MỤC LỤC

ĐỀ MỤC **TRANG**

MÔN HỌC AN TOÀN VÀ BẢO MẬT THÔNG TIN

Mã môn học: MH 25

Vị trí, tính chất, ý nghĩa và vai trò của môn học:

Vị trí của môn học: Môn học được bố trí sau khi sinh viên học xong mô đun: Mạng máy tính và Quản trị mạng 1.

Tính chất của môn học: Là môn học chuyên môn nghề

Mục tiêu của môn học:

- Trình bày được các nguy cơ đối với dữ liệu, các phương pháp đảm bảo an toàn dữ liệu.

- Ghi nhớ kiến thức về mật mã, mã hóa, và bảo mật dữ liệu (khái niệm, yêu cầu, chỉ dẫn, dịch vụ, kỹ thuật, thuật toán,...).

- Trình bày được quy trình khóa và chứng thực (khóa cơ sở dữ liệu / thư mục,chữ ký số, định danh,...).

- Trình bày chức năng an ninh mạng, trình bày được quy trình bảo mật thư điện tử và mã hóa thông điệp.

- Trình bày được những kiến thức về hệ thống thương mại điện tử (thanh toán tự động, đặt chỗ tự động, mô hình giao dịch mạng, bảo mật giao dịch điện tử...)

Nội dung của môn học:

CHƯƠNG 1
TỔNG QUAN VỀ AN TOÀN VÀ BẢO MẬT THÔNG TIN

Mục tiêu:

- Trình bày được nội dung tổng quan an toàn và bảo mật thông tin.
- Xác định được các mức bảo vệ hệ thống.
- Thực hiện các thao tác an toàn với máy tính bằng mật mã.

Nội dung chính:

1.1. Nội dung của an toàn và bảo mật thông tin

Mục tiêu: Trình bày được tổng quan an toàn và bảo mật thông tin.

Khi 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 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 có thể được quy tụ vào ba nhóm sau:

- Bảo vệ an toàn thông tin bằng các biện pháp hành chính.
- Bảo vệ an toàn thông tin bằng các biện pháp kỹ thuật (phần cứng).
- Bảo vệ an toàn thông tin bằng các biện pháp thuật toán (phần mềm).

Ba nhóm trên có thể được ứng dụng riêng rẽ hoặc phối kết hợp. Môi trường khó bảo vệ an toàn thông tin nhất và cũng là môi trường đối phương dễ xân nhập nhất đó là môi trường mạng và truyền tin. Biện pháp hiệu quả nhất và kinh tế nhất hiện nay trên mạng truyền tin và mạng máy tính là biện pháp thuật toán.

An toàn thông tin bao gồm các nội dung sau:

- Tính bí mật: tính kín đáo riêng tư của thông tin
- Tính xác thực của thông tin, bao gồm xác thực đối tác(bài toán nhận danh), xác thực thông tin trao đổi.
- Tính trách nhiệm: đảm bảo người gửi thông tin không thể thoái thác trách nhiệm về thông tin mà mình đã gửi.

Để đảm bảo an toàn thông tin dữ liệu trên đường truyền tin và trên mạng máy tính có hiệu quả thì điều trước tiên là phải lường trước hoặc dự đoán trước các khả năng không an toàn, khả năng xâm phạm, các sự cố rủi ro có thể xảy ra đối với thông tin dữ liệu được lưu trữ 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.

Có hai loại hành vi xâm phạm thông tin dữ liệu đó là: vi phạm chủ động và vi phạm thụ động. Vi phạm thụ động chỉ nhằm mục đích cuối cùng là nắm bắt được thông tin (đánh cắp thông tin). Việc làm đó có khi không biết được nội dung cụ thể nhưng có thể dò ra được người gửi, người nhận nhờ thông tin điều khiển giao thức chứa trong phần đầu các gói tin. Kẻ xâm nhập có thể kiểm tra được số lượng, độ dài và tần số trao đổi. Vì vậy vi pham thụ động không làm sai lệch hoặc hủy hoại nội dung thông tin dữ liệu được trao đổi. Vi phạm thụ động thường khó phát hiện nhưng có thể có những biện pháp ngăn chặn hiệu quả. Vi phạm chủ động là dạng vi phạm có thể làm thay đổi nội dung, xóa bỏ, làm trễ, xắp xếp lại thứ tự hoặc làm lặp lại 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. Vi phạm chủ động dễ phát hiện nhưng để ngăn chặn hiệu quả thì khó khăn hơn nhiều.

Một thực tế là không có một biện pháp bảo vệ an toàn thông tin dữ liệu nào là an toàn tuyệt đối. Một hệ thống dù được bảo vệ chắc chắn đến đâu cũng không thể đảm bảo là an toàn tuyệt đối.

1.2. Các chiến lược an toàn hệ thống

Mục tiêu: Trình bày được các chiến lược bảo vệ an toàn cho mạng.

1.2.1 Giới hạn quyền hạn tối thiểu (Last Privilege)

Đây là chiến lược cơ bản nhất theo nguyên tắc này bất kỳ một đối tượng nào cùng chỉ có những quyền hạn nhất định đối với tài nguyên mạng, khi thâm nhập vào mạng đối tượng đó chỉ được sử dụng một số tài nguyên nhất định.

1.2.2. Bảo vệ theo chiều sâu (Defence In Depth)

Nguyên tắc này nhắc nhở chúng ta : Không nên dựa vào một chế độ an toàn nào dù cho chúng rất mạnh, mà nên tạo nhiều cơ chế an toàn để tương hỗ lẫn nhau.

1.2.3. Nút thắt (Choke Point)

Tạo ra một "cửa khẩu" hẹp, và chỉ cho phép thông tin đi vào hệ thống của mình bằng con đường duy nhất chính là "cửa khẩu" này. => phải tổ chức một cơ cấu kiểm soát và điều khiển thông tin đi qua cửa này.

1.2.4. Điểm nối yếu nhất (Weakest Link)

Chiến lược này dựa trên nguyên tắc: " Một dây xích chỉ chắc tại mắt duy nhất, một bức tường chỉ cứng tại điểm yếu nhất"

Kẻ phá hoại thường tìm những chỗ yếu nhất của hệ thống để tấn công, do đó ta cần phải gia cố các yếu điểm của hệ thống. Thông thường chúng ta chỉ quan tâm đến kẻ tấn công trên mạng hơn là kẻ tiếp cận hệ thống, do đó an toàn vật lý được coi là yếu điểm nhất trong hệ thống của chúng ta.

1.2.5. Tính toàn cục

Các hệ thống an toàn đòi hỏi phải có tính toàn cục của các hệ thống cục bộ. Nếu có một kẻ nào đó có thể bẻ gãy một cơ chế an toàn thì chúng có thể thành công bằng cách tấn công hệ thống tự do của ai đó và sau đó tấn công hệ thống từ nội bộ bên trong.

1.2.6. Tính đa dạng bảo vệ

Cần phải sử dụng nhiều biện pháp bảo vệ khác nhau cho hệ thống khác nhau, nếu không có kẻ tấn công vào được một hệ thống thì chúng cũng dễ dàng tấn công vào các hệ thống khác.

1.3 Các mức bảo vệ trên mạng

Mục tiêu: Hiểu rõ và xác định được các mức bảo vệ hệ thống mạng.

Vì không thể có một giải pháp an toàn tuyệt đối nên người ta thường phải sử dụng đồng thời nhiều mức bảo vệ khác nhau tạo thành nhiều hàng rào chắn đối với các hoạt động xâm phạm. Việc bảo vệ thông tin trên mạng chủ yếu là bảo vệ thông tin cất giữ trong máy tính, đặc biệt là các server trên mạng. Bởi thế ngoài một số biện pháp nhằm chống thất thoát thông tin trên đường truyền mọi cố gắng tập trung vào việc xây dựng các mức rào chắn từ ngoài vào trong cho các hệ thống kết nối vào mạng. Thông thường bao gồm các mức bảo vệ sau:

1.3.1. Quyền truy nhập

Lớp bảo vệ trong cùng là quyền truy nhập nhằm kiểm soát các tài nguyên của mạng và quyền hạn trên tài nguyên đó. Dĩ nhiên là kiểm soát được các cấu trúc dữ liệu càng chi tiết càng tốt. Hiện tại việc kiểm soát thường ở mức tệp.

1.3.2. Đăng ký tên /mật khẩu.

Thực ra đây cũng là kiểm soát quyền truy nhập, nhưng không phải truy nhập ở mức thông tin mà ở mức hệ thống. Đây là phương pháp bảo vệ phổ biến nhất vì nó đơn giản ít phí tổn và cũng rất hiệu quả. Mỗi người sử dụng muốn được tham gia vào mạng để sử dụng tài nguyên đều phải có đăng ký tên và mật khẩu trước. Người quản trị mạng có trách nhiệm quản lý, kiểm soát mọi hoạt động của mạng và xác định quyền truy nhập của những người sử dụng khác theo thời gian và không gian (nghĩa là người sử dụng chỉ được truy nhập trong một khoảng thời gian nào đó tại một vị trí nhất định nào đó).

Về lý thuyết nếu mọi người đều giữ kín được mật khẩu và tên đăng ký của mình thì sẽ không xảy ra các truy nhập trái phép. Song điều đó khó đảm bảo trong thực tế vì nhiều nguyên nhân rất đời thường làm giảm hiệu quả của lớp bảo vệ này. Có thể khắc phục bằng cách người quản mạng chịu trách nhiệm đặt mật khẩu hoặc thay đổi mật khẩu theo thời gian.

1.3.3. Mã hoá dữ liệu

Để bảo mật thông tin trên đường truyền người ta sử dụng các phương pháp mã hoá. Dữ liệu bị biến đổi từ dạng nhận thức được sang dạng không nhận thức được theo một thuật toán nào đó và sẽ được biến đổi ngược lại ở trạm nhận (giải mã). Đây là lớp bảo vệ thông tin rất quan trọng.

1.3.4. Bảo vệ vật lý

Ngăn cản các truy nhập vật lý vào hệ thống. Thường dùng các biện pháp truyền thống như ngăn cấm tuyệt đối người không phận sự vào phòng đặt máy mạng, dùng ổ khoá trên máy tính hoặc các máy trạm không có ổ mềm.

1.3.5. Tường lửa

Ngăn chặn thâm nhập trái phép và lọc bỏ các gói tin không muốn gửi hoặc nhận vì các lý do nào đó để bảo vệ một máy tính hoặc cả mạng nội bộ (intranet)

1.3.6. Quản trị mạng

Trong thời đại phát triển của công nghệ thông tin, mạng máy tính quyết định toàn bộ hoạt động của một cơ quan, hay một công ty xí nghiệp. Vì vậy việc bảo đảm cho hệ thống mạng máy tính hoạt động một cách an toàn, không xảy ra sự cố là một công việc cấp thiết hàng đầu. Công tác quản trị mạng máy tính phải được thực hiện một cách khoa học đảm bảo các yêu cầu sau :

- Toàn bộ hệ thống hoạt động bình thường trong giờ làm việc.

- Có hệ thống dự phòng khi có sự cố về phần cứng hoặc phần mềm xảy ra.

- Backup dữ liệu quan trọng theo định kỳ.

- Bảo dưỡng mạng theo định kỳ.

- Bảo mật dữ liệu, phân quyền truy cập, tổ chức nhóm làm việc trên mạng.

1.4. An toàn thông tin bằng mật mã

Mục tiêu: Trình bày được cách bảo mật an toàn thông tin bằng mật mã.

Mật mã là một ngành khoa học chuyên nghiên cứu các phương pháp truyền tin bí mật. Mật mã bao gồm : Lập mã và phá mã. Lập mã bao gồm hai quá trình: mã hóa và giải mã.

Để bảo vệ thông tin trên đường truyền người ta thường biến đổi nó từ dạng nhận thức được sang dạng không nhận thức được trước khi truyền đi trên mạng, quá trình này được gọi là mã hoá thông tin (encryption), ở trạm nhận phải thực hiện quá trình ngược lại, tức là biến đổi thông tin từ dạng không nhận thức được (dữ liệu đã được mã hoá) về dạng nhận thức được (dạng gốc), quá trình này được gọi là giải mã. Đây là một lớp bảo vệ thông tin rất quan trọng và được sử dụng rộng rãi trong môi trường mạng.

Để bảo vệ thông tin bằng mật mã người ta thường tiếp cận theo hai hướng:

- Theo đường truyền (Link_Oriented_Security).

- Từ nút đến nút (End_to_End).

Theo cách thứ nhất thông tin được mã hoá để bảo vệ trên đường truyền giữa hai nút mà không quan tâm đến nguồn và đích của thông tin đó. Ở đây ta lưu ý rằng thông tin chỉ được bảo vệ trên đường truyền, tức là ở mỗi nút đều có quá trình giải mã sau đó mã hoá để truyền đi tiếp, do đó các nút cần phải được bảo vệ tốt.

Ngược lại theo cách thứ hai thông tin trên mạng được bảo vệ trên toàn đường truyền từ nguồn đến đích. Thông tin sẽ được mã hoá ngay sau khi mới tạo ra và chỉđược giải mã khi về đến đích. Cách này mắc phải nhược điểm là chỉ có dữ liệu của người ung thì mới có thể mã hóa được còn dữ liệu điều khiển thì giữ nguyên để có thể xử lý tại các nút.

1.5. Vai trò của hệ mật mã

Mục tiêu: phân tích được vai trò của hệ mật mã.

Các hệ mật mã phải thực hiện được các vai trò sau:

- Hệ mật mã phải che dấu được nội dung của văn bản rõ (PlainText) để đảm bảo sao cho chỉ người chủ hợp pháp của thông tin mới có quyền truy cập thông tin (Secrety), hay nói cách khác là chống truy nhập không đúng quyền hạn.

- Tạo các yếu tố xác thực thông tin, đảm bảo thông tin lưu hành trong hệ thống đến người nhận hợp pháp là xác thực (Authenticity).

- Tổ chức các sơ đồ chữ ký điện tử, đảm bảo không có hiện tượng giả mạo, mạo danh để gửi thông tin trên mạng.

Ưu điểm lớn nhất của bất kỳ hệ mật mã nào đó là có thể đánh giá được độ phức tạp tính toán mà "kẻ địch" phải giải quyết bài toán để có thể lấy được thông tin của dữ liệu đã được mã hoá. Tuy nhiên mỗi hệ mật mã có

một số ưu và nhược điểm khác nhau, nhưng nhờ đánh giá được độ phức tạp tính toán mà ta có thể áp dụng các thuật toán mã hoá khác nhau cho từng ứng dụng cụ thể tuỳ theo độ yêu cầu về độ an toàn.

Các thành phần của một hệ mật mã :

Định nghĩa: một hệ mật là một bộ 5 (P,C,K,E,D) thoả mãn các điều kiện sau:

- P là một tập hợp hữu hạn các bản rõ (PlainText), nó được gọi là không gian bản rõ.

- C là tập các hữu hạn các bản mã (Crypto), nó còn được gọi là không gian các bản mã. Mỗi phần tử của C có thể nhận được bằng cách áp dụng phép mã hoá Ek lên một phần tử của P, với k K.

- K là tập hữu hạn các khoá hay còn gọi là không gian khoá. Đối với mỗi

phần tử k của K được gọi là một khoá (Key). Số lượng của không gian khoá phải đủ lớn để "kẻ địch" không có đủ thời gian để thử mọi khoá có thể (phương pháp vét cạn).

- Đối với mỗi k K có một quy tắc mã eK: P → C và một quy tắc giải

mã tương ứng d_k D. Mỗi eK: P → C và d_k: C → P là những hàm mà:

dK (e_k(x))=x với mọi bản rõ x P.

1.6. Phân loại hệ mật mã

Mục tiêu: Biết phân loại các hệ mật mã khác nhau, so sánh được điểm ưu, nhược của từng hệ mật mã.

Có nhiều cách để phân loại hệ mật mã. Dựa vào cách truyền khóa có thể phân các hệ mật mã thành hai loại:

- Hệ mật đối xứng (hay còn gọi là mật mã khóa bí mật): là những hệ mật dung chung một khoá cả trong quá trình mã hoá dữ liệu và giải mã dữ liệu.

Do đó khoá phải được giữ bí mật tuyệt đối.

- Hệ mật mã bất đối xứng (hay còn gọi là mật mã khóa công khai) : Hay còn gọi là hệ mật mã công khai, các hệ mật này dùng một khoá để mã hoá sau đó dùng một khoá khác để giải mã, nghĩa là khoá để mã hoá và giải mã là khác nhau. Các khoá này tạo nên từng cặp chuyển đổi ngược nhau và không có khoá nào có thể suy được từ khoá kia. Khoá dùng để mã hoá có thể công khai nhưng khoá dùng để giải mã phải giữ bí mật.

Ngoài ra nếu dựa vào thời gian đưa ra hệ mật mã ta còn có thể phân làm hai loại: Mật mã cổ điển (là hệ mật mã ra đời trước năm 1970) và mật mã hiện đại (ra đời sau năm 1970). Còn nếu dựa vào cách thức tiến hành mã

thì hệ mật mã còn được chia làm hai loại là mã dòng (tiến hành mã từng khối dữ liệu, mỗi khối lại dựa vào các khóa khác nhau, các khóa này được sinh ra từ hàm sinh khóa, được gọi là dòng khóa) và mã khối (tiến hành mã từng khối dữ liệu với khóa như nhau)

1.7. Tiêu chuẩn đánh giá hệ mật mã

Mục tiêu: đánh giá được một hệ mật mã người ta thường đánh giá thông qua các tính chất như độ an toà, tốc độ giải mã, cách phân phối khóa.

1.7.1. Độ an toàn

Một hệ mật được đưa vào sử dụng điều đầu tiên phải có độ an toàn cao. Ưu điểm của mật mã là có thể đánh giá được độ an toàn thông qua độ an toàn tính toán mà không cần phải cài đặt. Một hệ mật được coi là an toàn nếu để phá hệ mật mã này phải dùng n phép toán. Mà để giải quyết n phép toán cần thời gian vô cùng lớn, không thể chấp nhận được.

Một hệ mật mã được gọi là tốt thì nó cần phải đảm bảo các tiêu chuẩn sau:

- Chúng phải có phương pháp bảo vệ mà chỉ dựa trên sự bí mật của các khoá, công khai thuật toán.

- Khi cho khoá công khai e_k và bản rõ P thì chúng ta dễ dàng tính được $e_k(P) = C$. Ngược lại khi cho d_k và bản mã C thì dễ dàng tính được $d_k(M)=P$.

Khi không biết dK thì không có khả năng để tìm được M từ C, nghĩa là khi cho hàm f: $X \rightarrow Y$ thì việc tính y=f(x) với mọi x X là dễ còn việc tìm x khi biết y lại là vấn đề khó và nó được gọi là hàm một chiều.

- Bản mã C không được có các đặc điểm gây chú ý, nghi ngờ.

1.7.2. Tốc độ mã và giải mã

Khi đánh giá hệ mật mã chúng ta phải chú ý đến tốc độ mã và giải mã. Hệ mật tốt thì thời gian mã và giải mã nhanh.

1.7.3. Phân phối khóa

Một hệ mật mã phụ thuộc vào khóa, khóa này được truyền công khai hay truyền khóa bí mật. Phân phối khóa bí mật thì chi phí sẽ cao hơn so với các hệ mật có khóa công khai. Vì vậy đây cũng là một tiêu chí khi lựa chọn hệ mật mã.

CHƯƠNG 2
CÁC PHƯƠNG PHÁP MÃ HÓA CỔ ĐIỂN

Mục tiêu:

- Trình bày được PKI, chữ ký số, chứng chỉ số, CA, CRL;
- Xây dựng một PKI trên ứng dụng cụ thể ;
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

2.1. Các hệ mật mã cổ điển

Mục tiêu: Trình bày được các hệ mật mã cổ điển.

2.1.1. Mã dịch vòng (shift cipher)

Phần này sẽ mô tả mã dịch (MD) dựa trên số học theo modulo. Trước tiên

sẽ điểm qua một số định nghĩa cơ bản của số học này.

Các định nghĩa

Giả sử a và b là các số nguyên và m là một số nguyên dương. Khi đó ta viết $a \equiv b \pmod m$ nếu m chia hết cho b-a. Mệnh đề $a \equiv b \pmod m$ được gọi là " a đồng dư với b theo modulo m". Số nguyên m được gọi là mudulus.

Giả sử chia a và b cho m và ta thu được phần thương nguyên và phần dư, các phần dư nằm giữa 0 và m-1, nghĩa là $a = q_1m + r_1$ và $b = q_2m + r_2$ trong đó $0 \le r_1 \le m-1$ và $0 \le r_2 \le m-1$. Khi đó có thể dễ dàng thấy rằng $a \equiv b \pmod m$ khi và chỉ khi $r_1 = r_2$. Ta sẽ dùng ký hiệu a mod m (không dùng các dấu ngoặc) để xác định phần dư khi a được chia cho m (chính là giá trị r1ở trên). Như vậy: $a \equiv b \pmod m$ khi và chỉ khi a mod m = b mod m. Nếu thay a bằng a mod m thì ta nói rằng a được rút gọn theo modulo m.

Nhận xét: Nhiều ngôn ngữ lập trình của máy tính xác định a mod m là phần dư trong dải - m+1,.. ., m-1 có cùng dấu với a. Ví dụ -18 mod 7 sẽ là -4, giá trị này khác với giá trị 3 là giá trị được xác định theo công thức trên. Tuy nhiên, để thuận tiện ta sẽ xác định a mod m luôn là một số không âm.

Bây giờ ta có thể định nghĩa số học modulo m: Z_m được coi là tập hợp {0,1,. . .,m-1} có trang bị hai phép toán cộng và nhân. Việc cộng và nhân trong Z_m được thực hiện giống như cộng và nhân các số thực ngoài trừ một điểm là các kết quả được rút gọn theo modulo m.

Ví dụ tính 11×13 trong Z_{16}. Tương tự như với các số nguyên ta có $11 \times 13 = 143$. Để rút gọn 143 theo modulo 16, ta thực hiện phép chia bình thường:

$143 = 8 \times 16 + 15$, bởi vậy $143 \bmod 16 = 15$ trong Z_{16}.

Giả sử $P = C = K = Z_{26}$ với $0 \le k \le 25$, định nghĩa:

$E_k(x) = x + K \bmod 26$

và (x,y Z_{26})

Nhận xét: Trong trường hợp K = 3, hệ mật thường được gọi là mã Caesar đã từng được Julius Caesar sử dụng.

Ta sẽ sử dụng MDV (với modulo 26) để mã hoá một văn bản tiếng Anh thông thường bằng cách thiết lập sự tương ứng giữa các kí tự và các thặng dư theo modulo 26 như sau: A ↔ 0,B ↔ 1, . . ., Z ↔ 25. Vì phép tương ứng này còn dùng trong một vài ví dụ nên ta sẽ ghi lại để còn tiện dùng sau này:

Ví dụ:

Cho bản mã

JBCRCLQRWCRVNBJENBWRWN

ta sẽ thử liên tiếp các khoá giải mã d0 ,d1 .. . và y thu được:

j b c r c l q r w c r v n b j e n b w r w n

i a b q b k p q v b q u m a i d m a v q v m

h z a p a j o p u a p t l z h c l z u p u l

g y z o z i n o t z o s k y g b k y t o t k

j x y n y h m n s y n r j e x f a j x s n s j

e w x m x g l m r x m q i w e z i w r m r i

d v w l w f k l q w l p h v o d y h v q l q h

c u v k v e j k p v k o g u c x g u p k p g

b t u j u d i j o u j n f t b w f o j o f

a s t i t c h i n t i m e s a v e s n i n e

Tới đây ta đã xác định được bản rõ và dừng lại. Khoá tương ứng K = 9. Trung bình có thể tính được bản rõ sau khi thử 26/2 = 13 quy tắc giải mã. Như đã chỉ ra trong ví dụ trên, điều kiện để một hệ mật an toàn là phép tìm khoá vét cạn phải không thể thực hiện được, tức không gian khoá phải rất lớn.

Tuy nhiên, một không gian khoá lớn vẫn chưa đủ đảm bảo độ mật.

2.1.2. Mã thay thế

Một hệ mật nổi tiếng khác là hệ mã thay thế. Hệ mật này đã được sử dụng hàng trăm năm. Trò chơi đố chữ "cryptogram" trong các bài báo là những ví dụ về MTT.

Trên thực tế MTT có thể lấy cả P và C đều là bộ chữ cái tiếng anh, gồm 26 chữ cái. Ta dùng Z26 trong MDV vì các phép mã và giải mã đều là các phép toán đại số. Tuy nhiên, trong MTT, thích hợp hơn là xem phép mã và giải mã như các hoán vị của các kí tự.

Mã thay thế

Cho P =C = Z_{26} . K chứa mọi hoán vị có thể của 26 kí hiệu 0,1, . . . , 25

Với mỗi phép hoán vị π K , ta định nghĩa:

eπ(x) = π(x)

và

dπ(y) = π -1(y)

trong đó π -1 là hoán vị ngược của π.

Sau đây là một ví dụ về phép hoán vị ngẫu nhiên π tạo nên một hàm mã hoá (cũng như trước, các ký hiệu của bản rõ được viết bằng chữ thường còn các ký hiệu của bản mã là chữ in hoa).

Như vậy, eπ (a) = X, eπ (b) = N,. . . . Hàm giải mã là phép hoán vị ngược. Điều này được thực hiện bằng cách viết hàng thứ hai lên trước rồi sắp xếp theo thứ tự chữ cái. Ta nhận được:

Bởi vậy dπ (A) = d, dπ(B) = 1, . . .

