô lệ, viết mã hiệu lên đó, đợi tóc mọc và chuyển cho người nhận cạo trọc đầu để đọc. Thời hiện đại cũng đã từng dùng hoá chất đặc biệt để che dấu đoạn tin trao đổi. Sự biến đổi mã hiệu sang một dạng bí mật khác là tiền đề cho các dạng mật mã sau này. Cũng có một phương pháp khác là sử dụng các hình ảnh để che dấu đoạn tin, ví dụ ở hình 1.1, nếu nhìn lướt qua chỉ là một cành cây, nhưng nếu đọc theo chiều kim đồng hồ với giá trị 1 cho nhánh có quả và giá trị 0 cho nhánh không quả thì cuối cùng sẽ có kết quả là một dãy số nhị phân. Nhóm các số 0 và 1 đó thành từng nhóm 4 phần tử sẽ có một đoạn tin theo mã hiệu nhị phân. Giữa các đoạn tin có ý nghĩa có thể chèn thêm các đoạn tin vô nghĩa để làm phức tạp thêm quá trình che dấu thông tin.



Hình 1.1. *Hình vẽ một cành cây che dấu một đoạn tin viết dưới dạng mã nhị phân*

Người ta cũng đã dùng phương pháp viết tắt bằng cách sử dụng hai chất tan chảy khác nhau. Một chất đặc trưng cho bit nhị phân 0 và một chất đặc trưng cho bit nhị phân 1 để mã hoá một đoạn tin mật cài trong một đoạn tin khác không liên quan đến nó. Cũng đã có một thời, một số nhà nghiên cứu mật mã học đã thử tìm xem có phải Bacon là tác giả đích thực các tác phẩm đã trao lại cho Shakespeare, bằng cách phát hiện các đoạn tin được che dấu, nhưng công việc không kết quả và cũng không công bố.

Cũng có một phương pháp mã hiệu mật khác là các đoạn tin được biểu thị bởi các hàng hoá bày trong tủ kính của một cửa hàng. Đoạn tin che dấu cũng có thể viết bằng mực hoá học không nhìn thấy. Tóm lại, có thể nói có vô vàn cách khác nhau để che dấu đoạn tin cần bảo vệ.

Để mã hoá các đoạn tin, người ta cũng có thể sử dụng một từ điển mật mã cho các từ hoặc các nhóm từ. Trong trường hợp này, tất nhiên phía nhận cũng phải có từ điển thứ hai giống như bên phát. Một ví dụ đơn giản cho mã hoá nhóm từ là mã liên lạc quốc tế trong thông tin vô tuyến. Ví dụ:

QRL: Tôi có cần tăng thêm công suất không?

QRH: Tôi có cần thay đổi tần số liên lạc không?

QSD: Anh có bận không?

QRO: Khoá mã của tôi đã đúng chưa?

QTR: Chính xác là mấy giờ?

QRZ: Ai đang gọi tôi đó?

Phương pháp mã hoá dùng từ điển như trên không thích hợp với việc mã hoá dữ liệu có sự tham gia của máy tính, bởi vì có nhiều khó khăn trong việc lưu giữ (do dung lượng) và các bài toán về truy nhập (bài toán về tra cứu). Việc tạo ra được một từ điển là một công việc công phu. Mỗi khi từ điển được lưu giữ thì việc bảo vệ nó cũng là một công việc không kém phần khó khăn.

Có hai kỹ thuật cơ bản sử dụng trong các mã hiệu mật cổ điển, đó là phép thay thế và phép chuyển dịch. Trong phép thay thế, một chữ cái này được thay thế bởi chữ cái khác và trong phép chuyển dịch, các chữ cái được sắp xếp theo một trật tự khác.

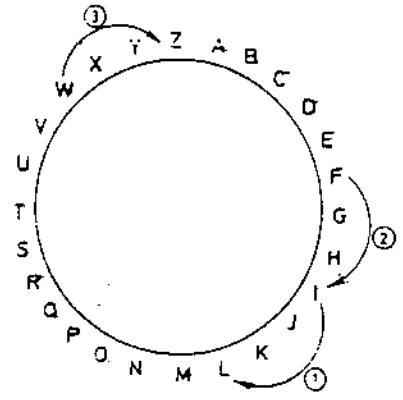
1.3.2. Mã hiệu mật theo phương pháp thay thế đơn giản

Hình 1.2. mô tả một dạng mã hiệu mật theo phương pháp thay thế đơn giản. Ở đây các chữ cái được sắp xếp thứ tự trên một vòng tròn và mỗi ký tự của mã hiệu mật được thay thế bởi một ký tự khác cách nó ba vị trí về phía bên phải

Mã hiệu mật theo phương pháp thay thế đơn giản trên có độ mật rất yếu, nó xuất phát từ trò chơi của học sinh. Các chữ cái được ghi trên chu vi hai vòng tròn đồng tâm có đường kính khác nhau. Việc mã hoá và giải mã được thực hiện bằng cách xoay một trong hai vòng tròn đồng tâm để xem kết quả. Có thể thay đổi số bước thay thế sau mỗi lần thực hiện mã hoá.

Trong phép thay thế đó nếu thay thế các chữ cái (A, B, C...) bằng các số (1, 2, 3...) thì trong bảng 26 chữ cái có thể thực hiện phép toán dưới dạng $C = p + 3$ (modulo 26).

Độ mật của phép thay thế đơn giản trên rất yếu bởi vì chỉ cần thực hiện từ 1 đến 25 lần thay thế là có thể phát hiện ra quy luật mã hoá.



Hình 1.2. *Mô tả một dạng mã hiệu mật theo phương pháp thay thế*

Phép thay thế đơn giản trên có thể thay bằng phép thay thế khác, phức tạp hơn một ít như mô tả ở hình 1.3. Ở đây bảng chữ cái được sử dụng cho phép thay thế, được xem như là khoá mã và khoá giải mã, được sắp xếp một cách ngẫu nhiên không theo quy luật trật tự thông thường.

1.3.3. Mã hiệu mật theo phương pháp thay thế chữ cái đơn.

Hình 1.3. mô tả mã hiệu mật được thực hiện theo phương pháp thay thế xuất phát từ một bảng chữ cái được sắp xếp "không theo trật tự", trong đó đoạn tin rõ sử dụng đoạn tin rõ ở hình 1.2.

Mã hiệu	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Mã hiệu mật	DKVQFIBJWPESCXHTMYAUOLRGZN
Bản tin rõ	IFWEWISHTOREPLACELETTERS
Bản tin đã mã hóa	WIRFRWAJUHYFTDVFSFUUFYA

Hình 1.3. *Phương pháp thay thế ký tự đơn*

Ở đây, việc dịch chuyển vị trí chữ cái phụ thuộc vào chữ cái của bản tin rõ. Kỹ thuật này được gọi là phương pháp thay thế chữ cái đơn bởi vì bảng khoá mã là bảng các chữ cái đơn. Số khả năng mã hoá trong phương pháp thay thế đơn giản là 25 còn số khả năng thay thế chữ cái đơn ở đây là 26! (khoảng 4.10^{26}), một độ mật được gia tăng đáng kể. Tuy vậy cũng có vấn đề nhược điểm của phương pháp, đó là việc phân tích có thể dựa vào phương pháp thống kê và bảng chữ cái thay thế có thể được xây dựng lại theo từ.

Trong ngôn ngữ tự nhiên của các quốc gia dùng bảng chữ cái a, b, c thì mỗi ngôn ngữ có một tần suất xuất hiện nào đó của các chữ cái. Dựa vào tần suất xuất hiện của các chữ cái đó có thể khám phá được quy luật thay thế theo chữ cái đơn dù đó là ngẫu nhiên. Hình 1.4. mô tả tần suất xuất hiện các chữ cái và nhóm chữ cái trong ngôn ngữ tiếng Anh.

	%		%		%
E	13,1	D	4,1	G	1,4
T	9,0	L	3,6	B	1,3
O	8,2	C	2,9	W	1,0
A	7,8	F	2,9	K	0,4
N	7,3	U	2,8	X	0,3
I	6,8	M	2,6	J	0,2
R	6,6	P	2,2	Q	0,1
S	6,5	Y	1,5	Z	0,1
H	5,9	W	1,5		

a) Tần suất xuất hiện các chữ cái trong tiếng Anh

TH	THE
HE	AND
AN	THA
IN	ENT
ER	ION
RE	TIO
ES	FOR
ON	NDE
EA	HAS
TI	NCE

b) Mười nhóm hai chữ cái và ba chữ cái thường xuất hiện trong tiếng Anh.

Hình 1.4. *Tần suất xuất hiện các chữ cái và các nhóm chữ cái thường xuất hiện trong tiếng Anh*

Các giá trị tần suất xuất hiện đó có thể thay đổi chút ít phụ thuộc vào bản tin khác nhau. Người phân tích có thể dựa vào các tần suất xuất hiện các từ trong bản tin mã hoá để phân tích dựa trên việc suy luận các từ nào có tần suất lớn và các từ nào có tần suất bé. Các từ có tần suất xuất hiện trung bình thường ít thông tin cho người phân tích. Các từ và các nhóm chữ cái mô tả trong hình 1.4b là thuộc tần suất xuất hiện lớn trong ngôn ngữ tiếng Anh.

Tần suất xuất hiện các chữ cái trong ngôn ngữ các nước khác nhau cũng sẽ khác nhau, ví dụ phụ âm Q rất ít xuất hiện trong Tiếng Anh và tiếng Đức nhưng xuất hiện nhiều trong tiếng Pháp, tiếng Ý, tiếng Tây Ban Nha; nguyên âm O thường xuất hiện nhiều trong tiếng Anh nhưng rất ít xuất hiện trong tiếng Đức v.v... Biết phân bố tần suất xuất hiện các chữ cái có thể phát hiện được ngôn ngữ nào được viết trong bản tin rõ trước khi mã hoá.

1.3.4. Mã hiệu mật theo phương pháp thay thế cụm các chữ cái

Nhược điểm của phương pháp thay thế chữ cái đơn có thể được cải thiện bằng phương pháp thay thế cụm chữ cái. Mặt khác, để giảm bớt khả năng phát hiện quy luật mã hoá được suy ra từ các từ trong bản tin rõ chỉ có một hoặc hai chữ cái trong ngôn ngữ viết, các bản tin rõ trước khi mã hoá thường được nhóm thành từng nhóm có số chữ cái bằng nhau. Số chữ cái trong mỗi nhóm thường được chọn là 5. Ví dụ đoạn tin rõ:

KHONG CO GI QUY HON DOC LAP TU DO sẽ được nhóm thành

KHONG COGIQ UYHON DOCLA PTUDO

Số thứ tự của các chữ cái trong mỗi nhóm cũng sẽ được chuyển vị theo một trật tự khác. Ví dụ nếu trật tự các chữ cái trong nhóm là 12345 được chuyển vị theo trật tự mới là 41532 thì đoạn tin rõ trên sẽ là:

NKGOH ICQGO OUNHY LDACO DPOUT

Từ đây sẽ tiến hành mã hoá theo phương pháp lựa chọn. Có thể đánh số các ký tự trong bảng chữ cái theo số thứ tự 1, 2... 26 và việc mã hoá và giải mã được tiến hành theo phép toán modulo-26. Ví dụ:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1				5					10					15					20					25	26

F + L = 6 + 12 = 18 = R

T + Q = 20 + 17 = 37 = 37 modul(26) = 11 = K

Hình 1.5 mô tả một đoạn tin rõ được thay đổi trật tự các chữ cái theo nhóm 5 và hình 1.6 mô tả đoạn tin rõ được thay đổi trật tự theo nhóm 8.

Bản tin rõ	THESIMPLESTPOSIBLETRANSPOSITIONSXX
Khóa	4 1 5 3 2
	S T I E H
	E M S L P
	S T S O P
	E I T L B
	S R P N A
	T O I I S
	X O X S N
Bản tin đã mã hóa	STIEHEMSLPSTSOPEITLBSRPNATOIISXOXSN

Hình 1.5. ***Đoạn tin rõ được thay đổi trật tự theo nhóm 5***

Từ khóa	C O M P U T E R
	1 4 3 5 8 7 2 6
	A N O V I N C E
	E W T A O T N Y
	E R P E T S *x* S
	H E P R T U E M
	A O I N Z Z T Z
Bản tin đã mã hóa	ANOVINCEEWTAOTNYERPESXSHEPRTUEMAOINZZTZ

Hình 1.6. ***Đoạn tin rõ được thay đổi trật tự theo nhóm 8 của từ khóa***

1.3.5. Mã hiệu mật theo phương pháp thay thế dòng và cột.

Một phương pháp thay thế có phức tạp hơn các phương pháp thay thế trên là phối hợp giữa dòng và cột. Ở đây bản tin rõ được viết theo dòng và quy tắc của từ khoá. Từ khoá đó được xếp theo cột dọc. Hình 1.7 mô tả đoạn tin rõ được mã hoá theo từ mã LEMON và được xếp theo nhóm 5.

Bản tin rõ	NOWISTHETIMEFORALLGOODMEN

		L	E	M	O	N
		2	1	3	5	4
L	2	O	N	W	S	I
E	1	H	T	E	I	T
M	3	E	M	F	R	O
O	5	L	A	L	O	G
N	4	D	O	M	N	E

Bản tin đã được mã hóa dòng và cột	HTEITONWSIEMFRODOMNELALOG
Bản tin đã được mã hóa đọc theo đường chéo	ONHETWSEMLDAFIITRLOMOOGNE

Hình 1.7. *Ví dụ mã hiệu mật kết hợp dòng và cột*

Việc thực hiện mã hoá theo dòng như trước kia, nhưng ở đây trật tự của việc lựa chọn các dòng được xác định bởi từ khoá theo cột dọc.

1.4. CÁC XÂM NHẬP VÀO CÁC DỮ LIỆU ĐƯỢC MÃ HOÁ

Trước khi mà vai trò của các máy tính được lên ngôi thì việc mã hoá và giải mã các bản tin mật chủ yếu dựa vào tài năng khôn khéo của con người và nó có thể là một bí quyết nào đó. Với tất cả các loại mã cổ điển thì việc phân tích một bản tin đã mã hoá đều có thể thực hiện được bằng cách này hoặc cách khác. Sự ra đời của các máy tính đã giúp cho công việc mã hoá và giải mã tiến một bước khá dài. Máy tính có thể thực hiện các phép tính phức tạp mà bằng các phương pháp khác phải mất hàng năm hoặc hàng chục năm mới thực hiện được.

Một người nào đó nhận được một tín hiệu vô tuyến mà nội dung đoạn tin được mã hoá sẽ gặp phải vấn đề là: hệ thống điều chế có thể phức tạp, ngôn ngữ của bản tin rõ chưa biết và phương thức mã hoá, khoá mã cũng chưa biết. Đoạn tin đó dù rất quan trọng nhưng người đó không thể nhận biết được nếu như công sức để khám phá ra đoạn tin đó vượt quá giới hạn cho phép.

Việc đánh giá độ mật của một phép mật mã thường được giả thiết rằng, thuật toán mã đã biết và vấn đề còn lại là giải mã đoạn tin đã được mã hoá bằng cách khám phá ra khoá mã. Công việc sẽ là khó khăn cho người phân tích mã bởi vì duy nhất chỉ dựa vào bản tin đã được mã hoá, không có thông tin gì về bản tin rõ. Nếu như không có một sự dư thừa nào trong bản tin, thì việc khám phá khoá mã gặp nhiều khó khăn. Nếu như biết được một phần nào đó của bản tin nguyên thuỷ thì bài toán có thể giải được, ví dụ trong bản tin có phần tiêu đề được viết theo chuẩn. Các khoá có thể được thử lần lượt cho đến khi phần tiêu đề được thừa nhận xuất hiện trong bản tin được giải mã. Nếu như phần tiêu đề theo chuẩn đó khá dài thì việc nhận dạng khoá càng chính xác.

Nếu tiêu đề ngắn thì có thể tập hợp nhiều bản được giải mã để lựa chọn các khoá có khả năng. Việc biết mã hoá sử dụng cho bản tin rõ, bao gồm ví dụ như bit kiểm tra chẵn lẻ cũng là một yếu tố có ích cho người phân tích mã. Cũng vì vậy mà trong nhiều trường hợp khoá mã không sử dụng bit kiểm tra chẵn lẻ.

Nếu như người phân tích mã không những chỉ có bản tin đã mã hoá, mà có cả bản tin rõ tương ứng thì việc khám phá khoá mã sẽ gặp nhiều thuận lợi. Công việc ở đây chỉ còn là khám phá

khoá mã tương ứng cho bản tin rõ và bản tin đã mã hoá. Nếu chiều dài bản tin khá đủ, thì khoá mã có thể nhận dạng với độ chính xác tuyệt đối.

Khi muốn tin chắc chắn vào độ bảo mật của mật mã đã thực hiện thì thường đặt giả thiết là kẻ xâm nhập có những yếu tố thuận lợi nhất để khám phá. Cụ thể là, kẻ xâm nhập có thể biết thuật toán thực hiện và chúng có thể có một số lượng khá đủ các bản tin rõ và bản tin đã được mã hoá tương ứng. Nói cách khác là giả thiết kiểu xâm nhập theo dạng bản tin rõ đã biết để khám phá khoá mã. Điều đó có nghĩa là để đánh giá một cách tốt nhất độ bền vững hoặc độ mật của mật mã cần phải tính đến trường hợp xấu nhất là kẻ xâm nhập có bản tin rõ và bản tin đã mã hoá tương ứng. Đó lã điều hoàn toàn có thể xảy ra trong thực tế. Có thể hình dung rằng, một kẻ xâm nhập có thể, ví dụ như, chèn một đoạn tin riêng vào trong đường truyền, và sau đó bằng cách này hoặc cách khác, có thể thu lại đoạn tin rõ đó tương ứng. Cũng chính vì vậy mà một mật mã tốt phải tính đến hết tất cả các khả năng xâm nhập.

Nếu như kẻ xâm nhập thực hiện một công việc tìm kiếm tất cả các khoá mã có thể sử dụng trong một máy tính theo phương pháp cặp "bản tin rõ - bản tin đã mã hoá" thì chúng ta có thể đánh giá thời gian cần thiết để tìm khoá mã xuất phát từ thời gian cần thiết để thử một khoá mã và tổng số khoá mã có thể. Hình 1.7 mô tả thời gian cần thiết để tìm khóa mã ứng với chiều dài của khóa và tốc độ phép thử.

Chiều dài khoá mã (bit)	Phép thử đơn giản		10^6 phép thử song song	
	1ms	1μs	1ms	1μs
24	2,33h	8,4s	8,4ms	8,4"s
32	24,9 d	35,8d	2,15s	2,15ms
40	17,4y	6,4m	9,2d	550ms
48	>100y	4,46y	1,64m	2,45h
56		>100y	1,14y	10,0d
64			>100y	107d

d = ngày; *m = tháng;* *y* = năm

Hình 1.7. ***Mô tả vùng khóa tốt (dưới nét đậm) có thể sử dụng***

1.5. CÁC MỐI ĐE DỌA ĐỐI VỚI MỘT HỆ THỐNG ĐƯỢC BẢO MẬT

1.5.1. Các xâm nhập thụ động

Các mối đe doạ đối với toàn bộ hệ thống được bảo mật thuộc một hệ thống truyền tin hoặc một hệ thống lưu trữ dữ liệu như đã biết có thể phân làm hai loại: các đe doạ dạng tích cực và các đe doạ dạng thụ động. Một xâm nhập thụ động đối với một hệ thống là các mưu toan vượt qua hàng rào bảo vệ để thu những thông tin dữ liệu truyền trên kênh truyền hoặc lưu giữ trong bộ nhớ mà không làm sai lạc các nội dung thông tin dữ liệu. Loại xâm nhập này đối với các hệ thống truyền tin

đã xuất hiện từ thời phát minh ra điện báo. Trước khi có việc ghép kênh thì việc thu trộm tin trên đường truyền tin rất đơn giản. Việc nghe trộm điện thoại một đường kết nội cục bộ cũng được thực hiện rất dễ dàng. Ngày nay trong các mạng truyền dữ liệu với các dạng ghép kênh khác nhau và các giao thức phức tạp (ví dụ mạng chuyển mạch gói đồng bộ STM, chuyển mạch gói không đồng bộ ATM, mạng số đa dịch vụ ISDN) thì việc thu trộm hoặc phá hoại thông tin dữ liệu trên đường truyền có nhiều khó khăn phức tạp hơn, cần có thiết bị đặc biệt và chuyên gia kỹ thuật. Cũng cần lưu ý rằng, các thiết bị đặc biệt đó có thể kiếm rất dễ trên thị trường và chuyên gia biết kỹ thuật đó cũng không hiếm.

Cũng có người từng cho rằng, việc ghép kênh phức tạp có thể bảo vệ chống lại các xâm nhập, điều đó là sai lầm và nguy hiểm.

Một sự rẽ nhánh thụ động không cần phải lọt qua các giao thức kiểm tra dòng dữ liệu. Người ta có thể kiểm tra phát hiện một sự rẽ nhánh thụ động trên đường truyền vật lý bằng cách đo chính xác các đặc tính kỹ thuật của đường truyền. Nhưng việc đo lường các thông số đó không thể thực hiện được trong trường hợp đường truyền kết nối vô tuyến. Để bảo vệ chống lại các xâm nhập thụ động trong các trường hợp này chỉ có cách dùng phương pháp mã hoá để bảo vệ dữ liệu truyền.

1.5.2. Các xâm nhập tích cực

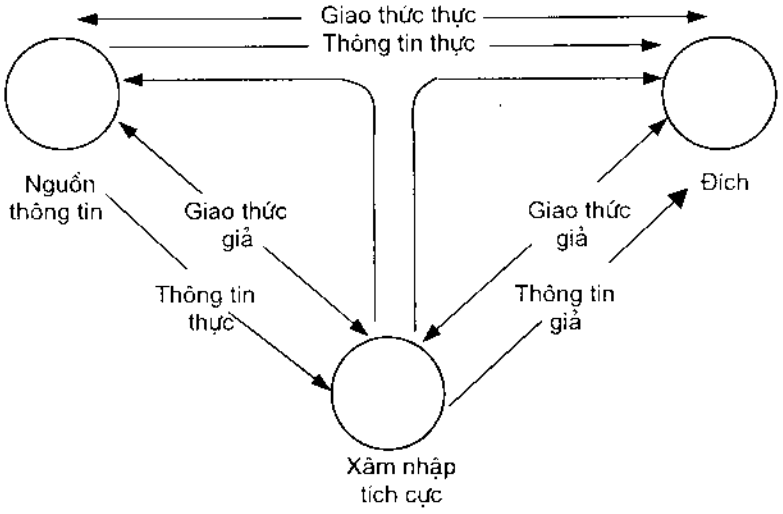
Các xâm nhập tích cực là một mối nguy hiểm tiềm tàng. Ở đây, kẻ xâm nhập tìm cách làm sai lạc các dữ liệu truyền hoặc các dữ liệu lưu trữ và hy vọng rằng, chủ sở hữu hoặc người sử dụng hợp pháp không nhận biết. Việc làm sai lạc các dữ liệu được lưu giữ có thể gây ra các sử dụng sai lạc của các đường truy nhập. Ở đây sẽ không đề cập đến các thủ tục truy nhập, mặt dù đó là một chủ đề quan trọng, mà chỉ đề cập đến vấn đề bảo vệ dữ liệu chống lại các xâm nhập bằng các phương pháp ám muội. Có thể sử dụng phương pháp mật mã để bảo vệ chống lại các xâm nhập tích cực làm sai lạc dữ liệu. Để làm được điều đó thì cần phải có một cấu trúc mã sao cho tất cả các việc làm sai lạc cấu trúc đó không thể thực hiện được nếu không phân tích được mã. Vấn đề sẽ được thảo luận chi tiết ở các chương tiếp theo. Một khi mà kẻ xâm nhập sử dụng phương tiện truy nhập bất hợp pháp vào một cơ sở dữ liệu thì việc phá hoại các dữ liệu không thể tránh khỏi. Đó là một chủ đề lớn thuộc lĩnh vực bảo vệ dữ liệu lưu giữ.

Một sự xâm nhập tích cực trên đường truyền tin hầu như cũng khó ngăn chặn giống như các xâm nhập thụ động. Tuy vậy việc phát hiện chúng thì dễ hơn nhiều, không cần phải mã hoá. Việc làm sai lạc các dữ liệu lưu trữ cần phải có thời gian và độ chính xác của thời gian truyền các dữ liệu cũng có thể giúp phát giác các xâm nhập.

Ở các đường truyền tin vô tuyến thì rất khó phát giác việc xâm nhập, và đây thực sự là một cuộc đấu trí. Nếu như một đường truyền vật lý được giám sát chặt chẽ và thường xuyên, lưu lượng truyền được kiểm tra thường xuyên thì lúc đó hầu như không một xâm nhập nào không được phát hiện. Hình 1.8 mô tả dạng xâm nhập tích cực.

Việc thực hiện một xâm nhập rẽ nhánh tích cực là một công việc không đơn giản. Ở hình 1.8 việc truyền tin giữa nguồn và đầu cuối được điều khiển bởi một giao thức được gọi là "giao thức thực". Việc xâm nhập tích cực phải ngắt giao thức đó và đưa vào một "giao thức giả". Như vậy các

thông tin thực sẽ từ nguồn về kẻ xâm nhập và các thông tin giả từ kẻ xâm nhập về đầu cuối mà đầu cuối không nhận biết được. Với một số biến đổi, việc xâm nhập tích cực như trên có thể thiết lập được với kênh truyền tin. Nếu mạng truyền tin là mạng diện rộng, hoặc nối mạng Internet thì việc thiết lập xâm nhập tích cực trên càng có điều kiện, bởi vì các giao thức truyền tin đã được công bố.



Hình 1.8. *Mô tả một xâm nhập tích cực*

1.5.3. Các phương pháp bảo vệ

Việc bảo vệ các dữ liệu truyền chống lại các xâm nhập tích cực cũng dựa trên các nguyên lý giống như bảo vệ các dữ liệu lưu giữ. Nó có thể ngăn ngừa việc làm sai lạc, thêm vào, phá hoại dữ liệu hoặc các trò chơi của kẻ xâm nhập. Trong các khối dữ liệu truyền thì việc làm sai lạc, việc thêm vào hoặc phá hoại dữ liệu được mã hoá, có nghĩa là toàn bộ khối dữ liệu mã hoá phụ thuộc vào khối dữ liệu rõ tương ứng cũng như cả khối dữ liệu rõ trước nó. Một sự xâm nhập tích cực sẽ không thực hiện được nếu không tìm được khoá mã theo kiểu thám mã. Vấn đề sẽ thảo luận chi tiết ở các chương sau.

Về trò chơi là một dạng xâm nhập mà kẻ xâm nhập ghi lại các dữ liệu mã hoá đã được truyền rồi sau đó lại sử dụng dữ liệu đó cho xâm nhập tích cực. Mục đích của chúng, ví dụ truyền nhiều lần cho một tài khoản ở ngân hàng. Có thể ngăn ngừa bằng các phương pháp xác thực và nhận dạng. Vấn đề sẽ chi tiết ở các phần sau.

1.6. CÁC KHOÁ MÃ

Trong tất cả các hệ thống mật mã (trừ các loại mã hiệu mật sơ đẳng) được đề cập trong chương, khoá mã đóng một vai trò đặc biệt quan trọng. Thuật toán mã hoá không dùng khoá mã rất dễ phán đoán.

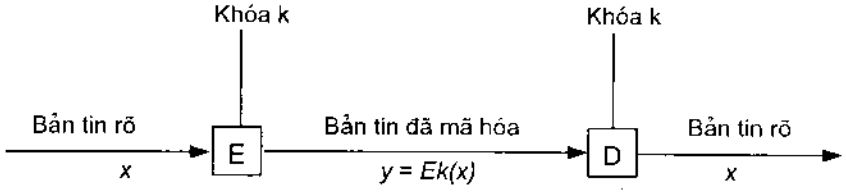
Trong các thuật toán có sử dụng khoá mã thì quá trình mã hoá được thực hiện dưới sự điều khiển của khoá mã, nó làm gia tăng độ hoàn thiện của thuật toán. Tính chất đó được ứng dụng giống nhau trong các phép mã theo phương pháp thay thế, chuyển vị hoặc hỗn hợp. Điều đó đặc biệt quan trọng khi dùng các thiết bị để thực hiện mã hoá.

Cũng cần lưu ý rằng, độ mật của mật mã không phải ở thuật toán mà ở khoá mã. Mọi thuật toán đối phương có thể có. Chính vì vậy mà một trách nhiệm lớn lao đối với người quản lý hệ thống trong việc tạo khoá, phân phối khoá, sử dụng khoá và huỷ khoá sau khi sử dụng. Trong quá khứ người ta sử dụng giao liên riêng để vận chuyển các khoá mã đến người sử dụng mật mã nhưng phương pháp đó không được thích hợp lắm trong điều kiện truyền tin hiện đại. Cần phải có một phương pháp vận chuyển khoá nhanh hơn, hiệu quả hơn. Điều đó dẫn đến một vấn đề là các khoá mã truyền trên các kênh truyền tin cần phải được bảo vệ cũng chính bằng mật mã bảo vệ khoá. Do đó cần phải có sự phân cấp các khoá, trong đó phân biệt các khoá sử dụng để bảo vệ dữ liệu (được sử dụng rộng rãi) và các khoá để bảo vệ các khoá truyền. Các khoá bảo vệ khoá truyền không sử dụng thường xuyên cho nên có thể chuyển vận bằng các phương tiện vật lý.

1.7. CÁC KÝ HIỆU SỬ DỤNG TRONG PHÉP MẬT MÃ

Các mã hiệu và các mã hiệu mật là các cấu thành của mật mã, là nghệ thuật che dấu thông tin theo một phương pháp mật để đảm bảo bí mật thông tin. Có hai phép toán thực hiện trong mật mã là phép "mã hoá" và phép "giải mã".

Mục đích của mã hoá là che dấu thông tin trước khi truyền trên kênh truyền. Có nhiều phương pháp mật mã khác nhau, tuy vậy tất cả chúng có thể biểu thị bởi mô hình tổng quát như mô tả ở hình 1.9. Có thể biểu thị phép toán mã hoá và phép toán giải mã như các hàm của hai biến số, hoặc có thể như một thuật toán, có nghĩa là một thủ tục đối xứng để tính kết quả khi giá trị các tham số đã cho.



Hình 1.9. *Các ký hiệu của phép mã hoá và phép giải mã*

Bản tin rõ ở đây là tập hợp các dữ liệu trước khi thực hiện mã hoá. Kết quả của phép mã hoá là bản tin đã được mã hoá. Việc giải mã bản tin đã mã hoá sẽ thu được bản tin rõ ban đầu. Các biểu thức "bản tin rõ" và "bản tin đã mã hoá" đều có liên quan đến một mật mã cụ thể. Các chữ cái viết hoa *D* (decipherment) và *E* (encipherment) là ký hiệu cho các hàm giải mã và mã hoá tương ứng. Ký hiệu *x* là bản tin rõ và *y* là bản tin đã mã hoá thì biểu thức toán học của phép mã hoá là:

$$y = Ek(x)$$

và của phép giải mã là

$$x = Dk(y)$$

trong đó tham số phụ *k* là khoá mã.

Khoá mã là một đặc tính quan trọng của thuật toán mật mã. Về nguyên lý, nếu hàm $y = E(x)$, không có một khoá mã nào, thì cũng có thể che dấu được giá trị của *x*.

Ký hiệu $Ek(x)$ là ký hiệu được sử dụng phổ biến. Trong một số tài liệu có ký hiệu $E_k(x)$, với k là chỉ số của E nói lên rằng chỉ số k là hằng số trong phép mã. Cũng có một số ký hiệu khác, ví dụ ký hiệu $\{..\}^k$ biểu thị các dữ liệu hoặc bản tin trong dấu ngoặc được mã hoá với khoá k. Trong sách này sử dụng ký hiệu $y = Ek(x)$ và $x = Dk(y)$.

Tập hợp các giá trị có thể của khoá k được gọi là "không gian các khoá". Trong một mật mã nào đó, nếu khoá mã có 20 số thập phân sẽ cho không gian các khoá là 10^{20}. Nếu khoá mã nào đó có 50 số nhị phân thì không gian các khoá sẽ là 2^{50}. Nếu khoá là một hoán vị của 26 chữ cái A, B, C... Z thì không gian các khoá sẽ là 26!.

Trong ký hiệu $y = Ek(x)$ thì x và y cũng xem như các biến số đơn giản, nhận các giá trị trong một tập hữu hạn. Nếu x đặc trưng cho một khối của bản tin thì mật mã được gọi là mật mã khối (block cipher). Ví dụ tám ký tự của mã ASCII-7 bit có thêm bit kiểm tra chẵn lẻ tạo nên một mã khối 64 bit.

Với các mật mã có độ dài đoạn tin không xác định thì lúc đó biến số x được thay thế bởi một chuỗi các biến số và bản tin được mã hoá sẽ là:

$$Ek(x_0,x_1,...x_i,...)$$

trong đó x_0, x_i là chuỗi các biến số mà kích cỡ của nó được xác định bởi bản tin.

1.8. ĐẶC TÍNH CỦA CÁC HÀM MẬT MÃ

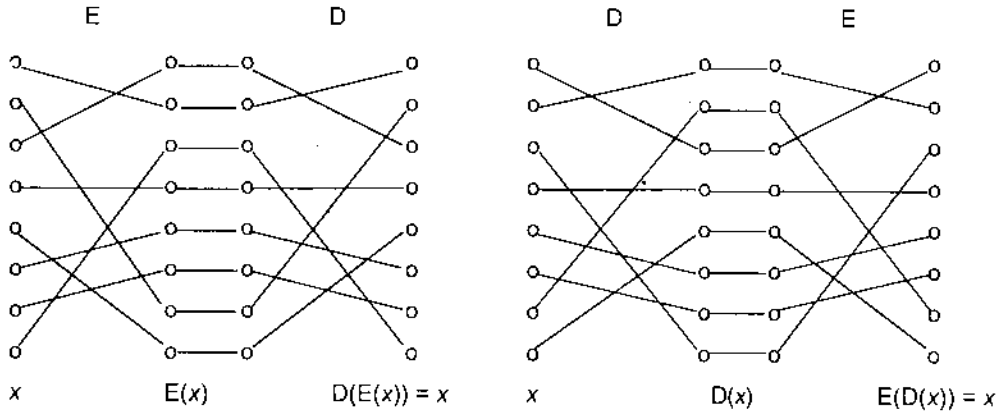
1.8.1. Các đặc tính chung

Các mật mã thông dụng thường là các mã không kéo dài thêm bản tin. Một mật mã dạng khối xử lý các khối n bit, tạo mỗi một giá trị trong $2n$ giá trị khác nhau của bản tin rõ thành một trong $2n$ giá trị khác nhau của bản tin đã mã hoá. Nó có thể là một sự hoán vị của $2n$ giá trị đó biến đổi với giá trị của khoá mã. Số các hoán vị có thể là $(2n)!$, số hoán vị đó không thể sử dụng hết vì nó lớn hơn không gian các khoá. Trong thực tế, các hoán vị thường được chọn một cách ngẫu nhiên, bởi vì nếu có tính quy luật thì kẻ thám mã có thể khai thác.

Với một giá trị của khoá k, nếu x đi qua tập hợp của tất cả các giá trị có thể thì $y = Ek(x)$ cũng đi qua tập hợp của tất cả các giá trị có thể. Mặt khác nếu giữ x là hằng số và k đi qua không gian toàn bộ các khoá thì các giá trị của y được tạo ra có thể xem như là sự lựa chọn không chắc chắn. Một số giá trị có thể không xuất hiện từ toàn bộ trong lúc có một số các giá trị khác lại xuất hiện nhiều lần.

Khi mà không có sự mở rộng các dữ liệu, thì đẳng thức $x = Dk[Ek(x)]$ có quan hệ với một đẳng thức khác là $x = Ek[Dk(x)]$. Từ đó có thể nói rằng, phép mã hoá và phép giải mã có thể trao đổi lẫn nhau. Hình 1.10 mô tả quan hệ trao đổi lẫn nhau giữa hàm mã hoá và hàm giải mã.

Trong trường hợp có sự mở rộng dữ liệu, lúc đó $Dk(x)$ sẽ giảm kích cỡ của bản tin và biểu thức thứ hai nêu trên không còn đúng. Hàm được mô tả ở hình 1.10 là thuộc dạng hàm song ánh xạ. Một hàm như vậy áp dụng một tập hợp các giá trị trên một tập hợp có cùng kích cỡ. Mỗi một giá trị chỉ liên kết với một và chỉ một giá trị, bởi một hàm hoặc bởi một hàm nghịch đảo. Đó là dạng *hàm một - một*.



Hình 1.10. *Sự trao đổi lẫn nhau giữa hàm mã hoá và hàm giải mã.*

Cũng có dạng mật mã được gọi là "cuốn chiếu", có nghĩa là một hàm f như là $f[f(x)] = x$. Trong trường hợp hàm mật mã là cuốn chiếu thì

$$Ek[Ek(x)] = x$$

ám chỉ rằng $Ek(x) = Dk(x)$, tức phép mã hoá và phép giải mã là đồng nhất. Ở thời đại mà các mật mã được thực hiện qua các máy mã thì mật mã theo kiểu "tự động - thuận nghịch" đó tỏ ra có nhiều ưu điểm, bởi vì không cần phải có các kết cấu khác nhau cho bên phát và bên thu. Tích của hai cuốn chiếu không nhất thiết là một cuốn chiếu, như mô tả ở hình 1.11. Tuy vậy nghịch đảo của tích đó có thể dễ dàng tìm được nếu như các thành phần được biết.

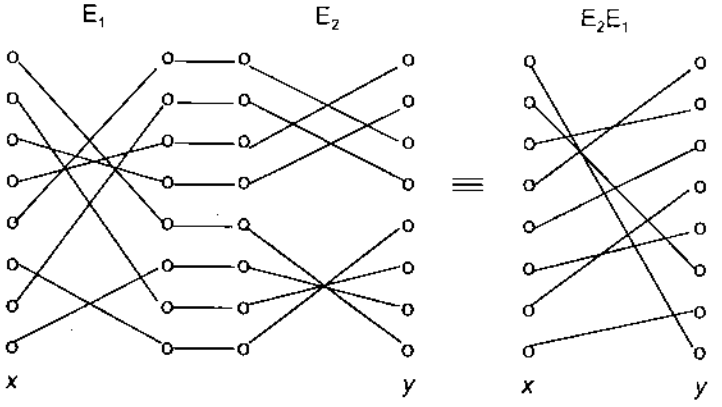
Nếu phép mã hoá là

$$y = E_2[E_1(x)]$$

và nếu E_1 và E_2 đều là những cuốn chiếu thì phép giải mã là

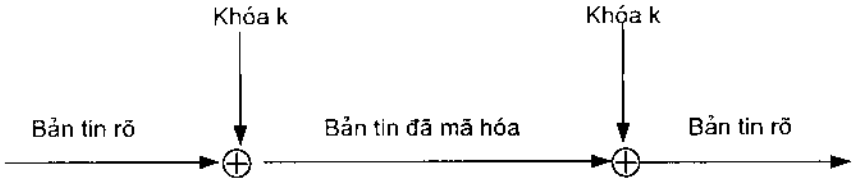
$$x = E_1[E_2(y)]$$

Một loạt các biến đổi cuốn chiếu có thể tạo ra một mật mã rất dễ nghịch đảo. Nguyên lý đó được sử dụng trong mật mã DES sẽ được trình bày ở chương sau.



Hình 1.11. *Tích của hai phép "cuốn chiếu"*

Phép cộng modul-2 cũng được sử dụng phổ biến trong các phép toán mã hoá và giải mã. Theo đại số logic, biết rằng nếu cộng modul-2 hai lần với cùng một giá trị sẽ cho kết quả ban đầu. Ký hiệu của phép cộng modul - 2 là ⊕. Hình 1.12 mô tả một khâu đơn giản của ứng dụng phép cộng modul-2.



Hình 1.12. *Dòng dữ liệu ứng dụng phép cộng modul-2*

1.8.2. Các phép thay thế và chuyển vị trong mật mã hiện đại

Trong các mật mã hiện đại, các phép thay thế thay một "từ" n bit bằng một từ khác dựa vào một hàm toán học (logic hoặc đại số) hoặc bằng cách sử dụng một hàm ngẫu nhiên chứa trong một bảng các giá trị. Phép thay thế các chữ cái như trong các mã cổ điển là sử dụng phép cộng modul-26. Trong các chương sau sẽ giới thiệu các hàm mũ với phép modulo cải tiến hơn. Phương pháp thay thế đơn giản nhất là thực hiện việc thay thế cho bởi một bảng các giá trị ngẫu nhiên. Ví dụ, một phép thay thế cho 8 bit. Có 256 giá trị có thể và với một giá trị đặc trưng cho 8 bit đầu ra. Điều đó có thể thực hiện với 256 byte của bộ nhớ của máy tính. Nếu sử dụng lợi thế không gian địa chỉ của máy tính thì có khả năng có các bảng thay thế rất lớn. Trong trường hợp mà bảng thay thế phụ thuộc vào khoá thì thời gian nạp cần phải tính đến.

Thông thường người ta sử dụng phép thay thế không thể nghịch đảo (hoặc song ánh xạ), điều đó có nghĩa là, ví dụ một tập hợp 256 giá trị các byte sẽ tạo nên một sự hoán vị các số từ 0 đến 255. Cũng có thể có những phương pháp khác hiệu quả hơn để tạo ra các hoán vị như vậy.

Ví dụ, giả thiết $y = S(x)$ là phép thay thế, trong đó x và y nhận tất cả giá trị nguyên từ 0 đến 255. Tại điểm xuất phát: $S(x) = x$. Sau đó sử dụng khoá mã hoặc một phần của khoá, xem như là *"mầm khoá"* để tạo ra một chuỗi số nguyên ngẫu nhiên $R(i)$, trong đó $i = 0, 1, ..., 255$ với $0 \leq R(i) \leq 255$. Các số ngẫu nhiên đó được sử dụng để trao đổi giao hoán $S(i)$ với $S[R(i)]$. Theo phương pháp đó, mỗi một phần tử của $S(x)$ được hoán vị với một phần tử khác được chọn một cách ngẫu nhiên trong bảng. Quá trình hoán vị đó có thể được lặp lại để lại nhận được một kết quả ngẫu nhiên. Các phép thay thế có thể thực hiện bằng phần mềm hoặc cài sẵn trong phần cứng. Các phép chuyển vị thường được thực hiện trên các thanh ghi dịch chuyển. Việc truyền các nội dung trong các phép toán số học có thể khai thác các hàm có sẵn trong máy tính để tạo các chuyển vị, tuy vậy chiều dài trung bình của một chuỗi nội dung khá nhỏ, khoảng $\log_2 n$ bit cho một số có n bit.

Các phép chuyển vị theo phương pháp cổ điển (chuyển vị theo chữ cái hoặc theo bit) có thể khó thực hiện bằng phần cứng nếu như chúng biến đổi hoặc phụ thuộc vào một khoá mã. Nếu như tất cả các hoán vị khả dĩ được thực hiện thì có thể bố trí giống như kiểu một chuyển mạch điện thoại và điều đó có thể thực hiện dễ dàng bằng phần mềm. Tuy nhiên cần lưu ý rằng việc chuyển vị

các bit luôn có khuynh hướng chậm, ảnh hưởng đến tốc độ truyền tin, bởi vì các phương pháp dựa trên các phép cộng, phép dịch chuyển là phép cộng modul-2 cho các từ số nguyên.

1.8.3. Lý thuyết Shannon về các hệ thống mật

Một trong những đóng góp quan trọng của *Claude Shannon* đối với lý thuyết các hệ thống mật mã là một số công trình được công bố trong *"lý thuyết truyền tin về các hệ thống mật"* (communication theory of secrecy systems). *Shannon* đồng nhất hai loại hệ thống mật mã: loại mật không điều kiện và loại mật có thể tính toán được. Một hệ thống mật không điều kiện được định nghĩa như một hệ thống chống tất cả các loại thám mã, ngay cả khi giả thiết rằng người thám mã có năng lực tính toán vô biên. Tất cả các bản tin có chiều dài đúng bằng chiều dài của bản mã có khả năng là bản tin rõ.

Một bản mật mã cụ thể có thể là mật không điều kiện, nếu như nó đủ ngắn để cho thông tin nội dung không đủ để xác định một lời giải chung. *Shannon* định nghĩa chiều dài tối thiểu của bản tin cần thiết để dẫn đến một lời giải chung xem như là "khoảng cách đơn vị". Tiêu chuẩn của *Shannon* là độ dư thừa của bản tin rõ phải lớn hơn thông tin nội dung trong khoá mã. Ví dụ, kích cỡ của khoá đối với phép thay thế chữ cái đơn là 26! và $\log_2(26!)$ có giá trị khoảng 88. Thừa nhận độ dư thừa của ngôn ngữ tiếng Anh là 80%, như vậy mỗi ký tự góp phần 3,8 bit dư thừa và toàn bộ bản tin được mã hoá có hơn 88/3,8 = 23 ký tự có thể sử dụng để xác định khoá mã. Shannon đánh giá độ dư thừa của tiếng Anh bao hàm các khoảng trống và việc thám mã bản tin không có các khoảng trống cần thiết cho việc phân tích. Tuy nhiên kết quả đó là gần giới hạn quan sát đối với việc thám mã theo kiểu thủ công.

Một bản tin được mã hoá chứa đầy đủ thông tin để có thể giải mã là loại mật có thể tính toán được nếu như có các phương tiện để tính toán. Các hàm một chiều (sẽ được phân tích ở phần sau) là các hàm được tính theo một chiều và rất khó hoặc không thể tính theo chiều ngược lại.

Thực tế cũng không có định nghĩa nào thừa nhận là không thể tính toán được. Vấn đề ở đây là thời gian cần thiết để đạt được lời giải của phép tính. Ở đây cũng cần lưu ý rằng, công nghệ tính toán và tốc độ xử lý các máy tính trong đó có các phép xử lý song song luôn được cải tiến phát triển.

Cũng đã có những ước tính thử về thời gian để giải các bài toán khó. Một giả thiết là, mỗi phép logic tiêu thụ một năng lượng là kT, trong đó k là hằng số *Boltzmann* và T là nhiệt độ tuyệt đối. Căn cứ và phép tính số lượng nhiệt mà quả đất nhận được từ mặt trời và so sánh với các phép tính thám mã cần phải thực hiện là ở một nhiệt độ 100K. Như vậy số các phép tính cơ bản được ước tính là khoảng 3.10^{48}, một khối lượng rất lớn các phép tính mà với một máy tính tốc độ nhanh cũng phải thực hiện trong nhiều năm.

Có thể hình dung là, việc thám mã mật mã Lucifer với khoá mã có chiều dài là 128 bit. Giả thiết rằng chỉ có thể xâm nhập bằng cách khám phá khoá mã và cho biết cặp đoạn tin đã được mã hoá và đoạn tin rõ tương ứng. Như vậy số phép toán cần phải thực hiện để thử các khoá là khoảng 3.10^{38}. Cho rằng mỗi một khoá Lucifer có thể thử tất cả các pico-giây, như vậy thời gian để tìm khoá cũng phải khoảng 10^{19} năm.

1.9. ĐIỂM YẾU CỦA PHẦN MỀM TRONG BẢO VỆ AN TOÀN THÔNG TIN DỮ LIỆU

Trong bảo vệ an toàn thông tin dữ liệu, nếu như các yếu tố về vật lý và hành chính được thiết kế đầy đủ, thì vấn đề còn lại là nằm trong phần mềm và người sử dụng khai thác hệ thống. Sự phức tạp của một hệ thống thông tin cũng hoàn toàn nằm trong phần mềm. Sự phát triển của phần mềm cho phép thiết kế phần cứng ngày càng đơn giản cũng như xây dựng các chuẩn, ví dụ các bộ vi xử lý và các bộ nhớ. Một đặc tính ưu việt của phần mềm, như tên gọi của nó là tính mềm dẻo, linh hoạt, nó có thể được thay đổi, cải tiến trong và sau khi phát triển tuỳ thuộc vào các ứng dụng cụ thể. Chính sự mềm dẻo, ưu việt trong sử dụng đó là yếu điểm trong bảo vệ an toàn thông tin dữ liệu. Cái khó lớn nhất đối với một phần mềm là có một sự hiểu biết đầy đủ về nó. Khi thiết bị được cấu thành từ nhiều bộ xử lý, mỗi bộ xử lý thực hiện một số giới hạn các chức năng và quan hệ kết nối với các bộ xử lý khác bằng các thể thức khá chính xác, thì việc kiểm tra phần mềm là một bài toán mà người ta hy vọng có thể thực hiện được. Ngược lại, người ta không thể nắm bắt được một cách hoàn toàn đầy đủ khi các máy tính lớn có các hệ thống xử lý phức tạp. Nếu có những yêu cầu về bảo mật nghiêm ngặt đối với các hệ thống đó thì phải bắt đầu từ nguyên lý mà tất cả hệ thống xử lý xem có những yếu kém sơ hở gì, một công việc không đơn giản.

Ngày nay, môi trường yếu điểm nhất của phần mềm trong bảo vệ an toàn thông tin dữ liệu, đặc biệt trên mạng máy tính, là nằm ở các phần mềm cung cấp cho các máy vi tính. Người sử dụng máy vi tính có thể sao chép các phần mềm từ nhiều nguồn khác nhau và tận dụng nó để xâm nhập các hệ thống được bảo vệ. Kẻ xâm nhập thường sử dụng một nhiệm vụ bình thường hữu ích nào đó, nhưng che dấu một nội dung xâm nhập mà từ chuyên môn còn gọi đó là "sự xâm nhập tấn công của con ngựa thành Troie".

Các xâm nhập phần mềm có thể từ bên ngoài hệ thống và cũng có thể từ bên trong hệ thống. Một lập trình viên nào đó làm việc trong hệ thống, nếu có ý đồ gian trá, có thể cài các bẫy trong phần mềm của mình hoặc sử dụng các phương tiện đặc biệt để thay đổi một chương trình sử dụng nào đó sang một dạng khác.

Điều đặc biệt nguy hiểm là nếu như một khi lập trình viên đó có sự cộng tác nào đó với đối phương, nhất là trong các hệ thống lớn có nhiều người sử dụng.

Việc giám định phần mềm để phát hiện những bất thường có thể thực hiện được nhưng không phải lúc nào hệ thống cũng thực hiện công việc giám định; có thể có những sơ hở.

Nếu như xem rằng các phần mềm của hệ thống là hoàn toàn tin cậy thì vấn đề đặt ra ở đây là việc bảo toàn các phần mềm đó. Các phần mềm có thể được bảo toàn bằng phương pháp vật lý, ví dụ chúng được lưu giữ trong các bộ nhớ ROM hoặc trong các bộ nhớ có bảo vệ chống ghi. Việc bảo toàn các phần mềm bằng phương pháp kỹ thuật không phải là khó, nhưng điều đó lại có mâu thuẫn với tính mềm dẻo của các phần mềm.

Không những chỉ có các phần mềm ứng dụng mà các đặc tính bảo mật của hệ thống có thể nằm trong các bảng chỉ dẫn các quy tắc truy cập vào các cơ sở dữ liệu mà nhiều người sử dụng hệ

thống có. Cũng vì vậy mà các bảng chỉ dẫn truy nhập bất hợp pháp là việc xác thực (sẽ giới thiệu chi tiết ở chương 7).

Còn một dạng xâm nhập nguy hiểm hơn nữa, đó là các "virut tin học". Có nhiều loại virut, chúng lây nhiễm gây bệnh các chương trình phần mềm, phá hoại dữ liệu trong các bộ nhớ, làm ngưng trệ hoạt động của máy tính hoặc của một mạng máy tính. Virút và phương pháp ngăn ngừa virút hiệu quả nhất và thông dụng, kinh tế nhất là phương pháp xác thực quyền truy nhập (sẽ giới thiệu ở các phần sau).

Chương 2

THUẬT TOÁN DES VÀ MẬT MÃ KHỐI

2.1. MÃ KHỐI VÀ THUẬT TOÁN LUCIFER

Mật mã khối được cấu trúc trên nguyên tắc là bản tin được chia thành các khối có độ dài bằng nhau và việc mã hoá được tiến hành theo từng khối độc lập nhau. Trong môi trường máy tính, độ dài của khối được tính bằng bit. Ví dụ bản tin được chia thành các khối B_i thì sau khi mã hoá chúng có các khối C_i tương ứng.

$$B_0B_1B_2\ldots B_{n-2}B_{n-1}$$
$$C_0C_1C_2\ldots C_{n-2}C_{n-1}$$

Độ bảo mật của mã trong trường hợp này phụ thuộc vào độ dài của khối và độ phức tạp của thuật toán mã. Nếu kích thước của khối quá bé thì việc giải mã không mấy khó khăn do dò tìm được đặc tính cấu trúc thống kê của bản tin rõ. Nếu tăng kích thước khối thì mức độ cấu trúc thống kê cũng tăng theo số mũ và nếu kích cỡ khối tiến đến đoạn tin thì tác dụng mã khối sẽ giảm.

Vào quãng đầu những năm 70, hãng IBM (Mỹ) đề xuất một thuật toán mã khối được gọi là thuật toán Lucifer. Thuật toán đó đáp ứng các yêu cầu của cơ quan chuẩn quốc gia Mỹ NBS (National Bureau of Standard) và được ứng dụng ở các trạm ngân hàng tự động. Thuật toán Lucifer sau này được phát triển cải tiến trở thành mã chuẩn DES (sẽ giới thiệu ở mục sau).

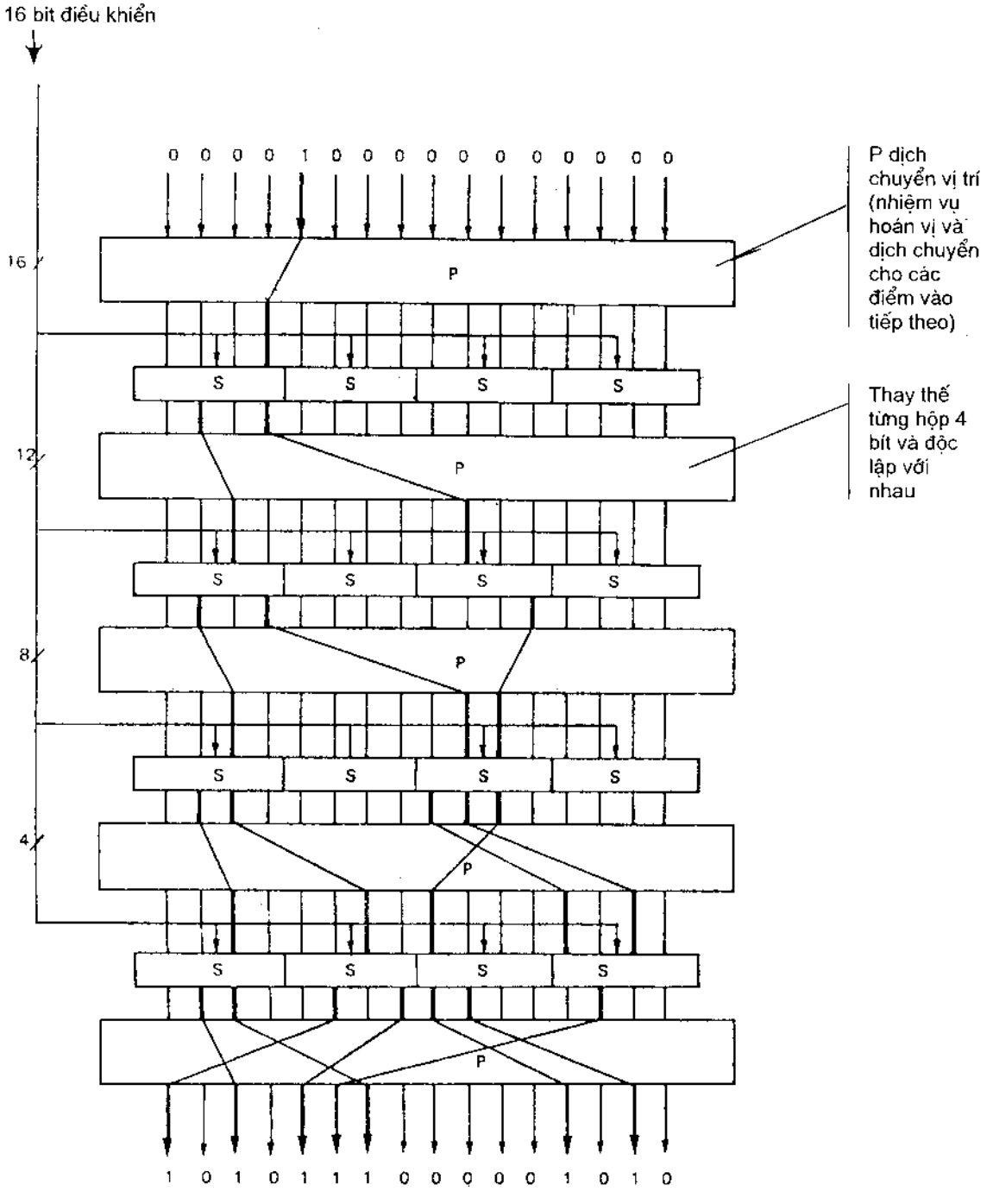
Trong chương một đã phân tích các dạng mã hiệu mật, các phép thay thế và chuyển vị; các dạng mã đó có độ mật rất kém. Ở thuật toán Lucifer có sự phối hợp các thuật toán trên để tạo nên mã có độ bảo mật lớn hơn nhiều. Thuật toán Lucifer có độ dài khối tin là 128 bit và độ dài khoá cũng là 128 bit. Hình 2.1. mô tả một phân đoạn của thuật toán Lucifer trong đó P là các hộp thay thế hoán vị (permutation) và S là các hộp chuyển vị (Substitution). Như vậy số khả năng các biến đổi với khoá 128 bit và độ dài khối 128 bit sẽ là $(2^{128})!$; một số khả năng khá lớn.

2.2. THUẬT TOÁN DES

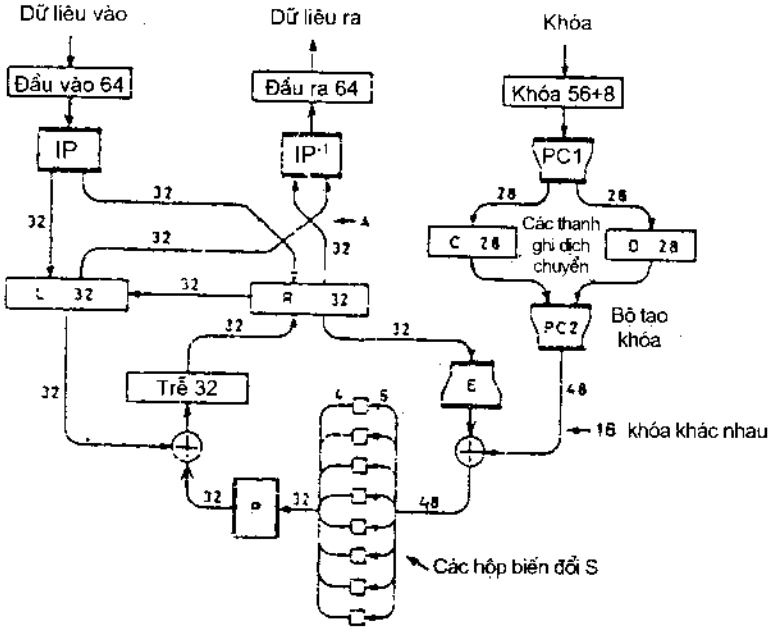
2.2.1. Cấu trúc thuật toán DES

Thuật toán DES do hãng IBM (Mỹ) đề xuất, lúc ban đầu chỉ dùng trong nội bộ nhưng sau đó được chấp nhận đưa ra dùng chung cho các ứng dụng khác cho nên có tên gọi là chuẩn mã bảo mật dữ liệu DES (data Encryption Standard). Thuật toán DES là một giải thuật mật mã đối xứng đang

được ứng dụng rộng rãi và còn có tên gọi là thuật toán mật mã dữ liệu, DEA (Data Encryption Algorithm). Hình 2.2 mô tả sơ đồ khối cấu trúc của thuật toán DES, trong đó bao gồm cả phần mã hoá và phần giải mã được tiến hành đồng thời.



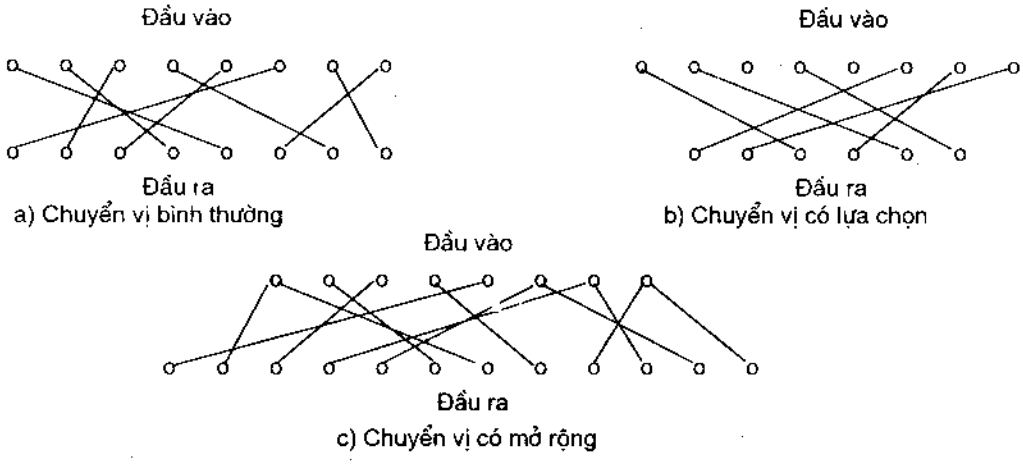
Hình 2.1. *Mã khối với độ dài đoạn tin 16 bit*



Hình 2.2. *Cấu trúc logic của thuật toán DES*

Thuật toán có hai đầu vào và một đầu ra. Hai đầu vào 64 bit đó là đầu vào bản tin rõ (hoặc bản tin đã được mã hoá) và đầu vào 64 bit của khoá mã. Một đầu ra 64 bit, đó là đầu ra bản tin đã được mã hoá (hoặc bản tin rõ). Thuật toán biến đổi bản tin rõ thành bản tin đã được mật mã hoá, hoặc biến đổi ngược lại, theo phương pháp bài toán lựa chọn. Ở 64 bit của khối khoá mã thì chỉ có 56 bit được sử dụng cho các phép biến đổi của thuật toán, còn 8 bit được sử dụng cho kiểm tra chẵn lẻ cho tám từ mã 8 bit của khoá mã.

Các phần tử cấu thành của thuật toán là các phép thay thế, chuyển vị và cộng modul-2. Phép chuyển vị của DES có ba dạng: chuyển vị bình thường, chuyển vị có mở rộng và chuyển vị có lựa chọn. Hình 2.3a, b và c mô tả ba phép chuyển vị đó.



Hình 2.3. *Mô tả ba dạng chuyển vị*

Các phép chuyển vị trong thuật toán DES được ký hiệu dưới dạng các hộp S và chúng được xác định bởi 8 bảng khác nhau. Ở thuật toán Lucifer thì kích cỡ đầu vào và đầu ra của các hộp S là 4 bit, nhưng ở thuật toán DES thì đầu vào các hộp S là 6 bit và đầu ra là 8 bit. Phép biến đổi dữ liệu đầu tiên là phép chuyển vị khởi đầu IP (initial permutation) theo phương pháp chuyển vị thông thường. Nhiệm vụ của nó là phân chia dữ liệu vào cho 8 thanh ghi dịch chuyển theo nguyên tắc là mỗi bit của byte dữ liệu vào được phân cho một vị trí của một thanh ghi tương ứng. Khi mà 64 bit của khối dữ liệu đã được đưa vào có nghĩa là các bit trên 8 thanh ghi đã được sắp xếp. Hình 2.4 mô tả cách sắp xếp các bit dữ liệu trong bộ chuyển vị khởi đầu (IP) của thuật toán DES.

IP

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 2.4. ***Sắp xếp các bit dữ liệu trong bộ chuyển vị khởi đầu (IP) của thuật toán DES***

Theo hình 2.4, cách sắp xếp được mô tả như sau: bit vào số 1 ở vị trí 40, bit vào số 2 ở vị trí 8... và bit ra 1 tương ứng với bit vào 58, bit ra 2 tương ứng với bit vào 50... Khi dữ liệu vào dùng mã hiệu ASCII có bit thứ 8 là bit kiểm tra chẵn lẻ thì chúng được tập trung vào một byte ở đầu khối và không tham gia vào các phép toán thay thế, chuyển vị. Khi kết thúc công việc mã hoá, trước khi tập trung dữ liệu vào thanh ghi đầu ra thì các dữ liệu được đưa vào bộ chuyển vị kết thúc và bộ chuyển vị kết thúc đó chính là nghịch đảo của bộ chuyển vị khởi đầu, được ký hiệu là IP^{-1}.

Sau khi công việc chuyển vị khởi đầu được thực hiện thì khối dữ liệu 64 bit được phân chia thành hai phân khối: phân khối trái (L) 32 bit và phân khối phải (R) 32 bit; chúng được đưa vào hai thanh ghi riêng biệt để bắt đầu thực hiện chu trình của thuật toán.

32 bit của thanh ghi R được đưa qua bộ chuyển vị có mở rộng E trong đó một nửa số bit được chuyển vị và một nửa số bit được lặp lại. Đầu ra của bộ chuyển vị E là 48 bit. Trong tất cả bốn byte của R thì các bit thứ nhất, bit thứ tư, bit thứ năm và bit thứ tám được lặp lại. Chuyển vị sau khi hoàn thành được mô tả ở hình 2.5.

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 2.5. ***Chuyển vị có mở rộng (E) của thuật toán DES***

Trong chuyển vị E đó, bit đầu tiên của đầu ra tương ứng với bit thứ 32 của đầu vào, bit thứ hai ở đầu ra tương ứng với bit đầu tiên của đầu vào... 32 bit của thanh ghi R lúc chưa chuyển vị cũng được đưa qua thanh ghi L để tham gia biến đổi bên phía L (sẽ xem xét ở phía L).

48 bit ở đầu ra của bộ chuyển vị mở rộng E được cộng modul-2 với các bit xuất phát từ khoá mã. 48 bit ở đầu ra của bộ cộng modul-2 đó lại được chia thành 8 nhóm mỗi nhóm 6 bit, và tất cả được đưa vào 8 hộp S. Mỗi một hộp S đó có 6 bit đầu vào và sẽ đưa ra 4 bit đầu ra. Hình 2.6 mô tả các biến đổi của 8 hộp S đó.

Sự biến đổi các bit theo mô tả ở bảng (hình 2.6) được giải thích thêm như sau: Đầu vào của mỗi bảng là 6 bit. Bit thứ nhất và bit cuối cùng của 6 bit đó tạo thành một từ 2 bit mà giá trị của nó được dùng để chọn dòng trong bảng các hộp S. Các bit từ thứ 2 đến thứ 5 tạo thành một từ 4 bit, mà giá trị của nó là từ 0 đến 15, được sử dụng để chọn phần tử trong dòng liên quan (bit về phía bên trái có trọng số lớn nhất). Mỗi một dòng của bảng các hộp S đặc trưng cho một số nhị phân xuất phát từ các bit 2 đến 5 ở đầu vào của các hộp S (cấu trúc các hộp S sẽ xem xét ở phần sau).

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	6	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	8	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	11	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	1	11	6
4	3	2	12	9	5	15	10	11	14	11	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 2.6. ***Bảng mô tả các biến đổi các hộp S của thuật toán DES***

Các đầu ra 4 bit của các hộp S được nhóm thành một trường 32 bit và sau đó đưa vào bộ chuyển vị P. Chi tiết biến đổi của bộ chuyển vị P đó được mô tả trong hình 2.7. Đầu ra của chuyển vị P là 32 bit, trong đó bit thứ nhất đầu ra tương ứng với bit 16 đầu vào, bit 2 đầu ra tương ứng với bit 7 đầu vào... Để kết thúc chu trình chính, 32 bit đầu ra của chuyển vị P được cộng modul-2 với 32 bit khởi đầu của thanh ghi L và kết quả được đặt vào thanh ghi R. Để thực hiện bài toán này mà không tạo ra sự sai lệch của các thanh ghi, một thanh ghi chờ 32 bit được đặt giữa bộ cộng modul-2 và thanh ghi R.

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 2.7. ***Chuyển vị P của thuật toán DES***

Chu trình chính giới thiệu trên sẽ được lặp lại 16 lần, sau đó nội dung của các thanh ghi R và L được tập hợp trong một khối 64 bit theo trật tự R sau đó L. Khối này sẽ được chuyển vị nghịch đảo (IP^{-1}) so với chuyển vị khởi đầu (IP) như đã giới thiệu trên.

Lưu ý là hãy đảo vị trí của hai khối L và R cho nhau trước khi thực hiện chuyển vị nghịch đảo IP^{-1}. Mỗi một lần thực hiện một chu trình chính đó được gọi là "một vòng".

Còn lại là vấn đề khoá mã sẽ tham gia như thế nào trong các quá trình đó. Như đã giới thiệu ở chu trình mã hoá trên, đầu ra của chuyển vị có mở rộng E là 48 bit dữ liệu và 48 bit dữ liệu đó được trộn (cộng modul-2) với 48 bit khoá mã. Với mỗi chu trình thì các bit đó có giá trị khác nhau. Phần bên phải của hình 2.2 mô tả phương pháp trộn dữ liệu với khoá mã đó.

Khoá mã được đưa vào thanh ghi khoá mã sau đó được đưa vào bộ chuyển vị PC1 (bộ chuyển vị lựa chọn 1). Cấu trúc của bộ chuyển vị PC1 cũng gần giống như bộ chuyển vị khởi đầu. Hình 2.8 mô tả biến đổi của bộ chuyển vị PC1.

C	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
D	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 2.8. ***Bộ chuyển vị PC1 có lựa chọn PC1 của thuật toán DES***

Đầu ra của bộ chuyển vị PC1 được đặt trong hai thanh ghi C và D. Bit thứ nhất của thanh ghi C tương ứng với bit 57 của khoá mã và bit thứ nhất của thanh ghi D tương ứng với bit 63 của khoá mã. Khoá mã 56 bit đó cũng được phân làm hai từ, mỗi từ 28 bit và được đặt trong các thanh ghi C

và D. Các thanh ghi C và D là các thanh ghi dịch chuyển vòng: cứ mỗi vòng chu kỳ mã nó chuyển dịch về phía trái một hoặc hai vị trí. Hình 2.9 mô tả bảng chuyển dịch đó.

Để có được 48 bit khoá mã đưa vào bộ cộng modul-2 với 48 bit dữ liệu, các nội dung của thanh ghi C và D được đưa tiếp vào bộ chuyển vị có lựa chọn PC2. Chuyển vị này có cấu trúc hơi đặc biệt và được mô tả trong hình 2.10. Ở đây bit thứ nhất của đầu ra tương ứng với bit 14 đầu vào (của các thanh ghi C và D), bit thứ 2 ở đầu ra tương ứng với bit 17 đầu vào... Đầu ra của bộ chuyển vị PC2 có kích cỡ là 48 bit. Cứ mỗi vòng của thuật toán thì đầu ra của chuyển vị PC2 lại cung cấp một khoá khác nhau để sử dụng cho chu trình.

Vòng	Khối khoá	Số chuyển dịch trái (trước PC2)
1	K1	1
2	K2	1
3	K3	2
4	K4	2
5	K5	2
6	K6	2
7	K7	2
8	K8	2
9	K9	1
10	K10	2
11	K11	2
12	K12	2
13	K13	2
14	K14	2
15	K15	2
16	K16	1

Hình 2.9. ***Bảng dịch chuyển trái của khoá mã (trước PC2)***

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	21	10
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 2.10. ***Chuyển vị có lựa chọn PC2 của thuật toán DES***

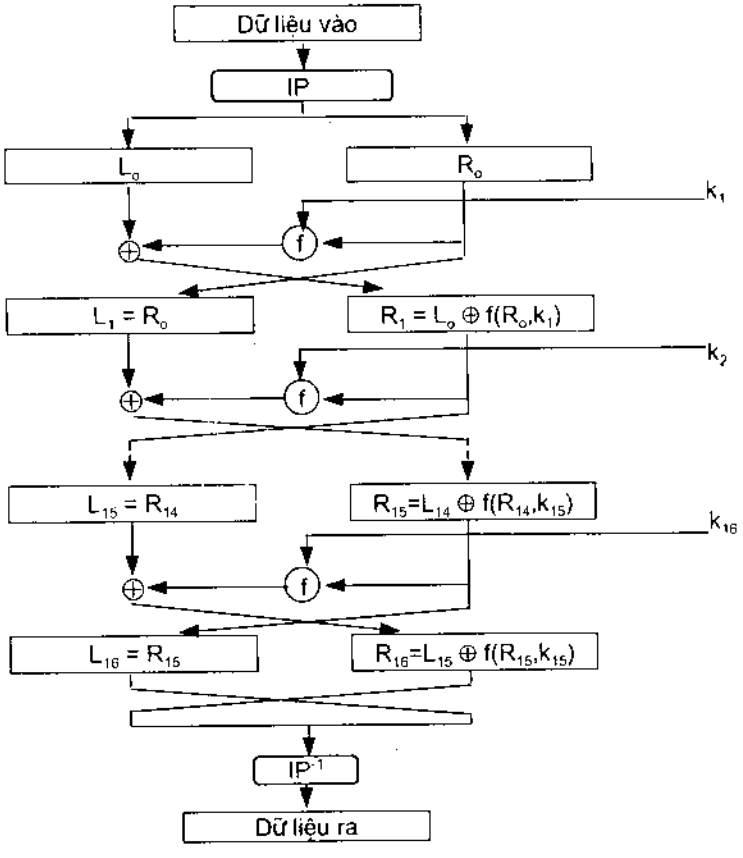
Đến đây, việc mã hoá theo thuật toán DES đã hoàn thành. Việc giải mã được thực hiện một cách dễ dàng bởi một quá trình tương tự như mã hoá, chỉ có một sự khác biệt duy nhất ở phần tạo ra các khoá mã riêng biệt. Ở giải mã, các thanh ghi dịch chuyển vòng về phía phải, trong lúc ở mã hoá chúng dịch chuyển vòng về phía trái. Bảng dịch chuyển vòng về phải của giải mã được mô tả trong hình 2.11. Ở đây cũng cần lưu ý rằng, trong khi mã hoá có một dịch chuyển được thực hiện trước khi khoá mã thứ nhất xuất hiện, điều đó không phải là trường hợp cho giải mã.

2.2.2. Lưu đồ thuật toán DES

Hoạt động của thuật toán DES có thể hiểu một cách dễ dàng nếu biểu thị thuật toán DES dưới dạng lưu đồ thuật toán như mô tả ở hình 2.12. Lưu đồ thuật toán biểu thị quan hệ giữa các chu trình kế tiếp nhau của thuật toán. Ở đây mô tả cách đổi vị trí lẫn nhau giữa các thanh ghi R và L cũng như việc trộn dữ liệu của các chu trình với các khoá mã riêng biệt từ k_1 đến k_{16}.

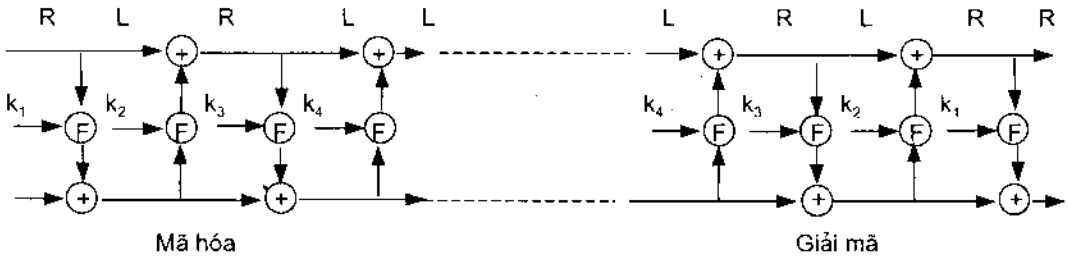
Vòng	Khóa	Số dịch chuyển phải (trước PC2)
1	k1	0
2	k2	1
3	k3	2
4	k4	2
5	k5	2
6	k6	2
7	k7	2
8	k8	2
9	k9	1
10	k10	2
11	k11	2
12	k12	2
13	k13	2
14	k14	2
15	k15	2
16	k16	1

Hình 2.11. *Bảng dịch chuyển phải của khoá mã khi giải mã*



Hình 2.12. *Mô tả lưu đồ thuật toán DES*

Trong lưu đồ thuật toán ở hình 2.12, các hoạt động của chuyển vị E, các hộp S và chuyển vị P được tóm tắt trong bộ chức năng F và để tiết kiệm không gian lưu đồ chỉ biểu thị ba trong 16 chu trình. Từ lưu đồ nhìn thấy rõ mối quan hệ giữa mã hoá và giải mã. Hình 2.13 mô tả quá trình giải mã kế tiếp sau quá trình mã hoá. Các ký hiệu R và L là đặc trưng cho nội dung các thanh ghi khác nhau cho mỗi chu trình. Cần lưu ý rằng, các khoá khác nhau để thực hiện việc giải mã là theo trật tự ngược lại với mã hoá. Tại mỗi điểm của lưu đồ có thể thiết lập trạng thái dữ liệu một cách chính xác cũng như trạng thái tương ứng của giải mã. Cấu trúc của bài toán mã hoá ở đây là hoàn toàn đối xứng với giải mã, do vậy thuật toán DES thuộc loại mã đối xứng.



Hình 2.13. ***Mô tả quan hệ giữa mã hoá và giải mã trong lưu đồ thuật toán DES***

2.2.3. Biểu thức đại số của thuật toán DES

Thuật toán DES có thể được biểu thị dưới dạng các biểu thức đại số. Cho rằng: j ký hiệu cho các chu trình khác nhau và P là đầu ra của chuyển vị P. Ký hiệu F có ý nghĩa như mô tả trong lưu đồ thuật toán. R và L là các giá trị nội dung trong các thanh ghi tương ứng; R_j và L_j là các giá trị tương ứng với chu trình thứ j; k_j là khoá mã tương ứng với chu trình j. Trong quá trình mã hoá có các quan hệ sau:

$$L_j = R_{j-1} \tag{2.1}$$

$$R_j = L_{j-1} \oplus P_j \tag{2.2}$$

$$P_j = F\,(R_{j-1}, k_j) \tag{2.3}$$

trong đó $\oplus$ là phép cộng modul-2.

Biểu thức (2.2) và (2.3) có thể tổng hợp thành:

$$R_j = L_{j-1} \oplus F(R_{j-1}, k_j) \tag{2.4}$$

Từ các biểu thức (2.1) và (2.4) cũng có thể viết:

$$R_{j-1} = L_j \tag{2.5}$$

$$L_{j-1} = R_j \oplus F(R_{j-1}, k_j) \tag{2.6}$$

Biểu thức (2.5) và (2.6) mô tả trạng thái thứ j theo trạng thái thứ j-1.

Thay thế (2.5) vào (2.6) có:

$$L_{j-1} = R_j \oplus F(L_j, k_j) \tag{2.7}$$

Biểu thức (2.5) và (2.7) mô tả trạng thái thứ j-1 theo trạng thái thứ j. Các biểu thức đó có thể xem như một phương tiện để có L_0 và R_0 xuất phát từ L_{16} và R_{16}, bằng cách sử dụng các khoá mã khác nhau theo trật tự k_{16}, k_{15}..., k_1. Chú ý rằng ở đây L và R được đảo cho nhau.

2.2.4. Đánh giá hiệu năng của mật mã DES

2.2.4.1. Hiệu năng thuật toán DES

Ý đồ của người phát minh ra mật mã DES là tạo ra một thuật toán biến đổi dữ liệu thật phức tạp để cho kẻ xâm nhập không thể tìm ra mối tương quan giữa đoạn tin đã mã hoá với đoạn tin rõ và cũng không thể thiết lập được mối quan hệ nào giữa đoạn tin đã được mã hoá và khoá mã.

Trong thuật toán, việc thay đổi một bit ở khối dữ liệu vào sẽ dẫn đến việc thay đổi một bit ở khối đầu ra với xác suất là 50%. Để đánh giá hiệu năng của nó sau đây sẽ xem xét một ví dụ cụ thể. Giả thiết rằng, sự thay đổi một bit ở đầu ra khó nhận biết và sự thay đổi một bit của khoá mã cũng sẽ dẫn đến đoạn tin đã mã hoá hoàn toàn khác.

Trong các ví dụ sau, đoạn tin rõ, đoạn tin đã mã hoá và các khoá mã đều sử dụng các ký hiệu thập lục phân. Phần bên trái của hình 2.14 mô tả các đoạn tin rõ, trong đó các dòng dưới chỉ khác nhau với dòng trên cùng là một bit. Tất cả các khối đều được mã hoá với cùng khoá mã là (0123456789ABCDF). Nếu nhìn lướt qua thì các đoạn tin đã được mã hoá chỉ thay đổi một bit, có vẻ không chắc chắn lắm. Để đánh giá các thay đổi đó, người ta đã tính khoảng cách Hamming giữa khối tin đã mã hoá khởi đầu và các khối khác đã mã hoá trong bảng. Khoảng cách Hamming trung bình tính được là 31,06; một giá trị rất gần với giá trị mong muốn, theo lý thuyết là 32.

Khoá 0 1 2 3 4 5 6 7 9 A B C D E F			
TT	Bản tin rõ	Bản tin mã hoá	Khoảng cách Hamming
1	A B C D E F A B C D E F A B C D	C D E 8 7 2 D 4 A 4 7 1 3 4 6 F	29
2	8 B C D E F A B C D E F A B C D	C D 3 D 0 A A 4 C 0 2 4 B 4 A	38
3	A 9 C D E F A B C D E F A B C D	8 0 1 F 8 A 2 9 6 8 B C 4 4 7 3	36
4	A B D D E F A B C D E F A B C D	5 D 9 8 C 4 7 D D B A 6 F 3 0	36
5	A B C F E F A B C D E F A B C D	9 9 8 9 5 6 2 A 7 4 F 4 0 1 C 9	26
6	A B C D 6 F A B C D E F A B C D	6 7 C 2 6 9 F 2 5 4 2 7 9 1 F 9	30
7	A B C D E B A B C D E F A B C D	F 8 C 9 8 F 7 9 A D C 0 6 E A 4	33
8	A B C D F F D B C D E F A B C D	8 7 D 3 2 4 0 A B B F 4 4 0 7 4	34
9	A B C D E F A 9 C D E F A B C D	D B 9 9 8 B 6 7 0 4 6 C D C E 7	30
10	A B C D E F A B E D E F A B C D	2 F 6 E 5 4 7 0 E 4 E 3 5 1 A C	25
11	A B C D E F A B C C E F A B C D	B 5 3 E 4 2 D E 3 0 F 9 7 A D 0	29
12	A B C D E F A B C D 6 F A B C D	4 F 4 0 6 7 7 2 6 B 3 5 B 0 1 4	28
13	A B C D E F A B C D E 7 A B C D	A B 1 5 5 2 8 9 6 6 0 C 6 0 B 2	35
14	A B C D E F A B C D E F A F C D	5 B D A 9 3 F 7 D 4 2 7 B 8 D 2	30
15	A B C D E F A B C D E F A B C D	9 8 5 3 C 5 1 1 E D 5 6 8 8 7 E	34
16	A B C D E F A B C D E F A B D D	7 0 A A 2 4 0 7 9 5 9 F 0 4 B 1	34
17	A B C D E F A B C D E F A B C 5	8 9 2 B E C 4 7 C 9 7 1 2 B E 3	26
		Khoảng cách Hamming trung bình	31.06

Hình 2.14. *Mật mã DES một đoạn các khối dữ liệu có giá trị khác nhau một bit so với khối khởi đầu*

Hình 2.15 mô tả mã hoá một khối đoạn tin rõ (ABCEFABCEFABCD) với một chuỗi liên tiếp các khoá mã khác nhau. Ở đây, trong dòng thứ nhất được thực hiện với một khoá mã nào đó và các khoá để mã hoá các dòng tiếp theo chỉ khác khoá mã dòng trước là một bit (nếu tính cả bit kiểm tra thì sự khác nhau sẽ là hai bit). Như vậy, các khoảng cách Hamming giữa khối đoạn tin được mã hoá đầu tiên và các khối đoạn tin kế tiếp sau đó có giá trị trung bình tính được là 32,88; giá trị đó cũng gần với giá trị mong muốn.

Bản tin rõ: ABCDEFABCDEFABCD			
TT	Khoá	Bản tin mã hoá	Khoảng cách Hamming
1	0123456789ABCDEF	CDE872D4A471346F	
2	0123456789ABCDEF	1B73FE8BC0B88606	35
3	0123456789ABCDEF	0F92F60D2FD4D8B7	32
4	0162456789ABCDEF	31AFD8C54FBF4BCD	37
5	0126456789ABCDEF	C79F5963D465A7EE	29
6	0123046789ABCDEF	36529CC10717A389	39
7	0123455789ABCDEF	7F35F7E6CEC57EE3	30
8	0123454689ABCDEF	E6928322EE7B9A69	36
10	01234567A8ABCDEF	0997A5AF6E4E1460	37
11	012345678AABCDEF	C7B41C4F38D9AF7A	31
12	0123456789EACDEF	4B4AA20AB2144DDD	30
13	0123456789ABCDEF	12997EE7001BD27C	32
14	0123456789AB4CEF	42D17B7DF5343B79	28
15	0123456789ABCEEF	87F6493C4C898327	33
16	0123456789ABCD6E	FC30F7F76BD42592	31
17	0123456789ABCDEC	1CBDEC4B79BCA7A1	39
		Khoảng cách Hamming trung bình	32,88

Hình 2.15. ***Mã hoá DES một khối dữ liệu với các khoá mã khác nhau trong đó các khoá sau khác khoá khởi đầu một bit.***

Để che dấu sự tương đồng đó, sau đây sẽ xem xét tiếp một ví dụ được thực hiện trong mã DES, trong đó có các biến đổi trong quá trình thực hiện các chu trình thuật toán. Hình 2.16 mô tả kết quả mã hoá của hai đoạn tin rõ, chúng khác nhau chỉ 1 bit, dùng cùng khoá mã (08192A3B4C5D6E7F). Các giá trị của các thanh ghi L và R khởi đầu của 16 chu trình được biểu thị trên bảng (các giá trị đó được gọi là các kết quả trung gian). Kết quả cuối cùng của phép mã hoá cũng được biểu thị và khoảng cách Hamming giữa các kết quả trung gian được tính toán (thường được biểu thị dưới dạng đồ thị). Từ các số liệu ở hình 2.16 nhận thấy rằng, ở hai kết quả đầu tiên thì khoảng cách Hamming giữ nguyên giá trị rất nhỏ, nhưng ở các chu trình tiếp theo thì chúng tăng luỹ tiến rất lớn. Sau chu trình thứ 5 thì có rất ít sự tương quan giữa các đầu vào và các kết quả trung gian. Có nhiều cách để giải thích tính chất đó nhưng tất cả đều dẫn đến kết luận là với 5 chu trình cũng đã khá đủ để khắc phục tính điều hoà của hàm số và thuật toán DES chọn 16 chu trình thực hiện cũng xem như là quá đủ để khắc phục tính điều hòa của hàm số.

Mã hoá với khoá mã: 08192A3B4C5D6E7F

Đoạn tin rõ 1: 0000000000000000			Đoạn tin rõ 2: 0000000000000001		
Chu trình	Thanh ghi L	Thanh ghi R	Thanh ghi L	Thanh ghi R	Khoảng cách Hamming
1	00000000	00000000	00000000	00000000	1
2	00000000	AFOD68FD	00000000	AFOD687D	1
3	AF0D68FD	CE0A36EA	AF0D687D	CE3A32E2	5
4	CE0A36EA	0BDCC5FE	CE3A32E2	81BED5F	15
5	0BDCC5FE	FD181CC3	81BED5F	F8CA39B2	26
6	5D181CC3	1744B978	F8CA39B2	A994B918	26
7	1744B978	9B1CB0D8	A994B918	3E9D05D8	22
8	9B1CB0D8	7AE8C7E0	3E9D05D8	23F48DFF	27
9	7AE8C7E0	A2AC7B3F	23F48DFF	9D58DCFB	34
10	A2AC7B3F	580E51EF	9D58DCFB	3F076303	34
11	580E51EF	2865DBD4	3F076303	97AE4AE3	35
12	2865DBD4	BF68F77C	97AE4AE3	68ABB612	37
13	BF68F77C	8F57F629	68ABB612	09D9C398	32
14	8F57F629	0287B240	09D9C398	44B0435C	31
15	0287B240	F2DEBFAC	44B0435C	319FD4B8	29
16	F2DEBFAC	16CD0EB8	319FD4B8	42684FF9	24

Đoạn tin 1 đã mã hoá: 125DDAC3E96176467

Đoạn tin 2 đã mã hoá: 1BDD183F1626FB4322

Hình 2.16. *Mã hoá DES một khối dữ liệu với hai khoá mã có giá trị khác nhau 1 bit (mô tả các kết quả trung gian trước mỗi chu trình thuật toán).*

2.2.4.2. Đặc tính bù của thuật toán DES

Một trong những tính chất cần lưu ý khi sử dụng mã DES là đặc tính bù. Có nghĩa là, nếu dùng giá trị bù (complement) của đoạn tin rõ cũng như của khoá mã thì lúc đó nội dung của đoạn tin được mã hoá sẽ là giá trị bù của đoạn tin được mã hoá nguyên thuỷ.

Điều đó có nghĩa là, nếu $y = Ek(x)$ thì $\overline{y} = E\overline{k}(\overline{x})$.

Sự xuất hiện kết quả đó là do trong thuật toán có hai bộ cộng modul 2; Một được đặt trước các hộp D trong cấu trúc của thuật toán và một đặt sau bộ hoán vị P. Tuy vậy, với các chu trình dùng khoá biến đổi của thuật toán, tính bảo mật của thuật toán vẫn được đảm bảo. Nó cũng không cho phép giảm thời gian dò tìm khoá mã nếu cho biết một cặp khoá.

Trong trường hợp cụ thể, đặc tính bù có thể biểu thị như sau. Giả thiết đoạn tin rõ x với các giá trị:

$$y_1 = Ek(x)$$

và $\overline{y_2} = Ek(\overline{x})$

thì lúc đó: $y_2 = E\overline{k}(x)$

Nếu như với mỗi giá trị của x, kiểm tra thấy nếu $Ek(x)$ hoặc nhận giá trị y_1, hoặc nhận giá trị y_2 thì lúc đó chúng đã như hai giá trị khoá k và khoá bù của k. Đó là một vấn đề đáng lo ngại. Để tránh điều đó thì hệ thống không thể cho phép đối phương nhận được cùng một lúc $Ek(x)$ và $Ek(\overline{x})$. Đó là một nhược điểm mà đối phương có thể khai thác.

2.2.4.3. Các khoá yếu của DES

Một đặc tính quy luật nữa của mã DES là vấn đề chọn các bit để hình thành các khoá mã từ k_1 đến k_{16}. Rõ ràng là, nếu như tất cả các bit của khoá mã có giá trị 1 hoặc giá trị 0 thì lúc đó đầu ra của bộ chuyển vị PC2 của mỗi chu trình sẽ là giống nhau đối với tất cả các chu trình. Như vậy, sự thay đổi của các khoá theo trật tự từ k_1 đến k_{16} cũng giống như theo trật tự từ k_{16} đến k_1. Điều đó có nghĩa là việc mã hoá là hoàn toàn tương đương. Hoặc nói cách khác, nếu như tất cả các bit của khoá mã có giá trị 1 hoặc giá trị 0 thì một đoạn tin rõ sẽ được biến đổi theo thuật toán mã hóa cũng giống như thuật toán giải mã. Kết quả là nếu như khối đoạn tin được mã hoá bởi một trong hai khoá đó, sau đó giải mã với cùng khoá đó, sẽ nhận được khối tin ban đầu. Mặt khác, ở đây các thanh ghi C và D là cách biệt nhau; nếu như các bit 0 trong C và các bit 1 trong D (hoặc ngược lại) tạo ra một tập các khoá không đổi. Có bốn khoá thuộc dạng đó và được mô tả ở hình 2.17. Cần tránh các khoá yếu đạng đó khi chọn các khoá có tính chất quyết định như các khoá chủ.

01	01	01	01	01	01	01	01
FE	FE	FE	FE	FE	FE	FE	FE
1F	1F	1F	1F	0E	0E	0E	0E
E0	E0	E0	E0	F1	F1	F1	F1

Hình 2.17. *Các khoá yếu của mã DES*

2.2.4.4. Các khoá hơi yếu của DES

01	FE	01	FE	01	FE	01	FE
FE	01	FE	01	FE	01	FE	01
1F	E0	1F	E0	1F	E0	1F	E0
E0	1F	E0	1F	E0	1F	E0	1F
01	E0	01	E0	01	F1	01	F1
E0	01	E0	01	F1	01	F1	01
1F	FE	1F	FE	0E	FE	0E	FE
FE	1F	FE	1F	FE	0E	FE	0E
01	1F	01	1F	01	0E	01	0E
1F	01	1F	01	0E	01	0E	01
E0	FE	E0	FE	F1	FE	F1	FE
FE	E0	FE	E0	FE	F1	FE	F1

Hình 2.18. *Các khoá hơi yếu của mã DES*

Ngoài các khoá yếu của DES như đã nêu trên, còn có một tập hợp các khoá khác mà chúng cũng có đặc tính yếu kém ít nhiều tương tự như các khoá yếu trên. Đó là "các khoá hơi yếu", nó xuất hiện thành từng cặp. Các cặp khoá trong tập hợp đó đều giống nhau nhưng theo trật tự đảo ngược nhau. Nói cách khác là khoá k_1 phát sinh từ khoá thứ nhất của cặp thì khoá k_{16} cũng phát sinh từ khoá thứ hai của cặp. Việc mã hoá với một khoá của cặp khoá, sau đó lại mã hoá với khoá thứ hai của cặp khoá thì sẽ cho kết quả ban đầu. Bộ 6 cặp khoá hơi yếu đó được liệt kê ở hình 2.18.

Trong các khoá hơi yếu thì một trong các thanh ghi C và D chứa dạng 1 và 0 lặp lại: 0101... Tính chất đó là một dạng đối xứng dịch chuyển cần tránh trong quá trình chọn khoá mã. Một số tài liệu dùng thuật ngữ các khoá kép (dual keys) để chỉ các dạng khoá hơi yếu mang tính cặp này.

2.2.4.5. Các chu kỳ Hamilton trong DES

Như đã biết, tính bền vững của thuật toán DES hoàn chỉnh phụ thuộc vào sự phức tạp của các hàm của các chu trình được lặp lại (R)ESP(R) trong đó (R) là nội dung của thanh ghi R, E là chuyển vị có mở rộng, S là các hộp biến đổi S và P là chuyển vị. Giả thiết rằng, bài toán biến đổi trong chu trình thực hiện được đặc trưng bởi các chuyển vị E và P. Việc sắp xếp một cách ngẫu nhiên các bit trong L và R không quan trọng. Việc đánh số các đầu vào và các đầu ra của một hộp S cụ thể cũng không quan trọng. Tất cả đầu ra đều có một trạng thái tương tự. Các đầu vào của các hộp S, do tính chất biến đổi của E, có thể chia làm 3 loại. Giả thiết ABCDEF là 6 bit đầu vào của hộp S. Các giá trị EF của một hộp S là bằng các giá trị AB của hộp S cạnh bên phải. Ví dụ, trong trường hợp các hộp 1 và 2, E1 = A2 và F1 = B2. Chúng ta sẽ không phân biệt được các bit vào A và B, bởi vì một sự chuyển vị các bit của các thanh ghi L và R có thể làm thay đổi chúng, một sự thay đổi hoàn toàn bằng cặp EF tương ứng.

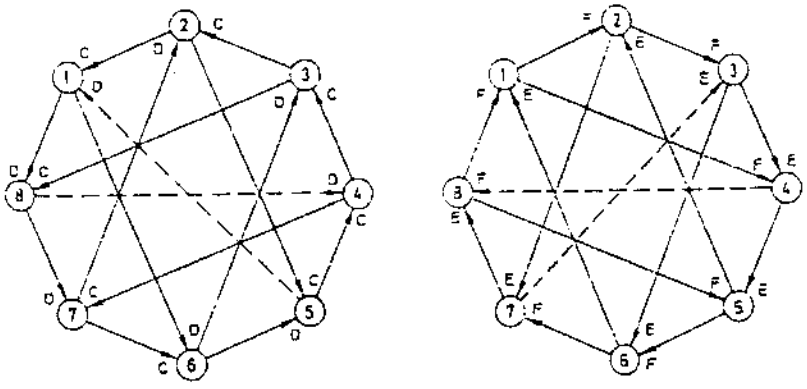
Điều quan trọng trong mắt xích các chuyển vị P và E không phải ở chỗ chúng tác động lên các bit mà lên các nhóm AB, CD và EF liên quan. Hình 2.19 mô tả quá trình đó, trong đó với mỗi đầu ra của hộp S có những đầu vào nào được gán sau chuyển vị PE. Ví dụ, bit ra 1 của hộp S1 nhận các bit F2 và B3 của các hộp S2 và S3 tương ứng. Điều đó nói lên rằng, 4 bit đầu ra của hộp S có hai đi qua chu trình đến các đầu vào C và D, còn hai đến các đầu vào A, B, E, F.

S1	S2	S3	S4
f2 f4 d6 d8 b3 b5	f3 e7 e1 c5 b4 a8	e6 e4 c8 c2 a7 a5	c7 e5 c3 f8 a6 b1

S5	S6	S7	S8
c2 c4 f6 d1 a3 b7	e1 f7 d3 d5 a2 b8	e8 e3 c6 d2 a1 a3	f1 d7 d4 f5 b2 b6

Hình 2.19. ***Mô tả sự kết nối lẫn nhau giữa các đầu vào và đầu ra của các hộp S***

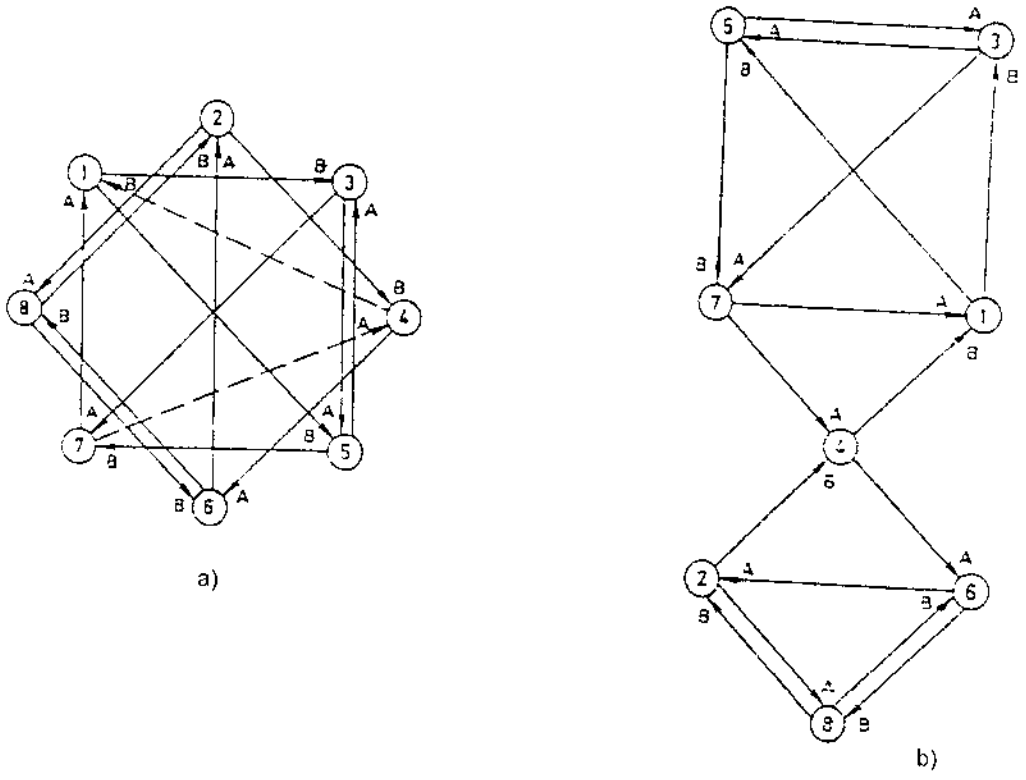
Đặc tính nói trên có thể được biểu thị bởi một đồ thị có các cung đặc trưng cho các đầu ra của các hộp S hướng đến các đầu vào. Hình 2.20 mô tả đồ thị cách biệt cho các đầu vào CD và EF. Đỉnh của hình bát giác đặc trưng cho 8 hộp S.



Hình 2.20. *Đồ thị mô tả hướng các cung ứng với các đầu vào CD và EF của các hộp S*

Trên đồ thị ở hình 2.20 với các đầu vào CD có thể tự tách thành 2 chu kỳ Hamilton (87654321) và (84725163). Đồ thị của EF là đối xứng qua một trục đứng. Cả hai đều là những đồ thị điều hoà và dường như đã được xác định.

Nếu vẽ đồ thị tương ứng với các đầu vào AB theo phương pháp tương tự như CD và EF, thì lúc đó sẽ có đồ thị như hình 2.21a mà ở đây sẽ có ít tính quy luật hơn so với đồ thị hình 2.20. Đồ thị 2.21a có thể vẽ lại theo dạng 2.21b để đỡ phức tạp hơn.



Hình 2.21. *a) Đồ thị mô tả hướng các cung ứng với các đầu vào AB của hộp S*
b) Đồ thị 2.21a được vẽ lại

Các bộ chuyển vị của mã DES đều là cố định và chúng không phụ thuộc vào khoá mã. Ở đây có một vấn đề cần lưu ý là các sự lựa chọn phân tích trên chưa tính đến việc đánh số các đầu vào và các đầu ra của các hộp S.

2.3. CÁC PHƯƠNG PHÁP ỨNG DỤNG MẬT MÃ KHỐI

2.3.1. Giới thiệu

Mật mã khối xử lý các khối dữ liệu có độ dài cố định (ví dụ độ dài 64 bit) và độ dài đoạn tin có thể bất kỳ. Có bốn phương pháp ứng dụng mã khối thường gặp trong bảo mật các hệ thống truyền tin số và truyền dữ liệu, đó là:

1. Phương pháp dùng từ điển điện tử, còn gọi là mật mã ECB (Electronic CodeBook).

2. Phương pháp móc xích các khối đã được mã hoá, còn gọi là mật mã CBC (Cipher Block Chaining).

3. Phương pháp phản hồi bản tin đã mã hoá, còn gọi là mật mã CFB (Cipher FeedBack)

4. Phương pháp phản hồi đầu ra, còn gọi là mật mã OFB (Output FeedBack).

Các phương pháp ứng dụng trên được phát triển mạnh sau khi xuất hiện mã DES. Trong thực tế có thể có các phương pháp khác, nhưng bốn phương pháp trên được ứng dụng phổ biến và cũng khá đầy đủ. Chúng có thể được sử dụng với bất kỳ dạng mã khối nào và cũng thường được sử dụng kết nối với mật mã DES.

2.3.2. Mật mã ECB

Khối	Bản tin rõ	Bản tin đã mã hóa
1	T H E Y C A N	60 99 46 42 52 B2 22 49
2	H A V E S E	FF BF BC 77 8B BB F2 06
3	V E R A L A C	0D 4D B6 DE B6 CD 92 5D
4	T I V E P E R	99 63 A8 OF 32 D3 E7 E9
5	M A N E N T V	10 49 1F 3B DE 67 21 B7
6	I R T U A L C	BD 2D 6D 61 42 06 C7 B8
7	I R C U I T S	19 F1 01 A4 69 5A AE 4C
8	A N D / O R V	84 DB CC EC 35 18 59 9C
9	I R T U A L C	BD 2D 6D 61 42 08 C7 B8
10	A L L S A T	D4 3C D4 5A 9E 0B A5 ED
11	T H E S A M E	84 52 01 AC 2D FE 98 3A
12	T I M E .	89 F1 89 E9 DB CC CB BB
	Khóa	01 23 45 67 89 AB CD EF

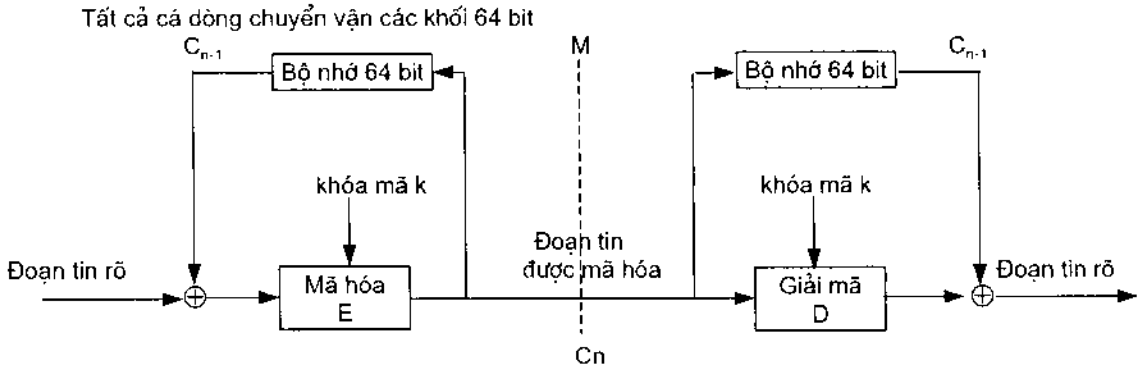
Hình 2.22. *Ví dụ sự yếu kém của mã ECB đối với một đoạn tin tiếng Anh*

Mật mã ECB sử dụng trực tiếp thuật toán để mã hoá từng khối tin một. Mật mã ECB sử dụng từ điển điện tử có một số giới hạn nhất định bởi vì trong các ứng dụng thực tế có những mẫu đoạn

tin có xu hướng lặp lại. Ngay các đoạn tin từ các máy tính cũng có tính chất lặp lại, nó có thể chứa những dãy 0 hoặc khoảng trống. Các định nghĩa về thể thức cho các tham số là các giá trị hằng số, đoạn tin có khuôn dạng cố định với các dữ liệu quan trọng luôn cùng vị trí. Đối phương có thể sử dụng các dạng dữ liệu tương tự để phát hiện các tính quy luật. Hình 2.22 mô tả một đoạn tin (tiếng Anh) được mã hoá theo mã ECB. Ở đây tính quy luật của mã hoá rất dễ bị phát hiện. Mặt khác sự yếu kém của mã còn ở chỗ các khối tin được mã hoá riêng rẽ, chúng không có quan hệ với nhau.

2.3.3. Mật mã CBC

Phương pháp mật mã CBC sử dụng đầu ra của phép toán mã hoá để biến đổi đầu vào tiếp theo. Như vậy mỗi một khối được mã hoá không những chỉ phụ thuộc vào đoạn tin rõ tương ứng mà còn phụ thuộc vào tất cả các khối của đoạn tin rõ trước nó. Hình 2.23 mô tả sơ đồ khối chứng năng của mật mã CBC (mã hoá và giải mã).



Hình 2.23. *Mô tả sơ đồ khối chức năng phương pháp mật mã CBC*

Ở đây, tất cả các phép toán được thực hiện với 64 bit song song. Trong các hệ truyền dẫn thực, việc truyền dữ liệu có thể là nối tiếp nhưng nguyên lý bài toán tốt nhất là thực hiện song song. Phương pháp mã hoá được mô tả ở phần bên trái của hình 2.23 đặc trưng cho sự móc xích khối dữ liệu đã được mã hoá ở đầu ra với khối dữ liệu rõ ở đầu vào. Trừ khối đầu tiên, mỗi một khối trước khi mã hoá sẽ được cộng modul-2 ($\oplus$) với khối đã được mã hoá trước đó, tức khối thứ n được mã hoá thành C_n phụ thuộc vào tất cả các khối dữ liệu rõ $P_1,...P_n$. Phần bên phải của hình 2.23 là quá trình giải mã được thực hiện theo phương pháp ngược lại. Ở đây, sau khi giải mã sẽ thực hiện phép cộng modul-2 với khối đã được mã hoá trước để có dữ liệu rõ ban đầu. Trên sơ đồ khối, phép cộng modul-2 chỉ biểu thị có tính chất nguyên lý. Trong quá trình thực hiện, mỗi một bit của khối dữ liệu vào, P_n được cộng modul-2 với một bit tương ứng của khối dữ liệu đã được mã hoá trước đó, C_{n-1}. Trong quá trình thực hiện, phép toán có thể thực hiện theo phương pháp nối tiếp hoặc song song từng byte một. Nếu thực hiện với mật mã DES thì tốc độ của chúng phù hợp với tốc độ của mã DES.

Có thể giải thích phương pháp mật mã CBC như sau:

- Với phép mã hoá:

$$C_n = Ek(P_n \oplus C_{n-1})$$

- Với phép giải mã:

$$Q_r = Dk(C_n) \oplus C_{n-1}$$

Để chứng minh rằng, sẽ xác định được đoạn tin rõ sau khi giải mã, ở đây sẽ dùng phép toán *Dk* cho biểu thức đầu tiên, sẽ có:

$$Dk(C_n) = P_n \oplus C_{n-1}$$

Thay thế giá trị đó vào biểu thức thứ hai, sẽ có:

$$Q_n = P_n \oplus C_{n-1} \oplus C_{n-1} = P_n.$$

Tính chất đối xứng đó được mô tả ở hình 2.23, có một sự đối xứng của phần bên phải so với phần bên trái qua điểm M cho các dòng dữ liệu. Điều đó cũng dễ nhận thấy qua tính chất của phép toán cộng modul-2 là đối xứng.

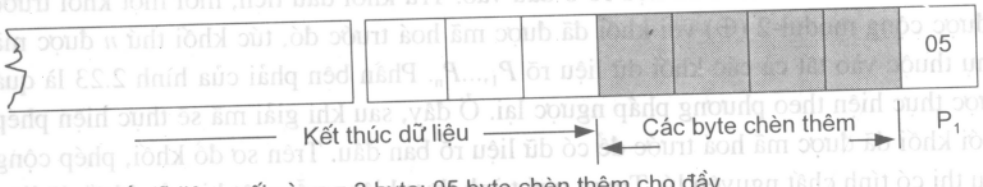
Như giới thiệu trên là nguyên lý. Để thực hiện, ở đây còn một vấn đề phải giải quyết là khối dữ liệu đầu tiên và khối dữ liệu cuối cùng cần một thủ tục bổ sung. Ở điểm xuất phát thì khối dữ liệu được mã hoá trước nó không có, cho nên nội dung khởi đầu của thanh ghi trong hình 2.23 được thực hiện riêng. Đại lượng đó được gọi là *giá trị khởi đầu* hoặc *véctơ khởi đầu* (trong các biểu thức toán học ký hiệu là *I* và trong lưu đồ thuật toán ký hiệu là IV). Việc chọn giá trị khởi đầu đó rất quan trọng để bảo đảm bí mật ở hai đầu nút (đầu và cuối đoạn tin). Với $n = 1$, các phép toán mã hoá và giải mã lúc đó cần phải được thay thế bởi:

$$C_1 = Ek(P_1 \oplus I) \quad \text{và} \quad Q_1 = Dk(C_1) \oplus I$$

và các biểu thức trên cũng có giá trị với $n = 2, 3....$

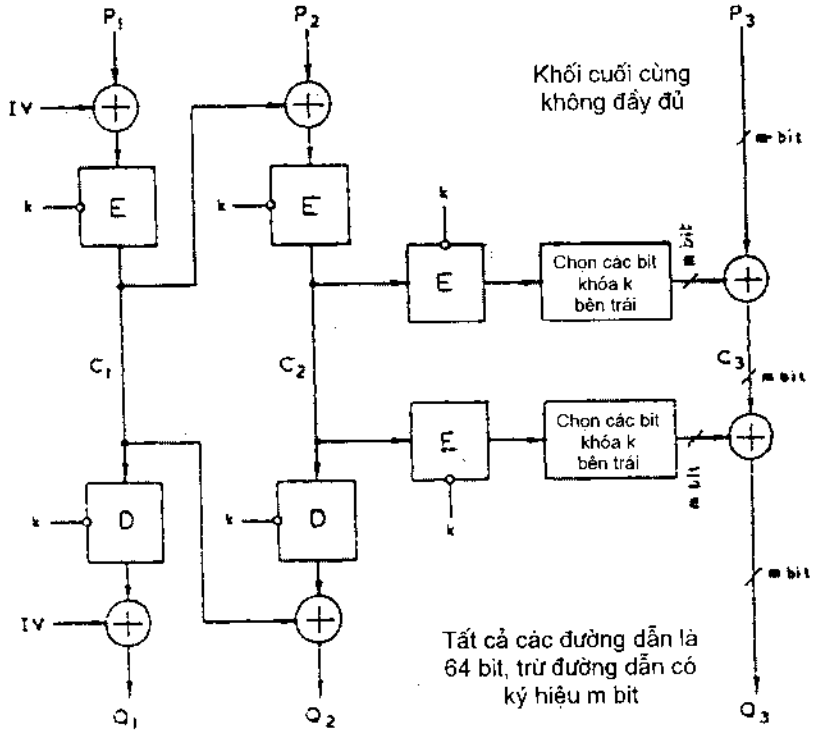
Trong trường hợp dùng mật mã CBC cho hệ thống truyền tin thì giá trị IV khởi đầu cần phải giữ hoàn toàn bí mật.

Một bản tin thường có độ dài tuỳ ý và nhiều trường hợp không thể chia chẵn cho các khối dữ liệu 64 bit, khối dữ liệu cuối cùng có thể ngắn hơn các khối khác. Có thể có hai cách giải quyết; một là chèn thêm các bit vô nghĩa để khối có độ dài bằng các khối khác và hai là khối tin được mã hoá theo một phương pháp riêng. Hình 2.24 mô tả phương pháp chèn thêm các byte vô nghĩa.



Hình 2.24. ***Mô tả chèn thêm dữ liệu ở khối cuối cùng của đoạn tin***

Trong nhiều trường hợp ứng dụng, người thiết kế hệ thống có thể xử lý khối dữ liệu cuối cùng bằng nhiều phương pháp khác nhau, nhưng tốt nhất là không làm gia tăng độ dài của bản tin. Hình 2.25 mô tả một phương pháp xử lý khối tin cuối theo nguyên lý giống như phương pháp mô tả ở hình 2.23 nhưng có khác là mỗi khối dữ liệu sẽ được xử lý riêng. Ở đây cần một đoạn tin được tạo thành bởi 2 khối hoàn chỉnh trước khối m bit. Khối cuối cùng được mã hoá CBC thêm một lần và được sử dụng để cộng modul-2 với đoạn tin ngắn cuối. Giá trị đó là hoàn toàn ngẫu nhiên để sử dụng cho xử lý đoạn dữ liệu cuối.



Hình 2.25. *Mô tả một phương pháp mật mã khối dữ liệu cuối cùng không dùng phương pháp chèn thêm bit*

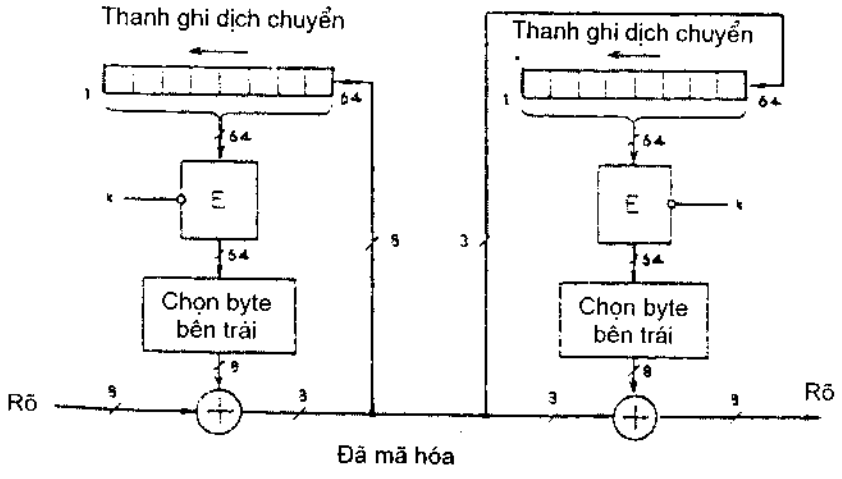
Phương pháp mật mã CBC có thể được đặc trưng như một sự phản hồi bản tin đã mã hoá về phía phát và một sự phản hồi về phía thu. Quá trình phản hồi đó dẫn đến một điều cần lưu ý là nếu có một bit lỗi trong bản tin sẽ dẫn đến tất cả các khối dữ liệu có quan hệ với khối đó đều bị lỗi.

Mật mã CBC được xây dựng trên cơ sở các khối dữ liệu có quan hệ móc xích với nhau, nhằm khắc phục nhược điểm của phương pháp dùng từ điển. Điều đó chỉ đúng bắt đầu từ khối dữ liệu thứ hai, còn khối dữ liệu đầu tiên lại phụ thuộc vào giá trị khởi đầu (IV). Nếu như giá trị khởi đầu luôn là hằng số với tất cả các bản tin thì điều đó tạo tính quy luật đối phương dễ phát hiện. Trong thực tế truyền tin các giá trị khởi đầu luôn được thay đổi và người ta cũng tránh việc dùng cùng giá trị khởi đầu và cùng khoá mã nhiều lần cho các bản tin được truyền.

2.3.4. Mật mã CFB

Dữ liệu được xử lý và được truyền dưới nhiều dạng khác nhau, chúng có thể dưới dạng các bản tin hoàn chỉnh, các khối dữ liệu, các gói, các ký tự riêng rẽ, theo byte hoặc theo bit. Khi phải xử lý các đoạn tin theo byte hoặc theo bit, người ta thường sử dụng một phương pháp mật mã dưới dạng phản hồi đoạn tin đã mã hoá, được gọi là mật mã CFB (Cipher FeedBack). Yêu cầu ở đây là các byte dữ liệu khi xuất hiện ra đường truyền cần được mã hoá ngay và việc mã hoá đó không ảnh hưởng đến sự chậm tốc độ truyền. Ở phía giải mã cũng có yêu cầu giải mã ngay khi một byte dữ liệu đến.

Hình 2.26 mô tả sơ đồ khối nguyên lý hoạt động của mật mã CFB. Nhìn lướt qua, phương pháp giống như phương pháp mật mã CBC, tuy vậy ở đây có sự khác biệt nằm trong phép toán mã hoá khối có trong pha phản hồi về sau ở thiết bị đầu cuối phát và trong pha phản hồi về trước ở thiết bị đầu cuối thu.



Hình 2.26. *Mô tả sơ đồ khối nguyên lý mã hoá và giải mã của mật mã CFB (theo 8 bit)*

Việc mã hoá chuỗi các ký tự được thực hiện theo phương pháp cộng modul-2 ký tự lấy ở đầu ra của thuật toán DES với ký tự bản tin rõ để tạo thành một ký tự được mã hoá. Ở phía thu cùng thực hiện phép cộng modul-2 của cùng ký tự lấy từ đầu ra của DES với ký tự đã được mã hoá để có ký tự của bản tin rõ. Cấu trúc của hệ thống như vậy đảm bảo rằng, dữ liệu được bổ sung thêm là hoàn toàn giả ngẫu nhiên.

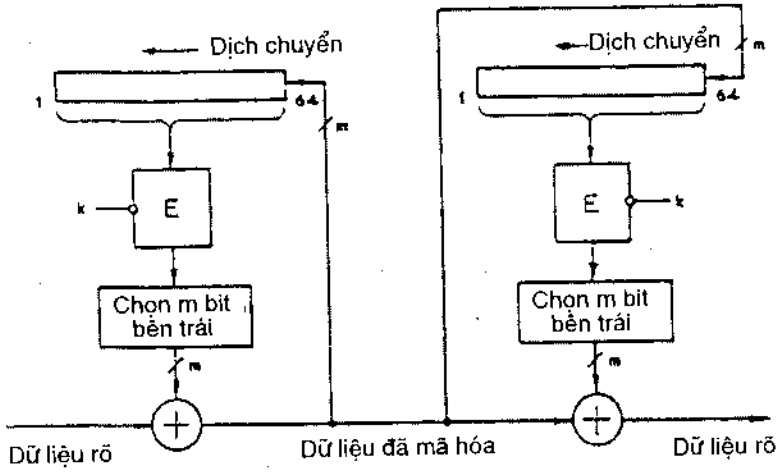
Trong khi mà phương pháp mật mã CBC thực hiện trên các khối dữ liệu hoàn chỉnh thì phương pháp mật mã CFB thực hiện mỗi lần theo một ký tự và chiều dài m có thể được chọn như một thông số trước kia. Chiều dài m nhỏ nhất là 1 bit và trường hợp này là mật mã CFC một bit. Về nguyên lý, giá trị m có thể từ 1 đến 64. Hiện nay trên các hệ truyền tin phổ biến nhất là sử dụng $m = 8$ như mô tả ở hình 2.26.

Cũng giống như mật mã CBC, phương pháp mật mã CFB liên kết các ký tự với nhau và làm cho bản tin được mã hoá vào toàn bộ bản tin rõ.

Khởi đầu thực hiện CFB thì thanh ghi dịch chuyển cũng phải có một giá trị khởi đầu. Thường sử dụng các giá trị khác nhau IV cho mỗi đoạn tin và trong thời gian thực hiện một vòng khoá mã để tránh đặc tính chu kỳ. Các giá trị IV có thể truyền công khai bởi vì chúng đã trải qua phép toán mã hoá.

Mật mã CFB cũng được sử dụng để mã hoá một chuỗi các ký tự mà mỗi ký tự được biểu thị bởi m bit. Trong một tập như thế, mỗi ký tự là một số nằm trong khoảng 0 và n-1 với $n = 2^m$. Các ký tự rõ cũng như các ký tự đã mã hoá đều nằm trong khoảng đó. Cũng có một số mã hiệu ký tự không

nằm trong khoảng 2*m* giá trị. Ví dụ trong trường hợp chỉ sử dụng các số thập phân 0, 1... 9. Một tập các giá trị như vậy cần lưu ý tránh các giá trị nhị phân tương ứng với các ký tự điều khiển. Ví dụ với mã ASCII có các giá trị ký tự điều khiển như: khởi đầu bản tin, kết thúc bản tin, bắt buộc thu, phát, thoát khỏi... mà điều đó dẫn đến hiểu nhầm là thể thức truyền dẫn trên mạng.



Hình 2.27. ***Mô tả sơ đồ khối nguyên lý mật mã CFB modulo-n với n < 2m***

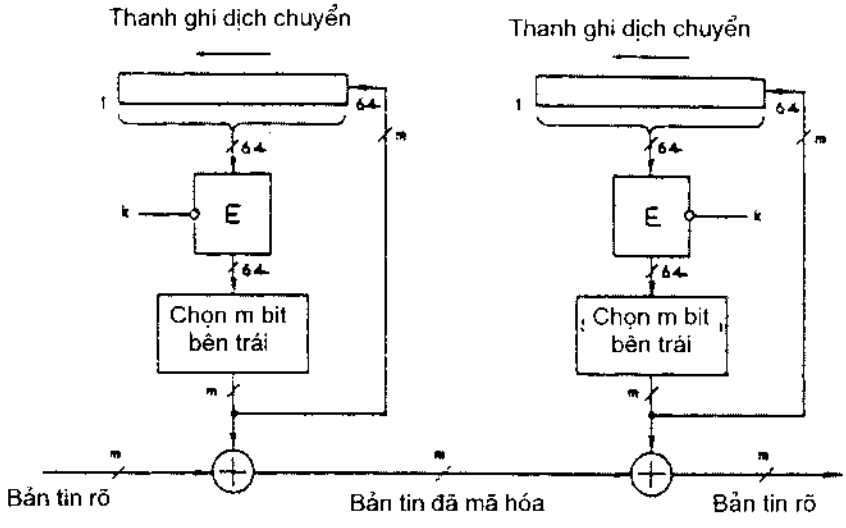
Nếu như n = 2^m thì có thể sử dụng mật mã CFB một cách bình thường. Trong trường hợp phải mã hoá các ký tự *m* bit với *m* là số nguyên khá nhỏ và $n < 2^m$ thì việc mã hoá và giải mã có hơi khác một ít. Hình 2.27 mô tả sơ đồ khối nguyên lý mật mã CFB trong trường hợp đối với một tập ký tự bất kỳ. Ở đây chuỗi các ký tự *m* bit là các bit có trọng số bé từ đầu ra của DES được cộng với các ký tự rõ. Cả hai phương pháp đều sử dụng các phép mã hoá cổ điển, chỉ khác là ở CFB bình thường thì phép cộng là cộng modul-2 gắn với phương pháp của Vernam còn ở đây là phép cộng modul-n gắn với phương pháp Vigenère.

2.3.5. Mật mã OFB

Trong số ba phương pháp mật mã được giới thiệu trên thì phương pháp mật mã ECB sử dụng hạn chế, còn hai phương pháp mật mã CBC và CFB được sử dụng tương đối thông dụng. Có một phương pháp mật mã thứ tư được dành riêng cho các ứng dụng mà quá trình truyền tin chấp nhận một sai số nào đó. Đó là phương pháp mật mã phản hồi từ đầu ra, OFB (output feedback). Mật mã OFB thường được sử dụng trong truyền tín hiệu số hoá âm thanh hoặc số hoá hình ảnh dưới dạng điều chế mã xung PCM, trên nền nhiễu bị sai số. Mã OFB sử dụng phương thức mã hoá kiểu Vernam, chỉ có nguồn giả ngẫu nhiên là mới. Cấu trúc của mã cũng gần giống như mã CFB, bởi vì nó có thể được sử dụng với các dãy ký tự *m* bit với m trong khoảng 1 và 64.

Hình 2.28 mô tả sơ đồ khối nguyên lý mật mã OFB, với các ký tự *m* bit và một đường phản hồi m bit. Nhìn lướt qua thì nó giống như mật mã CFB chỉ có khác ở phương thức thực hiện phản hồi. Chuỗi giả ngẫu nhiên được lấy từ mã DES và được phản hồi về chính nó. Việc đồng bộ cho các

bộ tạo giả ngẫu nhiên ở hai đầu đường truyền ở đây cần được chú ý. Nếu như việc đồng bộ không đảm bảo thì bản tin rõ sau khi mã hoá ở phía đầu thu cũng sẽ có dạng ngẫu nhiên. Để tạo lại đồng bộ ở phương pháp mã OFB thì các thanh ghi dịch chuyển phải được nạp các giá trị giống nhau. Các giá trị khởi đầu có thể gửi dưới dạng dữ liệu rõ bởi vì nếu đối phương nếu biết thì cũng không thể tạo được dãy giả ngẫu nhiên. Dãy giả ngẫu nhiên ở đây được cộng modul-2 với đoạn tin trong phép toán mã hoá và giải mã còn được gọi là dòng khoá (key stream).



Hình 2.28. *Mô tả sơ đồ khối nguyên lý mật mã OFB với m bit*

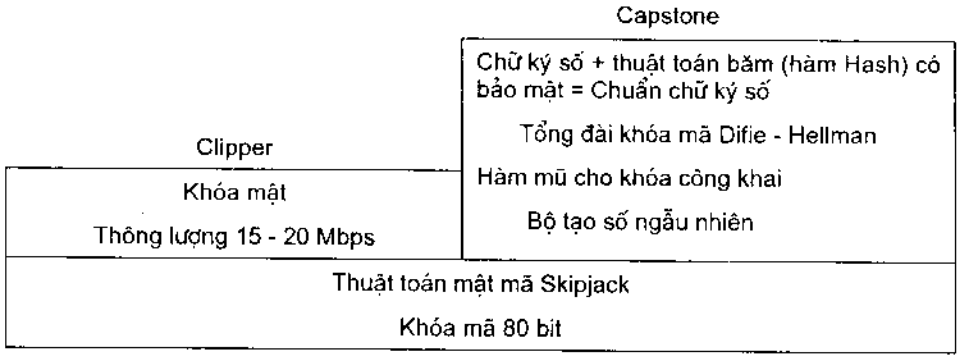
2.4. DES VÀ THUẬT TOÁN SKIPJACK

Thuật toán chuẩn DES được ứng dụng rộng rãi trong các hệ thông tin cần được bảo mật. Năm 1988 thuật toán được xem xét đánh giá lại và đến năm 1992 có một số phiên bản đổi mới. Biết rằng, đối với mật mã DES thì thuật toán mã hóa và giải mã là công khai, độ mật của mật mã phụ thuộc và không gian khóa hoặc kích thước của khóa. Số bít của khóa càng nhiều thì số khả năng càng lớn. Số khả năng của mật mã DES là 2^{56} hoặc lớn hơn 7.10^{16} một ít. Mật mã DES có một số nhược điểm như đã phân tích trong mục 2.2.4. Hơn nữa với các đường truyền tin hoặc mạng truyền tin chuyển mạch gói trong đó gói tin có trường tiêu đề và trường thông tin thì các gói tin đầu vào, đầu ra với khối tin cố định 64 bit của DES thì cũng cần phải tính toán sao cho phù hợp với các định khung của giao thức truyền dẫn sử dụng, ví dụ STM, ATM hoặc TCP/IP.

Trên các đường truyền tin có ghép kênh, theo tiêu chuẩn ghép kênh của Mỹ có định dạng khung chuẩn là 192 bit, như vậy cần liên kết ba khối DES để có 64x3 = 192 bit. Mật mã sử dụng 3 khóa mã 56 bit và được gọi là mật mã 3DES, thích hợp với các đường trực tuyến có ghép kênh theo chuẩn Mỹ. Chuẩn châu Âu phải sử dụng 4DES. Do sử dụng ba khóa mã 56 bit liên kết nhau cho nên độ bảo mật của mã cũng sẽ lớn hơn nhiều.

Phiên bản mới thay thế cho mật mã DES có khóa 56 bít là "*thuật toán Skipjack*", sử dụng không gian khóa 80 bit và sử dụng một thuật toán hoàn toàn khác. Thuật toán được giữ bí mật, không công bố. Đã từng có những chuyên viên nghiên cứu, khám phá và đánh giá nhưng cũng không công bố kết quả cụ thể.

Có một số chip cứng, ví dụ chip Clipper, được sản xuất để sử dụng thích hợp với thuật toán Skipjack và chữ ký số, có thể ứng dụng cho mật mã trong các hệ truyền tin bao gồm cả thoại, dữ liệu và facsimile. Chip Clipper đó sử dụng hệ thống khóa bí mật và được mô tả ở hình 2.29. Chip Clipper cũng có thể sử dụng thích hợp cho bảo mật trong các mạng số đa dịch vụ ISDN.



Hình 2.29. ***Chip Clipper và chip Capstone phối hợp sử dụng thuật toán Skipjack***

Việc phối hợp các chip cứng Clipper và Capstone như trên có thể kết hợp thực hiện các nhiệm vụ : mật mã dữ liệu (bao gồm mã hóa và giải mã), xác thực và bảo mật dữ liệu, tạo khóa và phân phối khóa công khai một cách mềm dẻo.

Thuật toán và các chip nêu trên chỉ mới được đưa vào trong một số trường hợp ứng dụng đặc biệt, chưa phải là chuẩn quốc tế.

Trong thực tế, mật mã DES ứng dụng trong các hệ thống truyền tin số hoặc truyền dữ liệu, thường được cài đặt cứng hóa trong các chip đặc biệt có phần mềm mã hóa và giải mã tăng cường. Phần mềm tăng cường đó có thể được nén, bảo vệ bằng các từ mật khẩu và sử dụng máy tính xách tay.

Viện tiêu chuẩn và công nghệ của Mỹ, NIST (National Institute of Standard and Technology) có quy định tập các quy tắc về các yêu cầu bảo mật, gồm 4 cấp bảo mật từ thấp đến cao cho các modul mật mã.

Mật mã DES sử dụng trong các hệ thống truyền tin hiện đại còn phải tính đến tốc độ truyền tin và định khung dữ liệu. Ví dụ với mạng số đa dịch vụ ISDN - 2B + D sử dụng chuẩn tốc độ 2 x 64 kbps + 16 kbps ; hoặc định khung ghép kênh theo chuẩn Mỹ là 192 + 1 bit và định khung ghép kênh theo chuẩn châu Âu là 256 bit (giới thiệu chi tiết ở Chương 8).

Bởi vì một số lớn phép toán của mật mã DES đã được thực hiện cứng hóa trong các chip cho nên mật mã DES có tốc độ nhanh hơn so với mật mã RSA. Tuy vậy mật mã DES cần sử dụng nhiều khóa hơn, cho nên việc quản lý khóa ở DES cũng phức tạp hơn. Trong nhiều hệ thống bảo mật trực tuyến, tốc độ cao, có thể phối hợp mật mã DES với mật mã có khóa công khai để tận dụng lợi thế tốc độ của DES và lợi thế quản lý khóa RSA (giới thiệu chi tiết ở các chương sau).

Viện Tiêu chuẩn và công nghệ của Mỹ, NIST (National Institute of Strandards and Technology) cũng đã đưa ra bốn mức bảo mật cho các modul mật mã (bảng 2.1).

Bảng 2.1. ***Các yêu cầu bảo mật đối với các modul mật mã***

Mức	Mô tả
1	- Mức bảo mật thấp nhất; hỗ trợ các chức năng mật mã cho các phần mềm ứng dụng, các bo mạch tích hợp và các thiết bị phụ trợ.
2	- Yêu cầu hệ điều hành tin cậy cấp C2 hoặc tương đương đối với các hệ điều hành nhiều người sử dụng; niêm phong và khóa chống các xâm nhập từ các máy tính xách tay nếu có yêu cầu.
3	- Yêu cầu hệ điều hành tin cậy cấp B1 hoặc tương đương đối với các hệ điều hành nhiều người sử dụng; cổng dữ liệu có bảo mật hoặc không bảo mật được cách ly vật lý; có khả năng xác thực và nhận dạng người sử dụng.
4	- Mức bảo mật cao nhất; yêu cầu hệ điều hành tin cậy cấp B2 hoặc tương đương đối với các hệ điều hành nhiều người sử dụng. Các hệ thống bảo mật mức 4 có thể phát hiện xâm nhập và sau đó tự động xóa các dữ liệu được lưu giữ.

Chương 3

QUẢN LÝ KHÓA MÃ

3.1. PHÂN CẤP KHOÁ

Chương hai đã đề cập một dạng mật mã (DES) dùng khóa bí mật, trong đó yếu tố bảo mật của hệ chủ yếu phụ thuộc vào khóa mã. Cũng vì vậy việc bảo vệ khóa mã đóng một vai trò quan trọng. Không những ở các thuật toán mật mã đối xứng mà ở các thuật toán xác thực, chữ ký số, bảo toàn thông tin và cả khóa giải mã ở mật mã không đối xứng thì việc bảo vệ khóa mã cũng được đặt ra một cách nghiêm ngặt. Nếu đối phương đánh cắp được hoặc khám phá được khóa mã thì việc giải mã không mấy khó khăn. *Quản lý khóa mã* cũng là một chức năng quan trọng của bảo vệ an toàn thông tin dữ liệu. Có nhiều phương pháp quản lý khóa mã, trong chương giới thiệu sau đây chỉ tập trung giới thiệu việc quản lý khóa mã liên quan đến truyền các đoạn tin trong mạng truyền tin.

Trong một mạng truyền dữ liệu lớn, các đoạn tin được trao đổi giữa nhiều thiết bị đầu cuối và các máy chủ, với sự tham gia của rất nhiều người sử dụng. Nguyên lý của các mật mã bao gồm việc cách biệt giữa những thiết bị đầu cuối, hoặc giữa những người sử dụng, họ không thể đọc được các bản tin của nhau nếu không được cấp quyền, không có trách nhiệm, theo phương thức của sự phân phối khoá. Có rất nhiều khóa mã và việc quản lý chúng cũng khá phức tạp.

Có những bài toán trong đó một máy chủ lưu giữ nhiều dữ liệu quan trọng của nhiều người sử dụng khác nhau và cần phải gửi một số trong các dữ liệu đó cho một số người sử dụng khác với một sự kiểm tra giám sát nghiêm ngặt. Một mô hình quản lý các khóa mã cần được thực hiện trong bối cảnh có nhiều người sử dụng, nhiều thiết bị đầu cuối, nhiều máy chủ được đặt ra cùng với việc bảo vệ các dữ liệu được truyền trên mạng truyền tin.

Ở đây, việc quản lý các khóa mã bao gồm tất cả các khái niệm của vấn đề xử lý khoá, từ khi tạo ra các khóa cho đến khi huỷ bỏ chúng. Vấn đề phức tạp hơn còn ở chỗ phân phối các khoá mã và lưu giữ chúng.

Muốn truyền các khóa mã qua mạng truyền tin thì khóa mã đó phải được mã hóa với một khóa mã khác. Ví dụ, nếu có một khóa *ks* được sử dụng để mã hóa các dữ liệu, và nếu khóa mã đó cần phải truyền qua mạng thì người ta sử dụng một khóa mã khác *kt* để mã hóa dưới dạng E*kt*(*ks*). Các khóa *ks* để mã hóa dữ liệu đó thường sử dụng trong một thời gian ngắn sau đó chúng được thay thế bằng các khóa mới.

Khóa mã *ks* được sử dụng chỉ trong một phiên liên lạc như vậy được gọi là *"khóa phiên liên lạc"* (sestion key). Khóa *kt* được sử dụng để vận chuyển là *"khóa thiết bị đầu cuối"* (terminal key). Khóa thiết bị đầu cuối được sử dụng trong một thời gian dài hơn khóa phiên liên lạc. Trong một số trường hợp, chúng được lưu giữ trong máy chủ cùng với các khóa thiết bị đầu cuối khác.

Để giảm bớt dung lượng nhớ cần phải được bảo mật, tất cả các khóa được nhớ lại phải được mã hóa bởi một khóa khác *km*, được gọi là *khóa chủ* (master key). Khóa *kt* được lưu giữ dưới dạng *Ekm*(*kt*).

Như vậy là hình thành một sự phân cấp bảo vệ khóa trong đó khóa chủ là cấp cao nhất (đỉnh).

3.2. TẠO KHOÁ

3.2.1. Tổng quan

Phương pháp lý tưởng để tạo khóa là xây dựng được một không gian khóa có xác suất đồng đều để có thể chọn một trong các giá trị có thể. Trong thực tế phương pháp đó khó thực hiện. Mọi sơ hở trong việc tạo khóa là kẽ hở cho đối phương xâm nhập tìm ra khóa để thám mã. Các phương pháp cổ điển cũng đã từng thử nghiệm các con xúc xắc để tạo ra các số ngẫu nhiên. Người ta cũng đã từng thử nghiệm trong hai xúc xắc đặc trưng cho hai số thấp lục phân và trong bảy lần cho 56 bit để dùng thử nghiệm với mã DES.

Việc tạo các khóa ngẫu nhiên bằng phương pháp thủ công như ví dụ trên không thích hợp với việc phân cấp khoá, có nhiều khoá con dưới khóa "bố mẹ, ông bà".

Hiện nay, để có thể tạo ra một số lớn các khoá, người ta thường sử dụng hai dạng nguồn các số ngẫu nhiên sau đây: nguồn vật lý thực tạo các bit ngẫu nhiên và các bộ tạo các số giả ngẫu nhiên.

3.2.2. Tạo các bit ngẫu nhiên

Biết rằng, tạp âm điện tử là một vấn đề không mong muốn đối với một bộ khuếch đại hoặc các hệ thống truyền tin điện tử, nhưng ở đây nó có thể là hữu ích. Tất cả các linh kiện và thiết bị điện tử đều tạo ra tạp âm (ngay cả khi tạp âm đó giảm ở một nhiệt độ rất thấp) và một bộ khuếch đại dải rộng có thể chuyển đổi tạp âm đó thành một tín hiệu để điều khiển đóng/mở một cổng. Có một phương pháp tạo các bit ngẫu nhiên bằng sử dụng giá trị vượt ngưỡng "không" của tín hiệu đó để sản sinh các xung làm thay đổi giá trị của một bộ đếm nhị phân. Khoảng giữa các xung biến đổi một cách ngẫu nhiên và sự phân bố xác suất của chúng phụ thuộc vào dải thông của bộ khuếch đại. Ở những khoảng thời gian điều hoà, trạng thái của bộ đếm nhị phân được lấy ra để cung cấp giá trị 0 hoặc 1. Các giá trị đó là một số nhị phân ngẫu nhiên, không tương quan với bất kỳ các giá trị nhị phân nào khác. Khi xây dựng các bộ tạo các bit ngẫu nhiên như vậy cần chú ý giảm thiểu độ tương quan. Phương pháp tạo các bit ngẫu nhiên như trên là phương pháp tương đối đơn giản và kinh tế. Trong thực tế có thể sử dụng các điốt hoặc các điện trở để tạo các bit ngẫu nhiên.

3.2.3. Tạo các số giả ngẫu nhiên

Có rất nhiều phương pháp toán học để tạo ra các số có vẻ ngẫu nhiên, nhưng khi sử dụng một số lớn trắc nghiệm thống kê thì người ta nhận thấy rằng, các bộ tạo các số giả ngẫu nhiên đó có tính

chu kỳ ở đầu ra. Nhưng một thuật toán tốt thì có thể dễ dàng tạo ra các dãy giả ngẫu nhiên và trong thực tế thuật toán đó là một thuật toán mật mã.

Ví dụ, nếu cố định giá trị k của khóa mã thì việc mã hoá dãy 0, 1, 2... sẽ tạo ra một dãy các số ngẫu nhiên mà mỗi một số của chúng chỉ có quan hệ với số trước nó. Nói một cách khác, có thể tạo ra một dãy R_n các số ngẫu nhiên bởi:

$$R_n = Ek(n)$$

Tuy vậy, ở đây có một vấn đề là, nếu hai dãy được tạo ra bởi phương pháp trên và cùng khóa thì chúng sẽ giống nhau. Do đó cần phải có các dữ liệu ngẫu nhiên để khởi đầu bộ tạo số ngẫu nhiên. Số ngẫu nhiên khởi đầu đó được gọi là các "mầm khoá" bởi vì chúng sản sinh ra một dãy các số. Một dãy ra của một mầm không phải là vô hạn, nhưng nó có thể rất dài. Trong ví dụ trên bộ tạo số giả mật mã khối tạo ra các số mật mã khối đến giá trị $n = 2^{64}$

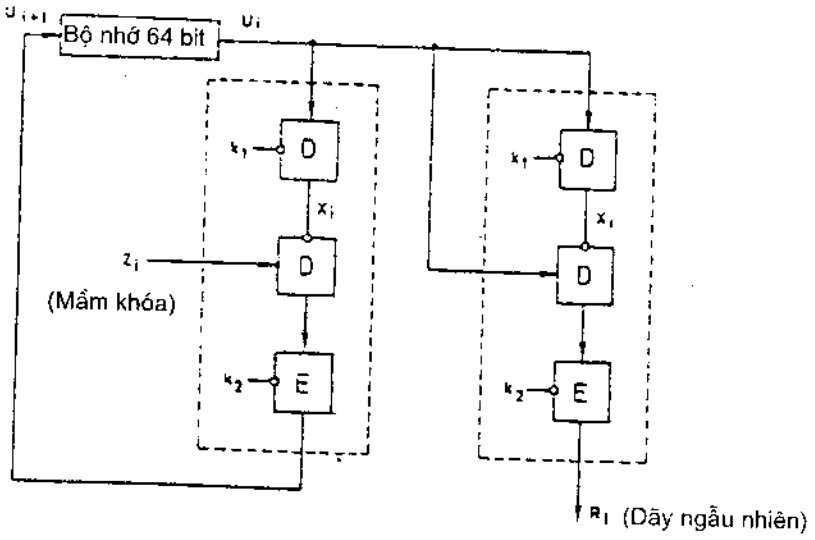
Mục đích của dãy ngẫu nhiên là để tạo ra các giá trị khoá mã và tất nhiên cả các giá trị của các biến số khởi đầu IV, cho mô hình quản lý khoá. Điều kiện ở đây là các khóa được tạo ra không thể phán đoán được, ngay cả khi nếu biết được các khoá trước đó. Trong bộ tạo số ngẫu nhiên đơn giản sẽ được giới thiệu sau đây, không thể tìm được giá trị n được sử dụng trong phép tính. Và ngay nếu như giá trị n đó được biết thì không thể tìm được giá trị mầm khóa và cũng không thể tìm được một giá trị ngẫu nhiên nào trong số các giá trị của các số ngẫu nhiên tiếp theo. Như vậy có nghĩa là, cần có một nguồn các giá trị mầm khoá lấy từ một quá trình ngẫu nhiên được tin cậy, ví dụ từ việc tung các con xúc xắc hoặc từ nguồn tạp âm ngẫu nhiên.

Với các hệ thống có yêu cầu bảo mật cao hơn thì các bộ tạo số ngẫu nhiên được kết cấu phức tạp hơn so với ví dụ trên một ít, sử dụng nhiều nguồn số ngẫu nhiên. Một số bí mật được chứa trong hệ thống thì nó cũng như một khóa chính ở mức cao, được bảo vệ cẩn thận, không được tiết lộ và được sử dụng trong các phép toán của hệ thống.

Một bộ tạo số giả ngẫu nhiên có thể sử dụng các phép toán đó mà không lo sự truy nhập bất hợp pháp từ ngoài hệ thống. Ví dụ, các giá trị mầm khóa được sử dụng để tạo ra dãy ngẫu nhiên có thể lấy từ các phép toán mã hóa (sử dụng khóa chính) được ứng dụng một số đặc trưng là ngày và giờ.

Hình 3.1. mô tả sơ đồ khối chức năng một bộ tạo các số ngẫu nhiên. Thanh ghi được mô tả ở thời điểm nó chứa giá trị u_i trước khi giá trị đó chưa thực hiện việc lặp lại kế tiếp. Các phép toán được mô tả trong hai hình chữ nhật đóng khung bằng các nét đứt là các chức năng mã hóa sử dụng các khóa chính bảo mật, ở đây ký hiệu là k_1 và k_2. Bài toán đó là bài toán quản lý khóa và được gọi là "mã hóa lại với khóa chủ sau giải mã", RTM (Re-encipher To Master key). Nó được ứng dụng trong quản lý khoá và chỉ có hệ điều hành mới có thể truy cập (không phải những người sử dụng máy chủ).

Giá trị xuất phát của bộ tạo là U_0 và nó sẽ tạo ra một dãy U_i bằng cách thực hiện các hàm mã hóa sử dụng hai khóa k_1 và k_2. Bộ tạo dãy ngẫu nhiên sử dụng một dãy phụ các giá trị mầm khóa Z_i, mà mỗi một giá trị đó có thể được xác định bằng cách đọc thời gian thực của một đồng hồ, với phép toán vào ra của thời gian không thể dự đoán được giữa hai lần đọc kế tiếp nhau. Phép toán U_i sử dụng các quan hệ sau:



Hình 3.1. *Mô tả sơ đồ khối chức năng bộ tạo các số ngẫu nhiên*

$$X_i = Dk_1(U_i)$$

$$u_{i+1} = Ek_2[DX_i(Z_i)]$$

Dãy ngẫu nhiên R_i nhận được xuất phát từ U_i theo quan hệ:

$$R_i = Ek_2[DX_i(U_i)]$$

3.3. CÁC KHÓA ĐẦU CUỐI VÀ CÁC KHÓA PHIÊN LÀM VIỆC

3.3.1. Nạp và huỷ khóa phiên làm việc

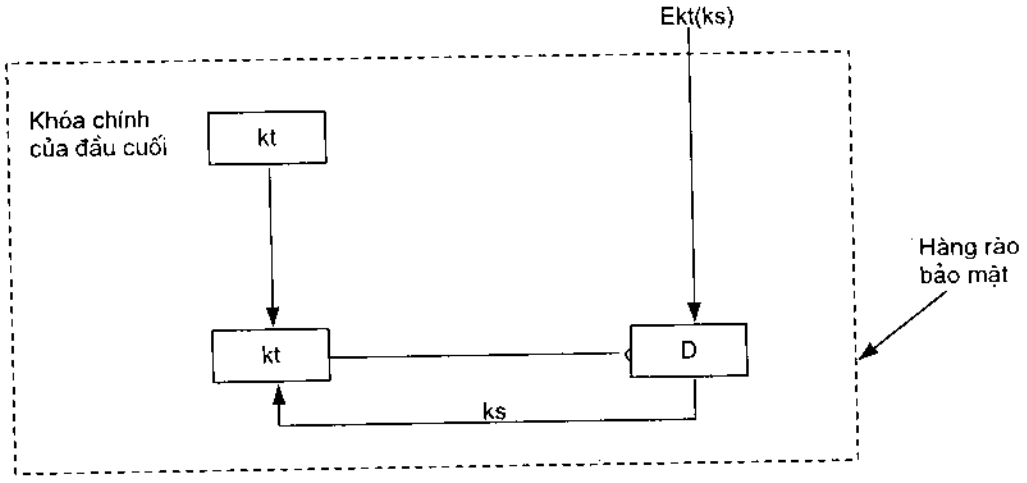
Giả thiết có một tập hợp n đầu cuối, gọi tập đó là T_1, hai đầu cuối một trong tập hợp đó muốn liên lạc với nhau. Nếu tất cả các khóa được cung cấp cho các cuộc kết nối thì mỗi đầu cuối phải lưu giữ N-1 khoá; và nếu mỗi khóa được sử dụng giữa hai đầu cuối theo hai chiều truyền dẫn thì phải có tất cả là $N(N$-$1)$ khoá. Trong thực tế, không phải cùng một thời điểm tất cả đầu cuối có cùng nhu cầu liên lạc. Thời gian mà hai đầu cuối liên lạc làm việc với nhau ở đây gọi là "*phiên liên lạc*" (sesion). Khóa mã cung cấp cho các phiên làm việc được gọi là "*khóa phiên làm việc*". Sau đây sẽ xác định thủ tục để thiết lập các khoá phiên làm việc khi cần thiết.

Việc sử dụng các khóa phiên làm việc làm gia tăng đáng kể tính bảo mật của hệ thống. Các khóa phiên làm việc chỉ tồn tại trong thời gian hai đầu cuối liên lạc, kẻ xâm nhập khó mà có khả năng truy nhập. Vấn đề còn lại ở đây là bảo vệ các khóa phiên làm việc.

Các khóa đầu cuối chỉ được sử dụng để phân phối các khoá phiên làm việc và các thông tin được truyền là dữ liệu ngẫu nhiên được xử lý bởi các khóa phiên làm việc. Như vậy, một tấn công vào khóa đầu cuối có thể khai thác được khóa phiên làm việc nhưng nếu tấn công vào khóa phiên làm việc phải hai lần thám khóa mã.

Khi mà hai đầu cuối T_i và T_j yêu cầu một khóa phiên làm việc ks_{ij} để liên lạc, thì lúc đó khóa có thể được truyền qua mạng theo hai phương pháp mã hóa là $Ekt_i(ks_{ij})$ cho đầu cuối T_i và $Ekt_j(ks_{ij})$ cho đầu cuối T_j. Các giá trị trên được tạo ra tại một trung tâm mà trung tâm đó có thể là một máy chủ điều hành các đầu cuối đó hoặc trong một mạng lớn thì có một trung tâm phân phối các khóa mã đảm nhiệm công việc đó (trung tâm này còn gọi là trung tâm bảo mật mạng).

Phép toán giải mã ở đây có thể sử dụng một mạch điện hoặc một phần cứng DES có khả năng lưu trữ hai khoá. Một trong các khóa đó, khóa *kt*, được sử dụng như là khóa chính của đầu cuối. Nó được lưu giữ trong một thanh ghi và có thể được đọc ra mỗi một lần khóa phiên làm việc cần phải nạp. Các giá trị mới được nạp trong thanh ghi các khóa làm việc. Thanh ghi đó cung cấp khóa cho các phép tính DES để giải mã khóa phiên làm việc đến và cho ra giá trị thực khóa phiên làm việc *ks*. Trong sơ đồ trên, giá trị *ks* không phải đưa ra mà nó được chuyển sau khi giải mã sang thanh ghi khóa làm việc, thay cho giá trị khóa *kt*. Với sự kết hợp với phần cứng như vậy, khóa phiên làm việc hoàn toàn được bảo vệ. Hình 3.2. mô tả sơ đồ khối của việc xử lý các khóa phiên làm việc ở đầu cuối.



Hình 3.2. ***Mô tả nạp một khóa phiên làm việc ở đầu cuối.***

Có hai phép toán khác trong quản lý các khóa ở các đầu cuối: đó là phép toán nạp khóa đầu cuối và phép toán huỷ khóa làm việc. Việc nạp khóa đầu cuối tham gia vào quá trình chuyển vận khóa đó đến đầu cuối và sẽ giới thiệu ở phần sau. Huỷ bỏ khoá phiên làm việc có thể được xử lý như một bài toán riêng, nó giống như việc nạp một khóa phiên làm việc với một giá trị giả, ví dụ như giá trị 0 được nạp. Việc huỷ khóa phiên làm việc thường diễn ra dưới sự điều khiển của máy chủ hoặc của trung tâm phân phối khoá, nhưng lưu ý rằng, một lệnh phát từ trung tâm đó có thể bị ngăn cản bởi kẻ xâm nhập. Trong nhiều trường hợp cụ thể, khóa phiên làm việc cần được xóa tại chỗ ngay sau khi kết thúc phiên làm việc.

3.3.2. Các đường phân phối khóa phiên làm việc

Hình 3.3. mô tả hai đầu cuối trong đó một khóa phiên làm việc cần được thiết lập. Trung tâm phân phối các khóa (có thể là một trung tâm điều khiển) cần phân phối sự nhận biết về khóa đầu cuối kt_1 cho đầu cuối 1 và khóa kt_2 cho đầu cuối 2. Các con đường trực tiếp nhất đối với các khóa

phiên làm việc được mô tả ở phần bên trái của hình (nhưng điều đó không phải lúc nào cũng thực hiện được trong thực tế).

Hình 3.3. Mô tả các đường phân phối khóa phiên làm việc

Quá trình làm việc được giải trình như sau, đầu tiên một trong các đầu cuối khởi đầu thiết lập liên lạc và ở đây giả thiết là đầu cuối 1. Đầu cuối 1 trước tiên cần liên lạc với trung tâm phân phối khóa và yêu cầu một khóa phiên làm việc, và cũng trình bày rõ phía đối tác làm việc là đầu cuối 2. Nếu như các khóa phiên làm việc được phân phối theo đường trực tiếp thì đầu cuối 2 nhận được khóa phiên làm việc trước cuộc gọi của đầu cuối 1. Có thể rằng, đầu cuối 2 trong thời điểm đó đang bận liên lạc với một đầu cuối khác, hoặc nó chưa muốn chấp nhận cuộc gọi. Tại thời điểm mà nó nhận được khóa phiên làm việc được mã hoá, nó cũng chưa biết rằng đầu cuối nào sẽ gọi nó. Trong bối cảnh lưu lượng lớn thì mỗi đầu cuối cần luôn luôn sẵn sàng để nhận nhiều khóa phiên làm việc mới và lưu giữ nó cho đến khi có cuộc gọi đến và lúc đó nó phải kết hợp với một trong các khóa đã được lưu giữ đó. Có những quy định cụ thể để tránh sai sót.

Có rất nhiều cách có thể truyền khóa phiên làm việc đến đầu cuối 2 qua đầu cuối trung gian 1, như mô tả ở phần bên phải của hình 3.3. Đầu cuối 1 lúc đó không những phải nhận khóa *ks* đã được mã hóa cho riêng nó, mà cả khóa phiên làm việc *ks* đã được mã hóa dưới dạng khóa mã kt_2. Sau đó nó cần phải chuyển giá trị thứ hai đó cho đầu cuối 2 bằng thiết lập cuộc gọi. Như vậy, về phía đầu cuối 1 sẽ có 2 pha trong thiết lập cuộc gọi:

- Nhận khóa đến từ trung tâm phân phối các khóa sau đó gọi đầu cuối 2.

- Truyền khóa phiên làm việc dưới dạng đã được mã hóa cho đầu cuối 2.

Như trên hình đã mô tả, khóa được truyền cho đầu cuối 2 là Ekt_2(ks), trong đó có đính kèm cả khóa phiên làm việc *ks* cho đầu cuối 2. Việc mã hóa dữ liệu đó theo khóa kt_1 cần được thực hiện theo kiểu móc xích như vậy cốt để kẻ xâm nhập không thể tách biệt hai phần của đoạn tin.

Phương pháp phân phối khóa như trên, nhìn bề ngoài có vẻ đơn giản nhưng cũng đã qua nhiều nghiên cứu thử nghiệm.

Còn một phương pháp thứ ba là, trung tâm phân phối khóa gửi cả hai khóa cho đầu cuối 2 và đầu cuối 2 trả về cho đầu cuối 1 một khoá.

3.3.3. Thể thức phân phối các khóa phiên làm việc

Hai pha phân khối khóa giới thiệu trên là việc trao đổi giữa đầu cuối 1 và đầu cuối 2, được gọi là "tậu khoá", và việc trao đổi giữa đầu cuối 1 và đầu cuối 2 ở đầu cuộc gọi được biểu thị bởi

"sự truyền khoá". Trong hai phép toán đó, việc mã hoá nhằm ngăn ngừa kẻ xâm nhập phát hiện được khóa phiên làm việc *ks*, nhưng nó chưa đủ để bảo vệ chống lại những xâm nhập tích cực. Cần phải ngăn chặn cả những kẻ "lừa đảo". Ví dụ trường hợp kẻ lừa đảo lợi dụng đường truyền tin để đóng vai một trung tâm phân phối các khoá, và cung cấp cho đầu cuối 1 một khóa đã được phân phối trước đó. Điều đó chúng có thể thực hiện được bằng cách lặp lại các đoạn tin trước. Một đầu cuối kẻ xâm nhập đóng vai trò đầu cuối 1 để được phân phối khóa sau đó gọi cho đầu cuối 2 bằng cách sử dụng một khóa được truyền trước đó.

Điều nguy hiểm ở đây là kẻ xâm nhập thiết lập một khóa phiên làm việc trước và bằng cách sử dụng nó thông qua trung tâm phân phối khóa và lừa đảo đầu cuối 1 sử dụng khóa đó hoặc thông qua như một đầu cuối 1 và bằng cách thiết lập khóa cũ để gọi lừa đảo đến đầu cuối 2. Các đoạn tin được trao đổi với nhau giữa các đầu cuối có thể được sử dụng lại các khóa đã dùng trước và kẻ xâm nhập không cần biết đó là khóa kt_1 hoặc khóa kt_2.

3.3.4. Xác thực trong pha tậu khoá

Khi có một cuộc gọi là trung tâm phân phối khoá, thì đầu cuối 1 cần phải kiểm tra là nó đang kết nối với một cách bố trí thông minh để biết được giá trị khóa kt_1 của mình và không phải là đó là của kẻ xâm nhập hoặc lừa đảo sử dụng lại các dữ liệu của một giao dịch cũ.

Sự xác thực đó (authentification) của trung tâm phân phối các khóa có thể được thực hiện bằng cách giao phó cho trung tâm phân phối khóa một nhiệm vụ phụ thuộc vào kt_1. Sự trao đổi sau đây thực hiện nhiệm vụ xác thực đó:

- Đầu cuối 1 gửi cho trung tâm phân phối khóa dưới dạng rõ:

$$a_1, r_1, a_2$$

- Trung tâm phân phối khóa gửi cho đầu cuối 1:

$$E_{kt_1}\{r_1, a_2, ks, E_{kt_2}(ks, a_1)\}$$

Dấu phẩy trong biểu thức biểu thị các yếu tố khác nhau tham gia trong đoạn tin chung. Khi một đoạn tin được mã hoá, nó sẽ được xử lý như một chuỗi. Có thể không nguy hiểm chọn giá trị 0 cho biến số khởi đầu IV, bởi vì mỗi một chuỗi bắt đầu với một số ngẫu nhiên. Các tham số a_1 và a_2 là địa chỉ của hai đầu cuối. Các số đó được gửi trong đoạn tin được truyền đến trung tâm phân phối khóa để nhận dạng có hai đầu cuối cần một khóa phiên làm việc. Tham số r_1 là một số ngẫu nhiên được đầu cuối 1 chọn cho cuộc trao đổi đó. Cùng số như vậy xuất hiện trong lời giải đã mã hóa và nó được đảm bảo rằng, nó không thuộc về một đoạn tin đã được lưu giữ trước, dù cho nó là một số ngẫu nhiên. Lời giải còn một tính chất nữa là, trong chuỗi đã mã hóa kt_2 bao gồm cả địa chỉ a_1 của bên đầu cuối gọi. Điều đó cho phép đầu cuối 2 kiểm tra xem đầu cuối 1 đúng là nguồn gửi tin hay không, bởi vì việc mã hóa với khóa kt_2 chứng tỏ rằng chuỗi đó đúng là được đến từ trung tâm phân phối khoá.

Người ta có thể tự đặt câu hỏi là nếu như trung tâm phân phối khóa cần phải xác thực đầu cuối 1. Một đầu cuối khác có thể có cuộc gọi và có các dữ liệu a_1, r_1 và a_2 như vậy. Điều đó không thể khai thác được, bởi vì muốn sử dụng các dữ liệu của trung tâm phân phối khóa thì phải giải mã chúng nhờ có khóa kt_1 mà khóa kt_1 chỉ có trung tâm phân phối khóa và đầu cuối 1 biết. Một sự xâm

nhập có thể làm thay đổi giá trị a_2 trong yêu cầu và làm sai lệch phiên giao dịch tiếp, nhưng việc lặp lại giá trị a_2 trong lời giải đã mã hóa có thể cho phép phát hiện ra. Trong pha "tậu khoá" việc xác thực theo nguyên lý chỉ được thực hiện một chiều. Người ta cũng mong muốn rằng, trung tâm phân phối khoá xác thực các đầu cuối gọi để tránh tình trạng có nhiều đầu cuối xâm nhập tạo ra nhiều cuộc gọi giả. Trung tâm phân phối khóa chỉ có thể xác thực một đầu cuối gọi dựa vào các biến số cho, bởi vì các dữ liệu cố định có thể được tạo ra bởi các đầu cuối lừa đảo. Đoạn tin gọi có thể được thay thế bởi:

$$a_1, Ekt_1(n_1,r_1,a_2)$$

Trong biểu thức trên, n_1 là một số của dãy được phân phối cho tất cả các cuộc gọi từ đầu cuối 1 đến trung tâm phân phối khóa kể từ khi thiết lập khóa kt_1. Người ta có thể sử dụng số liệu ngày và giờ cho giá trị n_1. Địa chỉ a_1 phải để ngoài phần được mã hóa để cho trung tâm có thể nhận dạng cuộc gọi từ đâu đến và sử dụng khóa tốt để giải mã.

Còn có những phương pháp khác để trung tâm phân phối khoá xác thực đầu cuối và các giới thiệu trên có tính nguyên lý.

3.3.5. Xác thực trong pha truyền khoá

Rõ ràng là, một sự xác thực tương hỗ giữa đầu cuối 1 và đầu cuối 2 là cần thiết, mục đích để cho mỗi một trong các đầu cuối kiểm tra sự đồng nhất của đầu cuối kia, và đảm bảo rằng không nhận phải sự lặp lại đoạn tin trước. Nếu như, bởi vì có sự xác thực của trung tâm phân phối khoá, đầu cuối 1 chắc chắn được nhận một khóa khóa phiên làm việc mới *ks*, lúc bấy giờ công việc là chuyển đổi mã hóa với đầu cuối 2 được thực hiện tiếp một cách bình thường để đảm bảo rằng đầu cuối đã trả lời. Không một đầu cuối nào khác có thể trả lời theo phương thức thông minh như trên, bởi vì việc thiết lập khóa *ks* như vậy, yêu cầu đầu cuối 2 phải mã hóa đoạn tin truyền khóa với khóa kt_2 của riêng mình. Nếu như đầu cuối 1 muốn tậu khóa sớm hơn lúc trao đổi, thì đầu cuối 2 có thể gửi một số ngẫu nhiên r_2 dưới dạng $Eks(r_2)$ và sẽ nhận được một trả lời có sự phụ thuộc của r_2 trong đoạn tin đã mã hóa đến từ đầu cuối 2.

Việc xác thực đầu cuối gọi từ đầu cuối 2 là có khó khăn hơn. Trường hợp đối phương phát hiện được khóa kt_1, thì rất nguy hiểm, bởi vì lúc đó chúng có thể nhận từ trung tâm khóa một loại các giá trị khóa *ks* để thực hiện các mưu đồ lừa đảo. Ngay cả khi đầu cuối 1 thay đổi khóa đầu cuối của mình thì việc sử dụng khóa cũ *ks* vẫn có thể thực hiện được việc lừa đảo.

Có một số tác giả đề xuất giải pháp trao đổi như sau:

- Yêu cầu từ đầu cuối 1 đến trung tâm phân phối khoá:

$$a_1, a_2$$

- Trả lời từ trung tâm phân phối khóa đến đầu cuối 1:

$$Ekt_1\{ks,a_2,d/t,Ekt_2(ks,a_1,d/t)\}$$

- Truyền khóa từ đầu cuối 1 đến đầu cuối 2:

$$Ekt_2(ks,a_1,d/t)$$

Trong các biểu thức trên, d/t biểu thị ngày và giờ.

Giá trị d/t cung cấp cho đầu cuối 1 được trung tâm phân phối khóa xác thực. Tất cả đầu cuối gọi trung tâm ở những khoảng thời gian trong một phút cần được bảo toàn một bảng liệt kê các giá trị khóa *ks* của phút cuối cùng và kiểm tra chung, không được lặp lại. Giá trị d/t cung cấp cho đầu cuối 2 khi truyền khóa nói lên rằng khóa phiên làm việc sẽ được tạo ra ngay sau đó. Các khóa nhận được từ trung tâm phân phối khóa phải được sử dụng ngay. Thời gian thiết lập cuộc gọi chỉ cần vài giây nếu như muốn mô hình trên làm việc tốt.

3.3.6. Phân phối các khóa đầu cuối

Ở đây chúng ta sẽ không đề cập đến các phương pháp khoá không dựa trên sự bảo mật (các phương pháp đó sẽ được giới thiệu ở chương sau về mật mã có khóa công khai) mà chỉ giới thiệu các phương pháp truyền thống thường được sử dụng trong các hệ thống mật mã đối xứng như mật mã DES. Trong trường hợp sự phân cấp khóa được sử dụng thì ở đầu cuối ít nhất có một khóa không phải gửi qua mạng. Trong hệ thống được giới thiệu sau đây, nó thuộc về khóa đầu cuối *kt*. Các khóa như vậy thường phải nạp thủ công ở đầu cuối và phải luôn luôn thay đổi theo thời gian để đảm bảo an toàn. Các khóa đó cần phải được vận chuyển bằng đường vật lý đến cho mỗi đầu cuối.