Ví dụ: Hãy giải mã bản mã:

M G Z V Y Z L G H C M H J M Y X S S E M N H A H Y C D L M H A.

Mỗi khoá của MTT là một phép hoán vị của 26 kí tự. Số các hoán vị này là 26!, lớn hơn 4 ×1026 là một số rất lớn. Bởi vậy, phép tìm khoá vét cạn không thể thực hiện được, thậm chí bằng máy tính. Tuy nhiên, sau này sẽ thấy rằng MTT có thể dễ dàng bị thám bằng các phương pháp khác.

2.1.3. Mã Affine

MDV là một trường hợpđặc biệt của MTT chỉ gồm 26 trong số 26! Các hoán vị có thể của 26 phần tử. Một trường hợp đặc biệt khác của MTT là mã Affine được mô tả dưới đây. Trong mã Affine, ta giới hạn chỉ xét các hàm mã có dạng:

e(x) = ax + b mod 26

a, b Z_{26} . Các hàm này được gọi là các hàm Affine (chú ý rằng khi a = 1, ta có MDV).

Để việc giải mã có thể thực hiện được, yêu cầu cần thiết là hàm Affine phải là đơn ánh. Nói cách khác, với bất kỳ y ∈ Z26, ta muốn có đồng nhất thức sau:

$ax + b \equiv y \pmod{26}$

phải có nghiệm x duy nhất. Đồng dư thức này tương đương với:

$ax \equiv y-b \pmod{26}$

Vì y thay đổi trên Z_{26} nên y-b cũng thay đổi trên Z_{26} . Bởi vậy, ta chỉ cần nghiên cứu phương trình đồng dư:

$ax \equiv y \pmod{26}$ (y Z_{26}).

Ta biết rằng, phương trình này có một nghiệm duy nhất đối với mỗi y khi và chỉ khi UCLN(a,26) = 1 (ở đây hàm UCLN là ước chung lớn nhất của các biến của nó). Trước tiên ta giả sử rằng, UCLN(a,26) = d >1. Khi đó, đồng dư thức $ax \equiv 0 \pmod{26}$ sẽ có ít nhất hai nghiệm phân biệt trong Z_{26} là x = 0 và x = 26/d. Trong trường hợp này, e(x) = ax + b mod 26 không phải là một hàm đơn ánh và bởi vậy nó không thể là hàm mã hoá hợp lệ.

Ví dụ, do UCLN(4,26) = 2 nên 4x +7 không là hàm mã hoá hợp lệ: x và

x+13 sẽ mã hoá thành cùng một giá trị đối với bất kì x Z_{26} .

Ta giả thiết UCLN(a,26) = 1. Giả sử với x1 và x2 nào đó thảo mãn:

$ax_1 \equiv ax_2 \pmod{26}$

Khi đó

$a(x_1- x_2) \equiv 0 \pmod{26}$

bởi vậy

$26 \mid a(x_1- x_2)$

Bây giờ ta sẽ sử dụng một tính chất của phép chia sau: Nếu UCLN(a,b)=1

và a bc thì a c. Vì 26 a(x1- x2) và UCLN(a,26) = 1 nên ta có:

26 (x1- x2)

tức là

$x1 \equiv x2 \pmod{26}$

Tới đây ta chứng tỏ rằng, nếu UCLN(a,26) = 1 thì một đồng dư thức dạng $ax \equiv y \pmod{26}$ chỉ có (nhiều nhất) một nghiệm trong Z26 . Do đó, nếu ta cho x thay đổi trên Z26 thì ax mod 26 sẽ nhận được 26 giá trị khác nhau

theo modulo 26 và đồng dư thức ax ≡ y (mod 26) chỉ có một nghiệm y duy nhất.

Không có gì đặc biệt đối với số 26 trong khẳng định này. Bởi vậy, bằng cách tương tự ta có thể chứng minh được kết quả sau:

** Định lí*

Đồng dư thức ax ≡ b mod m chỉ có một nghiệm duy nhất x Z_m với mọi b Z_m khi và chỉ khi UCLN(a,m) = 1.

Vì 26 = 2 ×13 nên các giá trị a Z_{26} thoả mãn UCLN(a,26) = 1 là a = 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23 và 25. Tham số b có thể là một phần tử bất kỳ trong Z_{26}. Như vậy, mã Affine có 12 × 26 = 312 khoá có thể (dĩ nhiên con số này quá nhỉ để bảo đảm an toàn).

Bây giờ ta sẽ xét bài toán chung với modulo m. Ta cần một định nghĩa khác trong lý thuyết số.

Định nghĩa

Giả sử a ≥ 1 và m ≥ 2 là các số nguyên. UCLN(a,m) = 1 thì ta nói rằng a và m là nguyên tố cùng nhau. Số các số nguyên trong Z_m nguyên tố cùng nhau với m thường được ký hiệu là φ(m) (hàm này được gọi là hàm Euler).

Một kết quả quan trọng trong lý thuyết số cho ta giá trị của φ(m) theo các thừa số trong phép phân tích theo luỹ thừa các số nguyên tố của m. (Một số nguyên p >1 là số nguyên tố nếu nó không có ước dương nào khác ngoài 1 và p.

Mọi số nguyên m >1 có thể phân tích được thành tích của các luỹ thừa các số nguyên tố theo cách duy nhất. Ví dụ 60 = 23× 3 × 5 và 98 = 2 × 72).

Số khoá trong mã Affine trên Zm bằng φ(m), trong đó φ(m) được cho theo công thức trên. (Số các phép chọn của b là m và số các phép chọn của a là φ(m) với hàm mã hoá là e(x) = ax + b). Ví dụ, khi m = 60, φ(60)=φ(5.22.3)=φ(5). φ(22). φ(3) = 2 × 2 × 4 = 16 và số các khoá trong mã Affine là 960.

Bây giờ ta sẽ xét xem các phép toán giải mã trong mật mã Affine với modulo m = 26. Giả sử UCLN(a,26) = 1. Để giải mã cần giải phương trình đồng dư y ≡ax+b (mod 26) theo x. Từ thảo luận trên thấy rằng, phương trình này có một nghiệm duy nhất trong Z_{26} . Tuy nhiên ta vẫn chưa biết một phương pháp hữu hiệu để tìm nghiệm. Điều cần thiết ở đây là có một thuật toán hữu hiệu để làm việc đó. Rất may là một số kết quả tiếp sau về số học modulo sẽ cung cấp một thuật toán giải mã hữu hiệu cần tìm.

Các định nghĩa trên phép cộng và phép nhân Z_m thảo mãn hầu hết các quy tắc quen thuộc trong số học. Sau đây ta sẽ liệt kê mà không chứng minh các tính chất này:

1. Phép cộng là đóng, tức với bất kì a,b Z_m ,a +b Z_m

2. Phép cộng là giao hoán, tức là với a,b bất kì Z_m a+b = b+a

3. Phép cộng là kết hợp, tức là với bất kì a,b,c Z_m (a+b)+c = a+(b+c)

4. 0 là phần tử đơn vị của phép cộng, có nghĩa là với a bất kì Z_m

a+0 = 0+a = a

5. Phần tử nghịch đảo của phép cộng của phần tử bất kì (a Z_m) là m-a, nghĩa là a+(m-a) = (m-a)+a = 0 với bất kì a Z_m .

6. Phép nhân là đóng , tức là với a,b bất kì Zm , ab Zm .

7. Phép nhân là giao hoán , nghĩa là với a,b bất kì Zm , ab = ba

8. Phép nhân là kết hợp, nghĩa là với a,b,c Zm , (ab)c = a(cb)

9. 1 là phần tử đơn vị của phép nhân, tức là với bất kỳ a Zm

a×1 = 1×a = a

10. Phép nhân có tính chất phân phối đối với phép cộng, tức là đối với

a,b,c Zm, (a+b)c = (ac)+(bc) và a(b+c) = (ab) + (ac)

Các tính chất 1,3-5 nói lên rằng Zm lập nên một cấu trúc đại số được gọi là một nhóm theo phép cộng. Vì có thêm tính chất 4 nhóm được gọi là nhóm Aben (hay nhóm giao hoán).

Các tính chất 1-10 sẽ thiết lập nên một vành Zm . Một số ví dụ quen thuộc của vành là các số nguyên Z, các số thực R và các số phức C. Tuy nhiên các vành này đều vô hạn, còn mối quan tâm của chúng ta chỉ giới hạn trên các vành hữu hạn.

Vì phần tử ngược của phép cộng tồn tại trong Zm nên cũng có thể trừ các phần tử trong Zm . Ta định nghĩa a-b trong Zm là a+m-b mod m. Một cách tương tự có thể tính số nguyên a-b rồi rút gon theo modulo m.

Ví dụ : Để tính 11-18 trong Z31, ta tính 11+31 – 18 mod 31= 11+13 mod 31= 24. Ngược lại, có thể lấy 11-18 được -7 rồi sau đó tính -7 mod 31 =31-7= 24.

Mã dịch vòng được xác định trên Z26 (do có 26 chữ cái trên bảng chữ cái tiếng Anh) mặc dù có thể xác định nó trên Zm với modulus m tuỳ ý. Dễ dàng thấy rằng, MDV sẽ tạo nên một hệ mật như đã xác định ở trên, tức là dK(eK(x)) = x với mọi x Z26 . Ta có sơ đồ mã như sau:

2.1.4. Mã Vigenère

Trong cả hai hệ MDV và MTT (một khi khoá đã được chọn) mỗi ký tự sẽ được ánh xạ vào một ký tự duy nhất. Vì lý do đó, các hệ mật còn được gọi hệ thay thế đơn biểu. Bây giờ ta sẽ trình bày một hệ mật không phải là bộ chữ đơn, đó là hệ mã Vigenère nổi tiếng. Mật mã này lấy tên của Blaise de Vigenère sống vào thế kỷ XVI.

Sử dụng phép tương ứng A 0, B 1, . . . , Z 25 mô tả ở trên, ta có thể gắn cho mỗi khóa K với một chuỗi kí tự có độ dài m được gọi là từ khoá.

Mật mã Vigenère sẽ mã hoá đồng thời m kí tự: Mỗi phần tử của bản rõ tương đương với m ký tự.

2.1.5. Mật mã Hill

Trong phần này sẽ mô tả một hệ mật thay thế đa biểu khác được gọi là mật mã Hill. Mật mã này do Lester S.Hill đưa ra năm 1929. Giả sử m là một số nguyên dương, đặt P = C = (Z26)m . Ý tưởng ở đây là lấy m tổ hợp tuyến tính của m ký tự trong một phần tử của bản rõ để tạo ra m ký tự ở một phần tử của bản mã.

2.2. Mã thám các hệ mã cổ điển

Trong phần này ta sẽ bàn tới một vài kỹ thuật mã thám. Giả thiết chung ở đây là luôn coi đối phương Oscar đã biết hệ mật đang dùng. Giả thiết này được gọi là nguyên lý Kerekhoff. Dĩ nhiên, nếu Oscar không biết hệ mật được dùng thì nhiệm vụ của anh ta sẽ khó khăn hơn. Tuy nhiên ta không muốn độ mật của một hệ mật lại dựa trên một giả thiết không chắc chắn là Oscar không biết hệ mật được sử dụng. Do đó, mục tiêu trong thiết kế một hệ mật là phải đạt được độ mật dưới giả thiết Kerekhoff.

Trước tiên ta phân biệt các mức độ tấn công khác nhau vào các hệ mật. Sau đây là một số loại thông dụng nhất.

Chỉ có bản mã:

Thám mã chỉ có xâu bản mã y.

Bản rõ đã biết:

Thám mã có xâu bản rõ x và xâu bản mã tương ứng y.

Bản rõ được lựa chọn:

Thám mã đã nhận được quyền truy nhập tạm thời vào cơ chế mã hoá. Bởi vậy, thám mã có thể chọn một xâu bản rõ x và tạo nên xâu bản mã y tương ứng.

Bản mã được lựa chọn:

Thám mã có được quyền truy nhập tạm thời vào cơ chế giải mã. Bởi vậy thám mã có thể chọn một bản mã y và tạo nên xâu bản rõ x tương ứng.

Trong mỗi trường hợp trên, đối tượng cần phải xác định chính là khoá đã sử dụng. Rõ ràng là 4 mức tấn công trên đã được liệt kê theo độ tăng của sức mạnh tấn công. Nhận thấy rằng, tấn công theo bản mã được lựa chọn là thích hợp với các hệ mật khoá công khai mà ta sẽ nói tới ở chương sau.

Trước tiên, ta sẽ xem xét cách tấn công yếu nhất, đó là tấn công chỉ có bản mã. Giả sử rằng, xâu bản rõ là một văn bản tiếng Anh thông thường không có chấm câu hoặc khoảng trống (mã thám sẽ khó khăn hơn nếu mã cả dấu chấm câu và khoảng trống).

Có nhiều kỹ thuật thám mã sử dụng các tính chất thống kê của ngôn ngữ

tiếng Anh. Nhiều tác giả đã ước lượng tần số tương đối của 26 chữ cái theo các tính toán thống kê từ nhiều tiểu thuyết, tạp chí và báo. Các ước lượng trong bảng dưới đây lấy theo tài liệu của Beker và Piper.

Kí tự	Xác suất	Kí tự	Xác suất	Kí tự	Xác suất
A	.082	J	.002	S	.063
B	.015	K	.008	T	.091
C	.028	L	.040	U	.028
D	.043	M	.024	V	.010
E	.0127	N	.067	W	.023
F	.022	O	.075	X	.001
G	.020	P	.019	Y	.020
H	.061	Q	.001	Z	.001
I	.070	R	.060		

Từ bảng trên, Beker và Piper phân 26 chữ cái thành 5 nhóm như sau:

1. E: có xác suất khoảng 1,120
2. T, A, O, I, N, S, H, R : mỗi ký tự có xac suất khoảng 0,06 đến 0,09
3. D, L : mỗi ký tự có xác suất chừng 0,04

4. C, U, M, W, F, G, Y, P, B: mỗi ký tự có xác suất khoảng 0,015 đến

0,023

5. V, K, J, X, Q, Z mỗi ký tự có xác suất nhỏ hơn 0,01

Việc xem xét các dãy gồm 2 hoặc 3 ký tự liên tiếp (được gọi là bộ đôi-

diagrams và bộ ba – Trigrams) cũng rất hữu ích. 30 bộ đôi thông dụng nhất (theo

thứ tự giảm dần) là: TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI và OF. 12 bộ ba thông dụng nhất (theo thứ tự giảm dần) là: THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR và DTH.

2.2.1. Thám hệ mã Affine

Mật mã Affine là một ví dụ đơn giản cho ta thấy cách thám hệ mã nhờ dùng các số liệu thống kê. Giả sử Oscar đã thu trộm được bản mã sau:

Kí tự	Tần suất	Kí tự	Tần suất	Kí tự	Tần suất	Kí tự	Tần suất
A	2	H	5	O	1	U	2
B	1	I	0	P	3	V	4
C	0	J	0	Q	0	W	0
D	6	K	5	R	8	X	2
E	5	L	2	S	3	Y	1
F	4	M	2	T	0	Z	0
G	0	N	1				

Bản mã nhận được từ mã Affine:

FMXVEDRAPHFERBNDKRXRSREFMORUDSDKDVSHVUFEDKPK

DLYEVLRHHRH

Phân tích tần suất của bản mã này được cho ở bảng dưới Bản mã chỉ có 57 ký tự. Tuy nhiên độ dài này cũng đủ phân tích thám mã đối với hệ

Affine. Các ký tự có tần suất cao nhất trong bản mã là: R (8 lần xuất hiện), D (6 lần xuất hiện), E, H, K (mỗi ký tự 5 lần) và F, S, V (mỗi ký tự 4 lần).

Trong phỏng đoán ban đầu, ta giả thiết rằng R là ký tự mã của chữ e và D là kí tự mã của t, vì e và t tương ứng là 2 chữ cái thông dụng nhất. Biểu thị bằng số ta có: eK(4) = 17 và eK(19) = 3. Nhớ lại rằng eK(x) = ax +b trong đó a và b là các số chưa biết. Bởi vậy ta có hai phương trình tuyến tính hai ẩn:

4a +b = 17

19a + b = 3

Hệ này có duy nhất nghiệm a = 6 và b = 19 (trong Z26). Tuy nhiên đây là một khoá không hợp lệ do UCLN(a,26) = 2 1. Bởi vậy giả thiết của ta là khôngđúng. Phỏng đoán tiếp theo của ta là: R là ký tự mã của e và E là mã của t. Thực hiện như trên, ta thu được a =13 và đây cũng là một khoá không hợp lệ. Bởi vậy ta phải thử một lần nữa: ta coi rằng R là mã hoá của e và H là mã hoá của t. Điều này dẫn tới a = 8 và đây cũng là một khoá không hợp lệ. Tiếp tục, giả sử rằng R là mã hoá của e và K là mã hoá của t. Theo giả thiết này ta thu được a = 3 và b = 5 là khóa hợp lệ.

Ta sẽ tính toán hàm giải mã ứng với K = (3,5) và giải mã bản mã để xem liệu có nhận được xâu tiếng Anh có nghĩa hay không. Điều này sẽ khẳng định tính hợp lệ của khoá (3,5). auk hi thực hiện các phép toán này, ta có dK (y) = 9y – 19 và giải mã bản mã đã cho, ta được:

algorithmsarequitegeneraldefinitionsof

arithmeticprocesses

Như vậy khoá xác định trên là khoá đúng.

2.2.2. Thám hệ mã thay thế

Sau đây ta phân tích một tình huống phức tạp hơn, đó là thay thế bản n

Ký tự	Tần suất	Ký tự	Tần suất	Ký tự	Tần suất	Ký tự	Tần suất
A	0	H	4	O	0	U	5
B	1	I	5	P	1	V	5
C	15	J	11	Q	4	W	8
D	13	K	1	R	10	X	6
E	7	L	0	S	3	Y	10
F	11	M	16	T	2	Z	20
G	1	N	9				

Do Z xuất hiện nhiều hơn nhiều so với bất kỳ một ký tự nào khác trong bản mã nên có thể phỏng đoán rằng, dZ(Z) = e. các ký tự còn lại xuất hiện ít nhất 10 lần (mỗi ký tự) là C, D, F, J, R, M, Y. Ta hy vọng rằng, các ký tự này là mã khoá của (một tập con trong) t, a, c, o, i, n, s, h, r, tuy nhiên sự khác biệt về tần suất không đủ cho ta có được sự phỏng đoán thích hợp.

Tới lúc này ta phải xem xét các bộ đôi, đặc biệt là các bộ đôi có dạng -Z hoặc Z- do ta đã giả sử rằng Z sẽ giải mã thành e. Nhận thấy rằng các bộ đôi thường gặp nhất ở dạng này là DZ và ZW (4 lần mỗi bộ); NZ và ZU (3 lần mỗi bộ); và RZ, HZ, XZ, FZ, ZR, ZV, ZC, ZD và ZJ (2 lần mỗi bộ). Vì ZW xuất hiện 4 lần còn WZ không xuất hiện lần nào và nói chung W xuất hiện ít hơn so với nhiều ký tự khác, nên ta có thể phỏng đoán là dK(W) = d. Vì DZ xuất hiện 4 lần và ZD xuất hiện 2 lần nên ta có thể nghĩ rằng dK(D) $\in$ {r,s,t}, tuy nhiên vẫn còn chưa rõ là ký tự nào trong 3 ký tự này là ký tự đúng.

Nêu tiến hành theo giả thiết dK(Z) = e và dK(W) = d thì ta phải nhìn trở lại bản mã và thấy rằng cả hai bộ ba ZRW và RZW xuất hiện ở gần đầu của bản mã và RW xuất hiện lại sau đó vì R thường xuất hiện trong bản mã và nd là một bộ đôi thường gặp nên ta nên thử dK(R) = n xem là một khả năng thích hợp nhất.

Tới lúc này ta có:

- - - - - - end - - - - - - - - - e - - - - ned- - - e - - - - - - - - -

YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

- - - - - - - - e- - - - e - - - - - - - - n - - d - - - en - - - - e - - - -e

NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

- e - - - n - - - - - n - - - - - - ed - - - e - - - - - - ne - nd- e- e - -

NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

- ed - - - - - n - - - - - - - - - - e - - - ed - - - - - - - d - - - e - - n

XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR

Bước tiếp theo là thử dK(N) = h vì NZ là một bộ đôi thường gặp còn ZN không xuất hiện. Nếu điều này đúng thì đoạn sau của bản rõ ne - ndhe sẽ gợi ý rằng dK(C) = a. Kết hợp các giả định này, ta có:

- - - - - -end- - - - - a- - -e -a - - nedh- -e- - - - - -a - - - - -

YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

h - - - - - - - a- - - e - a- - - a - - - nhad - a - -en -a - e - h- -e

NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

he - a - n- - - - - - n - - - - - - ed - - - e- - - e - - neandhe -e - -

NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

- ed - a - - -nh - - - ha - - - a- e - - - - ed - - - - -a -d - - he- -n

XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR

Bây giờ ta xét tới M là ký tự thường gặp nhất sau Z. Đoạn bản mã RNM mà ta tin là sẽ giải mã thành nh- gợi ý rằng h- sẽ bắt đầu một từ, bởi vậy chắc là M sẽ biểu thị một nguyên âm. Ta đã sử dụng a và e, bởi vậy, phỏng đoán rằng dK(M) = i hoặc o. Vì ai là bộ đôi thường gặp hơn ao nên bộ đôi CM trong bản mã gợi ý rằng, trước tiên nên thử dK(M) = i. Khi đó ta có:

- - - - -iend- - - - - a -i - e -a -inedhi - e- - - - - -a - - -i -

YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

h - - - - - i - ea - i - e -a - - -a - i -nhad -a - en - -a - e -hi -e

NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

he - a - n - - - - -in -i - - - - ed - - -e - - - e - ineandhe - e - -

NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

- ed - a - - inhi - - hai - - a - e - i- -ed- - - - - a - d - - he - -n

XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR

Tiếp theo thử xác định xem chữ nào được mã hoá thành o. Vì o là một chữ thường gặp nên giả định rằng chữ cái tương ứng trong bản mã là một trong các ký tự D,F,J,Y. Y có vẻ thích hợp nhất, nếu không ta sẽ có các xâu dài các nguyên âm, chủ yếu là aoi (từ CFM hoặc CJM). Bởi vậy giả thiết rằng dK(Y) =o.

Ba ký tự thường gặp nhất còn lại trong bản mã là D,F,J, ta phán đoán sẽ giải mã thành r,s,t theo thứ tự nào đó. Hai lần xuất hiện của bộ ba NMD gợi ý rằng dK(D) = s ứng với bộ ba his trong bản rõ (điều này phù hợp với

giả định trước kia là dK(D) ∈{r,s,t}). Đoạn HNCMF có thể là bản mã của chair, điều này sẽ cho dK(F) = r (và dK(H) = c) và bởi vậy (bằng cách loại trừ) sẽ có $d_K(J) = t$.

Ta có:

o- r - riend - ro - - arise - a - inedhise - - t - - - ass - it

YIFQFMZRWQFYVECFMDZPCVMRZNMDZVEJBTXCDDUMJ

hs - r - riseasi - e - a - orationhadta - - en - -ace - hi - e

NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREZCHZUNMXZ

he - asnt - oo - in - i - o - redso - e - ore - ineandhesett

NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ

- ed - ac - inhischair - aceti - ted - - to - ardsthes - n

XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR

Bây giờ việc xác định bản rõ và khoá cho ở ví dụ trên không còn gì khó khăn nữa. Bản rõ hoàn chỉnh như sau:

Our friend from Pais examined his empty glass with surprise, as if evaporation had taen place while he wasn't looking. I poured some more wine and he settled back in his chair, face tilted up towards the sun.

2.2.3. Thám hệ mã Vigenère

Trong phần này chúng ta sẽ mô tả một số phương pháp thám hệ mã Vigenère. Bước đầu tiên là phải xác định độ dài từ khoá mà ta ký hiệu là m. ở đây dùng hai kỹ thuật. Kỹ thuật thứ nhất là phép thử Kasiski và kỹ thuật thứ hai sử dụng chỉ số trùng hợp.

Phép thử Kasiski lần đầu tiên được Kasiski Friendrich mô tả vào năm 1863. Kỹ thuật này được xây dựng trên nhận xét là: hai đoạn giống nhau của bản rõ sẽ được mã hoá thành cùng một bản mã khi chúng xuất hiện trong bản rõ cách nhau x vị trí, trong đó $x \equiv o \bmod m$. Ngược lại, nếu ta thấy hai đoạn giống nhau của bản mã (mỗi đoạn có độ dài ít nhất là 3) thì đó là một dấu hiệu tốt để nói rằng chúng tương ứng với các đoạn bản rõ giống nhau.

Phép thử Kasiski như sau. Ta tìm trong bản mã các cặp gồm các đoạn như nhau có độ dài tối thiểu là 3 và ghi lại khoảng cách giữa các vị trí bắt đầu của hai đoạn. Nếu thu được một vài giá trị d1, d2,. . . thì có thể hy vọng rằng m sẽ chia hết cho ước chung lớn nhất của các di.

Việc xác minh tiếp cho giá trị của m có thể nhận được bằng chỉ số trùng hợp. Khái niệm này đã được Wolfe Friedman đưa ra vào 1920 như sau:

Định nghĩa:

Giả sử x = x1x2 . . . xn là một xâu ký tự. Chỉ số trùng hợp của x (ký hiệu là Ic(x)) được định nghĩa là xác suất để hai phần tử ngẫu nhiên của x là đồng nhất. Nếu ký hiệu các tần suất của A,B,C,. . . ,Z trong x tương ứng là f0,f1 ,. . . f25 , có thể chọn hai phần tử của x theo ??? cách. Với mỗi i, $0 \leq i \leq 25$, có ??? cách chọn hai phần tử là i.