Phương pháp cổ điển để nạp một khóa ở một đầu cuối là sử dụng các bộ quay cơ khí hoặc phím bấm. Bài toán đó chỉ được thực hiện với những người hoàn toàn tin cậy, và đầu cuối thường được trang bị một bộ khóa vật lý để mở vào khóa mới. Các bài toán nạp khóa như vậy, có thể thỉnh thoảng bị thám bởi kẻ xâm nhập. Nếu như khóa đó được ghi trên giấy khi vận chuyển đến đầu cuối thì khó mà đảm bảo rằng, khóa đó không bị sao chép lại trên đường vận chuyển. Với tất cả những lý do trên, việc nạp khóa thường dùng phương pháp ghi nhớ điện tử dưới dạng modul chuyển vận.

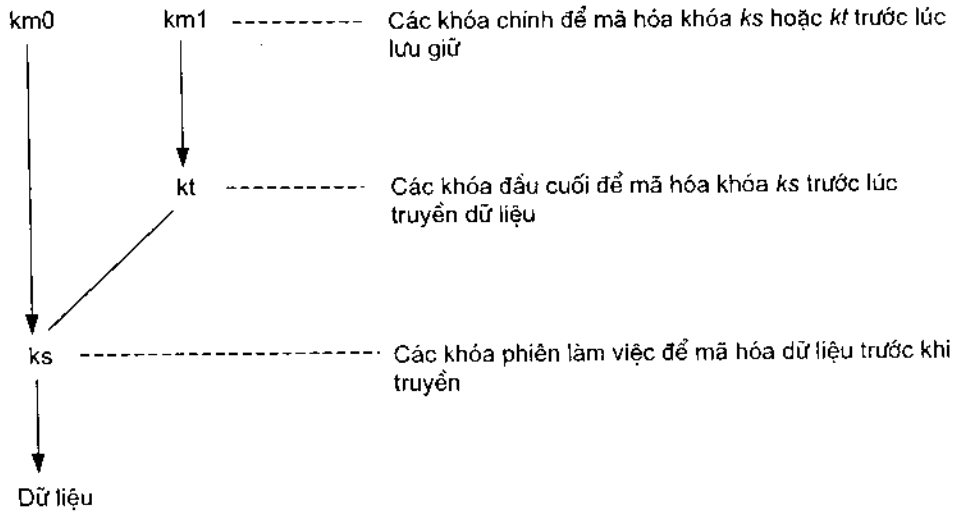
Modul chuyển vận khóa như vậy còn được gọi là" ống tiêm các khoá" (key gun), nó có kích thước như một máy tính bỏ túi, và là phương tiện ghép với đầu cuối để cung cấp khoá, với nguồn khoá khi có sự thay đổi khoá. Phương tiện ghép như vậy có thể là điện tử mà cũng có thể là quang học, và phương pháp quang học được ưa chuộng hơn vì rất khó phát hiện.

Một modul chuyển vận các khóa có thể chuyển vận nhiều khoá cho mỗi đầu nhận và có thể sử dụng để nạp các khóa theo nhiều cách theo hướng dẫn của trung tâm phân phối khoá. Các khóa được chọn một địa chỉ đã cho trong modul. Tất cả những gì cần ở bên trong modul đó là một bộ nhớ thích nghi, nó giống như một bộ nhớ E^2PROM (Electrically Ereasable Programable Read - Only Memory). Nhưng ở đây cũng không cung cấp bất kỳ một sự bảo mật nào để phòng ngừa khả năng đọc khóa trong quá trình vận chuyển. Để có một hệ thống đảm bảo hơn thì modul vận chuyển khóa thường được cấu trúc với một bộ vi xử lý và một chân cắm, loại có thể dùng cho bộ nhớ RAM, được xóa dưới sự điều khiển của bộ vi xử lý khi buộc phải đảm bảo bí mật.

Khi đọc một khóa không xóa có thể sử dụng mật khẩu để ngăn ngừa các cuộc đọc bất hợp pháp. Mỗi một nơi nhận chuẩn bị modul cho riêng mình và dùng riêng từ mật khẩu của mình. Nếu từ mật khẩu bị dùng sai nhiều lần thì các khóa sẽ bị xóa để để phòng kẻ xâm nhập lần tìm từ mật khẩu.

Hình 3.4 mô tả sự phân cấp khóa trong các hệ bảo mật truyền tin theo ba cấp. Ở cấp cao nhất có hai khóa chính là *km*0 và *km*1; ở cấp trung gian có các khóa đầu cuối, ký hiệu là *kt* và ở cấp thấp nhất là các khóa phiên làm việc *ks*.

Hai khóa chính của máy chủ, ký hiệu là *km*0 và *km*1 được sử dụng để mã hóa các khóa phiên làm việc và các khóa đầu cuối. Các khóa đầu cuối được lưu giữ trong các bộ nhớ của máy tính dưới dạng E*km*1(*kt*) và các khóa phiên làm việc được mã hóa dưới dạng E*km*0(*k*).



Hình 3.4. ***Sự phân cấp khóa trong bảo mật truyền tin***

Chương 4

MẬT MÃ CÓ KHÓA CÔNG KHAI

4.1. NGUYÊN LÝ CẤU TRÚC VÀ CÁC TÍNH CHẤT CỦA MẬT MÃ CÓ KHOÁ CÔNG KHAI

Phần lớn các mật mã đang sử dụng hiện nay sử dụng một khoá bí mật mà cả bên gửi cũng như bên nhận thông tin đều biết khóa mật đó. Các loại mật mã đó được gọi là "*mật mã đối xứng*" bởi vì cả A và B đều có khóa mật, cho phép trao đổi thông tin từ A đến B hoặc từ B đến A. Không thể phân biệt được A và B trong một hệ thống mật mã như vậy và kênh truyền phải được bảo vệ cả hai hướng.

Năm 1976, Diffie và Hellman công bố một phát kiến mới mang tên "các phương hướng mới trong mật mã" (*News directions in cryptography*). Công trình đề xuất một dạng mới của hệ thống mật mã, trong đó người gửi và người nhận sử dụng các khóa mã khác nhau nhưng có mối liên hệ với nhau, và một trong hai khóa mã đó được giữ bí mật. Bên nhận các dữ liệu giữ một khóa bí mật cho phép giải mã với khóa đó, còn bên gửi dữ liệu sử dụng một khoá khác để mã hóa dữ liệu và khóa mã hóa đó có thể công khai mà không sợ nguy hiểm phương hại đến hệ thống. Hệ thống dùng khóa như vậy được gọi là hệ thống "*không đối xứng*" bởi vì nó bảo đảm bí mật truyền tin chỉ một chiều. Thiết lập bí mật truyền tin theo hướng ngược lại phải có một cặp khoá thứ hai.

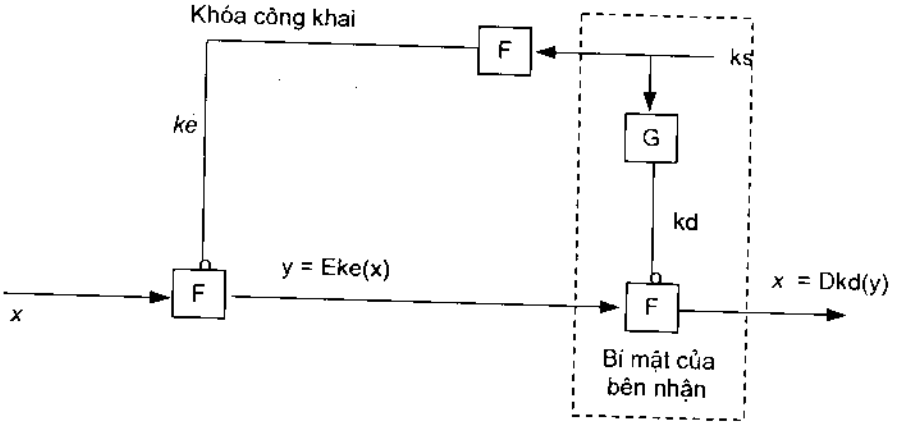
Một hệ thống mật mã có khóa công khai có cái lợi là không cần phải truyền khóa bí mật giữa A và B với sự thiết lập một kênh được bảo vệ. Việc truyền một khóa bí mật trên đường truyền có thể có nhiều nguy hiểm không thể lường trước được và các thể thức của chúng cũng phức tạp, rườm rà.

Việc lưu giữ một khóa bí mật cũng có nhiều nguy hiểm và trong một số trường hợp sử dụng mật mã có khóa công khai, các nguy hiểm đó được giảm thiểu. Hình 4.1. mô tả sơ đồ khối nguyên lý hoạt động của mật mã có khóa công khai.

Việc mã hóa sử dụng thuật toán mã hóa E và một khóa mã hoá *ke*. Việc giải mã sử dụng một thuật toán giải mã D và một khoá giải mã *kd*. Cả hai thuật toán đều công khai và là những phần tử không thể thay đổi được, dưới góc độ hệ thống. Cả hai khóa được tạo ra theo một phương pháp đặc biệt cho mỗi kênh mới được bảo vệ và cũng như tất cả các khóa khác, chúng có thể được thay đổi.

Để cho việc giải mã là một quá trình ngược với việc mã hoá, cả hai khóa mã hóa và khóa giải mã cần có một mối quan hệ với nhau và được chọn trong số rất nhiều khóa có thể. Trong hình 4.1, đặc trưng cho mối quan hệ đó là mầm khóa *ks*, hoặc còn gọi là khóa khởi thảo *ks* (seed) được chọn

theo phương pháp ngẫu nhiên. Hai thuật toán F và G được chọn để tính toán các khoá. Ở đây để phân biệt hai phía của hệ thống không đối xứng, chúng ta sẽ gọi là *bên gửi* và *bên nhận*.



Hình 4.1. ***Mô tả sơ đồi khối nguyên lý hoạt động của mật mã có khóa công khai.***

Chỉ có bên nhận thông tin là cần phải có biện pháp để giải mã và khóa giải mã *kd* phải được giữ bí mật, còn khóa *ke* được công khai cho phép một người nào đó có thể mã hóa dữ liệu. Ở đây, bên nhận sử dụng thuật toán F và G cùng với mầm khóa *ks* để tạo ra hai khoá: khóa giải mã *kd* bí mật được giữ lại sử dụng cho riêng mình và khóa *ke* công khai gửi cho bên gửi.

Một cách lý tưởng, các phép toán F, G và D của bên nhận được thực hiện bởi các vi mạch điện tử và tham số *kd* không bao giờ thất lạc, đó là phương pháp bảo vệ tốt nhất. Mỗi một lần huỷ bỏ số ngẫu nhiên xuất phát *ks*, bên nhận chỉ cần đảm bảo tính toàn vẹn vật lý của mạch. Có nhiều cách bố trí thực hiện việc giải mã, không được phân phối cho bất kỳ người nào khác và chỉ cho phép chính mình giải mã đoạn tin được mã hóa với khóa công khai do chính mình tạo ra.

Mật mã có khóa công khai trong nhiều trường hợp thực tế có chiều dài bản tin đã mã hóa bằng chiều dài của bản tin rõ. Như vậy, có thể biểu thị bản tin đã mã hóa y như một số biến hoá trong cùng một khoảng với x là bản tin rõ (trong các thuật toán sau sẽ sử dụng tính chất đó). Ở đây cũng nhận thấy rằng, hai thuật toán E và D là đồng nhất. Các kết quả mà chúng đưa ra khác nhau là bởi vì chúng sử dụng các khóa khác nhau.

Khóa mã hóa được công khai và bất kỳ ai cũng có thể sử dụng, không cần phải nhận thực đoạn tin nhận được. Kẻ phá hoại hoặc xâm nhập có thể ngụy tạo các đoạn tin giả và hệ thống (dạng đầu tiên) không thể báo điều đó cho bên nhận. Điều đó trái ngược với các mã đối xứng là bên gửi A và bên nhận B có cùng chung khóa bí mật, cùng chịu trách nhiệm. Trong trường hợp nếu như khóa mã bí mật ngoài A và B không một ai biết, thì A và B có thể tin rằng đoạn tin đã được giải mã một cách chính xác do phía đối tác gửi đến. Khi chúng ta nói "giải mã một cách chính xác" là giả thiết rằng có đầy đủ số dư trong đoạn tin để phía bên nhận có thể phân biệt đích xác giữa đoạn tin thực và các thông tin ngẫu nhiên.

Việc không có xác thực trong mật mã có khóa công khai có thể được chế ngự bằng nhiều phương pháp. Một trong những phương pháp thường dùng là chữ ký số, nó là một phương pháp quan trọng và đặc biệt hiệu nghiệm trong việc xác thực (sẽ giới thiệu ở chương sau). Khi mà A và B

đều tạo ra cả hai khóa dùng cho mật mã có khóa công khai và mỗi bên đều biết khóa công khai của bên kia, thì việc xác thực đoạn tin là điều cần thiết. Bên A đặt trong đoạn tin được mã hóa gửi cho B một số ngẫu nhiên R do B lựa chọn. Chỉ có B có thể giải mã đoạn tin đó và gửi trả lại R trong trả lời. Bên A có thể giải mã trả lời và kiểm tra lại số R đã được gửi trả. Bên A sau đó có thể xem xét đoạn tin được gửi trả về như một sự xác thực. Như vậy, việc xác thực theo hai hướng là có thể thực hiện được.

Người ta cũng có thể đặt ra một cầu hỏi khác với tính chất hoàn toàn công khai của phép mã hoá. Một kẻ xâm nhập có thể thực hiện một giả thiết đoạn tin và kiểm tra nó bằng phép mã hoá, sau đó so sánh kết quả với đoạn tin đã mã hoá. Cứ cố gắng thử nghiệm và có thể nhận được đoạn tin tốt? Với những khối dài của các đoạn tin (nó là cần thiết với các lý do khác), thì xác suất để có một giả thiết tốt là rất bé. Việc tìm ra các số lớn nguyên tố được sử dụng trong các phép mã hóa ở đây là một việc làm rất khó, có thể so sánh việc đó với các số ngẫu nhiên trong vùng 64 bit của mã DES. Mật mã có khóa công khai cho phép giải thoát nhu cầu chuyển vận khóa bí mật. Tuy vậy, nó cần phải đảm bảo sự xác thực rằng khóa công khai được bên gửi sử dụng. Một kẻ xâm nhập có khả năng lừa đảo bên nhận bằng cách chọn một khóa công khai nào đó mà chúng biết khóa giải mã quan hệ với khóa công khai đó. Tất cả các đoạn tin xuất phát từ người gửi sẽ được chúng giải mã. Để giữ cho việc liên lạc giữa bên gửi và bên nhận bề ngoài có vẻ được bình thường, kẻ xâm nhập phải mã hóa các đoạn tin được phát ra, mã hóa tiếp tục bằng khóa công khai của chúng, sau đó lại gửi cho người nhận chúng mong muốn. Điều đó trong thực tế là rất khó khăn cho kẻ xâm nhập để tiếp tục tạo các khóa giả để tiếp tục lừa đảo trong thời gian dài, bởi vì muốn làm điều đó phải nghe liên tục trên đường truyền tin. Liệu chúng có đạt được mong muốn với thời gian hạn chế. Cũng vì vậy mà bên gửi phải đảm bảo chắc chắn rằng, đó là khóa đúng.

Cũng đã nhiều tranh cãi và đề xuất về các thể thức cho các hệ bảo mật xử lý xác thực và độ mật của hệ thống.

4.2. CẤU TRÚC MỘT HỆ THỐNG MẬT MÃ CÓ KHÓA CÔNG KHAI

Các công bố đầu tiên về mật mã có khóa công khai chỉ đưa ra ý tưởng là có thể thực hiện những hệ như vậy, nhưng không chỉ rõ là các hàm D, E, F và G sẽ được thực hiện cụ thể như thế nào. Không bao lâu sau đó có ba sinh viên: *R.L.Rivest, A.Shamir* và *L. Adleman* của trường Đại học MIT (Massachusets Institute of Technology) công bố một ví dụ cụ thể về phương pháp thực hiện bài toán mật mã có khóa công khai. Cũng từ ngày đó (1978) mật mã có khóa công khai còn có tên gọi là mật mã RSA (chữ cái đầu tiên của tên ba sinh viên phát minh).

Theo sơ đồ hình 4.1 người ta tính rằng, khó để có một phương pháp thỏa mãn các hàm (hoặc thuật toán) khác nhau cần phải thỏa mãn nhiều điều kiện. Bởi vì khóa *ke* là đã biết, hàm mã hóa E sử dụng tham số đó để tạo ra y xuất phát từ x là một hàm công khai. Trong lúc đó yêu cầu của phép mã hóa là không được để cho kẻ xâm nhập có thể tính được x từ y. Điều đó trở về cách nói là hàm E phải là *"hàm một chiều"* (one-way function). Cũng như vậy, hàm F cũng phải là hàm một chiều. Vấn đề ở đây là làm sao suy ra được khóa *ks* từ khóa *ke* và tính được khóa *kd* giữ bí mật.

Hàm D cần phải có tất cả các tính chất của một hàm mã hoá. Ví dụ, cho các giá trị của một số khối của bản tin đã được mã hóa y và các khối rõ tương ứng x, nó phải không có khả năng tính

được khóa giải mã *kd*, bởi vì đó là phương pháp mà kẻ xâm nhập khai thác. Chúng có thể chọn bất kỳ một bản tin rõ x nào và sử dụng khóa công khai *ke* để tính theo hàm đã biết $y = Eke(x)$.

Các yêu cầu vừa nêu trên dường như không thích hợp giữa chúng với nhau, khi mà thuật toán E*ke* cần phải là một hàm một chiều, không thể nghịch đảo, trong lúc đó thuật toán D*kd* là nghịch đảo của hàm đó. Không biết được *kd*, hàm E không thể được nghịch đảo. Khóa bí mật *kd* cấu thành như một loại "cửa sập" (trapdoor) mà nếu biết được nó thì có thể nghịch đảo E một cách dễ dàng. Cũng vì lý do đó mà hàm mã hóa còn được gọi là "hàm một chiều với cửa sập". Thuật toán E*ke* là hàm một chiều. Cửa sập D*kd* cho phép nghịch đảo nó và nó có thể được suy luận ra nếu biết được khóa giải mã *kd*.

Một hệ thống mật mã có khóa công khai gồm các hàm mã hóa và giải mã cùng với các phương pháp tạo các cặp khóa (các mầm khoá) cốt để kẻ xâm nhập không có khả năng thám mã. Khó có hệ thống mật mã có khóa công khai thực sự thỏa mãn. Phương pháp đánh giá độ mật của một hệ thống cũng sử dụng các phương pháp giống như ở các hệ thống mật mã đối xứng. Cần phải giả thiết hết tất cả khả năng có thể xâm nhập (trong các trường hợp xấu nhất) để có biện pháp đề phòng. Thuật toán RSA cũng đã trải qua nhiều trắc nghiệm, nhưng biện pháp nào cũng không hoàn toàn tuyệt đối.

4.3. CÁC HÀM MỘT CHIỀU

Một hàm một chiều $y = f(x)$ có tính chất sau đây: cho giá trị x, có thể dễ dàng tính được giá trị y nhưng ngược lại, nếu cho giá trị của y thì nói chung là rất khó khăn để tính $x = f^{-1}(y)$. Một hàm như vậy thường được sử dụng để bảo vệ bảng lưu giữ các từ mật khẩu chống lại những kẻ xâm nhập tích cực như các mục trước đã đề cập.

Nếu biết một tập các cặp (x,y) có thể nghịch đảo f bằng các giá trị đó của y cụ thể. Bài toán nghịch đảo để tính x sẽ gặp rất nhiều khó khăn nếu y là ngẫu nhiên không tính đến trường hợp mà y tương ứng với một giá trị x đã biết.

Tất cả những quy luật của hàm số có thể thực hiện để nghịch đảo. Ví dụ, một tập các cặp (x, y) có thể được sử dụng để nội suy một hàm khác mà tất cả các sai số của x có thể được đính chính bằng cách tính $f(x)$ và lặp lại phép nội suy đó.

Từ đó có nhận xét rằng, không thể quyết đoán là hàm f không thể được nghịch đảo. Bởi vì, về nguyên lý, điều đó có thể thực hiện được. Tính chất một chiều của hàm ở đây có ý nghĩa là hàm khó khăn phức tạp cho phép tính. Phép tính $y = f(x)$ tương đối dễ, có thể thực hiện tương đối ít giai đoạn. Nhưng ngược lại, nếu cho y thì việc tính x sẽ gặp rất nhiều khó khăn, đặc biệt là trong các bài toán với số lớn.

4.4. LÝ THUYẾT CÁC SỐ VÀ ĐẠI SỐ HỮU HẠN

Các hệ thống mật mã có khóa công khai được xây dựng dựa trên lý thuyết các số. Cơ sở cần thiết để tạo nên *"đại số hữu hạn"* là nó sử dụng các số thông thường theo các quy tắc khác nhau. Tính chất cơ bản của đại số hữu hạn là nó chỉ sử dụng các tập hữu hạn các số thay vì tập vô hạn các số nguyên 0, 1, 2... mà đó là đối tượng của số học thông thường.

Các phép tính đại số thông thường chúng ta vẫn thường gặp trong cuộc sống hàng ngày. Tuy nhiên ở đây, đại số hữu hạn cũng được gọi là "đại số" bởi vì nó có quan hệ rất gần gũi với đại số thông thường, và nó sử dụng tên đó cho tiện trong việc thực hiện tên gọi các phép toán và các ký hiệu cho các phép toán như trong đại số thông thường. Ở đại số hữu hạn chỉ sử dụng giá trị các số nguyên như 0, 1, 2... và không sử dụng các số âm, không sử dụng phép toán phân số. Một đại số hữu hạn có thể được xây dựng độc nhất với hai số nguyên (0 và 1), hoặc với ba, bốn... hoặc một tập có kích cỡ ngẫu nhiên. Để thực hiện mã hoá, các phép đại số phải sử dụng các số rất lớn nhưng để giải thích về lý thuyết, người ta có thể dùng các tập số nhỏ.

Các mật mã thực hiện trên các đơn vị đoạn tin (các ký tự, các khối) có kích cỡ cố định và nó có thể lấy bất kỳ giá trị nào trong một tập hữu hạn. Kết quả phép toán mã cũng sẽ là một đơn vị có kích cỡ như vậy, nằm trong tập hữu hạn các giá trị.

Đại số hữu hạn với một modul m nào đó, biểu thị số các giá trị nguyên trong đại số được thực hiện và sẽ được giới thiệu ở mục sau. Ở đây cũng thực hiện các phép cộng, phép nhân cũng như nghịch đảo của chúng, các phép trừ và phép chia. Các tính chất của phép nhân sẽ được đơn giản hóa khi modul là một số nguyên tố, có nghĩa là số p đó không có số chia nào khác ngoài 1 và p. Với một modul nguyên tố, phép chia là luôn luôn thực hiện được, không cần sử dụng các số phân số.

4.5. HÀM MŨ VÀ SỰ PHÂN PHỐI KHÓA MÃ DÙNG HÀM MŨ

4.5.1. Hàm mũ và phép toán modulo

Các bài toán sơ đẳng của đại số thông thường như: phép đếm, phép cộng và phép nhân được định nghĩa phép toán này xuất phát từ phép toán kia. Phép đếm 0, 1, 2... là phép toán sơ đẳng nhất. Phép cộng b với a được định nghĩa là phép đếm xuất phát từ b với a bước. Nếu như a được tăng lên b lần cho một thanh ghi 0 lúc xuất phát thì kết quả đó là phép nhân b với a. Các hàm đại số còn nhiều và phức tạp hơn.

Nếu có một số bắt đầu từ 1, sau đó nhân với số a lặp lại n lần. Tích của nó là a lũy thừa n và được viết:

$$a^n = a.a...a \ (n \text{ số } a \text{ nhân với nhau})$$

Các phép toán được sử dụng trong đại số hữu hạn cũng sẽ dùng các ký hiệu như trên. Sau đây, sẽ xem xét hàm mũ và chỉ giới hạn với modul số nguyên tố p.

Phép toán a^n trên gồm hai số a và n mà chúng không thể đổi lẫn cho nhau. Có hai cách để miêu tả hàm đó dưới góc độ xem mỗi số là biến số.

Nếu xem là hàm của n thì đó là "*hàm mũ*", còn nếu xem là hàm của a thì đó là "*hàm lũy thừa*", nói chính xác hơn là lũy thừa n. Khi xem xét là hàm mũ, tức là xem a^n như một hàm của n. Biến số là n và có hai tham số hằng số, đó là modul p và gốc a.

Hình 4.2 mô tả một ví dụ cụ thể phép toán a^n (modulo-11) như đã phân tích trên. Ứng với mỗi giá trị của a và n có một giá trị tương ứng trong bảng. Giá trị $a = 0$ không sử dụng, bởi vì ở đây sử dụng nhóm tính nhân cho nên $a = 0$ không nằm trong nhóm đó.

Các giá trị a lớn hơn 10 không được xem xét trong phép toán modulo-11. Nếu nhìn theo hàng dọc thì ở hàng thứ 10 có giá trị tất cả bằng 1 và sau đó các giá trị của bảng lặp lại, bởi vì mỗi một dòng có kết quả như trước. Có thể giải thích điều đó:

$$a^{n+10} = a^n \text{ (modulo 11)}$$

Điều đó cũng giải thích một điều là, với phép modulo số ngẫu nhiên p (trừ trường hợp nếu $a = 0$) có:

$$a^{p-1} = 1 \text{ (modulo p)}$$

và với mọi a:

$$a^p = a \text{ (modulo p)}$$

Đây là cách để giải thích "định lý Fermat" và nó là cơ sở của lý thuyết các số. Hầu như tất cả các ứng dụng lý thuyết các số cho mật mã đều có quan hệ với nó.

n	1	②	3	4	5	⑥	⑦	⑧	9	10
0	1	1	1	1	1	1	1	1	1	1
1	1	2	3	4	5	6	7	8	9	10
2	1	4	9	5	3	3	5	9	4	1
3	1	8	5	9	4	7	2	5	3	10
4	1	5	4	3	9	9	3	4	5	1
5	1	10	1	1	1	10	10	10	1	10
6	1	9	3	4	5	5	4	3	9	1
7	1	7	9	5	3	8	5	3	4	10
8	1	3	5	9	4	4	9	5	3	1
9	1	6	4	3	9	2	3	7	5	10
10	1	1	1	1	1	1	1	1	1	1
11	1	2	3	4	5	6	7	8	9	10

Hình 4.2. ***Bảng các lũy thừa và các số mũ modulo-11***

Nếu với tư cách là hàm của n thì hàm mũ a^n là có chu kỳ, và chu kỳ đó là p - 1. Vậy cho nên n không được xem xét như một số theo đại số modulo p. Thực chất ở đây là phép đại số modulo p^{-1}. Để được chính xác hơn, có thể giảm sự phức tạp của biểu thức a^n với phép toán đại số modulo theo hai quy tắc sau:

1. Thu hẹp giá trị của a đến số dư modulo-p của nó.

2. Thu hẹp giá trị của n đến số dư modulo p-1 của nó.

Với hai quy tắc trên, tất cả phép toán của hàm mũ modulo-11 được mô tả trong phần ô vuông trên đường nét đứt của hình 4.2.

Trong bảng ở hình 4.2 cũng có một đặc điểm có thể nhìn thấy trực quan là, ở đòng thứ 10 chỉ có những giá trị 1 và ở dòng thứ 5 chỉ có hai giá trị là 1 và 10. Với phép toán modulo thì các dòng đó tương ứng $a^{(p-1)/2}$ và là căn bậc hai của $a^{p-1} = 1$. Căn bậc hai của 1 có thể là +1 hoặc -1. Nếu là -1 được hiểu là p-1 modulo p, tức bằng 10.

Có bốn cột trong bảng có các giá trị khác nhau, đó là các cột tương ứng với a = 2, 6, 7 và 8; bởi vì chúng chứa tất cả các giá trị số từ 1 đến 10, chúng chỉ hoán vị chỗ cho nhau. Nói một cách

khác, với các giá trị đó của a thì hàm mũ là một "*song ánh xạ*" (bijection). Khi sử dụng hàm mũ thì người ta sử dụng các giá trị đó của a. Trong thực tế, gần một nửa các giá trị có tính chất đó và được gọi là "các bộ tạo", bởi vì nó sản sinh ra tất cả các số của nhóm nhân bởi sự gia tăng các giá trị liên tiếp.

Khi sử dụng hàm a^n với tập các giá trị 0, 1, 2,...p-2 và a là một bộ tạo, thì hàm đó là một "*song ánh xạ*" với các giá trị trong tập 1, 2, 3...p-1. Trong các điều kiện đó, một nghịch đảo duy nhất có thể được xác định. Hàm và là nghịch đảo của hàm mũ $y = a^x$ là hàm logarit $x = \log_a y$. Hàm lôgarit cũng được lập như trong đại số thông thường, tuy nhiên ở đây kết quả của phép toán lôgarit luôn luôn là một số nguyên.

4.5.2. Hàm mũ dạng hàm một chiều

Một hàm mũ a^x với số sinh a và modulo p có dạng và được xem như là một hàm một chiều, bởi vì việc thực hiện phép toán số mũ thì dễ nhưng thực hiện phép toán lôgarit ngược lại là một công việc khó khăn.

Cũng có một số ngoại lệ cho tính chất một chiều đó như đã mô tả trong bảng ở hình 4.2. Phép toán lôgarit của 1 là 0 và của p - 1 là $(p-1)/2$. Trong thực tế ứng dụng cần tránh sử dụng các giá trị mà với các giá trị đó có thể thực hiện hàm nghịch đảo một cách dễ dàng. Có thể đạt được một ý tưởng tốt hơn của hàm bằng cách sử dụng một modul khá lớn.

Hình 4.3 mô tả một phần nhỏ của bảng kết quả phép toán tương ứng với $p = 999983$, một số lớn nguyên tố dưới một triệu. Nhìn lướt qua, bảng cho các giá trị ngẫu nhiên. Nếu lấy một góc khác của bảng kết quả, như mô tả ở hình 4.4 thì nhận thấy rằng, các kết quả có đặc tính không hoàn toàn ngẫu nhiên. Từ đó dẫn đến vấn đề là, trong thực tế ứng dụng, cần tránh các giá trị nằm gần giá trị 0 và giá trị p. Với các giá trị rất lớn của p được sử dụng trong các phép toán mật mã thì tỉ lệ không ngẫu nhiên của bảng trên là cực nhỏ, dưới 10^{-60}.

	8473	8474	8475	8476	8477	8478	8479
211	388600	125442	482038	713725	472600	893438	014498
212	663764	013579	341495	635933	298302	696132	930616
213	167980	070401	219323	259738	749030	822633	827194
214	318731	588206	794011	576705	635243	401132	897147
215	553663	542372	357618	234675	046456	854896	038732
216	550745	138460	864060	147589	814193	931487	414204
217	736025	329981	052991	985614	031395	281035	095420
218	445837	306526	603468	206282	139937	655224	079933
219	641110	545473	478238	475948	256111	083507	763416
220	217374	416776	135951	203826	861282	984365	114305
221	841199	819851	204309	658535	211631	588335	208558
222	600286	535473	548202	837537	026485	948909	478128
223	309740	675331	090932	084295	517153	113030	116230
224	471628	852168	661790	496558	960492	284626	530915
225	171976	394389	765586	897144	771971	100249	704802
226	177417	109200	451646	523812	109415	925455	117730
227	279792	376525	764909	724871	526714	140872	419216
228	717906	727080	713969	111044	030483	333114	592882
229	920932	380657	990125	224941	408777	188500	131937
230	189487	742243	452022	632318	261534	130166	712829

Hình 4.3. *Mô tả một phần của bảng các lũy thừa và các số mũ với phép toán modulo 999983*

Các số lớn được ứng dụng trong thực tế có thể có kích cỡ là 500 bit hoặc lớn hơn. Một phép tính như vậy là không thể nếu như bắt buộc phải thực hiện 2^{500} phép nhân hoặc một số lượng tương

đương. Số các phép nhân cần thiết nằm trong khoảng 500 và 1000. Cách đây hơn 2000 năm, các nhà toán học Ấn Độ đã đưa ra phương pháp tính, được mô tả ở hình 4.5 với số mũ là 41

Ở đây, số mũ x được biểu thị bởi một số nhị phân là 101001 và dãy kế tiếp 1,0... đó được sử dụng như một "chương trình" cho một dãy các phép nhân. Mỗi một pha thực hiện đó là:

	999976	999977	999978	999979	999980	999981	999982
1	999976	999977	999978	999979	999980	999981	999982
2	49	36	25	16	9	4	1
3	999640	999767	999853	999919	999956	999975	999982
4	2401	1296	625	256	81	16	1
5	983176	992207	996858	998959	999740	999951	999982
6	117649	46656	15625	4096	729	64	1
7	176440	720047	921858	983599	997796	999855	999982
8	764886	679633	390625	65536	6561	256	1
9	645696	922117	046841	737839	980300	999471	999982
10	480043	467196	765773	048593	59049	1024	1
11	639631	196773	171042	905611	822836	997935	999982
12	522498	819328	144773	777488	531441	4096	1
13	342446	083947	276118	889980	405543	991791	999982
14	602827	496301	619376	440012	783037	16384	1
15	780126	022143	903052	239918	650838	967215	999982
16	539016	867125	484655	040311	047452	65536	1
17	226820	797148	576674	838739	857627	868911	999982
18	412226	217027	116579	644976	427068	262144	1
19	114367	697804	417088	420045	718762	475695	999982

Hình 4.4. *Mô tả một phần của bảng các lũy thừa và các số mũ với phép toán modulo 999983*

- Bình phương số trước đó
- Nhân kết quả với a nếu nó tương ứng với bit 1.

Với 1 là giá trị lúc bắt đầu xuất phát, theo các pha thực hiện trên sẽ xác định được một cách nhanh chóng kết quả phép toán a^{41}. Số các phép nhân cần thiết với số mũ x là nằm giữa $\log_2 x$ và $2\log_2 x$ (hoặc $1\lg_2 x$ và $2\lg_2 x$). Trên hình 4.5 cũng mô tả ví dụ các giá trị của phép toán 7^{41} modulo97 và ở đây cũng chỉ dùng có 7 phép nhân.

41 = 101001

Số nhị phân	a^{41}	7^{41} modulo 97
1	1.a	1.7 = 7
0	a^2	49
1	$(a^2)^2.a = a^5$	73.7 = 26
0	$(a^5)^2.a = a^{10}$	94
0	$(a10)2.a = a^{20}$	9
1	$(a^{20})^2.a = a^{41}$	81.7 = 82

Hình 4.5. *Mô tả thực hiện phép toán a^{41} và 7^{41} modulo 97*

4.5.3. Độ phức tạp của phép toán lôgarit

Để chứng minh rằng, $y = a^x$ là một hàm một chiều thì cần phải chứng minh hàm lôgarit là đặc biệt khó khăn khi tính toán nó. Đã có một thời gian dài người ta đã dùng nhiều phương pháp

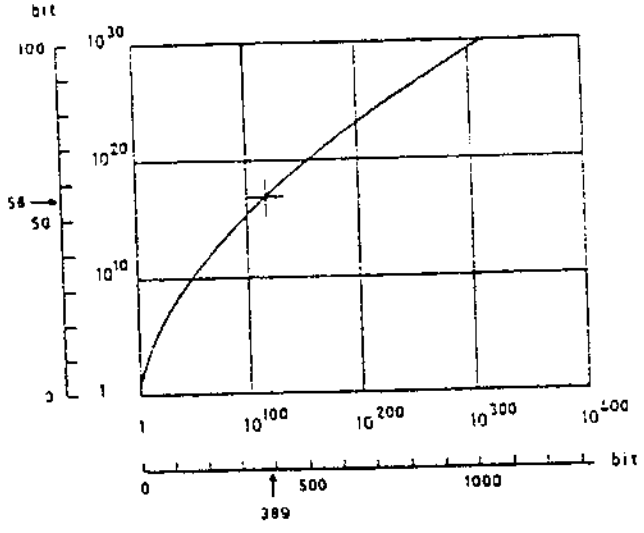
tính toán khác nhau, kể cả sử dụng các bộ nhớ, để tìm lời giải của bài toán $y = a^x$ và bài toán nghịch đảo của nó. Vấn đề liên quan đến các bài toán mật mã đã dẫn đến hai hướng phát triển.

Hướng thứ nhất (của tác Pohlig và Hellman) đã đề xuất một phương pháp nhanh hơn để tính phép toán lôgarit khi p-1 không có thừa số lớn. Độ phức tạp của phương pháp phụ thuộc vào số các thừa số của p - 1 luôn luôn chia hết cho 2 cho nên trường hợp phức tạp nhất là lúc $(p - 1)/2$ là số nguyên tố. Cũng không khó khăn lắm trong việc tìm các số nguyên tố loại ấy. Ngược lại, nếu muốn đơn giản phép toán lôgarit thì có thể chọn một số nguyên tố dạng $2^n + 1$, và phương pháp cho độ phức tạp tỉ lệ với giá trị n.

Hướng thứ hai (của tác giả Aldeman, một trong ba tác giả mật mã RSA) là sử dụng một số nguyên tố trong đó p - 1 có một thừa số lớn. Số các bài toán sơ cấp cần sử dụng để tính phép lôgarit modulo.p là xấp xỉ:

$$\exp \sqrt{[\ln p.\ln(\ln p)]}$$

Độ phức tạp của phép lôgarit theo phương pháp này được mô tả bởi đồ thị trên hình 4.6. Việc thực hiện bài toán nghịch đảo bằng các phép toán lôgarit như trên là khá phức tạp và đủ khó thường được sử dụng để ngăn ngừa xâm nhập trong mật mã (sẽ trở lại sau ở thuật toán RSA).



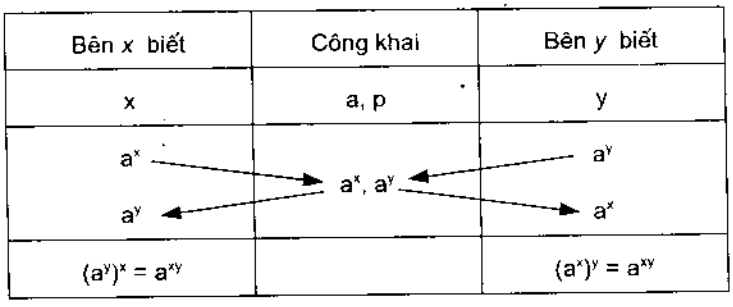
Hình 4.6. *Mô tả hàm* $y = exp \sqrt{\ln x.\ln(\ln x)}$.

4.5.4. Phân phối khoá

Trước đây, trong một công trình công bố, Diffie và Hellman đã đề xuất một phương thức cho phép việc phân phối các khóa bí mật bằng các đoạn tin không mật. Nguyên lý của phương pháp được mô tả ở hình 4.7.

Để bắt đầu, cả hai bên gửi và nhận (ở đây ký hiệu là bên X và bên Y) chọn một số nguyên tố lớn p và xây dựng thuật toán theo kiểu phép tính lôgarit của Pohlig và Hellman đã giới thiệu ở mục trên với độ phức tạp của phép lôgarit đủ lớn.

Chọn số sinh a cho phép đại số modul-p. Các số đó được trao đổi giữa hai bên theo phương thức truyền các đoạn tin không cần bảo mật và được miêu tả ở trong hình 4.7 dưới các tham số công khai là a và p.



Hình 4.7. Mô tả phương thức phân phối khóa theo hàm mũ

Tiếp theo, mỗi bên chọn ngẫu nhiên một số nguyên tự nhiên bí mật giữa 0 và p - 1 (tránh các giá trị gần hai biên). Các giá trị mật đó được ký hiệu là x và y. Hai bên sau đó tính các hàm mũ a^x và a^y tương ứng.

Pha tiếp theo, các giá trị a^x và a^y được hai bên trao đổi với nhau, chúng cũng là công khai bởi vì chẳng có bí mật nào trên mạng truyền tin.

Để kết thúc, mỗi bên thực hiện một phép số mũ phụ. Đối với bên x, giá trị nhận được a^y sẽ được nâng lên lũy thừa x. Đối với bên y, giá trị nhận được a^x sẽ được nâng lên lũy thừa y. Như vậy, cả hai bên đã tính được giá trị a^{xy} và chúng được sở hữu như số bí mật. Tất cả các phép tính trên được thực hiện với đại số modulo-p, kết quả là một số của tập 0, 1..., p - 1.

Ở đây cả hai bên cùng chịu trách nhiệm chung về bảo vệ bí mật. Phương thức này không dùng để truyền dữ liệu. Giá trị mật trên có thể được sử dụng để tạo một khóa chung cho hệ mật mã đối xứng ví dụ như mã DES.

Độ mật của phương thức trên phụ thuộc vào độ khó của phép tính logarit. Trong trường hợp kẻ xâm nhập biết được modul của phép đại số và giá trị a thì xuất phát từ các giá trị trao đổi a^x và a^y chúng cũng không thể tìm được x hoặc y bằng phương pháp tính logarít rất phức tạp.

Phương pháp phân phối khóa theo hàm mũ trên cũng có một giới hạn trong thực tế, đó là hai bên x và y không có một trao đổi thông tin gì khác ngoài việc trao đổi như trong hình 4.7. Điều đó sẽ khắc phục bằng biện pháp xác thực (sẽ giới thiệu ở phần sau) nếu không muốn để kẻ lừa đảo lợi dụng các sơ hở về xác thực.

4.6. HÀM LŨY THỪA VÀ MẬT MÃ DÙNG HÀM LŨY THỪA

Hình 4.8. mô tả hàm lũy thừa $y = x^n$ với đại số modulo-23. Hàm lũy thừa bậc m được biểu thị bởi dòng thứ m trong bảng.

Trong khi mà hàm mũ có đặc tính là rất khó thực hiện nghịch đảo, thì đó là một đặc tính của hàm lũy thừa để có được một hàm nghịch đảo mà nó cũng chính là một hàm lũy thừa. Ví dụ:

$$5^{13} = 21 \text{ và } 21^{17} = 5 \text{ (modulo 23)}$$

Sau đây sẽ kiểm tra quan hệ đó xem có đúng với tất cả các giá trị của x không.

Nếu $y = x^{13}$, như vậy $y^{17} = x^{13.17} = x^{221} = x$. Số mũ 221 được giảm xuống còn 1 bởi vì $x^{22} = 1$ (với x khác không) theo định lý Fermat mà theo đó $x^{220} = 1$.

Trong trường hợp tổng quát, với các hàm lũy thừa modulo-p, các số mũ m và n tương ứng với các hàm nghịch đảo lẫn nhau, nếu như $y = x^m$ thì sẽ kéo theo $x = y^m$, có nghĩa là nếu $x^{m.n} = x$ (modulo p). Bởi vì các phép tính của số mủ được hiện theo modulo p-1, cho nên điều kiện để m và n sản sinh ra các hàm lũy thừa nghịch đảo là:

$$m.n = 1 \text{ (modulo } p\text{-}1)$$

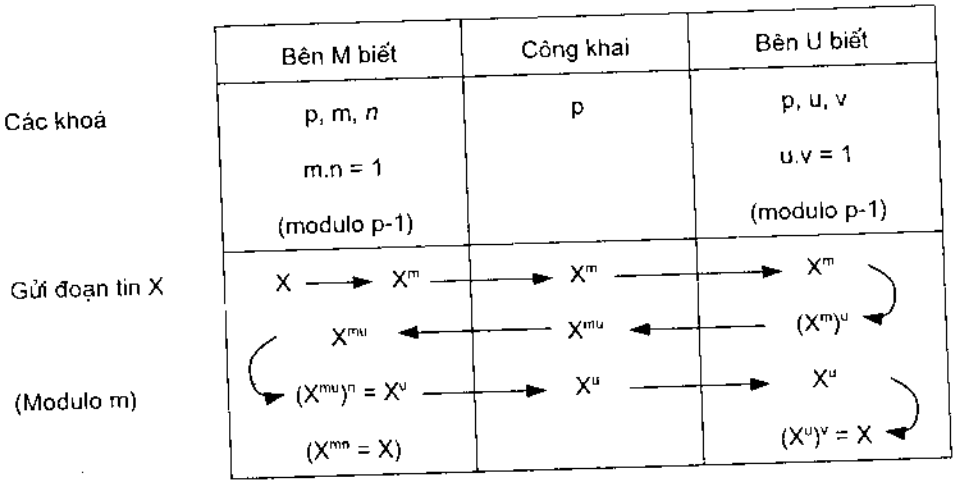
Không thể chọn m và n bất kỳ, bởi vì tất cả các hàm luỹ thừa đều ánh xạ. Ở đây chúng ta lại cần các hàm lũy thừa không thể nghịch đảo. Các hàm có tính chất đó được mô tả trong bảng của hình 4.8 bởi các dấu sao (*) ở bên trái dòng tương ứng. Đó là các dòng 1, 3, 5, 7, 9, 13, 15, 17, 19, 21. Các giá trị không thừa nhận đều là bội số của 2 hoặc 11, mà đều là các thừa số của p-1 = 22. Điều kiện tổng quát là m và n phải là các số nguyên tố với p-1. Nếu như n đúng trong trường hợp đó thì sẽ tồn tại một giá trị m và $m.n = 1$ (modulo p-1), và nó cũng là một số nguyên tố với p-1.

Ví dụ với phép toán modulo-23 là khá bé để có thể biểu thị hết tất cả các cặp của hàm lũy thừa nghịch đảo. Có thể thử các ví dụ sau đây trên bảng.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
						•		•			•	•			•	•		•		•	•	•	
0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
2		1	4	9	16	2	13	3	18	12	8	6	6	8	12	18	3	13	2	16	9	4	1
3	*	1	8	4	18	10	9	21	6	16	11	20	3	12	7	17	2	14	13	5	19	15	22
4		1	16	12	3	4	8	9	2	6	18	13	13	18	6	2	9	8	4	3	12	16	1
5	*	1	9	13	12	20	2	17	16	8	19	5	18	4	15	7	6	21	3	11	10	14	22
6		1	18	16	2	8	12	4	13	3	6	9	9	6	3	13	4	12	8	2	16	18	1
7	*	1	13	2	8	17	3	5	12	4	14	7	16	9	19	11	18	20	6	15	21	10	22
8		1	3	6	9	16	18	12	4	13	2	8	8	2	13	4	12	18	16	9	6	3	1
9	*	1	6	18	13	11	16	15	9	2	20	19	4	3	21	14	8	7	12	10	5	17	22
10		1	12	8	6	9	4	13	3	18	16	2	2	16	18	3	13	4	9	6	8	12	1
11		1	1	1	1	22	1	22	1	1	22	22	1	1	22	22	1	22	1	22	22	22	22
12		1	2	3	4	18	6	16	8	9	13	12	12	13	9	8	16	6	18	4	3	2	1
13	*	1	4	9	16	21	13	20	18	12	15	17	6	8	11	5	3	10	2	7	14	19	22
14		1	8	4	18	13	9	2	6	16	12	3	3	12	16	6	2	9	13	18	4	8	1
15	*	1	16	12	3	19	8	14	2	6	5	10	13	18	17	21	9	15	4	20	11	7	22
16		1	9	13	12	3	2	6	16	8	4	18	18	4	8	16	6	2	3	12	13	9	1
17	*	1	18	16	2	15	12	19	13	3	17	14	9	6	20	10	4	11	8	21	7	5	22
18		1	13	2	8	6	3	18	12	4	9	16	16	9	4	12	18	3	6	8	2	13	1
19	*	1	3	6	9	7	18	11	4	13	21	15	8	2	10	19	12	5	16	14	17	20	22
20		1	6	18	13	12	16	8	9	2	3	4	4	3	2	9	8	16	12	13	18	6	1
21	*	1	12	8	6	14	4	10	3	18	7	21	2	16	5	20	13	19	9	17	15	11	22
22		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
24		1	4	9	16	2	13	3	18	12	8	6	6	8	12	18	3	13	2	16	9	4	1

Hình 4.8. *Bảng giá trị các hàm lũy thừa và các hàm mũ modulo-23*

Việc sử dụng các khóa bấm để chuyển vận tài liệu mật mã không cần chuyển chìa khóa cho bên nhận như mô tả trên cũng tương đương như việc sử dụng các hàm lũy thừa trong mật mã. Hai khóa bấm tương đương với một mã khóa kép. Hình 4.10 mô tả quá trình mật mã đó. Hai phép mã hóa được ứng dụng cho đoạn tin gửi là loại có thể bắt đầu bởi nghịch đảo số nguyên tố, ngược lại với nguyên lý chung là bắt đầu bởi nghịch đảo thuật toán cuối cùng được áp dụng.



Hình 4.10. ***Mô tả phương pháp mật mã dùng hàm lũy thừa (mô hình tương đương với hình 4.9).***

Bên gửi M đóng và mở bằng cách áp dụng hàm lũy thừa với các số mũ m và n thỏa mãn quan hệ $m.n = 1$ (modulo p-1). Một quan hệ tương tự như vậy cho phép bên nhận U đóng và mở với mã hóa của mình với các số mũ u và v. Lần gửi đầu tiên, đoạn tin x được gửi dưới dạng được mã hóa x^m. Đoạn tin quay lại được mã hóa lần thứ hai dưới dạng x^{mu}. Người gửi M có thể rút lại mã hóa của mình nhờ số mũ n để đưa đoạn tin chỉ còn mã hóa x^u, sau đó gửi trả lại cho bên nhận U, và bên U sẽ giải mã đoạn tin đó theo hàm mũ: $x^{uv} = x$.

Có thể hình dung phương thức đó như trò chơi trộn và chia bài: tên của 52 con bài trước tiên được mã hoá, sau đó trộn bởi S và trao cho người phân phối D, ở đó sẽ trộn và chia các tên của các con bài đó đã được mã hoá. Mỗi một người chơi (trừ S có thể giải mã những con bài của mình) sẽ mã hóa các đoạn tin nhận được với khóa riêng của mình và gửi cho S để tháo khóa mã của S và gửi kết quả lại cho mỗi một người giữ bài. Những người chơi sau đó có thể giải mã các tên để xem những con bài của mình là gì.

Phương pháp truyền như trên có quan hệ chặt chẽ với phương pháp dùng hàm mũ để phân phối khoá. Nó cũng có một nhược điểm là không có xác thực người mà gửi lại đoạn tin x^{mu}, cũng như không đảm bảo rằng đoạn tin đó đã đến đúng địa chỉ người nhận.

4.7. MẬT MÃ CÓ KHÓA CÔNG KHAI RSA.

4.7.1. Nguyên lý cấu trúc mật mã RSA, mã hóa và giải mã.

Một trong những hệ mật mã có khóa công khai ra đời đầu tiên là mật mã RSA (Rivest, Shamir và Adleman). Trong phương pháp mật mã RSA, việc mã hóa và giải mã dùng hàm lũy thừa:

$$y = x^e \text{ và } x = y^d \text{ (modulo m)}$$

trong đó: x là đoạn tin, e là khóa công khai được sử dụng để mã hoá, y là đoạn tin đã được mã hóa và d là khóa bí mật được dùng để giải mã.

Không thể sử dụng các hàm lũy thừa với một modul số nguyên tố như là một thuật toán cho mật mã có khóa công khai, bởi vì các khóa mã hóa và khóa giải mã có thể suy từ cái này ra cái kia không mấy khó khăn. Cái mới trong thuật toán RSA là sử dụng một modul không nguyên tố.

Mật mã RSA sử dụng modul là tích của hai số lớn nguyên tố khác nhau được ký hiệu là p và q. Về toán học biết rằng: nếu cho hai số lớn nguyên tố p và q, xây dựng tích $m = p.q$ là rất dễ nhưng ngược lại cho m, yêu cầu phân tích m thành hai thừa số p và q là một công việc khó khăn. Cũng giống như trong tất cả các hệ thống có khóa công khai, bên nhận thông tin xây dựng các khoá của mình từ hai số lớn nguyên tố. Sau đó báo cho bên gửi biết tích m để xây dựng khóa công khai; cả bài toán mã hóa và giải mã đều cần biết tích m. Cái cốt yếu của mô hình RSA là phía nhận tin có thể giữ bí mật p và q mà chỉ công khai m.

Thực chất việc phân tích m thành thừa số p và q tuy khó nhưng không phải là không có khả năng đối với kẻ xâm nhập nếu số p và q không phải là số lớn nguyên tố.

Có thể giải thích thuật toán RSA bằng một ví dụ đơn giản sau đây với $m = 15$ và các thừa số p và q là 3 và 5. Hình 4.11 mô tả bảng kết quả các lũy thừa. Để tìm một phương pháp có hiệu quả, trước tiên hãy xem xét cấu trúc có tính chu kỳ trong bảng. Dòng đầu tiên tất cả có giá trị 1 và không bao giờ lặp lại. Chu kỳ là bằng 4. Trong trường hợp tổng quát, chu kỳ được ký hiệu là λ. Trong ví dụ ở bảng, $m = 15$. Có một định lý gần với định lý Fermat thừa nhận rằng, với mọi x có:

$$x^{\lambda+1} = x \text{ (modulo m)}$$

Phép đại số trong số mũ của các biểu thức trên là phép đại số modulo λ, ngoại trừ trường hợp không đúng là khi lấy $x^{\lambda} = x^{0} = 1$.

Từ đó có thể thực hiện dễ dàng các số mũ tương ứng e và d cho phép toán mã hóa và giải mã. Phép giải mã thiết lập lại bản tin rõ nếu như:

$$x = y^{d} = (x^{e})^{d} = x^{ed} \text{ (modulo m)}$$

Điều kiện trên được thỏa mãn nếu:

$$ed = 1 \text{ (modulo } \lambda)$$

Trong ví dụ trên, với giá trị $m = 15$ thì hàm mã hóa sẽ là $y = x^3$, bởi vì $x = x^9$ và $9 = 1$ modulo 4. Để hoàn thiện phương pháp, ở đây cần phải biết xác định λ như thế nào với tất cả modul m. Với một modul m nói chung thì đó là một công việc khá phức tạp, nhưng với $m = p.q$ với p và q là hai số nguyên tố khác biệt nhau, thì chu kỳ λ được xác định bởi biểu thức:

$$\lambda = \text{BSCNN } (p-1, q-1)$$

trong đó: BSCNN là bội số chung nhỏ nhất có thể tính một cách dễ dàng bằng cách chia tích $(p-1).(q-1)$ cho ước số chung lớn nhất theo thuật toán Euclide. Sau đó chọn phương pháp đơn giản để xây dựng cặp khóa có quan hệ với nhau là e và d (để mã hóa và giải mã) xuất phát từ p và q. Phương pháp sẽ giữ được bí mật nếu như các giá trị p và q được giữ bí mật và việc khai thác của kẻ xâm nhập trong trường hợp này là cực kỳ khó khăn.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	1
2	1	4	9	1	10	6	4	4	6	10	1	9	4	1	
3	1	8	12	4	5	6	13	2	9	10	11	3	7	14	
4	1	1	6	1	10	6	1	1	6	10	1	6	1	1	
5	1	2	3	4	5	6	7	8	9	10	11	12	13	14	$\lambda + 1$
6	1	4	9	1	10	6	4	4	6	10	1	9	4	1	
7	1	8	12	4	5	6	13	2	9	10	11	3	7	14	
8	1	1	6	1	10	6	1	1	6	10	1	6	1	1	
9	1	2	3	4	5	6	7	8	9	10	11	12	13	14	$\phi + 1$

Hình 4.11. *Bảng kết quả các hàm lũy thừa và hàm mũ modulo-15*

Nếu như chúng ta bắt đầu bằng việc chọn một số nguyên e là ngẫu nhiên, thì có thể có nghi ngờ về đặc tính song ánh xạ của hàm $y = x^e$. Khi modul đã là một số ngẫu nhiên p thì cần phải chọn một số mũ nguyên tố với p-1. Trong trường hợp đó, việc áp dụng tính chất song ánh xạ được đảm bảo bằng cách chọn một số e ngẫu nhiên với p-1 và q-1. Như vậy, một số nguyên tố không thể chia hết cho p-1 cũng như q-1 được thừa nhận. Điều kiện như vậy đảm bảo rằng, biểu thức ed = 1 (modulo λ) có một lời giải. Có thể bắt đầu bằng cách chọn e hoặc chọn d và từ đó suy ra các tham số khác. Ở đây cần lưu ý rằng, việc chọn tham số d có giá trị rất nhỏ thì kẻ xâm nhập có thể dễ dàng tìm ra tính hệ thống.

Mật mã RSA có thể tóm tắt ở hình 4.12, được bắt đầu bằng thực hiện các phép tính để tạo ra một cặp khóa mã sau đó là các phương pháp mã hóa và giải mã. Pha đầu tiên trong việc tính các khóa là tìm hai số nguyên tố có kích cỡ đủ lớn phù hợp với ứng dụng. Có nhiều phương pháp để chọn số nguyên tố lớn và phương pháp thường sử dụng là sử dụng nhiều trắc nghiệm độc lập (không đi sâu).

Bên gửi biết	Công khai	Bên nhận biết
		Các số nguyên tố: p, q m = p.q
m ←	m ←	
e ←	e ←	Các khóa mã d, e nguyên tố với p-1, q-1 de = 1 (modulo λ) λ = BSCNN(p-1, q-1)
Đoạn tin x $y = x^e$ (modulo m)	đã được mã hóa y →	đã được giải mã $x = y^d$ (modulo-m)

Hình 4.12. *Mô tả tóm tắt quá trình mật mã RSA*

Ví dụ sau đây, cho một bài toán với số liệu cụ thể qua các bước thực hiện mã hóa và giải mã của mật mã RSA với các số liệu đơn giản:

Bên nhận:

1. Chọn hai số nguyên tố, ví dụ:

p = 5

q = 7

2. Tính m = p.q

= 5.7 = 35

3. Tính p - 1 và q - 1

p - 1 = 5 - 1 = 4

q - 1 = 7 - 1 = 6

4. Tính chu kỳ λ

λ = BSCNN (4, 6) = 12

5. Chọn cặp khóa mã e, d

Chọn 85 = ed (modulo λ) = ed(modulo 12) = 1

e = 17

$\Rightarrow$ ed = 85

d = 5

6. Gửi e cho bên gửi tin

Bên gửi:

7. Mã hóa đoạn tin x và gửi cho bên nhận

$y = x^e$ (modulo.m)

Giả thiết: $x = 2$

$y = 2^{17}$ modulo35 = 131.072 (modulo 35) = 32

(giá trị 32 = y đó được gửi cho bên nhận)

Bên nhận:

8. Giải mã đoạn tin đã mã hóa nhận được và giải mã.

$x = y^d$ (modulo m)

$= 32^5$ (modulo.35) = 33.554.452 (modulo 35) = 2

Kết quả nhận được đúng đoạn tin gửi: $x = 2$

4.7.2. Thuật toán Euclide và phép toán tìm USCLN, tìm số nghịch đảo.

Phép toán tìm USCLN:

Trong phân tích trên, khi tìm USCLN có nói đến phương pháp sử dụng thuật toán Euclide. Trong lý thuyết số, có một bài toán thường gặp là tìm ước số chung lớn nhất (USCLN) cho hai số nguyên, tức là tìm một số nguyên lớn nhất mà cả hai số trên có thể chia hết cho nó, ví dụ USCLN của 60 và 84 là 12. Phương pháp đơn giản và hiệu quả nhất, đặc biệt khi các số là số lớn, là dùng thuật toán Euclide.

Ví dụ, tìm USCLN của hai số nguyên 323 và 238. Trong toán học biết rằng, nếu như có một số nguyên a và một số nguyên bé hơn là b đều có một số chia chung (ước số chung) là D thì lúc đó

a - b, a - 2b... cũng đều chia hết cho D (cho đến khi một trong chúng bằng không). áp dụng phương pháp trên, sẽ tìm được một số nhỏ nhất bằng cách dùng a - nb với n là số nguyên thương của phép chia a cho b và a - nb là số dư, nhỏ hơn b. Trong ví dụ trên, nếu n = 1 thì số dư là 323 - 238 = 85. Đến đây chung ta có hai số mới là 238 và 85, mà chúng đều là hai số cho hết cho D. Lặp lại quá trình đó và 238/85 cho hai lần số dư là 68. Như vậy 85 và 68 cũng chia hết cho D. Toàn bộ quá trình được bắt đầu với a = 323 và b = 238 được biểu thị ở hình 4.13a. Quá trình được lặp lại cho đến khi số dư bằng 0 thì số cuối cùng (số dư) đó sẽ là USCLN của hai số a và b cần tìm.

Nếu như hai số a và b từ khi bắt đầu xuất phát không có thừa số chung nào ngoài 1 thì thuật toán tìm được 1 và một trong hai số đó là số nguyên tố.

(a)	323		
(b)	238		
(c)	85	(a-b)	c = 323 - 238 = 85
(d)	68	(b - 2c)	d = 238 - 2.85 = 68
(e)	17 (USCLN)	(c - d)	e = 85 - 68 = 17
(f)	0	(d - 4e)	f = 68 - 4.17 = 0

Tìm được: 17 là USCLN của hai số 323 và 238

Hình 4.13a. ***Mô tả thuật toán Euclide để tìm USCLN của hai số nguyên***

Phép toán nghịch đảo:

Trong đại số modulo m, nghịch đảo của một số a là lời giải của biểu thức ax = 1 (modulo m). Một lời giải luôn luôn tồn tại khi mà a và m đều có 1 cho thừa số chung độc nhất. Có thể biến đổi thuật toán Euclide để tìm x như mô tả ở hình 4.13b sau đây, trong đó ví dụ tính nghịch đảo của 299 modulo 323. Các biểu thức xuất phát của a và b được thực hiện để tìm a - nb như mô tả ở hình 4.13a. Mỗi một vế bên phải của biểu thức nhận thừa số 299. Tiếp tục quá trình cho đến khi tìm được USCLN bằng 1 ở phía bên trái của biểu thức. Dòng cuối cùng chính là sự thiết lập quan hệ tìm được. Nếu chúng cho kết quả là đại lượng âm thì phải cộng thêm modul (323) để có giá trị đúng. Các ví dụ mô tả các phép biến đổi nhanh của thuật toán Euclide trong thực tế. Số các pha thực hiện là gần bằng số các bit của giá trị khởi đầu xuất phát.

Modulo 323

(a)	323	=	0.299	
(b)	299	=	1.299	
(c)	24	=	- 1.299	(a - b)
(d)	11	=	13.299	(b - 12c)
(e)	2	=	- 27.299	(d - 2d)
(f)	1	=	148.299	(d - 5e)

Hình 4.13b. ***Mô tả ứng dụng thuật toán Euclide để thực hiện phép toán nghịch đảo***

4.7.3. Đánh giá độ mật của mật mã RSA

Sau khi phương pháp mật mã RSA được công bố, đã có rất nhiều người xây dựng các bảng hàm lũy thừa (như kiểu ở hình 4.11) để tìm phương pháp thử xâm nhập thuật toán, tìm cách giải mã và phân tích các số nguyên tố bằng các phương pháp khác nhau. Cũng đã có nhiều bài báo vạch ra các điểm yếu của thuật toán RSA. Một trong các phương pháp khai thác đó là, dựa vào bản tin đã mã hóa và khóa công khai đã biết (modul m và số mũ e) để suy luận ra bản tin rõ. Việc suy luận theo phương pháp thông thường thì ít nhất phải thực hiện p hoặc q pha, một khối lượng phép tính rất lớn để tìm thừa số.

Có một phương pháp khai thác khác là thực hiện phép toán y^e lặp lại nhiều lần (vì y và e đã biết do công khai) cho đến khi nhận được một giá trị mới là y. Bởi vì bài toán y^e là một bài toán nghịch đảo và chỉ có một số hữu hạn (M) của các giá trị có thể xuất hiện và lúc đó nó trở về điểm xuất phát. Giá trị cuối cùng đã được mã hóa trong phép mã hóa y đó chính là x, bản tin rõ.

Có thể biểu thị phương pháp khai thác xâm nhập đó rõ hơn bằng một ví dụ bằng số cụ thể sau đây:

Có thể mô tả xâm nhập với modul là 23, như trong hình 4.8, dù rằng nó không thuộc về một modul RSA. Giả thiết rằng, khóa mã hóa được chọn là e = 9 và đoạn tin được mã hóa là y = 3. Phép mã hóa lặp lại nhiều lần sẽ cho kết quả sau:

$$3^9 = 18;\quad 18^9 = 12;\quad 12^9 = 4;\quad 4^9 = 13;\quad 13^9 = 3$$

(modulo 23)

Ở đây chỉ cần 5 pha là đã trở về phép mã hóa ban đầu y = 3. Quá trình xâm nhập đã kết thúc, bởi vì pha cuối cùng của quá trình là $13^9 = 3$, điều đó nói lên rằng 13 là bản tin rõ. Nếu như số chu trình chỉ có một số pha (vừa phải) thì kẻ xâm nhập đã thành công.

Trong ví dụ trên, các bản tin rõ và các phép mã hóa có thể lấy tất cả giá trị giữa 2 và 21. Số 20 giá trị đó có thể nhóm thành bốn nhóm 5 giá trị theo phương thức sau đây:

3,18,12,4,13 5,12,19,10,20 6,16,8,9,2 7,15,14,21,17

Phản bác lại lập luận trên, Rivest đã phân tích độ dài chu trình cho phương pháp lặp đó và chỉ ra rằng, có những điều kiện đơn giản trên các số nguyên tố p và q trở thành phương pháp không thể thực hiện được.

Cho một hàm mã hóa $y = x^e$ và nghịch đảo $x = y^d$, giá trị của d có thể tìm được một cách dễ dàng nếu như phép tính logarit $d = \log_y x$ là có thể thực hiện được. Biết rằng, trong phép toán modulo p người ta có thể khắc phục cách tính ngược đó bằng cách chọn số nguyên tố p đủ lớn và tránh giá trị p - 1 chỉ có các thừa số nhỏ, theo cách chơi cờ của phương pháp Pohlig và Hellman. Với modul là m = p.q, điều kiện như trên vẫn đúng: để tránh điểm yếu của phương pháp bởi một phép tính ngược logarit, giá trị p - 1 và q - 1 cần phải có tối thiểu một thừa số lớn. Giả thiết p' là thừa số lớn của p-1, thì điều kiện thỏa mãn để ngăn ngừa các xâm nhập theo phương pháp lặp lại nhiều lần đó là giá trị p'-1 cũng phải là một số lớn nguyên tố. Với q cũng phải có điều kiện tương tự. Có thể áp dụng những điểm nêu trên với ví dụ: chọn modul p = 23 là số nguyên tố. Trong trường

hợp này, p-1 có thừa số lớn nguyên tố là 11 và 11 - 1 cũng sẽ có thừa số lớn nguyên tố là 5. Đó chính là chu kỳ mà chúng ta đã thu được trong việc ứng dụng phép toán lặp lại nhiều lần của mã hóa với đại số modulo 23. Các chu kỳ lớn hơn cũng có thể được thực hiện như trên. Ví dụ, nếu chọn e = 13, sẽ tìm được hai chu kỳ có độ dài là 10. Từ sau khi có cuộc tranh luận vạch ra điểm yếu của mật mã RSA như trên, người ta đã rất thận trọng trong việc chọn các giá trị p và q để tránh điểm yếu có tính quy luật như đã nêu trên. Bài toán khai thác theo phương pháp lặp lại mã hóa nhiều lần như trên chỉ thích hợp trong một số trường hợp.

Kích cỡ của các số được sử dụng để bảo mật trong hệ thống mật mã RSA được xác định bởi bài toán phân tích thành thừa số. Độ phức tạp của phương pháp để phân tích một số nguyên đã cho thành thừa số cũng giống như độ phức tạp của việc thực hiện phép toán nghịch đảo lôgarit đã được mô tả ở hình 4.6. Tiêu chuẩn ở đây là: chọn kích cỡ m sao cho đảm bảo rằng, bằng thuật toán logarit không thể xâm nhập theo kiểu phép tính $d = \log_y x$ và bằng phép phân tích thành thừa số cũng không thể xâm nhập, ngoại trừ phải thực hiện một số phép toán khổng lồ.

Thuật toán RSA có một ý nghĩa trong ứng dụng thực tế là độ phức tạp của phép phân tích thành thừa số, mà nó cũng không thể xác định bằng cách tìm thuật toán tốt nhất và bằng cách tính toán độ phức tạp của nó. Các dạng bài toán "khó" như thế này đã là đối tượng nghiên cứu của toán học từ lâu. Các phương pháp tốt nhất đã chứng minh và biểu thị độ phức tạp đó dưới dạng:

$$L(x) = \exp \sqrt{[a \ln x . \ln(\ln x)]}$$

trong đó a là một hằng số và x là số cần phân tích thành thừa số. Trong những năm gần đây người ta đã thực hiện các phép toán và đã giảm hằng số a xuống trong khoảng từ 6 đến lớn hơn 1 một ít. Đơn vị của phép toán trong cách đánh giá đó là phép nhân các số có kích cỡ x. Độ phức tạp của phép phân tích thành thừa số cũng có dạng giống như bài toán lôgarit (xem phần trên đã phân tích). Bài toán phân tích thành thừa số cũng đã được nghiên cứu tương đối kỹ từ thời Gaussian. Các nghiên cứu về độ phức tạp đó hầu hết mang tính lý thuyết.

Độ phức tạp của phép phân tích thành thừa số là cơ sở của độ mật của phương pháp mật mã RSA. Có thể nói mật mã RSA đang chiếm vị trí quan trọng trong các dạng mật mã có khóa công khai và chữ ký số (sẽ giới thiệu ở phần sau) cho nên ở đây cần quan tâm đến độ khó của bài toán phân tích thành thừa số.

Lý thuyết về bài toán phân tích thành thừa số cũng rất trừu tượng và có thể xác định độ khó của bài toán bằng nhiều cách tiếp cận khác nhau. Sau đây là sáu phương pháp thường gặp:

- Thuật toán các đường cong elip (của Lenstra)

- Thuật toán nhóm các lớp (của Schnorr và Lenstra)

- Thuật toán sàng tuyến tính (của Schroeppel) - Thuật toán sàng quân phương (của Pomerance)

- Thuật toán sàng theo bảng số dư (của Coppersmith, Odlyzko và Schroeppel)

- Thuật toán các phân số liên tục (của Morrison và Brillhart)

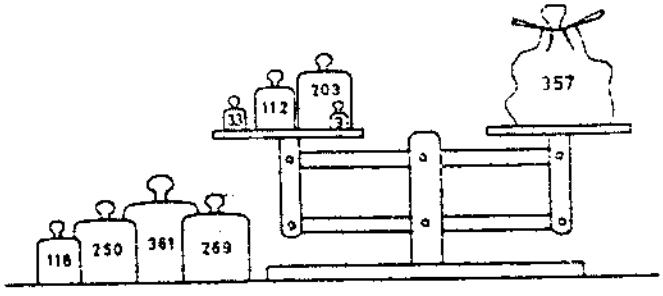
Các hệ thống mật mã có khóa công khai có thể sử dụng để đơn giản hóa việc phân phối các khóa mã mà vẫn đảm bảo độ bí mật tốt hơn. Điều đó có thể thực hiện, ví dụ dùng mật mã RSA để

truyền các khóa phiên làm việc và sau đó được thực hiện ngay với một mật mã đối xứng khá nhanh, ví dụ mật mã DES, để mã hóa và giải mã dữ liệu.

Bởi vì các khóa phiên làm việc chỉ chứa trong một khối của mật mã RSA, cho nên phần còn lại có thể dành cho các thông tin về xác thực, ví dụ như các thông tin về xác thực như nhận dạng, chữ ký số, ngày giờ và các giá trị khởi đầu nếu cần thiết. Thời gian truyền của các thông tin trên cũng rất ngắn, chưa đến một giây. Đó chính là một lợi thế quan trọng của mật mã RSA trong các ứng dụng thực tế.

4.8. THUẬT TOÁN MẬT MÃ "CHIẾC TÚI ĐEO LƯNG" CÓ KHÓA BẤM

Hình 4.14 mô tả bài toán "chiếc túi đeo lưng" (knapsack problem) với một ví dụ đơn giản. Cho tám quả cân với mỗi quả cân có giá trị nguyên lần lượt là 9, 33, 112, 118, 203, 250, 269 và 361 đơn vị. Hãy chọn một số trong số các quả cân đã cho đó để sao cho có thể cân bằng với "một túi đeo lưng" trong cũng chứa đầy các quả cân có tổng trọng lượng của túi là 357 đơn vị.



Hình 4.14. *Bài toán chiếc túi đeo lưng*

Bài toán chiếc túi đeo lưng cũng có thể mô tả dưới một dạng khác. Cho một véctơ tổng tương ứng với tổng trọng lượng của chiếc túi đeo lưng. Hãy tìm các véctơ thành phần trọng trường véctơ cho sao cho các véctơ thành phần đó tạo nên véctơ tổng, tức cộng tọa độ của các véctơ thành phần để cho véctơ tổng.

Bài toán chiếc túi đeo lưng có hai dạng: dạng bài toán dễ có thể thực hiện bằng các phép toán thông thường để tìm lời giải không mấy khó khăn và dạng bài toán khó mà thuật toán để giải nó có thể có độ phức tạp tính toán rất lớn.

Thuật toán mật mã "chiếc túi đeo lưng có khóa bấm" sử dụng một số trường hợp cụ thể của bài toán khó của bài toán chiếc túi đeo lưng và sử dụng một số thủ thuật biến đổi để cho bài toán thêm phức tạp và chỉ có những ai biết được thủ thuật biến đổi đó mới có thể giải được bài toán. Khóa bấm ngụ ý rằng, khóa thì dễ chỉ cần bấm nhưng mở khóa thì cần phải có chìa khoá.

Có thể xem tám quả cân trong ví dụ trên như tám trọng lượng của một véctơ và véctơ đó đặc trưng cho một đoạn tin hay một từ mã 8 bit được truyền. Trong trường hợp trong chiếc túi đeo lưng có 100 phần tử thì điều đó tương ứng với khối dữ liệu hoặc khối đoạn tin x có 100 phần tử, ví dụ đó là một dãy nhị phân có 100 phần tử $x_1, x_2, ..., x_{100}$. Véctơ chiếc túi đeo lưng sẽ là tập các trọng lượng

$w_1, w_2,...,w_{100}$. Việc mã hóa khối đoạn tin được thực hiện bằng cách tính tích vô hướng của các véctơ đó, tức là tính tổng S:

$$S = \sum_{i}^{100} x_i w_i$$

Tổng S của chiếc túi đeo lưng đó là bảng mật mã được gửi cho bên nhận. Để giải bài toán chiếc túi đeo lưng đó cần phải tìm các trọng lượng đã được lựa chọn khi thực hiện bài toán tính tổng S và thiết lập bảng nhị phân x_i của đoạn tin. Cái khó của bài toán ở đây là bằng cách nào đó làm cho bài toán thêm phức tạp mà kẻ xâm nhập không thể giải được với một thời gian ngắn.

Một thủ thuật, được xem như là một phương pháp cấu trúc mật, được mô tả ở hình 4.15. Ở đây sẽ đưa ra hai bài toán của chiếc túi đeo lưng, một bài toán dễ và một bài toán tương đối khó và bài toán được biến đổi từ bài toán dễ.

Dòng A ở trong hình, cho một dãy siêu lũy tiến sau: 1, 3, 5, 11, 23, 46, 136, 263 và tổng chúng là 488 (theo định nghĩa dãy siêu lũy tiến là dãy lũy tiến có giá trị sau lớn hơn tổng các giá trị trước nó).

Với véctơ cho đó thì toàn bộ bài toán của chiếc túi đeo lưng có thể giải bằng một phép toán thông thường. Đó là bản tin dễ. Bây giờ sẽ dùng một thủ thuật để che dấu cấu trúc đó. Các số trên được nhân với a = 203modulo m với m = 491. Kết quả được mô tả ở dòng B của hình 4.15. Đó chính là tập các trọng lượng được sử dụng ở hình 4.14.

Để đảm bảo rằng cấu trúc đã được che dấu, tốt nhất là biểu thị một véctơ mới dưới dạng một dãy lũy tiến như mô tả ở dòng C của hình 4.15. Như vậy bài toán đơn giản đã được biến đổi thành bài toán khác phức tạp hơn, nhưng bài toán phức tạp đó nếu biết các tham số a và m thì có thể thực hiện phép biến đổi ngược lại. Chỉ cần biết nghịch đảo của a = 203modulo491 là đủ. Dựa vào thuật toán Euclide (như đã ví dụ ở hình 4.13) có thể tìm được nghịch đảo đó là 387. Trong hình, bảng mật mã S có giá trị 357. Pha đầu tiên là đưa bài toán chiếc túi đeo lưng khó về bản tin dễ bằng cách nhân với 357 và 387 modulo 491 thì sẽ cho kết quả là 188. Đó là tổng của chiếc túi đeo lưng trong bài toán dễ.

Rõ ràng là trong lời giải của nó không chứa trọng lượng 263 (vì quá lớn) nhưng phải có 136. Số dư 188 - 136 = 32 phải chứa trọng lượng 46 và số dư 52 - 46 = 6, có 5 và 1. Như vậy, các trọng lượng 1, 5, 46, 135 tương ứng lần lượt với 203, 33, 9 và 11 trong véctơ khó và có thể kiểm tra dễ dàng là tổng của chúng có giá trị là 357, giá trị của bảng mật mã. Bài toán đã được giải.

Trong ví dụ trên véctơ "khó" là (9, 33, 112, 118, 203, 250, 269, 361) và đoạn tin nhị phân để lựa chọn các giá trị đó là 11101000. Có thể hình dung đó như một số rất lớn quan hệ mà kẻ xâm nhập khó có thể phát hiện.

Cũng giống như các mật mã có khóa công khai khác, thuật toán mật mã chiếc túi đeo lưng có khóa bấm cũng bao gồm một phương pháp xây dựng các khóa mã công khai và khóa mã bí mật, một phương pháp mã hóa và một phương pháp giải mã. Phía bên nhận xây dựng các khóa mã bắt đầu bằng việc xây dựng một véctơ túi đeo lưng theo kiểu dãy siêu lũy tiến, nhưng là bài toán dễ. Tổng các tọa độ của chúng cho số lớn hơn mà chiếc túi đeo lưng có thể cung cấp.

chọn, theo phương pháp ngẫu nhiên, một modul m lớn hơn số lớn nhất đó và số nhân a nguyên tố cùng với m. Bên nhận có thể sau đó, tính nghịch đảo của số nhân theo thuật toán Euclide. Tiếp theo là, xây dựng véctơ của túi đeo lưng loại khó bằng cách nhân lần lượt mỗi phần tử của véctơ dễ với a modulo m. Véctơ khó của túi đeo lưng được sắp xếp theo luỹ tiến và đó là "khóa công khai" của thuật toán. Véctơ dễ của túi đeo lưng, modul m và số nhân a được phía bên nhận giữ bí mật, đó là "khóa bí mật" của thuật toán.

Tạo khóa mã: a = 203; m = 491

Túi đeo lưng dễ	1	3	5	11	23	46	136	263	A	Tổng: 488
Túi đeo lưng được biến đổi	203	118	33	269	250	9	112	361	B	

Mã hoá:

Khóa công khai	9	33	112	118	203	259	269	361	C	
Bản tin rõ	1	1	1	0	1	0	0	0		
Bản tin đã mã hoá	9	+13	+112	+0	+203	+0	+0	+0	S = 357	

Giải mã:

1/a = 387 m = 491

Bản tin đã mã hóa được biến đổi: 357.387 = 188 (modulo 491)

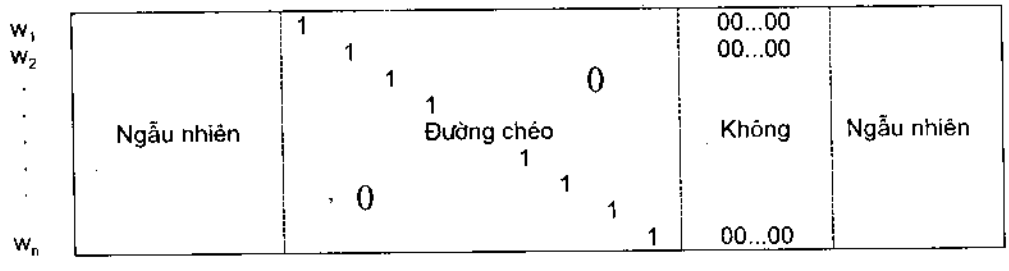
Lời giải của véctơ dễ 188 = 136 + 46 + 5 + 1 (xem A)

Véctơ khó tương ứng 357 = 112 + 9 + 33 + 203 (xem B)

Bản tin rõ 11101000

Hình 4.15. *Mô tả một ví dụ về thuật toán mật mã chiếc túi đeo lưng có khóa bấm*

Việc mã hóa sử dụng một khối bản tin rõ dưới dạng một dãy nhị phân để lựa chọn các tọa độ của túi đeo lưng (tọa độ được chọn tương ứng với các bit 1). Tổng được tính toán có dạng bản tin đã được mã hóa sẽ được gửi cho bên nhận. Bản tin giải mã bên nhận được thực hiện bằng cách nhân tổng của túi đeo lưng đó với nghịch đảo của số nhân để nhận được tổng tương ứng trong bài toán dễ của túi đeo lưng. Đáp số của bài toán dễ cho bản tin rõ, có nghĩa là véctơ nhị phân sẽ xác định các trọng lượng được sử dụng (như đã thực hiện trong dãy của véctơ khó).



Hình 4.16. *Mô tả một phương pháp khác xây dựng các phần tử (loại bài toán dễ) của thuật toán chiếc túi đeo lưng.*

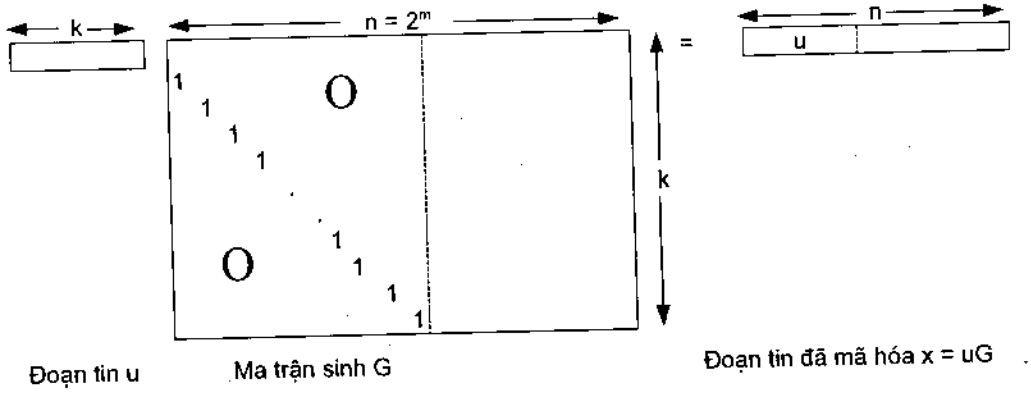
Hình 4.16 mô tả một phương pháp khác để xây dựng các thành phần của chiếc túi đeo lưng dạng bài toán đơn giản. Các thành phần ở đây được biểu thị dưới dạng nhị phân, các bit có trọng số

lớn ở phía bên trái. Có hai khối ngẫu nhiên được cách biệt nhau bởi một ma trận đường chéo và một ma trận không. Khi tiến hành phép cộng một số trong các thành phần đó thì các số dư nhận được xuất phát từ các số ngẫu nhiên nằm ở phía bên phải không tham gia, bởi vì $\log_2 n$ là bằng 0 mà n là số các phần tử của chiếc túi đeo lưng. Trong tổng của chiếc túi đeo lưng, các bit của đường chéo được sử dụng để nhận dạng các phần tử được thêm vào, tức là đoạn tin.

4.9. THUẬT TOÁN MẬT MÃ DỰA TRÊN MÃ SỬA LỖI

Có một phương pháp mật mã có khóa công khai được đề xuất dựa trên mã sửa lỗi. Phương pháp sử dụng một mã sửa lỗi và yêu cầu bên gửi mã hóa các dữ liệu gửi với một số lỗi lớn mà chỉ có bên nhận hợp pháp là biết sửa các lỗi đó.

Trong thực tế truyền tin và truyền dữ liệu có rất nhiều loại mã sửa lỗi. Trong số các mã sửa lỗi đó có một họ mã sửa lỗi quan trọng là các mã khối tuyến tính, hoặc còn gọi là mã Goppa. Dạng tổng quát của mã Goppa được mô tả ở hình 8.17. Đoạn tin u có k bit được mã hóa bởi phép nhân modul-2 với một ma trận $n \times k$ được ký hiệu là ma trận G. Kết quả là một từ mã có độ dài là n, có giá trị lớn hơn k để cho phép sửa lỗi (sửa các lỗi bit trong từ mã). Điều đó cũng tương tự như phép tính của chiếc túi đeo lưng, trong đó các dòng của ma trận là những phần tử của chiếc túi đeo lưng và các dòng thêm vào được lựa chọn bởi những bit của đoạn tin.



Hình 8.17. *Dạng tổng quát của một mã sửa lỗi Goppa.*

Lưu ý rằng, ở đây các phép cộng được thực hiện theo modulo-2 còn các phép toán cộng các số hạng ở bài toán chiếc túi đeo lưng như phân tích các mục trước là phép cộng số học thông thường.

Ma trận G thường được chọn là một ma trận đơn vị $k \times k$, có các bit 1 nằm trên đường chéo, các bit còn lại có thể giá trị 0 và được đặt phía bên trái (hình 4.17). Trong từ mã thì k bit đầu tiên là đoạn tin còn $n - k$ bit được thêm vào được sử dụng để sửa lỗi các bit trong từ mã (đúng theo nguyên lý mã khối tuyến tính sửa lỗi).

Loại mã sửa lỗi như trên có thể sửa được t lỗi và chiều dài n của các từ mã là một hàm mũ của 2; (2^m). Cần phải có m bit để xác định vị trí của một bit lỗi. Như vậy, để chỉ rõ vị trí của t bit lỗi cần tm bit (có thể ít hơn bởi vì một số lỗi có thể đổi chỗ cho nhau), và số dư thừa $n - k$ là bằng tm hoặc bé hơn chút ít. Về vấn đề này, một tác giả *McEliece* đã đưa ra một ví dụ với các giá trị thông

số $m = 10$ và $n = 1024$ và đề nghị sử dụng $t = 50$ lỗi với chiều dài k của đoạn tin là 524 bit hoặc lớn hơn một ít là tốt.

Có một thuật toán khá hiệu quả để sửa các lỗi đó và khôi phục lại k bit của đoạn tin nguyên thuỷ, được xây dựng từ một mã quen thuộc. Vấn đề ở đây là, toàn bộ đa thức bậc t nằm trong trường Galois $GF(2^m)$ tạo ra một mã như vậy. Tuy vậy, để tạo độ khó của thuật toán mà kẻ xâm nhập khó có thể phát hiện được thì ở đây, ma trận **G** cần phải được biến đổi theo phương pháp bí mật và chỉ có người nhận biết phương pháp biến đổi đó.

Tổ hợp các dòng của ma trận **G** theo phương pháp tuyến tính là tương đương với việc nhận ma trận **G** (nằm phía bên trái) với một ma trận **S** có thể nghịch đảo, nhưng phương pháp đó có thể nghịch đảo dễ dàng, bằng cách hồi phục phần đầu của ma trận theo dạng đường chéo như đã biết. Để khắc phục điều đó, làm khó khăn cho việc hồi phục theo phương pháp thông thường, ở đây sẽ hoán vị các cột của ma trận **G** bằng cách nhân **G** bên phải với một ma trận hoán vị $m \times m$, được ký hiệu là ma trận **P**. Như vậy, ma trận sinh thu được sẽ cung cấp cho bên gửi, có nghĩa là khóa công khai của mã sẽ là:

$$\mathbf{G'} = \mathbf{SGP}$$

Để mã hóa đoạn tin u, bên gửi tính véctơ:

$$x = \mathrm{u}\mathbf{G'} + \mathrm{Z}$$

trong đó Z là một véctơ ngẫu nhiên được tạo ra tại chỗ, có chiều dài n và nội dung một số bằng t còn các phần tử khác bằng không. Số hạng Z đó là số các lỗi yêu cầu được đưa thêm vào.

Để loại trừ các lỗi đó, bên nhận tính $x^{\mathbf{P}^{-1}}$ và đó là một từ mã của bộ tạo **SG**. Sau khi giải mã từ mã tương ứng u', bên nhận sẽ biến đổi thành đoạn tin u ban đầu theo biến đổi:

$$\mathrm{u} = \mathrm{u}'.\mathbf{S}^{-1}$$

Độ mật của phương pháp mật mã trên cũng đã được trắc nghiệm. Với các tham số như ví dụ nêu trên độ phức tạp khoảng 10^{19} phép nhân phải thực hiện. Tổng quát, có nhiều phương thức lựa chọn với các dạng mã sửa lỗi khác nhau để dựa vào đó thực hiện các thuật toán mật mã mà kẻ xâm nhập khó có thể phát hiện.

Thuật toán mật mã dựa trên mã sửa lỗi có ưu điểm là dễ dàng thực hiện khi mã hóa và giải mã, mặt khác trong thực tế ứng dụng các mã sửa lỗi được ứng dụng phổ biến trong truyền tin số và truyền dữ liệu. Nhược điểm của mật mã là không có xác thực.

Thuật toán cũng có thể thực hiện theo phương thức là phía bên gửi chọn các lỗi theo phần cứng, cố định các qui tắc và tạo ra một kênh truyền tin mật, có sửa lỗi.

Chương 5
CHỮ KÝ SỐ

5.1. GIỚI THIỆU TỔNG QUAN

Việc xác thực các đoạn tin là để đảm bảo cho bên nhận hai yêu cầu, đó là:

- Đúng đích xác bên gửi;
- Nếu có một sự giả mạo nào đó của đoạn tin sau khi đã gửi đi phải được phát hiện.

Nếu như cả bên gửi cũng như bên nhận không có sự không nhất quán nào về xuất xứ cũng như nội dung của đoạn tin, thì việc trao đổi như vậy là đủ. Cả hai bên đều tin rằng, không có một bên thứ ba nào tham gia vào trong quá trình truyền tin và xuất xứ của đoạn tin cũng không có nghi ngờ gì. Tuy nhiên cũng có nhiều trường hợp xảy ra là giữa bên gửi và bên nhận có sự không nhất quán.

Có thể có những đoạn tin gian lận xuất phát hoặc từ bên gửi hoặc do bên nhận tự tạo ra trong các giao dịch trao đổi và thương mại hoặc thanh toán tiền, chuyển tra ngân phiếu trên mạng. Ví dụ, người nhận có thể bịa ra một đoạn tin như tăng thêm số tiền trong tài khoản của mình và cho đó là do nhận được từ bên gửi, trong quá trình giao dịch thanh toán trên mạng.

Các tranh chấp xảy ra và cũng có nhiều trường hợp người bị lừa đảo khó nhận biết được nếu không có biện pháp ngăn ngừa hữu hiệu. Ngay cả khi dùng mật mã có khóa bí mật mà cả hai bên gửi và nhận đều biết, người gửi có thể tạo thêm một mã xác thực để bên nhận kiểm tra, nhưng về vấn đề đó, bên nhận cũng có thể xây dựng một mã xác thực cho một đoạn tin do bản thân mình xây dựng nên.

Trong nhiều giao dịch thương mại, các tài liệu trên giấy có giá trị cam kết giao hẹn với nhau (như ngân phiếu) thì bên gửi đoạn tin là bên có khả năng làm giả nhiều nhất. Ngay cả khi nếu như việc xác thực là một biện pháp hiệu quả ngăn ngừa các xâm nhập việc trao đổi đoạn tin thì nó cũng không thể ngăn ngừa được việc giả mạo do chính bên nhận tạo ra. Trong các trường hợp đó việc xác thực thường dựa vào chữ ký để xác nhận các điều khoản đã cam kết giao hẹn với nhau trên "giấy trắng mực đen", và đó cũng là cơ sở pháp lý khi có tranh chấp.

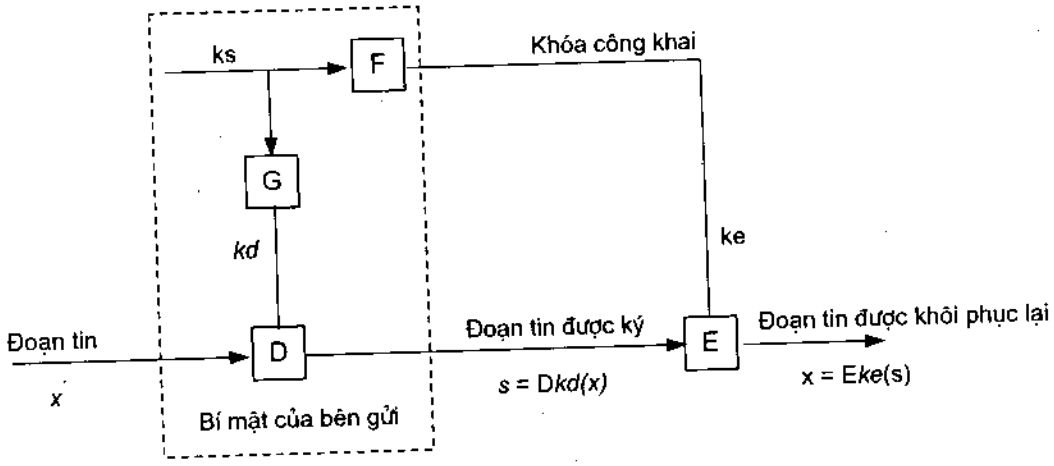
Ngược lại cũng có một số trường hợp về phía bên gửi chính thức lại chối bỏ trách nhiệm của mình vì thấy điều đó không có lợi cho mình. Ví dụ, nếu bên B mang đến tòa án một tài liệu nhận

được và nếu như bên A, bên đã gửi tài liệu đó qua mạng truyền tin số lại chối bỏ trách nhiệm gửi của mình thì tòa án lúc đó cũng rất khó phân xử rạch ròi. Bởi vì cũng có khả năng bên B làm giả đoạn tin và cũng có khả năng bên A có gửi thật nhưng chối bỏ trách nhiệm.

Đó chính là nhược điểm của mã xác thực trong trường hợp có tranh chấp khi sử dụng mật mã đối xứng giữa A và B. Một thuật toán đối xứng như mật mã DES có thể được sử dụng để mã hóa đoạn tin theo cả hai hướng, bởi vì cả hai bên A và B đều biết cả hai khóa bí mật. Các khóa bí mật đó có thể được sử dụng bởi một trong hai bên để xác thực đoạn tin bên kia đã gửi. Tuy vậy cũng cần phải có một cấu trúc nào đó giống như chữ ký tay, được tạo ra bởi bên gửi A và bên nhận B có thể kiểm tra và không thể tạo ra chữ ký đó.

5.2. CÁC CHỮ KÝ SỐ SỬ DỤNG MẬT MÃ CÓ KHÓA CÔNG KHAI

Như đã giới thiệu trong chương 4, mật mã có khóa công khai là không đối xứng. Do tính không đối xứng đó cho nên mật mã có khóa công khai có thể được sử dụng hữu hiệu trong việc xác thực, chữ ký số để giải quyết các tranh chấp giữa bên gửi và bên nhận. Hình 5.1. mô tả sơ đồ khối nguyên lý làm việc của chữ ký số bằng mật mã có khóa công khai, với các ký hiệu trong hình phù hợp với các ký hiệu sử dụng trong hình 4.1. Các hàm D, E, F và G là không thay đổi và các khóa mã *ke* khoá giải mã *kd* đều được tạo ra theo cùng phương pháp.



Hình 5.1. ***Mô tả sơ đồ khối nguyên lý hoạt động của chữ ký số dùng mật mã có khóa công khai***

Ở đây, các khóa mã công khai và bí mật đều do phía gửi tạo ra nhờ một giá trị khởi đầu ngẫu nhiên *ks* (mầm khoá) và khóa mã *ke* được công khai.

Biết rằng, nếu một đoạn tin x được mã hóa theo hàm $y = Eke(x)$ thì việc giải mã để khôi phục lại đoạn tin được thực hiện theo hàm giải mã $x = Dkd(y)$. Nếu ứng dụng các hàm trên theo trật tự ngược lại thì sẽ tạo ra một đại lượng $s = Dkd(x)$ và đó chính là tạo thành chữ ký. Bên nhận s có thể nhận được đoạn tin rõ nhờ có hàm mã hóa và để tin tưởng, tính $x = Eke(s)$. Điều đó rất dễ kiểm tra với thuật toán mật mã RSA, bởi vì bậc tính toán các luỹ thừa là cùng một số cho nên kết quả không thay đổi. Như vậy có nghĩa là:

$$(x^e)^d = x \text{ tức có nghĩa là } (x^d)^e = x$$

Hai hàm nghịch đảo luôn đổi chỗ cho nhau, chúng là các hàm *song ánh xạ*. Để cho hàm mã hóa là hàm tương hỗ duy nhất thì nó không có sự mở rộng dữ liệu trong quá trình mã hoá.

Ở các chữ ký bằng tay trên các văn bản giấy tờ thì giá trị *s* đó người nhận có thể đọc trực tiếp. Cũng tương tự như các chữ ký bằng tay, các chữ ký số có thể thực hiện bằng nhiều phương pháp khác nhau và các chữ ký đó không làm biến đổi đoạn tin mà nó chỉ là một giá trị phụ được bố trí thêm đi kèm với đoạn tin, ví dụ sử dụng một khối 512 bit với thuật toán mật mã RSA. Trong trường hợp bản tin dài thì chữ ký *s* thường được kèm với từng khối tin một.

Để kiểm tra chữ ký, bên nhận dùng hàm mã hóa với khóa công khai của bên gửi. Nếu như kết quả là một đoạn tin rõ đúng thì cơ cấu chữ ký đã hoạt động tốt và chữ ký được xem là có hiệu lực. Bằng chứng của sự hiệu lực đó là chỉ bên sở hữu khóa giải mã *kd* mới có thể tạo ra chữ ký *s* để bên nhận mã hóa bằng khóa mã hoá *ke* công khai để tạo ra đoạn tin. Xung quanh vấn đề này cũng có một số giả thiết.

Trước tiên, giả thiết rằng khóa *kd* được hoàn toàn giữ bí mật bởi bên sở hữu khóa công khai *ke*. Sự đồng nhất, "chủ sở hữu" khóa *ke* được cung cấp bởi một trung tâm lưu trữ các khóa và giá trị của nó, cũng đồng nhất được chủ sở hữu, tất cả đều đúng đắn và xác thực. Nhưng nếu như có một kẻ xâm nhập nào đó, muốn đánh lừa bên nhận một khóa công khai giả do chúng tạo ra (mà chúng có khóa bí mật tương ứng), thì lúc đó chúng có thể tạo ra các tài liệu với một chữ ký như là có hiệu lực. Vì vậy việc xác thực khoá công khai là một công việc quan trọng, tất cả nằm trong khâu mã hoá.

Một giả thiết khác là, toàn bộ biến đổi đoạn tin dưới dạng đã được ký *s* tạo ra sau khâu mã hóa một kết quả đưa ra không thể chấp nhận, tức bên nhận cho rằng nó khác với đoạn tin đúng. Có một dư thừa nào đó trong đoạn tin rõ và xác suất nhận đoạn tin ngẫu nhiên rất bé. Điều này trong thực tế có thể kiểm tra được khi sử dụng chữ ký dưới dạng một khối thêm vào đoạn tin rõ.

Với các giả thiết nêu trên, đến đây có thể khẳng định rằng, chỉ có bên sở hữu khóa bí mật mới tạo ra được một chữ ký có hiệu lực và việc biết khóa công khai của bên gửi đó cũng để kiểm tra một đoạn tin đã được ký bởi người đó.

Một chữ ký cần phải phụ thuộc vào tất cả các bit của đoạn tin với mục đích là giữ sự bền vững của đoạn tin. Điều quan trọng ở đây là, không một ai có thể thay đổi nội dung của bản tin rõ trong lúc phải giữ nguyên chữ ký. Chữ ký khác với phương pháp xác thực đối xứng là phía bên gửi nắm giữ nhiều thông tin hơn phía bên nhận (tức bên gửi giữ khóa bí mật). Đó chính là phương tiện để tạo ra chữ ký và bên nhận chỉ có kiểm tra chữ ký đó đúng hay không bằng khóa công khai.

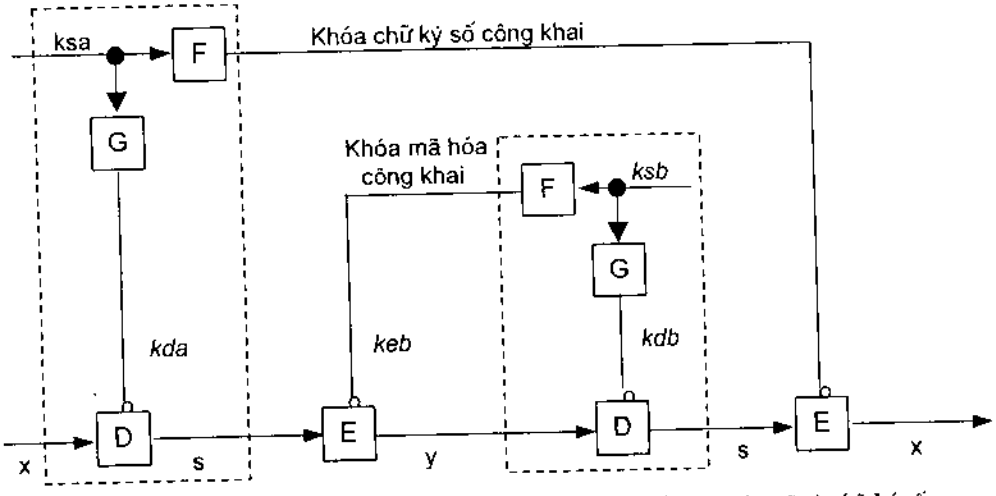
Chữ ký số cũng không phải là một biện pháp toàn năng để chống giả mạo và có thể có tranh chấp cho nên trong nhiều trường hợp cần có cơ chế trọng tài (sẽ giới thiệu ở phần sau). Chữ ký số có thể phối kết hợp với mật mã sử dụng.

5.3. KẾT HỢP CHỮ KÝ SỐ VÀ MẬT MÃ

Bằng cách kết hợp hai biến đổi: chữ ký số dùng khóa công khai và mật mã có khóa công khai, có thể vừa bảo vệ đoạn tin vừa chống lại các sự giả mạo. Việc sử dụng mật mã để đảm bảo tính bí mật của đoạn tin còn việc sử dụng chữ ký số để phát giác tất cả các gian lận. Hai biến đổi được xếp chồng lẫn nhau. Hình 5.2. mô tả sơ đồ khối nguyên lý hoạt động của phương pháp (các ký hiệu trong hình vẫn sử dụng như trước).

Đoạn tin từ A gửi đến B trước tiên thực hiện thuật toán chữ ký sau đó thực hiện tiếp thuật toán mật mã. Các khóa công khai và bí mật của cả hai bên đều được sử dụng nhưng có phân định rõ trách nhiệm: khóa *kda* và khóa *kea* thuộc bên gửi A còn khóa *kdb* và *keb* thuộc bên nhận B. Để ký, bên A sử dụng khóa bí mật *kda*. Tiếp theo, để mã hoá, bên A sử dụng khóa công khai *keb* do bên nhận B cung cấp. Khi nhận, bên B thực hiện các biến đổi ngược lại: trước tiên giải mã với khóa *kdb*, sau đó kiểm tra chữ ký với khóa *kea*.

Trình tự thực hiện các phép toán như trên là rất thuận tiện bởi hai lý do sau đây. Một là thuận tiện trong việc tạo chữ ký, trong đoạn tin có cả chứng cớ chữ ký và hai là các đoạn tin đã được ký, được giải mã có thể nhớ lưu giữ làm chứng cứ về sau hoặc đương sự giữ lại.



Hình 5.2. ***Mô tả sơ đồ khối một phương pháp kết hợp mật mã và chữ ký số***

Ở hình 5.2 xử lý chữ ký với trường hợp đoạn tin được gửi từ A đến B. Trong trường hợp đoạn tin được gửi theo hướng ngược lại thì cũng vẫn sử dụng các khóa đó, nhưng bên B ký với khóa *kdb* và mã với khóa *kea*, còn bên nhận A giải mã với khóa *kda* và kiểm tra chất lượng với khóa *keb*.

5.4. CHỮ KÝ SỐ SỬ DỤNG THUẬT TOÁN RSA

Ở mật mã RSA thì khóa công khai được xây dựng từ phép toán modulo m và số mũ mã hóa *e*. Các hàm mã hóa và hàm giải mã được biểu thị dưới dạng rất đơn giản:

$$y = x^e \qquad y^d = (x^e)^d = x \text{ (modulo m)}$$

Chữ ký số và phép biến đổi nghịch đảo, được sử dụng để kiểm tra, có dạng như sau:

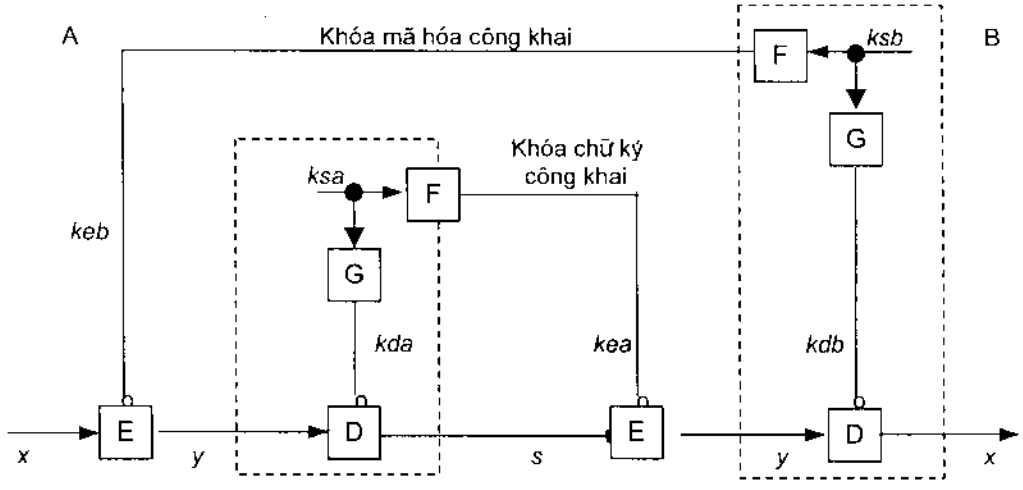
$$s = x^d \qquad s^e = (x^d)^e = x \text{ (modulo m)}$$

Thuật toán RSA tạo các chữ ký số rất dễ, bởi vì nó dựa trên một biến đổi song ánh xạ của tập 1, 2, 3... m-1. Thuật toán chiếc túi đeo lưng kéo dài thêm đoạn tin, do vậy nó không thích hợp với chữ ký số.

Việc thực hiện chữ ký số trước và sau đó mã hóa như mô tả ở hình 5.2 sẽ có trường hợp không thuận tiện, nếu như đoạn tin sau đó được phân thành các khối nhỏ.

Hình 5.3 mô tả một phương pháp gần giống như phương pháp ở hình 5.2 nhưng có khác là

thực hiện chữ ký số sau khi mã hoá. Với phương pháp này bản tin có thể phân thành các khối tuỳ ý. Phương pháp trở lại thuật toán RSA có chữ ký số.



Hình 5.3. *Mô tả sơ đồ khối một phương pháp thứ hai kết hợp mật mã và chữ ký số.*

5.5. CHỮ KÝ SỐ TÁCH BIỆT ĐOẠN TIN GỬI

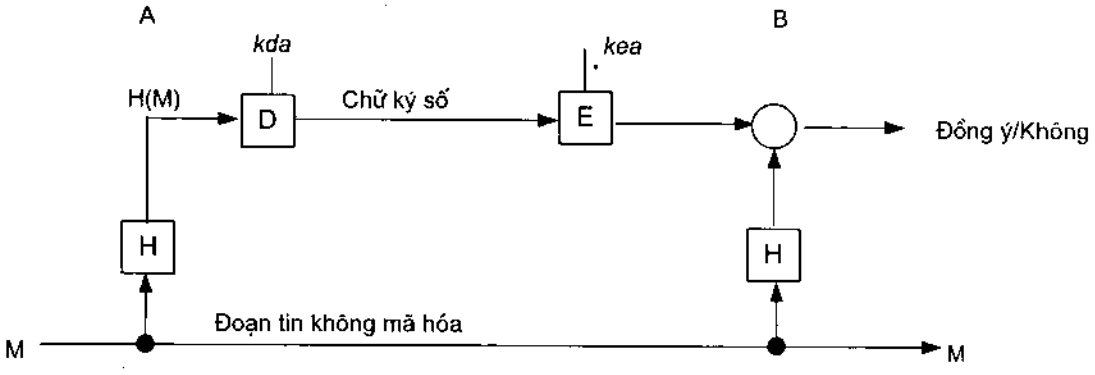
Trong các mục trên giới thiệu chữ ký số gắn liền với đoạn tin gửi. Trong một số trường hợp có yêu cầu chữ ký số và đoạn tin gửi tách biệt nhau. Hình 5.4 mô tả một phương pháp chữ ký số tách biệt đoạn tin gửi. Ở đây, bên gửi A gửi đoạn tin M (không mã) và chữ ký sử dụng khóa bí mật *kda*. Bởi vì đoạn tin ở đây không được mã hóa cho nên việc nhận dạng bên gửi không quan trọng. Nó thuộc dạng đoạn tin được phổ biến rộng hoặc tài liệu được công bố công khai. Đường phía trên ở hình 5.4 là đường chữ ký số. Một hàm một chiều, H (hàm băm Hast), được áp dụng cho toàn bộ đoạn tin. Đoạn tin ở bên gửi, sau khi qua hàm một chiều H được mã hóa với khóa *kda* và chuyển cho bên nhận B. Đó chính là chữ ký số do A gửi cho B.

Bên B sau khi nhận được chữ ký số, giải mã với khóa *kea* để có hàm một chiều H(M). Mặt khác B cũng nhận được đoạn tin rõ M, qua hàm một chiều H để có H(M). So sánh hai H(M) đó, nếu trùng nhau tức chữ ký số có hiệu lực, việc xác thực đoạn tin đã được thiết lập. Hàm một chiều ở đây được công khai và là một hàm chuẩn hoá. Giá trị của hàm một chiều đó không được vượt quá một khối của mật mã RSA sử dụng và cũng không nên nhỏ quá, vì như vậy kẻ xâm nhập sẽ dễ khám phá.

Giá trị hàm một chiều có thể sử dụng, ví dụ 64 bit, như vậy chữ ký sẽ được thiết lập với một giá trị ngẫu nhiên có xác suất là 2^{64}, một xác suất khá nhỏ.

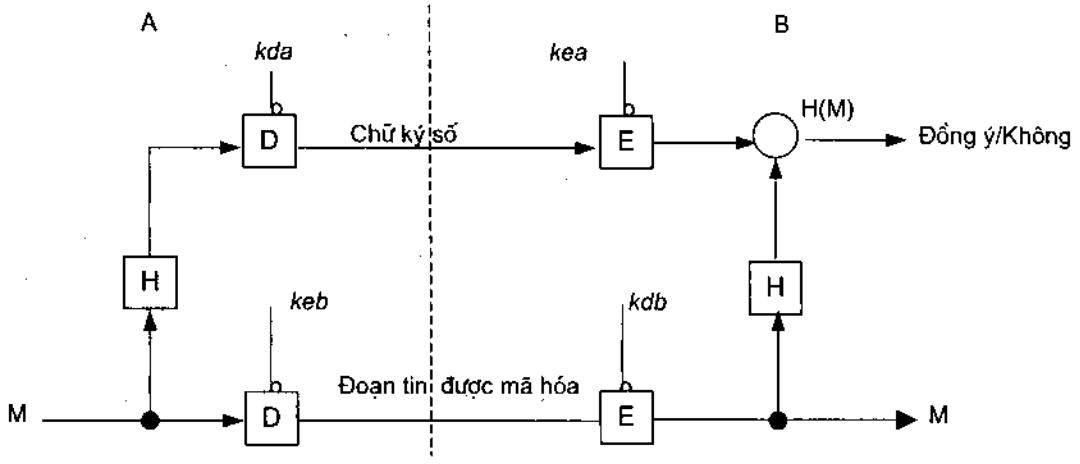
Trong một số trường hợp không cần thiết phải mã hóa chữ ký một cách tuyệt đối. Hình 5.5. mô tả một ví dụ đoạn tin được ký và được mã hóa bởi một hệ thống mật mã có khóa công khai. Phương pháp này chỉ dành riêng cho một bên gửi là A và một bên nhận là B, và nó phụ thuộc vào các khóa công khai của nhau. Chữ ký số được truyền rõ và đoạn tin được mã hóa bởi khóa công khai *keb* của bên nhận. Sau khi giải mã, việc kiểm tra chữ ký được thực hiện như trước. Với phương pháp này thì kẻ xâm nhập có khả năng làm thay đổi các đoạn tin và các chữ ký nhưng không thể phát hiện được nội dung, bởi vì xác suất rơi trên một chữ ký gắn với đoạn tin rất bé. Nếu như bên

nhận tìm được các chữ ký và các đoạn tin được tính toán từ các giá trị H(M) như nhau, với khoá công khai kia thì có nghĩa là đoạn tin nhận được có chữ ký đúng, đoạn tin chính xác đã được đến do chính A gửi.



Hình 5.4. ***Mô tả một phương pháp chữ ký số và đoạn tin (không mã hoá) tách biệt nhau.***

Ưu điểm của chữ ký số dựa trên hàm một chiều là ở chỗ, thuật toán mật mã RSA làm việc chỉ một lần để ký một đoạn tin có chiều dài bất kỳ.



Hình 5.5. ***Mô tả một phương pháp chữ ký số và đoạn tin có mã hóa tách biệt nhau***

5.6. CHỮ KÝ SỐ DÙNG MẬT MÃ ĐỐI XỨNG

5.6.1. Phương pháp chữ ký số dùng mật mã đối xứng của Lamport-Diffie

Có nhiều phương pháp chữ ký số sử dụng mật mã đối xứng được đề xuất. Lý do chủ yếu là có sự nghi ngờ về độ mật của một số thuật toán sử dụng mật mã có khóa công khai. Trong thực tế, loại mật mã nào cũng có một số ưu điểm và nhược điểm, tuỳ thuộc vào ứng dụng cụ thể để chọn lựa thích hợp.

Sau đây sẽ giới thiệu các nguyên lý của chữ ký số dựa trên thuật toán DES. Phương pháp tốt nhất để mô tả mô hình là làm thế nào để ký đoạn tin một bit. Với mỗi một giá trị 0 và 1 của đoạn tin, các giá trị ngẫu nhiên x và k được lựa chọn, và sau đó theo thuật toán DES tính các giá trị tương ứng $y = Ek(x)$. Như vậy sẽ dẫn đến các quan hệ với sáu giá trị (y_0, k_0, x_0 và y_1, k_1, x_1) sau đây:

$$y_0 = Ek_0(x_0)$$

$$y_1 = Ek_1(x_1)$$

Từ các biểu thức trên, có thể xem như chữ ký có khóa công khai với các giá trị x_0, y_0, x_1 và y_1. Các giá trị k_0 và k_1 được xem là khóa bí mật.

Khi đoạn tin được gửi, nếu giá trị đoạn tin là 0 thì chữ ký là k_0 và nếu giá trị đoạn tin là 1 thì chữ ký là k_1. Việc kiểm tra của bên nhận được thực hiện bằng cách mã hoá, với khóa k_0 hoặc k_1, đoạn tin tương ứng x_0 hoặc x_1, để nhận được giá trị tương ứng của y. Trong trường hợp này, nếu như có một xâm nhập nào đó thì cũng rất khó nếu không nói là không thể tìm được k xuất phát từ x và y. Chữ ký là giá trị hoặc k_0 hoặc k_1 chứng nhận rằng đoạn tin đến đúng từ người gửi mà người đó đã tính và đưa ra công khai các giá trị x và y trước đó. Điều mô tả trên có tính nguyên lý để ký đoạn tin một bit, khóa công khai là 256 bit và chữ ký số là 56 bit. Giả thiết trị rằng: M là đoạn tin, H là hàm một chiều và ảnh H(M) của đoạn tin (64 bit) bởi hàm một chiều cần phải được ký: chữ ký số lúc đó có 3584 bit và khóa công khai là 16384 bit.

Sau khi sử dụng chữ ký, khóa công khai tương ứng không thể dùng lại. Một nửa giá trị khóa bí mật k cũng bị lộ, các giá trị khóa k còn lại do đó cũng không thể sử dụng được. Vì vậy, với mỗi đoạn tin yêu cầu sử dụng một tập các thông số công khai hoàn toàn mới. Trong trường hợp gửi liên tục các đoạn tin được ký thì cần công bố một số lượng lớn các dữ liệu để sử dụng dần cho các chữ ký. Các dữ liệu mới đó cũng sẽ được bảo vệ theo phương pháp chữ ký từ các số liệu cũ.

5.6.2. Phương pháp chữ ký số dùng mật mã đối xứng của Rabin

Phương pháp chữ ký số của Rabin yêu cầu có sự can thiệp tích cực của bên gửi đoạn tin vào hiệu lực của chữ ký như mô tả ở hình 5.6. Bên gửi chọn ngẫu nhiên một số cặp khóa mã DES, ví dụ 36 khóa $k_1, k_2,...,k_{36}$. Sau khi đã áp dụng hàm một chiều cho đoạn tin và cũng đã nhận được một giá trị H, nó sẽ tạo ra các chữ ký với 36 giá trị:

$$S_i = Ek_i(H)$$

với $i = 1, 2,..., 36$

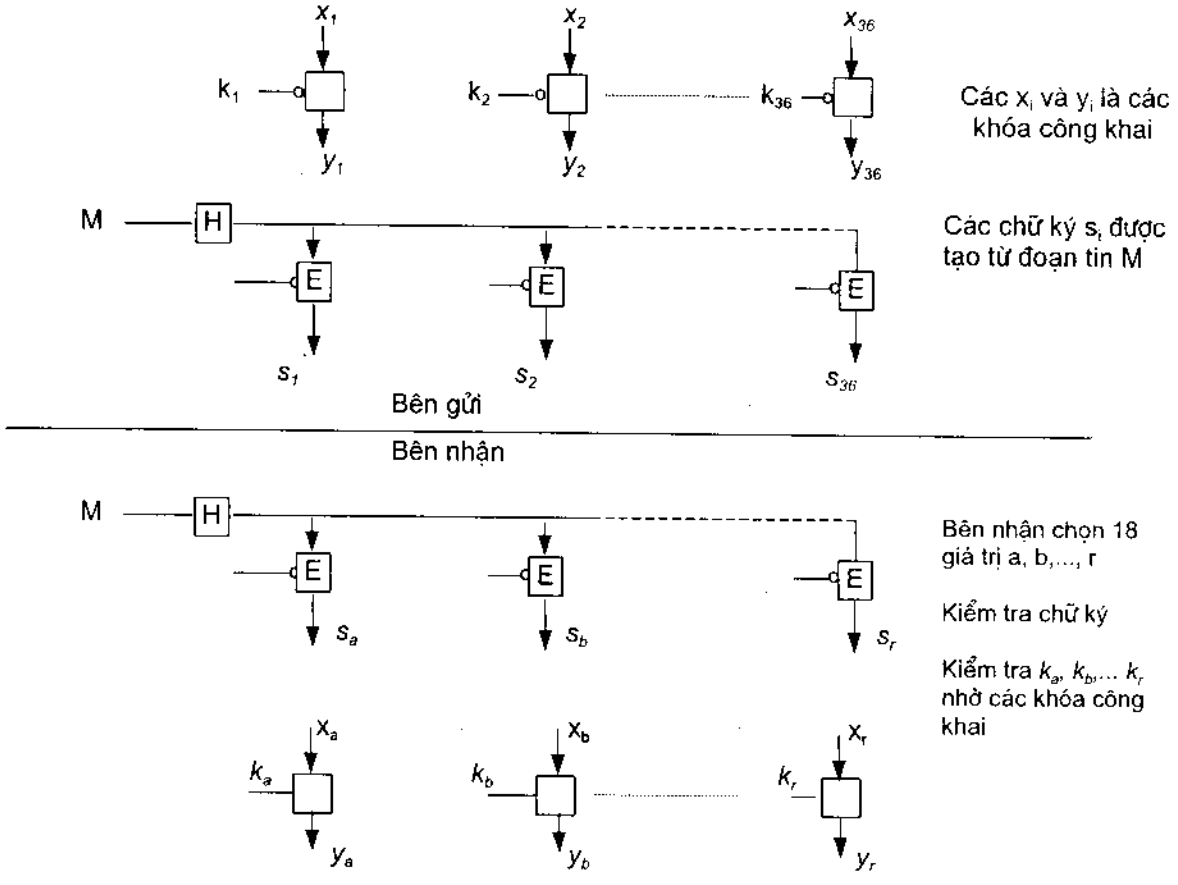
Khi bên nhận muốn kiểm tra chữ ký thì chọn 18 giá trị i bên ký tiết lộ tập khóa con đó. Bên nhận sử dụng tập khóa con đó để thử lại các biểu thức cho trên. Nếu bên gửi tiết lộ tất cả các khóa thì bên nhận có thể làm giả các đoạn tin tương ứng với chúng. Trong trường hợp có tranh chấp thì bên nhận trình bày đoạn tin và chữ ký đã được nhận, trong lúc đó bên gửi tiết lộ cho trọng tài tất cả các khóa và tài liệu liên quan.

Luật áp dụng để phân xử là, nếu như 19 (hoặc lớn hơn) các khóa trong chữ ký là đúng, người nhận thắng và chữ ký là có giá trị. Nhưng nếu như 18 (hoặc ít hơn) trong các khóa là đúng, thì xem xét lỗi là phía bên gửi.

Trong trường hợp trên, nếu như bên gửi có ý đồ gian trá thì rất dễ, bởi vì xác suất chọn 18 phần tử trong số 36 là rất lớn. Có thể khắc phục điều đó bằng cách sử dụng toàn bộ cặp khoá. Chữ ký 36 thành phần đó có $64 \times 36 = 2304$ bit và dữ liệu cung cấp cho chữ ký để kiểm tra là 1008 bit, một xác suất khá nhỏ. Tuy vậy việc kiểm tra 18 khóa cung cấp theo yêu cầu của bên gửi cần phải

dựa trên các dữ liệu được xây dựng từ khóa công khai của bên gửi và việc xác thực được thiết lập trước. Các giá trị x và y là cần thiết cho mỗi một trong 36 khoá, thường là 4608 bit và tiện nhất là sử dụng phương pháp Lamport và Diffie. Khóa công khai cần luôn thay đổi để tránh nguy hiểm do chữ ký xây dựng trên thuật toán đối xứng.

Phương pháp chữ ký số dùng thuật toán đối xứng hơi rườm rà và tốn nhiều công sức để tạo ra mối liên hệ với các khóa công khai.



Hình 5.6. ***Mô tả phương pháp chữ ký số dùng mật mã đối xứng của Rabin.***

5.7. CHỮ KÝ SỐ VÀ XÁC THỰC SỬ DỤNG HÀM MỘT CHIỀU

Như đã biết, hàm một chiều H giảm đoạn tin có chiều dài bất kỳ xuống một chiều dài chuẩn thích hợp và ở cuối đoạn tin có chữ ký, nếu sử dụng hàm H cho chữ ký số: Hàm H cũng có thể sử dụng cho việc xác thực (chi tiết về hàm băm H xem mục 7.10). Giả thiết rằng đoạn tin M được truyền từ A đến B trên đường truyền tin khả dĩ có xâm nhập tích cực. Có thể sử dụng hàm H(M) để kiểm tra M được truyền trên đường truyền không tin cậy đó.

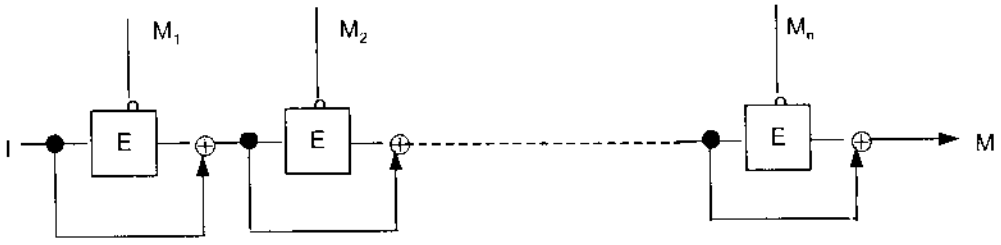
Tính chất quan trọng của hàm H là hàm một chiều. Hàm đó là công khai và không sử dụng khóa mã. Với một đoạn tin đã cho M thì đối phương có thể tính H(M), tuy vậy tính bảo mật sẽ bị nguy hại nếu như đối phương tìm được một đoạn tin M' khác, sao cho H(M) = H(M'). Bởi vì tập hợp các giá trị của H là rất nhỏ so với tập hợp các đoạn tin, cho nên có một số rất lớn các lời giải. Tuy vậy rất khó có phương pháp để tính toán chúng.

Mặt khác, phép mã hóa E*k*(*x*) cũng là hàm một chiều của *k*, nhưng không phải của *x*. Có thể tận dụng hàm mã hóa E*k*(*x*) đó để xây dựng chữ ký số và xác thực cho một đoạn tin có chiều dài bất kỳ theo một số đơn vị có dạng sau:

$$y = Ek(x) \oplus x$$

Hình 5.8 mô tả phương pháp thực hiện thuật toán trên. Trong trường hợp này, nếu cho giá trị *y* thì biểu thức không thể giải để cùng tìm được giá trị *k* và *x*. Trong hàm trên thì *k* là khóa mã và *y* là ảnh của *k* do hàm một chiều tạo nên. Nếu như E là một thuật toán mã hóa tốt thì *y* là hàm một chiều của *x* với tất cả *k* với lí do là *x* xuất hiện hai lần trong biểu thức.

Trong mô hình ở hình 5.8 thì đoạn tin M được phân chia thành các khối có độ dài 56 bit và I là giá trị khởi đầu và giá trị đó có thể là một phần của đoạn tin. Cũng có thể chọn I là một giá trị ngẫu nhiên đưa vào trước chữ ký.



Hình 5.7. *Ví dụ hàm một chiều cho chữ ký số.*

Trong trường hợp như mô tả ở hình 5.8, kẻ xâm nhập có thể tấn công vào hàm hỗn hợp H(M, I) bằng cách thực hiện quay ngược các mũi tên trên sơ đồ về phía trước. Điều đó có nghĩa là các phép tính ngược được thực hiện. Tuy vậy để thực hiện các phép tính ngược có liên quan đến hàm H(M, I) và I thì phải thực hiện khoảng 2^{32} phép tính trên các nửa đoạn tin. Cũng chính vì lý do đó mà hàm một chiều trong trường hợp này sử dụng hai biến số.

Có một phương pháp chuẩn hóa cho việc tính toán hàm một chiều và nhóm các kết quả theo khối để sử dụng chữ ký số theo mã có khoá công khai RSA. Khối RSA lúc đó chứa một độ dư thừa nào đó, ví dụ một tập hợp các số 0.

5.8. CHỮ KÝ SỐ VÀ ĐỒNG NHẤT SỬ DỤNG CÁC THỂ THỨC CỦA FIAT - SHAMIR

5.8.1. Cơ sở toán học các thể thức của Fiat - Shamir

Có một dạng chữ ký số theo các thể thức của Fiat - Shamir, nó không nhằm mục đích phục vụ chữ ký số nhưng nó cho phép một thực thể chứng tỏ sự đồng nhất của mình cho một thực thể khác một cách đích xác thông qua các thể thức của chúng. Tất cả các phép tính trong các thể thức của Fiat - Shamir đều được thực hiện theo đại số hữu hạn (modulo m với m là tích của hai số nguyên tố *p* và *q*). Nó cũng tương tự phép đại số của thuật toán RSA. Các thể thức đó dựa vào độ khó của phép toán tìm căn bậc hai khi mà việc phân tích thành thừa số của m là chưa biết. Giả thiết rằng: *m*, *p* và *q* là các số lớn. Nếu lấy tất cả các số nguyên giữa 0 và *m* - 1 và tính các bình phương của chúng modulo m thì chúng ta chỉ nhận được khoảng *m*/4 các kết quả khác nhau. Phần lớn các

số đó không có căn bậc hai nhưng một số trong đó có căn bậc hai sẽ được sử dụng sau này và các số đó được gọi là "*các số dư cầu phương*".

Trong hầu hết các trường hợp, các số dư cầu phương có bốn căn bậc hai khác nhau và $m - x$ được xem như là một số đối của x. Bốn căn bậc hai của một số dư cầu phương cũng sẽ xuất hiện dưới dạng hai cặp số của x, $m - x$.

Có thể nói rằng, việc tìm căn bậc hai theo phương pháp đại số modulo m là một bài toán khó. Giả thiết là luôn luôn có thể tìm được tối thiểu một căn bậc hai của x. Cho một số ngẫu nhiên t; tính $x = t^2$ sau đó tìm căn bậc hai s sao cho $s^2 = x$. Như vậy chúng ta có quan hệ $s^2 - t^2 = 0$ và điều đó có nghĩa là $(s + t)(s - t)$ là một bội số của m. Số s được chọn một cách ngẫu nhiên. Việc tính căn bậc hai theo đại số modulo m (trong đó $m = pq$ và các số lớn được chọn đúng đắn) cũng là một bài toán khó giống như bài toán khó phân tích m thành thừa số.

Ngược lại, nếu như việc phân tích thành thừa số của m là đã biết bài toán căn bậc hai cũng tương đối dễ. Mỗi một lần bài toán giải modulo p sau đó modulo q, theo lý thuyết các *số dư Trung hoa* có thể cho phép tính được một căn bậc hai của x modulo m. Bài toán modulo p và q sẽ đơn giản nếu như p và q có số dư là 3 trong phép chia cho 4.

Tóm lại, thể thức của Fiat - Shamir là dựa trên độ khó của phép toán căn bậc hai modulo m, nếu như việc phân tích m thành thừa số là chưa biết và nó cũng sẽ là bài toán dễ nếu việc phân tích thành thừa số là đã biết. Cái khó ở đây là trong phép tính căn bậc hai với các số dư cầu phương.

5.8.2. Mô hình đồng nhất cơ sở

Một mô hình đồng nhất cho hai đối tác A và B; phía bên A muốn chứng tỏ sự đồng nhất của mình cho phía bên B. Chứng cớ của sự đồng nhất đó có thể, ví dụ dùng một từ mật khẩu, nhưng điều đó có thể cho phép sau đó bên B làm tiết lộ từ mật khẩu của A. Trong một mô hình đồng nhất tin cậy, người ta mong muốn rằng, A có thể cung cấp cho B đồng nhất của mình nhưng không biểu lộ thông tin cho phép sau đó bên B làm tiết lộ cho thực thể khác. Có thể thực hiện điều đó bằng cách sử dụng thuật toán mật mã đối xứng như đã giới thiệu ở chương 2, nhưng sau đây sẽ giới thiệu một mô hình dạng khác ứng dụng các thể thức của Fiat - Shamir.

Thể thức gồm có sự lặp lại một trao đổi đơn giản của ba đoạn tin. Có thể biểu thị thực thể A như một thẻ có chip điện tử chứa một giá trị mật cho phép A được đồng nhất với tất cả các thực thể B khác có thể (dạng một thiết bị đầu cuối) để thực hiện thể thức. Gọi là dựa trên sự đồng nhất là bởi vì ở đây A (thẻ) lưu giữ một thông số mật có liên quan đến sự đồng nhất của A mà cái đó lại là công khai. Gọi tham số có liên quan đến sự đồng nhất đó là I: nó có thể bắt nguồn từ tên của người mang thẻ, số thẻ, số tài khoản hoặc một hỗn hợp của các số đó. Để giảm thông tin đến một kích cỡ có thể sử dụng được, người ta sử dụng một *hàm băm* (hàm hash) hoặc hàm một chiều f cho I và có $f(I)$ cho các phép tính đó. Thẻ có chứa I và có thể phân phối cho tất cả đối tác, nhưng phải sử dụng hàm $f(I)$ cho các phép tính nó. Trong thực tế, cần một số là số dư quân phương để sử dụng hàm $f(I, c)$ trong đó c là một số nhỏ có quan hệ với I và được xác định bởi các phép thử khá đủ cho đến ki $v = f(I, c)$ là số dư quân phương. Thẻ sử dụng c cho truy nhập tự do của một thực tể nào đó cần kiểm tra đồng nhất I nhờ có thể thức đó.

Một trung tâm chịu trách nhiệm, ví dụ nơi phát hành thẻ, biết các giá trị p và q: ở đó có thể dễ dàng tính được căn bậc hai của v, có nghĩa là một lời giải u của biểu thức $u^2 = v$ modulo m. Giá trị u đó là giá trị mật mà thẻ lưu giữ, và chỉ có trung tâm phân phối thẻ biết.

Giả thiết rằng, thực thể B cần kiểm tra đồng nhất của thẻ là B biết m và nhận được thẻ I và giá trị c. Từ đó có thể tính được v, bởi vì hàm băm f là công khai. Bài toán là một sự trao đổi qua lại giữa A và B để cho phép lần thứ nhất làm chứng cứ cho lần thứ hai mà việc biết căn bậc hai u cũng không làm tiết lộ giá trị thực của nó.

Việc trao đổi hội thoại qua lại được lặp nhiều lần và các phép tính được thực hiện theo modulo m như sau:

1. Bên A tạo một số ngẫu nhiên r và gửi cho bên B các số $x = r^2$ và $y = v/x$. Bên B kiểm tra lại một cách dễ dàng là $xy = v$.

2. Bên B gửi một bit ngẫu nhiên e cho A.

3. Nếu $e = 0$, A gửi r cho B; nếu $e = 1$, A gửi $s = u/r$ cho B.

Bên B (ví dụ thiết bị đầu cuối) có thể dễ dàng kiểm tra rằng, hoặc $r^2 = x$, hoặc $s^2 = y$ theo sự lựa chọn tham số e trong đó.

Thẻ A không nhất thiết phải gửi cùng một lúc các giá trị r và s cho B, bởi vì sau đó B có thể tính $u = rs$ và như vậy làm giảm độ mật của thẻ. Bên A chứng tỏ khả năng của mình trong việc cung cấp giá trị r hoặc s theo yêu cầu và điều đó nói lên rằng bên A biết cả hai; tức biết cả u.

Việc cá nhân hóa các thẻ là một vấn đề phức tạp, bởi vì sẽ có k giá trị $v_j = f(I, c_j)$, với các số nguyên nhỏ c_j khác nhau xác định cho các v_j là các số dư quân phương modulo m. Cũng đã có những phép thử với $c = 1, 2, 3...$ và lấy k số đầu tiên tạo số dư quân phương. Thẻ lưu giữ các giá trị mật s_j như là $s_j^2 = v_j^{-1}$ = modulo m. Thẻ cũng chứa đồng nhất I của mình, modul m cho các phép tính và các giá trị c_j tương ứng với s_j. Trước khi bắt đầu cuộc trao đổi chính, thẻ cung cấp cho đầu cuối giá trị I và các c_j để các đầu cuối có thể tính các v_j, từ $j = 1$ cho đến j = k, để chuẩn bị cho thủ tục đồng nhất.

Mỗi một chu trình đồng nhất gồm có ba giai đoạn thực hiện như đã nêu trên, đó là:

1. Bên A tạo số ngẫu nhiên r và gửi $x = r^2$ cho B;

2. Bên B lấy k số nhị phân ngẫu nhiên được lựa chọn là $e_1, e_2, ..., e_k$ và gửi véctơ đó cho A;

3. Bên A gửi:

$$y = r\prod_{j=1}^{j=k} s_j^{e_j}$$

Cho thiết bị đầu cuối. (Thẻ lựa chọn chính là một tập hợp các giá trị mật s_j cho yêu cầu của B, bằng cách tính tích, nhân với r và sau đó gửi kết quả y cho đầu cuối).

4. Việc kiểm tra bởi đầu cuối từ các phép toán đó sẽ nhận được kết quả bằng biểu thức bình phương:

$$y^2 = r^2 \prod_{j=1}^{j=k} s_j^{2e_j}$$

Bởi vì $r^2 = x$ và $s_j^2 = v_j^{-1}$; cho nên biểu thức trên có thể viết:

$$x = y^2 \prod_{j=1}^{j=k} v_j^{e_{ij}}$$

Việc kiểm tra đó có thể thực hiện bởi đầu cuối bằng cách sử dụng tập hợp các v_j đã được tính toán trước đó.

Mô hình đồng nhất của Fiat - Shamir rất thích hợp cho việc ứng dụng thẻ có chip điện tử bởi vì các chip có khả năng lưu giữ và các phép toán xử lý.

5.8.3. Mô hình chữ ký số của Fiat - Shamir

Các trao đổi qua lại như trình bày ở các mục trên có thể chứng giải cho B rằng, việc đồng nhất do A biểu thị là xác thực. Tuy vậy, cũng có thể xảy ra trong quá trình trao đổi bị lỗi hoặc xảy ra tranh chấp; trong trường hợp đó làm thế nào bên B có thể cung cấp cho trọng tài chứng cớ về sự đồng nhất của B với cuộc trao đổi đó ? Một cuộc trao đổi thuần tuý và đơn giản thường không chứa đựng các tang chứng, bởi vì thường được sắp xếp theo kiểu các bit là ngẫu nhiên. Về nguyên lý, một chữ ký có thể được kiểm tra sau đó bởi một trọng tài độc lập, xuất phát hoàn toàn từ A và không có trao đổi với B.

Việc xây dựng chữ ký là do A ký cho đoạn tin M. Mô hình dựa trên cơ sở của sự đồng nhất. Ký hiệu I là các dữ liệu đồng nhất, véctơ k thành phần được xác định bởi các $v_j = f(I, c_j)$. Các c_j là các số nguyên nhỏ, được chọn để cho các v_j tương ứng là các số dư quân phương. Các c_j cần phải được biểu thị đồng thời với I, để cho phép xác định các v_j ở cuối việc kiểm tra chữ ký. Cũng giống như trước kia, chữ ký được bên A giữ bí mật các giá trị s_j với $s_j^2 = v_j^{-1}$ modulo m.

Quá trình thực hiện chữ ký do A bao gồm ba giai đoạn, giống như các chu trình trong thể thức đồng nhất đã giới thiệu trên, nhưng chúng được thực hiện tất cả cùng lúc theo phương pháp sau đây:

1. Tạo t số ngẫu nhiên r_i và tính $x_i = r_i^2$.

2. Tính $f(M, x_1, x_2, ..., x_t)$ (trong đó f là một hàm một chiều và sử dụng kt bít đầu tiên của kết quả tính cho các hệ số e_{ij} của một ma trận với $i = 1, 2, ..., t$ và $j = 1, 2, ..., k$.

3. Tính véctơ chữ ký số cho t thành phần bởi biểu thức:

$$y_i = r_i \prod_{j=1}^{j=k} v_j^{e_{ij}}$$

Với $i = 1, 2, ..., t$.

Chúng chỉ xuất hiện trong tích các giá trị s_j tương ứng với $e_{ij} = 1$.

Khi bài toán kết thúc tức chữ ký đã hoàn chỉnh, và sau đó có thể đưa vào đoạn tin M và truyền cho phía kiểm tra (ví dụ B), để kiểm tra nhờ có biết được modulo m. Việc kiểm tra đó cần phải có các dữ liệu nhận dạng I, các số nguyên nhỏ c_j, ma trận e_{ij} và véctơ chữ ký y_i.

Việc kiểm tra trước tiên là tính véctơ đã cho với các thành phần $v_j = f(I, c_j)$ với $j = 1,2,..., k$. Sau đó tính tiếp:

$$z_i = y_i^2 \prod_{j=1}^{j=k} v_j^{e_{ij}}$$

Các giá trị z_i tính toán đó chính bằng các x_i được sử dụng trong chữ ký, như đã thực hiện trong thể thức đồng nhất.

Cuối cùng, cần phải đánh giá $f(M, z_1, z_2,..., z_t)$ và so sánh kt bit đầu tiên của nó với e_{ij}. Nếu chúng trùng khớp nhau tức chữ ký đã được kiểm tra là đúng.

Bảng 5.1 mô tả một số đặc tính so sánh của một số thuật toán được ứng dụng.

Bảng 5.1.

Thuật toán	Kích cỡ theo byte			Số các phép nhân thực hiện	
	Khóa công khai*	Khóa bí mật	Chữ ký số	Khởi tạo	Kiểm tra
RSA	64	64	64	768	2+
Fiat - Shamir:					
k = 9; t = 8	6	576	521	44	44
k = 64; t = 1	40	4096	72	33	33
Số v nhỏ:					
Căn bậc hai	104	4096	72	33	1
Căn bậc ba	64	4096	72	34	2
Căn bậc 16	76	1024	72	36	4
Số s nhỏ:	1024	128	72	22	66

* Bao gồm các giá trị c_i hoặc chọn các số nguyên nhỏ.

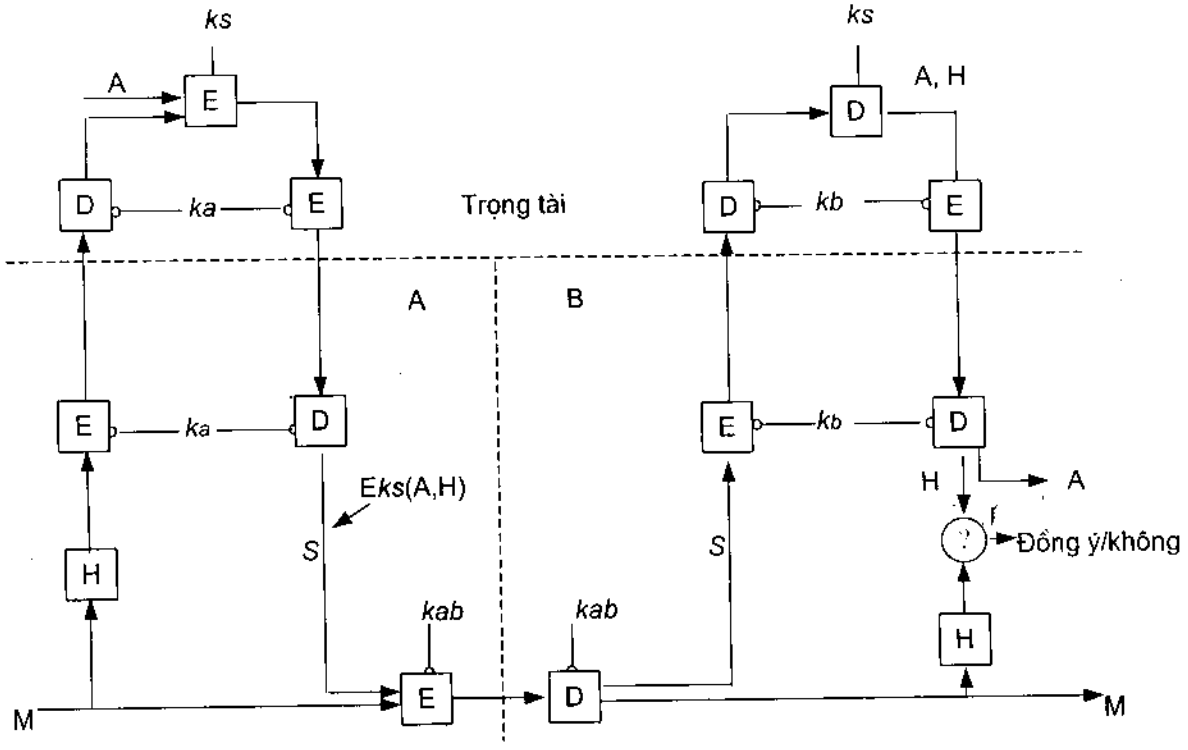
\+ Số mũ công khai được sử dụng là ba (mủ bậc ba và căn bậc ba)

5.9. CHỮ KÝ SỐ CÓ TRỌNG TÀI

Có khả năng sử dụng một mật mã đối xứng để tạo ra các chữ ký số mà không mở rộng khối dữ liệu, không dùng các khóa có kích cỡ lớn và chỉ dùng một lần như các phương pháp giới thiệu ở các mục trước. Để thực hiện được điều đó thì cả hai bên, bên gửi và bên nhận, cần phải trông cậy vào một dịch vụ trọng tài mà cả hai bên tín nhiệm. Hình 5.7 mô tả sơ đồ khối thực hiện nguyên lý đó.

Cũng giống như các phương pháp khác, ở đây cũng không ký đoạn tin nguyên vẹn mà ký đoạn tin rút gọn qua hàm một chiều $H(M)$, với một giá trị ngẫu nhiên khởi đầu I. Bên gửi nhận chữ ký bằng cách gửi $H(M)$ cho trọng tài, theo một kênh được bảo vệ bởi khóa mã ka, khóa ka đó chỉ có bên gửi và trọng tài biết. Cũng lúc đó đoạn rút gọn được mã hoá, Eka ($H(M)$), bên gửi truyền cho trọng tài, để nhận dạng nó với giá trị rõ với bên gửi là A. Nhờ có sự đồng nhất đó mà trọng tài có

thể lấy từ bộ nhớ ra khóa mã *ka* để giải mã đoạn tin nhận được theo *H*(M) của A và sau đó mã hóa với khóa riêng *ks* cho chữ ký. Chỉ có một mình trọng tài biết khóa mã *ks* được bảo vệ cẩn thận trong một modul chống trộm. Chữ ký *S* = *Eks*(A,*H*(M)) được trả về cho bên gửi, được mã hóa với khoá *ka* (khóa chung của bên gửi và trọng tài). Với người gửi, khóa *S* như vậy đã được giải mã và đoạn tin được ký (*M*, *S*) sẽ được địa chỉ hóa cho bên nhận với khóa *kab* (khóa *kab* cả hai bên A và B đều biết), sau đó gửi cho bên B như mô tả trên hình 5.7.



Hình 5.8. ***Sơ đồ khối nguyên lý hoạt động của chữ ký số có trọng tài.***

Việc kiểm tra chữ ký trong trường hợp này phải nhờ trọng tài. Bên nhận gửi cho trọng tài giá trị *S* đã mã hóa với khóa *kb* (khóa *kb* là khóa chung cho bên nhận và trọng tài) cũng như giá trị B để cho phép trọng tài lấy ra khóa đúng cho B để giải mã. Trọng tài xác định được *S* bởi phép giải mã với khóa *kS* và gửi (A, *H*(M)) đã mã hóa với khóa *kb* cho bên nhận. Bên nhận lúc này có hai đại lượng: đại lượng A chỉ rõ nguồn (nơi gửi) đoạn tin M và bản rút gọn H(M) của đoạn tin M đó. Cuối cùng, có thể tính ngược lại để so sánh bằng cách áp dụng hàm một chiều H cho đoạn tin M. Nếu như việc so sánh đó là đương tính tức đoạn tin được truyền chính xác, không thay đổi từ A đến.

Cũng có nhiều cách để thực hiện thủ tục trên. Có thể bảo vệ các kênh giữa trọng tài và các bên tham gia A và B nhờ một kiểm tra đồng nhất (mã xác thực đoạn tin) ngay khi mã hoá, nếu như cho phép. Chữ ký cũng có thể được xây dựng với một hàm một chiều thay vì việc mã hoá. Tiến trình kiểm tra cần phải có đầu vào các giá trị A và *H*(M) và trả lại chữ ký để so sánh với giá trị chữ ký *S* lấy từ đoạn tin. Phương pháp trọng tài và kiểm tra chữ ký như trình bày trên phụ thuộc vào sự công bằng (tính thanh liêm) của trọng tài.

Chương 6

KIỂM TRA NHẬN DẠNG

6.1. GIỚI THIỆU TỔNG QUAN

Có một yếu tố quan trọng không thể thiếu được trong việc bảo vệ an toàn các hệ thống thông tin, đặc biệt là trong bối cảnh các phương tiện thiết bị ngày càng hiện đại, các dịch vụ thông tin ngày càng đa dạng, đó là con người. Một ví dụ đơn giản là hệ thống "rút tiền tự động" của ngân hàng cung cấp tự động tiền mặt cho những ai được thông qua kiểm tra nhận dạng dưới một phương thức nào đó. Các máy tính và mạng máy tính đã phục vụ đắc lực và làm gia tăng đáng kể khả năng xử lý trao đổi thông tin, nhưng mặt khác cũng gia tăng các khả năng xâm nhập, các hoạt động gian trá mà trong nhiều trường hợp việc ngăn ngừa, phát hiện không đơn giản.

Một ví dụ khác, một ngân hàng có một triệu khách hàng. Nếu ngân hàng đó muốn biết các khách hàng của mình thông qua một "số nhận dạng cá nhân PIN" (Personal Identification Number) thì số đó phải chứa tối thiểu 6 con số. Nếu ngân hàng đó muốn biết khách hàng mình thông qua các chữ ký thì phải có phương pháp để phân biệt chữ ký của người này với người kia. Số khách hàng càng lớn thì việc nhận dạng càng khó.

Tất cả công việc trên dựa vào kỹ thuật hoặc phương pháp kiểm tra nhận dạng, hoặc còn gọi là kiểm tra đồng nhất.

Trong thực tế ứng dụng có thể chia các phương pháp kiểm tra nhận dang làm bốn nhóm lớn và trong đó có thể có một số hệ thống chứa đựng các phần tử phụ thuộc vào một hoặc nhiều trong số các nhóm đó. Có thể kiểm tra nhận dạng một con người nào đó thông qua:

(a) Điều mà con người đó biết
(b) Vật mà con người đó có
(c) Đặc tính vật lý của người đó
(d) Kết quả một hành động bột phát của người đó

Từ mật khẩu là một ví dụ thuộc nhóm (a); hộ chiếu hoặc thẻ căn cước là thuộc nhóm (b); các nhóm (c) và (d) không phải luôn luôn tách biệt nhau: vân tay số được xem như một đặc trưng vật lý và chữ ký được xem như một hành động bột phát bởi vì mỗi hành động không thể được kiểm tra riêng rẽ.

6.2. KIỂM TRA NHẬN DẠNG THÔNG QUA TỪ MẬT KHẨU

6.2.1. Các từ mật khẩu và trao đổi từ mật khẩu

Từ mật khẩu là một kiểu kiểm tra nhận dạng dựa trên sự hiểu biết của một người hoặc của một yếu tố nào đó. Từ mật khẩu thường được sử dụng ở các văn bản hoặc sự kiện luôn có sự biến đổi, từ thời AliBaba với cái hang đá và câu thần chú, trải qua các mật khẩu dùng trong quân sự cho đến ngày nay, các thể thức truy nhập các hệ thống thông tin máy tính. Cũng đã có nhiều nghiên cứu sử dụng các phương pháp từ mật khẩu khác nhau và có thể chia các từ mật khẩu thường được sử dụng trong các hệ thống truyền tin máy tính theo các dạng sau đây:

- Từ mật khẩu nhóm, chung cho nhiều người sử dụng hệ thống;
- Từ mật khẩu cá nhân, dành riêng cho mỗi cá nhân;
- Từ mật khẩu không cá nhân, không dành riêng cho mỗi cá nhân mà dùng để kiểm tra đồng nhất cá nhân;
- Từ mật khẩu được thay đổi sau mỗi lần truy nhập hệ thống.

Các từ mật khẩu nhóm thỉnh thoảng được sử dụng như một phần của một thủ tục truy nhập hệ thống, cho phép thừa nhận một dịch vụ nào đó từ một đầu cuối. Bởi vì loại mật khẩu này dành cho người sử dụng cho nên độ mật kém.

Các từ mật khẩu cá nhân có độ mật lớn hơn, có khả năng ngăn ngừa các truy nhập bất hợp pháp. Độ mật của mật khẩu một phần phụ thuộc vào phương pháp tạo và bảo vệ mật khẩu.

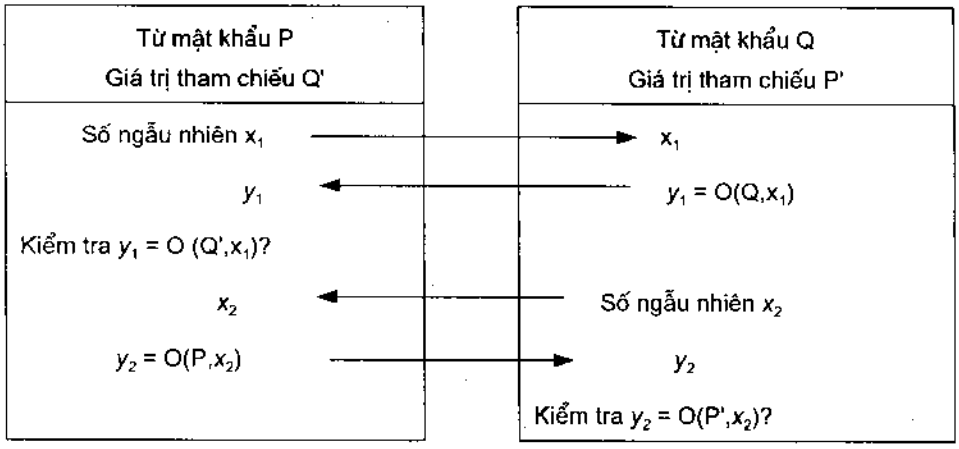
Các từ mật khẩu không cá nhân ngày nay cũng đang được sử dụng phổ biến. Nó thường được qui vào một từ mật khẩu tương đối ngắn (ví dụ bốn số thập phân) cho mỗi người sử dụng nhưng để nhận dạng nó thì phụ thuộc vào một số rất lớn mà người sử dụng xem như không nhớ được, ví dụ thẻ có dải băng từ.

Các từ mật khẩu thay đổi sau mỗi lần truy nhập hệ thống là loại thứ tư. Để thực hiện cần chuẩn bị trước một bản liệt kê các từ mật khẩu và cung cấp bản sao cho người sử dụng. Trong trường hợp này, tất nhiên các bản liệt kê các từ mật khẩu phải được bảo vệ cẩn thận. Có nhiều phương pháp bảo vệ, kể cả việc sử dụng mật mã. Cũng có thể có phương pháp đơn giản là bản liệt kê đó được cắt làm đôi, gửi cho người sử dụng hai lần cách biệt nhau để giảm nguy hiểm thu trộm một từ mật khẩu nguyên vẹn.

Chọn một từ mật khẩu như thế nào cũng là một bài toán lý thú. Người ta đã từng để cho những người sử dụng chọn các từ mật khẩu cho mình một cách tự do, không có lời khuyên nào và cũng không có ràng buộc nào. Khuynh hướng chung của những người sử dụng là chọn từ mật khẩu sao cho dễ nhớ. Qua thống kê trắc nghiệm thử 3289 từ mật khẩu được chọn một cách tự do đó có: 15 từ mật khẩu chỉ chọn duy nhất một chữ cái ASCII, 72 chọn hai ký tự ASCII, 464 chọn ba ký tự, 477 chọn bốn ký tự chữ cái/số, 706 chọn 5 chữ cái, tất cả đều viết hoa hoặc tất cả viết thường; và 605 chọn sáu chữ cái, tất cả viết thường.

Có nhận xét xu hướng chọn của người sử dụng: chữ viết tắt họ tên hoặc đảo ngược, tên thành phố, số xe, số bảo hiểm, số điện thoại... Tất nhiên cách chọn đó chỉ để phục vụ mục đích chính là dễ nhớ, còn độ mật rất kém. Trong các hệ thống thông tin thường dùng từ mật khẩu với tám ký tự và thường sử dụng những từ mật khẩu vô nghĩa nhưng dễ nhớ.

Khi hệ thống (A) và người sử dụng (B) cần trao đổi từ mật khẩu cho nhau thì hai bên phải trao từ mật khẩu cho nhau và phải đảm bảo rằng, trong thời điểm trao đổi đó không bị tiết lộ hoặc không bị kẻ xâm nhập khai thác. Bài toán có thể thực hiện theo hàm một chiều như mô tả ở hình 6.1. Giả thiết rằng, ở đây A và B sở hữu từ mật khẩu của mình lần lượt là P và Q.



Hình 6.1. ***Mô tả phương pháp dùng hàm một chiều để kiểm tra việc trao đổi từ mật khẩu cho nhau***

Mỗi một bên biết từ mật khẩu của bên kia và muốn kiểm tra tính đúng đắn của chúng. Thay vì việc trao đổi trực tiếp các từ mật khẩu cho nhau, ở đây sử dụng một hàm một chiều để trả lời cho một câu hỏi do phía bên kia đặt ra. Ví dụ A hỏi B bằng cách gửi cho B tham số ngẫu nhiên x_1, và B trả lời bằng cách tính ảnh y_1 của giá trị đó theo hàm một chiều,

$$y_1 = O(Q,x_1)$$

Hàm một chiều phải là giống nhau trong khi cả hai bên đều biết x_1 và y_1, và trong thực tế cũng không có khả năng tính ngược lại để có Q. Bên A không có khả năng tìm được Q xuất phát từ trả lời đó, nhưng có thể đối chiếu giá trị Q của B bằng cách áp dụng hàm một chiều cho giá trị x_1 mà nó đã tạo ra. Nếu như việc kiểm tra đó là dương tính thì thừa nhận B chính là chủ thể đúng đắn của Q.

Cũng bằng trình tự tương tự, B đặt câu hỏi x_2 và nhận được trả lời y_2 với

$$y_2 = O(Q, x_2)$$

Trả lời đó đến lượt nó được kiểm tra như A đã kiểm tra để chắc rằng P là từ mật khẩu của A.

Phương pháp trên chỉ tin tưởng khi số phần tử của từ mật khẩu đầy đủ. Với các từ mật khẩu ngắn thì độ bảo mật sẽ kém. Có thể khắc phục bằng phương pháp bổ sung thêm một số ngẫu nhiên sau khi trao đổi hỏi và đáp. Ví dụ, giả thiết từ mật khẩu Q là 16 bit được móc xích với một số ngẫu nhiên R1 có 40 bit, để tạo thành một biến số 56 bit và lúc đó trả lời của B cho câu hỏi đó sẽ là:

$$y_1 = O(Q||R_1,x_1)$$

trong đó $Q||R_1$ là kết quả trên 56 bit của móc xích của hai biến số đó. Sau khi mỗi bên đã thực hiện hỏi, đáp thì các số dư R_1 và R_2 được trao đổi để thực hiện pha kiểm tra lẫn nhau.

6.2.2. Các từ mật khẩu biến đổi dựa trên hàm một chiều.

Các từ mật khẩu biến đổi có thể đảm bảo độ mật tốt hơn chống lại các sự rẽ nhánh trên đường truyền nhưng chúng cũng đòi hỏi có sự phân phối các tập từ mật khẩu mới để thay thế các từ mật khẩu đã được sử dụng. Có một phương pháp hữu hiệu dựa trên cơ sở hàm một chiều cho phép cung cấp một dãy nguyên các từ mật khẩu bằng cách phân phối chỉ một số. Nó sử dụng một hàm một chiều $y = O(x)$ và lặp lại hàm đó nhiều lần. Ký hiệu $O^n(x)$ là kết quả nhận được do áp dụng hàm một chiều n lần cho biến số x. Ví dụ:

$$O^3(x) = O(O(O(x)))$$

Để thiết lập dãy các từ mật khẩu, A lấy một giá trị ngẫu nhiên và tính giá trị y_0 để gửi cho B

$$y_0 = O^n(x)$$

Cuộc truyền đó không cần bí mật, bởi vì giá trị xuất phát x không thể suy đoán ra từ nó.

Để thay cho từ mật khẩu đầu tiên, A gửi giá trị:

$$y_1 = O^{n-1}(x)$$

B có thể dễ dàng kiểm tra nó, bởi vì $O(y_1) = y_0$ và B đã nhận được y_0 ở giá trị đầu tiên.

Dãy các từ mật khẩu được tiếp tục theo phương pháp đó. Từ mật khẩu thứ i sẽ là:

$$y_i = O^{n-1}(x)$$

Mỗi một từ mật khẩu sẽ được kiểm tra theo quan hệ đó với từ mật khẩu trước nó, mà từ mật khẩu trước đã được kiểm tra. Dãy sẽ kết thúc ở từ mật khẩu thứ n mà nó chính là giá trị x.

6.3. KIỂM TRA NHẬN DẠNG THÔNG QUA THẺ SỞ HỮU

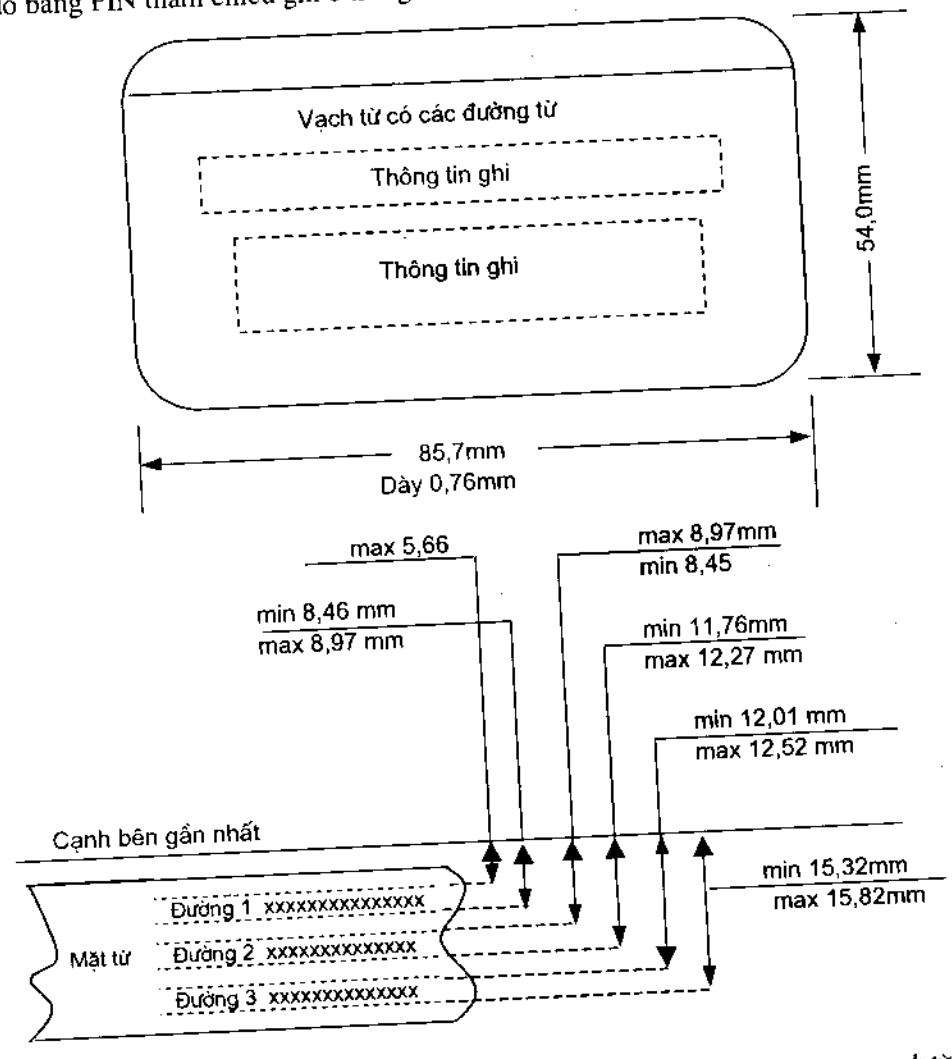
Theo nghĩa rộng, việc kiểm tra truy nhập thông qua việc sở hữu một vật nào đó của người sử dụng cũng là một dạng kiểm tra nhận dạng. Mỗi người chúng ta hầu như ai cũng có sở hữu một vật riêng mình; ví dụ chìa khóa nhà, chìa khóa xe... Việc truy nhập hệ thống thông tin cũng có thể xem như sở hữu một chìa khóa được cài đặt trong đầu cuối để truy nhập hệ thống. Chìa khóa đó nếu vô ý để kẻ khác sao chép (đánh chìa khoá) thì sẽ gây nguy hiểm cho hệ thống. Có nhiều dạng kiểm tra truy nhập thông qua vật sở hữu của người sử dụng, nhưng sau đây chỉ giới hạn ở kiểm tra nhận dạng thông qua các thẻ sở hữu được xử lý tự động hóa trong các hệ thống thông tin.

6.3.1. Thẻ nhận dạng các đường từ

Một dạng thẻ được sử dụng phổ biến để chứng minh tính hợp pháp của người sử dụng trên một lĩnh vực nào đó (ví dụ thẻ căn cước, thẻ bảo hiểm, thẻ rút tiền ở ngân hàng...), được xem như một chìa khóa để truy nhập hệ thống và đồng thời để hệ thống có thể kiểm tra nhận dạng quyền truy nhập hợp pháp là thẻ có đường từ.

Thẻ đã được cơ quan tiêu chuẩn quốc tế, ISO, chuẩn hóa về kích thước thẻ và các vạch từ để có thể kiểm tra thống nhất trên các thiết bị của các hệ thống. Khuôn dạng dữ liệu ghi có thể khác nhau. Hình 6.2 mô tả kích thước và cách bố trí thông tin trên một thẻ có vạch từ chuẩn.

Vạch từ thường có ba đường từ để ghi dữ liệu. Ngoài các dữ liệu cần thiết, trong vạch từ có phần kiểm tra nhận dạng. Từ mật khẩu sử dụng để nhận dạng thường được gọi là số nhận dạng cá nhân, PIN (personal indentification number), được nhớ trên một đường từ dưới dạng mã hóa và khi đọc kiểm tra nhận dạng phải giải mã. Trong một số hệ thống, PIN được lưu trữ ở trung tâm hệ thống. Việc nhận dạng được thực hiện do trung tâm ngay thời điểm mà PIN xuất hiện. Nếu so sánh PIN đó bằng PIN tham chiếu ghi ở trung tâm, việc truy nhập là hợp pháp.



Hình 6.2. ***Kích thước thẻ có vạch từ và các đường từ trong vạch từ***

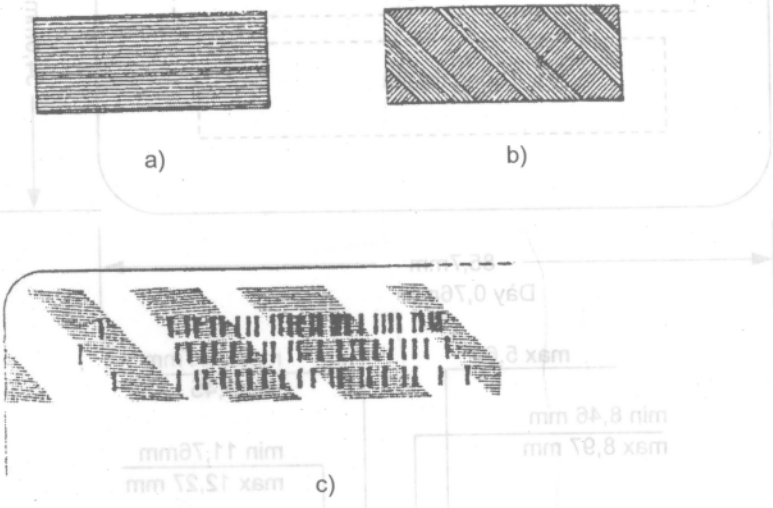
Nhược điểm của thẻ nhận dạng có các đường từ là các đường từ đó có thể được sao chép lại một cách dễ dàng bằng các kỹ thuật thông thường. Cũng vì vậy mà đã có khá nhiều phương pháp để ghi từ trên thẻ, đảm bảo tính bí mật, chống sao chép và chống làm giả. Sau đây sẽ giới thiệu hai phương pháp trong nhiều phương pháp đang được ứng dụng trong thực tế.