Bây giờ, giả sử x là một xâu văn bản tiếng Anh. Ta kí hiệu các xác suất xuất hiện của các kí tự A,B,. . .,Z trong bảng 1.1 là p0,...p25. Khi đó: do xác suất để hai phần tử ngẫu nhiên đều là A là p02, xác suất để cả hai phần tử này đều bằng B bằng p12 . . . Tình hình tương tự cũng xảy ra nếu x là một bản mã nhận được theo một hệ mã thay thế đơn bất kì. Trong trường hợp này, từng xác suất riêng rẽ sẽ bị hoán vị nhưng tổng ??? sẽ không thay đổi.

Bây giờ giả sử có một bản mã y = y1y2. . .ynđược cấu trúc theo mật mã

Vigenère. Ta xác định các xâu con m của y(y1,y2,. . .,ym) bằng cách viết ra bản mã thành một hình chữ nhật có kích thước m×(n/m). Các hàng của ma trận này là các xâu con yi, $1 \leq i \leq m$. Nếu m thực sự là độ dài khoá thì mỗi Ic(yi) phải xấp xỉ bằng 0,065. Ngược lại, nếu m không phải là độ dài khoá thì các xâu con yi sẽ có vẻ ngẫu nhiên hơn vì chúng nhận được bằng cách mã dịch vòng với các khoá khác nhau. Xét thấy rằng, một xâu hoàn toàn ngẫu nhiên sẽ có:

Hai giá trị 0,065 và 0,038 đủ cách xa nhau để có thể xác định được độ dài từ khoá đúng (hoặc xác nhận giả thuyết đã được làm theo phép thử Kasiski).

Hai kỹ thuật này sẽ được minh hoạ qua ví dụ dưới đây:

Ví dụ:

Bản mã nhận được từ mật mã Vigenère.

CHEEVOAHMAERATBTAXXWTNXBEEOPHBSBQMQEQERBW
RVXUOAKXAOSXXWEAHBWGJMMQMNKGRFVGXWTRZXWIAK
LXFPSKAUTEMNDCMGTSXMXBTUIADNGMGPSRELXNJELX
RVPRTULHDNQWTWDTYGBPHXTFEALJHASVBFXNGLLCHR
ZBWELEKMSJIKNBHWRJGNMGJSGLXFEYPHAGNRBIEQJT
MRVLCRRREMNDGLXRRIMGNSNRWCHRQHAEYEVTAQEBBI
EEWEVKAKOEWADREMXMTBHHCHRTKDNVRZCHRCLQOHP
WQAIIWXNRMGWOIIFKEE

Trước tiên, ta hãy thử bằng phép thử Kasiski xâu bản mã CHR xuất hiện ở bốn vị trí trong bản mã, bắt đầu ở các vị trí 1, 166,236 và 286. Khoảng cách từ lần xuất hiện đầu tiên tới 3 lần xuất hiện còn lại tương ứng là

165,235 và 285. UCLN của 3 số nguyên này là 5, bởi vậy giá trị này rất có thể là độ dài từ khoá.

Ta hãy xét xem liệu việc tính các chỉ số trùng hợp có cho kết luận tương tự không. Với m = 1 chỉ số trùng hợp là 0,045. Với m = 2, có 2 chỉ số là 0,046 và 0,041. Với m = 3 ta có 0,043; 0,050; 0,047. Với m = 4 các chỉ số là 0,042; 0,039; 0,046; 0,040. Với m = 5 ta có các giá trị 0,063; 0,068; 0,069; 0,061 và 0,072. Điều này càng chứng tỏ rằng độ dại từ khoá là 5.

CHƯƠNG 3
CHỨNG THỰC

Mục tiêu:

- Trình bày được chứng thực : chữ ký số,mật khẩu, sinh học;

- Cấu hình AAA trong Cisco;

Nội dung chính:

Giới thiệu: Trong chương này, chúng ta xem xét các sơ đồ chữ kí số (còn được gọi là chữ kí số). Chữ kí viết tay thông thường trên tài liệu thường được dùng để xác người kí nó. Chữ kí được dùng hàng ngày chẳng hạn như trên một bức thư nhận tiền từ nhà băng, kí hợp đồng…

Sơ đồ chữ kí là phương pháp kí một bức điện lưu dưới dạng điện tử.

Chẳng hạn một bức điện có ký hiệu được truyền trên mạng máy tinh. Chương này trình bày một vài sơ đồ chữ kí số. Ta sẽ thảo luận trên một vài khác biệt cơ bản giữa các chữ kí thông thường và chữ kí số.

Đầu tiên là một vấn đề kí một tài liệu. Với chữ kí thông thường, nó là một phần vật lý của tài liệu. Tuy nhiên, một chữ kí số không gắn theo kiểu vật lý vào bức điện nên thuật toán được dùng phải "không nhìn thấy" theo cách nào đó trên bức điện.

Thứ hai là vấn đề về kiểm tra. Chữ kí thông thường được kiểm tra bằng cách so sánh nó với các chữ kí xác thực khác. Ví dụ, ai đó kí một tấm séc để mua hàng, người bán phải so sánh chữ kí trên mảnh giấy với chữ kí nằm ở mặt sau của thẻ tín dụng để kiểm tra. Dĩ nhiên, đây không phải là phương pháp an toàn vì nó dể dàng giả mạo. Mắt khác, các chữ kí số có thể được kiểm tra nhờ dùng một thuật toán kiểm tra công khai. Như vậy, bất kỳ ai cũng có thể kiểm tra dược chữ kí số. Việc dùng một sơ đồ chữ kí an toàn có thể sẽ ngăn chặn được khả năng giả mạo.

Sự khác biệt cơ bản khác giữa chữ kí số và chữ kí thông thường bản copy tài liệu được kí băng chữ kí số đồng nhất với bản gốc, còn copy tài liệu có chữ kí trên giấy thường có thể khác với bản gốc. Điều này có nghĩa là phải cẩn thận ngăn chăn một bức kí số khỏi bị dung lại. Vì thế, bản thân bức điện cần chứa thông tin (chẳng hạn như ngày tháng) để ngăn nó khỏi bị dùng lại.

Một sơ đồ chữ kí số thường chứa hai thành phần: thuật toán kí và thuật toán xác minh. Bob có thể kí bức điện x dùng thuật toán kí an toàn. Chữ kí y=sig(x) nhận được có thể kiểm tra bằng thuật toán xác minh công khai ver(x,y). Khi cho trước cặp (x,y), thuật toán xác minh có giá trị TRUE hay FALSE tuỳ thuộc vào chữ kí được thực như thế nào. Dưới đây là định nghĩa hình thức của chữ kí:

3.1 Các định nghĩa

Một sơ đồ chữ kí số là bộ 5(P, A, K, S, V) thoả mãn các điều kiện dưới

đây:

1. P là tập hữu hạn các bức điện có thể.

2. A là tập hữu hạn các chữ kí có thể.

3. K không gian khoá là tập hữu hạn các khoá có thể.

4. Với mỗi k thuộc K tồn tại một thuật toán kí sigk S và là một thuật toán xác minh ver_k V. Mỗi sigk: P → A và ver_k: P×a →{true,false} là những hàm sao cho mỗi bức điện x P và mối chữ kí y A thoả mãn phương trình dưới đây.

True nếu y=sig(x)

ver_k

False nếu y# sig(x)

Với mỗi k thuộc K hàm sigk và verk là các hàm thời than đa thức. Ver_k sẽ là hàm công khai sigk là mật. Không thể dể dàng tính toán để giả mạo chữ kí của Bob trên bức điện x. Nghĩa là x cho trước, chỉ có Bob mới có thể tính được y để verk = True. Một sơ đồ chữ kí không thể an toàn vô điều kiện vì Oscar có thể kiểm tra tất cả các chữ số y có thể có trên bức điện x nhờ ung thuật toán ver công khai cho đến khi anh ta tìm thấy một chữ kí đúng. Vì thế, nếu có đủ thời gian. Oscar luôn luôn có thể giả mạo chữ kí của Bob. Như vậy, giống như trường hợp hệ thống mã khoá công khai, mục đích của chúng ta là tìm các sơ đồ chữ kí số an toan về mặt tính toán.

Xem thấy rằng, hệ thống mã khoá công khai RSA có thể ung làm sơ đồ chữ kí số.

Như vậy, Bob kí bức điện x dùng qui tắc giải mã RSA là dk. Bob là người tạo ra chữ kí vì dk = sigk là mật. Thuật toán xác minh dùng qui tắc mã RSA e_k.

Bất kì ai cũng có thể xác minh chữ kí vi ekđược công khai.

Chú ý rằng, ai đó có thể giả mạo chữ kí của Bob trên một bức điện " ngẫu nhiên" x bằng cách tìm x=ek(y) với y nào đó, khi đó y= sigk(x). Một giải pháp xung quanh vấn đề khó khăn này là yêu cầu bức điện chưa đủ phần dư để chữ kí giả mạo kiểu này không tương ứng với bức điện. Nghĩa là x trừ một xác suất rất bé. Có thể dùng các hàm hash trong việc kết nối với các sơ đồ chữ kí số sẽ loại trừ được phương pháp giả mạo này.

Sơ đồ chữ kí RSA

Cho n= p.q, p và q là các số nguyên tố. Cho P =A= Zn

$ab \equiv 1(\mathrm{mod}(\varphi(n)))$. Các giá trị n và b là công khai, a giữ bí mật.

Hàm kí:

$sig_k(x) = x^a \bmod n$

và kiểm tra chữ kí:

$ver_k(x,y) = true \quad x \equiv y^b \pmod n$

(x,y Zn)

3.2. Sơ đồ chữ kí ELGAMAL

Sau đây ta sẽ mô tả sơ đồ chữ kí Elgamal đã từng dưới thiệu trong bài báo năm 1985. Bản cả tiến của sơ đồ này đã được Viện Tiêu chuẩn và Công Nghệ Quốc Gia Mỹ (NIST) chấp nhận làm chữ kí số. Sơ đồ Elgamal (E.) được thiết kế với mục đích dành riêng cho chữ kí số, khác sơ đồ RSA dùng cho cả hệ thống mã khoá công khai lẫn chữ kí số.

Sơ đồ E, là không tất định giống như hệ thống mã khoá công khai Elgamal. Điều này có nghĩa là có nhiều chữ kí hợp lệ trên bức điện cho trước bất kỳ. Thuật toán xác minh phải có khả năng chấp nhận bất kì chữ kí hợp lệ khi xác thực.

Nếu chữ kí được thiết lập đúng khi xác minh sẽ thành công vì :

$\beta^{\gamma}\, \gamma^{\delta} \equiv \alpha^{a\gamma}\, \alpha^{k\gamma} \pmod p \equiv \alpha^{x} \pmod p$

là ở đây ta dùng hệ thức :

$a\gamma + k\delta \equiv x \pmod{p-1}$

Sơ đồ chữ kí số Elgamal.

Cho p là số nguyên tố sao cho bài toán logarit rời rạc trên Zp là khó và giả sử Znlà phần tử nguyên thuỷ p = Zp* , a = Zp*× Zp-1 và định nghĩa:

$K = \{(p, \alpha, a, \beta) : \beta \equiv \alpha a \pmod p\}$.

Giá trị p, α, β là công khai, còn a là mật.

Với K = (p, α , a, β) và một số ngẫu nhiên (mật) k Zp-1. định nghĩa :

Sigk(x,y) =(γ ,δ),

trong đó $\gamma = \alpha^k \bmod p$

và $\delta = (x-a)\, k^{-1} \bmod (p-1)$.

Với x,γ Zp và δ Zp-1 , ta định nghĩa :

Ver(x,γ ,δ) = true $\beta^{\gamma}\, \gamma^{\delta} \equiv \alpha^{x} \pmod p$.

3.3. Chuẩn chữ kí số.

Chuẩn chữ kí số(DSS) là phiên bản cải tiến của sơ đồ chữ kí Elgamal. Nó được công bố trong Hồ Sơ trong liên bang vào ngày 19/5/94 và được làm chuẩn voà 1/12/94 tuy đã được đề xuất từ 8/91. Trước hết ta sẽ nêu ra những

thay đổi của nó so với sơ đồ Elgamal và sau đó sẽ mô tả cách thực hiện nó.Trong nhiều tinh huống, thông báo có thể mã và giải mã chỉ một lần nên nó phù hợp cho việc dùng với hệ mật bất kì (an toàn tại thời điểm được mã).

Song trên thực tế, nhiều khi một bức điện được dùng làm một tài liệu đối chứng, chẳng hạn như bản hợp đồng hay một chúc thư và vì thế cần xác minh chữ kí sau nhiều năm kể từ lúc bức điện được kí. Bởi vậy, điều quan trọng là có phương án dự phòng liên quan đến sự an toàn của sơ đồ chữ kí khi đối mặt với hệ thống mã. Vì sơ đồ Elgamal không an toàn hơn bài toán logarithm rời rạc nên cần dung modulo p lớn. Chắc chắn p cần ít nhất là 512 bít và nhiều người nhất trí là p nên lấy p=1024 bít để có độ an toàn tốt.

3.4 Xác thực mẫu tin

3.4.1 Các khái niệm

Xác thực mẫu tin liên quan đến các khía cạnh sau khi truyền tin trên mạng

o Bảo vệ tính toàn vẹn của mẫu tin: bảo vệ mẫu tin không bị thay đổi hoặc có các biện pháp phát hiện nếu mẫu tin bị thay đổi trên đường truyền.

o Kiểm chứng danh tính và nguồn gốc: xem xét mẫu tin có đúng do người xưng tên gửi không hay một kẻ mạo danh nào khác gửi.

o Không chối từ bản gốc: trong trường hợp cần thiết, bản thân mẫu tin chứa các thông tin chứng tỏ chỉ có người xưng danh gửi, không một ai khác có thể làm điều đó. Như vậy người gửi không thể từ chối hành động gửi, thời gian gửi và nội dung của mẫu tin.

Ngoài ra có thể xem xét bổ sung thêm các yêu cầu bảo mật như mã hoá. Với mong muốn đáp ứng các yêu cầu trên, có 3 hàm lựa chọn sau đây được sử dụng:

o Mã mẫu tin bằng mã đối xứng hoặc mã công khai.

o Mã xác thực mẫu tin (MAC): dùng khoá và một hàm nén mẫu tin cần gửi để nhận được một đặc trưng đính kèm với mẫu tin và người gửi đó.

o Hàm hash (hàm băm) là hàm nén mẫu tin tạo thành "dấu vân tay" cho mẫu tin.

Các yêu cầu bảo mật khi truyền mẫu tin trên mạng.

Tìm các biện pháp cần thiết để chống đối lại các hành động phá hoại như sau:

o Để lộ bí mật: giữ bí mật nội dung mẫu tin, chỉ cho người có quyền biết.

o Thám mã đường truyền: không cho theo dõi hoặc làm trì hoãn việc truyền tin.

o Giả mạo: lấy danh nghĩa người khác để gửi tin.

o Sửa đổi nội dung: thay đổi, cắt xén, thêm bớt thông tin.

o Thay đổi trình tự các gói tin nhỏ của mẫu tin truyền.

o Sửa đổi thời gian: làm trì hoãn mẫu tin.

o Từ chối gốc: không cho phép người gửi từ chối trách nhiệm của tác giả mẫu tin.

o Từ chối đích: không cho phép người nhận phủ định sự tồn tại và đến đích của mẫu tin đã gửi.

3.4.2 Mã mẫu tin

• Mã mẫu tin bản thân đã cung cấp một phần tính xác thực, vì khoá được chia sẻ giữa người gửi và người nhận cũng như việc thay đổi nội dung cũng không dễ dàng thực hiện nếu không có khoá.

• Cụ thể nếu mã đối xứng được sử dụng thì người nhận biết người gửi phải tạo ra mẫu tin, vì chỉ có người gửi và người nhận biết được khoá sử dụng.

Người nhận có thể biết nội dung không bị sửa đổi, nếu mẫu tin có cấu trúc phù hợp, tính dư thừa và tổng kiểm tra để phát hiện bất cứ thay đổi nào.

• Nếu khoá công khai được sử dụng thì mã cung cấp không đủ độ tin cậy về người gửi, vì mọi người đều có thể biết khoá công khai của người nhận. Tuy nhiên nếu người gửi ký mẫu tin sử dụng khoá riêng của họ và sau đó mã với khoá công khai của người nhận, thì khi đó đảm bảo cả tính bảo mật và xác thực của mẫu tin. Cần phải bổ sung các biện pháp để phát hiện các mẫu tin đã bị làm hỏng. Việc sử dụng khoá riêng của người gửi kết hợp với khoá công khai của người nhận có nhiều ưu việt, nhưng với giá phải trả là chậm do dùng 2 mã khoá công khai trên mẫu tin.

3.4.3 Mã xác thực mẫu tin (MAC – Message Authentication Code)

Sinh ra bởi một thuật toán mà tạo ra một khối thông tin nhỏ có kích thước cố định

o Phụ thuộc vào cả mẫu tin và khoá nào đó.

o Giống như mã nhưng không cần phải giải mã.

• Bổ sung vào mẫu tin như chữ ký để gửi kèm theo làm bằng chứng xác thực.

• Người nhận thực hiện tính toán nào đó trên mẫu tin và kiểm tra xem nó có phù hợp với MAC đính kèm không.

• Tạo niềm tin rằng mẫu tin không bị thay đổi và đến từ người gửi.

Các mã xác thực mẫu tin MAC cung cấp sự tin cậy cho người nhận là mẫu tin không bị thay đổi và từ đích danh người gửi. Cũng có thể sử dụng mã xác thực MAC kèm theo với việc mã hoá để bảo mật. Nói chung người ta sử dụng các khoá riêng biệt cho mỗi MAC và có thể tính MAC trước hoặc sau mã hoá, tốt hơn là thực hiện MAC trước và mã hoá sau.

Sử dụng MAC có nhược điểm là MAC phụ thuộc vào cả mẫu tin và cả người gửi, nhưng đôi khi chỉ cần xác thực mẫu tin và thông tin xác thực đó chỉ phụ thuộc mẫu tin để lưu trữ làm bằng chứng cho tính toàn vẹn của nó. Khi đó người ta sử dụng hàm Hash thay vì MAC. Cần lưu ý rằng MAC không phải là chữ ký điện tử, vì cả người gửi và người nhận đều biết thông tin về khoá.

Các tính chất của MAC

MAC là thông tin nén của mẫu tin kết hợp với khoá MAC = CK(M)

o Nén bản tin M có độ dài tùy ý

o Sử dụng khoá mật K

o Tạo nên dấu xác thực có độ dài cố định

o Là hàm nhiều - một, nghĩa là có nhiều bản tin khác nhau nhưng có cùng MAC. Tuy nhiên ta phải lựa chọn hàm MAC sao cho xác suất để các mẫu tin có ý nghĩa có MAC trùng nhau là rất nhỏ. Việc tìm được các mẫu tin như vậy là rất khó khăn

Yêu cầu đối với MAC

Tuỳ thuộc vào kiểu tấn công mà MAC phải có các tính chất khác nhau để chống đối lại. Nhưng nói chung MAC phải thỏa mãn các điều sau

o Biết mẫu tin và MAC, không thể tìm được mẫu tin khác có cùng MAC.

o Các MAC cần phải phân bố đều

o MAC phải phụ thuộc như nhau vào tất cả các bit trong mẫu tin. Tức là khi thay đổi một bit thông tin nào đó, MAC sẽ có những thay đổi kéo theo.

3.4..4 Sử dụng mã đối xứng cho MAC

• Có thể dùng mã khối với chế độ chuỗi móc nối bất kỳ và sử dụng khối cuối cùng của mã khối làm MAC của mẫu tin.

• Thuật toán xác thực dữ liệu (DAA – Data Authentication Algorithm) là MAC được sử dụng rộng rãi dựa trên chế độ DES-CBC, trong đó

o Sử dụng véc tơ ban đầu IV = 0 và bộ đệm 0 của block cuối cùng

o Và mã mẫu tin sử dụng chuẩn mã dữ liệu DES trong chế độ CBC

o Gửi lấy block cuối cùng như là MAC của cả mẫu tin

Nhưng bây giờ MAC cuối cùng với kích thước 64 bit cũng là quá nhỏ để đảm bảo an toàn. Do đó người ta tìm cách tạo nên các MAC có kích thước lớn hơn.

3.5 Các hàm Hash (hay còn gọi là hàm băm).

3.5.1 Các yêu cầu

Nén mẫu tin bất kỳ về kích thước cố định. Và giả thiết là hàm hash là công khai và không dùng khoá. Hash chỉ phụ thuộc mẫu tin, còn MAC phụ thuộc thêm cả vào khoá.

Hash được sử dụng để phát hiện thay đổi của mẫu tin. Hash có thể sử dụng nhiều cách khác nhau với mẫu tin, Hash thường được kết hợp dùng để tạo chữ ký trên mẫu tin.

Các tính chất của hàm Hash

Hàm Hash tạo nên dấu vân tay (tức là thông tin đặc trưng) của một tệp, mẫu tin hay dữ liệu

h = H(M)

Nén mẫu tin có kích thước tùy ý về dấu vân tay có kích thước cố định. Hàm Hash được giả thiết là công khai, mọi người đều biết cách sử dụng

Các yêu cầu của hàm Hash

Có thể áp dụng cho mọi mẫu tin có kích thước tuỳ ý. Tuy nhiên phải tạo đầu ra h có kích thước cố định, thường là 128 bit đến 1024 bit.

Dễ tính h = H(M)cho mọi mẫu tin M, hàm H tính toán nhanh, hiệu quả phụ thuộc chặt vào mẫu tin M và không tính toán ngược lại.

Cho trước h không thể tìm được (rất khó) x sao cho H(x) = h. Tính chất này gọi là tính chất một chiều, chiều tìm nghịch ảnh rất khó khăn, tuy chiều tìm ảnh lại dễ dàng.

Cho x không thể tìm được y sao cho H(y) = H(x). Đây là tính chất chống đỡ va chạm yếu, không tìm được mẫu tin có cùng Hash với mẫu tin đã cho.

Và không thể tìm được x, y sao cho H(y) = H(x). Đây gọi là tính chất chống đỡ va chạm mạnh, đây là yêu cầu cao hơn tính chống đỡ va chạm yếu.

3.5.2 Các hàm hash đơn giản

Có một số đề xuất cho một số hàm hash đơn giản. Chẳng hạn biểu diễn mẫu tin dưới dạng bit sau đó chia chúng thành các khối bit có kích thước bằng kích thước mong muốn của Hash. Rồi dựa trên phép toán XOR các bit

thông tin ở cùng vị trí tương ứng của các khối, kết quả nhận được là Hash của cả mẫu tin. Hàm hash trên là không an toàn vì đối với mẫu tin bất kỳ có thể tìm được mẫu tin mà có cùng hàm hash.

Có thể nghĩ hash 64 bit là an toàn, có nghĩa là khó tìm được bản tin có cùng hash. Nhưng không phải vậy vì nghịch lý ngày sinh nhật như sau: trong lớp có ít nhất bao nhiêu sinh viên, để xác suất có ít nhất 2 sinh viên trùng ngày sinh nhật là lớn hơn 0.5. Theo lý thuyết xác suất thống kê gọi số sinh viên ít nhất trong lớp là k, khi đó xác suất q để không có 2 người nào trùng ngày sinh là tỷ số giữa cách chọn k ngày khác nhau trong 365 ngày trên số cách chọn k ngày bất kỳ trong 365 ngày. Vậy

$$q = Ck365 / 365k$$

Do đó, xác suất p để có ít nhất 2 người trùng ngày sinh là

$$p = 1 – q = 1 - Ck365 / 365k$$

Để $p > 0.5$ thì $k > 22$ hay $k = 23$, cụ thể khi đó $p = 0.5073$.

Khi chưa tính toán chi tiết chúng ta nghĩ là trong lớp phải có ít nhất khoảng 365/2 tức là 184 sinh viên. Nhưng trên thực tế con số đó ít hơn rất nhiều chỉ cần 23 sinh viên, chính vì vậy ta gọi đây là nghịch lý ngày sinh nhật.

Điều đó muốn nói lên rằng, trong nhiều trường hợp xác suất để hai mẫu tin có cùng bản Hash là không nhỏ như chúng ta tưởng.

3.5.3 Tính an toàn của hàm Hash và MAC.

Giống như đối với mã khối, hàm hash cũng có tấn công vét cạn, cụ thể: Hash chống va chạm mạnh có giá 2m/2, có nghĩa là với m là độ dài mã hash thì 2m/2 xác định sức mạnh của nó chống đối lại tấn công vét cạn. Ta cần lựa chọn m đủ lớn để việc duyệt tìm 2m/2 phương án là không khả thi. Có đề xuất Hash 128 bit cho MD5 phần cứng. Nhưng có thể tìm được va chạm sau 24 ngày. Do đó có thể coi là hash 128 bit có thể có lỗ hổng, không an toàn, tốt hơn dùng hash 160 bit.