6.3.2. Băng từ Watermark

Băng từ Watermark cho phép mã hóa các dữ liệu dưới dạng từ, cố định, không thể xóa trong cấu trúc của băng. Việc ghi dữ liệu cố định đó được thực hiện khi sản xuất băng, theo phương pháp

sắp xếp các phần tử của trường từ lúc còn ẩm. Trong phần lớn các băng (ví dụ loại băng từ dùng oxít sắt γ), các phần tử của chúng có dạng hình kim dài và rất mảnh. Để tạo được một cấu trúc theo kiểu "dệt đan chéo nhau" (watermark), trước tiên người ta sử dụng trường một chiều có phương nghiêng góc 45° với chiều dài của băng để sắp xếp ngay ngắn các phần tử. Tiếp theo sử dụng một đầu ghi từ đặc biệt, có các xung từ, cho băng lướt qua đầu ghi. Khi có dòng điện, hướng của các phần tử sẽ ngược nhau 45°. Băng từ hóa sản xuất dạng ướt sau đó được đem sấy khô và định hướng các phần tử theo dữ liệu ghi lúc đó được cố định.

Hình 6.3 mô tả sự sắp xếp các phần tử theo phương pháp trên. Phương pháp do Emidata đề xuất và đặt thương hiệu là Watermark, có nghĩa là dạng "gợn sóng" hoặc "bóng mờ".



Hình 6.3. *Mô tả sự sắp xếp các phần tử của băng từ Watermark*
a) Từ hóa theo phương pháp truyền thống
b) Từ hóa theo phương pháp Watermark
c) Dữ liệu ghi trên các đường từ

Để kiểm tra việc ghi cố định các nội dung (dữ liệu) trên các đường từ theo cấu trúc trên cần có một đầu đọc đặc biệt. Với đầu đọc đó, trước tiên băng từ được đặt trong từ trường một chiều, sau đó sẽ đọc các dữ liệu ghi theo các hướng thích ứng. Bởi vì băng lúc đầu được đặt trong trường một chiều trước khi đọc cho nên nó tương ứng với đường số 0, nói chung là không có số liệu ghi. Từ đường từ 1 đến đường từ 3 việc ghi và đọc là được thực hiện bởi các đầu đọc có định hướng chuẩn. Đường 0 cũng có thể ghi các dữ liệu cố định 0 và 1 để nhận dạng các đường từ và thẻ.

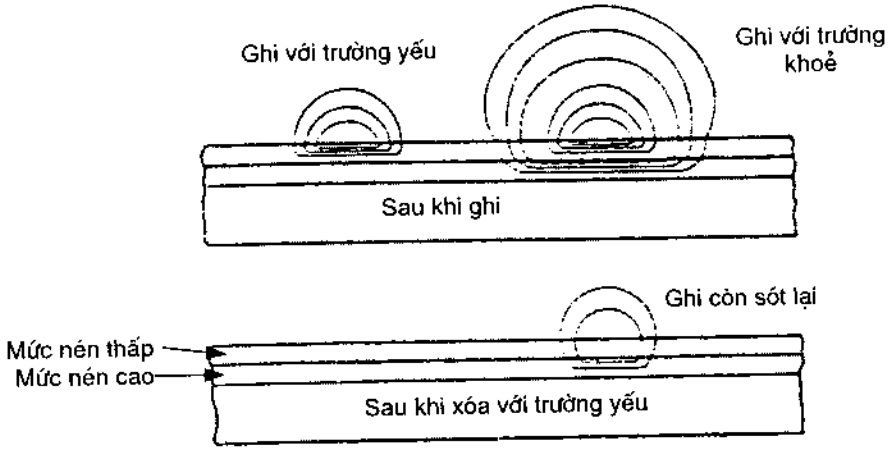
Trong nhiều trường hợp băng từ sử dụng các số ngẫu nhiên. Nếu gọi W là số Watermark và D là các dữ liệu bảo vệ có trên đường 1 và đường 3 thì có thể nhớ tất cả D một vùng điều khiển có mã hóa C = E*k*(W,D) trên một đường từ. Khi đọc có thể kiểm tra quan hệ mật mã đó với khóa *k* mà hệ thống kiểm tra đã biết.

Tính bền vững và bảo mật ở đây phụ thuộc vào độ khó mà kẻ xâm nhập gặp phải nếu muốn làm thẻ giả mạo. Độ khó đó là rất lớn bởi vì cả quá trình sản xuất đòi hỏi kỹ thuật công nghệ rất cao kết hợp với việc mã hoá.

6.3.3. Thẻ sử dụng băng từ hai lớp

Thẻ sử dụng băng từ hai lớp là thẻ có hai lớp chất có từ tính, một lớp có kháng từ cao và một lớp có kháng từ thấp, được đặt chồng lên nhau để ghi dữ liệu. Lớp trên, gần đầu đọc là lớp có kháng từ thấp. Việc ghi dữ liệu được thực hiện bởi một đầu ghi có kháng từ cao và từ được ghi trong cả hai lớp. Mỗi lần đọc, băng từ được đưa qua một từ trường xóa để sao cho xóa hết các nội dung ghi trong lớp kháng từ thấp. Các dữ liệu ghi cố định sau đó sẽ được đọc bình thường. Hình 6.4 mô tả cấu trúc các đường sức từ trường trên hai lớp từ tính và kiểu ghi/đọc dữ liệu loại băng từ hai lớp đó.

Bảo vệ bí mật của một hệ thống sử dụng thẻ có băng từ hai lớp dựa trên hai yếu tố sau đây. Đầu tiên thẻ được đưa vào một trường xóa của đầu đọc để ngăn ngừa việc làm giả theo kiểu băng từ thông thường. Hai là việc khó khăn tạo ra một từ trường kháng từ cao đủ mức cần thiết để ghi dữ liệu trong lớp kháng từ cao, ngăn ngừa việc làm giả do chúng đánh cắp được thẻ băng từ.



Hình 6.4. ***Cấu trúc băng từ có hai lớp***

6.3.4. Thẻ sử dụng chíp điện tử

Một loại thẻ đang được sử dụng phổ biến hiện nay là loại thẻ trên có gắn chíp điện tử (còn gọi là rệp điện tử); thẻ còn có tên gọi là thẻ tinh khôn (smart cards), trong chip điện tử đó có các bộ vi xử lý và các bộ nhớ.

Với công nghệ hiện nay có thể cài các bộ nhớ trong các chip điện tử của thẻ đến hàng Mbyte trong lúc các dải băng từ chỉ có thể nhớ tối đa vài trăm byte.

Kích thước của thẻ cũng sử dụng kích thước chuẩn (8,57cm×5,40cm×0,76cm), có một số thẻ có chiều dày mỏng hơn (< 0,76 mm). Các tiếp điểm mạch của chip là các tiếp điểm kim loại trên bề mặt chip, trong số đó có cả đầu tiếp điểm cung cấp điện và các đầu vào, đầu ra để làm việc với một thiết bị đầu cuối. Có hai loại chip thường được sử dụng, đó là loại chip có bộ vi xử lý kèm theo bộ nhớ và loại chỉ có bộ nhớ. Các bộ nhớ dùng loại E^2PROM hoặc ROM. Một sự làm biến đổi bất hợp pháp nào đó các dữ liệu trong ROM là hoàn toàn khó khăn.

Thẻ có chip điện tử được ứng dụng trong nhiều lĩnh vực. Việc nhận dạng thẻ được thực hiện bởi việc giải mã các dữ liệu được mã hóa trong chip tại trung tâm hệ thống hoặc tại thiết bị cuối có kiểm tra.

Thẻ có chip điện tử với bộ nhớ của chip không thể ghi lại được có thể phân phối cho một số lượng lớn khách hàng làm thẻ thanh toán khi mua hàng và thanh toán tiền với tài khoản của khách hàng ở ngân hàng. Hệ thống thanh toán bằng thẻ ngân phiếu, không sử dụng tiền mặt đó rất tiện lợi trong hoạt động ngân hàng và đang được sử dụng phổ biến ở nhiều nước.

Việc bảo mật các dữ liệu trong thẻ có thể được thực hiện theo phương pháp hợp nhất các dữ liệu, dung lượng xử lý và mã hóa vào một thể thống nhất.

6.4. KIỂM TRA NHẬN DẠNG THÔNG QUA CÁC ĐẶC ĐIỂM CÁ NHÂN

6.4.1. Giới thiệu

Các từ mật khẩu có thể bị lộ do sơ suất và các thẻ nhận dạng có thể bị mất hoặc bị làm giả mạo. Ngoài các hệ thống nhận dạng như đã để cập ở các mục trên, còn có một hệ thống nhận dạng khác mà một số hệ thống chức năng thường sử dụng, đó là các hệ thống nhận dạng thông qua các đặc điểm nhận dạng cá nhân.

Việc nhận dạng các đặc điểm nhận dạng cá nhân đã có một lịch sử lâu đời và được ứng dụng trong các công việc điều tra tội phạm. Vào thế kỷ 16, 17 cũng đã xuất hiện các hệ thống nhận dạng như lấy các số đo về chiều cao, chiều dài của cánh tay và các ngón tay, độ rộng của hộp sọ, chiều dài của chân v.v... về một con người phạm tội cần theo dõi kiểm tra. Phương pháp còn được gọi là phương pháp nhân chủng học. Ở thời kỳ mà các hệ thống đó đang được sử dụng, người ta cũng chưa biết rằng, hai người khác nhau có thể có cùng các số đo với dung sai rất bé. Thế kỷ thứ 18, một số nước châu Âu đưa vân tay vào các hệ thống nhận dạng. Trên thực tế, việc sử dụng vân tay để nhận dạng đã được ứng dụng ở Trung Quốc từ lâu đời trong các khế ước, văn tự thay cho chữ ký.

Việc nhận biết một người quen là một kinh nghiệm bẩm sinh của con người. Sự nhận biết đó dựa vào các đặc điểm vật lý khác nhau như khuôn mặt, màu da, bộ tóc, vóc dáng và có thể là bộ điệu, tiếng nói, tiếng bước chân, mùi hương v.v... Con người cũng có thể nhận biết nhau qua chữ ký hoặc chữ viết. Trong các đặc điểm nhận dạng trên có một số đặc điểm có thể tận dụng để xây dựng các hệ thống kiểm tra nhận dạng tự động hóa có sự tham gia của máy tính điện tử.

6.4.2. Nhận biết tự động và dung sai của hệ thống

Cho đến nay cũng đã có rất nhiều nghiên cứu, phần lớn là thực nghiệm, ứng dụng các phương pháp kỹ thuật khác nhau để nhận biết các đặc điểm nhân dạng cá nhân. Trong số các nghiên cứu đó, một số hướng về các đặc điểm chữ ký tay, các vân tay và giọng nói. Một số nghiên cứu khác có xu hướng nghiên cứu sâu về hộp sọ (não tướng học), khuôn mặt, đường vân của môi hoặc của gan bàn chân, các đường vân trên bàn tay hoặc cổ tay, hài cốt v.v... Sau đây chỉ đề cập một số đặc điểm đặc trưng, có thể sử dụng các hệ thống xử lý tự động hóa thường gặp trong ứng dụng thực tế đó là các đặc điểm về chữ ký tay, vân ngón tay và giọng nói. Các đặc điểm khác cũng có thể sử dụng để kiểm tra nhận dạng, nhưng thường chỉ sử dụng trong những trường hợp cá biệt, chúng không thích hợp

cho một hệ thống nhận dạng có sử dụng các thiết bị máy móc tự động hóa với số đông đối tượng tham gia hệ thống, ví dụ việc lấy đường vân môi, đo hộp sọ chỉ có thể thực hiện được nếu người sử dụng đồng ý.

Ở đây cũng cần lưu ý rằng, xác suất trùng nhau về đặc điểm nhân dạng của hai người trong thực tế là rất bé, tuy vậy nó cũng có một dung sai nào đó do sự biến đổi qua năm tháng mà hệ thống cũng cần phải tính đến. Một số hệ thống phân biệt hai loại dung sai. Dung sai loại I cho các biến đổi đặc điểm cá nhân "hợp pháp" và dung sai loại II cho các thay đổi đặc điểm cá nhân "bất hợp pháp". Vấn đề sẽ được xem xét sau khi phân tích một số hệ thống cụ thể.

6.4.3. Kiểm tra nhận dạng chữ ký tay

Đã từ lâu, chữ ký bằng tay được xem như là một sự chứng thực của chủ thể trong các văn bản, thư từ trao đổi, hợp đồng ký kết, hóa đơn biên nhận v.v... Các chữ ký bằng tay đó đã trở thành một phương tiện để kiểm tra nhận dạng về một người, ví dụ chủ sở hữu một thẻ tín dụng. Các phương pháp truyền thống kiểm tra chữ ký bằng mắt thường, chỉ cần sự tinh mắt của người kiểm tra. Một khi có sự tranh chấp về chữ ký, đôi khi phải đưa ra toà án, thì lúc đó cần phải có sự tham gia của các giám định viên chuyên nghiệp.

Việc thực hiện một chữ ký giả mạo, trong thực tế là một công việc không dễ dàng, bởi vì điều trước tiên là phải có cùng tốc độ ký với chữ ký thật. Cũng đã có khá nhiều công trình nghiên cứu về nhận dạng chữ ký tay. Ở đây, chúng ta sẽ không quan tâm đến những gì đã viết trong văn bản mà chỉ quan tâm đến việc nhận dạng tác giả của chữ ký tay đó. Việc nhận dạng có thể được thực hiện với một chữ ký đã được ký hoặc có thể là đang ký. Có nhiều nhận dạng quan tâm lúc đang ký. Tuy vậy, việc nhận dạng những chữ ký đã ký là một công việc khá quan trọng, thường gặp nhiều trường hợp giả mạo chữ ký với các ý đồ gian dối khác nhau trong thực tế.

Các chữ ký giả mạo có thể được phân làm ba dạng sau đây: ký giả mạo tự do (freehand), ký giả mạo bắt chước và sao chép (đồ) lại chữ ký thật.

Ký giả mạo tự do, ví dụ một ngân phiếu nào đó của chủ sở hữu bị mất mà một kẻ nào đó đánh cắp hoặc nhặt được. Trong ngân phiếu có họ tên chủ ngân phiếu nhưng không có chữ ký. Kẻ giả mạo có thể phóng tác ra một chữ ký căn cứ vào họ tên của chủ ngân phiếu. Chữ ký giả mạo bắt chước hoặc sao chép là khi kẻ giả mạo có được chữ ký thực của chủ nhân hợp pháp.

Việc phát hiện tự động một chữ ký bằng tay giả mạo có thể căn cứ vào việc phân tích các tỷ số các kích thước và các góc nghiêng của chữ ký thật với các số liệu được lưu trữ trong máy tính. Khi cần xác định một chữ ký tay có giả mạo hay không thì các số liệu về chữ ký mới sẽ được so sánh với các số liệu của chữ ký thực đã lưu trữ trong máy tính. Có thể sử dụng phương pháp đó để phân tích so sánh hai chữ ký cần giám định. Phương pháp cho kết quả tốt, nhưng không thích hợp với trường hợp chữ ký giả mạo sao chép nguyên bản.

Một phương pháp khác khá quan trọng trong việc kiểm tra thống kê các chữ ký là sử dụng các chương trình đồ hoạ. Phương pháp được tổ chức như sau:

Bằng kinh nghiệm chọn một số "đặc trưng" của các chữ ký. Các đặc trưng đó là không đổi đối với một cá nhân nhưng lại khác nhau giữa hai cá nhân. Chỉ những đặc trưng tổng quan chứa trong toàn bộ chữ ký được chọn. Bảy đặc trưng được xem xét và mỗi đặc trưng có thang độ từ 0 đến 9. Mức sử dụng để phân biệt với các tham số đó là đủ cho việc nhận dạng chữ ký trong các ứng dụng ngân hàng, thẻ tín dụng. Các đặc điểm lựa chọn với các thang đo tương ứng có thể được xây dựng thành tôpô đồ hoạ và sử dụng chương trình đồ hoạ để kiểm tra nhận dạng. Phương pháp có độ tin cậy khá lớn. Phương pháp cũng có thể gia tăng độ tin cậy nếu kết hợp với các thông số động (của chữ ký đang ký).

Việc kiểm tra truy nhập bằng một hệ thống kiểm tra chữ ký bắt đầu bằng công việc là mỗi người sử dụng cung cấp một số chữ ký mẫu. Hệ thống kiểm tra phân tích các chữ ký đó và lưu trữ các đặc trưng vào máy tính. Có thể yêu cầu nhiều chữ ký để có một sự ổn định các tham số. Với những người ký không đều tay thì có thể hệ thống không đạt được độ ổn định cần thiết. Lúc đó cần có thêm các số đo cụ thể khác ví dụ gia tăng dung sai hoặc sử dụng một phương pháp truy nhập khác. Nhưng việc gia tăng dung sai là một kẻ hở cho kẻ giả mạo chữ ký.

6.4.4. Kiểm tra nhận dạng đường vân ngón tay.

Từ xa xưa, người ta đã nhận biết rằng, có sự khác nhau giữa người này và người khác về các đường vân trên da ngón tay và trên các bộ phận khác của con người. Người Trung Hoa cổ đã dùng phương pháp "điểm chỉ" trên các văn tự chữ Hán xem như là một chứng thực cá nhân. Đây là một nhánh của khoa học nghiên cứu về da (dermatology). Các đường vân đó mang tính gien di truyền. Qua các nghiên cứu, nhận thấy rằng, xác suất để hai người có đường vân tay giống nhau rất rất bé (một phần vài tỷ) hoặc có thể nói không có trường hợp hai người có các đường vân tay giống nhau. Trong quá trình trưởng thành, kích thước các ngón tay có thể thay đổi, nhưng các vân tay không thay đổi và chúng giữ nguyên trong suốt cả cuộc đời (trừ trường hợp phải thông qua mổ xẻ tạo hình).

Khi con người cầm hoặc sờ vào một đồ vật nào đó thì dấu vân tay của người đó sẽ để lại trên bề mặt đồ vật đó. Dấu vân tay đó có thể nhìn thấy bằng mắt thường và cũng có thể không nhìn thấy bằng mắt thường nhưng với các phương pháp đặc biệt có thể nhận biết. Chính vì vậy mà cũng đã từ lâu, dấu vân tay được ứng dụng trong các công việc điều tra hình sự.

Việc kiểm tra nhận dạng vân tay được thực hiện bằng cách so sánh các vân tay của một người khả nghi nào đó với các vân tay đã được lưu trữ. Để có thể lưu trữ các vân tay, người ta phân loại lựa chọn và sắp xếp các vân tay theo các điểm đặc trưng của vân tay đó, gọi tắt là các *đặc trưng*. Hệ thống lưu trữ và tra cứu vân tay dựa trên các điểm đặc trưng của vân tay được Sir Edward Henrry (một cảnh sát trưởng) đề xuất và xây dựng đưa vào ứng dụng năm 1897. Các đặc trưng chủ yếu của vân tay là: *Cung, quai* và *xoáy* được mô tả ở hình 6.5a. Cũng theo Henry, mỗi một dạng chính trên có thể chia nhỏ thành hai hoặc nhiều dạng con. Mặt khác cũng có những dạng hỗn hợp mà các thành phần có liên quan đến các dạng khác.

Hình 6.5. Mô tả các đặc trưng của vân tay

Mỗi một vân ngón tay có một hoặc nhiều đặc trưng trên. Có những chi tiết rất nhỏ, rất vụn vặt nhưng chúng lại đóng vai trò quan trọng trong việc phân loại và nhận dạng vân tay. Các chi tiết đó ví dụ như đặc trưng rẽ nhánh hoặc một đường vân cụt (hình 6.6). Có thể có 50 đến 100 chi tiết dạng ấy trên một ngón tay. Trong thực tế người ta căn cứ vào vị trí và hướng của các chi tiết trong các vân tay để thực hiện nhận dạng. Có thể nói rằng, chỉ cần phát hiện 20 chi tiết là có thể nhận dạng được một vân ngón tay chưa biết và một vân ngón tay đã biết.

Hình 6.6. *Mô tả các chi tiết nhỏ của vân tay*

Việc phân tích các vân tay thu thập được tại các vị trí gây án hoặc tội phạm để kiểm tra nhận dạng chúng là công việc của các cơ quan chức năng cảnh sát. Ở nhiều nước, người ta xây dựng các kho lưu trữ vân tay lớn, còn gọi là "*tàng thư vân tay*" để có thể so sánh với các vân tay thu thập được cần so sánh. Các tàng thư đó có thể lưu trữ bằng phương pháp thủ công, phân loại, công thức hóa các đặc trưng và sắp xếp theo phương pháp Henry, hoặc xây dựng các hệ thống xử lý tự động hóa trên máy tính điện tử. Các hệ thông tin xử lý vân tay tự động hóa trên máy tính điện tử đó được gọi là các hệ AFIS (Automatic Finger Indentification System). Với các hệ AFIS hiện đại, một vân tay cần so sánh, có thể là không đầy đủ các đặc trưng, nhưng chỉ cần vài phút là có thể nhận dạng được với độ tin cậy khá cao. Công việc đòi hỏi không những kỹ thuật công nghệ mà cả trí tuệ và kinh nghiệm.

Trong những năm 1950 - 1960, người ta sử dụng các hệ thống xử lý vân tay quang học (phương pháp Holograph). Phương pháp sử dụng nguồn sáng laser để chiếu qua một tấm phim trong suốt có chứa dấu vân tay (microfilm chứa ảnh vân tay của 10 ngón tay - Hologram vân tay) để tạo ra ảnh biến đổi Fourier (phổ Fourier) sau đó thực hiện lọc so sánh trên mặt phẳng Fourier với ảnh

Holography. Sự nhận biết ảnh tiến hành qua việc so sánh các phổ. Phương pháp Holograph đơn giản và không sử dụng các thiết bị điện tử phức tạp nhưng có những hạn chế: sự phân bố sắp xếp dấu vân tay tiến hành một cách cơ học, khó tìm kiếm trong một tàng thư vân tay có số liệu lớn.

Trong những năm 1960 - 1970, các nghiên cứu về các hệ thống xử lý dấu vân tay theo kiểu ngôn ngữ cú pháp ra đời và được đưa vào ứng dụng ở một số nước. Phương pháp được xây dựng trên cơ sở là, dấu vân tay sau khi được đọc vào máy được chia thành nhiều vùng nhỏ để xử lý. Tại mỗi vùng, các đặc điểm như điểm cụt, điểm rẽ nhánh, đảo, đoạn ngắn... được chọn và mã hóa theo "*tập hợp ngôn ngữ hình cây*" (Set of tree languages). Sau đó việc so sánh vân tay được thực hiện theo "*kỹ thuật phân tích cú pháp tuần tự*" (Sequential Parsing technique).

Phân loại dấu vân tay là một công việc cần thiết trước tiên cho việc xây dựng các cơ sở dữ liệu dấu vân tay. Cho đến nay nhiều hệ thống cơ sở dữ liệu vân tay hiện đại vẫn được xây dựng trên cơ sở phân loại truyền thống của Henry (đề xuất năm 1900). Những năm đầu phương pháp Henry được xây dựng thủ công nhưng ngày nay đã được tự động hóa bằng máy tính tốc độ cao dung lượng lớn. Theo phương pháp Henry các dấu hoặc đặc điểm vân tay được biểu diễn thành công thức chứa các ký hiệu, số và chữ cái; nhờ đó mà các dấu vân tay được mã hóa và sắp xếp theo một trật tự nhất định trong tàng thư. Việc tìm kiếm do đó cũng nhiều thuận lợi và dễ dàng hơn, dù cho phải thực hiện bằng phương pháp thủ công.

Ví dụ vài quy ước về cách phân loại theo Henry:

- Cung → A, a
- Cung trồi → T, t
- Quai phải → / (cho ngón thuộc bàn tay trái)
  → \ (cho ngón thuộc bàn tay phải)
- Xoáy → W, w

Qui tắc đánh ký hiệu như sau:

- Ngón trỏ: các chữ cái hoa được dùng cho ngón này trừ trường hợp quai trái;
- Các ngón còn lại: Các ngón này được đánh ký hiệu là các chữ cái thường, trừ ngón có quai trái;
- Ngón có quai trái: Người ta sử dụng các gạch chéo như đã ký hiệu bên trên.

Dấu vân tay các ngón được phân loại theo nhóm 1 (có thể phân loại theo nhiều nhóm) và được gán theo mẫu sắp xếp như sau:

Phân loại theo nhóm 1:

	Cái	Trỏ	Giữa	Nhẫn	Út
Tay phải	16	16	8	8	4
Tay trái	4	2	2	1	1

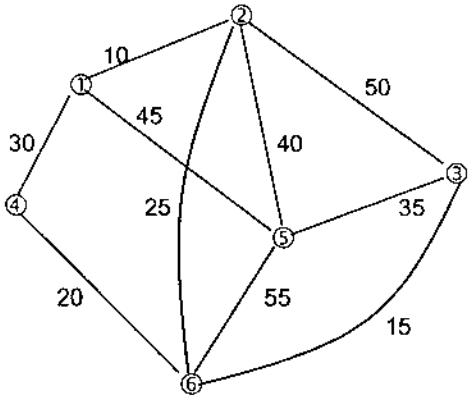
Phân loại theo nhóm 2:

- Tử số cho các ngón của bàn tay phải
- Mẫu số cho các ngón của bàn tay trái
- Đối với ngón trỏ: các chữ cái hoa được dùng như A, T, R, U và W.
- Đối với các ngón khác: Các chữ cái thường được dùng như a, t, r, u và w.

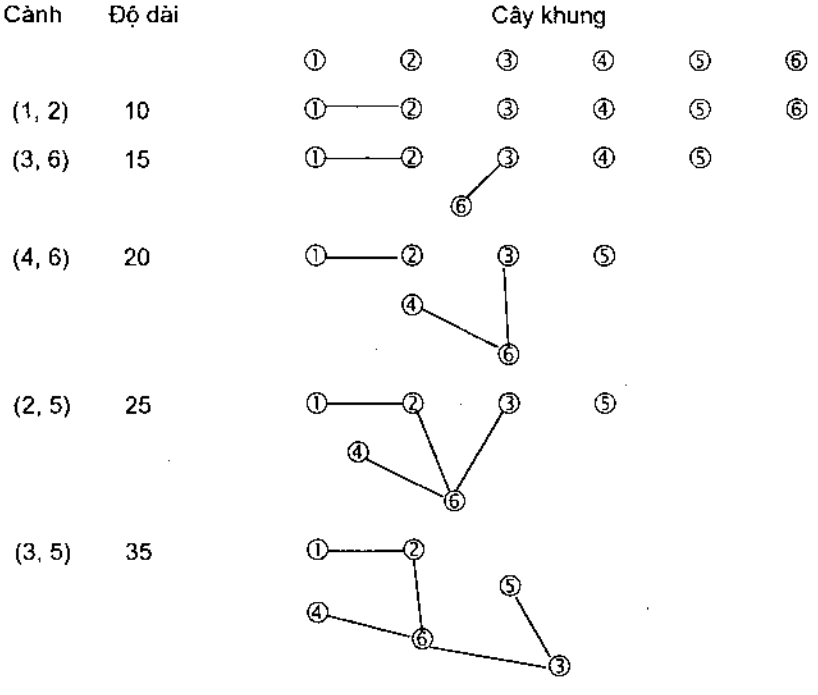
Ví dụ:

$$\frac{\text{tWaar}}{\text{tWaar}} \quad \text{hoặc} \quad \frac{\text{tW2ar}}{\text{tW2ar}}$$

Cho graph



Quá trình tạo cây khung như sau:

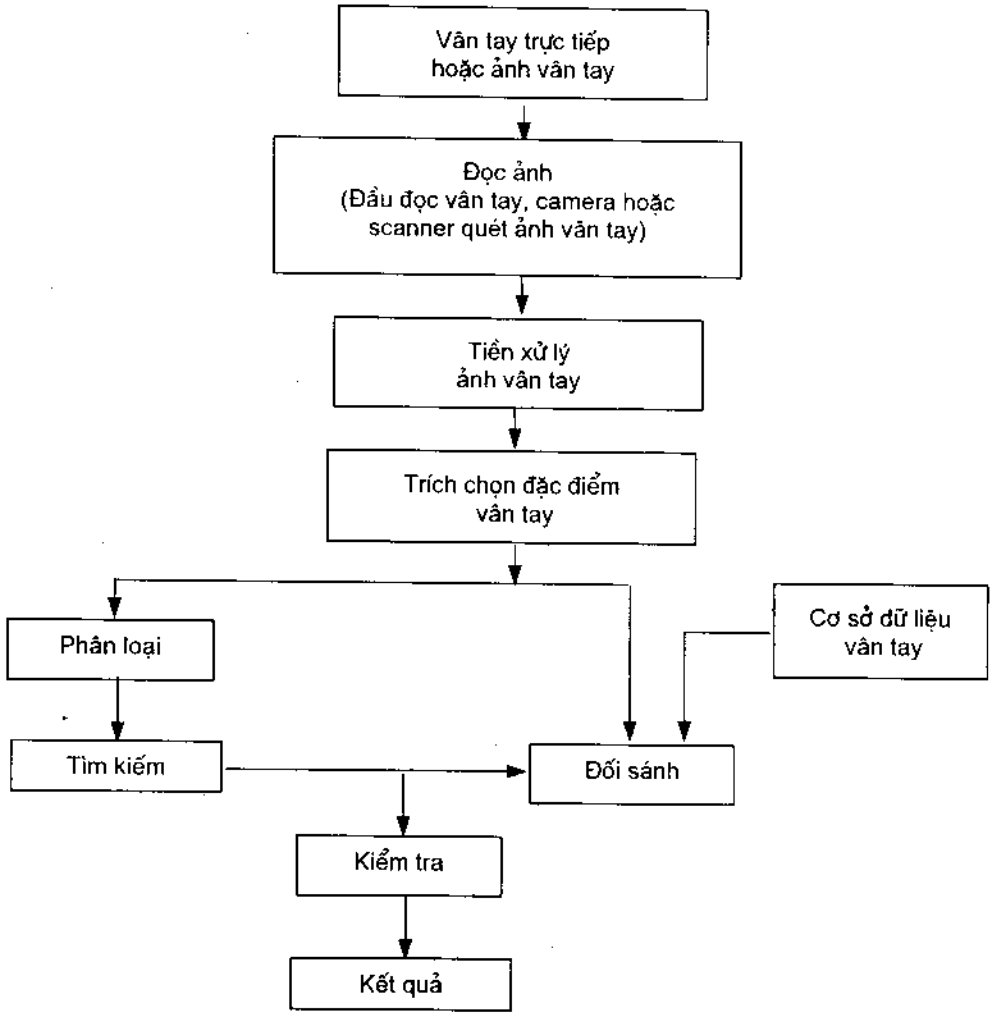


Hình 6.7. ***Mô tả ví dụ minh hoạ tạo cây cực tiểu***

Trong các thuật toán tự động hóa phân loại dấu vân tay để giảm thời gian tính toán, các khu vực có thể chứa các điểm đặc biệt (như tâm, delta) được tìm trước tiên. Đây là bước hạn chế vùng phải xử lý. Có thể có nhiều phương pháp xử lý khác nhau, ví dụ dựa vào dòng chảy của các đường vân, dựa vào các điểm đặc trưng hoặc topo của các điểm đặc trưng, xác định phân bố hướng, thuật toán tạo cây cực tiểu với các nút là các điểm đặc trưng...

Hình 6.7 mô tả ví dụ minh hoạ một quá trình tạo cây cực tiểu của thuật toán Kruskal với graph cho trước.

Hình 6.8 mô tả mô hình thuật toán đơn giản nhận dạng đường vân ngón tay xử lý tự động hóa trên máy tính.



Hình 6.8. ***Mô tả một thuật toán đơn giản nhận dạng vân tay***

6.4.5. Kiểm tra nhận dạng tiếng nói

Tiếng nói của con người là một biểu hiện rất nhạy cảm và thông qua nó có thể nhận biết một con người ngay cả trong trường hợp có tạp âm và nhận biết được nội dung ngữ nghĩa lời nói. Đã có khá nhiều công trình nghiên cứu về nhận dạng tiếng nói, kiểm tra nhận dạng tiếng nói.

Nhận dạng tiếng nói cũng đã được ứng dụng làm nhân chứng trong các vụ tranh chấp kiện tụng, trong giám định hình sự và trong giao tiếp người - máy, tự động hóa điều khiển.

Trong thực tế, có hai dạng nhận dạng tiếng nói, đó là:

- Nhận dạng tiếng nói của người nào nói;

- Nhận dạng ngữ nghĩa của tiếng nói, câu nói.

Hệ thống nhận dạng người nào nói được ứng dụng trong việc kiểm tra nhận dạng để phân biệt người này và người khác như các hệ thống nhận dạng vân tay, chữ ký.

Hệ thống nhận dạng ngữ nghĩa của tiếng nói được ứng dụng trong các hệ thống điều khiển xử lý tự động hóa giao tiếp người máy.

Về lý thuyết, hiện có ba khuynh hướng nhận dạng tiếng nói, đó là:

- Khuynh hướng âm học - ngữ âm học;

- Khuynh hướng nhận dạng mẫu;

- Khuynh hướng ứng dụng trí tuệ nhân tạo.

Khuynh hướng âm học - ngữ âm học dựa trên lý thuyết về âm học - ngữ âm học. Lý thuyết này cho rằng tồn tại các đơn vị ngữ âm xác định, có tính phân biệt trong lời nói và các đơn vị ngữ âm đó được đặc trưng bởi một tập các đặc tính tín hiệu tiếng nói. Mặc dù các đặc tính âm học của các đơn vị ngữ âm biến thiên rất lớn theo cả giọng người nói lẫn tác động của các đơn vị ngữ âm xung quanh (còn gọi là hiện tượng đồng phát âm), nhưng tồn tại các quy luật cho phép giải quyết vấn đề này. Bước đầu tiên trong nhận dạng tiếng nói tự động theo khuynh hướng âm học - ngữ âm học là phân đoạn và dán nhãn. Bước này chia tín hiệu tiếng nói thành các đoạn có đặc tính âm học đặc trưng cho một (và có thể vài) đơn vị ngữ âm, đồng thời gán cho mỗi đoạn âm thanh đó một hay nhiều nhãn ngữ âm phù hợp. Để có thể nhận dạng tiếng nói cần tiến đến bước thứ hai. Bước thứ hai tìm cách xác định một từ (hoặc một chuỗi từ) đúng trong số một chuỗi các nhãn ngữ âm được tạo ra sau bước thứ nhất, tuân thủ một số điều kiện ràng buộc (các từ được chọn ra trong một từ điển cho trước, các từ phải phù hợp với nguyên tắc ngữ pháp và có nghĩa...)

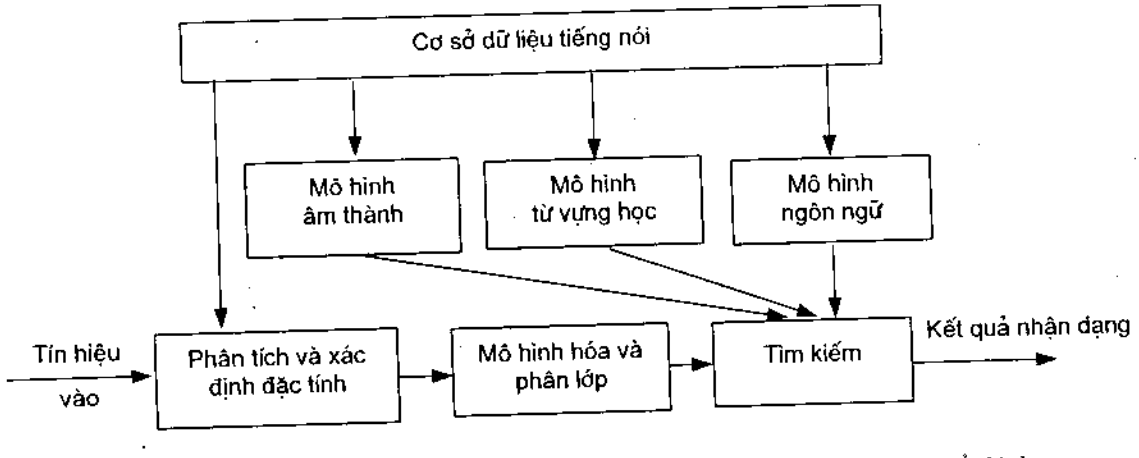
Nhận dạng tiếng nói theo khuynh hướng nhận dạng mẫu sử dụng trực tiếp các mẫu tín hiệu tiếng nói mà không phải xác định rõ ràng các đặc tính âm học (so với khuynh hướng âm học - ngữ âm học) và không phải phân đoạn tiếng nói. Các hệ thống nhận dạng tiếng nói theo khuynh hướng này, phải triển khai theo hai bước. Bước thứ nhất là huấn luyện hệ thống sử dụng một tập mẫu tiếng nói (cơ sở dữ liệu tiếng nói). "*Kiến thức*" về tiếng nói của hệ thống nhận dạng tiếng nói được tích lũy thông qua quá trình huấn luyện. Bước thứ hai là nhận dạng tiếng nói thông qua so sánh mẫu. Nguyên tắc cơ bản của khuynh hướng này là nếu cơ sở dữ liệu tiếng nói dùng cho huấn luyện có đủ các phiên bản của mẫu cần nhận dạng, quá trình huấn luyện có thể xác định chính xác các đặc tính âm học của mẫu (mẫu có thể là âm vị, từ hoặc cụm từ v.v. ..). Kỹ thuật nhận dạng mẫu là linh hồn của các hệ thống nhận dạng tiếng nói theo khuynh hướng này nên tên của kỹ thuật nhận dạng mẫu thường được đặt tên cho các hệ thống nhận dạng tiếng nói theo khuynh hướng này. Các kỹ thuật nhận dạng mẫu đơn giản (lượng tử hóa véc-tơ, hiệu chỉnh thời gian động, hiệu chỉnh thời gian tuyến tính...) thường chỉ được áp dụng cho các hệ thống nhận dạng tiếng nói có các từ rời rạc và từ điển cỡ nhỏ. Hiện tại có hai kỹ thuật nhận dạng mẫu được ứng dụng khá thành công trong nhận dạng tiếng nói liên tục, đó là mô hình Markov ẩn (HMM) và mô hình mạng nơron (NN).

Nhận dạng tiếng nói theo khuynh hướng ứng dụng trí tuệ nhân tạo là sự lai ghép giữa khuynh hướng âm học - ngữ âm học với khuynh hướng nhận dạng mẫu vì nó khai thác ý tưởng của cả hai khuynh hướng đó. Nhận dạng tiếng nói theo khuynh hướng ứng dụng trí tuệ nhân tạo tự động hóa thủ tục nhận dạng theo cách mà con người áp dụng trí tuệ của mình để hình tượng hoá, phân tích và đưa ra các quyết định về các âm thanh thu nhận được. Trong thực tế, các kỹ thuật nhận dạng tiếng nói theo khuynh hướng này chủ yếu ứng dụng một hệ chuyên gia cho công đoạn phân đoạn và gán nhãn sao cho công việc khó khăn nhất này có thể thực hiện không chỉ nhờ các thông tin âm học - ngữ âm học (ý tưởng của nhận dạng theo khuynh hướng âm học - ngữ âm học) mà nó còn học và phân biệt các mẫu âm thanh (ý tưởng của nhận dạng tiếng nói theo khuynh hướng mẫu). Một ví dụ đơn giản về hệ thống nhận dạng xác định tiếng nói của người nào nói:

Chọn một câu hoặc một chủ đề ngắn, gọn trong có 16 âm điển hình. Tránh các đồng âm. Các âm đặc trưng cho cả nguyên âm và phụ âm. Người được lưu trữ tiếng nói trong hệ thống đọc 32 lần chủ đề chuẩn được chọn đó để hệ thống lưu giữ. Bằng phương pháp phân tích phổ của Fourier, hoặc thiết bị phân tích phổ, với tần số lấy mẫu là 10 ms, để ghi nhận các vùng có biên độ lớn trong miền tần số và phân bố của phổ. Năng lượng phổ tập trung nhiều trong dải tần từ 300 đến 2500 Hz và có các mẫu tham chiếu. Khi cần kiểm tra nhận dạng tiếng nói của một người nào đó thì cũng đọc các câu chuẩn và so sánh với các mẫu tham chiếu của phổ đã được lưu trữ với một sai số cho phép.

Nhược điểm của hệ thống nhận dạng tiếng nói trên là phải có một thời gian nào đó để đọc các câu chuẩn và phải có thời gian để phân tích. Ngoài ra tiếng nói còn phụ thuộc vào trạng thái vật lý của con người như thở gấp, sợ sệt, buồn chán, tức giận... có sự biến đổi khác nhau.

Hình 6.9 mô tả các phần tử cơ bản của một hệ thống nhận dạng tiếng nói điển hình.



Hình 6.9. ***Các phần tử cơ bản của một hệ thống nhận dạng tiếng nói điển hình***

Biết rằng, tiếng nói được hình thành thông qua một loạt các tác động sinh học được điều phối trong cơ quan phát âm của con người. Sóng âm thanh bao gồm các xung được lan truyền trong thanh quản; Cộng hưởng khi đi qua khoang hầu, khoang miệng hoặc khoang mũi (gọi chung là khoang thanh của ống thanh) và thoát ra ngoài. Các khoang cộng hưởng đó có sự khác nhau về hình dáng, thể tích, lối thoát không khí... do đó chúng tạo ra các âm khác nhau của một ngôn ngữ.

Tần số cộng hưởng ở các khoang thanh của ống thanh, trong kỹ thuật nhận dạng tiếng nói gọi là *tần số formant*, hay gọi tắt là *formant*. Bởi vì hình dáng của ống thanh luôn biến đổi theo thời gian cho nên có một phổ tiếng nói luôn biến đổi theo thời gian. Căn cứ vào dạng phổ và các đặc tính biến đổi của phổ để nhận dạng tiếng nói là một hướng quan trọng trong kỹ thuật nhận dạng tiếng nói.

Cũng giống như các phổ tín hiệu điện khác, các ảnh phổ tiếng nói cũng được biểu thị dưới dạng các tham số về năng lượng (biên độ), tần số và thời gian.

Quá trình nhận dạng tiếng nói có sự tham gia của các máy tính số bao gồm các khâu: chuyển tiếng nói thành tín hiệu điện, lọc nhiễu, biến đổi thành dạng số (A/D, lấy mẫu, lượng tử và mã hóa), phân tích lựa chọn đặc trưng và xử lý so sánh nhận dạng.

Các phương pháp nhận dạng có thể sử dụng mô hình Markov ẩn, HMM (Hidden Markov Modell); mạng nơron hoặc ứng dụng trí tuệ nhân tạo.

Mô hình Markov ẩn là một công cụ toán học mạnh có thể sử dụng để mô hình hóa cấu trúc và tính biến đổi theo thời gian của tín hiệu tiếng nói. Phương pháp này mô hình một chuỗi các mẫu tiếng nói như đầu ra của một quá trình ngẫu nhiên.

Mô hình được gọi là ẩn bởi vì không xác định được chuỗi trạng thái tạo ra một chuỗi mẫu quan sát cho trước. Về nguyên tắc luôn tồn tại khả năng chuyển từ trạng thái này sang trạng thái khác; song để giảm bớt độ phức tạp khi tính toán người ta thường hạn chế số bước chuyển và hướng chuyển. Để tạo được một mô hình tốt, cần một số lượng lớn các mẫu để huấn luyện.

Mạng nơ-ron cũng là một công cụ nhận dạng có hiệu quả, do vậy cũng đã có nhiều hệ thống ứng dụng mạng nơ-ron vào việc nhận dạng tiếng nói. Một dạng mạng nơ-ron được ứng dụng tương đối phổ biến là mạng nơ-ron có cấu trúc perceptron nhiều lớp.

Ý tưởng cơ bản của việc ứng dụng trí tuệ nhân tạo (AI) vào nhận dạng tiếng nói là thu thập kiến thức từ các nguồn kiến thức khác nhau để giải quyết vấn đề đặt ra. Ví dụ, ứng dụng trí tuệ nhân tạo để làm công đoạn phân đoạn và gán nhãn tiếng nói trong đó bao gồm ngữ âm, cú pháp, ngữ nghĩa v.v... Trong thực tế, hầu hết các kỹ thuật nhận dạng theo khuynh hướng này đều sử dụng các hệ chuyên gia cho công đoạn phân đoạn và gán nhãn các tín hiệu âm thanh.

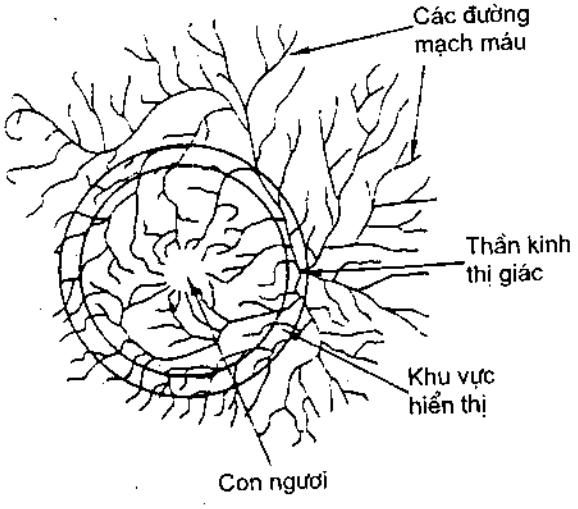
Tiếng Việt có những đặc điểm riêng không giống những ngôn ngữ đa âm và cho đến nay cũng đã có nhiều công trình nghiên cứu được công bố.

6.4.6. Kiểm tra nhận dạng dựa vào võng mạc

Đã từ lâu, người ta cũng đã biết rằng, các đường mạch máu li ti trên võng mạc mắt người mang đặc tính hoàn toàn cá nhân. Có thể dựa vào các đường mạch máu trên võng mạc đó để thực hiện kiểm tra nhận dạng. Phương pháp được thực hiện như sau.

Sử dụng một thiết bị tia hồng ngoại cường độ yếu để chụp hình các đường mạch máu trên võng mạc. Ảnh thu được, hoặc được hiển thị, như mô tả ví dụ ở hình 6.10, trong đó có các nút và các nhánh của các đường mạch máu.

Cũng giống như các phương pháp kiểm tra nhận dạng đã được phân tích ở vân tay, có thể chọn các điểm đặc trưng, lưu giữ và so sánh khi cần kiểm tra nhận dạng. Các thuật toán được thực hiện như các thuật toán sử dụng ở các hệ thống xử lý tự động hóa vân tay.



Hình 6.10. *Mô tả ví dụ một võng mạc với các đường mạch máu*

Chương 7

XÁC THỰC ĐỒNG NHẤT VÀ BẢO TOÀN DỮ LIỆU

7.1. Ý NGHĨA CỦA XÁC THỰC ĐỒNG NHẤT VÀ BẢO TOÀN DỮ LIỆU

Các dữ liệu, một đoạn tin hoặc một tài liệu được xác thực khi mà chúng hoàn toàn đúng với nguyên bản xuất xứ và đến từ nguồn được thừa nhận. Với một nghĩa rộng, có thể nói đó là nguồn hợp pháp để truyền các đoạn tin đó. Một câu hỏi đặt ra là, liệu các đoạn tin đó đã được xác thực? Thực ra, khái niệm về xác thực (authentify) rất rộng. Trong chương, chỉ giới hạn chủ đề về xác thực ở các dạng sau đây:

- Một đoạn tin được truyền từ A đến B qua một mạng truyền tin. A và B có thể là những con người cụ thể, là những thể thức truyền tin và cũng có thể là một "thực thể" trong sự phân cấp các thể thức. Mong muốn là B nhận được đoạn tin đích xác là của A gửi, nó không bị làm thay đổi nội dung trong quá trình truyền từ A đến B, hoặc nói cách khác đó đích thực là đoạn tin A gửi cho B.

- Một đoạn tin được A ghi vào bộ nhớ, sau đó được đọc bởi A hoặc bởi một thực thể B khác. Mong muốn rằng, đoạn tin được đọc đó là đích thực của A đưa vào bộ nhớ và không bị làm thay đổi nội dung bởi một người nào đó trước khi đọc. Bởi vì các dữ liệu, trong một số trường hợp, có thể được thay đổi theo phương thức hợp pháp nhưng người đọc cần biết thực thể xuất phát và các biến đổi sau đó. Về vấn đề này thường người ta đánh số phiên bản trong các dữ liệu đã lưu giữ để đánh dấu các thay đổi.

- Hai thực thể A và B có sự trao đổi nhau các đoạn tin, từ A đến B hoặc từ B đến A. Để cho cuộc trao đổi được đích xác thì A cần phải kiểm tra nhận dạng B và B cần phải kiểm tra nhận dạng A; nội dung của đoạn tin trao đổi phải đích thực là của phía bên đối tác gửi.

Việc xác thực là một thủ tục đảm bảo sự chính xác của mỗi một trong các tình huống đề cập trên. Ở đây thuật ngữ "*mã xác thực*" mô tả một khối dữ liệu được tạo ra theo một phương pháp riêng nhằm phục vụ các yêu cầu trên. Ở đây cần phân biệt khái niệm *mật mã các thông tin* với khái niệm *xác thực*. Mật mã là để bảo vệ chống lại sự xâm nhập thụ động còn xác thực là để bảo vệ chống lại các xâm nhập tích cực.

Trong thuật ngữ của OSI, có sự phân biệt giữa *xác thực* (authentication) và *bảo toàn dữ liệu* (data integrity). Xác thực mô tả các phương thức mà các đối tác truyền tin sử dụng để kiểm tra nhận dạng lẫn nhau. Khi mà sự truyền tin giữa hai bên có sự thiết lập phối hợp với nhau trước thì việc xác thực là đồng cấp (hoặc ngang hàng - peer entity authentication). Trong trường hợp đoạn tin truyền

không cần phải chờ trả lời thì việc xác thực chỉ là xác thực nguồn xuất xứ của dữ liệu. Công việc bảo mật là bảo vệ chống lại các thay đổi dữ liệu bất hợp pháp hoặc phá hoại dữ liệu và được gọi là sự bảo toàn dữ liệu. Xác thực và bảo toàn dữ liệu đều là đảm bảo an toàn truyền tin chống lại các xâm nhập tích cực. Tuy vậy, việc tách riêng như vậy có cái lợi trong thực tế ứng dụng. Trong thực tế ít khi cùng một lúc phải xác thực và thực hiện đảm bảo tính bảo toàn dữ liệu và thuật ngữ xác thực nhiều lúc được hiểu là tất cả các bảo vệ cần thiết chống lại xâm nhập tích cực.

Các xâm nhập tích cực phức tạp hơn nhiều so với các xâm nhập thụ động, bởi vì có rất nhiều phương pháp để làm sai lạc, hư hỏng dữ liệu. Trong số các phương pháp đó có thể kể ra một số là:

- Làm sai lệch, rút bớt hoặc thêm vào;
- Thay đổi nội dung gốc;
- Thay đổi bên nhận;
- Làm sai lệch trật tự các khối hoặc trật tự các dữ liệu;
- Sử dụng lại các dữ liệu đã truyền trước;
- Giả mạo các đoạn tin.

Ở đây cần lưu ý rằng, việc sử dụng các phương pháp mật mã không bao gồm việc chống lại việc kẻ gian trá chèn thêm dữ liệu vào. Phần lớn các xâm nhập trên, về nguyên lý, có thể thực hiện được bằng cách sử dụng các dữ liệu xâm nhập đã được mã hoá.

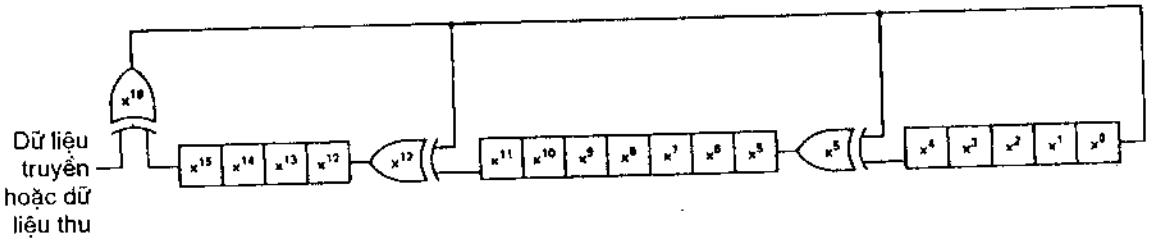
7.2. BẢO VỆ ĐỐI VỚI CÁC LỖI NGẪU NHIÊN TRONG TRUYỀN DỮ LIỆU

Trong tất cả các hệ thống truyền tin, các bên nhận có yêu cầu là đảm bảo các đoạn tin nhận được không bị sai hoặc lỗi kể cả vô tình hoặc cố ý. Các lỗi ngẫu nhiên của một đoạn tin truyền trên kênh có thể được phát hiện và sửa lỗi bằng cách yêu cầu truyền lại hoặc bằng cách áp dụng mã sửa lỗi. Trong kỹ thuật truyền tin, có rất nhiều dạng mã sửa lỗi khác nhau. Tuỳ thuộc vào dạng mã lỗi xuất hiện mà có thể chọn ứng dụng loại mã sửa lỗi thích hợp.

Một loại mã sửa lỗi thường được ứng dụng phổ biến để phát hiện và sửa lỗi theo phương pháp *kiểm tra độ dư vòng*, (CRC - Cyclic Redundancy Check). Mã hoạt động theo phương thức xử lý các dãy nhị phân như các đa thức và "chia" khối dữ liệu truyền cho một đa thức cố định trước được gọi là *đa thức sinh*. "Số dư" của *phép tính đại số đa thức đồng dư* đó được sử dụng như một trường kiểm tra. Ở bên thu, cũng sử dụng phép chia đa thức đồng dư đó và số dư nhận được của phép toán được đem so sánh với trường kiểm tra nhận được. Nếu như hai giá trị đó là bằng nhau thì lỗi truyền sẽ được bỏ qua. Khuyến nghị V41 của CCITT có một phiên bản chuẩn hoá quá trình đó với mã sửa lỗi CRC - 16 bít cho các đường truyền tin. Phiên bản đó được sử dụng trong các thể thức đường truyền như HDLC. Mã sửa lỗi CRC đó có thể phát hiện tất cả các lỗi một hoặc hai bit, tất cả các lỗi có số lẻ bit, tất cả các gói lỗi có chiều dài dưới hoặc bằng 16 bit và với độ chính xác 99,997% cho các gói có chiều dài lớn hơn 16 bit. Hình 7.1 mô tả thuật toán tạo mã vòng sửa lỗi CRC-16 trong đó đa thức sinh $P(x) = x^{16} + x^{12} + x^5 + 1$ theo khuyến nghị V41.

Mã CRC-16 thích hợp cho việc sửa các lỗi bit đơn và kết cấu mã khối. Để sửa lỗi cụm bit hoặc nhóm bít có thể sử dụng các loại mã xoắn. Mã sửa lỗi cũng có thể kết hợp với mã bảo mật như giới thiệu ở chương 4.

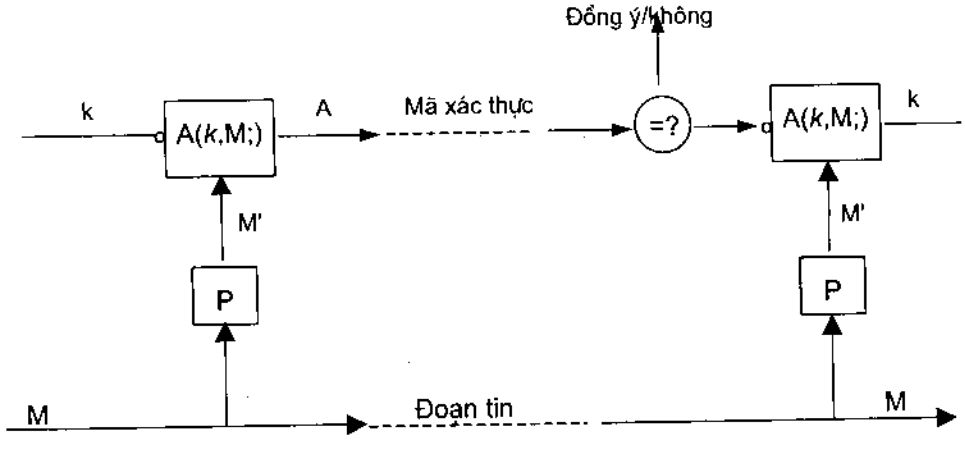
Các mã sửa lỗi có thể bảo vệ chống lại tất cả các lỗi ngẫu nhiên do tạp nhiễu của kênh truyền tin gây nên nhưng không thể bảo vệ được dữ liệu truyền chống lại xác xâm nhập tích cực. Để chống các xâm nhập tích cực trong trường hợp này, có thể kết hợp mã sửa lỗi với mật mã hóa dữ liệu. Phương pháp sử dụng các mã sửa lỗi là công khai, trong đó bao gồm cả các hệ số của đa thức mã sửa lỗi. Do đó, kẻ xâm nhập có thể dễ dàng làm giảm đoạn tin tuỳ ý bằng cách tính toán và thay đổi trường kiểm tra. Trong trường hợp này phía bên nhận không thể nhận biết được đó là đoạn tin giả nếu không có sự xác thực.



Hình 7.1. *Mô tả sơ đồi khối nguyên lý bộ tạo mã sửa lỗi CRC-16 theo khuyến nghị CCITT-V41.*

7.3. BẢO TOÀN DỮ LIỆU DỰA TRÊN CÁC THAM SỐ MẬT

Nhược điểm của các phương pháp xác thực dựa trên các tham số không mật là bất kỳ kẻ xâm nhập nào cũng có thể làm giả mạo các dữ liệu trong trường hợp kiểm tra của đoạn tin có mã sửa lỗi. Do đó cần phải nhờ cậy đến sự tham gia của một khoá mật, và khoá mật đó chỉ có bên gửi và bên nhận biết, không một người thứ ba nào có thể truy nhập. Trong trường hợp này sẽ trở lại vấn đề phân phối khoá mã như đã phân tích ở chương 3. Hình 7.2. mô tả sơ đồ khối nguyên lý hoạt động tổng quát phương pháp bảo toàn dữ liệu sử dụng khoá mật.



Hình 7.2. *Mô tả sơ đồ khối nguyên lý kiểm tra bảo toàn dữ liệu*

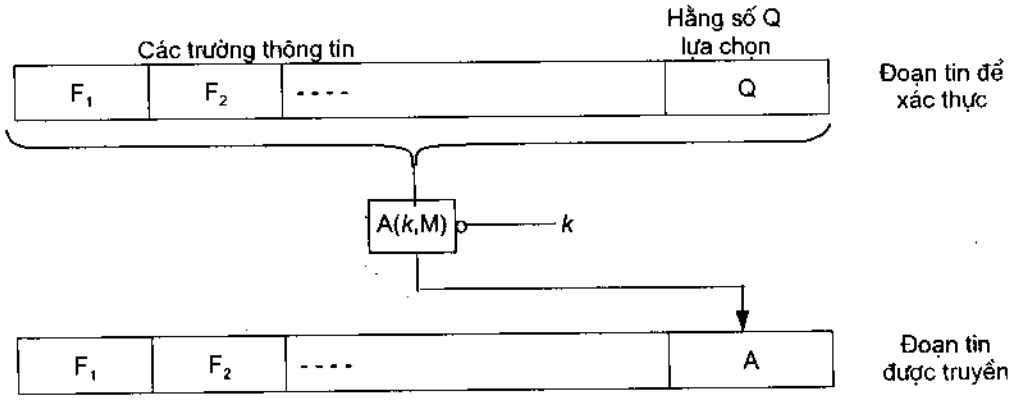
Hoạt động của phương pháp dựa trên một thuật toán công khai được đặc trưng bởi hàm A(k,M). Cả bên gửi và bên nhận đều biết khoá mật k, ngoài ra không ai biết. Đoạn tin cần nhận dạng M có thể có chiều dài bất kỳ. Giá trị của hàm A được gọi là "*mã xác thực*" MAC (message Authentication Code), cùng với đoạn tin từ bên gửi đến bên nhận. Bên nhận cũng sẽ tính giá trị

A(k,M) và so sánh giá trị đó với giá trị A nhận được. Nếu các kết quả đó là bằng nhau có nghĩa là đoạn tin được chấp nhận.

Các mã xác thực cũng đã từng được sử dụng cho các đoạn tin gửi qua đường truyền tin télex và thời đó các phép tính phải xử lý theo phương pháp thủ công và thuật toán dựa trên các bảng tính.

Để đơn giản thủ tục, có thể một phần của các trường được đưa vào thuật toán. Giải pháp đó giống như vấn đề "tiền xử lý" đoạn tin đặc trưng bởi chức năng P trên hình. Nếu như đoạn tin được truyền bởi hệ thống có thể thay đổi phép mã hoá thì việc tiền xử lý chỉ cần bảo toàn những phần tử thiết yếu và không biến đổi. Ví dụ, một hệ thống truyền tin hoặc hệ thống nhớ có thể xử lý theo phương pháp nhảy đòng.

Kích thước của trường mã xác thực được xác định bởi hàm A(k, M). Nó nằm trong khoảng 16 và 32 bit và nói chung là rất ít hơn so với đoạn tin và thường được bổ sung them một giá trị Q đã biết để tính mã xác thực. Hình 7.3 mô tả nguyên lý thực hiện phương pháp đó (đoạn tin được chứa trong các trường F_1, F_2,..., F[n]. Nếu như hằng số Q đó chỉ có bên gửi và bên nhận biết thì nó trở thành một phần của khoá mã.



Hình 7.3. *Mô tả sử dụng hằng số Q trong trường mã xác thực.*

Ở đây, các khoá cho các đoạn tin cần phải khác nhau, có nghĩa là phải sử dụng hai giá trị khác nhau cho hai mã xác thực A(k_1, M) và A(k_2, M) và chúng có liên quan chặt chẽ với nhau. Cũng giống như trong thuật toán mật mã, một trong những phương pháp của kẻ xâm nhập là bằng mọi cách tìm ra khoá mã. Mỗi lần đọc đoạn tin hoặc mã xác thực của nó thì kẻ xâm nhập cũng có số liệu đó nhưng để tìm ra mã xác thực thì còn phải khó khăn hơn việc khám phá khoá một thuật toán mật mã, bởi vì hai lý do sau đây:

Một là, giá trị của mã xác thực là yếu, nó có chiều dài trong khoảng 16 đến 32 bit. Nếu như làm phép kiểm tra tất cả các giá trị của khoá, và ví dụ như chiều dài của khoá là 56 bit thì khoá đó của mã xác thực là 24 bit, đoạn tin thứ nhất được kiểm tra khoảng 2^{32} khả năng khoá. Với đoạn tin thứ hai thì giá trị đó giảm xuống còn 2^8 số khả năng và đến đoạn tin thứ ba thì nó giảm xuống còn duy nhất một khả năng để có thể tìm ra khoá đã được sử dụng.

Hai là, để kiểm tra một khoá phải can thiệp vào toàn bộ đoạn tin. Giả thiết ví dụ đoạn tin là 640 bit được mã hoá bằng mật mã CBC theo thuật toán mật mã DES và mã xác thực cũng dựa trên phương pháp đó. Có 10 khối đoạn tin. Lúc đó việc tìm khoá trong số 2^{56} khả năng khoá có thể thì

phải thực hiện 2^{55} phép kiểm tra. Còn một vấn đề nữa là giá trị khởi đầu IV chưa biết. Bởi vì chiều dài của khối tin trong DES là 64 bít cho nên khoá mã sử dụng là 56 bit. Trong trường hợp mã xác thực, phải thực hiện trung bình 2^{55} phép thử và mỗi phép thử đó đều phải thực hiện với các phép tính của mật mã DES. Số 10.2^{55} phép tính để tìm khoá là một công việc hoàn toàn khó khăn. Tiếp tục còn phải 10.2^{31}, sau đó 10.2^{7} phép tính để tiếp tục thử cho khối tin thứ hai và khối tin thứ ba. Rõ ràng là một công việc tốn nhiều công sức bỏ ra, có thể nói là khó thực hiện được để tìm khoá mã trong trường hợp xác thực có kết hợp mật mã.

7.4. THUẬT TOÁN DỊCH CHUYỂN VÀ CỘNG THẬP PHÂN

Có một phương pháp để xác thực các đoạn tin dựa trên cơ sở là bên gửi và bên nhận sở hữu hai số mật và mỗi số mật đó có mười số thập phân. Các số mật đó là b_1 và b_2 được xem như là khoá của thuật toán. Các số mật đó cung cấp điểm xuất pháp cho hai phép tính được tiến hành song song. Đoạn tin được xác thực được xem là một dãy các số thập phân và nó cho phép xử lý phần số trong các đoạn tin trao đổi, ví dụ các trao đổi về thanh toán, kế toán, giá trị hợp đồng, v.v... Với phần chữ cái có thể sử dụng hai số cho việc mã hoá mỗi ký tự. Đoạn tin được chia thành các khối có mười số thập phân. Đoạn tin cuối cùng thiếu sẽ bổ sung thêm các số 0 ở bên phải để cho đủ mười số thập phân. Thuật toán làm việc theo bài toán đại số thập phân, có thể sử dụng một máy tính bỏ túi có dung lượng tối thiểu mười chữ số. Phương pháp thuật toán được gọi là "*dịch chuyển và cộng thập phân*" (Decimal shift and Add).

Các khối đoạn tin cần xác thực được xếp khối này sau khối kia và hai phép toán được thực hiện song song. Với hai phép toán được thực hiện song song đó, hai đầu ra mười số thập phân là Z_1 và Z_2 được tổng hợp lại để tạo thành mã xác thực của đoạn tin.

Phép toán là một sự phối hợp "*dịch chuyển và cộng*", và được thực hiện theo phương pháp sau đây: Cho D là một số có mười số thập phân và x cũng là một số thập phân. R(x)D là kết quả của một phép chuyển vòng của D với *x* vị trí về phía phải.

Ví dụ: R(3){1234567890} = 8901234567

Hàm "dịch chuyển và cộng" được định nghĩa là:

$$S(x)D = R(x)D + D \text{ (modulo } 10^{10})$$

Như vậy, với ví dụ trên có:

D = 1234567890

R(3)D = 8901234567

Bởi vì ở đây là thực hiện với phép toán modulo, cho nên giá trị của vị trí thứ mười là không biết.

Bây giờ, nếu giả thiết xác thực một đoạn tin có hai khối là m_1 = (1583492637) và m_2 = (5283586900). Chọn các giá trị b_1 và b_2 là:

$b_1 = 5236179902$ $\qquad$ $b_2 = 4893524771$

Như vậy, hai dãy biến đổi bắt đầu bởi phép cộng m_1 với b_1 và m_1 với b_2 modulo 10^{10} là:

	Dãy 1		Dãy 2
m_1	1583492637	m_1	1583492637
$+ b_1$	5236179902	$+ b_2$	4893524771
Kết quả p	6819672539	Kết quả q	64777017408

Kết quả cho các giá trị trung gian p và q và sẽ được sử dụng cho phép toán dịch chuyển và cộng. Với dãy 1 cho số vị trí dịch chuyển của p là 4 và với dãy 2 cho số vị trí dịch chuyển của q là 5. Như vậy có:

p	= 6819672539	q	= 6477017408
R(4)p	= 2539681967	R(5)q	= 1740864770
S(4)p	= 9359354506	S(5)q	= 8217882178

Gọi r và s là kết quả của phép tính cho dãy 1 và dãy 2 trên, bây giờ cộng tiếp đoạn tin m_2 vào cho mỗi dãy, sẽ có:

r	= 9359354506	s	= 8217882178
$+ m_2$	= 5283586900	$+ m_2$	= 5283586900
Kết quả u	= 4642941406	Kết quả v	= 3501469078

Đến đây, chúng ta có kết quả trung gian thứ hai là u và v. Thực hiện tiếp tục phép toán thứ hai về "dịch chuyển và cộng". Với dãy 1, số dịch chuyển được xác định bởi số thứ hai của b_2 (ở đây là 8) và với dãy 2, số dịch chuyển được xác định bởi số thứ hai của b_1 (ở đây là 2). Như vậy có:

u	= 4642941406	v	= 3501469078
R(8) u	= 4294140646	R(2) v	= 7835014690
S(8) u	= 8937082052	S(2) v	= 1336483768

Ở đoạn này của phép tính, các dữ liệu của đoạn tin đã hoàn toàn được sử dụng và đến đây có thể tính toán được mã xác thực. Gọi z_1 và z_2 là kết quả của hai phép tính trung gian trên. Thuật toán được thực hiện bởi phép cộng z_1 với z_2 modulo 10^{10}.

z_1	8937082052
$+ z_2$	1336483768
Kết quả	0273565820

Mã xác thực có mười con số nhưng để giảm kích thước của mã ở đây chỉ sử dụng sáu con số cuối và gập bốn số trước lại theo phương thức sau:

	565820
	0273
Kết quả	566093

Giá trị cuối cùng đó chính là mã xác thực. Ví dụ đã cho trên giải thích phương pháp thực hiện.

Thuật toán thực hiện theo các quá trình trên rất khó khám phá. Nếu như tất cả các phép toán chỉ được thực hiện với modulo 10^{10} thì việc thực hiện bài toán ngược không mấy khó khăn, nhưng ở đây có sự dịch chuyển vòng mà sự dịch chuyển đó không có quan hệ với phép toán. Điều đó cũng giống như các phép toán modulo $10^{10} - 1$, bởi vì mỗi lần một số d được dịch chuyển vòng thì giá trị sẽ bị thay đổi là $(10^{10} - 1)d$. Mặt khác, ở đây hai phép toán được thực hiện song song. Ngay cả khi kết quả của mười số được biết thì cũng rất khó mà khám phá.

Thuật toán dịch chuyển và cộng thập phân rất thích hợp với các máy tính thập phân có lập trình. Nó cũng được xem như một chuẩn quốc tế, được ứng dụng cùng với việc bảo toàn đoạn tin và thường được ứng dụng trong các hệ thống ngân hàng.

7.5. THUẬT TOÁN XÁC THỰC ĐOẠN TIN MAA

Thuật toán xác thực đoạn tin MAA (Message Authentication Alogorithm) về nguyên lý cũng giống như thuật toán dịch chuyển và cộng thập phân, chỉ có khác ở đây là các phép tính được thực hiện trên các số nhị phân. Phép toán "dịch chuyển và cộng" của thuật toán được giới thiệu ở mục 7.4 có thể so sánh với một phép nhân modulo $10^{10} - 1$ trong đó số nhân được biểu thị bởi các số nhị phân 10...01.

Mô hình dựa trên phép nhân có hai nguy hiểm, đó là:

- Nếu có kết quả nhận giá trị 0 (hoặc ngược lại;

- Nếu như modul là một số tổng hợp thì toàn bộ số chia của modul xuất hiện trong kết quả trung gian sẽ xuất hiện trong tất cả số dư của phép tính.

Để tránh nhược điểm thứ nhất, các số nhân được cố định một số bít 0 hoặc 1; để tránh nhược điểm thứ hai, một trong những phép tính song song được thực hiện theo modulo $2^{32} - 1$ và phép tính kia thực hiện theo $2^{32} - 2$. Số thứ hai của các số đó là $2(2^{31} - 1)$ và $2^{31} - 1$.

Hình 7.4 mô tả chu trình chính của phép toán mã xác thực.

```
        V:= CYC(V);
        D:= XOR(V,W);                                  V và W được lấy từ khoá mã
X:= XOR(X, Mi);          Y:= XOR(Y, Mi);               Mi là một khối đoạn tin
F:= ADD(E, Y);           G:= ADD(E, X);
F:= OR(F, A);            G:= OR(G, B);                 A, B, C, D là các hằng số
F:= AND(F, C);           G:= AND(G, D);
X:= MUL1(X, F);          Y:= MUL2A(Y, G);              Phép nhân modulo
```

Hình 7.4. ***Mô tả chu trình chính của thuật toán MAA***

Trên hình 7.4, hai dãy phép tính được mô tả cạnh nhau. CYC(V) là quay vòng của V một bit về trái. MUL1 là phép nhân modulo $2^{32} - 1$ và MUL2 là phép nhân modulo $2^{32} - 2$. Các giá trị A, B, C, D là các hằng số dùng cho các giá trị của các số nhân F và G. Phép toán ADD được thực hiện theo modulo 2^{32} và các phép toán logic được thực hiện song song trên 32 bit. Mỗi số nhân được tạo ra từ dãy kia của các phép tính bằng cách sử dụng giá trị E, lấy từ V và W, chúng thay đổi cho mỗi chu trình.

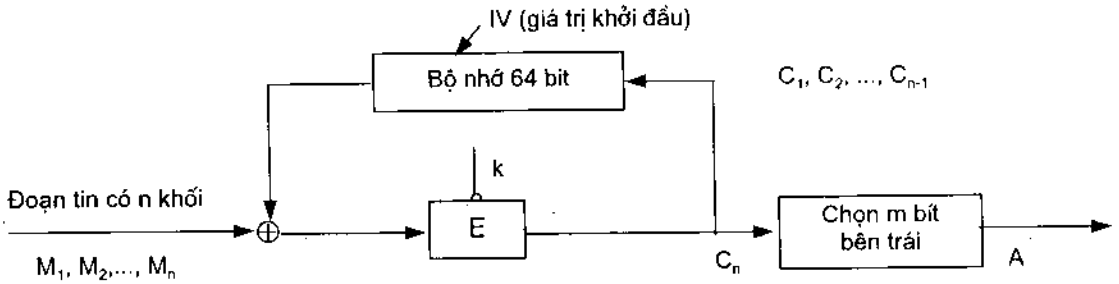
Cũng giống như ở thuật toán dịch chuyển và cộng thập phân, mã xác thực cuối cùng được tạo ra từ hai dãy phép tính bằng cách cộng modul 2 (XOR) các kết quả của các phép tính cuối cùng. Kích cỡ của mã xác thực là 32 bit.

7.6. XÁC THỰC DỰA TRÊN PHƯƠNG PHÁP MẬT MÃ CFB VÀ MẬT MÃ CBC

Có hai phương pháp xác thực dựa trên phương pháp mật mã CFB và mật mã CBC (về mật mã CFB và mật mã CBC đã được giới thiệu trong chương 2).

Ở phương pháp sử dụng mật mã CFB, phép toán được thực hiện trên các ký tự có kích cỡ thích hợp, ví dụ các ký tự 8 bit nếu sử dụng mã ASCII. Ở phương pháp sử dụng mật mã CBC, phép toán luôn luôn thực hiện trên các khối 64 bit, và khối cuối cùng được chèn thêm các bit 0 cho đủ 64 bit. Việc lựa chọn sử dụng phương pháp mật mã nào là tuỳ thuộc vào biểu hiện của các dữ liệu.

Để tạo ra một mã xác thực theo phương pháp mật mã CBC người ta mã hoá đoạn tin để xác thực như một chuỗi các khối 64 bit, và chèn thêm các số 0 vào khối cuối cùng nếu khối đó thiếu bit. Hình 7.5 mô tả sơ đồ khối chức năng hoạt động của xác thực theo phương pháp sử dụng mật mã CBC. Tất cả các khối đã mã hoá, trừ khối cuối cùng, chỉ được sử dụng trong quá trình phản hồi. *m* bit có trọng số lớn nhất của khối cuối cùng C_n được sử dụng để cung cấp mã xác thực. Mã xác thực như vậy còn có tên gọi là *mã xác thực đoạn tin*, MAC (Message Authentication Code).



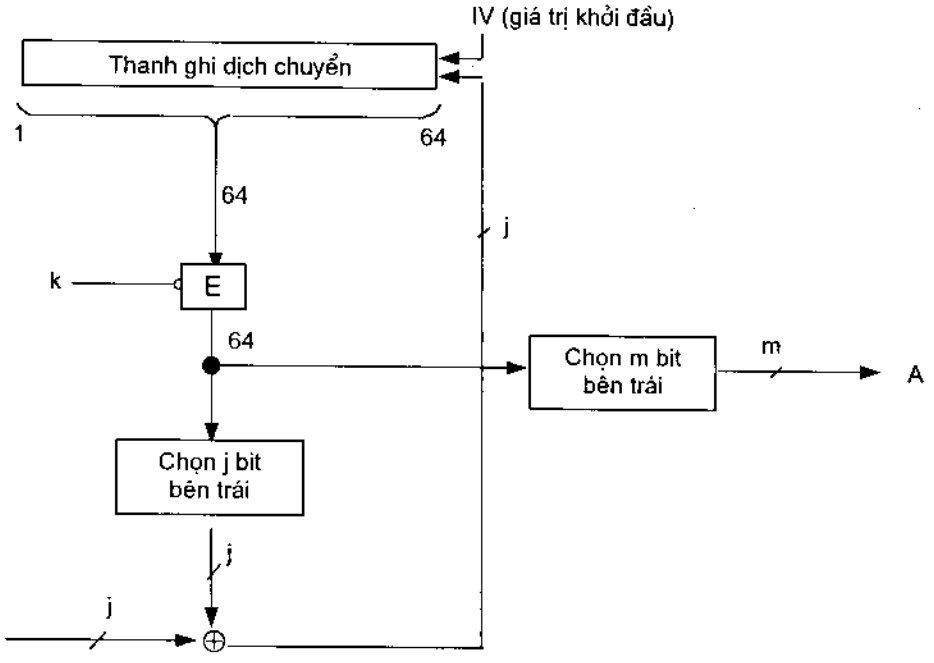
Hình 7.5. ***Mô tả khối chức năng phương pháp xác thực dựa vào mật mã CBC***

Ở phương pháp sử dụng mật mã CFB, bản tin cần nhận dạng bắt buộc phải mã hoá theo CFB. Khi tất cả dữ liệu đã được mã hoá và đơn vị cuối cùng của bản tin đã mã hoá đó được nạp trong thanh ghi đầu vào của DES thì lúc đó sử dụng quá trình xử lý của DES thêm một lần nữa. Thanh ghi đầu ra sẽ có một trạng thái mới. Sử dụng m bit có trọng số lớn nhất của thanh ghi đầu ra đó để tạo mã xác thực. Hình 7.6 mô tả sơ đồ khối nguyên lý của phương pháp.

Ở đây cần lưu ý rằng, nếu như *m* bit của mã xác thực đoạn tin (MAC) đã được lấy ở thanh ghi đầu vào của DES ở chu kỳ cuối cùng của mã hoá thì để cho giá trị *m* tương đối lớn, có thể biến đổi ký tự cuối cùng của bản tin rõ với mã xác thực. Trong thực tế, phương pháp sử dụng mật mã CFB không được thông dụng lắm mà thường sử dụng phương pháp mật mã CBC.

Các mã xác thực theo phương pháp ứng dụng mật mã CBC hoặc mật mã CFB phụ thuộc vào toàn bộ dãy hoặc chuỗi tạo thành bởi đoạn tin. Trong tất cả ứng dụng, đầu và cuối của dãy đó (hoặc chuỗi đó) cần phải được xác định, cũng như vị trí, kích cỡ và khuôn dạng của trường chứa mã xác thực. Việc xác thực cũng phụ thuộc vào giá trị khởi đầu (IV) được nạp trong thanh ghi 64 bit ở đầu

quá trình thực hiện. Có thể sử dụng một giá trị mật cho giá trị khởi đầu (IV) đó, giống như phương pháp thông thường đối với mật mã CBC. Mục đích cua giá trị khởi đầu IV là để ngăn chặn các xâm nhập theo phương pháp tra cứu mở đầu của bản tin đã mã hoá nhưng điều đó ở mã xác thực không cần thiết. Ở đây, đơn giản nhất là sử dụng giá trị khởi đầu IV bằng không, một giải pháp mà nhiều chuẩn đã sử dụng.



Hình 7.6. *Mô tả sơ đồi khối nguyên lý hoạt động phương pháp xác thực sử dụng mật mã CFB.*

7.7. BẢO TOÀN CÁC ĐOẠN TIN BẰNG MẬT MÃ.

7.7.1. Giới thiệu

Việc ứng dụng mật mã cho một đoạn tin thì bản thân công việc đó cũng là một dạng bảo toàn dữ liệu. Nếu như bên gửi và bên nhận một đoạn tin cùng có khoá mật giống nhau, thì việc thu đoạn tin được mã hoá, sau khi giải mã, bên nhận sẽ có bản tin rõ và bên nhận cũng có một sự đảm bảo nào đó là đoạn tin không bị làm biến đổi một cách bất hợp pháp và nó đến từ người cùng có khoá mật. Trong toàn bộ bản tin được xem là đúng đắn đó, có chứa một độ dư thừa vừa đủ. Khi có sử dụng mã xác thực thì bản thân mã xác thực cũng bổ sung thêm một dư thừa cần thiết. Hầu hết các phương thức truyền tin đều có độ dư thừa, nhưng điều đó sẽ không phải luôn luôn thuận lợi cho phía bên nhận khi xử lý và kiểm tra độ dư thừa đó là thoả mãn. Trong thực tế, người ta có thể bổ sung độ dư thừa vào các đoạn tin khi mã hoá đoạn tin cuối để xác thực. Độ dư thừa được thêm vào một đoạn tin đó còn được gọi là *mã phát hiện gian lận*, MDC (Manipulation Detection Code).

Việc thay đổi gian lận một số bit nào đó của bản tin đã mã hoá bằng mật mã CBC sẽ gây nên những biến đổi ngẫu nhiên ở bản tin được giải mã và điều đó có thể nhận biết không khó khăn lắm. Tuy nhiên nếu sự biến đổi đó chỉ có duy nhất một bit của khối đã mã hoá thì việc đó cũng dẫn đến

những biến đổi ngẫu nhiên trong bản tin rõ tương ứng và kéo theo bit tương ứng của khối kế tiếp. Trong nhiều trường hợp nghi ngờ phải cần đến yêu cầu truyền lại đoạn tin. Nếu như trong trường hợp truyền theo mật mã CFB, kẻ xâm nhập có thể chọn một bit trong bản tin rõ để làm sai lệch đi thì phía bên nhận cũng khó phát hiện. Trong nhiều trường hợp phải cần đến xác thực.

7.7.2. Phương pháp kiểm tra so sánh để xác thực

Để dễ phân tích, ở đây giả thiết mật mã sử dụng là CBC. Hình 2.23 (chương 2) đã mô tả nguyên lý hoạt động của mật mã CBC. Tất cả các phép toán trong hình được thực hiện với 64 bit song song. Trong các hệ truyền tin thực, việc truyền dữ liệu có thể là nối tiếp nhưng nguyên lý giải thích để dễ hiểu tốt nhất là sử dụng phép toán song song. Từ song song chuyển thành nối tiếp chỉ là quá trình biến đổi logic.

Phần bên phải của hình 2.23 mô tả quá trình giải mã chuỗi của bản tin đã được mã hoá C_1, C_2, ... C_n. Nếu giá trị khởi đầu IV là I thì chuỗi của bản tin rõ sau giải mã sẽ là:

$$I \oplus Dk(C_1), C_1 \oplus Dk(C_2), \ldots C_{n-1} \oplus Dk(C_n)$$

Tổng modul2 của các từ 64 bit đó là:

$$I \oplus \sum_{i=1}^{n-1} C_i \oplus \sum_{i=1}^{n-1} Dk(C_i)$$

Nếu như có một sự thay đổi bản tin đã mã hoá nhưng không thay đổi tổng kiểm tra 64 bit thì nó vẫn giữ nguyên không đổi tổng kiểm tra nhận được do phép modulo 2 các từ 16 bit. Rất dễ thực hiện dạng thay đổi ấy. Có thể, ví dụ như đảo lộn thứ tự các cặp khối, bởi vì trật tự các khối được thực hiện trong hai phép cộng. Ở đây không thể sử dụng khối cuối cùng C_n, bởi vì khối cuối cùng không liên quan với khối đầu tiên. Bất kỳ khối được mã hoá nào cũng có thể được chèn hai lần trong dãy bản tin đã mã hoá và đặt cạnh nhau, như vậy sẽ dẫn đến chúng sẽ triệt tiêu nhau trong phép cộng modulo-2 và dạng này rất khó phát hiện.

Ở phương pháp dùng mật mã CFB thì vấn đề có phần phức tạp hơn. Sau đây nêu một ví dụ làm thay đổi dữ liệu và tránh các kiểu phát hiện. Giả thiết mật mã sử dụng là CFB với 8 bit, thanh ghi dịch chuyển chứa 8 ký tự (64 bit). Lúc đó biến đổi bản tin được tạo thành bởi dãy các ký tự:

A B C D E F G H

thành dãy

A B C D E F G X A B C D E F G X A B C D E D G H

mà không làm thay đổi tổng kiểm tra 16 bit (ở đây sẽ không phân tích sâu tiếp tục).

Qua các phân tích trên, nhận thấy rằng, việc sử dụng mật mã có khoá mã mật đã bao hàm việc bảo toàn các đoạn tin. Ở đây có thể có câu hỏi đặt ra là, trong các trường hợp này tại sao còn đặt ra vấn đề xác thực. Lý do ở đây là, trong một số trường hợp, việc xác thực là cần thiết dù cho mật mã đã đủ độ khó. Ví dụ, một hệ thống trong đó các đoạn tin được phân bố cho nhiều bên nhận và một số trong số đó có trách nhiệm kiểm tra xác thực. Trong trường hợp như vậy, việc truyền cần phải thực hiện ở dạng rõ, bổ sung phụ trợ thêm cho trường hợp xác thực. Một trong các đầu cuối

kiểm tra trường hợp xác thực và có khoá để thực hiện công việc đó, còn các đầu cuối khác không thể.

Một ví dụ khác là trong một hệ truyền dữ liệu điểm - điểm, một đầu cuối có trách nhiệm xử lý lưu giữ nhưng không có trách nhiệm giải mã tất cả đoạn tin nhận được. Việc xác thực trong trường hợp đó được thực hiện theo phương pháp chọn lọc, chọn ngẫu nhiên các đoạn tin để xác thực.

Cũng một ứng dụng rất quan trọng của việc xác thực một bản tin rõ là sự xác thực bằng các chương trình phần mềm. Trong trường hợp đó, điều quan trọng là bản tin không bị sao chép lại bởi một bộ xử lý. Sẽ là rất tốn kém nếu phải thực hiện phép giải mã cho mỗi lần thực hiện chương trình. Có thể sử dụng việc xác thực chương trình bằng cách đưa vào chương trình một trường xác thực và sử dụng nó để kiểm tra mỗi lần cần đảm bảo tính toàn vẹn của chương trình.

7.7.3. Xác thực không sử dụng khoá bảo mật

Các phương pháp xác thực được đề cập ở các mục trên đều có liên quan đến một khoá mật chung cho cả bên gửi và bên nhận. Có một giải pháp không sử dụng các khoá mật cho cả hai bên, đó là thay vì sử dụng mã xác thực A(k, M) ở đây sẽ sử dụng một hàm một chiều A(M) mà hàm đó chỉ đơn thuần phụ thuộc vào đoạn tin.

Hàm một chiều đó, như đã phân tích trong chương 4 và chương 5, có các tính chất sau: Nếu biết một tập các đoạn tin M và các giá trị khác A_0, đặc biệt khó khăn để xác định một giá trị của đoạn tin M để có A(M) = A_0. Một điều kiện cần thiết là A nhận các giá trị của nó trong một tập có kích cỡ lớn, ví dụ tập từ 64 bit. Kích cỡ của tập đó phải đủ để không thể thiết lập được một bảng các đoạn tin được phân loại theo giá trị của A.

Các hàm một chiều dạng này còn được gọi là *hàm hash* hoặc *hàm băm* (hash-functions). Nhưng các hàm "giả ngẫu nhiên" này được ứng dụng cho nhiều mục đích khác nhau và một số hàm băm không thích hợp với việc xác thực.

Để xác thực một đoạn tin, ở đây sẽ tính đại lượng A(M) và truyền đại lượng đó cho thực thể nhận hoặc đọc đoạn tin, một phương pháp đảm bảo xác thực của A(M). Nhờ có đại lượng đó, bên nhận có thể kiểm tra M và tin chắc rằng A(M) là đồng nhất với giá trị xác thực.

Các ứng dụng của giải pháp xác thực này trong thực tế không được phát triển lắm. Có thể nêu một ví dụ sau đây về ứng dụng giải pháp đã nêu. Giả thiết A gửi cho B một tập lớn các dữ liệu và B muốn có đảm bảo xác thực. Cả hai bên đồng ý với nhau sử dụng một hàm một chiều A(M). Một cuộc gọi điện thoại cho phép kiểm tra giá trị A(Mb) do B nhận được có phù hợp với giá trị A(Ma) do A tính không. Nếu các giá trị đó bằng nhau, có nghĩa là Ma bằng Mb. Bên B chấp nhận phiên bản đó. Cuộc gọi điện thoại ở đây cũng là một sự xác thực của A(M), bởi vì hai đối tác có thể trao đổi những vấn đề chung và dễ dàng phát giác kẻ xâm nhập.

7.8. BÀI TOÁN TRÒ CHƠI

7.8.1. Giới thiệu

Việc sử dụng mật mã hoặc mã xác thực trong trao đổi dữ liệu trên mạng truyền tin thì kẻ xâm nhập vẫn đọc được và chúng có thể sử dụng lại các dữ liệu đó cho một giao dịch gian lận (tất nhiên

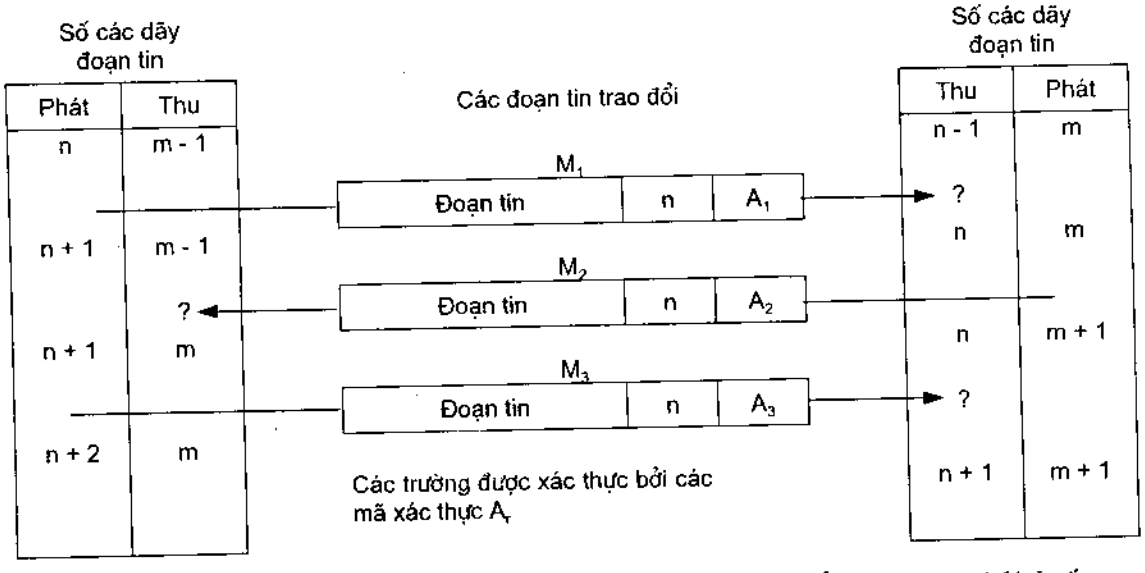
với điều kiện là trong thời gian khoá mã còn giá trị). Khả năng "trò chơi các đoạn tin" làm khó khăn thêm việc xác thực. Trò chơi cũng có thể dưới dạng làm thay đổi trật tự các đoạn tin hoặc thay đổi vị trí dữ liệu trong bộ nhớ bằng một giá trị trước đó của cùng dữ liệu. Có rất nhiều dạng trò chơi xâm nhập, sau đây chỉ tập trung đề cập đến giao dịch giữa hai thực thể truyền tin.

Lời giải của bài toán trò chơi về nguyên lý cũng đơn giản; Chỉ cần mỗi đoạn tin có sự khác biệt nào đó với các đoạn tin trước nếu sử dụng cùng khoá và bên nhận có thể kiểm tra sự "giả mạo" của mỗi đoạn tin. Việc bảo vệ chống lại trò chơi các đoạn tin sẽ được hoàn thiện nếu như mỗi thực thể của truyền tin so sánh tất cả đoạn tin nhận với tất cả các đoạn tin nhận trước đó trong cùng khoá và loại bỏ những đoạn tin sao chép đoạn tin trước.

Việc xác thực các giao dịch như vậy cần thực hiện hai chiều: mỗi thực thể này phải xác thực thực thể kia. Trong một số trường hợp việc xác thực chỉ một chiều là đủ nhưng cần lưu ý việc xác thực lẫn nhau.

7.8.2. Phương pháp đánh số dãy đoạn tin

Hai thực thể thường xuyên thực hiện các giao dịch với nhau có thể thực hiện việc đánh số các đoạn tin theo kiểu gia tăng một đơn vị cho mỗi đoạn tin được trao đổi. Hình 7.7 mô tả việc sử dụng phương pháp đánh số như nói trên để trao đổi các đoạn tin M_1, M_2, ... với các mã xác thực cho mỗi đoạn tin.

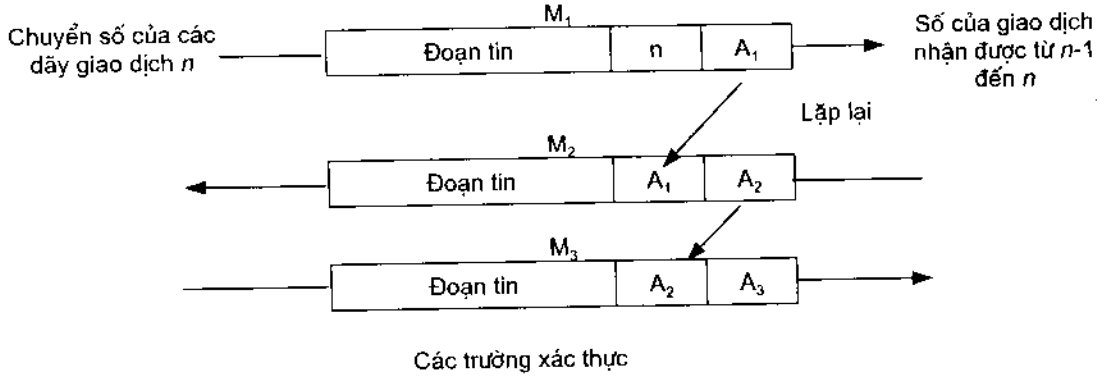


Hình 7.7. ***Mô tả phân phối số các dãy đoạn tin trong việc trao đổi các đoạn tin có đánh số.***

Ở đây mỗi đoạn tin có một số dãy và mỗi thực thể có số riêng của mình, các đoạn tin trao đổi được đánh số như kiểu đánh số công văn đi đến của văn thư. Mã xác thực được tính trên toàn bộ đoạn tin còn lại trong đó có số của dãy. Mỗi thực tế biết số của đoạn tin mình đang chờ và như vậy tất cả việc phá hoại đoạn tin cũng như tất cả trò chơi truyền lại đoạn tin trước, tất cả việc thay đổi trật tự các đoạn tin đều dễ dàng bị bên nhận phát hiện.

Có một phương pháp khác, đó là có sự móc xích giữa các đoạn tin của một giao dịch. Hình

7.8 mô tả một mô hình trong đó mã xác thực của đoạn tin trước được lặp lại cho đoạn tin tiếp theo. Đoạn tin M_2 có một bản sao mã xác thực A_1 của đoạn tin M_1. Mã xác thực A_2 của đoạn tin M_2 được tính với nội dung của đoạn tin và với giá trị của A_1. Trong thực tế, mã xác thực là một số ngẫu nhiên có quan hệ hoặc có móc xích với dãy các đoạn tin theo kiểu mà kẻ xâm nhập khai thác được. Để chọn kích cỡ các mã xác thực có sự ràng buộc đó, người ta sử dụng tiêu chuẩn sau: xác suất để kẻ xâm nhập tìm được quan hệ ràng buộc một sự kiện cần phải nhỏ rất nhiều so với công sức bỏ ra để thăm dò tính toán nó. Trong thực tế thường sử dụng là 16 bit hoặc 32 bit.



Hình 7.8. ***Mô tả sự móc xích các đoạn tin bởi các giá trị của các mã xác thực***

Khi mà các đoạn tin được mã hoá thì điều đó rất thuận lợi cho việc tiến hành móc xích các đoạn tin: Có thể sử dụng phương pháp mật mã CBC liên tiếp nhau giữa một đoạn tin và trả lời của nó. Trong trường hợp này, 64 bit cuối cùng của một đoạn tin sẽ là tham số khởi đầu cho đoạn tin tiếp theo sau. Kiểu móc xích này không kéo dài thêm đoạn tin. Đối với mật mã CFB cũng có thể sử dụng phương pháp móc xích tương tự.

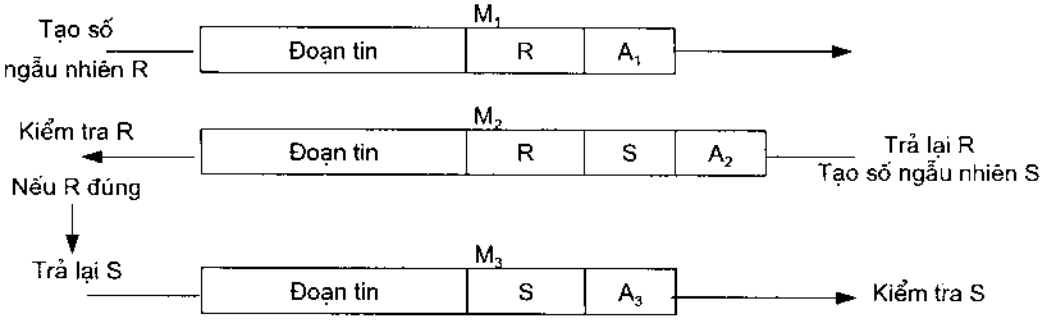
Một khi mà các đoạn tin được móc xích theo một trong các phương pháp nói trên (tức là lặp lại mã xác thực hoặc tính giá trị khởi đầu IV xuất phát từ đoạn tin trước nó) thì số dãy n của đoạn tin đầu tiên là đồng nhất trong toàn bộ giao dịch. Số dãy đầu tiên đó là chung hay riêng cần được thống nhất giữa hai thực thể giao dịch.

7.8.3. Sử dụng các số ngẫu nhiên để xác thực bảo toàn dữ liệu

Trong trường hợp có nhiều người sử dụng, việc xác thực có thể dựa trên các khoá phiên làm việc do trung tâm phân phối khoá cấp. Điều đó tránh được việc mỗi thực thể nắm giữ các khóa mà thực thể đó cần để liên lạc. Cũng như vậy, để đơn giản vấn đề phân phối khoá, có thể sử dụng các số ở mật mã có khoá công khai (xem cụ thể ở chương 4). Trong trường hợp đó, nói chung không thể mỗi thực thể giữ cho mình các số của các dãy để thực hiện truyền tin cho mình. Ở đây có thể (với một xác suất khá lớn) mỗi đoạn tin khác với tất cả các đoạn tin bằng cách đưa vào đó một số ngẫu nhiên. Ví dụ có giao dịch ba đoạn tin như mô tả ở hình 7.9. Thực thể xuất phát đầu tiên đưa vào trong đoạn tin của mình một số ngẫu nhiên R và nó sẽ được gửi trả lại trong trả lời. Mỗi đoạn tin chứa một mã xác thực được tính với toàn bộ nội dung cả đoạn tin, trong đó gồm các số ngẫu nhiên, và đồng thời cũng cho phép tách các số ngẫu nhiên của nội dung đó trong mạng truyền tin. Việc gửi

trả lại số ngẫu nhiên đảm bảo cho bên bắt đầu giao dịch rằng, trả lời xuất phát từ thực thể đúng. Trong trường hợp này, nếu như có N giao dịch cùng khoá mã giống nhau, thì kích cỡ của trường chứa các số ngẫu nhiên phải lớn hơn N^2, mục đích là để cho xác suất lặp lại của một số ngẫu nhiên sẽ nhỏ.

Để thực hiện một xác thực lẫn nhau thì thực thể phía bên kia đưa vào trả lời một số ngẫu nhiên S. Số đó được gửi cho đoạn tin thứ ba của giao dịch để đảm bảo rằng, giao dịch là mới, có nghĩa là không phải lặp lại giao dịch cũ.



Hình 7.9. ***Mô tả phương pháp sử dụng các số ngẫu nhiên để xác thực bảo toàn đoạn tin truyền.***

Mỗi thực thể cài số ngẫu nhiên của mình trong thể thức để tự bảo vệ chống các xâm nhập bất thường.

Giả thiết, trong trường hợp M_1 là sự lặp lại của đoạn tin khởi đầu của một giao dịch trước. Một bên nhận không nghi ngờ và gửi trả lại M_2 và trong trả lời có chứa chính xác R và số ngẫu nhiên mới S. Nếu M_3 là được nhận bằng trả lời thì điều đó chỉ có thể do một thực thể có khoá mật để tính giá trị A_3 của mã xác thực. Việc xác thực của A_3 chứng minh cho bên nhận rằng chính đó là M_1. Nếu đó chỉ là một trò chơi thì người phát chỉ cần nhìn qua M_2 là biết, không cần chờ đến M_3.

Nếu việc giao dịch phải tiếp tục với các đoạn tin khác thì cần phải tiến hành theo phương pháp móc xích các đoạn tin như đã giới thiệu ở mục trước.

7.8.4. Bảo toàn các dữ liệu lưu giữ

Các dữ liệu lưu giữ có thể được bảo vệ chống lại các thay đổi bất hợp pháp bằng cách sử dụng các kỹ thuật đã được sử dụng để trao đổi dữ liệu như đã giới thiệu ở các mục trước. Các giá trị của mã xác thực được tính dựa vào một khoá mã mật và được lưu giữ cùng với các phần của dữ liệu mà nó bảo vệ. Những người sử dụng các dữ liệu đều nắm giữ khoá và họ có thể kiểm tra giá trị của mã xác thực.

Có thể bảo vệ toàn bộ tập hồ sơ dữ liệu chỉ bằng một mã xác thực, nhưng như vậy một người sử dụng muốn truy nhập một trong các dữ liệu đó thì trước đó phải thực hiện các phép tính xác thực trên toàn bộ hồ sơ. Mọi thay đổi trong hồ sơ, dù là rất nhỏ, cũng phải tính lại mã xác thực cho toàn bộ hồ sơ. Thông thường người ta phân chia bộ hồ sơ thành các đơn vị nhỏ thích hợp với số lượng bản ghi hoặc đọc chỉ một lần. Nhưng như vậy cũng có vấn đề nảy sinh, đó là nếu như mỗi đơn vị được phân chia đó có một mã xác thực riêng, kẻ xâm nhập phá hoại có thể sắp xếp lại trật tự theo

kiểu khác. Có loại hồ sơ thì việc thay đổi trật tự đó không quan trọng nhưng có loại hồ sơ thì việc thay đổi trật tự đó dẫn đến thay đổi nội dung khi truy nhập.

Cũng có thể có những xâm nhập tích cực vào các dữ liệu lưu giữ hoặc lợi dụng kênh truyền tin để truy nhập vào dữ liệu lưu giữ. Các xâm nhập đó có thể xảy ta bởi vì mã xác thực các dữ liệu lưu giữ thường có đời sống khá dài; việc thay đổi mã xác thực phải xử lý lại toàn bộ hồ sơ. Để chống lại dạng xâm nhập này cần thực hiện theo phương pháp đánh số dãy hoặc sử dụng bộ định thời theo năm, tháng, ngày, giờ, phút, giây. Các bộ định thời nên thường xuyên đồng bộ với đồng hồ mạng truyền tin. Sự tham gia của các bộ định thời có thể phức tạp thêm dữ liệu lưu trữ, cho nên chỉ sử dụng trong một số trường hợp cần thiết cùng với mã xác thực.

7.9. BÀI TOÁN TRANH CHẤP

Việc xác thực sử dụng khoá mã bí mật là đối xứng, có nghĩa là hai bên giao dịch cùng biết khoá mật và cả hai bên đều có thể tạo ra mã xác thực. Việc sử dụng khoá mật như vậy chỉ có tác dụng bảo vệ chống lại người thứ ba chứ không phải để chống lại nhau. Tuy vậy cũng có thể có tranh chấp xảy ra. Ví dụ, A có thể gửi cho B một đoạn tin được xác thực, nhưng nếu như B có ý đồ gian lận thì B có thể bịa hoặc làm giả một đoạn tin khác và cho rằng, đoạn tin đó được nhận từ A. Mã xác thực trong trường hợp này sẽ không cung cấp được chứng cứ để giải quyết tranh chấp và để chứng minh rằng đoạn tin của B là một đoạn tin giả mạo. Hoặc ngược lại A có thể từ chối rằng A không gửi đoạn tin mà thực tế A đã gửi. Lúc đó B cũng không thể nào chứng minh được bằng mã xác thực. Phương pháp dùng khoá bí mật đối xứng trên là đối với hai bên thực thà chân thực, hoàn toàn tin cậy nhau, còn trong trường hợp tranh chấp thì không thể giải quyết được với giá trị khoá mã. Bài toán không có khả năng cung cấp chứng cứ để giải quyết tranh chấp đó về chuyên môn được gọi là "*bài toán tranh chấp*".

Có thể có nhiều phương pháp để khắc phục tranh chấp hoặc nói cách khác là tìm lời giải cho bài toán trên. Giải pháp thích hợp nhất là sử dụng chữ ký số, như đã được trình bày giới thiệu ở chương 5, chữ ký đó kéo dài mô tả đến cả phần xác thực.

7.10. CÁC THUẬT TOÁN BĂM (HÀM HASH)

7.10.1. Giới thiệu chung và thuật toán băm đơn giản

Một sự xác thực đơn giản nhất là chỉ cần biết chính xác là ai đã gửi đoạn tin chứ không cần phải che dấu đoạn tin hoặc cũng không cần biết đoạn tin đó đã đến bằng cách nào, con đường nào. Đoạn tin trên đường truyền tải có thể bị sai lạc một cách ngẫu nhiên (ví dụ do bị lỗi đường truyền) hoặc bị sai lạc một cách cố ý.

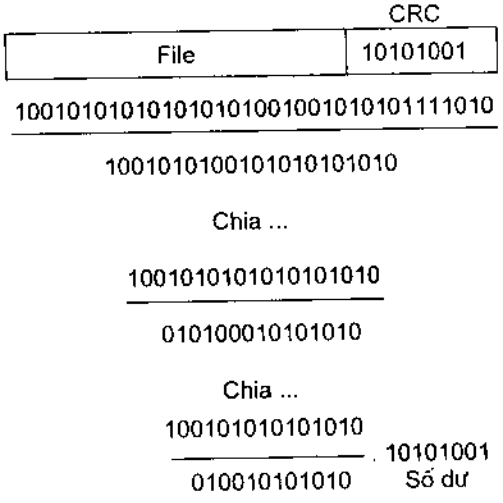
Để tăng cường tính toàn vẹn của đoạn tin thì đoạn tin có thể được mã hoá, như đã giới thiệu ở mục trên. Có một cách khác có thể tăng cường tính bảo toàn của đoạn tin truyền, đó là một kỹ thuật được gọi là "*băm*" (hashing) hoặc *thuật toán băm* (hashing algorithm), trong một số tài liệu gọi là "*hàm hash*".

Kỹ thuật băm là một dạng xác thực để gia tăng tính an toàn dữ liệu truyền nhưng không gửi và cũng không nhận xác thực. Thuật toán băm thực hiện việc "băm" dữ liệu theo một phương pháp

được tính toán và thêm vào đó chữ ký phản ánh kết quả. Các thuật toán băm thường gặp là thuật toán MD2 (của Kaliski, 1992), thuật toán MD5 (của Rivest, 1992) và thuật toán băm có bảo mật (của NIST).

Ở phía nhận cũng thực hiện phép toán tương tự như phía gửi. Các kết quả tính toán được so sánh với một bảng. Nếu so sánh phù hợp thì xem như dữ liệu trao đổi giữa hai bên trao đổi không có gì thay đổi. Việc kiểm tra theo dõi độ dư vòng (CRC) để phát hiện virút là ví dụ một thuật toán băm đơn giản dựa trên phép chia dữ liệu với một đa thức theo lý thuyết đại số đa thức đồng dư. Kết quả phép chia, tức số dư sẽ sử dụng là chữ ký CRC. Việc so sánh CRC tính toán được với một bảng tham chiếu là cần thiết. Hình 7.10 mô tả việc thực hiện phép chia đoạn tin có mã kiểm tra CRC.

Có một cách sử dụng kỹ thuật băm là chia và đánh dấu file, sử dụng phần mềm công khai. Các bộ kiểm tra virut có chữ ký và tính toán chúng là sử dụng phần mềm công khai. Nếu như việc kiểm tra đánh dấu file và chữ ký kiểm tra virút là không phù hợp tức có sự sai lạc file xuất hiện. Chữ ký không có trong phần mềm công khai và chỉ xuất hiện trong phần mềm kiểm tra virút.



Hình 7.10. ***Một ví dụ về kỹ thuật băm thực hiện theo phép chia đa thức.***

Ở đây cần lưu ý một vấn đề là, có thể có một số file trong thuật toán hàm băm nào đó có thể được thay đổi một cách hợp pháp để cài đặt thông tin người dùng, ví dụ như tên hoặc tuỳ chọn của khách hàng. Các thông tin thay đổi đó có các file hoặc các vị trí cách biệt với việc tính toán chữ ký và đó chính là điều kiện tốt cho virút tấn công hoặc các giả mạo tấn công.

Có thể hiểu rằng, hàm băm là hàm dạng một chiều, không thể thực hiện phép biến đổi ngược như trong quá trình mã hoá và giải mã; có nghĩa là với một đoạn tin cho trước khó có thể tìm được đoạn tin nào có hàm băm bằng đoạn tin số đã cho.

7.10.2. Thuật toán băm MD5

Thuật toán băm MD5 là thuật toán được sử dụng tương đối phổ biến để kiểm tra tính toàn vẹn của khối dữ liệu lớn. Thuật toán nhận đầu vào là một đoạn tin có chiều dài bất kỳ, xử lý nó thành các khối 512 bít và tạo đầu ra là một đoạn tin 128 bit. Quá trình thực hiện bao gồm các bước sau:

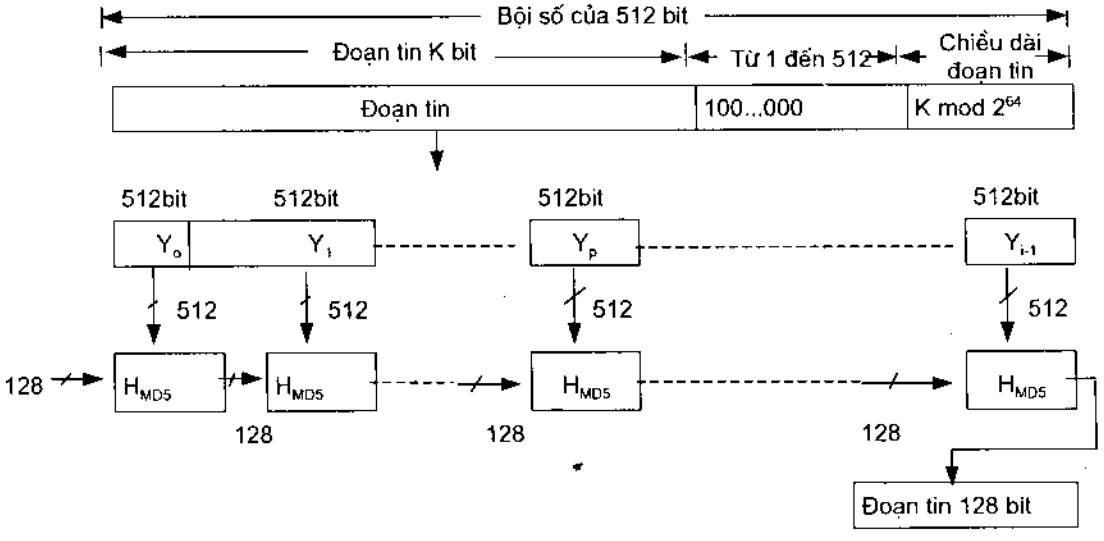
Bước 1: Đoạn tin ban đầu được nhồi thêm (padding) một số bit (bắt đầu là bit 1, kế tiếp là các bit 0, số bit thêm vào từ 1 đến 512 bit) sao cho tổng số bit của đoạn tin (sau khi cộng thêm giá trị khởi đầu IV là 64 bit) sẽ là bội số của 512.

Bước 2: Khởi tạo bộ đệm MD. Bộ đệm 128 bit được sử dụng để chứa kết quả trung gian và kết quả cuối cùng của hàm băm. Có thể xem bộ đệm như là bốn thanh ghi 32 bit. Các thanh ghi này được khởi tạo (dạng số thập lục phân, hex) như sau:

A = 0 1 2 3 4 5 6 7; B = 8 9 a b c d e f

C = f e d c b a 9 8; D = 7 6 5 4 3 2 1 0

Hình 7.11 mô tả ví dụ tạo đoạn tin sử dụng thuật toán băm MD5.



Hình 7.11. ***Tạo đoạn tin sử dụng thuật toán băm MD5***

Bước 3: Xử lý đoạn tin thành từng khối 512 bit (16 từ 32 bit). Quá trình tính toán được chia thành từng giai đoạn, số giai đoạn bằng số chiều dài (tính theo bit) của đoạn tin sau khi đã nhồi thêm chia cho 512. Mỗi giai đoạn nhận đầu vào là khối 512 bit của đoạn tin đã được nhồi thêm và đoạn tin kết quả của giai đoạn trước nó, để cho ra kết quả đoạn tin mới (xem hình 7.11). Mỗi giai đoạn thực hiện trong bốn vòng, các vòng có cấu trúc giống nhau nhưng mỗi vòng sử dụng một hàm luận lý khác nhau, được đặc trưng là F, G, H, I. Trong hình 7.12, bốn vòng được đặt nhãn là f_F, f_G, f_H và f_I để chỉ rằng, mỗi vòng có cấu trúc hàm tổng quát như nhau, nhưng tuỳ thuộc vào sự khác nhau của hàm thao tác (F, G, H, I).

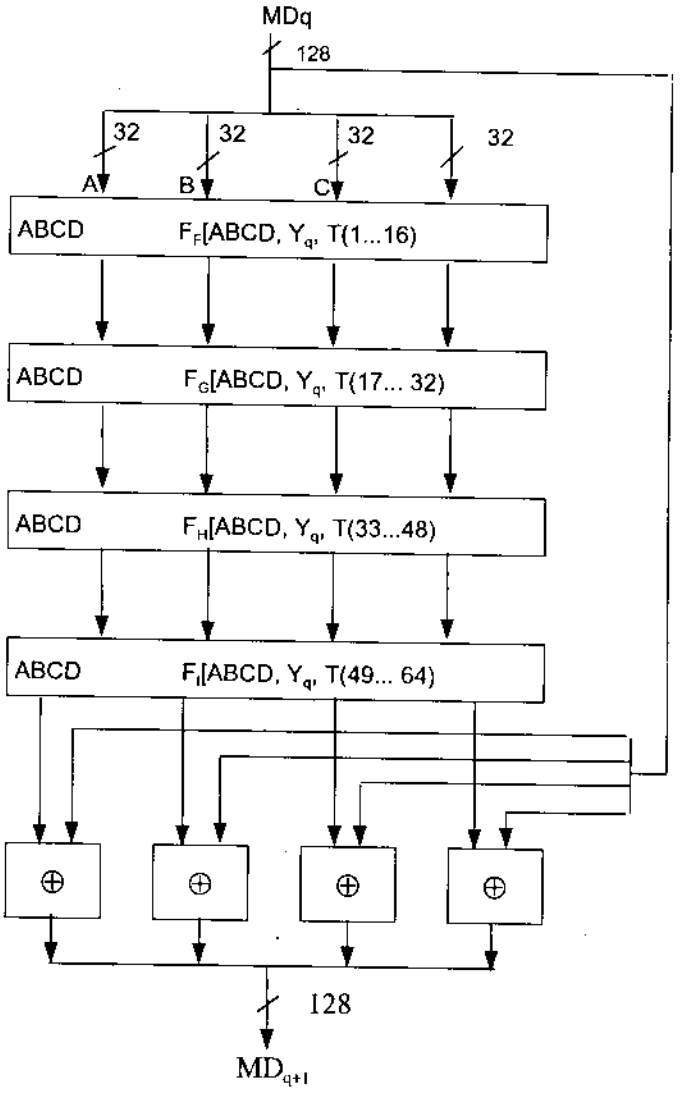
Mỗi vòng được thực hiện 16 bước tuần tự trên các data A, B, C, D như mô tả ở hình 7.13.

Biểu thức tính toán được sử dụng cho mỗi vòng có dạng:

$$a = b + \mathrm{CLS}_s\{a + g(b, c, d) + X[k] + T[i]\}$$

trong đó:

a, b, c, d là bốn từ A, B, C, D theo thứ tự nào đó;



Hình 7.12. ***Xử lý thuật toán băm MD5 của khối đơn 512 bit (HD[MD5])***

g là một trong các hàm F, G, H, I;

$$F(b, c, d) = (b*c)|(b*d)$$

$$G(b, c, d) = (b*d)|(c*d)$$

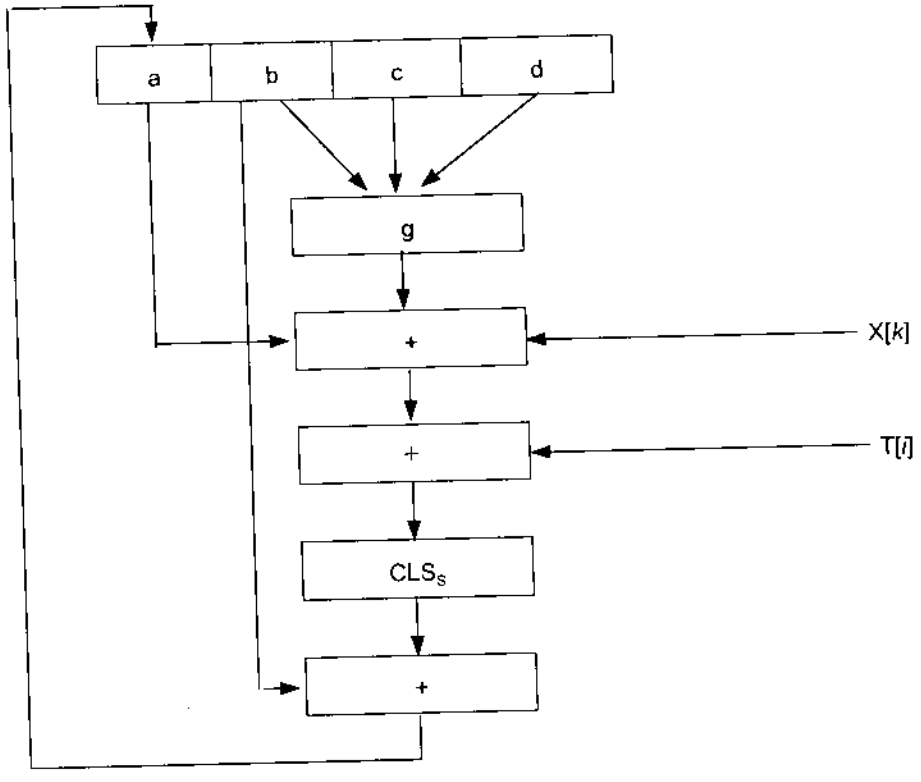
$$H(b, c, d) = b \oplus c \oplus d$$

$$I(b, c, d) = c \oplus (b*d)$$

CLS_s là dịch vòng phía trái s bít;

$X[k] = M[q * 16 + k]$: từ 32 bit thứ k của khối 512 bit thứ q của đoạn tin;

$T[i] = abs[sin(i)]$: từ 32 bit thứ i (số liệu tra cứu theo bảng); Phép toán cộng (+) tính theo modulo 2^{32}.



Hình 7.13. *Mô tả tác vụ của MD5*

Bước 4: Đoạn tin đầu ra có độ dài là 128 bit sau khi thực hiện phép cộng modul-2 của hai toán hạng 32 bit với nhau trong bốn khối tin.

Có thể tóm tắt hoạt động của thuật toán MD5 như sau:

$$MD_0 = IV$$

$$MD_{q+1} = MD_q + f_I(Y_q, f_H(Y_q, f_G(Y_q, f_F(Y_q, MD_q))))$$

$$MD = MD_{L-1}$$

trong đó:

IV là giá trị khởi đầu của bộ đệm ABCD được xác định ở bước 2.

Y_q là khối tin 512 bit thứ q.

L là số khối đoạn tin (đã được nhồi thêm)

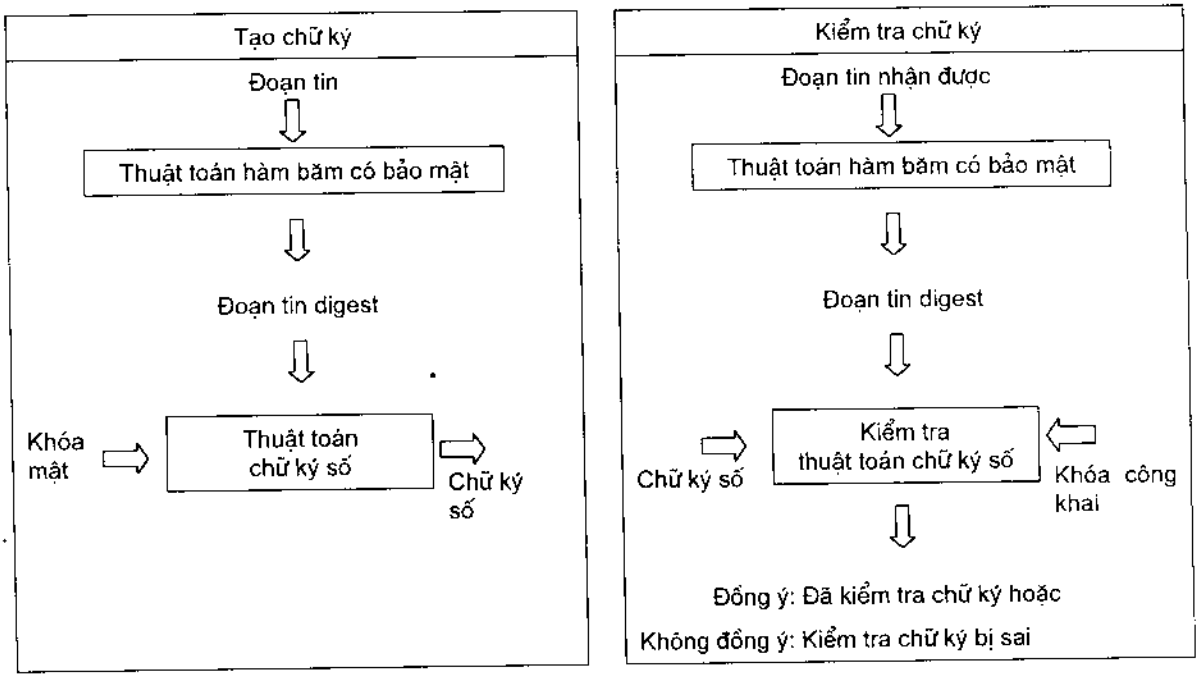
MD là giá trị đoạn tin cuối cùng.

7.10.3. Thuật toán băm có bảo mật

Hình 7.14 mô tả một đoạn tin nhỏ hơn 2^{64} bit được đưa vào *thuật toán băm có bảo mật*, SHA (Secure Hashing Algorithm). Kết quả sau khi băm là đầu ra 160 bit được gọi là đoạn tin digest. Một khoá bí mật tăng cường được sử dụng như một chữ ký. Giữa thuật toán chữ ký số, DSA (digital signature algorithm) và thuật toán băm có bảo mật (SHA) có một đoạn tin được xác thực của độ toàn vẹn đã biết được tạo ra.

Tại phía thu, thuật toán SHA được thực hiện trên đoạn tin và lại cũng tạo ra ở đầu ra đoạn tin digest. Nếu như kết quả băm đó là giống và chữ ký được kiểm tra (ở phía thu sử dụng một khoá công khai để kiểm tra), thì như vậy đoạn tin là đã được xác thực đúng người gửi và đã có bảo toàn dữ liệu.

Một sự thay đổi nào đó trong bản thân đoạn tin sẽ dẫn đến sự thay đổi đoạn tin digest đầu ra với một xác suất rất lớn và điều đó sẽ dẫn đến sai khi kiểm tra chữ ký băm.



Hình 7.14. ***Thuật toán hàm băm và chữ ký số***

Thuật toán băm (SHA) có thể được cài đặt bằng phần cứng, phần mềm, phần sụn hoặc bằng hỗn hợp giữa các phương pháp.

Chương 8

BẢO VỆ THÔNG TIN DỮ LIỆU TRONG MẠNG TRUYỀN TIN

8.1. CÁC DỊCH VỤ BẢO MẬT TRONG CÁC LỚP CỦA CẤU TRÚC HỆ THỐNG MỞ

Biết rằng, việc kết nối các hệ thống mở theo các tiêu chuẩn khuyến nghị của ISO được cấu trúc theo mô hình bảy lớp, trong đó lớp thấp nhất là lớp vật lý và lớp cao nhất là lớp ứng dụng. Ba lớp đầu tiên (lớp vật lý, lớp liên kết dữ liệu và lớp mạng) hình thành cấu trúc truyền tin. Các chuẩn của các lớp đó được giới thiệu trong các khuyến nghị của CCITT. Các lớp cao (chuyển vận, tập hợp, trình bày và ứng dụng) sử dụng các khả năng truyền tin của lớp mạng. Các lớp đó được xem như hoạt động từ đầu cuối này đến đầu cuối kia thông qua hệ thống truyền tin.

Về nguyên lý, các hệ bảo mật có thể cài trong bất kỳ lớp nào của mô hình ISO-7 lớp nhưng cấu trúc phải rõ ràng và loại dịch vụ nào cần phải cài đặt. Theo DSI, có một sự phân biệt rõ ràng giữa "dịch vụ" bảo mật và "cấu trúc" bảo mật. Ví dụ, mật mã là một "cấu trúc" bảo mật có thể được sử dụng để cung cấp cho các "dịch vụ" bảo mật khác nhau mà thường gặp nhất là đảm bảo tính bí mật. Các tài liệu của ISO nhấn mạnh về các dịch vụ nhiều hơn là cấu trúc.

Các dịch vụ bảo mật được phân chia làm năm nhóm, đó là: đảm bảo bí mật, xác thực, bảo toàn dữ liệu, không chối bỏ trách nhiệm và kiểm tra truy nhập. Mỗi nhóm dịch vụ có thể được phân chia thành nhiều dịch vụ phụ thuộc vào phương thức truyền tin sử dụng và dịch vụ yêu cầu.

Đảm bảo bí mật dữ liệu là đảm bảo sao cho thông tin không sử dụng được bởi những cá nhân, những thực thể hoặc những chương trình không có trách nhiệm. Ở đây có thể là toàn bộ đoạn tin hoặc có thể là các trường trong đoạn tin. Việc đảm bảo bí mật dữ liệu được phân biệt trường hợp các thể thức truyền tin không kết nối (không chiếm kênh) và trường hợp có thiết lập kết nối (chiếm kênh) giữa hai thực thể tương đồng.

OSI sử dụng từ "*xác thực*" với một ý nghĩa cụ thể, có hơi khác với những phân tích ở các mục trước theo nghĩa rộng. Với OSI, xác thực có nghĩa là xác định các dữ liệu nhận được đến đúng từ nguồn xuất phát đã được tổ chức xếp đặt. Với một dịch vụ không kết nối, thì thuật ngữ "xác thực nguồn gốc các dữ liệu" có nghĩa là "xác nhận nguồn các dữ liệu nhận được chính là nguồn đã công

bố". Một khi có sự đồng ý trước giữa hai thực thể trao đổi thì đó là sự xác thực của hai thực thể trao đổi. Thuật ngữ xác thực ở đây không bao hàm sự bảo toàn các dữ liệu được truyền.

"*Bảo toàn dữ liệu*" là bảo vệ sao cho dữ liệu không bị sai lạc hoặc không bị phá huỷ do các xâm nhập bất hợp pháp. Trong thực tế việc bảo toàn dữ liệu và xác thực dữ liệu có sự liên quan chặt chẽ với nhau và chúng sử dụng kết cấu giống nhau. Thực vậy, nếu như các dữ liệu nhận được là đảm bảo đến đúng từ một nguồn đã được công bố thì điều đó trong thực tế còn phải xem xét chúng có bị làm sai lạc hoặc biến đổi một cách bất hợp pháp không. Hoặc ngược lại, biết được các dữ liệu truyền là đúng đắn, không bị sai lạc chỉ thực sự hữu ích khi biết chắc rằng nó đến từ một nguồn giả mạo. Chính vì vậy mà việc bảo toàn dữ liệu và xác thực thường được gắn với nhau chứ không tách riêng.

Cấu trúc cho phép phục vụ các dịch vụ trên phụ thuộc vào các thể thức mật mã sử dụng và việc phân chia khóa mã mật giữa hai thực thể truyền tin. Trong nhiều trường hợp, việc sở hữu khóa mã mật đó cho phép thiết lập việc xác thực và đồng thời cho phép kiểm tra bảo toàn dữ liệu. Cơ cấu xác thực không phải dùng đến mật mã và để đảm bảo là xác thực có tính hệ thống.

Dịch vụ bảo toàn dữ liệu có thể thay đổi theo phương thức truyền tin là có kết nối hay không kết nối. Dịch vụ bảo toàn dữ liệu cũng có thể thực hiện với toàn bộ đoạn hoặc các trường được lựa chọn trong đoạn tin, giống như trường hợp bảo đảm bí mật dữ liệu.

Thuật ngữ "*chối bỏ trách nhiệm*" ở đây được hiểu là, các thực thể có liên quan trong truyền tin chối bỏ trách nhiệm là đã tham gia các cuộc trao đổi, có thể là chối bỏ hoàn toàn hoặc chối bỏ một phần. Dịch vụ "không chối bỏ trách nhiệm" theo cấu trúc của OSI có thể dưới hai dạng sau đây: hoặc là với các chứng cớ xuất xứ từ nguồn xuất xứ hoặc là với chứng cớ để lại. Khái niệm về việc không thể chối bỏ trách nhiệm được trình bày kỹ trong chương 5 về chữ ký số. Chữ ký số là một giải pháp rất tốt đảm bảo người tham gia truyền tin hoặc giao dịch không thể chối bỏ trách nhiệm và lưu ý rằng ở đây phải có sự công chứng của thực thể thứ ba được tín nhiệm cả hai bên (cơ chế trọng tài).

Dịch vụ "*kiểm tra truy nhập*" có thể ứng dụng hoặc cho nguồn truyền tin, hoặc tại một điểm trung gian hoặc tại bên nhận. Kiểm tra truy nhập có thể cần thiết để bảo vệ mạng chống lại các truy nhập bất hợp pháp, và thông thường được sử dụng để bảo vệ máy chủ chống lại các truy nhập không được phép. Kiểm tra truy nhập thường sử dụng trong lớp ứng dụng. Trong lớp mạng, dịch vụ kiểm tra truy nhập có thể phục vụ bảo vệ một mạng con bằng cách chỉ cho phép các thực thể hợp pháp thiết lập liên lạc với mạng. Ở lớp chuyển vận việc kiểm tra truy nhập cũng có thể được thực hiện tương tự.

Một dạng đặc biệt của sự đảm bảo bí mật là đảm bảo bí mật dòng các dữ liệu, tức bảo vệ chống lại việc phân tích lưu lượng, cụ thể là chống sự suy đoán thông tin xuất phát từ việc quan sát các dòng lưu lượng dữ liệu. Cũng rất khó có một sự bảo vệ hoàn chỉnh chống lại việc phân tích lưu lượng và nhiều cơ cấu có thể phải sử dụng đồng thời.

Các cấu trúc được sử dụng để cung cấp cho các dịch vụ trên bao gồm mật mã với thuật toán

đối xứng và không đối xứng, chữ ký số, cấu trúc bảo toàn dữ liệu và xác thực, chèn thêm lưu lượng, kiểm tra định tuyến và công chứng; trong số đó cũng có những cấu trúc có thể áp dụng cho tất cả các mức của mô hình cấu trúc mở OSI.

Bảng 8.1 mô tả các dịch vụ bảo mật có thể thực hiện trong các lớp của mô hình cấu trúc mở ISO-7 lớp, trong đó lớp 7 (ứng dụng) có tất cả các dịch vụ bảo mật được thực hiện và lớp 5 (tập hợp) không có dịch vụ nào thực hiện.

BẢNG 8.1. *Mô tả các dịch vụ bảo mật trong các lớp ISO.*

Các dịch vụ bảo mật	Lớp						
	1	2	3	4	5	6	7
	(Vật lý)	(Liên kết dữ liệu)	(Mạng)	(Chuyển tải)	(Tập hợp)	(Trình bày)	(Ứng dụng)
Đảm bảo bí mật:							
Phương thức có kết nối	x	x	x	x		x	x
Phương thức không kết nối		x	x	x		x	x
Lựa chọn theo trường						x	x
Đảm bảo bí mật dòng dữ liệu	x		x				x
Xác thực (cả hai dạng)			x	x			x
Bảo toàn dữ liệu:							
Có kết nối không hồi tiếp			x	x			x
Có kết nối, có hồi tiếp				x			x
Có kết nối, chọn trường							x
Không chối bỏ trách nhiệm							x
Kiểm tra truy nhập			x	x			x

x: dịch vụ được thực hiện.

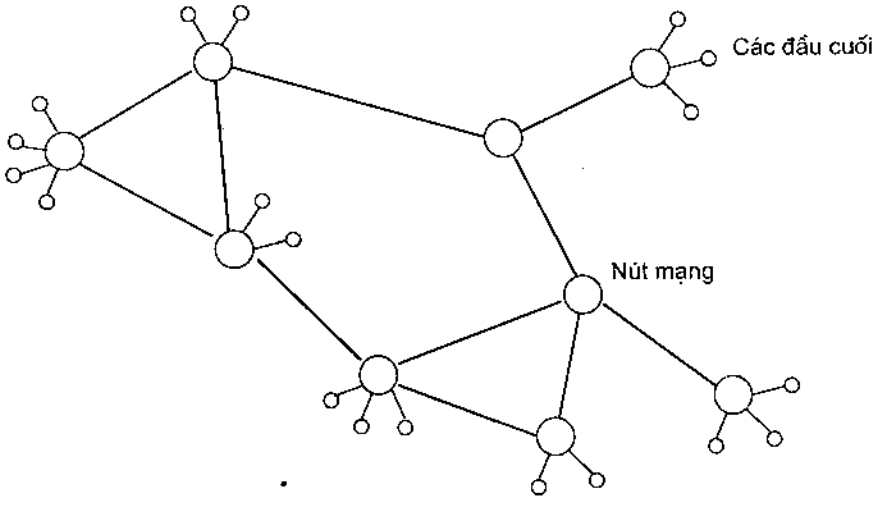
8.2. VỊ TRÍ CỦA MẬT MÃ TRONG CẤU TRÚC MẠNG TRUYỀN TIN

8.2.1. Giới thiệu tổng quan

Một mạng truyền tin số hoặc một mạng máy tính được xem như bao gồm các trung tâm chuyển mạch, thường được gọi là các nút mạng, nối với nhau bởi các đường truyền tin và các nút mạng đó lại được kết nối với các bộ ghép kênh hoặc các bộ tập trung để dẫn đến trung tâm (hoặc máy tính chủ) và các đầu cuối của mạng. Trong một cấu trúc tương đối phức tạp như vậy có thể có nhiều phương thức mật mã được sử dụng với các cấp độ khác nhau.

Một khi mà việc mật mã được sử dụng trong một mạng thì mức bảo mật mà mật mã đó cung cấp phụ thuộc vào lớp đang được thực hiện. Theo cấu trúc mô hình OSI thì dịch vụ đó có thể thực hiện trong bất kỳ lớp nào của sáu lớp phụ thuộc vào cấu trúc của mạng và chức năng các bộ phận khác nhau của mạng. Ví dụ, trong một mạng riêng, các nút mạng có thể xem như đảm bảo không nguy hiểm không giống như các nút mạng trong một mạng công cộng. Bởi vì ở đây xem rằng, mạng máy tính cũng như mạng truyền tin số, cho nên có ba dạng mật mã sau đây được sử dụng: mật mã

đường truyền, mật mã đầu cuối - đầu cuối và mật mã nút - nút. Hình 8.1 mô tả ví dụ một mạng điển hình gồm một tập hợp các trung tâm chuyển mạch hoặc các nút được nối với nhau bởi các kênh truyền tin và là nơi dễ bị xâm nhập nhất trong hệ thống.



Hình 8.1. *Mô tả một mạng gồm các nút mạng (chuyển mạch, ghép kênh...)*

8.2.2. Mật mã đường kết nối giữa các nút mạng

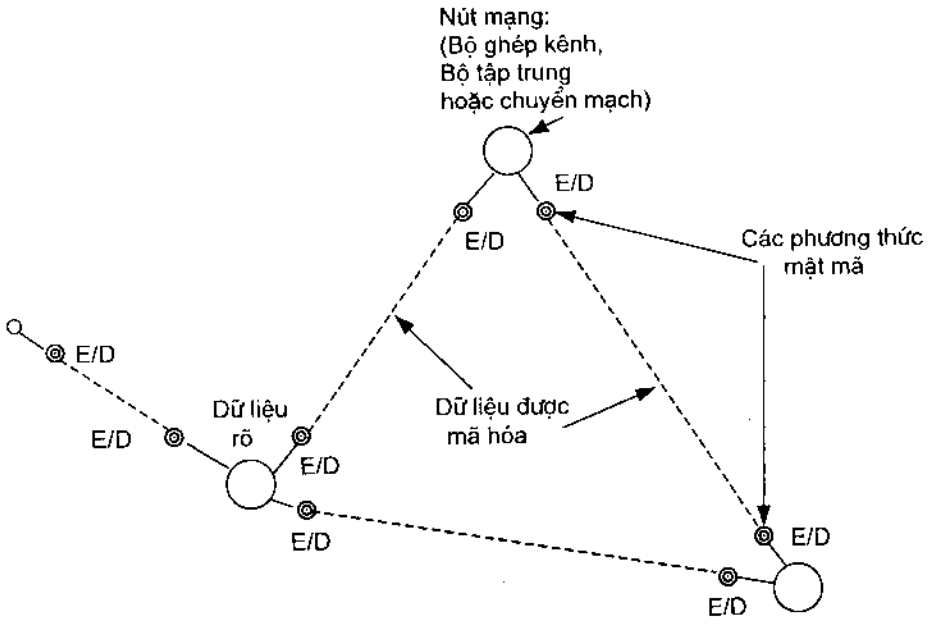
Đường kết nối giữa các nút mạng thực hiện việc truyền tin giữa các nút mạng theo một thể thức hoặc một thủ tục truyền tin nào đó. Quá trình sử dụng mật mã để bảo vệ đường kết nối được gọi là mật mã đường kết nối (hoặc kênh kết nối). Thông tin truyền trên đường kết nối có thể dưới dạng bit hoặc dạng ký tự từ nút này đến nút khác và việc sử dụng mật mã không quan tâm đến chi tiết cấu trúc các đoạn tin, ví dụ cấu trúc tiêu đề, mào đầu, kết thúc như thế nào.

Mật mã đường kết nối là các mật mã ứng dụng trong lớp 1 (vật lý) và lớp 2 (liên kết dữ liệu) của mô hình mở ISO - 7 lớp. Hình 8.2 mô tả một mật mã đường kết nối.

Nếu thực hiện mật mã ở mức thấp, ví dụ bằng cách xử lý các bit hoặc các ký tự qua đường truyền thì kẻ xâm nhập đường truyền cũng không biết gì về cấu trúc thông tin cũng như không biết gì về bên gửi cũng như bên nhận. Trường hợp này còn được gọi là "bảo vệ bí mật dòng dữ liệu", có nghĩa là không những che dấu thông tin mà còn che dấu cả đường đi và khối lượng thông tin truyền. Cũng có thể bảo vệ bí mật dòng dữ liệu bằng cách thực hiện mật mã trong các lớp thấp nhất.

Nếu phương thức truyền tin trên đường truyền là đồng bộ thì trên đường truyền có dãy bit được truyền. Dãy bit đó có thể được mã hóa theo phương thức mật mã CFB cho 1 bit. Sau khi khởi đầu, tiến trình sẽ tiếp tục thực hiện xem như đường truyền mở, có sửa lỗi bit tự động và đồng bộ. Kẻ xâm nhập trong trường hợp đó chỉ thu được một dãy ngẫu nhiên các bit mà không thể biết rằng trong đó có dữ liệu hoặc không có. Do không có các thông tin khởi đầu và kết thúc của đoạn tin cho nên kẻ xâm nhập không thể thực hiện các xâm nhập tích cực một cách hiệu quả. Tuy vậy, trong

trường hợp này nếu kẻ xâm nhập làm thay đổi một phần bản tin đã được mã hóa thì sẽ gây ra các biến đổi ngẫu nhiên ở phần giải mã. Khắc phục hiện tượng này ở các lớp thấp rất khó và chỉ có thể thực hiện ở các lớp cao hơn với cơ cấu mã sửa lỗi được lựa chọn thích hợp.



Hình 8.2. ***Mô tả mật mã đường kết nối***

Tình hình sẽ không sáng sủa hơn nếu phương thức truyền tin là không đồng bộ (có các bit khởi đầu và các bit kết thúc ký tự). Trong trường hợp đó, toàn bộ nội dung của đoạn tin có thể được mã hóa theo phương thức mật mã CFB cho 1 bit hoặc 1 ký tự, và giữ nguyên các bit khởi đầu cũng như các bit kết thúc. Đó là một kẽ hở mà kẻ xâm nhập có thể biết được cách đọc và phân tích các ký tự truyền. Trong trường hợp này, để che giấu thông tin, hệ thống cần có cơ cấu phụ trợ tạo ra các đoạn tin vô nghĩa ở các chu kỳ nghỉ. Các đoạn tin vô nghĩa đó thường được tạo ra tại các nút mạng.

Nếu phương thức truyền tin là chuyển mạch gói thì một đường logic có thể thực hiện chức năng đó.

Các phân tích ở trên là thuộc lớp vật lý. Trong lớp liên kết dữ liệu (lớp 2) có các thể thức ví dụ như: thể thức truyền tin đồng bộ nhị phân (BISYNC), điều khiển liên kết dữ liệu đồng bộ (SDLC), điều khiển dữ liệu mức cao (HDLC) hoặc truyền không đồng bộ (ATM) thì cần xem xét trong từng trường hợp cụ thể.

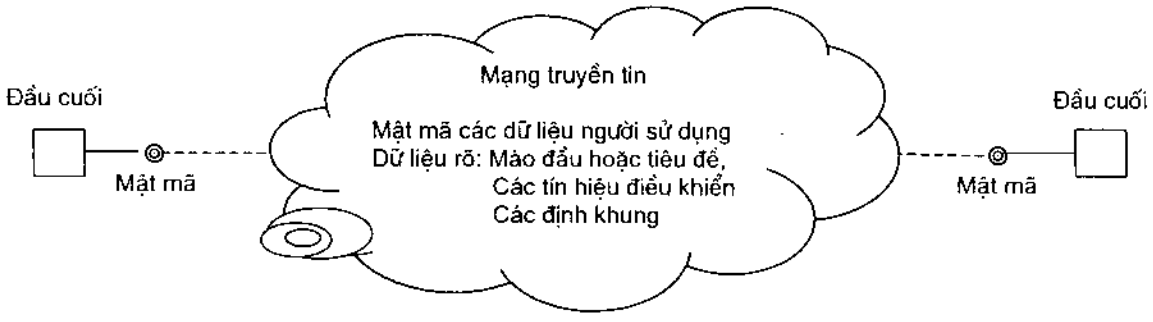
Việc phân phối khóa mã trong các lớp thấp không có vấn đề gì lớn. Việc trao đổi đó chỉ có quan hệ giữa hai nút liên hệ với nhau và được xem như là công việc nội bộ của hai bên.

Qua các lớp phân tích trên có nhận xét rằng, việc sử dụng mật mã ở các lớp thấp có ưu điểm là đảm bảo bí mật dòng các dữ liệu (lưu lượng) nhưng có nhược điểm là chỉ bảo vệ các dữ liệu trên đường truyền chứ không ngăn được giữa hai người sử dụng mạng. Do đó có thể tận dụng các ưu điểm của lớp thấp có và khắc phục các nhược điểm của chúng bằng cách sử dụng mật mã ở các lớp cao hơn. Cụ thể là, nếu sử dụng mật mã đường kết nối thì nên đặt ngay trong lớp vật lý, có nghĩa là áp dụng trực tiếp bit hoặc ký tự, không quan tâm đến cấu trúc và thể thức kết nối.

8.2.3. Mật mã từ đầu cuối đến đầu cuối

Hình 8.3 mô tả mô hình mật mã từ đầu cuối đến đầu cuối trong đó bố trí mật mã được đặt ở cổng của đầu cuối kết nối với mạng truyền tin.

Tiến trình thực hiện mật mã ở đây cần phải tính đến các thể thức truyền tin giữa các đầu cuối và mạng. Một cuộc gọi từ đầu cuối này đến đầu cuối kia để thiết lập kết nối có thể có nhiều phương thức kết nối, phụ thuộc vào thể thức truyền thông. Các kết nối đó có thể là kết nối thực (chiếm kênh) hoặc kết nối ảo (không chiếm kênh) để trao đổi dữ liệu.



Hình 8.3. ***Mô tả mật mã từ đầu cuối đến đầu cuối***

Có một số dữ liệu trên đường truyền chỉ được dành riêng cho các điều khiển của mạng mà không phải cho các đầu cuối, do đó nó không cần phải mã hoá, không cần phải dùng đến khóa mã, để cho mạng và các nút chuyển tiếp nhận biết thể thức truyền tin. Cũng vì vậy, trong thực tế thường phân làm hai loại dữ liệu: "dữ liệu của người sử dụng" cho đầu cuối từ xa và các dữ liệu khác dành cho truyền tin.

Ví dụ trong truyền tin theo phương thức chuyển mạch gói, có thể sử dụng giao thức X25. Giao thức X25 là "giao diện giữa một thiết bị đầu cuối (DTE) và một thiết bị truyền tin (DCE). Nó bao hàm việc sử dụng các gói dữ liệu làm cơ sở để tạo ra một đường ảo. Với một đường ảo, hai người sử dụng sẽ có một kết nối riêng giữa họ mặc dù thực ra họ đang dùng chung kênh vật lý với những người khác. Điều này đạt được bằng cách cắt chuỗi dữ liệu nối tiếp của từng người sử dụng thành các "gói" dữ liệu có chiều dài xác định. Mỗi gói đó được gắn thêm tiêu đề vào trong đó có địa chỉ đích của gói và các thông tin điều khiển khác. Các dữ liệu này không thuộc người sử dụng mà thuộc mạng truyền tin. Các gói của người sử dụng được gắn thêm các dữ liệu khác đó được đặt rải rác giữa các gói của người sử dụng khác trên đường truyền tin theo phương thức chuyển mạch gói sử dụng giao diện X25.

Lớp thấp nhất của giao diện X25 là lớp vật lý và trên lớp vật lý là lớp liên kết dữ liệu sử dụng cơ cấu điều khiển liên kết dữ liệu mức cao (HDLC) để truyền các gói. Ở lớp 3, các gói đã được hình thành và có khoảng 26 loại gói khác nhau.

Nếu như việc mật mã được thực hiện ở khâu giữa đầu cuối và mạng như mô tả ở hình 8.3 thì cấu trúc đó cần phải biểu thị một phần của giao diện X25 để có thể phân biệt hai loại trường "dữ liệu người sử dụng" đã được mã hóa đó. Trong thực tế, mật mã chỉ cần thực hiện trên các dữ liệu trước khi chúng được đưa vào giao diện X25 của đầu cuối, mục đích là để giảm độ phức tạp. Cũng

có thể có những cuộc gọi ảo khác nhau được thực hiện với cùng một giao diện X25 nhưng sử dụng khóa mã khác nhau, bởi vì với các người sử dụng khác nhau.

Việc thực hiện mật mã trong các phương thức kết nối ảo cần có các phần tử của các thể thức phụ trợ. Trong thực tế các phần tử của thể thức phối hợp này cũng hơi phức tạp. Cũng có thể xuất phát từ một giao thức dành riêng cho việc bảo mật, ví dụ có một sự cải biên hoặc sự thêm bớt nào đó trong các giao thức ở lớp trên. Tất cả các bước nhằm thiết lập một đường kết nối ảo là không được mã hoá. "Các gói cuộc gọi và các gói thu" có thể bị kẻ xâm nhập phân tích khi chúng được truyền qua mạng và chúng có thể có thông tin về trạng thái cuộc gọi và số lượng thông tin được truyền tin. Ở đây chưa có bảo mật dòng lưu lượng trong mật mã từ đầu cuối đến đầu cuối. Một giải pháp tốt nhất cho vấn đề này là sử dụng mật mã hai lần, một lần ở các lớp thấp và một lần ở các lớp cao với các dịch vụ được bổ sung. Như vậy, việc áp dụng mật mã từ đầu cuối đến đầu cuối bảo vệ "các dữ liệu người sử dụng" trong mạng, còn mật mã đường kết nối bảo vệ bí mật dòng lưu lượng trên các tuyến.

Việc bảo vệ thông tin dòng lưu lượng ở đây có khác với trường hợp sử dụng mật mã trên đường kết nối giữa các nút mạng, bởi vì ở đây còn phải định tuyến và các thông tin về lưu lượng là đã có sẵn. Cũng có thể có một vài phương pháp bảo vệ khác, ví dụ sử dụng mật mã hoặc phân chia khóa mã cho người sử dụng tại bộ định tuyến. Các trung tâm định tuyến cũng có thể tạo ra một lưu lượng giả tạo để che dấu lưu lượng thật. Một mô hình như vậy đòi hỏi những người sử dụng phải hoàn toàn tin tưởng vào trung tâm định tuyến.

Các chức năng mã hóa và giải mã ở hai phía đầu cuối như mô tả ở hình 8.3 đòi hỏi phải làm việc cùng khóa mã. Do đó mà mỗi cặp đầu cuối muốn trao đổi các đoạn tin thì bằng cách này hoặc cách khác phải có khóa mã giống nhau. Nếu một mạng có N đầu cuối thì phải có N(N-1)/2 khóa mã. Ở chương 3 đã giới thiệu các phương pháp phân phối và quản lý các khóa mã nhưng chỉ với các khóa phiên làm việc với N khóa chính. Bài toán phân phối khóa ở đây sẽ phức tạp hơn nhiều. Sự gia tăng các khóa đó chính là một nhược điểm trong mật mã từ đầu cuối đến đầu cuối. Ở đây, nếu như một gói nào đó bị thất lạc đến một nơi nào đó thì ở đó cũng không thể giải mã được.

8.2.4. Mật mã từ nút mạng đến nút mạng

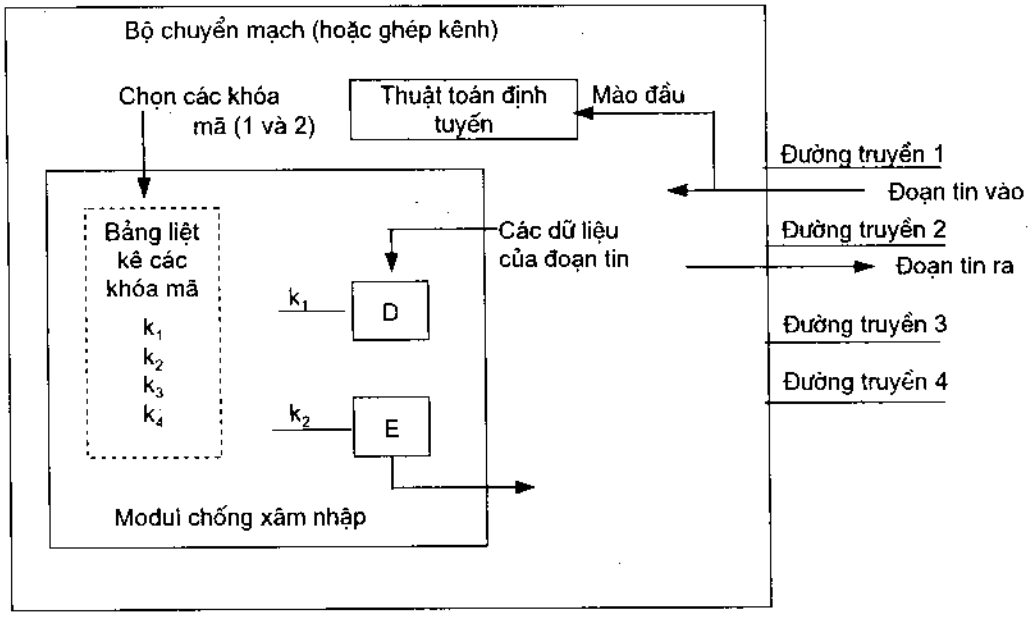
Có một loại mật mã khác của mạng được gọi là "mật mã từ nút mạng đến nút mạng". Loại mật mã này có điểm giống với mật mã đường kết nối là mỗi một khóa mã được gắn với một đường truyền cụ thể nhưng bài toán phân phối khóa đơn giản hơn. Điều này có thể dẫn đến đoạn tin gửi có thể bị sai lạc địa chỉ nhận, tuy nhiên ở đây yếu tố về phân phối khóa được đặt lên hàng đầu.

Ở đây việc mã hóa (ngược với mật mã đường kết nối) chỉ thực hiện với các dữ liệu người sử dụng còn các dữ liệu về điều khiển và tiêu đề không mã hoá. Như vậy cho phép các chuyển mạch, các bộ ghép kênh, các bộ tập trung xử lý các đoạn tin dưới dạng rõ mà không truy nhập vào các dữ liệu của những người sử dụng.

Vấn đề mã hóa và giải mã ở đây với các khóa mã khác nhau khi các dữ liệu đi vào một đường truyền này và đi ra một đường truyền khác. Chức năng đó được thực hiện bởi một modul vật lý bảo mật đặt ở nút mạng và được bảo vệ đặc biệt. Modul bảo mật đó chứa tất cả các khóa mã mà nút

mạng cần sử dụng cho mỗi đường truyền của nó, Hình 8.4 mô tả nguyên lý thực hiện mật mã từ nút mạng đến nút mạng.

Để thực hiện việc giải mã thì modul bảo mật phải nhận biết các đường truyền vào và ra mỗi khi nút mạng thực hiện việc định tuyến. Với các dữ liệu đó, modul chọn ra các khóa cần thiết, giải mã dòng các dữ liệu với khóa mã liên quan với đường truyền đầu vào và mã hóa lại với khóa mã liên quan với đường truyền đầu ra. Các dữ liệu của người sử dụng không bao giờ xuất hiện ở dạng rõ trong các phần không được bảo vệ của nút mạng và như vậy có thể tránh được các nguy hiểm gặp phải ở mật mã đường kết nối. Tuy vậy, ở đây cũng có một nhược điểm khác là, tất cả các tiêu đề và các dữ liệu định tuyến đều được truyền trong mạng dưới dạng rõ. Cũng chính vì vậy mà mật mã từ nút mạng đến nút mạng cũng chỉ được ứng dụng trong một số trường hợp lựa chọn, tuỳ thuộc vào yêu cầu bảo mật cụ thể.



Hình 8.4. ***Mô tả nguyên lý mật mã từ nút mạng đến nút mạng.***

8.2.5. Mật mã trong lớp vật lý của các đường truyền dữ liệu

8.2.5.1. Vị trí thiết bị mật mã trong lớp vật lý và một số chuẩn

Lớp vật lý là lớp thấp nhất trong mô hình cấu trúc mở 7 lớp OSI. Một trong những chuẩn đầu tiên về bảo vệ thông tin dữ liệu có liên quan đến lớp vật lý là chuẩn ISO 9160 về "xử lý thông tin - mật mã dữ liệu - các đặc tính chức năng trong lớp vật lý". Thông thường, mật mã chỉ có thể hoạt động ở mức liên kết. Tuy vậy việc mật mã trong lớp vật lý cũng có ưu điểm là mã hóa tất cả các bit dữ liệu và điều đó đảm bảo độ tin cậy của dòng dữ liệu. Chuẩn được phân theo phương thức truyền đồng bộ và không đồng bộ. Ở phương thức truyền đồng bộ thì mỗi phần tử truyền được mã hoá. Ở phương thức truyền không đồng bộ theo ký tự thì các bit khởi đầu (start) và kết thúc (stop) của các ký tự truyền cần phải được giữ nguyên, chỉ cần mã hóa các bit dữ liệu (thông thường là 8 bit). Vì

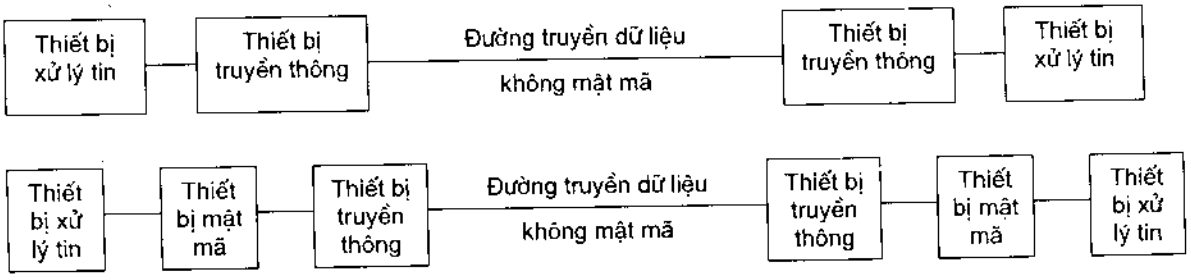
vậy các ký tự dạng start/stop xuất hiện trên đường truyền có thể nhận biết theo ký tự. Để đảm bảo độ tin cậy dòng dữ liệu có thể thêm vào một dòng giả định.

Khi phát triển chuẩn ISO- 9160 thì người ta cũng đã dự tính trước việc sử dụng thuật toán mật mã DES theo phương pháp phản hồi từng bit một của bản tin được mã hoá. Các chuẩn cũng luôn được thay đổi. Việc phản hồi có thể thực hiện theo 1 bit hoặc theo 8 bit.

Ở phương thức truyền tin đồng bộ thì việc phản hồi trên 1 bit là lý tưởng, bởi vì nó phù hợp với tất cả các bài toán của các dữ liệu được đóng khung. Còn ở truyền không đồng bộ theo ký tự 8 bit thì việc thực hiện phản hồi có thể lựa chọn theo bit hoặc theo 8 bit. Các hệ thống truyền tin hiện nay thích hợp với phương pháp phản hồi 1 bit, bởi vì nó đơn giản hơn, không phải thay đổi giao thức, chỉ cần các bổ sung để thiết lập kênh trước khi truyền.

Việc sử dụng phương pháp phản hồi trong bản tin đã được mã hóa riêng biệt theo chuẩn ISO-8372 cần phải truyền trước giá trị khởi đầu IV.

Hình 8.5 mô tả một đường kết nối truyền dữ liệu không sử dụng thiết bị mật mã và có sử dụng thiết bị mật mã. Trong mô hình kết nối đó, thiết bị mật mã được đặt giữa thiết bị xử lý tin và thiết bị truyền thông, ví dụ modem hoặc các thiết bị thu phát trong trường hợp kênh truyền là vô tuyến. Cũng vì vậy mà các thiết bị mật mã phải có khả năng vừa giao diện với thiết bị xử lý tin, vừa có khả năng giao diện với thiết bị truyền thông. Trong thực tế ứng dụng, để đảm bảo kinh tế có thể đưa mật mã vào một trong hai thiết bị xử lý tin hoặc thiết bị truyền thông để chỉ cần một giao diện giữa thiết bị xử lý tin và thiết bị truyền thông.

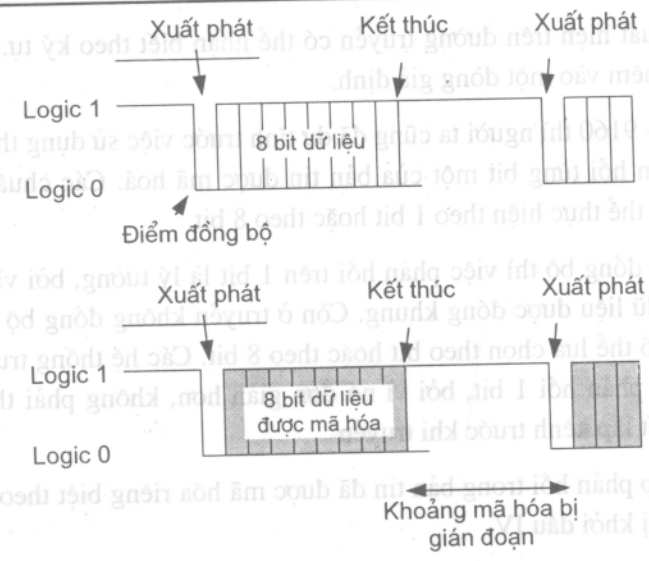


Hình 8.5. ***Bố trí thiết bị mật mã của lớp vật lý.***

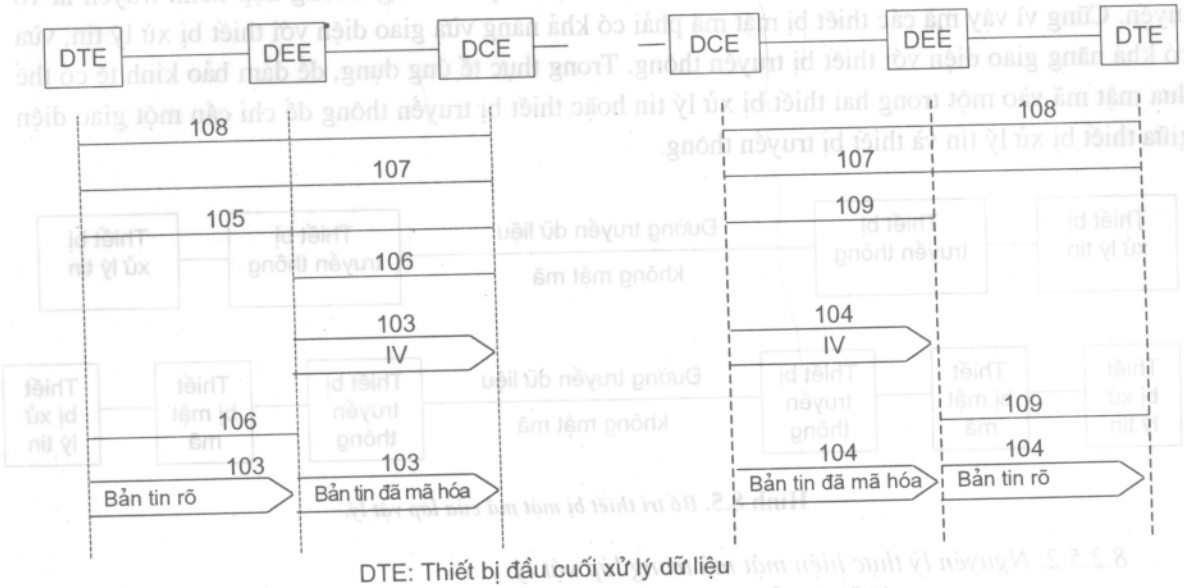
8.2.5.2. Nguyên lý thực hiện mật mã trong lớp vật lý

Việc mật mã trong lớp vật lý có thể thực hiện theo bít hoặc theo ký tự. Để có sự đồng nhất các bit, tất cả các phiên bản đều có một yêu cầu chung là thực hiện mật mã CFB trên một bit. Trong trường hợp truyền đồng bộ thì mỗi bit truyền được mã hóa theo phương thức trên. Nếu là truyền không đồng bộ theo ký tự thì các bit Start và Stop phải giữ nguyên và chỉ mã hóa các bit dữ liệu giữa chúng. Quá trình mã trong lớp vật lý của phương thức truyền không đồng bộ theo ký tự. Lưu ý rằng, trước khi việc mã hóa và giải mã bắt đầu thì phải truyền giá trị khởi đầu IV để tiến hành mật mã theo CFB.

Việc truyền có thể bắt đầu ngay sau khi trao đổi các tín hiệu điều khiển cho phép thiết lập kết nối qua giao diện thiết bị xử lý tin và thiết bị truyền thông. Tuỳ thuộc vào yêu cầu truyền tin và kênh truyền tin, các giao diện có thể là V24, X20, X21 hoặc X25... Hình 8.7. mô tả các tín hiệu khi sử dụng giao diện V24.



Hình 8.6. *Mật mã dữ liệu truyền không đồng bộ theo ký tự trong lớp vật lý*



DTE: Thiết bị đầu cuối xử lý dữ liệu
DEE: Thiết bị đầu cuối mã hóa hoặc giải mã dữ liệu
DCE: Thiết bị đầu cuối truyền thông
IV: Giá trị khởi đầu

Hình 8.7. *Mô tả trao đổi điều khiển trong quá trình thiết lập kết nối của giao diện V24 có sử dụng thiết bị mật mã*

Ở giao diện V24, giá trị khởi đầu IV được gửi trên đường 103 (đường gửi dữ liệu đi) và nhận ở đường 104 (đường nhận các dữ liệu đến).

Ở phương thức tryền đồng bộ, giá trị khởi đầu IV chỉ truyền trong một khối dữ liệu.

Việc sử dụng mật mã trong lớp vật lý cần chú ý đến các ảnh hưởng của nó đến độ trễ và tốc độ truyền dữ liệu. ISO có các chuẩn khuyến nghị cho các mức độ ứng dụng.

8.3. XÁC THỰC CÁC THỰC THỂ TƯƠNG ĐỒNG

Trong lĩnh vực truyền tin có bảo mật thường gặp hai loại xác thực, đó là xác thực các đoạn tin (đảm bảo tính toàn vẹn) và xác thực các thực thể tương đồng. Việc xác thực các đoạn tin đã được phân tích trong chương 7. Mục đích của việc xác thực các thực thể tương đồng là cho phép các thực thể truyền tin (ví dụ các quá trình ứng dụng) có một sự đảm bảo chắc chắn tương hỗ lẫn nhau.

Để thực hiện công việc đó, có thể có nhiều phương pháp khác nhau, nhưng ý tưởng cơ bản của tất cả các hệ thống là một thực thể chứng minh thực thể kia cũng đang nắm giữ một số bí mật. Để thực hiện việc xác thực có thể sử dụng các hệ thống mật mã có khóa bí mật hoặc mật mã có khóa công khai. Có bốn cơ cấu khác nhau thường được đồng nhất, đó là:

1. Cơ cấu thứ nhất dự đoán rằng mỗi thực thể truyền tin hợp pháp là đang sở hữu cùng khóa bí mật trước khi việc xác thực được thực hiện.