Tấn công vét cạn trên MAC khó hơn, vì chúng đòi hỏi một cặp MAC của mẫu tin đã biết, do nó phụ thuộc thêm vào khoá. Có thể tấn công vào không gian khoá (như là tìm khoá) hoặc MAC. Độ dài ít nhất 128 bit MAC là cần thiết để đảm bảo an toàn

Thám mã tấn công có cấu trúc

Giống như mã khối muốn dùng tấn công vét cạn, có một số các tấn công thám mã là lựa chọn tốt nhất hiện có. Chẳng hạn

- Nếu $CVi = f[CVi-1, Mi]$; $H(M)=CVN$
- Thì ở đây thông thường khai thác sự va chạm của hàm f

§ Giống mã khối thường gồm một số vòng lặp

§ Khi đó tân công sử dụng các tính chất của các hàm vòng.

3.6 Các thuật toán Hash và MAC

3.6.1 Các thuật toán Hash và MAC

Hàm Hash: thực hiện việc nén mẫu tin vê kích thước cố định bằng cách xử lý mẫu tin theo từng khối kết hợp dùng một hàm nén nào đó và có thể sử dụng mã khối.

Mã xác thực mẫu tin (MAC): thực hiện tạo phần xác thực cho mẫu tin có kích thước cố định, để cung cấp tính toàn vẹn của mẫu tin và tính xác thực thông qua việc sử dụng khoá. Có thể tiíen hành bằng cách sử dụng mã khối với chế độ móc nối hoặc hàm Hash.

3.6.2 Thuật toán Hash an toàn SHA (Secure Hash Algorithm)

SHA có nguồn gốc từ Viện chuẩn công nghệ quốc gia Hoa kỳ - NIST & NSA vào năm 1993, sau đó được nâng cấp vào 1995 theo chuẩn US và chuẩn là FIPS 180-1 1995 và Internet RFC3174, được nhắc đến như SHA-1. Nó được sử dụng với sơ đồ chữ ký điện tử DSA (Digital Signature Algorithm).

Thuật toán là SHA dựa trên thiết kế MD4 với một số khác biệt tạo nên giá trị Hash 160 bit. Các kết quả nghiên cứu 2005 về an toàn của SHA-1 đề xuất sử dụng nó trong tương lai.

Sau đây ta mô tả chi tiết thuật toán SHA-1 và MD5:

a. Thuật toán SHA-1

Mô tả thụât toán

Đầu vào của thuật toán là một thông điệp có chiều dài bất kỳ nhỏ hơn 264 bit, SHA-1 cho ra kết quả là một thông điệp rút gọn có độ dài là 160 bit

Mở rộng thông điệp:

f(t;B,C,D) được định nghĩa như sau.

f(t;B,C,D) = (B AND C) OR ((NOT B) AND D) (0≤t≤19)

f(t;B,C,D) = B XOR C XOR D (20≤t≤39)

f(t;B,C,D) = (B AND C) OR (B AND D) OR (C AND D) (40≤t≤59)

f(t;B,C,D) = B XOR C XOR D (60≤t≤79).

Thông điệp M được mở rộng trước khi thực hiện băm. Mục đích của việc mở rộng này là để đảm bảo cho thông điệp mở rộng có độ dài là bội số của 512.

Giả sử độ dài của thông điệp là l bit. Thêm bit 1 vào cuối thông điệp, theo sau là k bit 0 (k là số dương không âm nhỏ nhất sao cho l+1+k=448 (mod512)) . Sau đó thêm khối 64 bit là biểu diễn nhị phân của l.

Phân tích thông điệp mở rộng:

Sau khi thông điệp đã được mở rộng, thông điệp mở rộng được phân tích thành N khối 512 bit M(1),M(2),…,M(N). Trong đó 512 bit của khối dữ liệu đầu vào có thể được thể hiện bằng 16 từ 32 bit,

Khởi tạo giá trị băm:

Giá trị băm là một chuỗi bit có kích thước bằng kích thước của thông điệp băm (trừ SHA-384) gồm các từ ghép lại. Trong đó Hj(i) là từ j trong giá trị băm ở lần lặp i với 0≤i≤N (số block có được sau khi chia văn bản được đệm) và 0≤j≤(số từ trong giá trị băm -1).Trước khi thực hiện giá trị băm, với mỗi thuật toán băm an toàn, giá trị băm ban đầu H(0) phải được thiết lập. Kích thước và số lượng từ trong H(0) tuỳ thuộc vào kích thước thông điệp rút gọn.

SHA-1 sử dụng dãy các hằng số K(0),…K(79) có giá trị như sau:

K(t) = 5A827999 (0 <= t <= 19)

K(t) = 6ED9EBA1 (20 <= t <= 39)

K(t) = 8F1BBCDC (40 <= t <= 59)

K(t) = CA62C1D6 (60 <= t <= 79).

Thuật toán của bước tính giá trị băm SHA-1

SHA-1 được sử dụng để băm thông điệp M có độ dài l bit thoả mãn điều kiện 0≤l≤264 . Thuật toán sử dụng:

- Một bảng phân bố thông điệp gồm 80 từ 32 bit
- 5 biến 32 bit
- Một giá trị băm gồm 5 từ 32 bit

Kết quả của SHA-1 là một thông điệp rút gọn có độ dài 160 bit. Các từ của bảng phân bố thông điệp được ký hiệu W(0),W(1),…,W(79). 5 biến được ký hiệu là a,b,c,d,e. Các từ của giá trị băm ký hiệu H0(i),H1(i), H2(i), H3(i),H4(i). H(0) giữ giá trị băm ban đầu và được thay thế bằng các giá trị băm thành công.H(i) sau mỗi khối thông điệp được xử lý và kết thúc bằng giá trị băm cuối cùng H(N).

Tính toán thông điệp băm

Định nghĩa: S^n(X)=(X<<n) or (X>>32-n).

X<<n có nghĩa là loại bỏ từ trái sang phải n bit và thêm vào kết quả n số 0 vào bên phải. X>> có nghĩa là loại bỏ từ phải qua trái n bit và thêm vào kết quả n số 0 vào bên trái.

Khởi tạo H

H0 = 67452301 ; H1 = EFCDAB89

H2 = 98BADCFE ; H3 = 10325476

H4 = C3D2E1F0.

Chia M(i) thành 16 từ W(0), W(1),…,W(15)

For t = 16 to 79

- W(t) = S^1(W(t-3) XOR W(t-8) XOR W(t-14) XOR W(t-16)).
- Đặt a=H0 , b=H1,c=H2,d=H3,e=H4

For t = 0 to 79 do

- TEMP = S^5(A) + f(t;B,C,D) + E + W(t) + K(t);
- e = d; d = c; c = S^30(b); b = a; a = TEMP;
- Đặt H0 = H0 + a,H1 = H1 + b,H2 = H2 + c,H3 = H3 + d,H4 = H4+ e.

Sau khi tính toán được hết M(n), thông điệp rút gọn là một chuỗi 160 bit là biểu diễn của 5 từ: H0 H1 H2 H3 H4

Đánh giá thuật toán

- SHA-1 được xem là an toàn đối với hiện tượng đụng độ vì rất khó tìm được hai thông điệp khác nhau có giá trị băm giống nhau
- SHA-1 được coi là chuẩn của việc bảo vệ các kênh liên lạc trực tuyến tồn tại trong 9 năm qua.
- SHA-1 được thiết kế cho bộ xử lý 32 bit, thế hệ sắp tới của máy tính dùng các bộ xử lý 64 bit mà SHA-1 không hiệu quả trên bộ xử lý này.
- Tháng 2 năm 2005 SHA-1 bị tấn công bởi 3 chuyên gia người Trung Quốc. Thuật toán này đã bị giải mã thông qua phương pháp tính phân bổ.

b. Thuật toán MD5

Mô tả thuật toán

Thuật toán có đầu vào là một thông điệp có độ dài tuỳ ý và có đầu ra là một chuỗi có độ dài cố định là 128 bit. Thuật toán được thiết kế để chạy trên các máy tính 32 bit.

Thuật toán:

Thông điệp đầu vào có độ dài b bit bất kỳ. Biểu diễn các bit dưới dạng như sau: m[0] m[1] m[2] ... m[b-1]

Bước1: Các bit gắn thêm : Thông điệp được mở rộng, thêm bit vào phía sau sao cho độ dài của nó (bit) đồng dư với 448 theo môđun 512. Nghĩa là thông điệp được mở rộng sao cho nó còn thiếu 64 bit nữa thì sẽ có một độ dài chia hết cho 512. Việc thêm bit này được thực hiện như sau: một bit '1' được thêm vào sau thông điệp, sau đó các bit '0' được thêm vào để có một độ dài đồng dư với 448 môđun 512.

Bước 2: Gắn thêm độ dài: Dạng biểu diễn 64 bit độ dài b của chuỗi ban đầu được thêm vào phía sau kết quả của bước 1.

Bước 3: Khởi tạo bộ đệm MD: Một bộ đệm 4 từ (A,B,C,D) được dùng để tính mã số thông điệp. Ở đây mỗi A,B,C,D là một thanh ghi 32 bit. Những thanh ghi này được khởi tạo theo những giá trị hex sau :

A=0x01234567

B=0x89abcdef

C=0xfedcba98

D=0x76543210

Bước 4 :Xử lý thông điệp theo từng khối 16 từ. Định nghĩa các hàm phụ, các hàm này nhận giá trị đầu vào là 3 từ 32 bit và tạo tạo ra một word 32 bit.

F(X,Y,Z) = XY v not(X) Z

G(X,Y,Z)= XZ v Y not(Z)

H(X,Y,Z) = X xor Y xor Z

I(X,Y,Z) = Y xor (X v not(Z))

Bước này sử dụng một bảng 64 giá trị T[1 .. 64] được tạo ra từ hàm sin. Gọi T là phần tử thứ i của bảng, thì T là phần nguyên của 4294967296*| sin(i)| , i được tính theo radian.

Thuật toán

```
/* Xử lý với mỗi khối 16 bit từ */
  For i = 0 to N/16-1 do
/* Sao khối i vào X. */
    For j = 0 to 15 do
     Set X[j] to M[i*16+j].
    end
```

AA = A

BB = B

CC = C

DD = D

/* Vòng 1: Ký hiệu [abcd k s i] là thao tác sau

a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s). */

/* Làm 16 thao tác sau đây*/

[ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4]
[ABCD 4 7 5] [DABC 5 12 6] [CDAB 6 17 7] [BCDA 7 22 8]
[ABCD 8 7 9] [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]
[ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]

/* Vòng 2: Ký hiệu [abcd k s i] là thao tác sau đây

a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s). */

/* Làm 16 thao tác sau đây*/

[ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]
[ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]
[ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]
[ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]

/* Vòng 3: Ký hiệu [abcd k s t] là thao tác sau đây

a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s). */

/* Làm 16 thao tác sau đây*/

[ABCD 5 4 33] [DABC 8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]
[ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]
[ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]
[ABCD 9 4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA 2 23 48]

/* Vòng 4: Ký hiệu [abcd k s t] là thao tác sau đây

a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s). */

/* Làm 16 thao tác sau đây*/

[ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]
[ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]
[ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]

[ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]

/* Tính */

A = A + AA

B = B + BB

C = C + CC

D = D + DD

end /* Kêt thúc vòng lặp trên i*/

Bước 5: Thông điệp rút gọn = A||B||C||D.

Đánh giá thuật toán MD5

Về tốc độ sinh ra chuỗi cốt yếu thì MD5 chậm hơn so với MD4 nhưng nó lại an toàn hơn rất nhiều so với MD4. Thuật toán số hóa thông điệp MD5 khá đơn giản để thực hiện, cung cấp một giá trị băm của thông điệp với độ dài tuỳ ý. Người ta cho rằng độ khó để tìm được 2 thông điệp có cùng giá trị băm là khoảng 264 bước tính, và độ khó để tìm được một thông điệp với giá trị băm cho trước là 2128 bước tính. Tuy nhiên lỗ hổng mới phát hiện trong thuật toán MD5 sẽ cho phép kẻ tấn công có thể tạo ra file giả mạo trong vòng vài giờ với loại máy tính đạt chuẩn.

Chuẩn Hash an toàn nâng cao

Viện chuẩn công nghệ quốc gia NIST xuất bản bản sửa FIPS 180-2 vào năm 2002, đề nghị bổ sung 3 phiên bản mới của SHA: SHA-256, SHA-384, SHA-512. Các phiên bản trên được thiết kế tương thích với việc tăng độ an toàn được cung cấp bởi chuẩn mã nâng cao AES. Về cấu trúc và chi tiết giống SHA-1, suy ra việc phân tích cũng tương tự, nhưng mức độ an toàn cao hơn nhiều so với SHA-1.

Tổng quan SHA 512

Hàm nén SHA-512

SHA-512 là trọng tâm của thuật toán. Ở đây xử lý mẫu tin với các khối 1024 bit và bao gồm 80 vòng cập nhật bộ đệm 512 bit. Sử dụng giá trị Wt 64 bit được lấy ra từ block hiện tại của mẫu tin và hằng số quay vòng dựa trên căn bậc ba của 80 số nguyên tố đầu tiên

3.7 Các ứng dụng xác thực

Chúng ta sẽ xem xét các hàm xác thực được phát triển để hỗ trợ xác thực mức ứng dụng và chữ ký điện tử. Đồng thời cũng xem xét dịch vụ xác thực dùng khoá riêng Kerrberros. Và sau đó xét dịch vụ xác thực dùng khoá công khai X.509.

3.7.1 Kerberos

Đây là mô hình Hệ thống khoá máy chủ tin cậy của MIT (Trường Đại học Kỹ thuật Massachusetts) để cung cấp xác thực có bên thứ ba dùng khoá riêng và tập trung. Cho phép người sử dụng truy cập vào các dịch vụ phân tán trong mạng. Tuy nhiên không cần thiết phải tin cậy mọi máy trạm, thay vì đó chỉ cần tin cậy máy chủ xác thực trung tâm. Đã có hai phiên bản đang sử dụng là: Kerberos 4 và Kerberos 5.

a. Các yêu cầu của Kerrberos

Báo cáo đầu tiên của: Kerberos nêu các yêu cầu sau

- An toàn
- Tin cậy
- Trong suốt
- Có thể mở rộng

Ở đây cài đặt sử dụng thủ tục xác thực Needham-Schroeder.

b.Tổng quan Kerberos 4

Là sơ đồ xác thực dùng bên thứ ba cơ bản và có máy chủ xác thực (AS – Authentication Server). Người dùng thỏa thuận với AS về danh tính của mình, AS cung cấp sự tin cậy xác thực thông qua thẻ cấp thẻ TGT (Ticket Granting Ticket) và máy chủ cung cấp thẻ (TGS – Ticket Granting Server). Người sử dụng thường xuyên yêu cầu TGS cho truy cập đến các dịch vụ khác dựa trên thẻ cấp thẻ TGT của người sử dụng.

c.Trao đổi Kerberos 4

Người sử dụng nhận thẻ được cấp từ máy chủ xác thực AS, mỗi thẻ cho một phiên làm việc và cũng nhận thẻ cấp dùng dịch vụ (service granting ticket) từ TGT. Mỗi thẻ dùng cho một dịch vụ khác nhau được yêu cầu, thông qua việc trao đổi giữa máy chủ/trạm để nhận được dịch vụ.

d. Các lãnh địa Kerberos

Môi trường Kerberos bao gồm: máy chủ Kerberos, một số máy trạm đã được đăng ký với máy chủ, các máy chủ ứng dụng chia sẻ khoá với máy chủ. Một hệ thống như vậy được gọi là một lãnh địa Kerberos. Thông thường là một miền hành chính duy nhất. Nếu có nhiều lãnh địa, thì các máy chủ Kerberos cần phải chia sẻ khoá và tin cậy nhau.

e. Kerberos phiên bản 5

Kerberos 5 được phát triển vào giữa những năm 1990, được thiết kế theo chuẩn RFC 1510. Nó cung cấp những cải tiến so với phiên bản 4, cụ thể hướng tới các thiếu xót về môi trường, thuật toán mã, thủ tục mạng thứ tự byte, thời gian sử dụng thẻ, truyền tiếp xác thực, xác thực lãnh địa con. Và

các sự khác biệt về kỹ thuật như: mã kép, các dạng sử dụng không chuẩn, khoá phiên, chống tấn công mật khẩu.

Kerberos la`một giao thức xác thực mạng, no´cho phep cac ca´nhân giao tiêp vơí nhau trên một mạng không an toaàn băèng cách xac thực ngươì duèng nay vơí ngươì duèng khac theo một cơ chê´bảo mật va`an toaàn. Kerberos ngăn chặn việc nghe trộm thông tin cuñg như tấn công thay thế va`đảm bảo tính toaàn vẹn của dư˜liệu. Kerberos hoạt động theo mô hình máy trạm/máy chủ va`no´thực hiện qua´trình xác thực 2 chiêù - cả ngươì duèng va`dịch vụ xac thực lâñ nhau. Kerberos được xây dựng dựa trên mô hình ma˜hoá khoá đôí xứng va`đoì hỏi một thaành phâàn thư´ba tin cậy tham gia vào qua´trình xác thực.

Kerberos sử dụng một đôí tac tin cậy thư´ba để thực hiện qua´trình chứng thực được gọi la`Trung tâm phân phôí khoá bao gôìm 2 phâàn riêng biệt: một máy chủ chứng thực (AS) va`một máy chủ câp thẻ (TGS). Kerberos laàm việc dựa trên cac thẻ để thực hiện qua´trình chứng thực ngươì duèng.

Kerberos duy tri`một cơ sở dư˜liệu chưá cac khoá bí mật. Môĩ thực thể trên mạng (máy trạm hoặc máy chủ) đêù chia se˜một khoá bí mật chỉ giưã bản thân no´vơí Kerberos. Để thực hiện qua´trình giao tiêp giưã 2 thực thể, Kerberos tạo ra một khoá phiên. Khoá nay duèng để bảo mật qua´trình tương tac giưã cac thực thể vơí nhau.

Hoạt động của Kerberos:

Qua´trình hoạt động của giao thức (AS = Máy chủ xác thực, TGS = Máy chủ câp thẻ, C = Máy trạm, S = Dịch vụ):

Ngươì duèng nhập vào tên truy cập và mật khẩu ở phiá máy trạm.

Máy trạm thực hiện thuật toaán băm một chiêù trên mật khẩu được nhập vào va`no´trở thaành khoá bí mật của máy trạm.

Máy trạm gởi một thông điệp dươí dạng bản rõ đến AS để yêu câù dịch vụ. Không co´khoá bí mật cuñg như mật khẩu naò được gởi đến AS.

AS kiểm tra xem co´tôn tại ngươì duèng C trong cở sở dư˜liệu của no´hay không. Nêú co´, no´gởi ngược lại cho máy trạm 2 thông điệp:

Thông điệp A: chưá khoá phiên Máy trạm/TGS được ma˜hoá bởi khoá bí mật của ngươì duèng.

Thông điệp B: chưá Thẻ (bao gôìm ID của máy trạm, địa chỉ mạng của máy trạm, kỳ hạn thẻ có giá trị va`một khoá phiên máy trạm/TGS) được ma˜ hoá sử dụng khoá bí mật của TGS.

5. Khi máy trạm nhận được thông điệp A va`B, no´giải ma˜thông điệp A để lâý khoá phiên máy trạm/TGS. Khoá phiên nay được sử dụng cho qua´trình giao đổi tiêp theo vơí TGS. Ở đây máy trạm không thể giải ma˜thông điệp B bởi vi`no´được ma˜hoá bởi khoá bí mật của TGS.

6. Khi yêu câù dịch vụ (S), máy trạm gởi 2 thông điệp sau đến TGS:

o Thông điệp C: Gồm thông điệp B va`ID của dịch vụ được yêu câù

o Thông điệp D: chứa Authenticator (gồm ID máy trạm va`nhãn thời gian -timestamp) được ma~hoá bởi khoá phiên Máy trạm/TGS.

7. Khi nhận được thông điệp C va`D, TGS giải ma~thông điệp D sử dụng khoá phiên máy trạm/TGS va`gởi 2 thông điệp ngược lại cho máy trạm:

o Thông điệp E: chứa thẻ (máy trạm đến máy chủ) (bao gồm ID máy trạm, địa chỉ mạng của máy trạm, kỳ hạn thẻ có giá trị va`một khoá phiên máy trạm/dịch vụ) được ma~hoá bởi khoá bí mật của dịch vụ.

o Thông điệp F: chứa khoá phiên của máy trạm/máy chủ được ma~ hoá bởi khoá phiên máy trạm/TGS.

8. Khi nhận được thông điệp E va` F, máy trạm sau đo´ gởi một Authenticator mơí va`một thẻ (máy trạm đến máy chủ) đến máy chủ chứa dịch vụ được yêu câù.

o Thông điệp G: chứa thẻ (máy trạm đến máy chủ) được ma~hoá sử dụng khoá bí mật của máy chủ.

o Thông điệp H: một Authenticator mơí chứa ID máy trạm, Timestamp va`được ma~hoá sử dụng khoá phiên máy trạm/máy chủ.

9. Sau đo´, máy chủ giải ma~thẻ sử dụng khoá bí mật của chính no´, va` gởi một thông điệp cho máy trạm để xac nhận tính hợp lệ thực sự của máy trạm va`sự săn saàng cung câp dịch vụ cho máy trạm.

o Thông điệp I: chứa gia´trị Timestamp trong Authenticator được gởi bởi máy trạm se~được cộng thêm 1, được ma~hoá bởi khoá phiên máy trạm/máy chủ.

10. Máy trạm se~giải ma~sự xac nhận nay sử dụng khoá chia se~giữa no´vơí máy chủ, va`kiểm tra xem gia´trị timestamp co´được cập nhật đuíng hay không. Nêú đuíng, máy trạm co´thể tin tưởng máy chủ va`bắt đâù đưa ra cac yêu câù dịch vụ gởi đến máy chủ.

11. Máy chủ cung câp dịch vụ được yêu câù đến máy trạm.

Hạn chế của Kerberos

Kerberos thích hợp cho việc cung cấp các dịch vụ xác thực, phân quyền và bảo đảm tính mật của thông tin trao đổi trong phạm vi một mạng hay một tập hợp nhỏ các mạng. Tuy nhiên, nó không thật thích hợp cho một số chức năng khác, chẳng hạn như ký điện tử (yêu cầu đáp ứng cả hai nhu cầu xác thực và bảo đảm không chối cãi được). Một trong những giả thiết quan trọng của giao thức Kerberos là các máy chủ trên mạng cần phải tin cậy được. Ngoài ra, nếu người dùng chọn những mật khẩu dễ đoán thì hệ thống dễ bị

mất an toàn trước kiểu tấn công từ điển, tức là kẻ tấn công sẽ sử dụng phương thức đơn giản là thử nhiều mật khẩu khác nhau cho đến khi tìm được giá trị đúng.

Do hệ thống hoàn toàn dựa trên mật khẩu để xác thực người dùng, nếu bản thân các mật khẩu bị đánh cắp thì khả năng tấn công hệ thống là không có giới hạn. Điều này dẫn đến một yêu cầu rất căn bản là Trung tâm phân phối khóa cần được bảo vệ nghiêm ngặt. Nếu không thì toàn bộ hệ thống sẽ trở nên mất an toàn.

Toàn vẹn dữ liệu

Đối với mỗi hệ bảo mật toàn vẹn dữ liệu là một yêu cầu không thể thiếu, để đảm bảo tính toàn vẹn dữ liệu thực sự, các thuật mã hoá như mã hoá băm, mã xác nhận thông điệp (MAC) và chữ ký điện tử có thể cùng được triển khai đồng loạt. Về cơ bản, những biện pháp này sử dụng các hàm một chiều, nghĩa là dữ liệu không thể bị giải mã ngay cả khi đã biết khoá để mã hoá nó.

3.7.2 Dịch vụ xác thực X.509

Dịch vụ xác thực X.509 là một phần của chuẩn dịch vụ thư mục CCITT X.500. Ở đây các máy chủ phân tán bảo trì cơ sở dữ liệu thông tin của người sử dụng và xác định khung cho các dịch vụ xác thực. Thư mục chứa các chứng nhận khoá công khai, khoá công khai của người sử dụng được ký bởi chủ quyền chứng nhận. Để thống nhất dịch vụ cũng xác định các thủ tục xác thực, sử dụng mã khoá công khai và chữ ký điện tử. Tuy thuật toán không chuẩn nhưng được RSA đề xuất. Các chứng nhận X.509 được sử dụng rộng rãi.

1. Các chứng nhận X.509

Được phát hành bởi Chủ quyền chứng nhận (Certification Authority – CA) bao gồm:

- Các phiên bản 1,2 hoặc 3
- Số sổ (duy nhất với CA) xác định chứng nhận
- Thuật toán xác định chữ ký
- Xuất bản tên X.500 (CA)
- Chu kỳ hiệu lực (từ-đến ngày)
- Đối tượng của tên X.500 (tên của người sở hữu)
- Đối tượng thông tin khoá công khai (thuật toán, các tham số,khoá)
- Định danh duy nhất xuất bản (phiên bản 2+)
- Định danh duy nhất đối tượng (phiên bản 2+)

- Các trường mở rộng (phiên bản 3)
- Chữ ký (hoặc hash của các trường trong chứng nhận)

Ký hiệu CA<<A>> là chứng nhận cho A được ký bởi CA

2. Nhận chứng nhận

Người sử dụng bất kỳ có thể trao đổi với CA để nhận được chứng nhận. Chỉ CA có thể sửa chứng nhận. Vì không thể bị giả mạo nên chứng nhận có thể được đặt trong thư mục công cộng.

3. Sơ đồ phân cấp CA

Nếu cả hai người sử dụng chia sẻ chung CA thì họ được giả thiết là biết khoá công khai của CA đó. Ngược lại các CA cần tạo nên sơ đồ phân cấp để trao đổi chứng nhận với nhau. Sử dụng chứng nhận liên kết các thành viên của sơ đồ để có được chứng nhận của các CA khác. Mỗi CA có thể gửi tiếp (forward) các chứng nhận của mình cho clients và có thể gửi lại (backward) chứng nhận của mình cho cha của nó. Mỗi client tin tưởng các chứng nhận của cha. Có thể kiểm chứng chứng nhận bất kỳ của một CA cho người sử dụng bằng các CA khác trong sơ đồ phân cấp.

4. Sự thu hồi chứng nhận

Giấy chứng nhận có chu kỳ sử dụng, có thể thu hồi trước thời hạn trong những trường hợp cần thiết như: khoá riêng của người sử dụng bị lộ, người dùng không tiếp tục được chứng nhận bởi CA đó, Giấy chứng nhận của CA bị làm hại. Nói chung CA bảo trì danh sách các chứng nhận bị thu hôi (CRL – Certificate Revocation List). Người sử dụng có thể kiểm tra lại các chứng nhận đã bị thu hồi.

5. Các thủ tục xác thực

X.509 bao gồm ba thủ tục xác thực tùy chọn: xác thực một chiều, xác thực hai chiều và xác thực ba chiều. Mọi thủ tục trên đều sử dụng các chữ ký khoá công khai.

Xác thực một chiều

Một chiều A->B được sử dụng để thiết lập

- Danh tính của A và rằng mẩu tin là từ A
- Mẩu tin được gửi cho B
- Tính toàn vẹn và gốc gác của mẩu tin

Mẩu tin có thể bao gồm cả nhãn thời gian, ký hiệu đặc trưng của mẩu tin (nonce), danh tính của B và nó được ký bởi A. Có thể bao gồm một số thông tin bổ sung cho B như khoá phiên.

Xác thực hai chiều

Hai mẩu tin A->B và B->A được thiết lập, ngoài mẩu tin từ A đến B như trên còn có:

- Danh tính của B và trả lời từ B
- Trả lời này dành cho A
- Tính toàn vẹn và gốc gác của trả lời

Trả lời bao gồm cả ký hiệu đặc trưng của mẩu tin (nonce) từ A, cả nhãn thời gian và ký hiệu đặc trưng trả lời từ B. Có thể bao gồm một số thông tin bổ sung cho A.

Xác thực ba chiều

Ba mẩu tin A->B, B->A và A->B được thiết lập như trên mà không có đồng hồ đồng bộ. Ngoài 2 chiều như trên còn có trả lời lại từ A đến B chứa bản sao nonce của trả lời từ B, nghĩa là các nhãn thời gian mà không cần kiểm tra.

X.509 phiên bản 3

Trong phiên bản 3 được bổ sung một số thông tin cần thiết trong giấy chứng nhận như: Email/URL, chi tiết về đợt phát hành, các ràng buộc sử dụng. Tốt hơn hết là đặt tên tường minh cho các cột mới xác định trong phương pháp mở rộng tổng quát. Các mở rộng bao gồm:

- Danh tính mở rộng
- Chỉ dẫn tính quan trọng
- Giá trị mở rộng

Các mở rộng xác thực

Khoá và các thông tin đợt phát hành

- Bao trùm thông tin về đối tượng, khoá người phát hành, chỉ thị kiểu phát hành, chứng nhận

Đối tượng chứng nhận và các thuộc tính người phát hành

- Hỗ trợ có tên phụ, định dạng phụ cho các đối tượng và người phát hành

Chứng nhận các ràng buộc phát hành

- Cho phép sử dụng các ràng buộc trong chứng nhận bởi các CA khác

Bài tập

1. Chữ ký điện tử DSA:

Cho p = 23, q = 11, h=3

Tính g

NSA A chọn khoá riêng xA = 7, tính khoá công khai của yA của A

Cho bản Hash của M là H(M) = 15

Chọn số ngẫu nhiên k = 6

Tính chữ ký điện tử của A: (r, s)

Nêu cách người nhận kiểm chứng chữ ký điện tử của A trên bản tin M.

2. Chữ ký điện tử DSA:

Cho p = 53, q = 13, h=5

Tính g

NSA A chọn khoá riêng xA = 11, tính khoá công khai của yA của A

Cho bản Hash của M là H(M) = 17

Chọn số ngẫu nhiên k = 9

Tính chữ ký điện tử của A: (r, s)

Nêu cách người nhận kiểm chứng chữ ký điện tử của A trên bản tin M.

3. Hãy cho biết các phương pháp phân phối khoá công khai. Và các cách trao đổi công khai khoá mật giữa hai người sử dụng

4. Nêu sự khác biệt giữa MAC và Hash và nêu tác dụng của chúng. Cho một số ví dụ về các hàm MAC và Hash.

5. Cho biết HMAC là gì, sử dụng chúng vào mục đích nào.

6. Nêu một số cách tạo và kiểm chứng chữ ký điện tử

7. Chứng minh "Nghịch lý Ngày sinh nhật", tức là có ít nhất 23 người, thì xác suất để có hai người trùng ngày sinh nhật sẽ lớn hơn hoặc bằng 0.5.

8. Các hàm số học và logic cơ bản nào dùng trong MD5?

9. Các hàm số học và logic cơ bản nào dùng trong SHA-1?

10. Các hàm số học và logic cơ bản nào dùng trong RIPEMD-160?

11. Trình bày hoạt động của các giao thức xác thực trên mô hình Kerberos.

12. Nêu nội dung dịch vụ xác thực X.509.

CHƯƠNG 4
MÃ KHỐI VÀ CHUẨN MÃ DỮ LIỆU DES

Mục tiêu:

- Trình bày được các chuẩn mã dữ liệu DES;
- Xác định được các thành phần của một hệ thống bảo mật.
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

3.1. Giới thiệu chung về DES

Chuẩn mã hoá dữ liệu DES được Văn phòng tiêu chuẩn của Mỹ (U.S National Bureau for Standards) công bố năm 1971 để sử dụng trong các cơ quan chính phủ liên bang. Giải thuật được phát triển tại Công ty IBM dựa trên hệ mã hoá LUCIFER của Feistel. DES là thuật toán mã hoá khối (block algrithm), với cỡ của một khối là 64 bít. Một khối 64 bít bản rõ được đưa vào, sau khi mã hoá dữ liệu đưa ra là một khối bản mã 64 bít. Cả mã hoá và giải mã đều sử dụng cùng một thuật toán và khoá.

Khoá mã có độ dài 64 bít, trong đó có 8 bít chẵn lẻ được sử dụng để kiểm soát lỗi. Các bít chẵn lẻ nằm ở các vị trí 8, 16, 24,... , 64. Tức là cứ 8 bít khoá thì trong đó có 1 bít kiểm soát lỗi, bít này qui định số bít có giá trị "1" của khối 8 bít đó theo tính bù chẵn.

Nền tảng để xây dựng khối của DES là sự kết hợp đơn giản của các kỹ thuật thay thế và hoán vị bản rõ dựa trên khoá. Đó là các vòng lặp. DES sử dụng 16 vòng lặp, nó áp dụng cùng một kiểu kết hợp của các kỹ thuật trên khối bản rõ 16 lần (Như hình vẽ) Thuật toán chỉ sử dụng các phép toán số học và lôgíc trên các số 64 bít, vì vậy nó dễ dàng thực hiện vào những năm 1970 trong điều kiện về công nghệ phần cứng lúc bấy giờ. Ban đầu, sự thực hiện các phần mềm kiểu này rất thô sơ, nhưng hiện tại thì việc đó đã tốt hơn, và với đặc tính lặp đi lặp lại của thuật toán đã tạo nên ý tưởng sử dụng chíp với mục đích đặc biệt này.

Tóm lại DES có một số đặc điểm sau:

- Sử dụng khoá 56 bít.
- Xử lý khối vào 64 bít, biến đổi khối vào thành khối ra 64 bít.
- Mã hoá và giải mã được sử dụng cùng một khoá.
- DES được thiết kế để chạy trên phần cứng.

DES thường được sử dụng để mã hoá các dòng dữ liệu mạng và mã hoá dữ liệu được lưu trữ trên đĩa.

3.2. Mô tả thuật toán

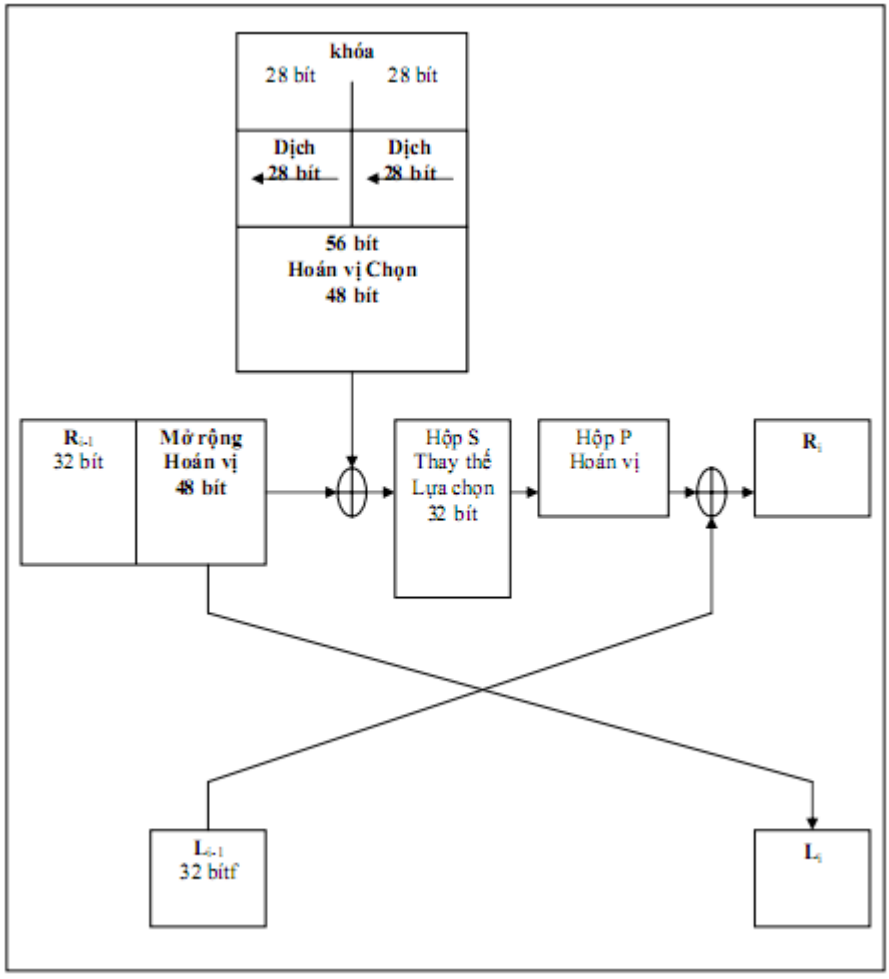
DES thực hiện trên từng khối 64 bít bản rõ. Sau khi thực hiện hoán vị khởi đầu, khối dữ liệu được chia làm hai nửa trái và phải, mỗi nửa 32 bít. Tiếp đó, có 16 vòng lặp giống hệt nhau được thực hiện, được gọi là các hàm f, trong đó dữ liệu được kết hợp với khoá. Sau 16 vòng lặp, hai nửa trái và phải được kết hợp lại và hoán vị cuối cùng (hoán vị ngược) sẽ kết thúc thuật toán.

Trong mỗi vòng lặp, các bít của khoá được dịch đi và có 48 bít được chọn ra từ 56 bít của khoá. Nửa phải của dữ liệu được mở rộng thành 48 bít bằng một phép hoán vị mở rộng, tiếp đó khối 48 bít này được kết hợp với khối 48 bít đã được thay đổi và hoán vị của khoá bằng toán tử XOR. Khối kết quả của phép tính XOR được lựa chọn ra 32 bít bằng cách sử dụng thuật toán thay thế và hoán vị lần nữa. Đó là bốn thao tác tạo nên hàm f. Tiếp đó, đầu racủa hàm f được kết hợp với nửa trái bằng một toán tử XOR. Kết quả của các bước thực hiện này trở thành nửa phải mới; nửa phải cũ trở thành nửa trái mới. Sự thực hiện này được lặp lại 16 lần, tạo thành 16 vòng của DES.

Nếu Bi là kết quả của vòng thứ i, Li và Ri là hai nửa trái và phải của Bi, Ki là khoá 48 bít của vòng thứ i, và f là hàm thực hiện thay thế, hoán vị và XOR với khoá, ta có biểu diễn của một vòng sẽ như sau:

Li=Ri-1

Ri=Li-1 XOR f(Ri-1,Ki)



3.3. Hoán vị khởi đầu

Hoán vị khởi đầu đổi chỗ khối dữ liệu vào, thay đổi vị trí của các bít trong khối dữ liệu vào, như được mô tả trong Bảng 1. Bảng này, và tất cả các bảng khác sau này, được đọc từ trái qua phải, từ trên xuống dưới. Ví dụ, hoán vị khởi đầu chuyển bít 1 thành bít 58, bít 2 thành bít 50, bít 3 thành bít 42,...

3.4. Khoá chuyển đổi

Đầu tiên, khoá 64 bít được giảm xuống thành một khoá 56 bít bằng cách bỏ qua 8 bít chẵn lẻ. Sự loại bỏ được thực hiện theo Bảng sau:

Các bít chẵn lẻ này có thể được sử dụng để đảm bảo rằng không có lỗi nào xảy ra khi đưa khoá vào. Sau khi khoá 56 bít được trích ra, một khoá khác 48 bít được sinh ra cho mỗi vòng của DES. Những khoá này, ki, được xác định bằng cách:

+ Đầu tiên, khoá 56 bít được chia làm hai phần mỗi phần 28 bít. Sau đó, các phần này được dịch trái một hoặc hai bít, phụ thuộc vào vòng đó.

+ Sau khi được dịch, 48 bít được lựa chọn ra từ 56 bít. Bởi vì sự thực hiện này đổi chỗ thứ tự các bít như là sự lựa chọn một tập con các bít, nó được gọi là hoán vị nén (compression permutation), hoặc hoán vị lựa chọn (permuted choice). Sự thực hiện này cung cấp một tập hợp các bít cùng cỡ với đầu ra của hoán vị mở rộng. Bảng 4 định nghĩa hoán vị nén (cũng gọi là hoán vị lựa chọn). Ví dụ, bít ở vị trí 33 của khoá dịch được chuyển tới vị trí 35 của đầu ra, và bít ở vị trí 18 của khoá dịch bị bỏ qua.

3.5. Hoán vị mở rộng

Ở thao tác này, nửa phải của dữ liệu, Ri, được mở rộng từ 32 bít thành 48 bít. Bởi vì sự thực hiện này thay đổi thứ tự của các bít bằng cách lặp lại một bít nào đó, nó được hiểu như là một sự hoán vị mở rộng. Sự thực hiện này nhằm mục đích tạo ra kết quả là dữ liệu cùng cỡ với khoá để thực hiện thao tác XOR. Định nghĩa hoán vị mở rộng - hộp E. Với mỗi bộ 4 bít của khối dữ liệu vào, bít đầu tiên và bít thứ tư mỗi bít tương ứng với 2 bít của khối dữ liệu ra, trong khi bít thứ hai và bít thứ ba mỗi bít tương ứng với một bít của khối dữ liệu ra. Bảng dưới mô tả vị trí của các bít trong khối dữ liệu ra theo khối dữ liệu vào. Ví dụ, bít ở vị trí thứ 3 của khối dữ liệu vào được chuyển tới vị trí thứ 4 trong khối dữ liệu ra. Và bít ở vị trí 21 của khối dữ liệu vào được chuyển tới vị trí 30 và 32 trong khối dữ liệu ra.

3.6. Hộp thay thế S

Sau khi được nén, khoá được XOR với khối mở rộng, 48 bít kết quả được chuyển sang giai đoạn thay thế. Sự thay thế được thực hiện bởi 8 hộp thay thế (substitution boxes, S-boxes). Khối 48 bít được chia thành 8 khối 6 bít. Mỗi khối được thực hiện trên một hộp S riêng biệt (separate S-box): khối được thực hiện trên hộp S1, khối 2 được thực hiện trên hộp S2,... , khối 8 được thực hiện trên hộp S8.

Mỗi hộp S là một bảng gồm 4 hàng và 16 cột. Mỗi phần tử của hộp là một số 4 bít. Sáu bít vào hộp S sẽ xác định số hàng và số cột để tìm kết quả ra. Bảng 6 biểu diễn 8 hộp S.

Những bít vào xác định một phần tử trong hộp S một cách riêng biệt.

Sáu bít vào của hộp được ký hiệu là b1, b2, b3, b4, b5 và b6. Bít b1 và b6 được kết hợp thành một số 2 bít, nhận giá trị từ 0 đến 3, tương ứng với một hàng trong bảng. Bốn bít ở giữa, từ b2 tới b5, được kết hợp thành một số 4 bít, nhận giá trị từ 0 đến 15, tương ứng với một cột trong bảng.

Ví dụ, giả sử ta đưa dữ liệu vào hộp S thứ 6 (bít 31 tới bít 36 của hàm XOR) là 110010. Bít đầu tiên và bít cuối cùng kết hợp thành 10, tương ứng với hàng thứ 3 của hộp S thứ 6. Bốn bít giữa kết hợp thành 1001, tương ứng với cột thứ 10 của hộp S thứ 6. Phần tử hàng 3 cột 9 của hộp S thứ 6 là 0. Giá trị 0000 được thay thế cho 110010.

Kết quả của sự thay thế là 8 khối 4 bít, và chúng được kết hợp lại thành một khối 32 bít. Khối này được chuyển tới bước tiếp theo: hộp hoán vị P (P- box permutation).

3.7. Hộp hoán vị P

Khối dữ liệu 32 bít ra của hộp thay thế S được hoán vị tiếp trong hộp P. Sự hoán vị này ánh xạ mỗi bít dữ liệu vào tới một vị trí trong khối dữ liệu ra; không bít nào được sử dụng hai lần và cũng không bít nào bị bỏ qua. Nó đượcgọi là hoán vị trực tiếp (straight permutation). Bảng hoán vị cho ta vị trí của mỗi bí cần chuyển. Ví dụ, bít 4 chuyển tới bít 21, trong khi bít 32 chuyển tới bít 4.

Cuối cùng, kết quả của hộp hoá vị P được XOR với nửa trái của khối 64 bít khởi đầu. Sau đó, nửa trái và phải được chuyển đổi cho nhau và một vòng mới được tiếp tục.

3.8. Hoán vị cuối cùng

Hoán vị cuối cùng là nghịch đảo của hoán vị khởi đầu, và nó được mô tả trong bảng dưới. Chú ý rằng nửa trái và nửa phải không được tráo đổi sau vòng cuối cùng của DES; thay vào đó khối nối R16L16được sử dụng như khối dữ liệu ra của hoán vị cuối cùng. Không có gì đưa ra ở đây; tráo đổi các nửa và dịch vòng hoán vị sẽ cho chính xác như kết quả trước; điều đó có nghĩa là thuật toán có thể được sử dụng cho cả mã hoá và giải mã.

Bảng hoán vị cuối cùng:

40 8 48 16 56 24 64 32 39 7 47 15 55 23 63 31

38 6 46 14 54 22 62 30 37 5 45 13 53 21 61 29

36 4 44 12 52 20 60 28 35 3 43 11 51 19 59 27

34 2 42 10 50 18 58 26 33 1 41 9 49 17 57 25

3.9. Giải mã DES

Sau khi thay đổi, hoán vị, XOR, và dịch vòng, chúng ta có thể nghĩ rằng thuật toán giải mã phức tạp, khó hiểu như thuật toán mã hoá và hoàn toàn khác thuật toán mã hoá. Trái lại, sự hoạt động được lựa chọn để đưa ra một đặc tính hữu ích: cùng thuật toán làm việc cho cả mã hoá và giải mã.

Với DES, có thể sử dụng cùng chức năng để giải mã hoặc mã hoá một khối. Chỉ có sự khác nhau đó là các khoá phải được sử dụng theo thứ tự ngược lại. Nghĩa là, nếu các khoá mã hoá cho mỗi vòng là k1, k2, k3 ,... , k15, k16 thì các khoá giải là k16, k15,... , k3, k2, k1. Giải thuật để tổng hợp khoá cho mỗi vòng cũng tương tự. Có khác là các khoá được dịch phải và số vị trí bit để dịch được lấy theo chiều ngược lại.

3.10. Phần cứng và phần mềm thực hiện DES

Việc mô tả DES khá dài dòng song việc thực hiện DES rất hữu hiệu bằng cả phần cứng lẫn phần mềm. Các phép tính số học duy nhất được thực hiện là phép XOR các xâu bít. Hàm mở rộng E, các hộp S, các hoán vị khởi đầu IP, hoán vị cuối cùng IP-1 và việc tính toán các khoá k1, k2,... , k16đều có thể thực hiện được cùng lúc bằng tra bảng (trong phần mềm) hoặc bằng cách nối cứng chúng thành mạch.

Một phần mềm DES trên máy tính lớn IBM 3090 có thể thực hiện32.000 phép tính mã hoá trong một giây. Với máy vi tính thì tốc độ thấp hơn.

(Chú ý : Phần mềm này được viết trên C và Assembler, và có thể mua được từ Utimaco-Belgium, Interleuvenlaan 62A, B-300 leuven, Belgium. Cỡ mã xấp xỉ 64K. ANSI C thực hiện chậm hơn khoảng 20%.)

Một ứng dụng rất quan trọng của DES là trong giao dịch ngân hàng Mỹ. DES được dùng để mã hoá các số định danh các nhân (PIN) và việc chuyển tài khoản được thực hiện bằng máy thủ quỹ tự động (ATM). DES còn được sử dụng rộng dãi trong các tổ chức chính phủ.

3.11. Sự an toàn của DES

Đã có rất nhiều sự nghiên cứu về độ dài của khoá, số vòng lặp, và thiết kế của hộp S (S-boxes). Hộp S có đặc điểm là khó hiểu, không có bất cứ sự rõ ang nào như tại sao chúng phải như vậy. Mọi tính toán trong DES ngoại trừ

các hộp S đều tuyến tính, tức việc tính XOR của hai đầu ra cũng giống như phép XOR hai đầu vào rồi tính toán đầu ra. Các hộp S chứa đựng thành phần phi tuyến của hệ là yếu tố quan trọng nhất đối với sự an toàn của hệ thống.

Tính bảo mật của một hệ mã hoá đối xứng là một hàm hai tham số: độ phức tạp của thuật toán và độ dài của khoá.

Giả sử rằng tính bảo mật chỉ phụ thuộc vào độ phức tạp của thuật toán. Có nghĩa rằng sẽ không có phương pháp nào để phá vỡ hệ thống mật mã hơn là cố gắng thử mọi khoá có thể, phương pháp đó được gọi là brute-force attack. Nếu khoá có độ dài 8 bít, suy ra sẽ có 28=256 khoá. Vì vậy, sẽ mất nhiều nhất 256 lần thử để tìm ra khoá đúng. Nếu khoá có độ dài 56 bít, thì sẽ có 256 khoá có thể sử dụng. Giả sử một Suppercomputer có thể thử một triệu khoá trong một giây, thì nó sẽ cần 2000 năm để tìm ra khoá đúng. Nếu khoá có độ dài 64 bít, thì với chiếc máy trên sẽ cần 600,000 năm để tìm ra khoá đúng trong số 264 khoá. Nếu khoá có độ dài 128 bít, thì sẽ mất 1025 năm để tìm ra khoá đúng. Vũ trụ chỉ mới tồn tại 1010 năm, vì vậy 1025 thì một thời gian quá dài. Với một khoá 2048 bít, một máy tính song song thực hiện hàng tỉ tỉ phép thử trong một giây sẽ tiêu tốn một khoảng thời gian là 10597 năm để tìm ra khoá. Lúc đố vũ trụ có lẽ không còn tồn tại nữa.

Khi IBM đưa ra thiết kế đầu tiên của hệ mã hoá LUCIFER, nó có khoá dài 128 bít. Ngày nay, DES đã trở thành một chuẩn về mã hoá dữ liệu sử dụng khoá 56 bít, tức kích thước không gian khoá là 256. Rất nhiều nhà mã hoá hiện đang tranh luận về một khoá dài hơn của DES. Nhiều thiết bị chuyên dụng đã được đề xuất nhằm phục vụ cho việc tấn công DES với bản rõ đã biết. Sự tấn công này chủ yếu thực hiện tìm khoá theo phương pháp vét cạn.

Tức với bản rõ X 64 bít và bản mã Y tương ứng, mỗi khoá có thể đều được kiểm tra cho tới khi tìm được một khoá k thoả mãn Ek(X)=Y (có thể có nhiều hơn một khoá k như vậy).

Vào năm 1979, Diffie và Hellman tuyên bố rằng với một máy tính chuyên dụng bản mã hoá DES có thể được phá bằng cách thử mọi trường hợp của khoá trong vòng một ngày – giá của máy tính đó là 20 triệu đôla. Vào năm 1981, Diffie đã tăng lên là cần hai ngày để tìm kiếm và giá của chiếc máy tính đó là 50 triệu đôla.