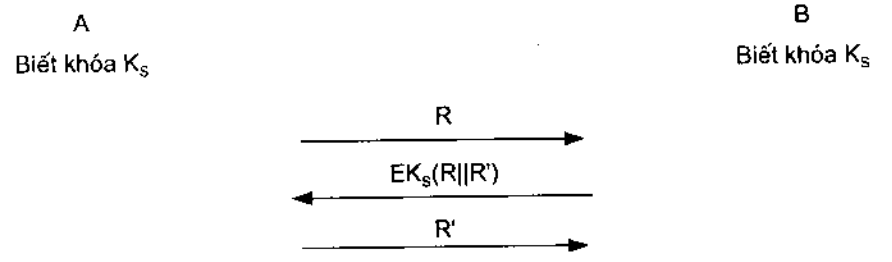
2. Cơ cấu thứ hai không dựa vào cùng nguyên lý là ở xuất phát không có khóa chung. Nhưng một bộ ba (một trung tâm xác thực) được sử dụng.

3. Cơ cấu thứ ba là sự cải tiến của cơ cấu thứ hai, giả thiết rằng các bên có bố trí đồng hồ. Các kiểu gian trá về trò chơi các đoạn tin dễ bị phát hiện.

4. Cơ cấu thứ tư được ghi một cách đặc biệt để tối thiểu hóa các thông tin bí mật trao đổi giữa các thực thể muốn được xác thực.

Để mô tả nguyên lý xác thực các thực thể tương đồng, sau đây sẽ giải thích cơ cấu thứ nhất của các cơ cấu đó.

Thực thể A, khởi đầu quá trình, tạo một số ngẫu nhiên R (như mô tả ở hình 8.8) và gửi rõ cho thực thể B. Thực thể B chuẩn bị một trả lời trong đó số ngẫu nhiên R được biến đổi với một số ngẫu nhiên R' do B tạo ra. Trước khi gửi trả lời, B mã hóa nó với khóa K_s, trong đó A và B không có sự thỏa thuận trước với nhau. Khi bên A nhận được trả lời đã mã hóa của B, thì A có thể giải mã với khóa K_s để có số ngẫu nhiên ban đầu R của A và số ngẫu nhiên mới R' của B. Để hoàn thiện việc xác thực, A sau đó gửi số ngẫu nhiên rõ R' cho B. Kết quả là các thực thể A và B, cả hai đều đảm bảo rằng thực thể mà mình quan hệ có cùng khóa K_s. Việc xác thực đó dựa trên sự nhận biết được phân chia. Tất nhiên ở đây việc có trước các khóa K_s của hai thực thể phải được bảo mật để tránh có một phần tử xâm nhập thứ ba nắm bắt được. Ba cơ cấu khác có phức tạp hợp nhưng cũng dựa vào nguyên lý trên.



Hình 8.8. *Mô tả một phương pháp xác thực các thực thể tương đồng bằng một thuật toán dùng khóa bí mật.*

Việc xác thực các thực thể tương đồng dựa vào các thuật toán mật mã dùng khóa công khai cũng thường được sử dụng.

8.4. MẬT MÃ TRỰC TUYẾN TRÊN ĐƯỜNG TRUYỀN TIN SỐ, CHUYỂN MẠCH GÓI VÀ CÓ GHÉP KÊNH

8.4.1. Mật mã trường thông tin, không mật mã trường tiêu đề

Các đường điện thoại và mạng điện thoại truyền thống đã được xây dựng từ lâu. Với mục đích kinh tế, các mạng truyền dữ liệu, các mạng máy tính diện rộng, mạng kết nối Internet thường được kết hợp và xây dựng trên cơ sở các mạng điện thoại. Các hệ thống điện thoại tương tự (analog) đã lỗi thời được thay thế bằng các hệ thống điện thoại số, chuyển mạch số và gói. Các tín hiệu thoại số hoá, dữ liệu, hình ảnh thường được truyền trên cùng một kênh truyền. Công nghệ kỹ thuật số chuyển mạch gói đồng bộ (STM) và không đồng bộ (ATM) đang được ứng dụng rộng rãi. Các bộ ghép kênh hoặc phân kênh được đặt ở các nút mạng hoặc trung tâm chuyển mạch để ghép kênh (dồn kênh) từ tốc độ thấp lên tốc độ cao hoặc phân kênh từ tốc độ cao về tốc độ thấp. Để có thể ghép kênh, dữ liệu được định khung (framing) và có các tốc độ được phân cấp theo các chuẩn tương ứng.

Theo định lý Nyquist, tốc độ lấy mẫu tín hiệu thoại là 8000 mẫu/s, mã PCM là 8 bit/mẫu, do đó tốc độ khởi đầu của các kênh tín hiệu số, ký hiệu là DS-O, là:

$$\frac{8000\text{mẫu}}{\text{s}} \cdot \frac{8\text{bit}}{\text{mẫu}} = 64 \text{ kbps}$$

Có hai chuẩn ghép kênh được sử dụng phổ biến cho tất cả các nước trên thế giới là chuẩn của Mỹ và chuẩn của châu Âu.

Cấp 1 theo chuẩn của Mỹ (DS-1) ghép 24 kênh DS-O để có tốc độ là:

$$64 \text{ kbps}.24 = 1{,}536 \text{ Mbps}.$$

Cấp 1 theo chuẩn của châu Âu ghép 32 kênh (30 kênh thông tin và 2 kênh điều khiển) để có luồng E1 tốc độ là 2,048 Mbps.

Cấu trúc khung theo chuẩn của Mỹ là 192 bit, cộng thêm một bit đồng bộ là 193 bit. Như vậy thực tế tốc độ truyền cấp 1 (DS-1) theo chuẩn của Mỹ là:

$$\frac{193}{\text{khung}} \cdot \frac{8000 \text{ khung}}{\text{s}} = 1{,}544 \text{ Mbps}$$

Cấu trúc khung theo chuẩn châu Âu là 256 bit được xác định như sau:

$$\frac{8 \text{ bit}}{\text{khe thời gian}} \cdot \frac{32 \text{ khe thời gian}}{\text{khung}} = \frac{256}{\text{khung}}$$

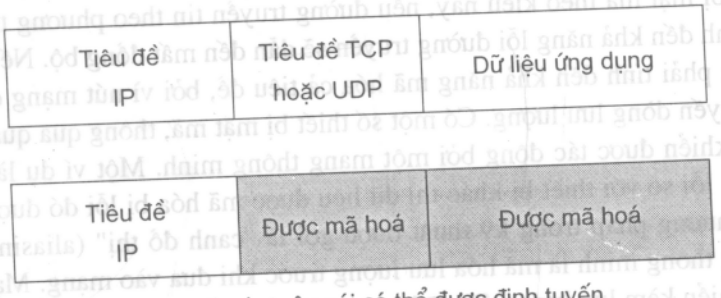
và tốc độ đường dây của luồng cấp 1 sẽ là:

$$\frac{256 \text{ bit}}{\text{khung}} \cdot \frac{8000 \text{ khung}}{\text{s}} = 2{,}048 \text{ Mbps}$$

Dữ liệu trước khi truyền được đóng gói. Cấu trúc gói tin gồm có hai trường: trường tiêu đề (header - chứa các thông tin về điều khiển, báo lỗi, địa chỉ...) và trường thông tin dữ liệu. Phụ thuộc vào giao thức truyền dẫn sử dụng mà cấu trúc và độ dài của các trường có khác nhau. Nếu như việc mật mã chỉ thực hiện mã hóa phần thông tin còn phần tiêu đề không mã hóa thì chỉ trường thông tin được mã hoá. Mật mã phần thông tin có thể sử dụng các mật mã, ví dụ DES hoặc mã có khóa công khai như đã trình bày ở các chương trước.

Trong truyền dữ liệu chuyển mạch gói, bởi vì đoạn tin được cắt thành các gói có địa chỉ đích và nguồn, cho nên các nút mạng hoặc bộ chuyển kênh phải có địa chỉ rõ để định tuyến các gói.

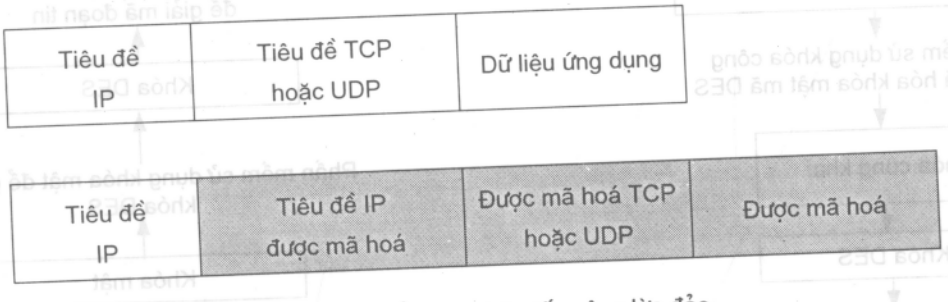
Việc mã hóa trường thông tin như trên chỉ có tác dụng bảo vệ nội dung thông tin truyền chứ không ngăn ngừa được các hành vi xâm nhập lừa đảo. Ví dụ như hệ thống truyền tin chuyển mạch gói sử dụng giao thức TCP/IP, nếu như được mã hóa cả phần trường thông tin và trường tiêu đề TCP (hoặc UDP) thì kẻ xâm nhập có thể khai thác địa chỉ rõ ở trường IP để đóng giả làm máy chủ hoặc thực hiện các hành vi lừa đảo khác. Hình 8.9 mô tả phương pháp mật mã trường thông tin và trường tiêu đề TCP.



- Tiêu đề IP vẫn để nguyên dạng rõ, như vậy gói có thể được định tuyến.
- Phương pháp mật mã này không bảo vệ địa chỉ nguồn và địa chỉ đích chống các tấn công lừa đảo.

Hình 8.9. ***Mật mã trường thông tin, không mật mã trường tiêu đề IP***

8.4.2. Mật mã trường thông tin và trường tiêu đề



- Ở đây tiêu đề IP được mã hoá, có thể ngăn chặn tấn công lừa đảo.
- Tiêu đề được giải mã sẽ được so sánh với tiêu đề rõ để phát hiện xâm nhập lừa đảo.
- Có nhược điểm là thêm tiêu đề

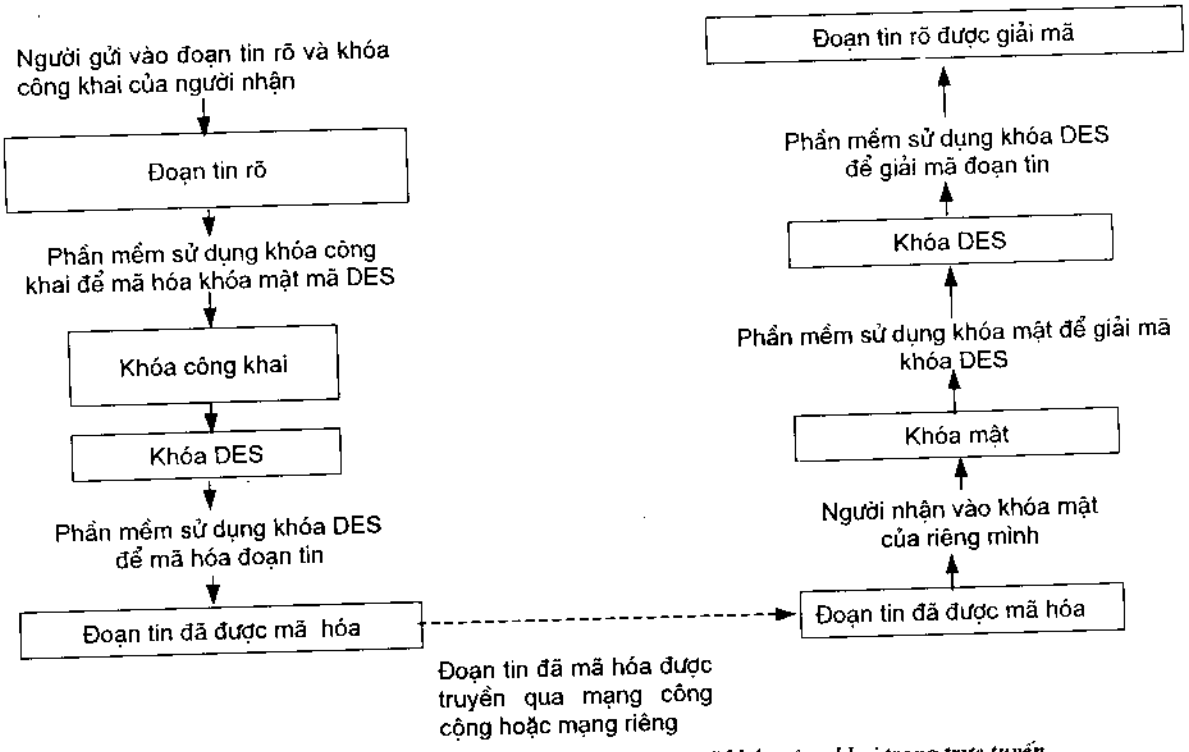
Hình 8.10. ***Mật mã trường thông tin và trường tiêu đề IP***

Để ngăn chặn các xâm nhập lợi dụng khai thác các thông tin rõ ở phần tiêu đề, có thể sử dụng phương pháp mật mã cả hai trường thông tin và tiêu đề. Như vậy dữ liệu truyền trong trường hợp này sẽ có hai trường tiêu đề: một trường tiêu đề được mật mã hóa và một trường tiêu đề rõ để

các nút có thể định tuyến. Tại phía nhận trường tiêu đề rõ sẽ được tách riêng ra, đoạn tin hoặc trường thông tin và trường tiêu đề mã hóa được giải mã. So sánh trường tiêu đề được giải mã và trường tiêu đề rõ, nếu chúng không phù hợp tức có một địa chỉ IP xâm nhập lừa đảo. Phương pháp được thực hiện giống như xác thực. Nhược điểm của phương pháp là phải sử dụng hai tiêu đề, do đó giảm tốc độ truyền tin. Hình 8.10 mô tả phương pháp mã hóa cả trường tiêu đề và trường thông tin.

8.5. PHỐI HỢP MẬT MÃ DES VÀ MẬT MÃ CÓ KHÓA CÔNG KHAI TRÊN ĐƯỜNG TRỰC TUYẾN

Trong nhiều trường hợp ứng dụng trong các hệ thống truyền tin, thuật toán DES thường được cài đặt sẵn trong các chip đặc biệt và được bố trí trong các thiết bị mật mã. Phần mềm được sử dụng để tăng cường. Một số trường hợp khác sử dụng kết hợp vừa mềm vừa cứng. Mật mã DES trực tuyến được sử dụng trong các mạng diện rộng, trong các hệ truyền tin có nhiều nút mạng như mô tả ở hình 8.2. Ở đây các thiết bị mật mã được bố trí giữa modem và thiết bị dịch vụ dữ liệu, thiết bị dịch vụ kênh hoặc thiết bị đầu cuối, trạm làm việc, máy tính chủ, bộ ghép kênh, thiết bị facsimile... Việc cài đặt các thiết bị mật mã theo kiểu này, nếu đường truyền tin theo phương thức đồng bộ, có định khung thì phải tính đến khả năng lỗi đường truyền sẽ dẫn đến mất đồng bộ. Nếu phương thức truyền gói có tiêu đề thì phải tính đến khả năng mã hóa cả tiêu đề, bởi vì nút mạng cần có địa chỉ rõ của tiêu đề để định tuyến dòng lưu lượng. Có một số thiết bị mật mã, thông qua quá trình ngẫu nhiên để tạo các mã điều khiển được tác động bởi một mạng thông minh. Một ví dụ là, khi có một thiết bị mật mã nào đó bị lỗi so với thiết bị khác thì dữ liệu được mã hóa bị lỗi đó được sử dụng cho thông tin điều khiển. Phương pháp trong kỹ thuật được gọi là "canh đồ thị" (aliasing). Cách tốt nhất để mật mã qua mạng thông minh là mã hóa lưu lượng trước khi đưa vào mạng. Mạng lúc đó có thể tạo ra một gói điều khiển kèm lưu lượng đã mã hoá.



Hình 8.11. ***Phối hợp mật mã DES và mật mã khóa công khai trong trực tuyến***

Các trạm làm việc trong mạng cũng có thể sử dụng thuật toán DES dưới dạng cứng hoặc mềm để bắt tay với các thiết bị bảo vệ cổng của một PBX hoặc một bộ chuyển đổi giao thức. Trong các ứng dụng thời gian thực thì việc ứng dụng thuật toán mật mã DES có ưu điểm là tốc độ nhanh hơn mật mã dùng khóa công khai RSA. Các hệ thống mật mã dùng khóa công khai cần phải đánh dấu đoạn tin và kiểm tra nó ở phía đầu cuối. Như đã giới thiệu trong phần phân phối khoá, có thể phối hợp giữa mật mã DES và RSA bằng cách sử dụng RSA để phân phối các khóa mã DES và sau đó sử dụng DES trực tuyến như mô tả ở hình 8.9.

Mật mã DES cần sử dụng nhiều khóa mã, trong lúc đó mật mã dùng khóa công khai sử dụng ít khóa mã hơn, do đó mà việc quản lý khóa trong trường hợp phối hợp hai mật mã này cũng đơn giản hơn. Các khóa mật mã DES cần phải giống nhau ở cả bên gửi và bên nhận và việc xác thực trong thực tế đã được thực hiện trước, việc mã hóa trong trường hợp này chỉ là bổ sung thêm.

8.6. PHÁT HIỆN XÂM NHẬP MẠNG

Phát hiện xâm nhập mạng là việc phán doán, rà xét hệ thống xem có một sự xâm nhập bất hợp pháp nào đó vào mạng hoặc có một sự sử dụng khai thác nào đó không hợp lệ. Công việc đó thường do một hệ thống kiểm tra phát hiện xâm nhập, IDS (Instrusion Detection System) đảm nhiệm. Hệ thống IDS được thiết kế để kiểm soát tài nguyên và hoạt động của mạng, thông báo cho bộ phận có trách nhiệm mỗi khi xác định được khả năng có sự xâm nhập. Nó hoạt động như một mạng lưới cảm biến để thông báo cho người quản trị mạng biết khi có ai đó đang xâm nhập, ở đâu và hình thức xâm nhập. Bức tường lửa là các vòng rào án ngữ vòng ngoài, vòng trong, nhưng một khi kẻ xâm nhập chui qua được các lỗ hổng của bức tường lửa thì hệ thống IDS có trách nhiệm phát hiện.

Trong thực tế, các hệ thống IDS được xây dựng theo hai hướng:

- Kiểm tra xâm nhập trên hệ thống máy chủ (host) (HIDS)

- Kiểm tra xâm nhập trên hướng mạng (NIDS).

HIDS: thực chất là một phần mềm được cài đặt trên hệ thống máy chủ để phát hiện các xâm nhập, kiểm tra những thay đổi trên hệ thống, đảm bảo cho những tập tin quan trọng không bị sửa đổi.

NIDS: kiểm tra mọi gói tin được trao đổi trên từng đoạn mạng, đánh dấu các gói tin bị nghi ngờ và có biểu hiện xâm nhập. Những dấu hiệu đáng ngờ có thể tìm thấy bằng cách xem xét mọi bit trong nội dung thực của gói tin và so sánh với từng bit tham chiếu đã biết về dạng tấn công, ví dụ tìm kiếm những kiểu tấn công tthay đổi các tập tin của hệ thống. Những kiểu tấn công cổng dịch vụ sẽ được xác định bằng cách theo dõi những cố gắng kết nối vào các cổng dịch vụ. Các dấu hiệu tấn công sẽ được phát hiện thông qua tiêu đề gói tin TCP/IP, ví dụ kẻ tấn công có thể gửi một gói tin yêu cầu đồng thời đóng và mở kết nối TCP, kiểu gói tin này có thể gây ra tình trạng nghẽn mạng, dạng tấn công khước từ dịch vụ DoS (Denial of Service).

Các hệ thống kiểm tra phát hiện kẻ xâm nhập cũng có thể phân loại theo tri thức hay hành vi.

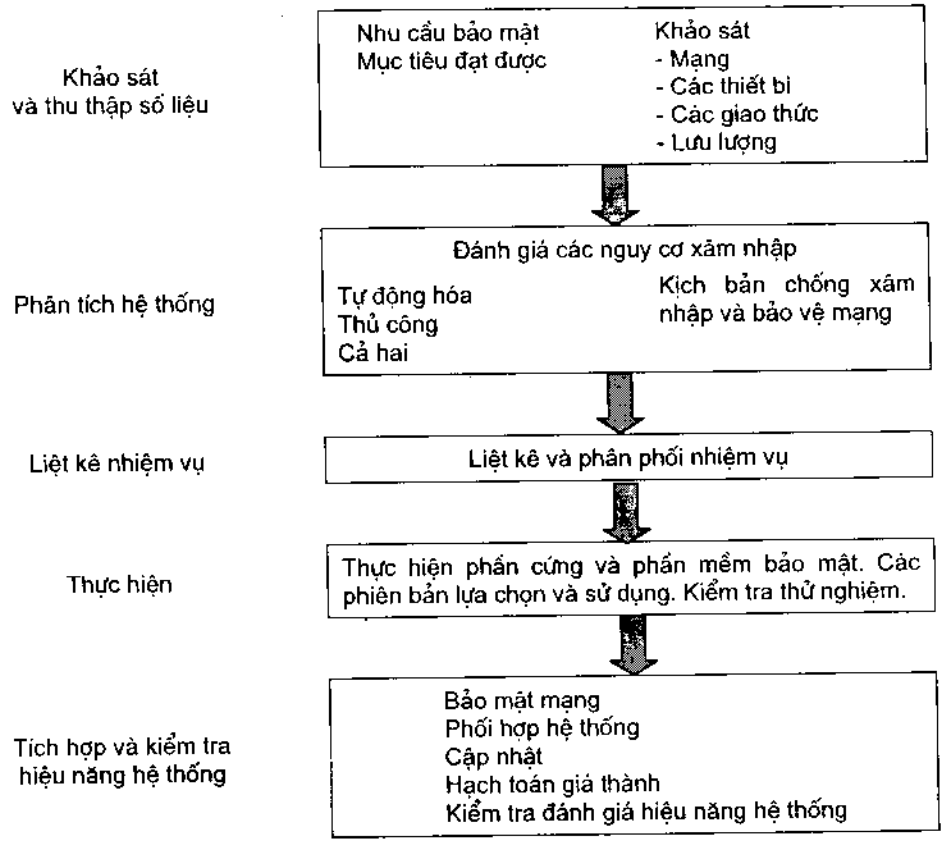
- Các hệ thống IDS có sẵn trên thị trường là dạng dựa trên "tri thức", hoạt động theo kiểu so sánh trùng các dấu hiệu tấn công thuộc dạng đã biết với những thay đổi của hệ thống hay các gói

tin trên mạng. Những hệ thống phát hiện kiểu này khá tin cậy và ít sai sót, nhưng chúng chỉ có khả năng phát hiện các kiểu tấn công đã được biết trước, chúng không có tác dụng đối với các dạng tấn công chưa được biết trước do đó hệ thống luôn cần được cập nhật các dạng tấn công mới.

- Các hệ thống IDS dựa vào "hành vi", xem xét các hoạt động bằng cách kiểm soát hoạt động của hệ thống hay mạng và đánh dấu bất kỳ hoạt động nào có nghi vấn. Những hành vi như vậy có thể kích hoạt sự cảnh báo, trong đó bao gồm những cảnh báo sai. Tuy vậy, trong những cảnh báo sai lại có khả năng phát hiện dạng tấn công xâm nhập mới.

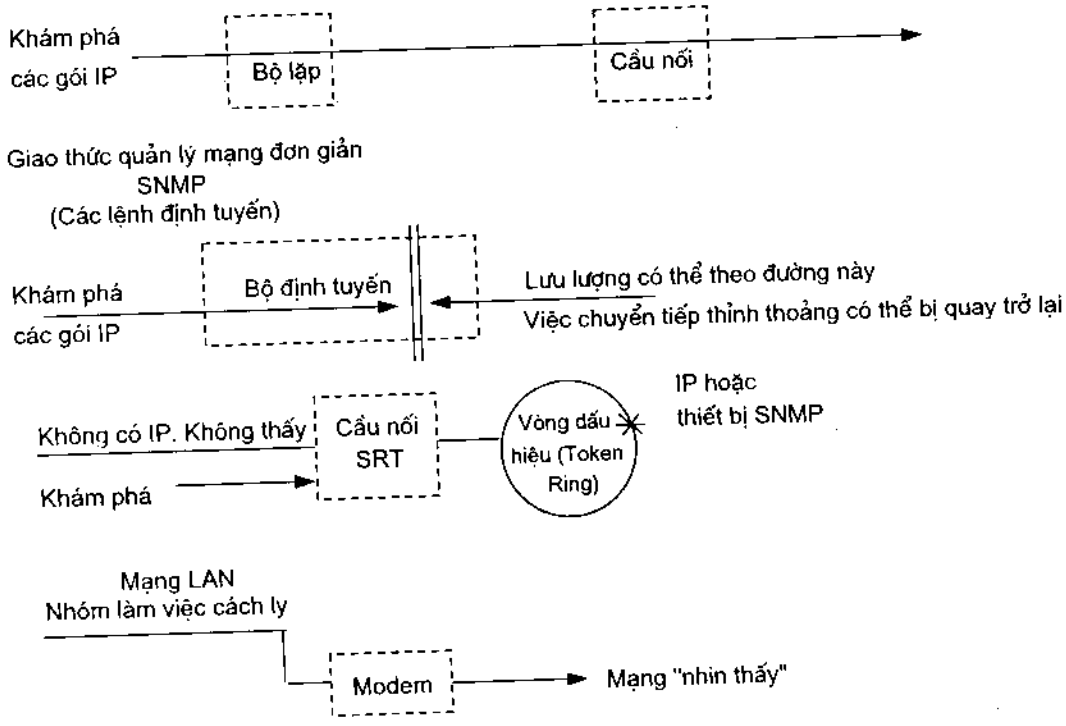
8.7. XÂY DỰNG MỘT HỆ BẢO MẬT MẠNG

Hình 8.12 mô tả các công việc cần thực hiện qua các bước của quá trình tổ chức xây dựng một hệ bảo mật cho một mạng truyền tin hoặc một mạng máy tính điển hình.



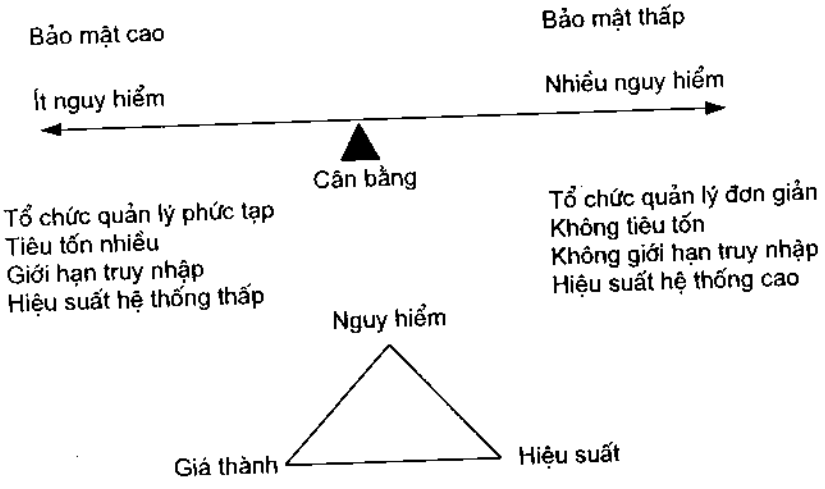
Hình 8.12. ***Các bước xây dựng hệ bảo mật mạng***

Có những mạng mà bằng các phương thức quản lý thông thường rất khó nhận biết các nguy cơ xâm nhập. Ví dụ các đường kết nối logic không hợp lý, phần lập trình không chuẩn xác ở một thời điểm nào đó tại các cầu nối, bộ định tuyến hoặc cổng chính có thể cho phép các kết nối logic sai. Các công cụ phần mềm khám phá như mô tả ở hình 8.13 có thể khai thác thăm dò môi trường mạng. Một số trong đó có thể là xâm nhập thụ động nhưng một số khác có thể là tấn công tích cực hoặc thăm dò khai thác thông qua các lệnh thông dụng. Các lệnh đó có thể trong bảng định tuyến giao thức quản lý mạng đơn giản (SNMP MIB) hoặc các bắt tay để tìm các thiết bị IP.



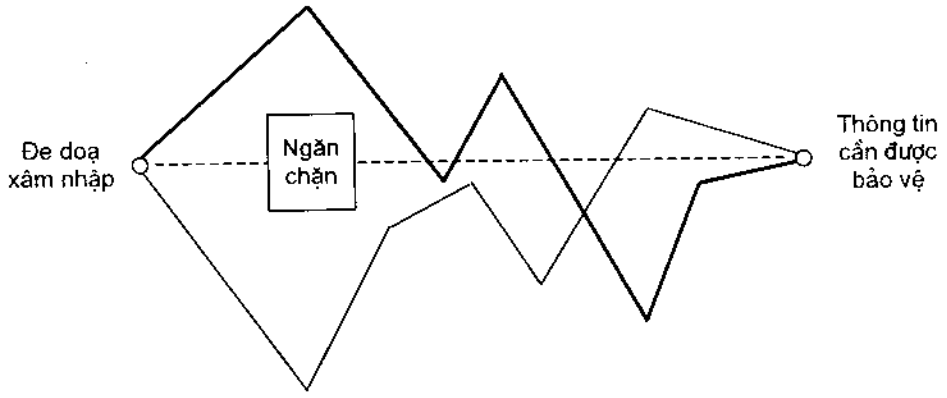
Hình 8.13. ***Các khối và các công cụ khám phá xâm nhập***

Thực ra rất khó mà đánh giá một hệ bảo mật được thiết kế như thế nào là hoàn hảo. Chúng có giới hạn tối thiểu và giới hạn tối đa. Một hệ không được bảo mật hoặc bảo mật ít là một hệ mở, không phải tốn kém và có thể tận dụng mọi khả năng của hệ thống. Một hệ có độ bảo mật cao thì số truy nhập bị hạn chế, tiêu tốn nhiều hơn và khả năng sử dụng khai thác hệ thống bị giới hạn. Vấn đề ở đây là người thiết kế cần căn cứ vào yêu cầu cụ thể của mạng và thông tin truyền hoặc lưu giữ cần được bảo vệ; dự đoán hết các nguy hiểm và khả năng xâm nhập, đe doạ an toàn hệ thống; các mức hoặc cấp bảo vệ, để trên cơ sở đó có một sự cân bằng thích hợp tối ưu. Hình 8-14 mô tả quá trình lựa chọn cân bằng đó. Quá trình lựa chọn cân bằng là một nghệ thuật đồng thời cũng là một khoa học.



Hình 8.14. ***Mô tả quá trình cân bằng khi xây dựng hệ bảo mật***

Các đe doạ xâm nhập đối với một mạng dữ liệu là khá lớn, bởi vì mạng dữ liệu trải rộng ra đến nhiều người sử dụng, nhiều người có thể truy cập mạng, rất khó kiểm soát. Các con đường đe doạ xâm nhập phần lớn là "zic-zắc" chứ không đơn giản trực tiếp qua các cổng kiểm soát. Hình 8.15 mô tả các đường "zic-zắc" xâm nhập. Cũng vì vậy mà quá trình thiết kế cần dự đoán hết các khả năng xâm nhập đó. Một số hệ bảo mật được xây dựng trên các tiêu chí về giá trị thông tin, giá trị thời gian tồn tại của thông tin và độ khó hoặc công sức bỏ ra của việc xâm nhập so với kết quả thu được.



Hình 8.15. *Mô tả các đường xâm nhập "zic-zắc" không qua trạm kiểm soát*

Chương 9

INTERNET, TCP/IP VÀ BỨC TƯỜNG LỬA

9.1. INTERNET

9.1.1. Giới thiệu

Internet là mạng toàn cầu của các máy tính và các mạng máy tính được kết nối với nhau. Nó là mạng của các mạng - kết nối các mạng của các trường học, thư viện, bệnh viện, trung tâm thương mại, các văn phòng nhà nước và các thực thể khác vào một mạng giao tiếp thông tin lớn, duy nhất kết nối toàn cầu. Những mạng này chứa nhiều kiểu máy tính khác nhau, vì vậy người ta đã thiết lập một phương pháp tổng quát chung, để chúng có thể kết nối được với nhau. Mạng Internet phục vụ cho việc chia sẻ các tài nguyên thông tin nhiều hơn là chia sẻ phần cứng.

Mạng Internet hoạt động có hiệu quả là nhờ vào các chương trình. Các chương trình cho phép người dùng nhận và gửi thư điện tử, truy cập vào mạng khác, chuyển các tệp dữ liệu từ các máy chủ (máy lưu trữ các thông tin), chia sẻ và thu thập các thông tin. Các chương trình này phân làm hai loại: trình chủ và trình khách. Trình chủ là chương trình cung cấp các dịch vụ, cư trú trên các máy tính chủ hoặc máy tính trung tâm. Trình khách là chương trình cư trú trên các máy tính riêng rẽ yêu cầu dịch vụ cho người dùng.

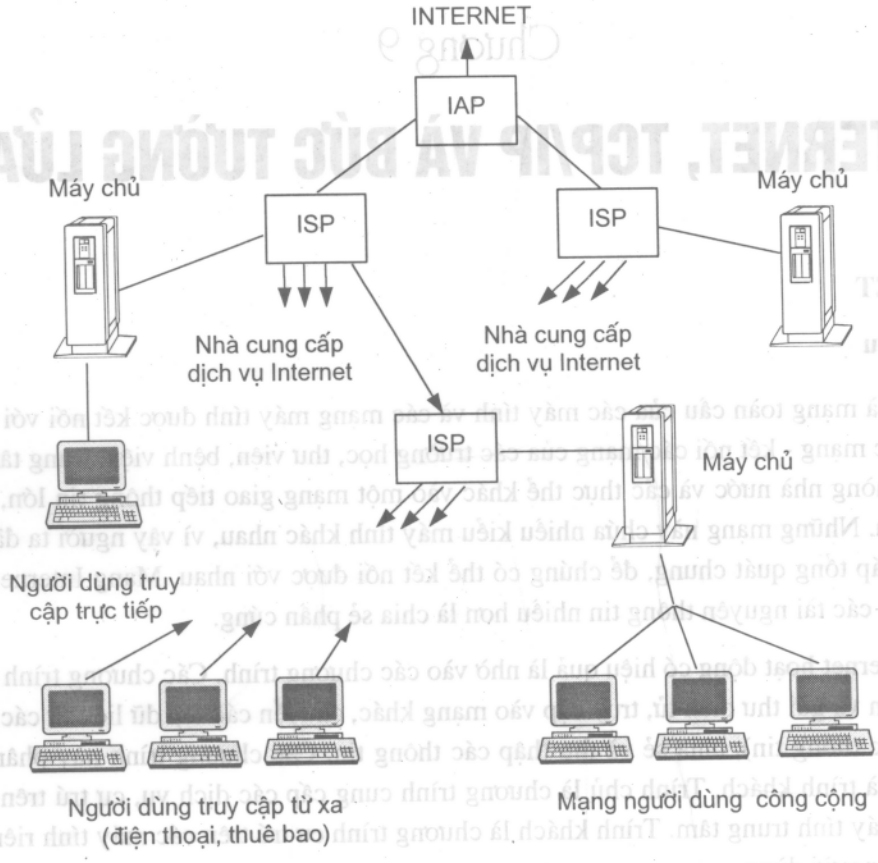
Các nguồn thông tin được đặt trong các máy tính trên mạng Internet. Như vậy, mạng Internet còn được hiểu là một bộ sưu tập các thông tin lưu trữ trong các máy tính nằm rải rác trên toàn cầu. Các tư liệu thông tin được tổ chức theo trang. Người ta đưa một trang lên màn hình máy tính, xem nội dung và nhờ các tuỳ chọn của chương trình để mang về và hiển thị nhiều trang thông tin khác. Các trang trên Internet mang lại cho người dùng các thông tin dưới dạng văn bản được định dạng hoặc có màu sắc, dưới dạng hình ảnh và âm thanh. Các trang này còn có khả năng tương tác, tạo ra một giao diện thuận lợi cho người dùng.

Chính sự thuận lợi, đa dạng và phong phú phục vụ người dùng trên diện rộng toàn cầu đó lại là môi trường rất thuận lợi cho tất cả các dạng xâm nhập mà người sử dụng cần được bảo vệ nghiêm ngặt. Hình 9.1. mô tả một mô hình kết cấu dịch vụ Internet.

9.1.2. Các lớp trong kiến trúc Internet

Có bốn lớp trong kiến trúc Internet, đó là: lớp truy nhập mạng, lớp Internet, lớp chuyển tải và lớp ứng dụng. Hình 9.2 mô tả bốn lớp kiến trúc Internet so với kiến trúc mô hình mở OSI - 7 lớp.

- *Lớp 1* (lớp truy nhập mạng). Lớp này tương ứng với lớp vật lý và lớp liên kết dữ liệu trong mô hình OSI-7 lớp. Lớp truy nhập mạng chấp nhận các gói dữ liệu lớp cao và truyền chúng trên mạng đến những nơi hệ thống khác nhau. Đối với các mạng vùng nội hạt dựa trên Ethernet thì mỗi khung là giữa 64 và 1518 byte.



Hình 9.1. *Sơ đồ kết cấu dịch vụ Internet*

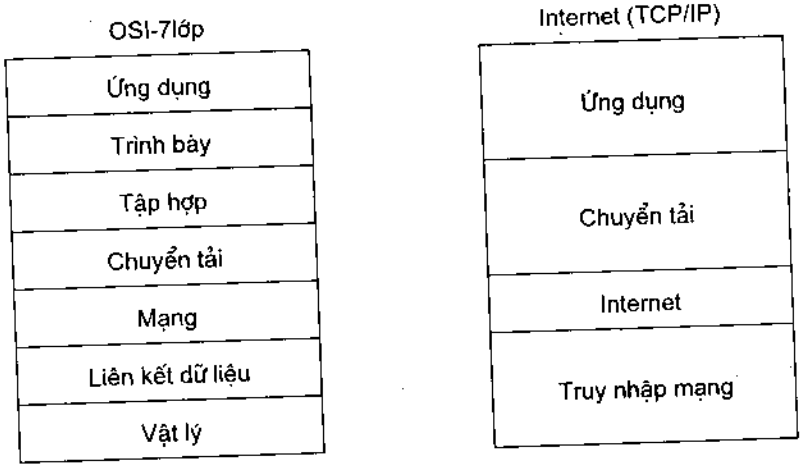
- *Lớp 2* (lớp Internet). Lớp Internet tương ứng với lớp mạng trong mô hình OSI-7 lớp. Lớp này xử lý giao tiếp thông tin giữa máy với máy, với một máy có thể là máy chủ hay bộ định tuyến Internet. Gói được nhận từ lớp chuyển tải được gói trong gói dữ liệu của giao thức Internet (IP). Dựa trên thông tin máy chủ đích, lớp Internet sử dụng thuật toán định tuyến để xác định hoặc là phân phát gói dữ liệu trực tiếp hoặc là gửi nó tới cổng chính (gateway) vào. Đối với các gói dữ liệu đến thì lớp Internet kiểm tra tính hợp lệ, xóa thông tin tiêu đề (header) và sử dụng thuật toán định tuyến để xác định có gói dữ liệu được xử lý trên hệ thống địa phương hay gửi chuyển tiếp. Nếu gói dữ liệu là cho hệ thống địa phương thì phần mềm của lớp Internet xác định giao thức của lớp chuyển tải nào sẽ kế tiếp xử lý gói. Các giao thức như giao thức phân tích địa chỉ (ARP) và giao thức phân tích địa chỉ đảo (RARP) hoạt động ở lớp Internet.

- *Lớp 3* (lớp chuyển tải). Lớp này trong Internet tương ứng với chức năng được cung cấp bởi lớp chuyển tải và lớp tập hợp trong mô hình OSI - 7 lớp. Phần mềm chuyển tải chia dữ liệu nhận được từ các lớp cao hơn thành các phần nhỏ hơn, gọi là các đoạn tin (message). Mỗi đoạn tin được chuyển tới lớp Internet. Các giao thức làm việc ở lớp chuyển tải khác nhau về chức năng. Ví dụ,

giao thức điều khiển truyền dẫn (TCP) điều chỉnh luồng thông tin và chịu trách nhiệm đảm bảo rằng, dữ liệu truyền không có lỗi, liên tục và không bị trùng lặp. Bởi vậy, lớp này cũng được xem như lớp máy chủ đến máy chủ (host-to-host), lớp đầu cuối đến đầu cuối (end-to-end) hay lớp nguồn tới đích (sourse-to-destination).

- *Lớp 4* (lớp ứng dụng). Lớp ứng dụng trong kiến trúc Internet tương ứng với các lớp trình bày và ứng dụng trong mô hình OSI-7 lớp. Các khách hàng có thể đưa ra các gói dữ liệu ứng dụng như telnet.ftp hay rlogin để truy nhập các nút trên mạng Internet. Lớp ứng dụng phối hợp hoạt động với giao thức mức chuyển tải để gửi hay nhận dữ liệu.

Dữ liệu mà có thể là trình tự của các bản tin hay chuỗi các byte được chuyển tới lớp chuyển tải trên hệ thống nguồn. Lớp chuyển tải sau đó có thể thiết lập với tập hợp và gửi số liệu đến hệ thống đích. Lớp ứng dụng cũng được đề cập tới như là lớp xử lý vì trên các hệ thống UNIX thì các giao thức lớp ứng dụng được thực thi theo dạng xử lý.



Hình 9.2. ***So sánh mô hình OSI-7 lớp và kiến trúc Internet***

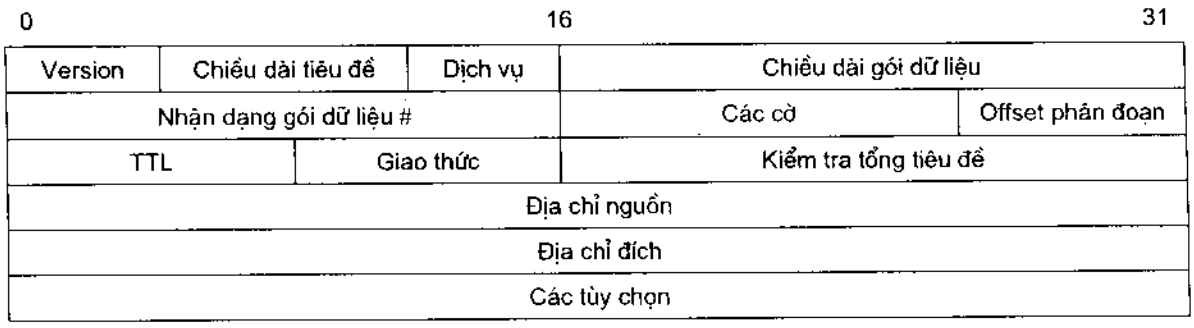
9.2. Giao thức Internet (IP)

Giao thức Internet là giao thức lớp mạng của OSI/RM. Các gói dữ liệu Internet có thể chuyển qua một vài mạng trước khi tới máy chủ đích của chúng. Gói dữ liệu là một gói tự chứa, độc lập với các gói khác, không yêu cầu sự xác nhận và nó mang thông tin đầy đủ cho việc định tuyến từ máy chủ gốc tới máy chủ đích. Bởi vì không có giai đoạn thiết lập khác nhau rõ ràng nên giao thức IP được gọi là *giao thức không kết nối*. Gói dữ liệu IP được định tuyến thông suốt đến máy chủ đích không nhất thiết là đáng tin cậy. Đảm bảo độ tin cậy là trách nhiệm của các lớp chuyển tải và ứng dụng. Hình 9.3 mô tả tiêu đề của giao thức IP.

Các trường trong tiêu đề IP được biểu thị như sau:

1. Version (4 bit): Xác định số phiên bản của giao thức IP. Số này hiện tại đặt là 4.

2. Chiều dài tiêu đề (4 bit): Chiều dài của tiêu đề IP, tiêu đề của IP có thể thay đổi về kích thước. Nhỏ nhất và điển hình, kích thước của IP là 20 byte. Trường này được trình bày theo các từ 32 bit; bởi vậy, nếu tiêu đề của IP là 20 byte thì số 5 được xác định trong trường hợp này.



Hình 9.3. ***Tiêu đề của giao thức Internet (IP)***

3. Dịch vụ (1 byte): Xác định các thông số độ tin cậy, trễ và lưu lượng truyền qua. Trường này còn được gọi là trường kiểu của dịch vụ.

4. Chiều dài gói dữ liệu (2 byte): Tổng chiều dài gói dữ liệu, bao gồm tiêu đề được sử dụng ở lớp truy cập mạng (ví dụ tiêu đề Ethernet)

5. Số nhận dạng (ID) gói dữ liệu (2 byte): Trường này cùng với hai trường kế tiếp, các cờ và offset đoạn, được sử dụng bởi hệ thống đích để tập hợp lại các gói dữ liệu bị phân đoạn.

6. Các cờ (3 bit): được sử dụng cho việc phân đoạn và ghép lại.

7. Offset phân đoạn (13 bit): Chỉ ra nơi trong gói dữ liệu mà phân đoạn này thuộc về; đơn vị đo là 64bit.

8. Thời gian sống, TTL (1 byte): Trường này là cho cả bộ đến và bộ định thời. Giá trị cực đại có thể xác định được trong trường này là 255. Mỗi khi một gói dữ liệu được nhận bởi bộ định tuyến thì trường này giảm đi 1. Bộ định tuyến cũng theo dõi thời gian mà nó xử lý gói dữ liệu (để xác định địa chỉ của bước nhảy kế tiếp). Trường TTL cũng trừ đi thời gian xử lý này trước khi bộ định tuyến giải phóng gói dữ liệu.

9. Giao thức (1 byte): chỉ ra giao thức lớp cao hơn nằm trong gói dữ liệu. Nếu trường giao thức trong gói IP là 1 thì nó là gói ICMP; nếu là 6 thì nó là gói TCP; nếu là 17 thì nó là gói UDP.

10. Kiểm tra tổng (checsum) tiêu đề (2 byte): được sử dụng cho phát hiện lỗi.

11. Địa chỉ nguồn (4 byte). Địa chỉ Internet của hệ thống nguồn (bao gồm phần mạng và máy chủ của Internet).

12. Địa chỉ đích (4 byte). Địa chỉ Internet của hệ thống cuối cùng hay đích (bao gồm phần mạng và máy chủ của Internet)

13. Các tùy chọn (biến đổi). Bao gồm các tùy chọn như định tuyến nguồn an toàn, nới lỏng hay nghiêm ngặt, thông báo lỗi, nhãn thời gian hay gỡ rối trong những loại khác.

14. Đệm (biến đổi). Được sử dụng để đảm bảo rằng, tiêu đề IP kết thúc ở ranh giới 32 bit.

15. Dữ liệu (biến đổi). Trường phải là bội số của 8 bit theo chiều dài; tổng chiều dài của trường dữ liệu cộng tiêu đề có cực đại là 65.535 byte.

Phân đoạn gói dữ liệu IP: Giao thức IP cố gắng tối đa để đưa gói dữ liệu tới hệ thống đích càng nhanh càng tốt. Các thiết bị giao tiếp như các bộ định tuyến xử lý các gói dữ liệu IP và xác định tuyến đường mà gói dữ liệu đi qua để tới hệ thống đích.

Có khả năng là bộ định tuyến có thể nhận một gói dữ liệu quá lớn đối với nó để xử lý. Trong trường hợp này, nó có thể ngắt gói dữ liệu thành các phần nhỏ hơn, được gọi là các *phân đoạn*. Việc ghép các phân đoạn này lại không phải là trách nhiệm của bộ định tuyến mà là của hệ thống đích cuối cùng. Các cờ và các trường offset phân đoạn trong tiêu đề IP được sử dụng cho các chức năng phân đoạn và ghép lại.

Ngay khi hệ thống đích nhận được phân đoạn đầu tiên, bộ đếm (bộ đếm thời gian ghép đoạn) sẽ được khởi động. Hệ thống đích phải nhận tất cả các đoạn từ gói dữ liệu đầu tiên trước khi bộ đếm thời gian ngừng hoạt động và hệ thống đích nếu không nhận được tất cả các đoạn thì bản tin ICMP được khởi phát và thông báo cho hệ thống nguồn.

9.3. GIAO THỨC ĐIỀU KHIỂN TRUYỀN DẪN (TCP)

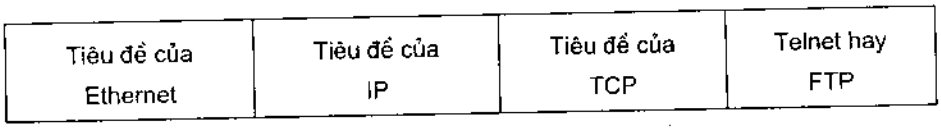
Giao thức điều khiển truyền dẫn (TCP) là giao thức của lớp chuyển tải có chức năng cung cấp cơ chế tin cậy cho việc trao đổi dữ liệu giữa các quá trình trong các hệ thống khác nhau. Lớp chuyển tải là lớp *end-to-end* đầu tiên, nói cách khác đó là lớp *nguồn - đích* (source - to - destination). Do vậy, chương trình trên hệ thống nguồn giao tiếp với chương trình khác trên hệ thống đích. Các lớp thấp hơn, như lớp mạng và bên dưới, giao tiếp giữa các hệ thống và các láng giềng trung gian của nó (như các bộ định tuyến) và không giao tiếp trực tiếp giữa các hệ thống nguồn và đích.

Khi lớp mạng không bảo đảm việc phân phát gói thì TCP chịu trách nhiệm về độ tin cậy. Các gói cũng có thể được phân phát nhờ lớp mạng không theo trình tự. TCP chịu trách nhiệm về trình tự gói. Modul IP gói các đoạn TCP bên trong các gói IP và định tuyến các gói này sau cùng đối với máy chủ đích.

TCP gọi modul IP đến lượt sau đó lại gọi driver của thiết bị mạng. Nói chung các dịch vụ được cung cấp bởi TCP bao gồm như sau:

- Dữ liệu phân phát được bảo đảm;
- Dữ liệu phân phát theo trình tự;
- Không trùng lặp số liệu;
- Quản lý lớp tập hợp được thiết lập giữa hệ thống nguồn và đích.

Hình 9.4. mô tả TCP sử dụng các dịch vụ của IP và Ehternet như thế nào.



Hình 9.4. ***Mô tả Telnet hay FTP sử dụng các dịch vụ của TCP, IP và Ethernet***

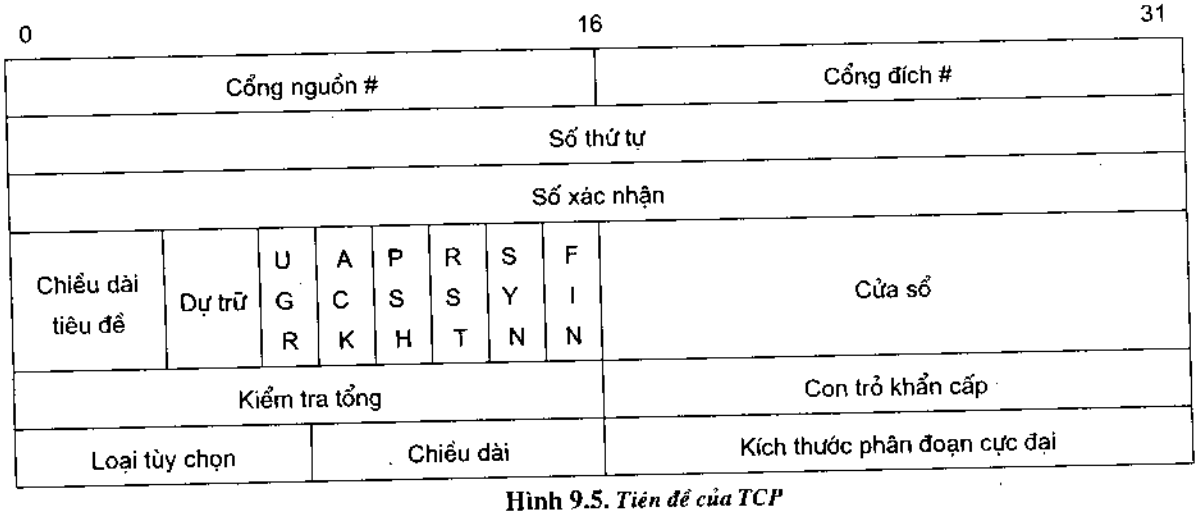
Hình 9.5 mô tả chi tiết các trường của tiêu đề TCP. Số các cổng là các điểm vào trong hệ thống, nơi hệ thống khác có thể truy nhập một ứng dụng. Ví dụ, số cổng 23 được kết hợp với ứng dụng telnet.

Sau đây là mô tả sơ lược mỗi trường trong tiêu đề TCP:

1. *Cổng nguồn* (2 byte): Nhận dạng cổng nguồn, cũng có thể được xem như số cổng của ứng dụng khách hàng trong kết nối ban đầu.

2. *Cổng đích* (2 byte): Nhận dạng cổng đích; cũng có thể được xem như số cổng của ứng dụng server cho kết nối ban đầu.

3. *Số trình tự* (4 byte): Số trình tự của byte dữ liệu đầu tiên trong đoạn này, ngoại trừ khi SYN (đề cập tới đồng bộ số các trình tự) được đưa ra; nếu SYN được đưa ra thì số trình tự là số được lựa chọn ngẫu nhiên.



Hình 9.5. ***Tiên đề của TCP***

4. *Số xác nhận* (4 byte): Sự xác nhận truy nhập ngược (piggy-backed acknowlegment); chứa số trình tự là số được lựa chọn ngẫu nhiên.

5. *Chiều dài tiêu đề* (4 bit): Xác định số lượng các từ 32- bit trong tiêu đề.

6. *Dự trữ* (6 bit): Phục vụ cho mục đích sử dụng trong tương lai.

7. *Các cờ* (6 bit): Có 6 trường cờ, mỗi trường có chiều dài 1 bit.

URG là con trỏ khẩn cấp (nếu như có bất kỳ một dữ liệu khẩn nào trong gói này).

ACK là xác nhận (nếu như có thông tin xác nhận trong gói này).

PHS là chức năng thúc đẩy (để buộc TCP đọc và giải phóng dữ liệu ngay lập tức).

RST là thiết lập kết nối (để nhanh chóng đưa vào kết nối TCP)

TCP ở lớp chuyển tải; bit này được thiết lập là 1.

FIN có nghĩa là không có thêm dữ liệu từ người gửi nữa (cách thông thường để đưa xuống kết nối TCP).

8. *Cửa sổ* (2 byte): Sự sắp xếp thẻ (credit) điều khiển luồng theo byte: chứa số lượng các byte dữ liệu bắt đầu với 1 được chỉ ra trong trường xác nhận mà người gửi sẵn sàng chấp nhận. Nếu bit

này được thiết lập là O thì hệ thống đích sẽ yêu cầu hệ thống nguồn giảm tốc độ tại chỗ nào đang khởi phát dữ liệu. Giá trị của trường này có thể thay đổi trong kết nối TCP định trước. Trường này có thể được xem như một cách để điều khiển luồng dữ liệu giữa các hệ thống sử dụng các dịch vụ của TCP.

9. *Kiểm tra tổng* (2 byte): được sử dụng cho phát hiện lỗi.

10. *Con trỏ khẩn cấp* (2 byte): Trỏ tới byte tiếp sau dữ liệu khẩn cấp; do vậy người nhận có thể xác định có bao nhiêu dữ liệu khẩn cấp trong bản tin.

11. *Kiểu tùy chọn* (1 byte): hiện nay chỉ có một tùy chọn được định nghĩa, xác định kích thước đoạn cực đại sẽ được chấp nhận, số tùy chọn được định nghĩa là 2.

12. *Chiều dài tùy chọn* (1 byte): Đối với số tùy chọn là 2 thì chiều dài tùy chọn sẽ là 4 byte.

13. *Kích thước đoạn cực đại* (2 byte): Cũng được coi là trường MSS: trường này xác định kích thước cực đại mà người gửi có thể nhận được cho kết nối đã định trước (nếu nguồn và đích có cùng phần mạng của địa chỉ Internet thì số này điển hình được dựa trên kích thước khung lớn nhất ở lớp truy nhập mạng). Giá trị chứa trong trường này liên quan tới đơn vị truyền dẫn cực đại (MTU) của giao thức đang sử dụng tại lớp truy nhập mạng. Ví dụ trên mạng Ethernet, giá trị được xác định trong trường này là 1460. Đó là vì:

Kích thước khung Ethernet cực đại: 1518 byte;

Header + trailer của Ethernet: 18 byte;

Header IP: 20 byte;

Header TCP: 20 byte.

Nếu cộng 18 byte tiêu đề của Ethernet, 20 byte tiêu đề IP, và 20 byte tiêu đề TCP thì sẽ có 58 byte. Lấy 1518 byte là giá trị kích thước khung Ethernet cực đại trừ đi 58 byte sẽ còn lại 1460 byte được xác định trong trường MSS.

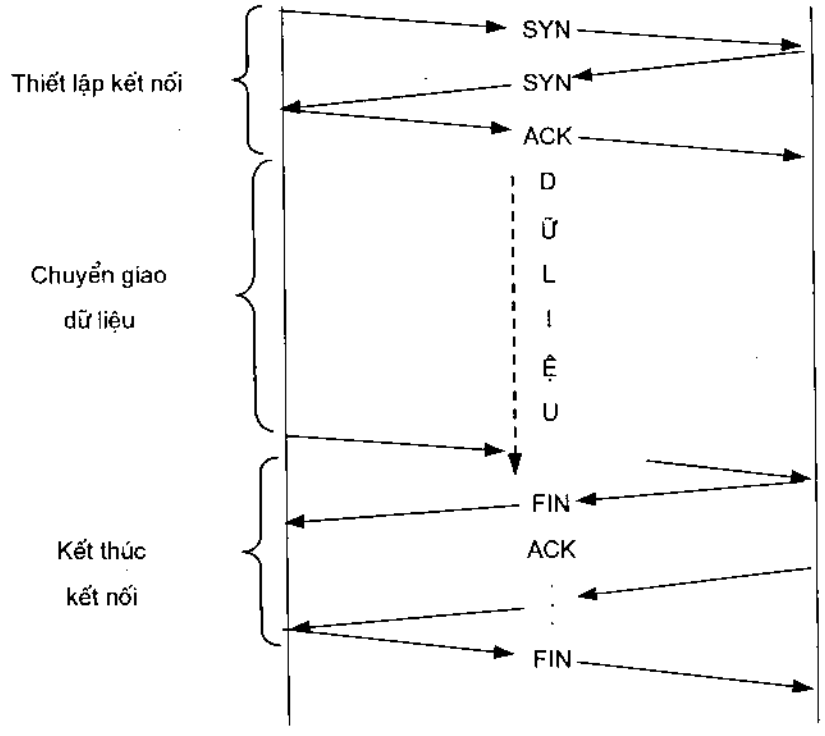
14. *Dữ liệu* (biến đổi): Trường dữ liệu có giá trị cực đại là 65.535 byte.

Hình 9.6 mô tả tóm tắt quá trình kết nối, chuyển giao dữ liệu và kết thúc kết nối kết hợp với bất kỳ một kết nối TCP nào. Quá trình thiết lập kết nối cũng được xem như sự *đồng bộ*, bao gồm 3 bước:

1. Hệ thống nguồn thiết lập bit SYN và gửi đi bản tin TCP bao gồm thông tin theo số trình tự mà hệ thống đích sẽ sử dụng để xác nhận các byte đã nhận; đây là bản tin đầu tiên.

2. Hệ thống đích đáp ứng nhờ việc thiết lập các bit SYN và ACK. Bit SYN được thiết lập để thông báo cho hệ thống nguồn về số trình tự của hệ thống đích; bit ACK được thiết lập để thông báo cho nguồn là hệ thống đích đã nhận bản tin xác định số trình tự của hệ thống nguồn.

3. Bản tin thứ 3 được gửi đi từ hệ thống nguồn tới hệ thống đích. Ở đây, bit ACK được thiết lập, nguồn xác định số trình tự của hệ thống đích. Từ khi 3 bản tin được trao đổi giữa các hệ thống nguồn và đích để thiết lập kết nối TCP, điều này được xem như quá trình bắt tay 3 tuyến đường.



Hình 9.6. ***Mô tả quá trình kết nối, chuyển giao dữ liệu và kết thúc kết nối TCP***

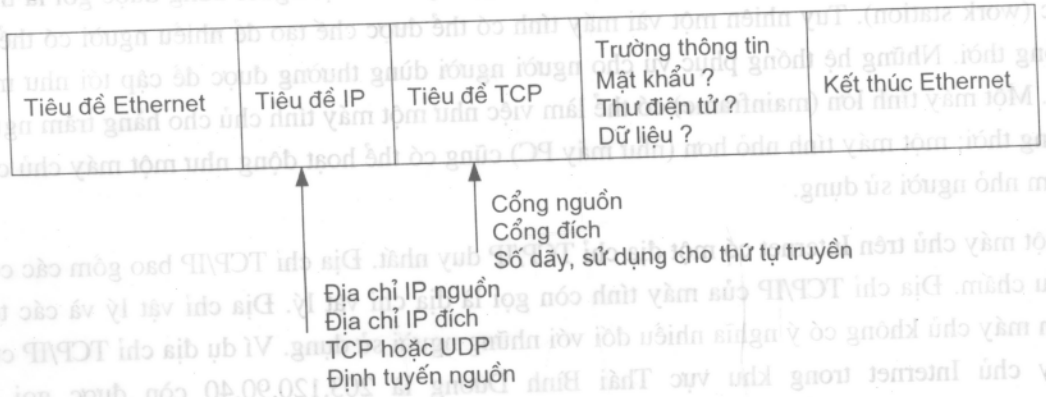
9.4. TCP/IP VÀ CUNG CẤP DỊCH VỤ INTERNET

Giao thức điều khiển truyền dẫn và giao thức Internet, viết tắt là TCP/IP (Transmission Control Protocol/Internet Protocol) là tên chung cho một tập hợp trên 100 giao thức được sử dụng để kết nối các máy tính vào mạng.

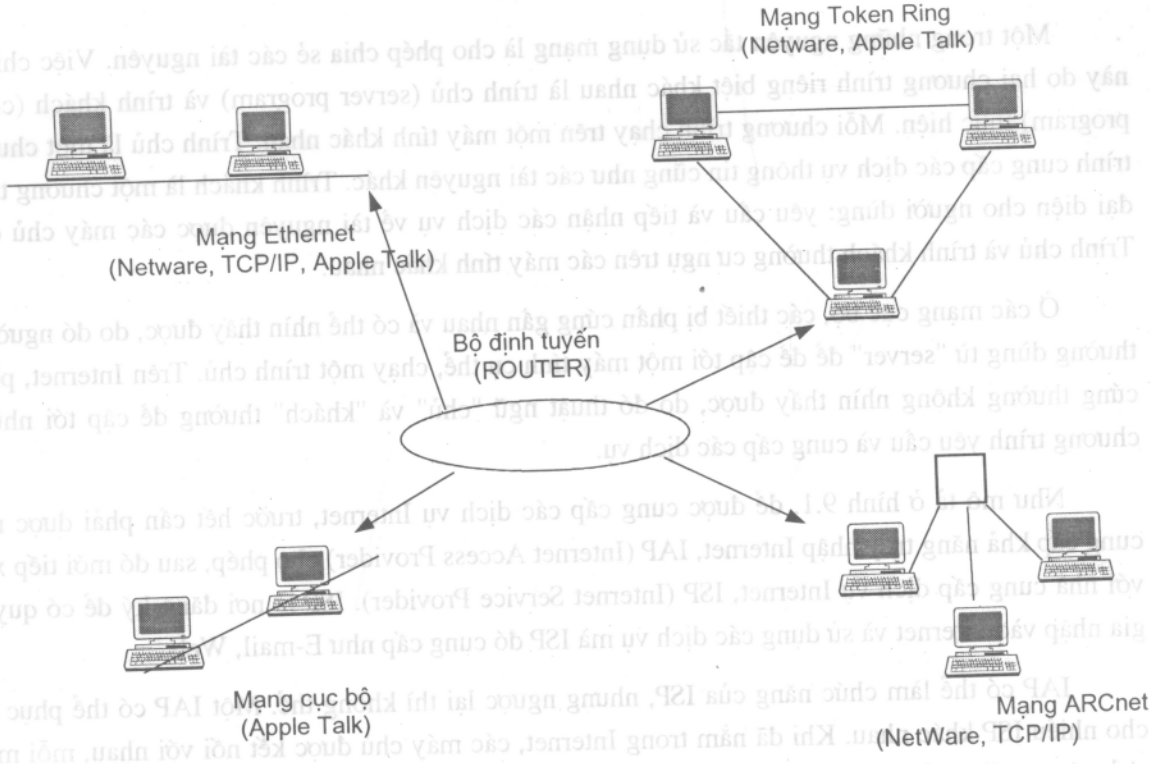
Trong phạm vi mạng Internet cũng như trong các mạng số hiện đại của các quốc gia kết nối, thông tin không được truyền tải như một dòng liên tục từ máy chủ này đến máy chủ khác. Thay vào đó, dữ liệu được chia thành gói nhỏ để truyền, trong kỹ thuật viễn thông và truyền tin gọi là "*chuyển mạch gói*". Mỗi gói thông tin gồm có hai phần: phần tiêu đề và phần thông tin. Phần tiêu đề chứa địa chỉ nguồn, địa chỉ đích và các thông tin điều khiển khác. Phần thông tin chứa các nội dung thông tin trao đổi như mật khẩu, thư điện tử và dữ liệu cần trao đổi. Tại phía nguồn TCP chia đoạn tin thành các gói và sắp xếp theo khuôn mẫu qui định thống nhất và tại phía nhận TCP nhận, sắp xếp các gói và kiểm tra lỗi. Nếu xuất hiện lỗi trong một gói, TCP yêu cầu gói riêng biệt đó phải được gửi lại. Chỉ khi tất cả các gói đã được nhận đúng theo quy cách, TCP sẽ sử dụng thứ tự để tạo lại bản tin ban đầu. IP làm nhiệm vụ truyền tải các gói. Nói cách khác, công việc của IP là chuyển dữ liệu thô (các gói) từ nơi này đến nơi khác còn công việc của TCP là quản lý dòng lưu lượng và bảo đảm sự chính xác cho dữ liệu truyền.

Khi các gói đi chuyển, chúng hành trình từ máy chủ này sang máy chủ khác cho đến khi chúng tới được đích cuối cùng. Đây là sự linh hoạt của Internet. Nếu một lộ trình nào đó bị ngắt, những gói có thể đi theo lộ trình khác. Như vậy, khi những điều kiện thay đổi, mạng có thể sử dụng mối liên kết tốt nhất có thể tại mỗi thời điểm. Ví dụ khi một phần riêng biệt của mạng bị quá tải,

các gói có thể hành trình theo các tuyến đường dây khác, ít bận rộn hơn. Một lợi ích khác của việc chuyển mạch gói là khi có một sai lỗi nào đó, chỉ có một gói đơn lẽ cần truyền lại, thay vì toàn bộ bản tin. Điều này làm tăng rất nhiều tốc độ chung của Internet. Tất cả sự linh hoạt này làm cho việc truyền dữ liệu có độ tin cậy cao. Bằng cách này hay cách khác, TCP/IP bảo đảm rằng dữ liệu truyền tải được thông suốt. Trên thực tế, Internet chạy tốt đến mức chỉ cần vài giây để gửi một tệp tin từ máy chủ này đến máy chủ khác, thậm chí chúng cách xa hàng ngàn dặm và tất cả các gói phải truyền qua nhiều máy tính. Có thể nói TCP/IP là một thứ keo dán để duy trì sự liên kết hàng ngàn mạng máy tính và hàng triệu máy tính trên Internet. Hình 9.7 mô tả ví dụ cấu trúc một gói IP trong khung Ethernet và hình 9.8 mô tả giao thức truyền dữ liệu trong Internet.



Hình 9.7. *Gói IP trong khung Ethernet.*



Hình 9.8. *Giao thức truyền dữ liệu trong Internet*

Ở đây cần lưu ý một số thuật ngữ sử dụng khi nghiên cứu về Internet. Máy chủ (host) là một máy tính được nối vào mạng và có khả năng cung cấp thông tin cho máy tính khác trong mạng. Nói cách khác máy chủ là máy tính trong mạng mà các máy tính khác trong mạng đó có thể truy cập được.

Trạm đầu cuối (terminal) là một máy tính được ghép nối vào một mạng và hoạt động do nguồn lực máy tính chủ hỗ trợ.

Như vậy, một máy tính có thể được sử dụng theo hai cách: một là sử dụng toàn bộ nó và hai là sử dụng với hình thức chia sẻ các tài nguyên. Những máy tính một người dùng được gọi là trạm làm việc (work station). Tuy nhiên một vài máy tính có thể được chế tạo để nhiều người có thể sử dụng đồng thời. Những hệ thống phục vụ cho người người dùng thường được đề cập tới như máy tính chủ. Một máy tính lớn (mainframe) có thể làm việc như một máy tính chủ cho hàng trăm người dùng đồng thời; một máy tính nhỏ hơn (như máy PC) cũng có thể hoạt động như một máy chủ cho một nhóm nhỏ người sử dụng.

Một máy chủ trên Internet có một địa chỉ TCP/IP duy nhất. Địa chỉ TCP/IP bao gồm các con số và dấu chấm. Địa chỉ TCP/IP của máy tính còn gọi là địa chỉ vật lý. Địa chỉ vật lý và các tên logic trên máy chủ không có ý nghĩa nhiều đối với những người sử dụng. Ví dụ địa chỉ TCP/IP của một máy chủ Internet trong khu vực Thái Bình Dương là 203.120.90.40 còn được gọi là www.pacific.net.sg. Giao tiếp được thiết lập giữa người dùng và máy chủ thông qua các địa chỉ TCP/IP.

Một trong những nguyên tắc sử dụng mạng là cho phép chia sẻ các tài nguyên. Việc chia sẻ này do hai chương trình riêng biệt khác nhau là trình chủ (server program) và trình khách (client program) thực hiện. Mỗi chương trình chạy trên một máy tính khác nhau. Trình chủ là một chương trình cung cấp các dịch vụ thông tin cũng như các tài nguyên khác. Trình khách là một chương trình đại diện cho người dùng: yêu cầu và tiếp nhận các dịch vụ về tài nguyên được các máy chủ cấp. Trình chủ và trình khách thường cư ngụ trên các máy tính khác nhau.

Ở các mạng cục bộ, các thiết bị phần cứng gần nhau và có thể nhìn thấy được, do đó người ta thường dùng từ "server" để đề cập tới một máy tính cụ thể, chạy một trình chủ. Trên Internet, phần cứng thường không nhìn thấy được, do đó thuật ngữ "chủ" và "khách" thường đề cập tới những chương trình yêu cầu và cung cấp các dịch vụ.

Như mô tả ở hình 9.1, để được cung cấp các dịch vụ Internet, trước hết cần phải được nhà cung cấp khả năng truy nhập Internet, IAP (Internet Access Provider) cho phép, sau đó mới tiếp xúc với nhà cung cấp dịch vụ Internet, ISP (Internet Service Provider). ISP là nơi đăng ký để có quyền gia nhập vào Internet và sử dụng các dịch vụ mà ISP đó cung cấp như E-mail, Web...

IAP có thể làm chức năng của ISP, nhưng ngược lại thì không thể. Một IAP có thể phục vụ cho nhiều ISP khác nhau. Khi đã nằm trong Internet, các máy chủ được kết nối với nhau, mỗi máy chủ có một địa chỉ riêng gọi là URL (Uniform Resource Locator), người dùng có thể truy cập trực tiếp thông qua cấp mạng, nhưng phần lớn là truy nhập từ xa thông qua mạng điện thoại (hoặc vệ

tinh viễn thông). Các ISP được kết nối với IAP để giao tiếp với Internet. Mỗi người dùng Internet thường có yêu cầu riêng của mình. Trên cơ sở đó, nhiều ISP đã phân chia thị trường thành các lĩnh vực khác nhau, cũng như giá cả và tính cước.

9.5. INTERNET VÀ AN TOÀN THÔNG TIN DỮ LIỆU

ISO định nghĩa tính an toàn thông tin dữ liệu có nghĩa là giảm tối thiểu việc tấn công các tài nguyên và dữ liệu quý giá. Một dữ liệu quý giá được định nghĩa là bất cứ dữ liệu gì có giá trị. Tính chất bị tấn công là bất kỳ điểm yếu nào có thể bị lợi dụng để phá rối hệ thống hoặc các thông tin trong nó. Mối đe doạ là một sự vi phạm có tính tiềm năng đến tính an toàn.

Như đã phân tích ở chương một, các đe dọa an toàn tới một hệ thống có thể được phân loại thành *đe dọa thụ động* và *đe doạ tích cực*. Đe dọa thụ động là các đe dọa mà nếu như được thực hiện thì không dẫn đến bất kỳ thay đổi nào đối với bất kỳ thông tin nào trong hệ thống và những nơi, hoặc sự vận hành hoặc trạng thái của hệ thống không thay đổi. Ví dụ, một khách hàng chạy ứng dụng *Snoop* từ hệ thống của họ và nghe xen các gói thông tin trao đổi giữa hai nút trên mạng. Trong ví dụ này, cho dù khách hàng có được một bản sao của các gói mạng đó để phân tích số liệu trao đổi thì vẫn không thay đổi được gói gốc được gửi từ hệ thống nguồn tới hệ thống đích.

Sự lựa chọn thông tin, hoặc các thay đổi trạng thái hoặc khai thác hệ thống được định nghĩa như một đe dọa tích cực tới hệ thống.

Các hệ thống tồn tại trên mạng có thể là mục tiêu của các kiểu tấn công nhất định. Có thể là một thực thể giả danh một thực thể khác để đánh lừa. Một thực thể được hiểu là một khách hàng, một quá trình hoặc một nút trên mạng. Sự giả dạng thường sử dụng cùng với các dạng tấn công tích cực khác.

Sự tấn công cũng có thể ngăn chặn hoạt động của mạng, đặc biệt nếu mạng có các thực thể chuyển tiếp, mà các thực thể này thực hiện các quyết định định tuyến dựa trên các thông báo trạng thái nhận được từ các thực thể chuyển tiếp khác.

Các tấn công từ nội bộ (từ bên trong hệ thống) xảy ra khi các đối tượng sử dụng hợp pháp của hệ thống thực hiện các thao tác không được phép và ngoài dự kiến. Hầu hết các vi phạm máy tính như đã biết đều liên quan đến các tấn công bên trong làm tổn hại đến tính an toàn của hệ thống.

Các dịch vụ an toàn thông tin dữ liệu có thể thực hiện trong các lớp của mô hình OSI và các dịch vụ an toàn thường sử dụng được định nghĩa như sau (theo ISO - 7498/2):

Dịch vụ an toàn	*Mô tả*
- Sự cho phép	- Sự cho phép thường là bước đầu tiên đạt được sự truy nhập tới hệ thống. Đánh vào tên đối tượng sử dụng và một từ khóa là một ví dụ sự cho phép sử dụng hệ thống. Sự cho phép được xem là một quá trình chứng minh sự nhận dạng.
- Điều khiển truy nhập	- Dịch vụ này bảo vệ chống lại truy nhập các tài nguyên không được phép qua việc sử dụng các giao thức mạng. Sự cho phép các file, các thư mục và các quá trình xử lý liên quan đến các vùng điều khiển

truy nhập - là những người truy nhập đến các tài nguyên này (đối tượng) trên hệ thống. Điều khiển truy nhập liên quan tới một đối tượng sử dụng có thể truy nhập tới các tài nguyên nào trên hệ thống hoặc trên mạng.

- Sự bí mật dữ liệu

- Dịch vụ này để bảo vệ dữ liệu. Các dịch vụ bí mật dữ liệu bao gồm sự bí mật kết nối, bí mật phi kết nối, bí mật có chọn lọc và bí mật chuỗi lưu lượng. Bí mật dữ liệu bao gồm bí mật dữ liệu trên hệ thống và trên mạng, bảo vệ chống các xâm nhập thụ động và xâm nhập tích cực.

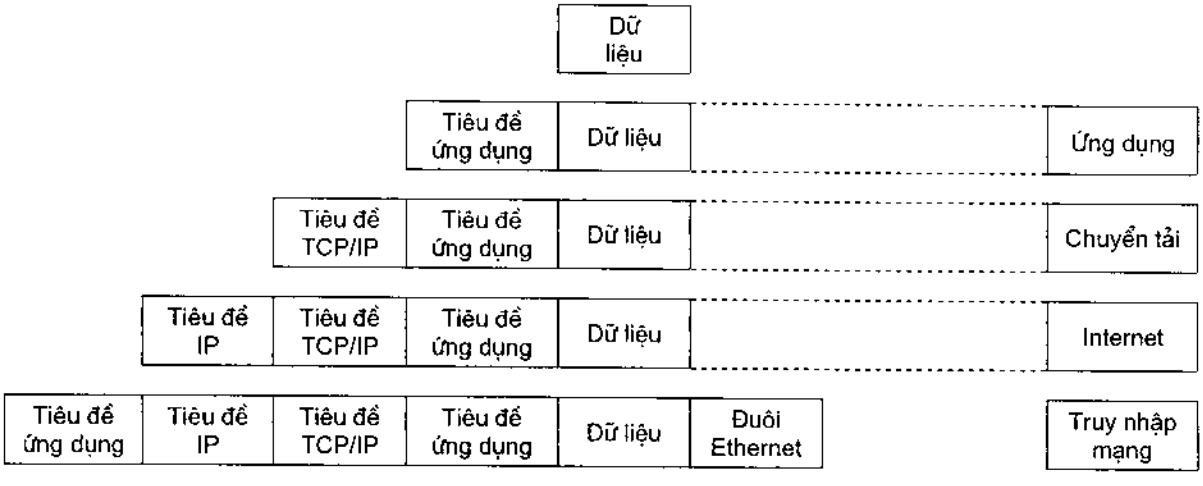
- Sự toàn vẹn dữ liệu

- Các dịch vụ toàn vẹn dữ liệu bao gồm tính toàn vẹn kết nối với khả năng khôi phục, tính toàn vẹn kết nối mà không có khả năng khôi phục, toàn vẹn kết nối thường có chọn lọc và toàn vẹn không kết nối có chọn lọc.

- Sự không thừa nhận

- Sự không thừa nhận được xác định như một sự phủ nhận bởi một trong các thực thể tham gia vào toàn bộ hay một phần quá trình trao đổi thông tin.

Có thể có nhiều phương pháp lựa chọn để bảo vệ an toàn thông tin dữ liệu trên hệ thống và trên mạng. Để có thể sử dụng tốt phương pháp lựa chọn bao gồm cả việc sử dụng bức tường lửa cần dựa vào cấu trúc dữ liệu ở trong Internet sử dụng giao thức TCP/IP như mô tả ở hình 9.9.



Hình 9.9. ***Kiến trúc TCP/IP và gói dữ liệu***

Mỗi một hệ thống tùy thuộc vào tình hình cụ thể cần xác định một chính sách an toàn cho toàn bộ hệ thống, mạng và các ứng dụng. Một chính sách an toàn trước hết thể hiện cái gì được phép và cái gì không được phép trong hoạt động tổng thể của hệ thống hoặc các phần tử mạng trong đó bao gồm phần an toàn dữ liệu, an toàn máy tính và an toàn mạng.

9.6. BỨC TƯỜNG LỬA

9.6.1. Giới thiệu

Thuật ngữ bức tường lửa (firewall) được các tác giả ban đầu đặt tên cho một phương thức ngăn chặn hữu hiệu các truy cập và thông tin bất hợp pháp giữa một mạng này và mạng khác, đặc biệt khi Internet xuất hiện. Rất tiếc là việc lựa chọn sử dụng thuật ngữ "bức tường lửa" đó đã dẫn đến nhiều người sử dụng hiểu nhầm, hiểu một cách đơn giản rằng, đó là một hàng rào chắn bất khả xâm phạm, như một bức tường lửa thật sự, không gì phá vỡ và lọt qua được. Một quan niệm đơn giản có thể nói là nguy hiểm nếu sử dụng nó nhưng không hiểu rõ về nó.

Các bức tường lửa thực chất là cung cấp một cấp độ cách ly giữa hai mạng. Nếu như bức tường lửa đó được lựa chọn, cấu trúc và bảo trì một cách đúng đắn thì nó sẽ cung cấp một cấp độ bảo mật.

Cũng đã có một số ý kiến cho rằng, không những thuật ngữ bức tường lửa mà một số thuật ngữ liên quan khác cũng được sử dụng không chặt chẽ. Các ý kiến đó là: các cấu hình vật lý có thể nhầm lẫn do nhìn giống nhau nhưng từ ngữ dùng cần chính xác tránh hiểu nhầm. Cũng vì vậy, một số thuật ngữ sử dụng trong chương được hiểu như sau: Bức tường lửa dựa vào máy chủ (host-based firewall) là bức tường lửa ở máy tính dùng cho mục đích chung trong đó có cài phần mềm bảo mật (thuật ngữ máy chủ có thể hiểu khác nhau). Các máy tính đó kiểm tra hoạt động ở lớp ứng dụng (lớp 7) đúng hơn là ở lớp mạng (lớp 3) trong mô hình OSI - 7 lớp. Phần mềm dựa vào một uỷ quyền ứng dụng và thường bao gồm một khách và chủ cho dịch vụ cả hai phía của cổng chính (gateway).

Bởi vì chúng được cấu hình để cách ly xâm nhập, các bức tường lửa dựa vào máy chủ ở một số tài liệu gọi là "pháo đài" (bastion) hoặc các máy chủ cổng chính (gateway hosts). Các máy chủ uỷ quyền cũng được gọi là cổng chính ứng dụng. Trong chương sẽ dùng thuật ngữ cổng chính ứng dụng (application gateway).

Bức tường lửa dựa vào thiết bị định tuyến (router-based firewall) sử dụng các thiết bị định tuyến để sàng lọc các tiêu đề TCP/IP như là một cơ sở để chặn hoặc cho thông lưu lượng ở lớp mạng. Các thiết bị định tuyến như vậy được gọi là các định tuyến sàng lọc (screening routers) hoặc các định tuyến lọc gói (packet-filtering roaters). Trong chương thuật ngữ định tuyến sàng lọc, được hiểu là định tuyến lọc gói.

"Dual homing" là một thuật ngữ TCP/IP qui về cho một thiết bị có nhiều hơn một địa chỉ IP. Vì vậy một cổng chính có hai lối vào bao gồm một cổng chính ứng dụng với hai giao diện mạng, một cho mạng được bảo vệ và một giao diện khác cho Internet hoặc cho mạng không được bảo mật.

Mạng không bảo mật là một khái niệm rộng hơn mạng Internet. Trong khi nhiều bức tường lửa là các hàng rào chắn đối với mạng Internet thì chúng cũng có thể sử dụng để làm hàng rào chắn các mạng khác.

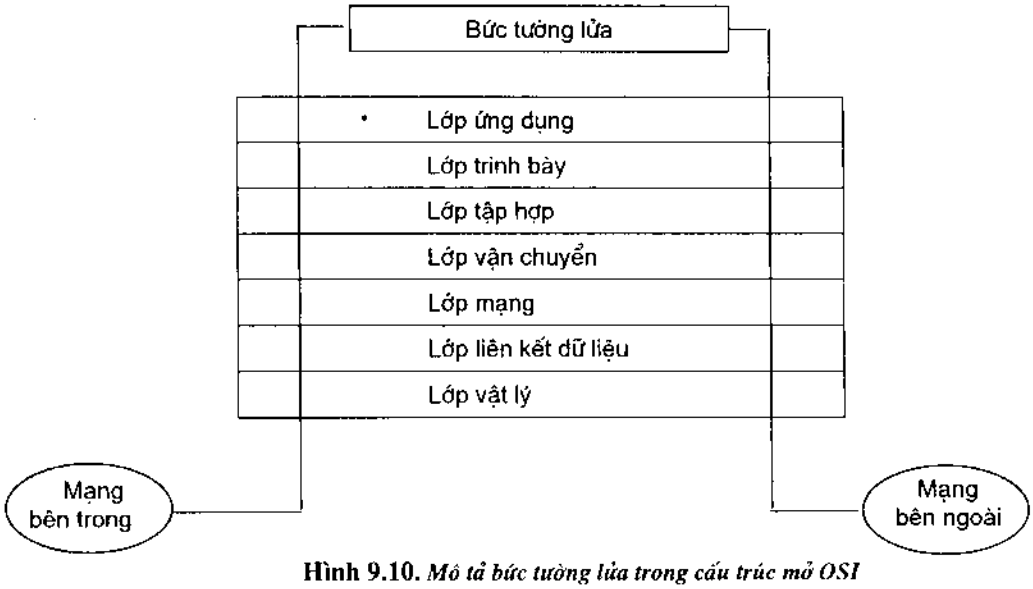
Một số cấu hình bức tường lửa có một điểm yếu cố hữu, đó là nếu như một kẻ xâm nhập bằng cách nào đó chọc thủng được bức tường lửa đơn, cụ thể là bức tường lửa lọc gói thì chúng hoàn toàn truy cập vào hệ thống. Sự xâm nhập đó chỉ là vòng ngoài nếu như hệ thống có những vòng trong bảo vệ vững chắc hơn. Có thể sử dụng phương pháp lọc dần từ mức thấp đến mức cao.

9.6.2. Bức tường lửa trong mô hình cấu trúc OSI

Cũng có thể hiểu rằng, bức tường lửa là một cơ cấu để bảo vệ mạng tin cậy (trusted network) ngăn ngừa sự xâm nhập của các mạng không tin cậy (untrusted network) mà thông thường là khi kết nối với Internet.

Thuật ngữ bức tường lửa cũng được nhiều người sử dụng như một thuật ngữ chung để mô tả một phạm vi rộng các chức năng và cấu trúc của các thiết bị bảo vệ mạng. Trên thực tế, có một số người sử dụng thuật ngữ để ám chỉ hầu như bất kỳ thiết bị an toàn nào, chẳng hạn như một thiết bị mật mã phần cứng, một bộ định tuyến sàng lọc hoặc một cổng ở lớp ứng dụng.

Nói chung, một bức tường lửa được bố trí giữa một mạng bên trong tin cậy và một mạng bên ngoài không tin cậy. Bức tường lửa hoạt động như một điểm chặn để giám sát và loại bỏ lưu lượng cần loại bỏ tại lớp ứng dụng như mô tả ở hình 9.10. Các bức tường lửa cũng có thể hoạt động tại lớp mạng và lớp vận chuyển, trong trường hợp đó chúng kiểm tra các tiêu đề của giao thức điều khiển truyền dẫn (TCP) và giao thức bức tường lửa (IP) của các gói gọi đến, gọi đi và loại bỏ hoặc cho đi qua các gói dựa vào các quy tắc lọc gói đã được lập trình sẵn.



Hình 9.10. *Mô tả bức tường lửa trong cấu trúc mở OSI*

9.6.3. Chức năng bức tường lửa

Về mặt chức năng, bức tường lửa là thành phần đặt giữa hai mạng để kiểm soát tất cả các trao đổi và truy nhập giữa chúng với nhau bao gồm:

- Tất cả các trao đổi dữ liệu từ trong ra ngoài và ngược lại phải thực hiện thông qua bức tường lửa.

- Chỉ có những chế độ nào được phép của an ninh hệ thống mạng tin cậy mới được quyền lưu thông qua bức tường lửa.

Về mặt vật lý, bức tường lửa bao gồm:

- Một hay nhiều máy chủ kết nối với các bộ định tuyến (router) hoặc có chức năng như bộ định tuyến.

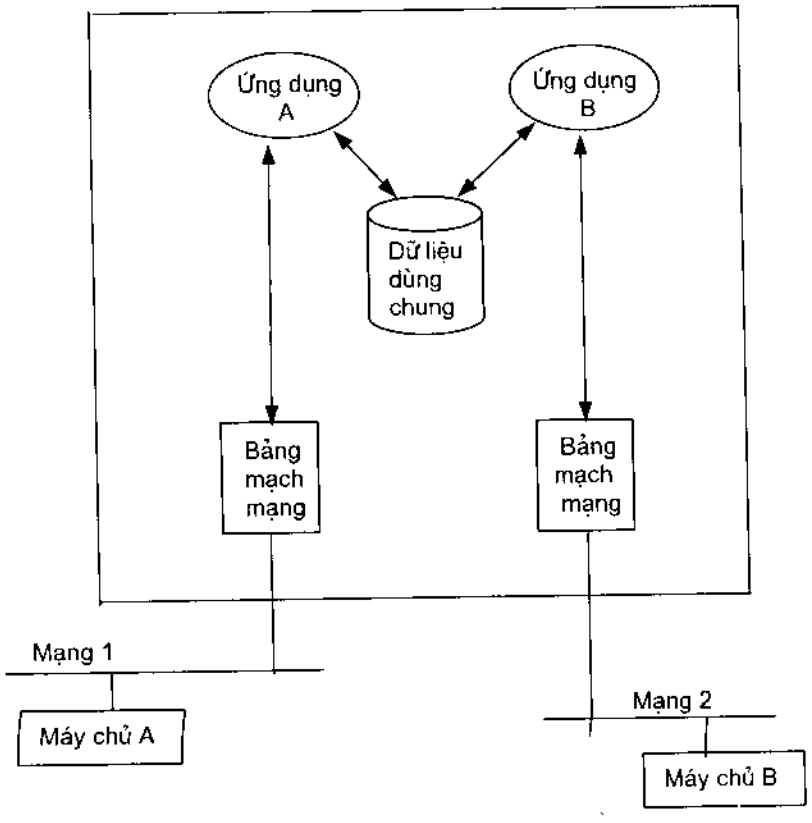
- Cấu trúc bảo vệ kiểm soát đảm bảo an ninh mạng. Phần này cũng do máy chủ phụ trách (có tài liệu còn gọi là máy chủ thành trì hoặc pháo đài bảo vệ).

- Các phần mềm quản lý bảo vệ an toàn chạy trên các hệ thống máy chủ, bao gồm các phần mềm bảo mật, các hệ quản trị xác thực, cấp quyền và các phần mềm khác.

Hình 9.11. Mô tả ví dụ một máy tính chủ hai cửa được sử dụng với các chức năng định tuyến bị vô hiệu hoá.

Máy tính chủ A trên mạng 1 có thể truy nhập ứng dụng A trên máy tính chủ hai cửa. Tương tự như vậy, máy tính B có thể truy nhập ứng dụng B trên máy tính chủ hai cửa.

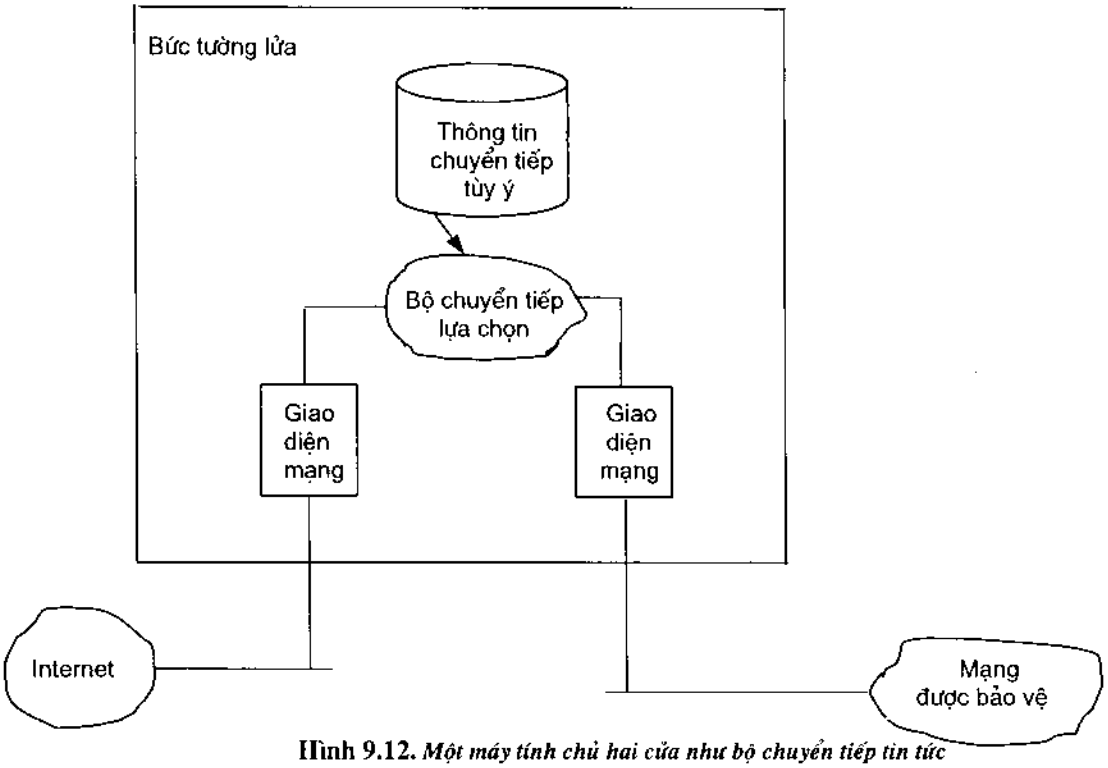
Thậm chí hai ứng dụng trên các máy tính chủ hai cửa có thể dùng chung số liệu. Các máy tính chủ A và B có khả năng trao đổi thông tin qua các số liệu dùng chung trên các máy tính chủ hai cửa nhưng không có sự trao đổi lưu lượng mạng giữa hai nhánh mạng được kết nối với máy chủ hai cửa này.



Hình 9.11. ***Máy tính chủ hai cửa***

Có thể sử dụng máy tính chủ hai cửa để ngăn cách một mạng bên trong với một mạng bên ngoài không tin cậy. Trong trường hợp này máy tính chủ hai cửa không chuyển tiếp bất kỳ lưu lượng TCP/IP nào, cho nên nó hoàn toàn ngăn chặn bất kỳ lưu lượng IP nào giữa mạng bên trong và mạng bên ngoài không tin cậy.

Nhiều dịch vụ Internet chủ yếu là các dịch vụ lưu trữ và chuyển tiếp. Nếu các dịch vụ này chạy trên máy tính chủ hai cửa thì chúng có thể được cấu hình để phát các dịch vụ ứng dụng từ máy này đến mạng kia. Nếu các số liệu ứng dụng phải đi qua bức tường lửa thì các tác nhân chuyển tiếp ứng dụng (application forwarder agents) có thể được thiết lập để chạy trên máy tính chủ hai cửa này. Các tác nhân chuyển tiếp ứng dụng là một phần mềm đặc biệt được sử dụng để chuyển tiếp các yêu cầu ứng dụng giữa hai mạng kết nối với nhau. Một phương pháp khác là cho phép người sử dụng đăng nhập (login) vào máy chủ hai cửa và sau đó truy nhập các dịch vụ bên ngoài từ giao diện mạng bên ngoài của máy chủ hai cửa này. Hình 9.12. mô tả một ví dụ bức tường lửa có sử dụng bộ chuyển tiếp.



Hình 9.12. ***Một máy tính chủ hai cửa như bộ chuyển tiếp tin tức***

9.7. ĐỊNH TUYẾN SÀNG LỌC

9.7.1. Giới thiệu

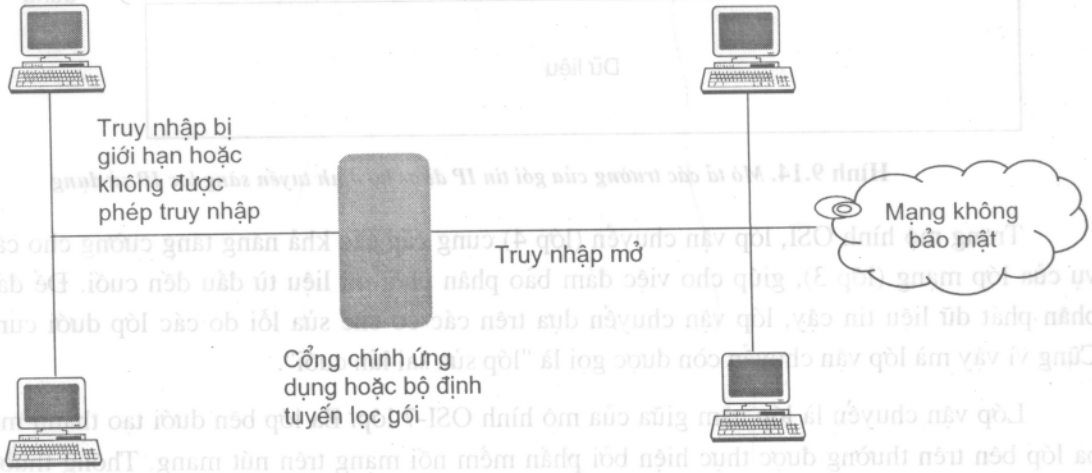
Định tuyến sàng lọc là một chức năng quan trọng trong cấu trúc bức tường lửa. Nó có khả năng định tuyến và sàng lọc các gói tin dựa trên các tiêu chuẩn như loại giao thức, các trường địa chỉ nguồn và địa chỉ đích đối với một loại giao thức nhất định và các trường điều khiển vốn là thành phần của giao thức.

Hiện nay trên thị trường có nhiều loại bộ định tuyến sàng lọc (screening router) có khả năng cung cấp một cơ chế rất mạnh để điều khiển các loại lưu lượng trên mạng, các lưu lượng này có thể tồn tại trên bất kỳ đoạn mạng nào. Như vậy có thể hạn chế các dịch vụ có khả năng làm hại đến an toàn mạng.

Các bộ định tuyến sàng lọc có thể phân biệt giữa lưu lượng mạng dựa trên loại giao thức và các giá trị của trường giao thức trong gói tin. Khả năng của bộ định tuyến phân biệt giữa các gói tin và hạn chế các gói tin trên các cổng của nó dựa trên các tiêu chuẩn đặc trưng giao thức và được gọi là lọc gói tin. Cũng vì lý do đó mà các bộ định tuyến sàng lọc còn có tên gọi là các bộ định tuyến lọc gói tin (packet filter router).

Ở đây cũng cần lưu ý rằng, các bộ định tuyến được thiết kế nhằm mục đích tạo mọi điều kiện thuận lợi cho công tác truyền tin trong lúc các bộ lọc gói tin được hiểu như một cái sàng được cài thêm vào để lọc và ngăn chặn các gói tin không được phép truyền tiếp tục.

Các bộ lọc gói tin khi được cài vào bộ định tuyến thì bộ định tuyến đó trở thành bộ định tuyến sàng lọc của bức tường lửa. Các gói không được phép sẽ bị loại bỏ. Việc lọc đó có thể thực hiện trong bộ định tuyến sàng lọc, trong bộ định tuyến của nhà cung cấp dịch vụ Internet hoặc đặt cả hai nơi. Hình 9.13 mô tả bộ định tuyến sàng lọc được cài đặt giữa mạng được bảo vệ và mạng ngoài không tin cậy. Việc cài đặt các bảng lọc là một công việc khá phức tạp và mỗi bộ định tuyến của nhà sản xuất có cú pháp và khuôn dạng khác nhau. Nếu như trong bảng lọc không tính hết các điều kiện có thể thì đó là lỗ hổng bảo mật.



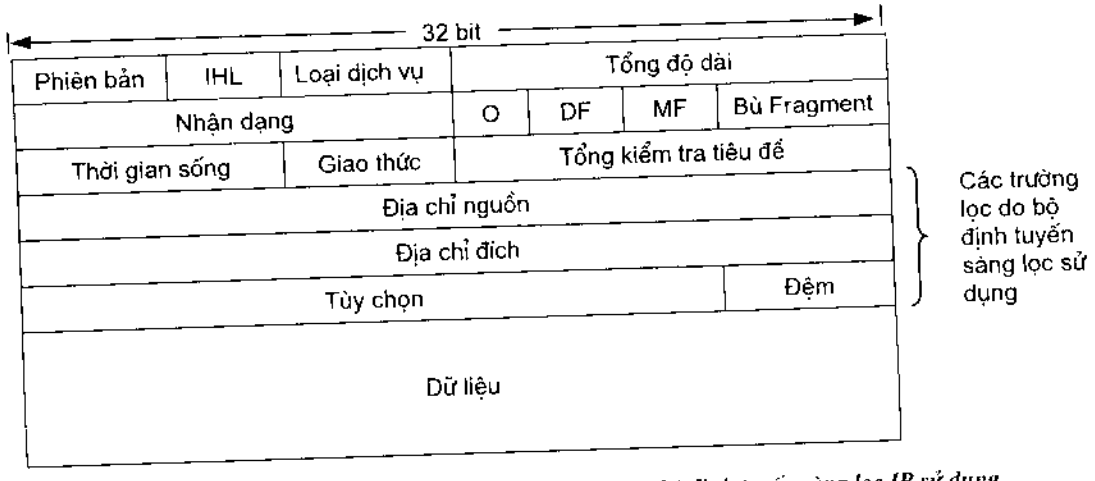
Hình 9.13. ***Bộ định tuyến sàng lọc và mạng hỗn hợp***

9.7.2. Chức năng và hoạt động của bộ định tuyến sàng lọc

Các bộ định tuyến và định tuyến sàng lọc hoạt động ở *lớp mạng* (lớp 3) của mô hình cấu trúc mở OSI - 7 lớp. Biết rằng, lớp mạng được xây dựng trên cơ sở kết nối nút - tới- nút do lớp liên kết dữ liệu (lớp 2) cung cấp, bằng việc mở rộng các dịch vụ liên kết dữ liệu nút - tới - nút qua một mạng. Lớp mạng còn cung cấp dịch vụ định tuyến các gói tin giữa các nút được kết nối với nhau thông qua một mạng có độ phức tạp tùy ý. Ngoài việc định tuyến, lớp mạng còn có nhiệm vụ điều khiển lưu lượng và loại trừ tắc nghẽn. Lớp mạng cho phép hai mạng đấu nối liên kết với nhau bằng cách cài đặt một cơ chế định địa chỉ đồng nhất. Ví dụ các mạng cục bộ Token Ring và Ethernet có những loại địa chỉ liên kết dữ liệu khác nhau. Đối với các mạng TCP/IP thì khả năng này được cung cấp bởi giao thức liên kết mạng (IP).

Biết rằng, một địa chỉ IP bao gồm số mạng (còn được gọi là số nhận dạng của mạng) và số của máy tính chủ (còn được gọi là số nhận dạng của máy tính chủ). Bộ định tuyến có nhiệm vụ kiểm tra phần số mạng của địa chỉ IP đích trong gói IP và so sánh nó với các đầu vào của bảng định tuyến (routing table). Nếu có một đầu vào trong bảng này khớp với số mạng thì bộ định tuyến sẽ chuyển tiếp gói IP đó như chỉ thị trong bảng định tuyến. Nếu không có đầu vào nào khớp với số mạng và cũng chẳng có tuyến ngầm định thì gói IP sẽ bị loại bỏ.

Hình 9.14 mô tả các trường hợp trong gói IP mà theo đó bộ định tuyến sàng lọc có thể thực hiện chức năng lọc. Các chức năng sàng lọc có thể theo tùy chọn.



Hình 9.14. *Mô tả các trường của gói tin IP được bộ định tuyến sàng lọc IP sử dụng*

Trong mô hình OSI, lớp vận chuyển (lớp 4) cung cấp các khả năng tăng cường cho các dịch vụ của lớp mạng (lớp 3), giúp cho việc đảm bảo phân phối dữ liệu từ đầu đến cuối. Để đảm bảo phân phát dữ liệu tin cậy, lớp vận chuyển dựa trên các cơ chế sửa lỗi do các lớp dưới cung cấp. Cũng vì vậy mà lớp vận chuyển còn được gọi là "lớp sửa sai lần cuối".

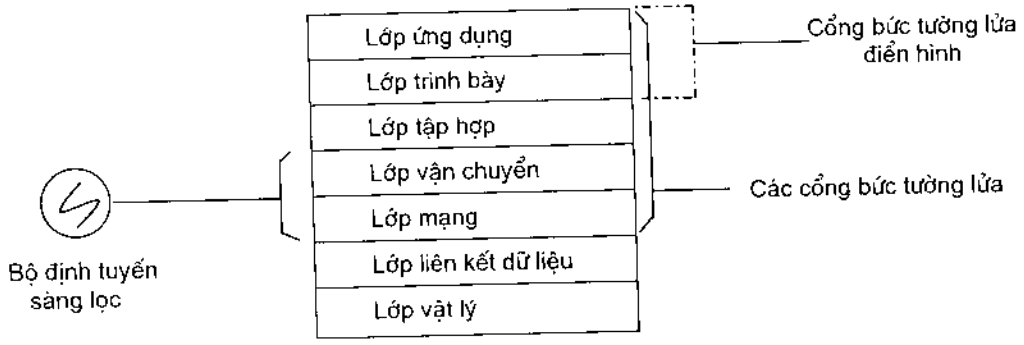
Lớp vận chuyển là lớp nằm giữa của mô hình OSI-7 lớp. Ba lớp bên dưới tạo thành mạng và ba lớp bên trên thường được thực hiện bởi phần mềm nối mạng trên nút mạng. Thông thường lớp vận chuyển cũng được cài đặt trên nút mạng và nhiệm vụ của nó là biến mạng con không tin cậy thành một mạng có độ tin cậy cao hơn.

Lớp vận chuyển còn chịu trách nhiệm tạo ra một số kết nối logic trên cùng một kết nối mạng, một quá trình được gọi là "ghép kênh" (multiplexing). Để xác định được đúng các yếu tố phần mềm trong lớp vận chuyển phải có một hình thức định địa chỉ chung hơn. Các địa chỉ này, được gọi là địa chỉ lớp vận chuyển và thường là một tổ hợp của địa chỉ mạng và số của điểm truy nhập dịch vụ lớp vận chuyển (SAP). Trong TCP/IP thuật ngữ số cổng (port number) được sử dụng để xác định các địa chỉ của lớp vận chuyển.

Các bộ định tuyến thường không thực hiện việc xử lý ở lớp vận chuyển nhưng bộ định tuyến sàng lọc có thể kiểm tra các trường của số cổng trong tiêu đề của TCP.

Lớp tập hợp (lớp 5) là lớp lợi dụng lớp vận chuyển để cung cấp các dịch vụ tăng cường và lớp trình bày (lớp 6) quản lý cách mà dữ liệu được biểu thị (mã dữ liệu, cú pháp, ngữ nghĩa).

Các bức tường lửa được sử dụng để bảo vệ mạng hoạt động ở lớp ứng dụng của mô hình OSI. Do các bức tường lửa hoạt động ở lớp cao nhất của mô hình OSI, cho nên chúng có sự truy nhập đến nhiều thông tin hơn so với các bộ định tuyến sàng lọc và có thể được lập trình để hoạt động thông minh hơn các bộ định tuyến sàng lọc. Hình 9.15 mô tả các bộ định tuyến sàng lọc và các bức tường lửa trong mối quan hệ với mô hình OSI.



Hình 9.15. ***Các bộ định tuyến sàng lọc, các bức tường lửa và mô hình OSI.***

Đôi khi các bộ định tuyến sàng lọc cũng còn được gọi là các cổng lọc gói tin (packet filter gateway). Một lý do để lý giải cho việc sử dụng thuật ngữ cổng (gateway) đặt tên cho thiết bị lọc gói tin là việc lọc dựa trên các cờ TCP được thực hiện tại lớp vận chuyển chứ không phải là chức năng của bộ định tuyến chỉ hoạt động tại lớp mạng của mô hình OSI. Các thiết bị hoạt động trên lớp mạng cũng được gọi là các cổng. Việc lọc gói tin được xem như là một biện pháp tăng cường an toàn mạng. Bộ lọc gói tin có thể được đặt giữa một hay nhiều nhánh mạng. Việc lọc gói tin là để hạn chế lưu lượng mạng đối với các dịch vụ cần phải từ chối và bộ lọc gói tin trên mỗi phía của các cổng của bộ định tuyến sàng lọc phải hoạt động khác nhau. Nói cách khác, các bộ lọc này không phải là đối xứng.

Bộ định tuyến sàng lọc được sử dụng như là hàng phòng vệ đầu tiên giữa mạng bên trong cần được bảo vệ và mạng bên ngoài không tin cậy (hình 9.16).

Hầu hết các bộ lọc gói hiện nay (bộ định tuyến sàng lọc hoặc các cửa ngỏ lọc gói) đều hoạt động theo phương thức sau đây:

1. Các tiêu chuẩn lọc gói phải được nhớ đối với các cổng của thiết bị lọc gói. Các tiêu chuẩn lọc gói đó được gọi là các "qui tắc lọc gói tin"

2. Khi một gói nào đó về đến cổng, chỉ có mào đầu của gói đó được phân tích. Hầu hết các thiết bị lọc gói tin chỉ kiểm tra các trường thuộc mào đầu của các gói IP, TCP hoặc UDP.

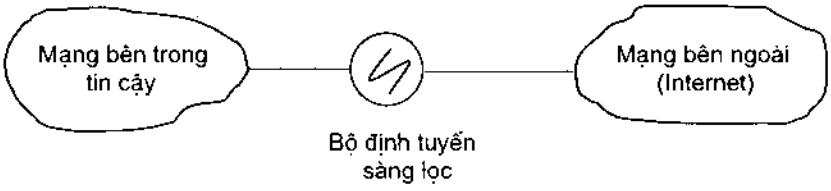
3. Các qui tắc gói tin phải được lưu trữ theo một thứ tự qui định. Mỗi qui tắc được áp dụng cho gói theo thứ tự nó được lưu trữ.

4. Nếu việc gửi đi hoặc thu nhận một gói bị chặn lại thì gói đó không được phép.

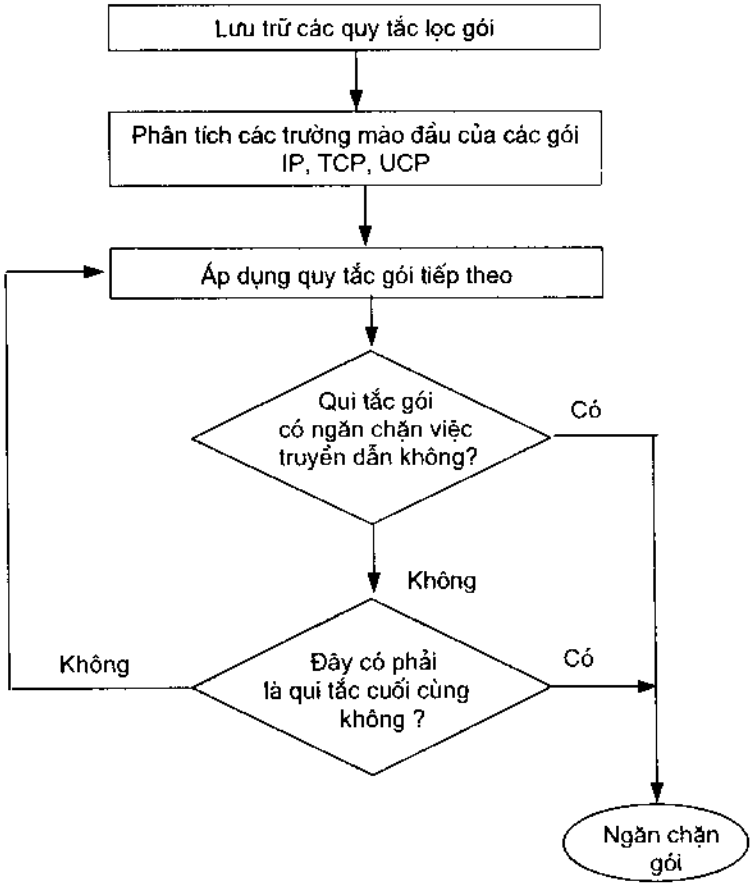
5. Nếu một qui tắc cho phép truyền hoặc thu một gói thì gói tin đó được phép tiếp tục.

6. Nếu gói tin không thỏa mãn một qui tắc nào đó thì nó sẽ bị chặn (các qui tắc này được mô tả ở lưu đồ thuật toán hình 9.17).

Việc đặt các qui tắc theo một thứ tự đúng đắn là hết sức quan trọng. Nếu các qui tắc lọc gói bị đặt theo thứ tự sai thì có thể sẽ từ chối các dịch vụ hợp lệ trong khi lại để lọt các dịch vụ cần từ chối.



Hình 9.16. ***Bộ định tuyến sàng lọc ở hàng phòng vệ đầu tiên***



Hình 9.17. ***Lưu đồ thuật ngữ hoạt động của bộ lọc gói***

Việc sử dụng bộ định tuyến sàng lọc ở bức tường lửa cũng có một số ưu điểm và nhược điểm. Nếu được xây dựng thiết kế tốt, bộ định tuyến sàng lọc có thể ngăn chặn hoặc loại bỏ những gói tin không mong muốn xâm nhập vào mạng cần được bảo vệ. Các con đường xâm nhập của các gói tin đó là từ các mạng không tin cậy, chủ yếu là khi mạng được kết nối với Internet hoặc mạng công cộng. Tuy vậy, nó có nhược điểm là với các phần mềm mua sẵn khi cần nâng cấp thì hiệu quả sàng lọc bị suy giảm. Giá thành phần cứng, phần mềm và bảo trì cho hệ thống cũng gia tăng.

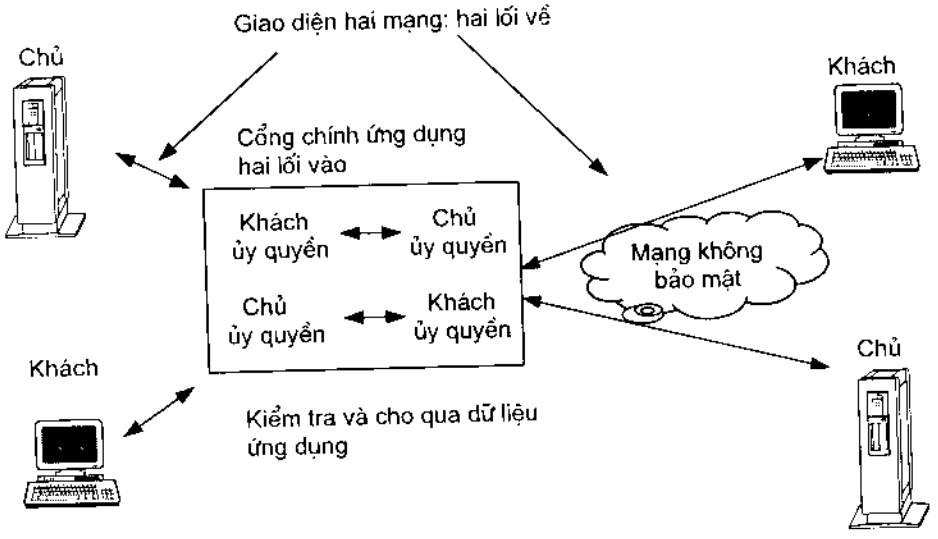
Bộ lọc gói tin đơn độc có thể sử dụng thích hợp nếu như mạng được tập trung, không có trung tâm bảo mật chắc chắn và số lượng máy chủ trong mạng ít. Nó được sử dụng riêng lẻ nếu các máy chủ không sử dụng Unix mà sử dụng Novell NetWare cũng như các trạm làm việc sử dụng DOS/Windows hoặc các hệ điều hành không Unix.

Ở đây cũng cần lưu ý rằng, việc bảo mật trên máy chủ chưa phải là một cấp tốt, nhất là khi số máy chủ tăng thì khả năng có thể bị chọc thủng đâu đó.

9.8. CÁC BỨC TƯỜNG LỬA CỔNG CHÍNH MỨC ÁP ỨNG DỤNG VÀ MỨC KẾT NỐI.

Các bức tường lửa cổng chính được chia thành hai loại: các cổng chính mức ứng dụng và các cổng chính mức kết nối.

Bức tường lửa cổng chính ứng dụng là ngược lại với bức tường lửa định tuyến. Trong khi các bức tường lửa định tuyến hướng về việc điều phối các loại lưu lượng khác nhau như đã phân tích ở các mục trên thì bức tường lửa cổng chính ứng dụng sử dụng các mã được viết riêng biệt cho các ứng dụng riêng biệt. Các tập mã đó được gọi là "các dịch vụ uỷ quyền" và nó tồn tại ở cả hai bên khách và chủ trong cổng chính ứng dụng vật lý. Tất cả các dịch vụ giữa khách và chủ đi qua cổng chính. Hình 9.18 mô tả một cổng chính hai lối về cùng với cấu trúc logic của một cổng chính ứng dụng.



Hình 9.18. ***Cổng chính hai lối về và cấu trúc logic của một cổng chính ứng dụng.***

Các bức tường lửa cổng chính mức ứng dụng có ưu điểm là:

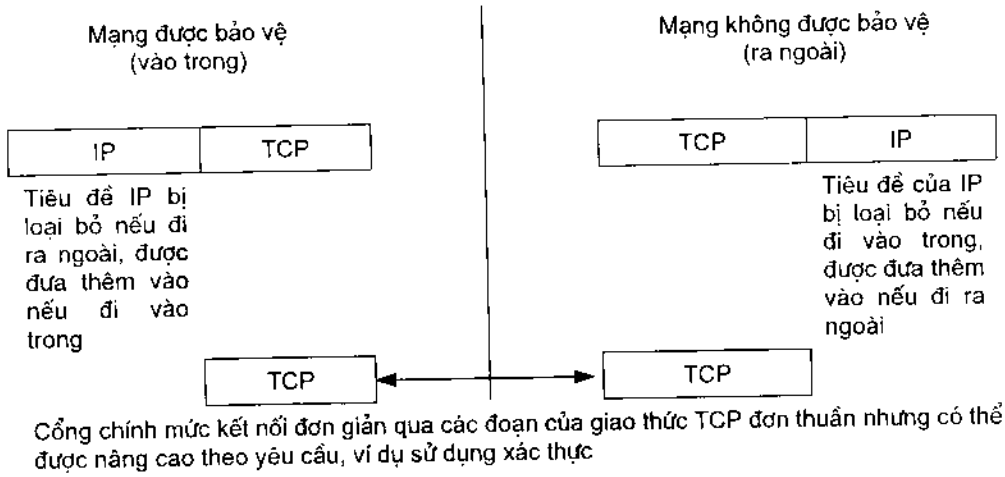
- Chỉ cho phép các dịch vụ có uỷ quyền;
- Giao thức cũng có thể bị lọc;
- Che dấu tên các máy chủ và các địa chỉ IP, bên ngoài chỉ nhìn thấy cổng chính;
- Thuận lợi cho việc xác thực;
- Thủ tục lọc đơn giản;
- Cổng chính có thể kiểm tra các địa chỉ IP nguồn;

- Thường được sử dụng cho TELNET, FTP, E-mail, X-Windows và các dịch vụ khác.

Tuy vậy, nó có nhược điểm là:

- Phải tiến hành hai bước cho các ứng dụng khách - chủ: kết nối với cổng chính máy chủ; kết nối cổng chính như là khách với máy chủ bên ngoài (hoặc ngược lại);
- Tốn kém hơn so với bộ định tuyến lọc gói;
- Cần phải có dịch vụ phụ trợ là các dịch vụ uỷ quyền;
- Khó thích nghi với dịch vụ mới ví dụ như WWW và Gopher;
- Việc cài đặt trong cổng chính giới hạn bởi thời gian tồn tại của ứng dụng.

Hình 9.19 mô tả cấu trúc logic của một cổng chính mức kết nối của một mạng được bảo vệ kết nối với một máy chủ bên ngoài. Nếu như việc sử dụng là thường xuyên thì quá trình có thể là tự động. Các cổng chính mức kết nối đôi lúc cũng được ưa chuộng nếu như mạng có nhiều lưu lượng đi ra ngoài. Trong trường hợp này, các kết nối của người gọi đến một cổng TCP trên cổng chính; cổng chính sau đó lại kết nối với bên nhận. Mỗi một lần kết nối được thiết lập thì chương trình chuyển tiếp sao chép dữ liệu và chuyển theo hướng đó.



Hình 9.19. ***Cổng chính mức kết nối***

9.9. CÁC CẤU HÌNH BỨC TƯỜNG LỬA

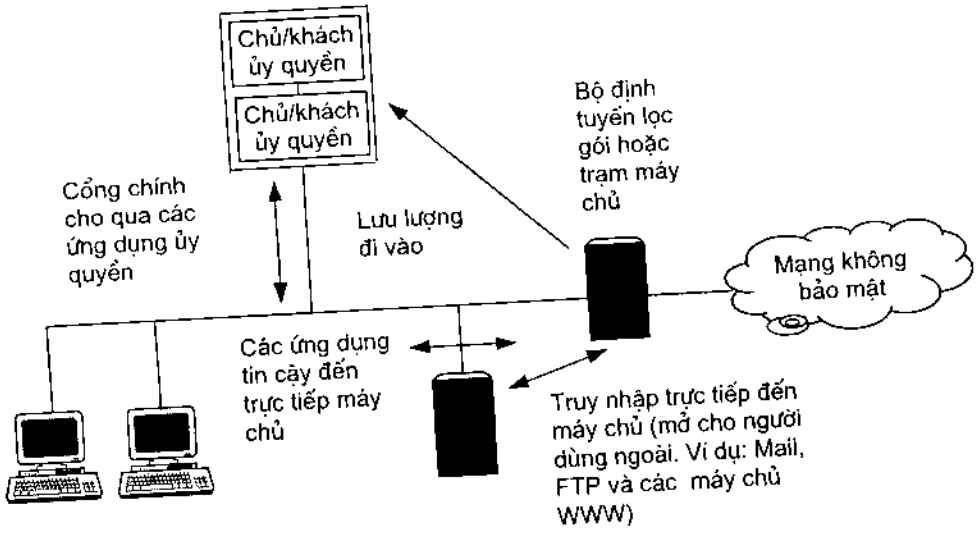
9.9.1. Định tuyến logic trong cấu hình bức tường lửa

Trong các mục trên đã giới thiệu nguyên lý cơ bản về hoạt động của các bức tường lửa sử dụng bộ lọc gói và bức tường lửa cổng chính mức ứng dụng. Chúng có thể được sử dụng riêng rẽ hoặc phối hợp với nhau.

Theo sự gia tăng của mức bảo mật, có thể sắp xếp các cấu hình theo thứ tự sau:

1. Các bức tường lửa lọc gói chỉ lọc đơn giản các gói;
2. Các bức tường lửa máy chủ sàng lọc cho phép các dịch vụ tin cậy qua cổng chính;
3. Các bức tường lửa mạng phụ sàng lọc có bảo mật như cổng chính hai lối về;
4. Các cổng chính hai hướng tuyệt đối không cho phép truy cập vào hệ thống bên trong.

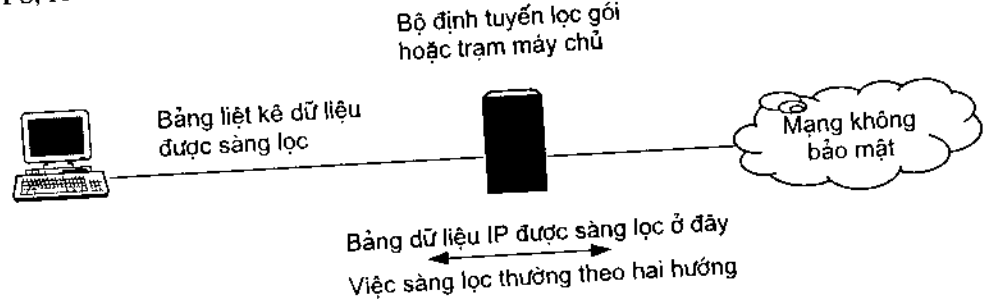
Hình 9.20 mô tả quan hệ logic giữa các thiết bị. Ở một thời điểm có thể có lưu lượng của một số máy đi về máy chủ thông tin qua một cổng chính ứng dụng, trong lúc đó có lưu lượng của một số máy khác trực tiếp về máy chủ thông tin. Việc quyết định đó phụ thuộc vào phần mềm cổng chính có sẵn hoặc được viết và đánh giá các nguy hiểm nằm ở việc cho phép truy cập trực tiếp vào máy chủ thông tin. Nếu được xây dựng theo kiểu trạm chủ pháo đài thì các nguy hiểm sẽ ít. Với các mạng phụ được sử dụng thì sử dụng một địa chỉ loại C để giới hạn số các địa chỉ IP.



Hình 9.20. ***Định tuyến logic***

9.9.2. Bức tường lửa lọc gói

Hình 9.21 mô tả mô hình đơn giản bức tường lửa lọc gói được sử dụng trong bảo mật nối mạng với Internet. Bộ lọc có thể được cài đặt ở cổng kết nối với Internet hoặc ở một mạng phụ nào đó, phần còn lại là để mở. Ở các hệ thống được bảo vệ bởi bức tường lửa thì có truy cập vào Internet nhưng hướng truy cập từ Internet về thì chủ yếu là bị khoá, và đặc biệt nguy hiểm với các dịch vụ như NFS, X-Windows và NiS.



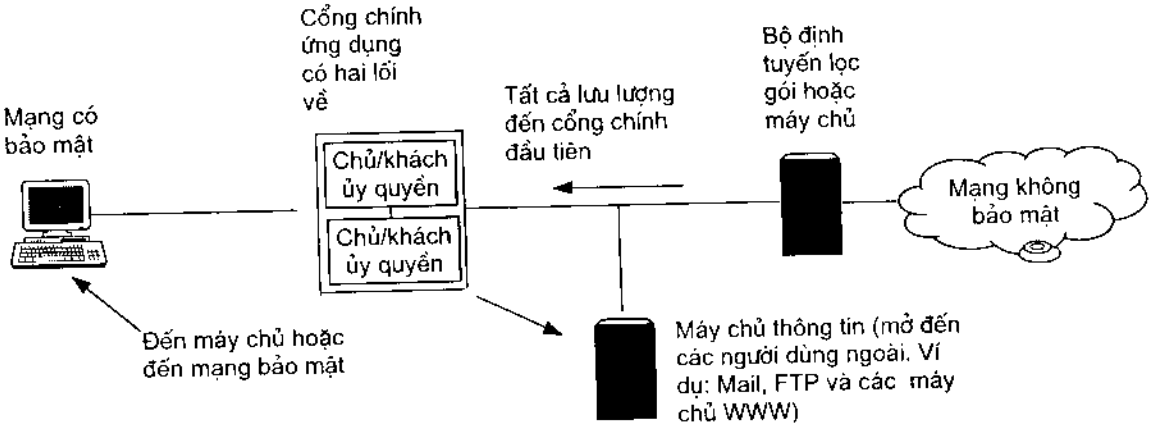
Hình 9.21. ***Bức tường lửa lọc gói***

Ưu điểm của bức tường lửa lọc gói là thuật tiện sử dụng, tốn kém ít và có thể chặn các kết nối đi và về với các trạm chủ hoặc các mạng riêng lẻ hoặc các cổng như các địa chỉ IP của các nguồn không tin cậy.

Nhược điểm của bức tường lửa là không có nhiều mức bảo mật khi yêu cầu bảo mật gia tăng và việc loại trừ là được cài đặt cứng không linh hoạt trong bảo mật.

9.9.3. Cổng chính hai lối về

Cổng chính hai lối về là một kiểu cổng chính mức ứng dụng, nó có nhiều ưu điểm hơn bộ định tuyến sàng lọc đơn giản. Thuật ngữ cổng chính hai lối về được qui cho một trạm chủ cho nhiều hơn một địa chỉ IP. Trong trường hợp này mỗi giao diện mạng có một địa chỉ IP của nó như mô tả ở hình 9.22.



Hình 9.22. *Bức tường lửa cổng chính hai lối về*

Trạm chủ không thể gửi chuyển tiếp các gói từ một giao diện mạng này đến một giao diện mạng khác nhưng trạm chủ chuyển tiếp các gói đến các máy chủ thông tin mở như E-mail, Gopher hoặc các máy WWW.

Trạm chủ cũng không thể nhận hoặc phân phát các gói được định tuyến từ nguồn. Trạm chủ có thể xác thực các truy cập đi và về của người sử dụng, cho qua các yêu cầu có uỷ quyền và các trả lời của các máy nội bộ có bảo mật. Mạng giữa bộ định tuyến lọc gói và cổng chính trong đó có các máy mở được gọi là mạng phụ được sàng lọc. Bởi vì ở đây có hai tầng bảo mật cho nên cổng chính hai lối về rất bảo mật và cũng không phức tạp lắm trong việc cài đặt và quản lý.

Cũng giống như ở cổng chính ứng dụng, tên và số của mạng nội bộ là hoàn toàn riêng và nó không thể qua cổng chính. Một số cổng chính đảm bảo rằng, các tiêu đề của E-mail chứa trong các địa chỉ IP và tên của trạm chủ thực, như vậy tại một thời điểm nào đó nó không thể ngẫu nhiên qua cổng chính ra ngoài.

Ưu điểm của cấu hình này là một số các dịch vụ có thể công khai nhưng vẫn đảm bảo an toàn. Máy chủ thông tin trong trường hợp này có thể là một thư viện thông tin cho người dùng truy cập tìm kiếm thông tin, truy cập các dịch vụ Web v.v... Các máy chủ thông tin đó được bảo vệ do cổng chính hoạt động theo uỷ quyền. Nếu như máy chủ bị thâm nhập thì cổng chính thực hiện việc ngăn chặn truy cập đó.

Cổng chính hai lối về cần được bảo mật nghiêm khắc, tất cả các dịch vụ không cần thiết và có hại đều được ngăn chặn. Các hoạt động TCP đặc biệt hoạt động như các tác nhân uỷ quyền.

Cổng chính hai lối về có ưu điểm là bảo mật tốt nhưng có khả năng dễ bị tắc nghẽn khi một điểm nào đó bị lỗi.

9.9.4. Bức tường lửa trạm chủ có sàng lọc

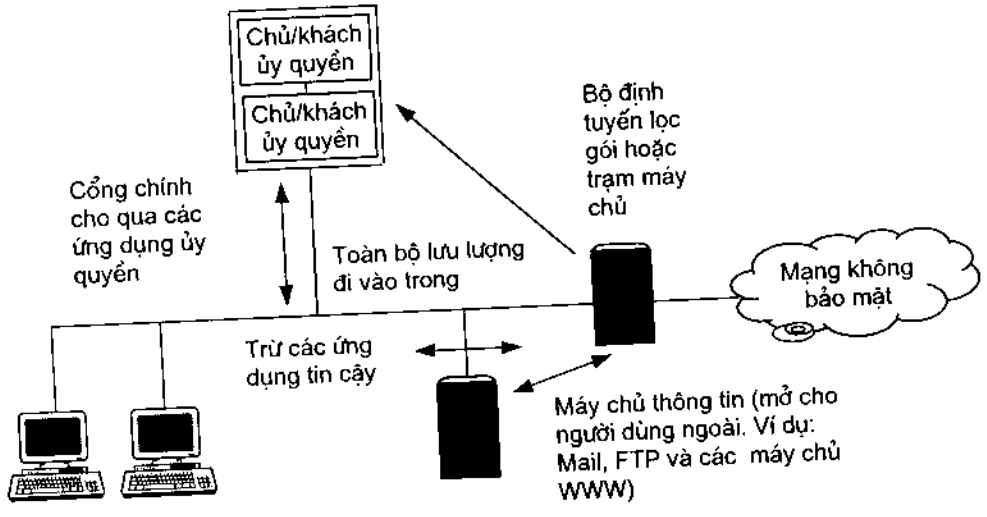
Hình 9.23 mô tả cấu hình bức tường lửa trạm chủ có sàng lọc mềm dẻo trong bảo mậtỞ đây các ứng dụng tin tưởng đi về trong có thể qua trực tiếp đến mạng nội bộ sau khi qua bộ lọc gói. Nếu như các dịch vụ uỷ quyền không có hoặc các nguy hiểm sử dụng các dịch vụ như vậy được đánh giá là có thể chấp nhận thì các ứng dụng đó được xem là tin cậy.

Bộ định tuyến lọc gói ở đây có các định nghĩa về thủ tục không đến nỗi phức tạp nếu như bộ lọc gói được sử dụng riêng rẽ trong đó tất cả lưu lượng ứng dụng không tin cậy được gửi qua cổng chính.

Cần lưu ý rằng, do các ứng dụng tin cậy được truy cập trực tiếp cho nên nếu như có ai đó bên trong mạng tấn công vào một ứng dụng tin cậy thì không có cách nào biết được.

Ưu điểm của cấu hình trạm chủ có sàng lọc là chỉ cần một giao diện mạng cổng chính ứng dụng. Dòng lưu lượng hướng vào trong dành cho cổng chính ứng dụng được đi qua bộ lọc gói. Những lưu lượng nào không qua bộ lọc gói sẽ bị từ chối. Với lưu lượng hướng ra ngoài thì bộ định tuyến sẽ từ chối các lưu lượng không được đến từ cổng chính ứng dụng. Các điều phân tích trên là với giả thiết ở cổng chính luôn luôn theo tuyến logic nối tiếp.

Cấu hình trạm chủ có sàng lọc có thể làm việc để định tuyến lưu lượng trực tiếp ở cổng chính thông tin. Trong trường hợp này bộ định tuyến có ba đích hướng về, đó là: cổng chính, máy chủ thông tin và một trạm chủ nội bộ nhận một ứng dụng tin cậy.



Hình 9.23. ***Bức tường lửa trạm chủ có sàng lọc***

9.9.5. Bức tường lửa mạng phụ có sàng lọc

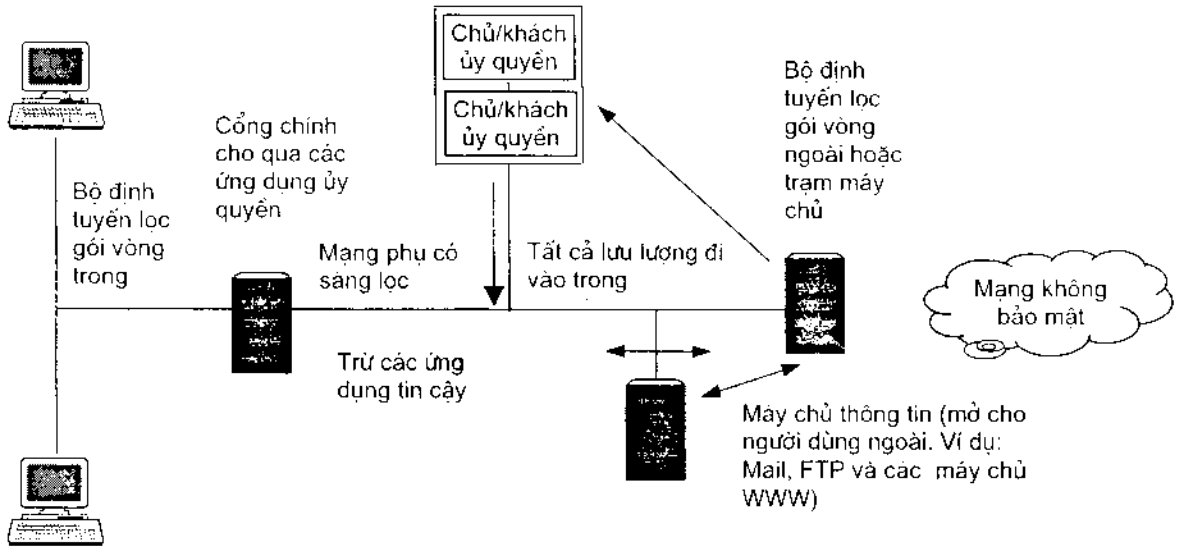
Hình 9.24 mô tả bức tường lửa mạng phụ có sàng lọc. Ở đây có thêm một bộ định tuyến lọc gói khác để ngăn cách mạng nội bộ với mạng phía bên kia. Mạng kết nối hai bộ định tuyến, cổng chính và một số máy chủ thông tin được gọi là mạng phụ có sàng lọc. Ở đây các modem để kết nối với máy chủ truyền thông, các thiết bị mạng LAN tương thích hoặc các máy chủ đầu cuối (xem như là không bảo mật) được đặt ngoài bộ định tuyến lọc gói.

Bộ định tuyến vòng ngoài: hướng vào, bộ định tuyến hoặc trạm chủ có lọc gói ở vòng ngoài sẽ gửi E-mail và lưu lượng ứng dụng đến cổng chính riêng (dịch vụ uỷ quyền E-mail thường thực hiện trên các cổng chính ứng dụng). Một số lưu lượng nào đó ở hướng về có thể bị từ chối. Hướng ra, đi theo đường ngược lại. Lưu lượng hướng ra đến Internet được định tuyến nhưng có một số lưu lượng nào đó sẽ bị loại bỏ tại trạm chủ. Mặt khác bộ định tuyến cũng loại bỏ một số lưu lượng không rõ kết nối xuất xứ.

Bộ định tuyến vòng trong: Lưu lượng hướng vào từ cổng chính ứng dụng (bao gồm cả E-mail) đi qua bộ định tuyến vòng trong và tại đây cũng được lọc gói.

Mạng phụ có sàng lọc bảo mật tốt hơn so với trạm chủ có sàng lọc nhưng nó cũng có thể cho phép các ứng dụng tin cậy như thế qua cổng chính ứng dụng. Với lý do đó, chúng được liệt kê trước những gì không đáng tin cậy đến bức tường lửa cổng chính có hai lối về. Mạng phụ có sàng lọc là bộ lọc tinh để xác định vị trí các máy chủ thông tin, các modem và các hệ thống có khả năng nguy hiểm.

Mạng phụ có sàng lọc có ưu điểm là bảo mật tốt, không kết nối trực tiếp với mạng không được bảo mật nhưng nó có nhược điểm là phụ thuộc vào các bộ định tuyến ở các cấp bảo mật và cần phải có thêm một bộ định tuyến ở vòng trong.



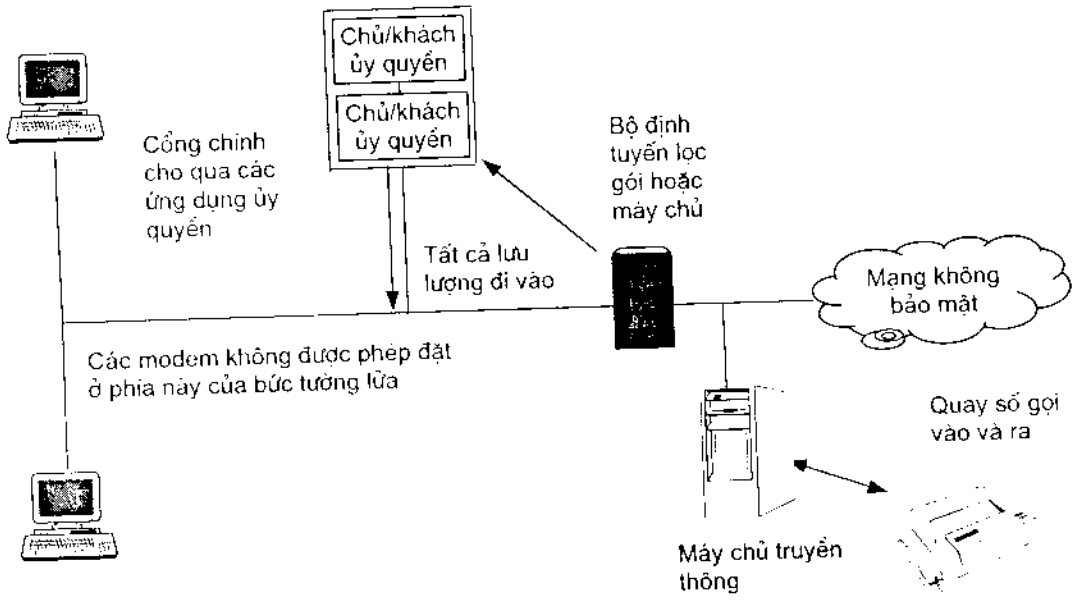
Hình 9.24. *Bức tường lửa mạng phụ có sàng lọc*

9.9.6. Vị trí của modem trong các cấu hình bức tường lửa

Modem là từ viết tắt của bộ điều chế (MOdulator) và bộ giải điều chế (DEModulator). Các tín hiệu máy tính muốn truyền tải trên đường truyền tin, ví dụ mạng điện thoại công cộng, thì phải biến đổi tín hiệu đó thành một tín hiệu có dạng thích hợp (điều chế) và phía nhận sẽ biến đổi ngược lại (giải điều chế).

Chức năng điều chế và giải điều chế đó do một thiết bị modem đảm nhiệm. Thiết bị modem có thể là riêng rẽ hoặc cũng có thể được bộ trí ngay trong máy tính chủ hoặc máy tính truyền thông. Người sử dụng khi cần quay số truy cập mạng thường phải vào một mật khẩu hoặc sử dụng dấu hiệu xác thực để truy cập. Người sử dụng có thể sử dụng máy tính xách tay, qua modem để truy cập mạng. Đây là một sự tiện lợi cho người sử dụng nhưng đồng thời là một nguy hiểm mà kẻ xâm nhập có thể lợi dụng để xâm nhập truy cập mạng.

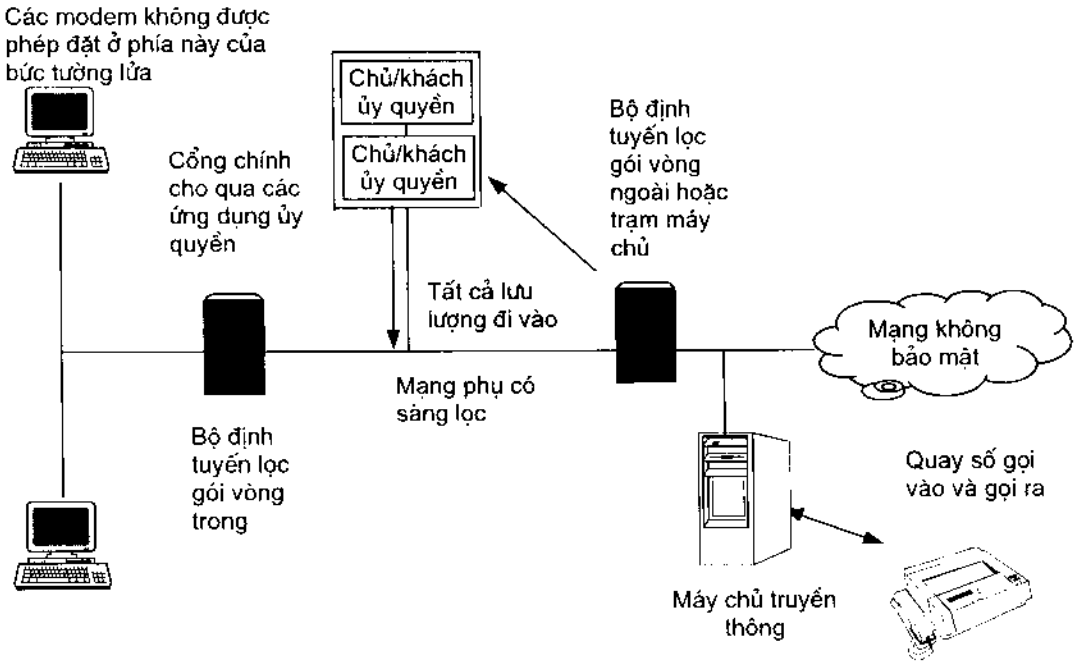
Hình 9.25 mô tả bức tường lửa trạm chủ có sàng lọc và modem ngoài. Ở đây việc lọc và chuyển tiếp lưu lượng modem cũng giống như từ Internet. Thêm vào đó, việc xác thực hướng vào được cung cấp bởi cổng chính ứng dụng. Đối với hướng ra, các trạm chủ bên trong có thể ngăn chặn nhờ bộ định tuyến lọc gói.



Hình 9.25. *Vị trí modem ở bức tường lửa trạm chủ có sàng lọc*

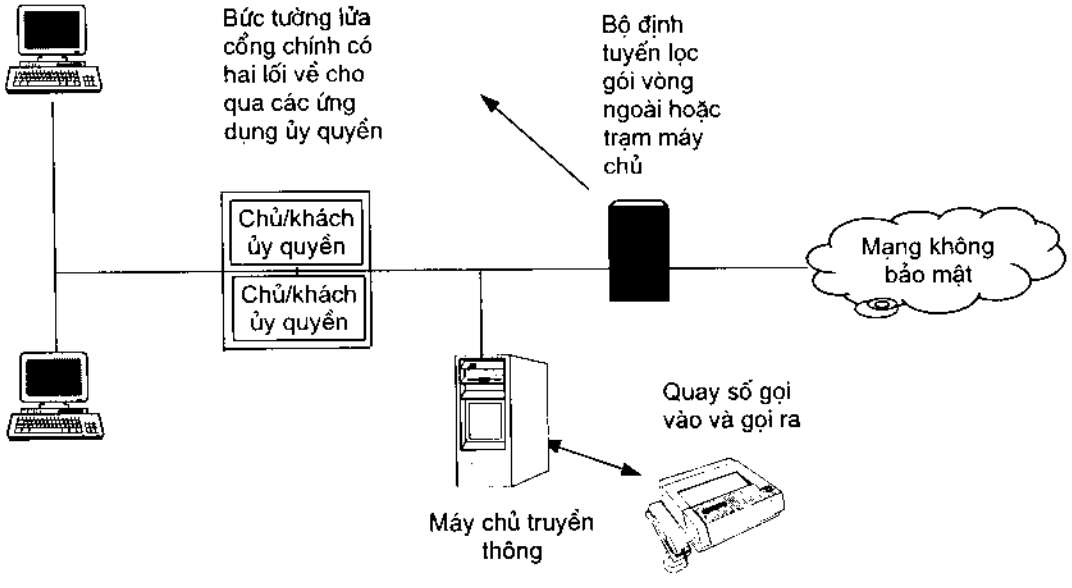
Những kẻ xâm nhập từ mạng không được bảo mật hướng mục tiêu qua modem và nếu như chúng đạt được việc truy cập thì sau đó chúng có thể sử dụng truy cập đó để xâm nhập các hệ thống khác. Ngăn chặn điều đó được thực hiện ở máy chủ truyền thông hoặc ở máy chủ đầu cuối để loại bỏ các xâm phạm kết nối có sự khác biệt nào đó ở cổng chính ứng dụng.

Hình 9.26 mô tả một cấu hình có modem đặt ở bức tường lửa mạng phụ có sàng lọc. Ở đây cho phép kiểm tra việc sử dụng modem được xuất xứ từ bên trong hoặc từ bên ngoài. Bộ lọc vòng ngoài ngăn chặn các kết nối bất cứ từ đâu đến chỉ trừ cổng chính ứng dụng. Bộ lọc vòng trong bắt buộc các yêu cầu của modem theo hướng ra phải qua cổng chính.



Hình 9.26. *Bức tường lửa mạng phụ có sàng lọc và vị trí của modem.*

Hình 9.27 mô tả vị trí modem trong cấu hình bức tường lửa cổng chính có hai lối về và cổng chính hạn chế truy cập đến modem từ bên trong. Cũng giống như ở bức tường lửa mạng phụ có sàng lọc, việc truy cập từ ngoài đến modem được ngăn chặn bởi bộ định tuyến lọc gói.



Hình 9.27. *Bức tường lửa cổng chính hai lối về và vị trí đặt modem*

Trong tất cả các cấu hình trên thì việc truy cập vào được hạn chế bởi việc xác thực cải tiến do cổng chính ứng dụng cung cấp. Việc xác thực ở máy chủ truyền thông có thể không cần thiết. Tuy

vậy, nếu việc truy cập được phép trực tiếp đến Internet thì điều kiện bảo mật bổ sung cần được đảm bảo.

Ở đây cần lưu ý một điều quan trọng là: Các máy chủ có trang bị modem trong đó không được kết nối với mạng bảo mật. Chúng có thể đi qua tất cả các bức tường lửa.

9.10. VIRÚT VÀ CÁC PHƯƠNG PHÁP PHÒNG CHỐNG

Virút là một chương trình máy tính có khả năng tự ghi và xâm nhập vào các chương trình khác bằng cách "lây nhiễm" vào các chương trình đó và thực hiện các tác động không mong muốn. Cũng vì vậy một virút cần một chương trình chủ điều hành hoạt động như một vật chủ trung gian, một đường dẫn. Các tác động không mong muốn gây ra do một virút có thể xếp từ loại trò chơi nghịch ngợm đến các tác hại về chương trình và dữ liệu rất quan trọng. Các lây nhiễm của virút cũng thường theo từng đợt, có thể cùng lúc lây nhiễm nhiều thiết bị trong một thời gian dài. Các lây nhiễm file tự chúng xâm nhập vào các file điều hành, điển hình là các file .COM và các file .EXE. Cũng chính vì vậy mà một số file nào đó chứa mã điều hành có thể là vật chủ trung gian cho virút. Cho đến nay có rất nhiều loại, nhiều dạng virút khác nhau. Có thể xếp chúng vào các dạng virút lây nhiễm thường gặp sau đây:

- Các lây nhiễm file tác động trực tiếp chọn một hoặc một vài chương trình khác để lây nhiễm mỗi lúc mà chương trình chứa đựng nó làm việc. Virút Vienna là một loại lây nhiễm file tác động trực tiếp.

- Các lây nhiễm cư trú được đặt trong bộ nhớ lúc ban đầu một chương trình bị lây nhiễm làm việc. Virút Jerusalem là ví dụ virút lây nhiễm cư trú.

- Các virút kết bạn tạo ra một chương trình lây nhiễm mới xem như một chương trình có chủ ý.

- Các virút có vỏ bọc làm bất cứ gì có thể để tránh bị phát hiện và làm phức tạp thêm các chương trình phát hiện.

Thông thường các virút lây nhiễm không tác động ngay sau khi lây nhiễm mà với một thời gian nào đó mới tác động, do đó chúng còn có tên gọi là các quả bom thời gian hoặc bom logic:

- Bom thời gian (hẹn giờ) là một chương trình sẽ phát nổ ở một ngày hoặc một thời gian nào đó. Ví dụ loại virut Michelangelo là một bom thời gian phát nổ vào ngày 6 tháng 3, là ngày sinh của nghệ sĩ.

- Bom logic là một chương trình được phát nổ do một vài sự kiện nào đó, có thể là khi mở file. Một sự bực tức nào đó có thể cài bom logic và khi phát nổ kèm theo một thông báo. Các bom logic thường hướng vào máy chủ.

Cũng đã có những nhà bán hàng cài đặt bom thời gian và/hoặc bom logic vào trong phần mềm của mình. Nếu như khách hàng không trả tiền sòng phẳng hoặc khách hàng xâm phạm trong việc sao chép thì các bom đó phát nổ, lúc đó chương trình hoặc hệ thống điều hành sẽ ngừng hoạt động. Người sử dụng nhận được thông báo đến tiếp xúc với nhà bán hàng.

Theo một nghiên cứu, con đường vật lý lây nhiễm của các virút được liệt kê theo tỉ lệ % sau đây:

- Các đĩa ngoài luồng (61%)
- Xuất xứ không biết (27%)
- Phân chia trên đĩa (19%)
- Phần mềm thương mại (14%)
- Phân chia trên board (13%)
- Các mạng LAN (6%)
- Thư điện tử (3%)
- Các mạng diện rộng (1%)
- Các đĩa quảng cáo, các đĩa chẩn đoán của kỹ thuật viên (1%)

Để phòng chống virút có thể sử dụng công cụ phần cứng, công cụ phần mềm hoặc kết hợp cả hai.

Về phần cứng, có bốn phương thức để giảm thiểu các lây nhiễm của virút, đó là:

1- Không sử dụng đĩa mềm ở các trạm làm việc trong mạng. Tất cả phần mềm cần thiết được cài ở máy chủ.

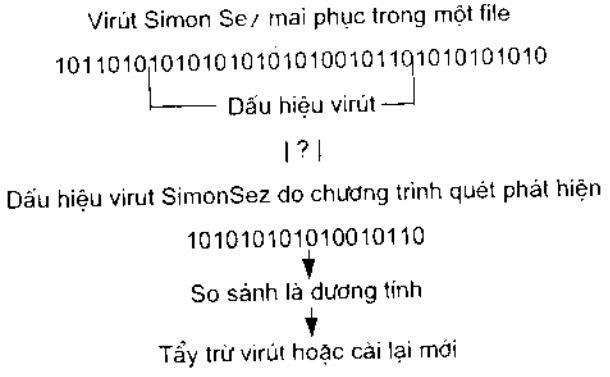
2- Khởi động máy từ xa. Sử dụng bộ nhớ PROM trong các giao diện mạng LAN để khởi động tại chỗ hoặc khởi động từ xa.

3- Sử dụng mật khẩu dựa vào phần cứng. Đây cũng là một phương thức bảo mật ngăn chặn các truy nhập bất hợp pháp.

4- Sử dụng khóa bảo vệ ghi. Các khóa này ngăn chặn các virút lây nhiễm từ các đĩa mềm.

Về phần mềm có ba dạng cơ bản thường được sử dụng để phòng chống các virút được liệt kê sau đây:

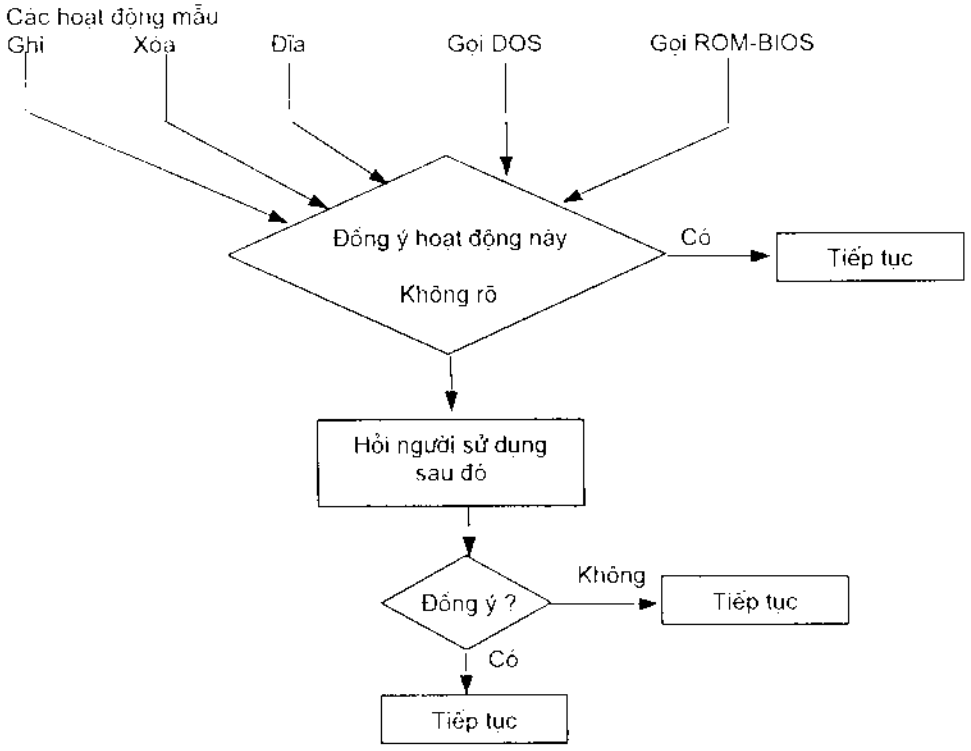
1- Các phần mềm quét. Các phần mềm này quét và phát hiện các chuỗi bit có dấu hiệu đặc trưng cho các chương trình virút. Phần mềm quét nà sẽ quét và kiểm tra bộ nhớ, đĩa, các sector khởi động, các file lệnh và thực hiện để xem có dấu hiệu của virút không. Hình 9.28 mô tả ví dụ một phương pháp quét để phát hiện dấu hiệu của virút.



Hình 9.28. *Mô tả một phương pháp quét virút*

Có thể thường xuyên sử dụng chương trình quét virút để kiểm tra virút trước khi thực hiện một công việc trên máy tính. Các chương trình quét thường được chạy với AUTOEXE.BAT trong các trạm làm việc. Vì vậy có thể mất nhiều thời gian để quét cho một ổ đĩa cứng mỗi lúc khởi động.

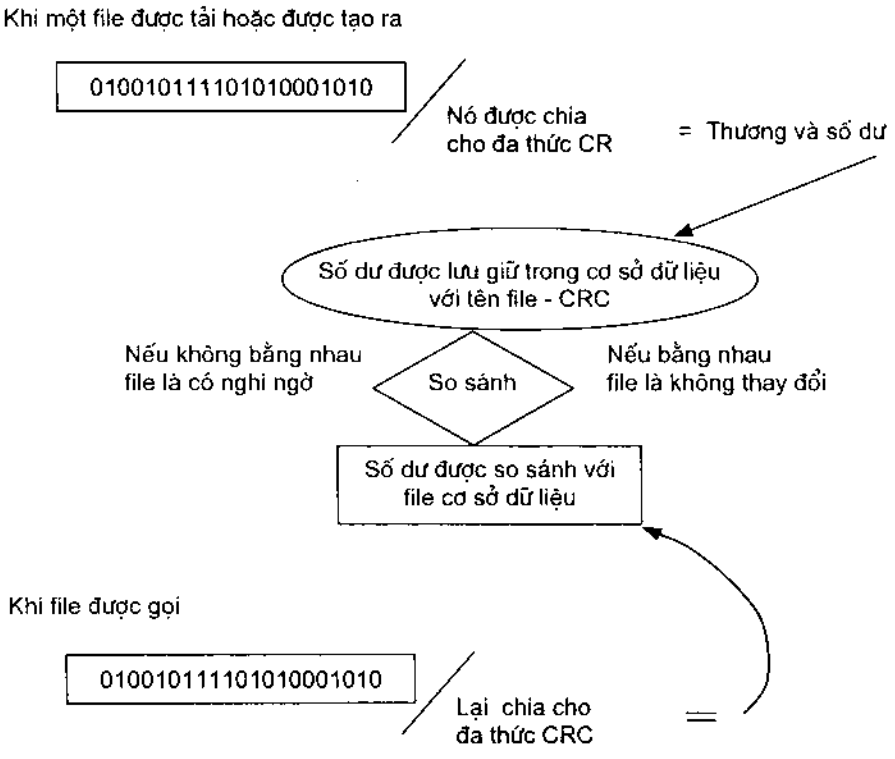
2- Giám sát tích cực. Hình 9.29 mô tả hoạt động của một phần mềm giám sát các tác động của virút liên quan đến các tác vụ: ghi, xoá, đĩa, ROM và ROM-BIOS



Hình 9.29. *Mô tả giám sát tích cực*

3- Kiểm tra tính toàn vẹn. Hình 9.30 mô tả một hoạt động của phần mềm kiểm tra tính toàn vẹn, xác định số nhị phân trên một chương trình không có virut đã biết và sau đó lưu giữ vào file cơ sở dữ liệu. Số kiểm tra được gọi là kiểm tra độ dư vòng, CRC (cyclical redundancy check). Đôi lúc giá trị "hàm băm" mật mã cũng được tính thay thế nhưng hàm đó cũng giống như CRC.

Khi chương trình được gọi thực hiện, phần mềm kiểm tra sẽ tính CRC trên chương trình sắp được thực hiện và so sánh nó với số kiểm tra trong cơ sở dữ liệu để phát hiện có bị lây nhiễm virút hay không.



Hình 9.30. ***Mô tả phần mềm kiểm tra tính toàn vẹn theo mã CRC***

Phụ lục

TẠO CÁC SỐ GIẢ NGẪU NHIÊN

Việc tạo các chuỗi số giả ngẫu nhiên là một công việc khá phức tạo. Có rất nhiều phương pháp khác nhau để tạo các chuỗi số giả ngẫu nhiên. Sau đây chỉ đề cập một số phương pháp thường được sử dụng trong thực tế và dễ thực hiện, đó là phương pháp được gọi là "phương pháp đồng dư tuyến tính" và "phương pháp cộng đồng dư".

Việc tạo các số giả ngẫu nhiên được xếp vào loại tốt khi nó thỏa mãn các điều kiện sau:

1. Các số ngẫu nhiên kế tiếp có thể được tính toán không quá phức tạp, giá thành chấp nhận.

2. Dãy được tạo ra xuất hiện đúng thực sự là ngẫu nhiên, có nghĩa là các số ngẫu nhiên kế tiếp là hoàn toàn độc lập, không có tương quan với các số khác. Chúng phù hợp với phân bố cho.

3. Chu kỳ của chúng (thời gian sau đó nó lặp lại) là rất lớn.

Ý tưởng cơ bản của việc tạo ra các số giả ngẫu nhiên cộng đồng dư là khá đơn giản. Bài toán được khởi đầu bởi giá trị Z_0, thường được gọi là hạt giống (seed), các giá trị Z_{i+1} được tính từ giá trị Z_i như sau:

$$Z_{i+1} = (aZ_i + c) \text{ modulo-m}$$

Bằng cách chọn một cách đúng đắn các thông số a, c, và m, thuật toán này sẽ tạo ra m giá trị khác nhau, sau đó lại bắt đầu tính toán tiếp. Đại lượng m được gọi là chiều dài của chu kỳ. Bởi vì giá trị sau của dãy phụ thuộc vào giá trị đang thực hiện cho nên chu kỳ khởi động sau không bao giờ có giá trị lặp lại. Phương pháp tạo số ngẫu nhiên cộng đồng dư tạo nên chu kỳ có độ dài là m nếu như ba điều kiện sau đây được thỏa mãn:

1. Các giá trị m và c là các số nguyên tố tương đối, có nghĩa là ước số chung lớn nhất của chúng có giá trị bằng 1.

2. Tất cả các thừa số nguyên tố của m chia hết cho a - 1

3. Nếu 4 chia hết cho m, có nghĩa là 4 cũng chia hết cho a -1.

Các điều kiện trên chỉ biểu thị về độ dài chu kỳ nào đó chứ nó không có nghĩa chu kỳ là ngẫu nhiên thực.

Ví dụ (phương pháp đồng dư tuyến tính):

Giả thiết với trường hợp: m = 16, c = 7 và a = 5. Kiểm tra các điều kiện nêu trên.

Bắt đầu xuất phát với $Z_0 = 0$, sẽ có:

$$Z_1 = (a.Z_0 + c) \text{ modulo m} = (5.0 + 7) \text{ modulo16} = 7$$

Tiếp tục tính toán theo biểu thức

$Z_{i+1} = (aZ_i + c)$ modulo-m, chúng ta sẽ có dãy:

0, 7, 10, 9, 4, 11, 14...

Phương pháp đồng dư tuyến tính tạo nên các dãy số ngẫu nhiên có chu kỳ tương đối ngắn, do tính ngẫu nhiên kém. Phương pháp có thể được khắc phục bằng cách sử dụng các phương pháp đồng dư cộng. Ở phương pháp này thì giá trị Z_i sẽ được xác định từ giá trị trước nó $(Z_{i-1}...Z_{i-k})$ theo quy luật sau:

$$Z_i = \left(\sum_{j=1}^{k} a_j . Z_{i-j} \right) \text{ modulo-m}$$

Nếu lựa chọn các hệ số a_j thích hợp thì các chu kỳ có độ dài là $m^k - 1$.

Ví dụ (phương pháp đồng dư cộng):

Chọn giá trị k = 7 và các hệ số $a_1 = a_7 = 1$;

$$a_2 = a_3 = a_4 = a_5 = a_6 = 0$$

Từ đây có thể mở rộng ví dụ trước. Bắt đầu chuỗi có thể tính được số hạng đầu tiên là: 0, 7, 10, 9, 4, 11, 14. Giá trị tiếp theo sẽ là (14+7)modulo-16 = 5; (5+10)modulo = 15. Cứ tiếp tục tính tiếp sẽ nhận được chuỗi các giá trị: 0, 7, 10, 9, 4, 11, 14, 5, 15...

Quan sát không thấy chu kỳ lặp lại. Với ví dụ trên thì độ dài chu kỳ được giới hạn bởi $m^k - 1 = 16^7 - 1 = 286.435.455$.

Việc dùng phương pháp tạo số ngẫu nhiên rất thích hợp cho mỗi chuỗi số ngẫu nhiên trong việc mô phỏng. Ở đây lưu ý rằng, việc chọn mầm giống (seed) đóng vai trò quan trọng. Phương pháp tạo được các số ngẫu nhiên tốt tức là chúng không là hàm của các mầm giống chọn không tốt.

TÀI LIỆU THAM KHẢO

- FRED SIMONDS. Network Security-Data and voice commucations MC.Graw-Hill Comp. New York.1995

- Németh Jozsef. Adatvédclem számitógépes és hirkozlo renszerekben. Budapest-1990.

- DIFFIE.W, HELLMAN.M.E. Privacy and Authentication. An introduction to Cryptography. IEEE. 1979.

- MERKLE.B.C. Protocols for public key Cryptosystems. IEEE 1980.

- D.W.DAVIES, W.L.PRICE. Security for Computer Networks John Wiley & Sons Ltd. New York. 1995

- SHANNON.C.E. Communication theory of secrecy systems.

- MATYAS S.M. Digital signatures. Computer Networks - 1979

- BLAKLEY.B and BLACKLEY G.R. Security of number theoretic public key cryptosystems againt random attack. Cryptologia. 1979. - WAYNE TOMASI. Electronic Communications Systems New Jersey - 1990

- THÁI HỒNG NHỊ - PHẠM MINH VIỆT - Hệ thống viễn thông - Tập 1 và tập 2 - NXB GD - Hà Nội - 2001.

- PHẠM MINH VIỆT - TRẦN CÔNG NHƯỢNG - Cơ sở kỹ thuật mạng Internet. NXB GD - Hà Nội - 2001.

- THÁI HỒNG NHỊ - Tài liệu giáo trình.
  An toàn thông tin
  Bảo vệ thông tin dữ liệu trên các hệ thống truyền tin
  ĐHAN - Hà Nội - 2000.

- Tài liệu Internet

MỤC LỤC

AN TOÀN THÔNG TIN

Mạng máy tính, truyền tin số và truyền dữ liệu

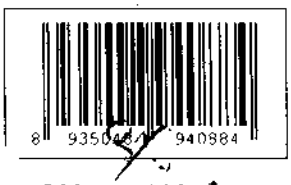
Tác giả: PGS. TS. THÁI HỒNG NHỊ
TS. PHẠM MINH VIỆT

Chịu trách nhiệm xuất bản: PGS. TS. TÔ ĐĂNG HẢI
Biên tập: NGUYỄN NGỌC
ĐỖ THỊ CẢNH
Vẽ bìa: LAN HƯƠNG
Chế bản: HỮU BÌNH

NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
70. TRẦN HƯNG ĐẠO - HÀ NỘI

In 1000 cuốn, khổ 19x27cm tại Nhà in **KHCN**
Giấy phép xuất bản số 469-89 do Cục XB cấp ngày 19/4/2004
In xong và nộp lưu chiểu tháng 5 năm 2004

204088



Giá 32.000 đ