3.12. Tranh luận về DES.

Khi DES được đề xuất như một chuẩn mật mã, đã có rất nhiều ý kiến phê phán. Một lý do phản đối DES có liên quan đến các hộp S. Mọi tính toán liên quan đến DES ngoại trừ các hộp S đều tuyến tính, tức việc tính phép hoặc loại trừ của hai đầu ra cũng giống như phép hoặc loại trừ của hai đầu vào rồi tính tóan đầu ra. Các hộp S – chứa đựng thành phần phi tuyến của hệ mật là yếu tố quan trong nhất đối với độ mật của hệ thống(Ta đã thấy trong chương 1 là các hệ mật tuyến tính – chẳng hạn như Hill – có thể dễ dàng bị

mã thám khi bị tấn công bằng bản rõ đã biết). Tuy nhiên tiêu chuẩn xây dựng các hộp S không được biết đầy đủ. Một số người đã gợi ý là các hộp S phải chứa các "cửa sập" được dấu kín, cho phép Cục An ninh Quốc gia Mỹ (NSA) giải mã được các thông báo nhưng vẫn giữ được mức độ an toàn của DES. Dĩ nhiên ta không thể bác bỏ được khẳng định này, tuy nhiên không có một chứng cớ nào được đưa ra để chứng tỏ rằng trong thực tế có các cửa sập như vậy.

Năm 1976 NSA đã khẳng định rằng, các tính chất sau của hộp S là tiêu chuẩn thiết kế:

P0 Mỗi hàng trong mỗi hộp S là một hoán vị của các số nguyên 0, 1, . . . , 15. P1 Không một hộp S nào là một hàm Affine hoặc tuyến tính các đầu vào của nó. P2 Việc thay đổi một bít vào của S phải tạo nên sự thay đổi ít nhất là hai bít ra. P3 Đối với hộp S bất kì và với đầu vào x bất kì S(x) và S(x 001100) phải khác nhau tối thiểu là hai bít (trong đó x là xâu bít độ dài 6).

Hai tính chất khác nhau sau đây của các hộp S có thể coi là được rút ra từ tiêu chuẩn thiết kế của NSA. P4 Với hộp S bất kì, đầu vào x bất kì và với e, f {0,1}: S(x) ≠S(x 11ef00). P5 Với hộp S bất kì , nếu cố định một bít vào và xem xét giá trị của một bít đầu ra cố định thì các mẫu vào để bít ra này bằng 0 sẽ xấp xỉ bằng số mẫu ra để bít đó bằng 1.(Chú ý rằng, nếu cố định giá trị bít vào thứ nhất hoặc bít vào thứ 6 thì có 16 mẫu vào làm cho một bít ra cụ thể bằng 0 và có 16 mẫu vào làm cho bít này bằng 1. Với các bít vào từ bít thứ hai đến bít thứ 5 thì điều này không còn đúng nữa. Tuy nhiên phân bố kết quả vẫn gần với phân bố đều. Chính xác hơn, với một hộp S bất kì, nếu ta cố định giá trị của một bít vào bất kì thì số mẫu vào làm cho một bít ra cố định nào đó có giá trị 0 (hoặc 1) luôn nằm trong khoảng từ 13 đến 19.

Người ta không biết rõ là liệu có còn một chuẩn thiết kế nào đầy đủ hơn được dùng trong việc xây dựng hộp S hay không. Sự phản đối xác đáng nhất về DES chính là kích thước của không gian khoá: 256 là quá nhỏ để đảm bảo an toàn thực sự. Nhiều thiết bi chuyên dụngđã được đè xuất nhằm phục vụ cho việc tấn công với bản rõ đã biết. Phép tấn công này chủ yếu thực hiện tìm khoá theo phương pháp vét cạn. Tức với bản rõ x 64 bít và bản mã y tương ứng, mỗi khoá đều có thể được kiểm tra cho tới khi tìm được một khoá K thảo mãn eK(x) = y. Cần chú ý là có thể có nhiều hơn một khoá K như vậy).

Ngay từ năm 1977, Diffie và Hellman đã gợi ý rằng có thể xây dựng một chíp VLSI (mạch tích hợp mật độ lớn) có khả năng kiểm tra được 106khoá/giây. Một máy có thể tìm toàn bộ không gian khoá cỡ 106 trong khoảng 1 ngày. Họ ước tính chi phí để tạo một máy như vậy khoảng 2.107$.

Trong cuộc hội thảo tại hội nghị CRYPTO'93, Michael Wiener đã đưa ra một thiết kế rất cụ thể về máy tìm khoá. Máy này có khả năng thực hiện

đồng thời 16 phép mã và tốc độ tới 5×107 khoá/giây. Với công nghệ hiện nay, chi phí chế tạo khoảng 10,5$/khung. Giá của một khung máy chứa 5760 chíp vào khoảng 100.000$ và như vậy nó có khả năng tìm ra một khoá của DES trong khoảng 1,5 ngày. Một thiết bị khung 10 khung máy như vậy có giá chừng 106 $ sẽ giảm thời gian tìm kiếm khoá trng bình xuống còn 3,5 giờ.

3.13. DES trong thực tế.

Mặc dù việc mô tả DES khá dài dòng song người ta có thể thực hiện DES rất hữa hiệu bằng cả phần cứng lẫn phần mền. Các phép toán duy nhất cần được thực hiện là phép hoặc loại trừ các xâu bít. Hàm mở rộng E, các hộp S, các hoán vị IP và P và việc tính toán các giá tri K1,.. . ,K16đều có thể thực hiện được cùng lúc bằng tra bảng (trong phần mền) hoặc bằng cách nối cứng chúng thành một mạch.

Các ứng dụng phần cứng hiện thời có thể đạt được tốc độ mã hoá cực nhanh. Công ty Digital Equipment đã thông báo tại hội nghị CRUPTO'92 rằng họ sẽ chế tạo một xung có 50 ngàn xung có thể mã hoá với tốc độ 1 Gbít/s bằng cách xung nhịp có tốc độ 250MHz. Giá của xung này vào khoảng 300$. Tới năm 1991 đã có 45 ứng dụng phần cứng và chương trình cơ sở của

DES được Uỷ ban tiêu Chuẩn quốc gia Mỹ (NBS) chấp thuận. Một ứng dụng quan trọng của DES là trong giao dịch ngân hàng Mỹ - (ABA) DES được dùng để mã hoá các số định danh cá nhân (PIN) và việc chuyển tài khoản bằng máy thủ quỹ tự động (ATM). DES cũng được Hệ thống chi trả giữa các nhà băng của Ngân hàng hối đoái (CHIPS) dùng để xác

thực các giao dịch vào khoản trên 1,5×1012 USA/tuần. DES còn được sử dụng rộng rãi trong các tổ chức chính phủ. Chẳng hạn như bộ năng lượng, Bộ Tư pháp và Hệ thống dự trữ liên bang.

3.14. Các chế độ hoạt động của DES.

Có 4 chế độ làm việc đã được phát triển cho DES: Chế độ chuyển mã điện tử (ECB), chế độ phản hồi mã (CFB), chế độ liên kết khối mã (CBC) và chế độ phản hồi đầu ra (OFB). Chế độ ECB tương ứng với cách dùng thông thường của mã khối: với một dãy các khối bản rõ cho trước x1,x2,. . .(mỗi khối có 64 bít), mỗi xi sẽ được mã hoá bằng cùng một khoá K để tạo thành một chuỗi các khối bản mã y1y2 ... theo quy tắc yi = eK(yi-1 xi) i ≥ 1.

Trong các chế độ OFB và CFB dòng khoá được tạo ra sẽ được cộng mod 2 với bản rõ (tức là nó hoạt động như một hệ mã dòng, xem phần 1.1.7). OFB thực sự là một hệ mã dòng đồng bộ: dòng khoá được tạo bởi việc mã lặp véc tơ khởi tạo 64 bít (véc tơ IV). Ta xác định z0 =IV và rồi tính dòng khoá z1z2 . . . theo quy tắc zi = eK(zi-1), i≥1. Dãy bản rõ x1x2 . . . sau đó sẽ được mã hoá bằng cách tính yi = xi zi,i ≥1.

Trong chế độ CFB, ta bắt đầu với y0 = IV (là một véc tơ khởi tạo 64 bít) và tạo phần tử zi của dòng khoá bằng cách mã hoá khối bản mã trước đó.

Tức zi = eK(yi-1), i ≥1. Cũng như trong chế độ OFB: yi = xi⊕ zi,i ≥1. Việc sử dụng CFB đượcc mô tả trên hình 3.5 (chú ý rằng hàm mã DES eKđượcc dùng cho cả phép mã và phép giải mã ở các chế độ CFB và OFB).

Cũng còn một số biến tấu của OFB và CFB được gọi là các chế độ phản hồi K bít (1 < K < 64). ở đây ta đã mô tả các chế độ phản hồi 64 bít. Các chế độ phản hồi 1 bít và 8 bít thường được dùng trong thực tế cho phép mã hoá đồng thời 1 bit (hoặc byte) số liệu.

Bốn chế độ công tác có những ưu, nhược điểm khác nhau. Ở chế độ ECB và OFB, sự thay đổi của một khối bản rõ xi 64 bít sẽ làm thay đổi khối bản mã yi tương ứng, nhưng các khối bản mã khác không bị ảnh hưởng.

Trong một số tình huống đây là một tính chất đáng mong muốn. Ví dụ, chế độ OFB thường được dùng để mã khi truyền vệ tinh. Mặt khác ở các chế độ CBC và CFB, nếu một khối bản rõ xi bị thay đổi thì yi và tất cả các khối bản mã tiếp theo sẽ bi ảnh hưởng. Như vậy các chế độ CBC và CFB có thể được sử dụng rất hiệu quả cho mục đích xác thực. Đặc biệt hơn, các chế độ này có thể được dùng để tạo mã xác thực bản tin (MAC - message authentication code). MAC được gắn thêm vào các khối bản rõ để thuyết phục Bob tin rằng, dãy bản rõ đó thực sự là của Alice mà không bị Oscar giả mạo. Như vậy MAC đảm bảo tính toàn vẹn (hay tính xác thực) của một bản tin (nhưng tất nhiên là MAC không đảm bảo độ mật).

Ta sẽ mô tả cáchb sử dụng chế độ BCB để tạo ra một MAC. Ta bắt đầu bằng véc tơ khởi tạ IV chứa toàn số 0. Sau đó dùng chế đô CBC để tạo các khối bản mã y1,. . . ,yn theo khoá K. Cuối cùng ta xác định MAC là yn. Alice sẽ phát đi dãy các khối bản rõ x1,x2,. . . ,xn cùng với MAC. Khi Bob thu được x1. . .xn anh ta sẽ khôi phục lại y1. . .yn bằng khoá K bí mật và xác minh xem liệu yn có giống với MAC mà mình đã thu được hay không.

Nhận thấy Oscar không thể tạo ra một MAC hợp lệ do anh ta không biết khoá K mà Alice và Bob đang dùng. Hơn nữa Oscar thu chặn được dãy khối bản rõ x1. . .xn và thay đổi ít nhiều nội dung thì thì chắc chắn là Oscar không thể thay đổi MAC để được Bob chấp nhận.

Thông thường ta muốn kết hợp cả tính xác thực lẫn độ bảo mật. Điều đó có thể thực hiện như sau: Trước tiên Alice dùng khoá K1để tạo MAC cho x1. . . xn . Sau đó Alice xác định xn+1 là MAC rồi mã hoá dãy x1. . .xn+1 bằng khoá thứ hai K2để tạo ra bản mã y1. . .yn+1 . Khi Bob thu được y1. . .yn+1, trước tiên Bob sẽ giải mã (bằng K2) và kiểm tra xem xn+1 có phải là MAC đối với dãy x1. . .xn dùng K1 hay không.

Ngược lại, Alice có thể dùng K1để mã hoá x1. . .xn và tạo ra được 1...yn, sau đó dùng K2để tạo MAC yn+1đối với dãy y1. . .yn. Bob sẽ dùng K2 để xác minh MAC và dung K1để giải mã y1. . .yn.

CHƯƠNG 5
PHÁT HIỆN XÂM NHẬP

Mục tiêu:

- Trình bày được các hình thức tấn công vào hệ thống mạng;
- Xác định được các thành phần của một hệ thống bảo mật.
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

5.1 Kẻ xâm nhập

5.1.1 Khái niệm

Vấn đề quan trọng đối với hệ thống mạng là chống lại việc truy cập không mong muốn qua mạng máy tính lớn hoặc cục bộ. Chúng ta có thể phân loại kẻ xâm nhập như sau:

- o Kẻ giả danh
- o Kẻ lạm quyền
- o Người sử dụng giấu mặt

Có nhiều mức độ khả năng khác nhau xâm nhập khác nhau. Rõ ràng vấn đề trên được công khai và trở nên bức xúc

- o Từ Wily Hacker" trong năm 1986/1987
- o đến việc tăng nhanh các đội ứng cứu tình trạng khẩn cấp của máy tính

Với đội ứng cứu có thể cảm thấy bình an nhưng đòi hỏi các nguồn chi bổ sung để phát triển và duy trì hoạt động. Kẻ xâm nhập có thể sử dụng các hệ thống làm hại để tấn công.

5.1.2 Các kỹ thuật xâm phạm

Mục tiêu của kẻ xâm nhập là dành quyền truy cập hoặc tăng quyền trong hệ thống. Các phương pháp tấn công cơ bản bao gồm

- o Tìm mục tiêu và thu thập thông tin
- o Truy cập ban đầu
- o Leo thang quyền
- o Lần vết khôi phục

Mục tiêu chính là giành được mật khẩu và sau đó dùng quyền truy cập của người sở hữu.

5.1.3 Đoán mật khẩu

Đoán mật khẩu là một trong các hướng tấn công chung nhất. Kẻ tấn công đã biết tên người sử dụng đăng nhập (từ trang email/web) và tìm cách đoán mật khẩu.

- Mặc định, mật khẩu ngắn, tìm kiếm các từ chung
- Thông tin của người dùng (thay đổi tên, ngày sinh, số điện thoại, các mối quan tâm và từ chung)
- Tìm kiếm tổng thể mọi khả năng của mật khẩu

Kẻ xâm nhập kiểm tra đăng nhập với tệp mật khẩu đánh cắp được. Sự thành công của việc đoán mật khẩu phụ thuộc vào mật khẩu được chọn bởi người dùng. Tổng quan chỉ ra rằng nhiều người sử dụng chọn mật khẩu không cẩn thận.

Nắm bắt mật khẩu

Tấn công khác bao gồm nắm bắt mật khẩu

- Theo dõi qua vai khi nhập password
- Sử dụng chương trình ngựa thành Toroa để thu thập
- Theo dõi login mạng không an toàn, chẳng hạn Telnet, FTP, Web, email.
- Chắt lọc thông tin ghi lại được sau lần vào mạng thành công (đệm/lịch sử web, số quay cuối,…)
- Sử dụng login/password đúng để nhại lại người sử dụng

Người sử dụng cần được học để dùng các biện pháp đề phòng và ngăn ngừa thích hợp.

5.1.4 Phát hiện xâm nhập

Chắc chắn có lỗi an toàn ở đâu đó. Như vậy để phát hiện xâm nhập cần phải

- Chia khối để phát hiện nhanh
- Hành động ngăn chặn
- Thu thập thông tin để tăng cường an toàn

Giả thiết rằng kẻ xâm nhập sẽ hành động khác so với người dùng hợp pháp

- Nhưng sẽ có sự khác biệt nhỏ giữa họ

a. Các cách tiếp cận phát hiện xâm nhập

Phát hiện thống kê bất thường

- Vượt qua ngưỡng thống kê nào đó
- Dựa trên mô tả

Dựa trên qui tắc

- Hành động bất thường
- Định danh thâm nhập

b. Kiểm tra các bản ghi

Công cụ cơ bản để phát hiện xâm nhập là kiểm tra bản ghi đơn giản

- Một phần của hệ điều hành đa người sử dụng
- Sẵn sàng để sử dụng
- Có thể không có thông tin trong định dạng mong muốn

Tiến hành kiểm tra các bản ghi chuyên dùng để phát hiện

- Được tạo chuyên dùng để thu thập một số thông tin mong muốn
- Trả giá chi phí bổ sung trong hệ thống

c. Phát hiện thống kê bất thường

Phát hiện ngưỡng

- Đếm sự xuất hiện của sự kiện đặc biệt theo thời gian
- Nếu vượt quá giá trị nào đó thì cho là đã có xâm nhập
- Nếu chỉ dùng nó thì đây là phát hiện thô không hiệu quả

Dựa trên mô tả

- Đặc trưng hành vi quá khứ của người sử dụng
- Phát hiện hệ quả quan trọng từ đó
- Mô tả bằng nhiều tham số

d. Phân tích kiểm tra bản ghi

Đây là cơ sở của cách tiếp cận thống kê. Phân tích bản ghi để nhận được các số đo theo thời gian

- Số đếm, đo, thời gian khoảng, sử dụng nguồn

Sử dụng các kiểm tra khác nhau trên số liệu phân tích để xác định hành vi hiện tại có chấp nhận được không

- Tính kỳ vọng, phương sai, biến nhiều chiều, quá trình Markov, chuỗi thời gian, thao tác

Ưu điểm chính là không sử dụng kiến thức biết trước

e. Phát hiện xâm nhập dựa trên qui tắc

Quan sát các sự kiện trong hệ thống và áp dụng các qui tắc để quyết định hoạt động đó có đáng nghi ngờ hay không. Phát hiện bất thường dựa trên qui tắc

- Phân tích các bản ghi kiểm tra cũ để xác định mẫu sử dụng và qui tắc tự sinh cho chúng
- Sau đó quan sát hành vi hiện tại và sánh với các qui tắc để nhận thấy nếu nó phù hợp
- Giống như phát hiện thống kê bất thường không đòi hỏi kiến thức biết trước về sai lầm an toàn

Định danh sự thâm nhập dựa vào qui tắc

- Sử dụng công nghệ hệ chuyên gia
- Với qui tắc định danh sự xâm nhập đã biết, các mẫu điểm yếu, hoặc các hành vi nghi ngờ
- So sánh các bản ghi kiểm tra hoặc các trạng thái theo qui tắc
- Qui tắc được sinh bởi các chuyên gia những người đã phỏng vấn và hệ thống kiến thức của các quản trị an toàn
- Chất lượng phụ thuộc vào cách thức thực hiện các điều trên

Ảo tưởng dựa trên tỷ lệ

- Thực tế phát hiện xâm nhập hệ thống cần phát hiện tỷ lệ xâm nhập đúng với rất ít cảnh báo sai
  - Nếu rất ít sự xâm nhập được phát hiện -> an toàn không tốt
  - Nếu rất nhiều cảnh báo sai -> bỏ qua/phí thời gian
- Điều đó rất khó thực hiện
- Các hệ thống tồn tại hình như không có các bản ghi tốt

Phát hiện xâm nhập phân tán

- Truyền thống thường tập trung hệ thống đơn lẻ
- Nhưng thông thường có các hệ thống máy tính
- Bảo vệ hiệu quả cần làm việc cùng nhau để phát hiện xâm nhập
- Các vấn đề
  - Làm việc với nhiều định dạng bản ghi kiểm tra khác nhau
  - Toàn vẹn và bảo mật dữ liệu trên mạng
  - Kiến trúc tập trung và phân tán

Sử dụng bình mật ong

- Chăng lưới thu hút các kẻ tấn công

§ Tách khỏi sự truy cập đến các hệ thống then chốt

§ Để thu thập các thông tin về hoạt động của chúng

§ Kích thích kẻ tấn công ở lại trong hệ thống để người quản trị có thể phán đoán

o Được cấp đầy đủ các thông tin bịa đặt

o Được trang bị để thu thập chi tiết thông tin về hoạt động của kẻ tấn công

o Hệ thống mạng đơn và lặp

5.1.5 Quản trị mật khẩu

o Là bảo vệ tuyến đầu chống kẻ xâm nhập

o Người sử dụng được cung cấp cả hai:

§ Login – xác định đặc quyền của người sử dụng

§ Password – xác định danh tính của họ

o Passwords thường được lưu trữ mã hoá

§ Unix sử dụng DES lặp

§ Các hệ thống gần đây sử dụng hàm hash

o Cần phải bảo vệ file passwords trong hệ thống

Tìm hiểu về mật khẩu

o Purdue 1992 – có nhiều mật khẩu ngắn

o Klein 1990 – có nhiều mật khẩu đoán được

o Kết luận là người sử dụng thường chọn các mật khẩu không tốt

o Cần một cách tiếp cận để chống lại điều đó

Tạo mật khẩu - cần giáo dục cách tạo mật khẩu

o Cần có chính sách và giáo dục người sử dụng

o Giáo dục tầm quan trọng của mật khẩu tốt

o Cho định hướng mật khẩu tốt

§ độ dài tối thiểu > 6

§ đòi hỏi trộn chữ hoa và chữ thường, số và dấu chấm

§ không chọn từ trong từ điển

o Nhưng nên chọn sao cho nhiều người không để ý

Tạo mật khẩu – máy tính tự sinh

o Cho máy tính tự tạo mật khẩu

- Nếu ngẫu nhiên không dễ nhớ, thì sẽ viết xuống (hội chứng nhãn khó chịu)
- Ngay cả phát âm được cũng không nhớ
- Có câu chuyện về việc chấp nhận của người sử dụng tồi
- FIPS PUB 181 là một trong những bộ sinh tốt nhất
  - Có cả mô tả và code ví dụ
  - Sinh từ việc ghép ngẫu nhiên các âm tiết phát âm được

Tạo mật khẩu - kiểm tra trước

- Cách tiếp cận hứa hẹn nhất để có thể cải thiện an toàn mật khẩu
- Cho phép người sử dụng chọn trước mật khẩu của mình
- Nhưng để cho hệ thống kiểm chứng xem nó có chấp nhận được không
  - Bắt buộc theo qui tắc đơn giản
  - So sánh với từ điển các mật khẩu tồi
  - Sử dụng mô hình thuật toán Markov hoặc bộ lọc để chống các cách chọn tồi

5.2 Phần mềm có hại

5.2.1 Các kiểu phần mềm có hại khác ngoài Virus

Virus máy tính đã được công bố rất nhiều, là một trong những phần mềm có hại. Tác động của nó mọi người đều biết, đã được nêu trong các báo cáo, viễn tưởng và phim ảnh, gây nhiều chú ý hơn là tán thưởng và được quan tâm nhiều để phòng chống.

5.2.21. Cửa sau hoặc cửa sập

Điểm vào chương trình bí mật, cho phép những người biết truy cập mà bỏ qua các thủ tục an toàn thông thường. Kỹ thuật này có thể được sử dụng chung bởi những người phát triển và là mối đe doạ khi để trong chương trình sản phẩm cho phép khai thác bởi các kẻ tấn công. Rất khó ngăn chặn trong hệ điều hành, đòi hỏi sự phát triển và cập nhật phần mềm tốt.

5.2.3 Bom logic

Đây là một trong những phần mềm có hại kiểu cổ, code được nhúng trong chương trình hợp pháp. Nó được kích hoạt khi gặp điều kiện xác định

- Có mặt hoặc vắng mặt một số file

- Ngày tháng/thời gian cụ thể
- Người sử dụng nào đó

Khi được kích hoạt thông thường nó làm hỏng hệ thống

- Biến đổi/xoá file/đĩa, làm dừng máy,…

5.2.4. Ngựa thành Tơ roa

Chương trình với các tác động phụ được dấu kín, mà thông thường rất hấp dẫn như trò chơi hoặc phần mềm nâng cấp. Khi chạy thực hiện những nhiệm vụ bổ sung, cho phép kẻ tấn công gián tiếp dành quyền truy cập mà họ không thể trực tiếp. Thông thường sử dụng lan truyền virrus/sâu (worm) hoặc cài đặt cửa sau hoặc đơn giản phá hoại dữ liệu.

5.2.5. Zombie

Đây là chương trình bí mật điều khiển máy tính của mạng khác và sử dụng nó để gián tiếp tiến hành các tấn công. Thông thường sử dụng để khởi động tấn công từ chối các dịch vụ phân tán (DdoS). Khai thác các lỗ hổng trong các hệ thống.

5.3. Virus

Virus là đoạn code tự sinh lặp đính kèm với code khác như virus sinh học. Cả hai đều lan truyền tự nó và mang đi bộ tải

- Mang theo code để tạo các bản sao của chính nó
- Và cũng như mọi code nó cũng thực hiện nhiệm vụ ngầm nào đó

Thao tác của virus

Các giai đoạn của virus

- Nằm im - chờ sự kiện kích hoạt
- Lan truyền – lặp sinh ra chương trình/đĩa
- Kích hoạt - bởi sự kiện để thực hiện bộ tải
- Thực hiện bộ tải
- Cụ thể thông thường mang tính chất chuyên biệt của các máy và hệ điều hành. Nó khai thác các tính chất và điểm yếu

Cấu trúc Virus

```
program V :=
        {goto main;
        1234567;
        subroutine infect-executable := {loop:
                        file := get-random-executable-file;
```

```
                    if (first-line-of-file = 1234567) then goto loop
                    else prepend V to file; }
        subroutine do-damage := {whatever damage is to be done}
        subroutine trigger-pulled := {return true if condition holds}
        main: main-program :=  {infect-executable;
                    if trigger-pulled then do-damage;
                    goto next;}
        next:
}
```

Các kiểu Virus

Có thể phân loại dựa trên kiểu tấn công

- Virus ăn bám
- Virus cư trú ở bộ nhớ
- Virus ở sector khởi động
- Lén lút
- Virus nhiều hình thái
- Virus biến hoá

5.3.1. Marco Virus

Marco code đính kèm file dữ liệu, được dịch bởi chương trình sử dụng file

- Như marco của Word/Excel
- Sử dụng lệnh tự động và lệnh marco

Đây là đoạn code là độc lập với nền tảng, là đoạn nguồn chính của sự lan nhiễm virus. Có sự khác biệt không rõ ràng giữa dữ liệu và file chương trình, thông thường có sự thoả hiệp truyền thống: "dễ dàng sử dụng" và "an toàn". Đã có sự cải thiện an toàn trong Word, không trội hơn sự đe doạ của virus.

5.3.2. Virus email

Đây là loại virus lan truyền sử dụng email được đính kèm chứa marco virus như Melissa. Thường được kích hoạt khi người sử dụng mở file đính kèm hoặc ít khi hơn khi mail được xem sử dụng một tính chất script của tác

nhân mail. Do đó sẽ lan truyền rất nhanh, thông thường đích là tác nhân mail Microsoft Outlook hoặc tài liệu Word /Excel. Cần an toàn ứng dụng và hệ điều hành tốt hơn

5.3.3. Sâu

Đây là chương trình sinh lặp nhưng không có tác động, thường lan truyền trên mạng

- Như sâu Internet Morris 1988
- Dẫn đến việc tạo ra các đội ứng cứu khẩn cấp máy tính CERT
- Dùng đặc quyền phân tán hoặc khai thác các điểm yếu hệ thống
- Được sử dụng rộng rãi bởi Hackers để tạo zombie PC, kéo theo sử dụng các tấn công khác, đặc biệt từ chối dich vụ DoS

Vấn đề chính là mất sự an toàn của hệ thống kết nối thường xuyên như PC.

Thao tác của sâu

Các giai đoạn của sâu giống như virus:

- Nằm im
- Lan truyền
  - § Tìm hệ thống khác để tác động
  - § Thiết lập kết nội với hệ thống đích từ xa
  - § Tự sinh lặp mình cho hệ thống từ xa
- Kích hoạt
- Thực hiện

a. Sâu Morrris

Sâu Morris là loại sâu cổ điển, được tạo bởi Robert Morris vào 1988, nhằm tới các hệ thống Unix. Ở đây sử dụng một số kỹ thuật lan truyền, như

- Phá mật khẩu đơn giản trong file mật khẩu cục bộ
- Khai thác lỗ hổng
- Tìm lỗi cửa sập trong hệ thống mail

Mọi tấn công thành công sẽ sinh lặp nó.

b. Tấn công của sâu đương thời

Làn sóng tấn công của sâu đương thời mới từ giữa 2001 như:

- Code Red - sử dụng lỗ hổng MS IIS:

o Thử IP ngẫu nhiên cho hệ thống chạy IIS

o Có kích hoạt thời gian cho tấn công từ chối dịch vụ

o Làn sóng thứ hai tác động đến 360000 máy chủ trong vòng 14 giờ

- Code Red 2 – cài đặt cửa sập

- Nimda – cơ chế tác động lặp

- SQL Slammer – đã tấn công máy chủ MS SQL

- Sobig – đã tấn công máy chủ proxy mở

- Mydoom – sâu email có số lượng lớn và có cửa sau

c. Công nghệ sâu

Các đặc tính của công nghệ sâu là tấn công đa nền tảng, khai thác nhiều chiều, lan truyền cực nhanh, có nhiều kiểu tác động, biến hoá, cơ động và khai thác zero day.

5.3.4. Các biện pháp chống Virus

Biện pháp tốt nhất là ngăn ngừa, nhưng nói chung là không thể. Do đó cần phải có một trong nhiều biện pháp sau:

o Phát hiện virus nhiễm trong hệ thống

o Định danh loại virus nhiễm

o Loại bỏ khôi phục hệ thống về trạng thái sạch

5.3.5. Phần mềm chống Virus

Phần mềm thuộc thế hệ đầu tiên

o Quét sử dụng chữ ký của virus để định danh

o Hoặc phát hiện sự thay đổi độ dài của chương trình

Phần mềm thuộc thế hệ thứ hai

o Sử dụng các qui tắc trực quan để phát hiện nhiễm virus

o Sử dụng mã hash của chương trình để phát hiện sự thay đổi

Phần mềm thuộc thế hệ thứ ba

o Chương trình thường trú trong bộ nhớ định danh virus theo hành động

Phần mềm thuộc thế hệ thứ tư

§ Đóng gói với rất nhiều kiểu kỹ thuật chống virus

§ Quét và lần vết tích cực, kiểm soát truy cập

Phương pháp diệt bằng tay vẫn được dùng.

5.3.6. Kỹ thuật chống Virus nâng cao

Giải mã mẫu

- Sử dụng mô phỏng CPU kiểm tra chương trình, chữ ký và hành vi trước khi chạy chúng

Dùng Hệ thống miễn dịch số (IBM)

- Hành động đa muc tiêu và chống Virus
- Mọi virus nhập vào tổ chức được nắm bắt, phân tích, phát hiện/tấm chắn tạo ra chống nó và loại bỏ

Sau đây là sơ đồ Hệ miễn dịch số (Digital Immune System)

5.3.7 Phần mềm ngăn chặn hành vi

Các phần mềm này được tích hợp với hệ điều hành của máy chủ.

Chương trình theo dõi các hành vi trong thời gian thực

- Chẳng hạn truy cập file, định dạng đĩa, các chế độ thực hiện, thay đổi tham số hệ thống, truy cập mạng

Đối với các hành động có khả năng có hại

- Nếu phát hiện thì ngăn chặn, chấm dứt hoặc tìm kiếm

Có ưu điểm so với quét, nhưng code có hại chạy trước khi phát hiện.

1. Tấn công từ chối dịch vụ từ xa

Tấn công từ chối dịch vụ từ xa (DDoS) tạo thành đe dọa đáng kể, làm cho hệ thống trở nên không sẵn sàng, làm tràn bởi sự vận chuyển vô ích.

Kẻ tấn công thường sử dụng một số lớn các "zombies", tăng độ khó của các tấn công.

Công nghệ bảo vệ tìm các biện pháp đương đầu chống lại

2. Tìm hiểu cách kẻ thù xây dựng mạng lưới tấn công từ chối dịch vụ từ xa

Từ chối dịch vụ có hiệu lực khi bị nhiễm rất nhiều "zombies". Để thực hiện được điều đó cần có:

- Phần mềm cài đặt tấn công từ chối dịch vụ từ xa
- Các lỗ hổng không vá được trong nhiều hệ thống

- Chiến lược quét để tìm lỗ hổng hệ thống: sử dụng các yếu tố ngẫu nhiên, lập danh sách va chạm, tìm hiểu cấu trúc topo, mạng con cục bộ.

3. Chống tấn công từ chối dịch vụ từ xa (DDoS)

Có ba cách bảo vệ sau đây được dùng rộng rãi

- Ngăn ngừa tấn công và chiếm lĩnh trước.

- Phát hiện tấn công và lọc trong quá trình sử dụng dịch vụ

- Lần vết nguồn tấn công và xác định sự tấn công sau khi sử dụng xong dịch vụ.

Nói chung có phạm vi rộng các khả năng tấn công, vì vậy phải có nhiều biện pháp chống và sử dụng kết hợp chúng.

5.3.8 Tràn bộ đệm

Tràn bộ đệm là cơ chế tấn công rất phổ biến bắt đầu từ 1988 xuất hiện sâu Morris đến Code Red, Slammer, Sasser và nhiều cái khác nữa. Các kỹ thuật phòng chống đều đã biết. Tuy nhiên vẫn còn là vấn đề phải quan tâm vì di truyền từ các con rệp đã lây lan rộng rãi. Vì vẫn còn các kỹ thuật lập trình không cẩn thận.

Cơ sở của việc tràn bộ nhớ: sinh bởi do lỗi lập trình, cho quá nhiều dữ liệu lưu trữ hơn khả năng cho phép trong bộ đệm kích thước cố định. Bộ đệm có thể trên ngăn xếp, đống, dữ liệu tổng thể. Viết đè các vị trí nhớ cận kề, làm hỏng dữ liệu của chương trình, truyền điều khiển không mong muốn, vi phạm truy cập bộ nhớ, thực hiện code của kẻ tấn công

Xét chương trình C trên. Ở đây có ba biến, thông thường lưu trong vùng nhớ liền kề. Gọi chương trình con copy vào str1 dữ liệu Start. Sau đó đọc đầu vào sử dụng hàm gets lưu vào str2. Sau đó so sánh đầu vào với xâu Start. Nếu thành công valid = true. Vấn đề hàm thư viện gets của C không kiểm tra độ lớn dữ liệu đọc vào. Nếu nhiều hơn 7 ký tự nó đòi hỏi bộ nhớ nhiều hơn. Khi đó dữ liệu thừa viết đè dữ liệu của biến kề, trong trường hợp này là str1. Giả sử xâu đầu vào là "EVILINPUTVALUE", kết quả xâu Str1 bị viết đè với các ký tự "TVALUE". Xâu str2 không chỉ sử dụng 8 ký tự của nó mà còn thêm 7 ký tự từ str1. Biết cấu trúc trên, kẻ tấn công có thể thu xếp sao cho giá trị xâu Str1 vẫn bằng Str2. Chẳng hạn nếu nhập xâu đầu vào là "BADINPUTBADINPUT" thì trong phép so sánh kết quả vẫn đúng như trong lần chạy thứ ba trong ví dụ trên.

5.3.9. Tấn công tràn bộ nhớ.

Để làm tràn bộ đệm, kẻ tấn công cần phải phát hiện lỗ hổng tràn bộ đệm trong chương trình nào đó. Theo dõi, lần theo vết thực hiện, sử dụng các công cụ ẩn. Hiểu bộ đệm lưu trong bộ nhớ như thế nào và xác định khả năng phá hỏng

Một chút về lịch sử ngôn ngữ lập trình. Ở mức ngôn ngữ máy mọi dữ liệu là mảng các bytes, thông dịch phụ thuộc vào các chỉ lệnh được dùng. Ngôn ngữ bậc cao hiện đại có định nghĩa chặt về kiểu và các phép toán đúng. Không có lỗ hổng tràn bộ đệm, được lường trước, giới hạn khi dùng.

C và ngôn ngữ liên quan có cấu trúc điều khiển bậc cao, nhưng cho phép truy cập trực tiếp đến bộ nhớ. Vì vậy có lỗ hổng tràn bộ đệm. Có kế thừa lớn các code không an toàn, đang được sử dụng rộng rãi, nên có lỗ hổng.

Để hiểu hơn tạo sao lại tràn bộ nhớ, mà không khắc phục được, ta xem xét cơ chế mà lời gọi hàm quản lý trạng thái cục bộ cho mỗi lời gọi. Khi một hàm gọi hàm khác, nó cần phải lưu ở đâu đó địa chỉ trả về để hàm được gọi khi kết thúc trả điều khiển cho hàm gọi. Bên cạnh đó cũng cần có chỗ để cất một số tham số mà cần truyền cho hàm được gọi và cũng cần lưu các giá trị thanh ghi của hàm gọi mà cần được sử dụng khi hàm được gọi kết thúc. Thông thường mọi dữ liệu này được cất ở khung ngăn xếp (stack frame). Mỗi lần gọi hàm lại sinh ra một khung ngăn xếp liên kết.

Tràn bộ đệm ngăn xếp xảy ra khi bộ đệm đặt trên ngăn xếp. Nó được khai thác bởi sâu Morris. Bài báo "Smashing the Stack" tuyên truyền nó. Có biến cục bộ phía dưới con trỏ khung lưu trữ và địa chỉ trả lại. Vì vậy tràn bộ đệm cục bộ có nhiều khả năng viết đè các mục điều khiển chính. Kẻ tấn công viết đè địa chỉ trả về với địa chỉ của đoạn code cài vào. Đó có thể là địa chỉ của chương trình, thư viện hệ thống hoặc tải vào bộ đệm.

Chúng ta xem cấu tạo vùng nhớ, nơi cất chương trình đang chạy, dữ liệu tổng thể, đống và ngăn xếp. Khi chương trình chạy, hệ điều hành tạo ra một tiến trình cho nó. Tiến trình được cho bởi không gian ảo của riêng nó với cấu trúc như trên hình vẽ sau. Nó bao gồm nội dung chương trình đang chạy, trong đó có dữ liệu tổng thể, bảng cấp bộ nhớ, code của chương trình ở gần đáy của không gian nhớ này. Trên đó không gian cho đống tăng dần lên và trên nữa là không gian cho ngăn xếp giảm dần xuống.

Để mô tả tràn bộ đệm ngăn xếp ta xét chương trình C sau. Nó chứa biến cục bộ duy nhất, bộ đệm inp. Hàm Hello nhắc nhập tên mà được đọc vào bộ đệm nhờ hàm thư viện không an toàn gets(). Sau đó hiện kết quả đọc được nhờ hàm thư viện printf (). Nếu giá trị nhỏ được đọc thì ở đây không có vấn đề gì. Chương trình gọi hàm sẽ chạy thành công như trong lần chạy thứ nhất trên hình sau đó. Nếu dữ liệu lớn như trong lần chạy thứ hai, dữ liệu sẽ vượt qua cuối bộ đệm và ghi đè lên con trỏ khung lưu trữ và trả về địa chỉ sai tương ứng với biểu diễn nhị phân của các ký tự. Và khi hàm trả điều khiển cho địa chỉ trả về, nó nhảy đến vị trí bộ nhớ không hợp lệ, báo lỗi "Segmentation Fault" và dừng chương trình không bình thường như thông báo trên hình sau đó. Kẻ tấn công tận dụng cơ hội này để truyền điều khiển về chương trình đã định trước.

5.3.10 Code che đậy (Shellcode)

Đây là code chương trình được chuẩn bị bởi kẻ tấn công. Nó được lưu trong bộ nhớ bị tràn và kẻ tấn công tìm cách chuyển điều khiển sang cho shell.

Phát triển code che đậy

Trong ví dụ trên mô tả với Intel Linux shellcode cơ bản để chạy bản dịch Bourne shell. Shellcode cần phải phù hợp với đối số cho execve(). Nó bao gồm mọi code để gọi hàm của hệ thống, phải độc lập với vị trí và không chứa NULLs – xâu kết thúc của C.

Thông thường chương trình che đậy dưới vỏ bọc là tiện ích hệ thống đáng tin cậy, dịch vụ mạng đặc biệt, code thư viện được sử dụng chung, như hình ảnh.

Các hàm của shellcode: giao diện tự sinh, tạo đối tượng nghe để khởi tạo giao diện kết nối, tạo kết nối ngược lại tới kẻ tấn công, vượt qua các qui tắc tường lửa, thoát khỏi môi trường thực thi hạn chế.

5.3.11 Bảo vệ tràn bộ nhớ

Tràn bộ đệm được khai thác rộng rãi, có nhiều code có lỗ hổng đang dùng. Mặc dù nguyên nhân và các biện pháp chống đã biết. Có hai cách chống rộng rãi: chương trình mới được gia cố trong thời gian dịch và kiểm soát tấn công chương trình đang có trong thời gian chạy.

Nếu sử dụng ngôn ngữ bậc cao với kiểu mạnh , thì sẽ không có lỗ hổng tràn bộ đệm. Chương trình dịch buộc kiểm tra cỡ và các thao tác cho phép trên các biến. Khi đó phải trả giá khi sử dụng nguồn và hạn chế truy cập đến phần cứng. Tuy nhiên vẫn cần một số code của các ngôn ngữ giống C.

Bảo vệ trong thời gian dịch

Thiết lập các kỹ thuật lập trình an toàn. Nếu sử dụng ngôn ngữ tiềm ẩn không an toàn như C, lập trình viên cần viết code an toàn một cách tường minh. Bằng thiết kế với code mới, sau khi xem xét code cũ. Xem an toàn tràn bộ đệm như tập con các kỹ thuật lập trình an toàn nói chung. Chú ý đến các lỗi nhỏ, kiểm tra đủ không gian trong bộ đệm bất kỳ.

Có đề nghị mở rộng an toàn cho C như tạo điểm phạt thực thi, cần dịch chương trình với chương trình dịch đặc biệt. Có một số phương án thư viện chuẩn an toàn, các hàm mới, như strlcpy(). Cài đặt lại an toàn hơn một số hàm chuẩn như thư viện động, chẳng hạn Libsafe.

Bổ sung code của chức năng nhập và thoát để kiểm tra ngăn xếp ghi nhận việc ghi đè, sử dụng yếu tố ngẫu nhiên như bảo vệ ngăn xếp, kiểm tra viết đè giữa biến cục bộ và con trỏ khung lưu trữ và địa chỉ trả về. Chương

trình dừng nếu phát hiện thay đổi. Phát hành: bản dịch lại, hỗ trợ phát hiện lỗi hoặc copy an toàn lưu trữ/kiểm tra địa chỉ trả về.

Bảo vệ trong thời gian chạy

Sử dụng hỗ trợ bộ nhớ ảo để tạo một số vùng bộ nhớ không thực thi được như stack, heap, global data. Cần hỗ trợ từ các phần cứng bộ nhớ như trong SPARC / Solaris systems, x86 Linux/Unix/Windows systems. Phát hành hỗ trợ cho code ngăn xếp thực thi, cần một số dự phòng đặc biệt.

Thao tác trên vị trí của các cấu trúc dữ liệu chính, sử dụng tịnh tiến ngẫu nhiên cho mỗi tiến trình, có vùng địa chỉ lớn trên các phương tiện của các hệ thống hiện đại chống các va chạm và đoán địa chỉ bộ đệm đích là không thể. Vị trí ngẫu nhiên cho bộ đệm heap và vị trí các hàm thư viện chuẩn. Đặt các trang bảo vệ giữa các vùng quan trọng của bộ nhớ, đặt cờ trong bộ nhớ như địa chỉ không hợp lệ. Có thể ngay cả đặt giữa các khung ngăn xếp và các bộ đệm heap trong thời gian thực thi và phải trả giá về không gian.

Có nhiều các phương án tấn công khác: phương án tràn ngăn xếp, tràn heap, tràn dữ liệu tổng thể, tràn xâu định dạng, tràn số nguyên. Có thể có nhiều hơn nữa được phát hiện trong tương lai. Một số không thể ngăn chặn trừ khi code an toàn lúc ban đầu.

Phương án tràn ngăn xếp chỉ viết đè bộ đệm và con trỏ khung lưu trữ trả về xảy ra nhưng đến khung giả trả về lời gọi hàm điều khiển bởi kẻ tấn công được dùng khi có tràn bộ đệm giới hạn. Ví dụ tách ra bởi một khung. Tuy có các hạn chế: cần biết địa chỉ chính xác của bộ đệm, hàm gọi thực hiện với khung giả, phương án tràn ngăn xếp thay địa chỉ trả về bằng hàm thư viện chuẩn để đáp lại sự bảo vệ ngăn xếp không thực thi. Kẻ tấn công xây dựng các tham số phù hợp trên ngăn xếp phía trên địa chi trả về. Kẻ tấn công có thể cần địa chỉ chính xác của bộ đệm, có thể ngay cả kết nối hai lời gọi thư viện.

Cũng có tấn công bộ đệm đặt trong heap. Thông thường đặt trên code của chương trình, bộ nhớ được yêu cầu bởi chương trình để sử dụng cho các cấu trúc dữ liệu động, ví dụ như danh sách móc nối. Không có địa chỉ trả về, nên không có chuyển giao quyền điều khiển dễ dàng. Có thể có con trỏ hàm để khai thác hoặc thao tác cấu trúc dữ liệu quản trị. Cách bảo vệ là dùng heap ngẫu nhiên và không thực thi.

Có thể tấn công bộ đệm đặt trong dữ liệu tổng thể. Có thể đặt phía trên code của chương trình. Nếu có con trỏ hàm và bộ đệm có lỗ hổng hoặc bảng quản trị các quá trình liền kề. Nhắm tới viết đè con trỏ hàm được gọi sau đó. Cách bảo vệ là dùng vùng dữ liệu tổng thể ngẫu nhiên và không thực thi, dịch chuyển con trỏ hàm, các trang bảo vệ.

5.4 Bức tường lửa

5.4.1 Mở đầu

Bức tường lửa phát triển mạnh mẽ, được ứng dụng trong các các hệ thống thông tin. Bây giờ mọi người đều muốn lên Internet và các mạng liên kết với nhau. Vì vậy cần quan tâm thường xuyên về an toàn. Không dễ dàng bảo vệ từng hệ thống trong tổ chức. Thông thường sử dụng bức tường lửa, cung cấp vòng bảo vệ như một phần của chiến lược an toàn toàn diện.

Bức tường lửa là gì

Là điểm cổ chai để kiểm soát và theo dõi. Các mạng liên kết với độ tin cậy khác nhau, buộc có hạn chế trên các dịch vụ của mạng. Chẳng hạn, vận chuyển phải có giấy phép. Kiểm tra và kiểm soát truy cập, có thể cài đặt cảnh báo các hành vi bất thường. Cung cấp bảng NAT và sử dụng theo dõi giám sát. Cài đặt mạng riêng ảo (VPN) sử dụng cơ chế an toàn IPSec. Có thể miễn dịch trước.

Hạn chế của bức tường lửa

Không bảo vệ được các tấn công đi vòng qua nó, chẳng hạn mạng lén lút, thiết bị modems. Nó ngăn cản cả các tổ chức tin cậy và dịch vụ tin cậy (SSL/SSH).

Không bảo vệ chống các mối đe dọa từ bên trong, chẳng hạn như những nhân viên bực tức hoặc thông đồng với kẻ xấu. Không thể bảo vệ chống việc truyền các chương trình hoặc file nhiễm virus, vì có phạm vi rất rộng các dạng file và các hệ điều hành

5.4.2 Bức tường lửa – các lọc gói

Là thành phần của bức tường lửa nhanh nhất và đơn giản nhất, là cơ sở của mọi hệ thống tường lửa. Nó kiểm tra mỗi gói IP (không có ngữ cảnh) và cho phép hay từ chối tuỳ theo qui tắc xác định. Suy ra có hạn chế truy cập đến các dịch vụ và các cổng.

Các đường lối mặc định có thể

- Rằng không cho phép tức là cấm

Rằng không cấm tức là cho phép

Tấn công các lọc gói

Địa chỉ IP lừa đảo: giả địa chỉ nguồn làm cho tin tưởng, bổ sung bộ lọc lên mạch chuyển để ngăn chặn.

Tấn công mạch truyền gốc: kẻ tấn công đặt được truyền khác với mặc định, ngăn chặn các gói truyền gốc

Tấn công các đoạn tin (fragment) nhỏ. Chia thông tin phần đầu thành một số đoạn nhỏ. Hoặc bỏ qua hoặc sắp xếp lại trước khi kiểm tra

Bức tường lửa – các lọc gói trạng thái

Lọc gói truyền thống không kiểm tra ngữ cảnh của tầng cao hơn, tức là sánh các gói về với dòng chảy ra. Lọc gói trạng thái hướng đến yêu cầu đó. Chúng kiểm tra mỗi gói IP trong ngữ cảnh: giữ vết theo dõi với các kỳ client-server, kiểm tra từng gói đúng thuộc vào một phiên. Suy ra có khả năng tốt hơn phát hiện các gói giả tách khỏi ngữ cảnh.

5.4.3 Bức tường lửa – cổng giao tiếp ở tầng ứng dụng (hoặc proxy)

Có cổng giao tiếp chuyên dùng cho ứng dụng – proxy (người được uỷ quyền).

Có truy cập đầy đủ đến giao thức

- Người sử dụng yêu cầu dịch vụ từ proxy
- Proxy kiểm tra các yêu cầu có hợp lệ không
- Sau đó xử lý yêu cầu và trả lời cho người sử dụng
- Có thể vào/theo dõi vận chuyển ở tầng ứng dụng

Cần các proxies khác nhau cho mỗi dịch vụ

- Một số dịch vụ hỗ trợ một cách tự nhiên proxy
- Những loại khác thì cần giải quyết một số vấn đề

5.4.4 Bức tường lửa - cổng giao tiếp mức mạch vòng

Chuyển tiếp 2 kết nối TCP. Có sự an toàn bằng cách hạn chế mà các kết nối này cho phép. Mỗi lần tạo ra chuyển tiếp thông thường không kiểm tra nội dung. Thông thường được sử dụng khi tin cậy người sử dụng bên trong bằng cách cho phép các kết nối ra ngoài nói chung. Gói SOCKS được sử dụng rộng rãi cho mục đích này.

5.4.5 Máy chủ Bastion

Hệ thống máy chủ an toàn cao. Chạy cổng giao tiếp mức ứng dụng và mạch vòng. Hoặc cung cấp các dịch vụ truy cập bên ngoài. Có tiềm năng thể hiện các yếu tố của máy chủ. Vì an toàn bền vững, nên hệ điều hành nặng nề hơn, các dịch vụ chính, bổ sung xác thực, proxies nhỏ, an toàn, độc lập, không đặc quyền.

Có thể hỗ trợ 2 hay nhiều hơn kết nối mạng và có thể được tin cậy để ép buộc chính sách tách bạch tin cậy giữa các kết nối mạng.

Cấu hình bức tường lửa (Firewall Configurations)

5.4.6 Kiểm soát truy cập

Hệ thống đã xác định được định danh như người sử dụng, xác định các nguồn gốc nào nó có thể truy cập. Mô hình tổng quát là ma trận truy cập với

- Chủ thể - thực thể chủ động (người sử dụng, quá trình)
- Đối tượng - thực thể bị động (file hoặc nguồn)
- Quyền truy cập – cách mà đối tượng được truy cập

Có thể được phân tách bởi

- Các cột như danh sách kiểm soát truy cập
- Các hàng như các thẻ về khả năng

Ma trận kiểm soát quyền truy cập

5.4.7 Các hệ thống máy tính tin cậy

An toàn thông tin ngày càng quan trọng. Có các mức độ khác nhau về sự nhạy cảm của thông tin

- Phân loại thông tin quân sự: bảo mật, bí mật

Chủ thể (người hoặc chương trình) có nhiều quyền khác nhau truy cập đến các đối tượng thông tin. Được biết như an toàn nhiều tầng

- Chủ thể có mức độ an toàn tối đa và hiện tại
- Đối tượng có phân loại mức độ tin cậy cố định

Muốn xem xét các cách tăng độ tin tưởng trong hệ thống để củng cố các quyền đó.

5.4.8 Mô hình Bell LaPadula

Một trong những mô hình an toàn nổi tiếng nhất. Được cài đặt như các chính sách bắt buộc trong hệ thống. Có hai chính sách chính

- Không đọc lên (tính chất an toàn đơn giản)

§ Chủ thể chỉ có thể đọc/viết các đối tượng nếu mức độ an toàn hiện tại của chủ thể trội hơn (>=) phân loại của đối tượng

- Không viết xuống (tính chất *)

§ Chủ thể chỉ có thể bổ sung/viết lên đối tượng nếu mức độ an toàn hiện tại của chủ thể được trội (<=) bởi phân loại của đối tượng

Reference Monitor

(Giao diện chỉ dẫn)

Các hệ thống máy tính triển khai

Chính phủ có thể phát triển các hệ thống IT. Đương đầu với phạm vi rộng các chuẩn

- TCSEC, IPSEC và bây giờ là Tiêu chuẩn Chung

Xác định một số mức độ triển khai với tăng cường kiểm tra qui tắc. Đã xuất bản danh sách các sản phẩm triển khai

- Chỉ hướng tới sử dụng cho chính phủ/quốc phòng
- Cũng có thể hữu ích trong công nghiệp

5.4.9 Tiêu chuẩn chung

Đặc tả yêu cầu an toàn quốc tế khởi đầu và xác định tiêu chuẩn triển khai. Tích hợp với các chuẩn khác

- Chẳng hạn CSEC, ITSEC, CTCPEC (Canada), Federal (US)

Đặc tả các chuẩn cho

- Tiêu chuẩn triển khai
- Phương pháp luận cho ứng dụng của Tiêu chuẩn
- Các thủ tục hành chính triển khai, chứng nhận và các sơ đồ chỉ định

Xác định tập các yêu cầu an toàn, có đích triển khai (TOE). Yêu cầu rơi vào trong 2 loại sau

- Chức năng
- Sự tin cậy

Cả hai được tổ chức theo lớp classes của họ hoặc cấu thành

Các yêu cầu Tiêu chuẩn chung

Yêu cầu chức năng

- Kiểm soát an toàn, hỗ trợ mã, trao đổi thông tin, bảo vệ dữ liệu người sử dụng, định danh và xác thực, quản lý an toàn, tính riêng tư, bảo vệ các hàm an toàn tin cậy, nguồn thiết thực, truy cập TOE, đường dẫn tin cậy

Yêu cầu sự tin cậy

- Quản lý tham số hệ thống, phân phối và thao tác, phát triển, tài liệu chỉ dẫn, hỗ trợ thời gian sống, kiểm tra, đánh giá lỗ hổng, bảo trì sự tin cậy

5.5 Bài tập

1. Liệt kê và phân loại các phần mềm có hại và các biện pháp phòng chống.

2. Phân tích các kỹ thuật xâm nhập hệ thống và cách phòng ngừa.

3. Nêu các biện pháp tăng cường an ninh, bảo mật máy tính cá nhân dựa trên các phần mềm thông dụng hiện có.

4. Mục đích yêu cầu của việc xây dựng bức tường lửa. Có những loại bức tường lửa nào.

5. Nêu cách thiết lập bức tường lửa sử dụng công cụ hỗ trợ trong hệ điều hành.

6. Phân tích các lỗi tràn bộ nhớ có thể xảy ra, nêu nguyên nhân.

7. Tìm hiểu các yêu cầu lập trình an toàn.

Trình bày một số mô hình hệ thống máy tính tin cậy.

CHƯƠNG 6
AN TOÀN IP VÀ WEB

Mục tiêu:

- Trình bày được các hình thức tấn công vào hệ thống mạng;
- Xác định được các thành phần của một hệ thống bảo mật.
- Thực hiện các thao tác an toàn với máy tính.

Nội dung chính:

Trong chương này chúng ta sẽ xét đến cơ chế an toàn IPSec và một số giao thức bảo mật lớp vận chuyển ứng dụng trên Web.

6.1 An toàn IP

Có khá nhiều cơ chế an toàn ứng dụng chuyên biệt như: S/MIME, PGP, Kerberos, SSL/HTTPS. Tuy nhiên có những cơ chế an toàn mà xuyên suốt nhiều tầng ứng dụng như là cơ chế an toàn IP được cài đặt trên mạng cho mọi ứng dụng.

6.1.1 IPSec

IPSec là cơ chế an toàn IP tổng quan. Nó cung cấp: xác thực, bảo mật và quản trị khoá. IPSec được dùng trên mạng LAN, mạng WAN riêng và chung và trên cả mạng Internet.

Lợi ích của IPSec

IPSec trên bức tường lửa/router cung cấp an toàn mạnh cho mọi việc truyền qua vành đai. Nó chống lại việc đi vòng qua bức tường lửa/router.

IPSec nằm ở tầng vận chuyển bên dưới nên trong suốt với mọi ứng dụng và có thể trong suốt với người sử dụng đầu cuối. Nó có thể cung cấp an toàn cho người sử dụng riêng biệt và bảo vệ kiến trúc rẽ nhánh.

6.1.2 Kiến trúc an toàn IP

Đặc tả an toàn IP rất phức tạp, được định nghĩa qua một số chuẩn (RFC): bao gồm RFC 2401/2402/2406/2408 và có nhiều chuẩn khác được nhóm theo loại. Điều này là bắt buộc đối với IP6 và tuỳ chọn với IP4. Có hai mở rộng an toàn cho phần đầu:

Phần đầu xác thực (AH – Authentication Header)

Tải trọng an toàn đóng gói (ESP – Encapsulating Security Payload)

a.Dịch vụ IPSec

IPSec nhằm đạt các mục đích sau: kiểm soát truy cập, toàn vẹn không kết nối, xác thực nguồn gốc dữ liệu, từ chối tải lại gói (đây là một dạng của

toàn vẹn liên kết từng phần), bảo mật (mã hoá), bảo mật luồng vận chuyển có giới hạn.

b.Liên kết an toàn

Quan hệ một chiều giữa người gửi và người nhận mà cung cấp sự an toàn cho luồng vận chuyển và được xác định bởi 3 tham số

- Chỉ số tham số an toàn
- Địa chỉ IP đích
- Tên của thủ tục an toàn

Ngoài ra có một số các tham số khác như: chỉ số dãy (sequence number), thông tin về phần đầu xác thực và phần đầu mở rộng AH & EH, thời gian sống. Có lưu trữ cơ sở dữ liệu của các liên kết an toàn.

c.Phần đầu xác thực (Authentication Header - AH)

AH cung cấp sự hỗ trợ cho an toàn dữ liệu và xác thực của các gói IP:

- Hệ thống đầu cuối/chuyển mạch có thể xác thực người sử dụng/ứng dụng
- Ngăn tấn công theo dõi địa chỉ bằng việc theo dõi các chỉ số dãy.

AH dựa trên sử dụng MAC: HMAC–MD5–96 hoặc HMAC – SHA -1-96

Muốn vậy các bên cần chia sẻ khoá mật.

d.Tải trọng an toàn đóng gói (ESP)

ESP đảm bảo bảo mật nội dung mẫu tin và luồng vận chuyển giới hạn, có lựa chọn cung cấp dịch vụ xác thực và hỗ trợ phạm vi rộng các mã, các chế độ mã, bộ đệm

- Bao gồm DES, Triple DES, RC5, IDEA, CAST,…
- CBC và các chế độ khác
- Bộ đệm cần thiết để lấp đầy các kích thước khối, các trường cho luồng vận chuyển

e.Chế độ vận chuyển và chế độ ống ESP

ESP được sử dụng với 2 chế độ: vận chuyển và ống. Trong chế độ ống không cần giữ tường minh địa chỉ đích.

Chế độ vận tải được sử dụng để mã và tuỳ chọn xác thực dữ liệu IP:

- Dữ liệu được bảo vệ nhưng phần đầu vẫn để rõ để biết địa chỉ đích
- Có thể phân tích vận chuyển một cách hiệu quả

- Tốt đối với ESP máy chủ vận chuyển tới máy chủ

Chế độ ống mã toàn bộ gói IP

- Bổ sung phần đầu mới cho bước nhảy tiếp
- Tốt cho mạng riêng ảo VPN (Virtual Private Network), cổng đến cổng an toàn

f. kết hợp các liên kết an toàn

Các liên kết an toàn có thể cài đặt qua AH hoặc ESP. Để cài đặt cả hai cần kết hợp các liên kết an toàn

- Tạo nên bó các liên kết an toàn
- Có thể kết thúc tại các điểm cuối cùng nhau hoặc khác nhau
- Kết hợp bởi kề vận chuyển và ống lặp

Cần bàn luận về thứ tự xác thực và mã hoá

g. Quản trị khoá

Quản lý sinh khoá và phân phối khoá giữa các bên trao đổi thông tin, thông thường cần hai cặp khoá, 2 khoá trên một hướng cho AH và ESP.

Trong cơ chế Quản trị khoá thủ công, người quản trị hệ thống thiết lập cấu hình cho từng hệ thống.

Trong cơ chế Quản trị khoá tự động:

- Hệ thống tự động dựa vào yêu cầu về khoá cho các liên kết an toàn trong hệ thống lớn.
- Có các thành phần như thủ tục trao đổi khóa Oakley và liên kết an toàn trên mạng ISAKMP

h. Oakley

Oakley là thủ tục trao đổi khoá, dựa trên trao đổi khoá Diffie-Hellman. Ở đây bổ sung các đặc trưng để khắc phục các điểm yếu như Cookies, nhóm (tham số tổng thể), các chỉ số đặc trưng (nonces), trao đổi khoá Diffie Hellman với việc xác thực. Có thể sử dụng số học trên trường số nguyên tố hoặc đường cong elip.

i. ISAKMP

ISAKMP liên kết an toàn trên Internet và thủ tục quản trị khoá. Nó cung cấp khung để quản lý khoá, xác định các thủ tục và định dạng gói để thiết lập, thỏa thuận, điều chỉnh và xoá các liên kết an toàn (SA – Secure Associations). ISAKMP độc lập với thủ tục trao đổi khoá, thuật toán mã hoá và phương pháp xác thực

Trao đổi và tải trọng ISAKMP

Có một số kiểu tải trọng ISAKMP: an toàn, đề xuất, dạng vận chuyển, khoá, định danh, chứng nhận, hash, chữ ký, nonce và xoá.

ISAKMP có bộ khung cho 5 kiểu trao đổi mẫu tin:cơ sở, bảo vệ định danh, xác thực, tích cực và thông tin.

6.2 An toàn Web

6.2.1 Khái niệm

Web ngày càng được sử dụng rộng rãi bởi các công ty, chính phủ và cá nhân, nhưng Internet và Web có những lỗ hổng lớn và có nhiều mối đe doạ an toàn như:

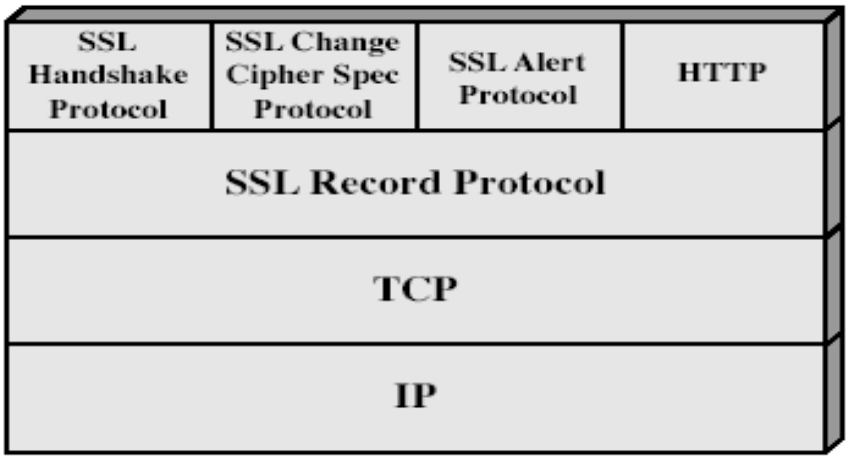
- Tính toàn vẹn
- Bảo mật
- Từ chối dịch vụ
- Xác thực

Như vậy cần bổ sung cơ chế bảo mật cho Web.

6.2.2 SSL (Secure Socket Layer)

SSL là dịch vụ an toàn tầng vận chuyển, ban đầu được phát triển bởi Netscape. Sau đó phiên bản 3 của nó được thiết kế cho đầu vào công cộng và trở thành chuẩn Internet, được biết đến như an toàn tầng vận chuyển TLS (Transport Layer Security).

SSL sử dụng giao thức TCP để cung cấp dịch vụ đầu cuối đến cuối tin cậy và có 2 tầng thủ tục



Ở đây kết nối SSL là:

- Tạm thời, đầu cuối đến đầu cuối, liên kết trao đổi
- Gắn chặt với 1 phiên SSL

Và phiên SSL:

- Liên kết giữa người sử dụng và máy chủ
- Được tạo bởi thủ tục HandShake Protocol
- Xác định một tập các tham số mã hoá
- Có thể chia sẻ bởi kết nối SSL lặp

a. Dịch vụ thủ tục bản ghi SSL

Dịch vụ thủ tục bản ghi SSL đảm bảo tính toàn vẹn của bản tin:

- Sử dụng MAC với khoá mật chia sẻ
- Giống như HMAC nhưng với bộ đệm khác

và cung cấp bảo mật:

- Sử dụng mã đối xứng với khoá chung xác định bởi thủ tục HandShake.
- IDEA, RC2-40, DES-40, DES, 3DES, Fortezza, RC4-40, RC4-128
- Bản tin được nén trước khi mã

b. Thủ tục thay đổi đặc tả mã SSL (SSL Change Cipher Spec Protocol):

Đây là một trong 3 giao thức chuyên biệt của SSL sử dụng thủ tục bản ghi SSL. Đây là mẫu tin đơn, buộc trạng thái treo trở thành hiện thời và cập nhật bộ mã đang dùng

c. Thủ tục nhắc nhở SSL (SSL Alert Protocol)

Truyền đi lời nhắc của SSL liên quan cho thành viên. Nghiêm khắc: nhắc nhở hoặc cảnh báo

Nhắc nhở đặc biệt:

- cảnh báo: mẫu tin không chờ đợi, bản ghi MAC tồi, lỗi giải nén, lỗi Handshake, tham số không hợp lệ
- Nhắc nhở: đóng ghi chú, không chứng nhận, chứng nhận tồi, chứng nhận không được hỗ trợ, chứng nhận bị thu hồi, chứng nhận quá hạn, chứng nhận không được biết đến.

Nén và mã như mọi dữ liệu SSL

d. Thủ tục bắt tay SSL (SSL HandShake Protocol)

Thủ tục này cho phép máy chủ và máy trạm:

- Xác thực nhau
- Thỏa thuận thuật toán mã hoá và MAC
- Thỏa thuận khoá mã sẽ dùng

Nó bao gồm một loạt các thông tin:

- Thiết lập các khả năng an toàn
- Xác thực máy chủ và trao đổi khoá
- Xác thực máy trạm và trao đổi khoá
- Kết thúc

e. An toàn tầng vận chuyển

IETF chuẩn RFC 2246 giống như SSLv3.

Với khác biệt nhỏ:

- số ký hiệu kích thước bản ghi
- sử dụng HMAC thay cho MAC
- hàm giả ngẫu nhiên tăng độ mật
- có mã ghi chú bổ sung
- có một số thay đổi hỗ trợ mã
- thay đổi kiểu chứng nhận và thỏa thuận
- thay đổi bộ đệm và tính toán mã

Sau đây ta xem xét chi tiết giao thức xác thực người dùng RADIUS và giao thức SSL:

Giao thức RADIUS

RADIUS la`một dịch vụ danh cho việc xac nhận va`cho phep ngươi dung truy cập từ xa qua các thiết bị như môdem, DSL, cáp mạng hoặc các thiết bị không dây khác. Một site thông thương co´một máy chủ truy cập được kêt nôi vaò một modem. Một máy chủ dịch vụ RADIUS được kêt nôi vaò mạng như một dịch vụ xac nhận. Nhưng ngươi dung tư`xa gọi vaò máy chủ truy cập, máy chủ se˜yêu câu nhưng dịch vụ xac nhận tư`máy chủ dịch vụ RADIUS. Máy chủ dịch vụ RADIUS se˜ xac nhận ngươi dung va` cho phep họ truy cập tai nguyên.Nhưng nha`quản trị mạng tạo ra nhưng hô`sơ vê`ngươi dung ở máy chủ RADIUS,xac định cac quyên hạn câp cho ngươi dung tư`xa. Nhưng giao thưc hỏi đap được sử dụng trong suôt qua´trình ngươi dung vaò mạng.

Giao thức SSL

Được phát triển bởi Netscape, giao thức SSL đã được sử dụng rộng rãi trên mạng Internet trong việc xác thực và mã hoá thông tin giữa máy trạm và máy chủ. Trong khi SSL có thể sử dụng để hỗ trợ các giao dịch an toàn cho rất nhiều ứng dụng khác nhau trên Internet. SSL không phải là một giao thức đơn lẻ, mà là một tập các thủ tục đã được chuẩn hoá để thực hiện các nhiệm vụ bảo mật sau:

- Xác thực máy chủ: Cho phép người sử dụng xác thực được máy chủ muốn kết nối. Lúc này, phía browser sử dụng các kỹ thuật mã hoá công khai để chắc chắn rằng chứng chỉ và khoá công cộng của máy chủ là có giá trị và được cấp phát bởi một CA trong danh sách các CA đáng tin cậy của máy trạm.

- Xác thực máy trạm: Cho phép phía máy chủ xác thực được người sử dụng muốn kết nối. Phía máy chủ cũng sử dụng các kỹ thuật mã hoá công khai để kiểm tra xem chứng chỉ và khoá công cộng của máy chủ có giá trị hay không và được cấp phát bởi một CA trong danh sách các CA đáng tin cậy không.

- Mã hoá kết nối: Tất cả các thông tin trao đổi giữa máy trạm và máy chủ được mã hoá trên đường truyền nhằm nâng cao khả năng bảo mật.

Hoạt động của SSL

Giao thức SSL hoạt động dựa trên hai nhóm con giao thức là giao thức "bắt tay" và giao thức "bản ghi". Giao thức bắt tay xác định các tham số giao dịch giữa hai đối tượng có nhu cầu trao đổi thông tin hoặc dữ liệu, còn giao thức bản ghi xác định khuôn dạng cho tiến hành mã hoá và truyền tin hai chiều giữa hai đối tượng đó.Giao thức SSL "bắt tay" sẽ sử dụng SSL "bản ghi" để trao đổi một số thông tin giữa máy chủ và máy trạm vào lần đầu tiên thiết lập kết nối SSL.

Một giao dịch SSL thường bắt đầu bởi quá trình "bắt tay" giữa hai bên. Các bước trong quá trình "bắt tay" có thể như sau:

1. Máy trạm sẽ gửi cho máy chủ số phiên bản SSL đang dùng, các tham số của thuật toán mã hoá, dữ liệu được tạo ra ngẫu nhiên (chữ ký số) và một số thông tin khác mà máy chủ cần để thiết lập kết nối với máy trạm

2. Máy chủ gửi cho máy trạm số phiên bản SSL đang dùng, các tham số của thuật toán mã hoá, dữ liệu được tạo ra ngẫu nhiên và một số thông tin khác mà máy trạm cần để thiết lập kết nối với máy chủ. Ngoài ra máy chủ cũng gửi chứng chỉ của nó đến máy trạm và yêu cầu chứng chỉ của máy trạm nếu cần.

3. Máy trạm sử dụng một số thông tin mà máy chủ gửi đến để xác thực máy chủ. Nếu như máy chủ không được xác thực thì người sử dụng sẽ được cảnh báo và kết nối không được thiết lập. Còn nếu như xác thực được máy chủ thì phía máy trạm sẽ thực hiện tiếp bước 4.

4. Sử dụng tất cả các thông tin được tạo ra trong giai đoạn bắt tay ở trên, máy trạm (cùng với sự cộng tác của máy chủ và phụ thuộc vào thuật toán được sử dụng) sẽ tạo ra premaster secret cho phiên làm việc, mã hoá bằng khoá công khai mà máy chủ gửi đến trong chứng chỉ ở bước 2, và gửi đến máy chủ.

5. Nếu máy chủ có yêu cầu xác thực máy trạm, thì phía máy trạm sẽ đánh dấu vào phần thông tin riêng chỉ liên quan đến quá trình "bắt tay" này mà hai bên đều biết. Trong trường hợp này, máy trạm sẽ gửi cả thông tin được đánh dấu và chứng chỉ của mình cùng với premaster secret đã được mã hoá tới máy chủ.

6. Máy chủ sẽ xác thực máy trạm. Trường hợp máy trạm không được xác thực, phiên làm việc sẽ bị ngắt. Còn nếu máy trạm được xác thực thành công, máy chủ sẽ sử dụng khoá bí mật để giải mã premaster secret, sau đó thực hiện một số bước để tạo ra master secret.

7. Máy trạm và máy chủ sẽ sử dụng master secret để tạo ra các khoá phiên , đó chính là các khoá đối xứng được sử dụng để mã hoá và giải mã các thông tin trong phiên làm việc và kiểm tra tính toàn vẹn dữ liệu.

8. Máy trạm sẽ gửi một lời nhắn đến máy chủ thông báo rằng các thông điệp tiếp theo sẽ được mã hoá bằng khoá phiên. Sau đó nó gửi một lời nhắn đã được mã hoá để thông báo rằng phía máy trạm đã kết thúc giai đoạn "bắt tay".

9. Máy chủ cũng gửi một lời nhắn đến máy trạm thông báo rằng các thông điệp tiếp theo sẽ được mã hoá bằng khoá phiên. Sau đó nó gửi một lời nhắn đã được mã hoá để thông báo rằng máy chủ đã kết thúc giai đoạn "bắt tay".

10. Lúc này giai đoạn "bắt tay" đã hoàn thành, và phiên làm việc SSL bắt đầu. Cả hai phía máy trạm và máy chủ sẽ sử dụng các khoá phiên để mã hoá và giải mã thông tin trao đổi giữa hai bên, và kiểm tra tính toàn vẹn dữ liệu

6.3 Thanh toán điện tử an toàn

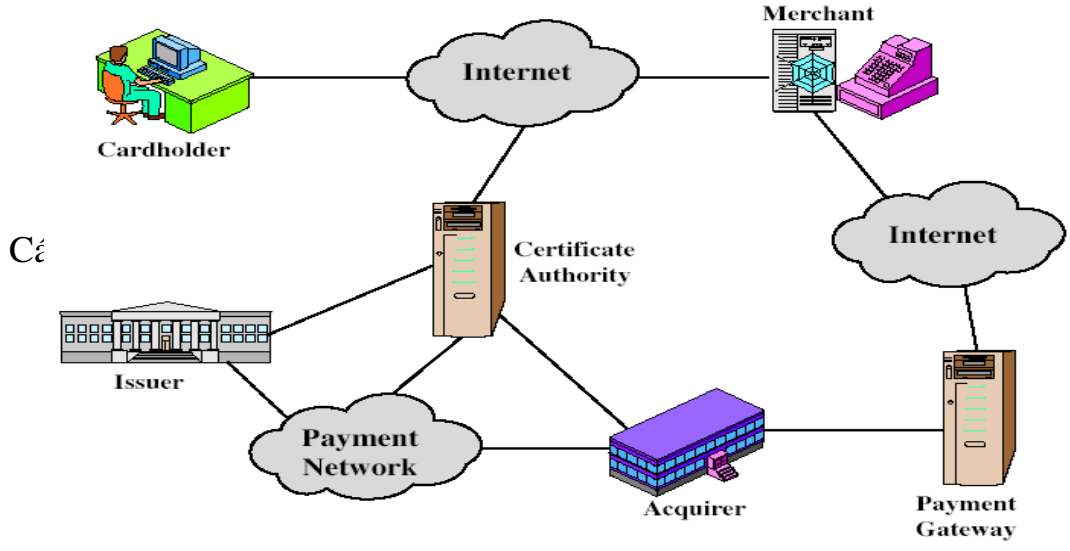
6.3.1 Yêu cầu

Đây là mã mở và đặc tả an toàn nhằm bảo vệ thanh toán thẻ tín dụng trên Internet. Nó được phát triển năm 1996 bởi Master, Visa Card và không phải hệ thống trả tiền. Thanh toán điện tử an toàn là tập các giao thức và định dạng an toàn dùng để

o

o

o



6.3.2 Thanh toán điện tử an toàn

a. Người mua mở tài khoản

b. Người mua nhận được chứng nhận

c. Người bán có chứng nhận của họ

d. Người mua đặt hàng

e. Người bán được kiểm chứng

f. Đơn đặt hàng và trả tiền được gửi

g. Người bán yêu cầu giấy phép trả tiền

h. Người bán duyệt đơn đặt hàng

i. Người bán cung cấp hàng và dịch vụ

j. Người bán yêu cầu trả tiền

6.3.3 Chữ ký kép

Người mua tạo chữ ký kép

- Thông tin đơn đặt OI cho người bán
- Thông tin trả tiền PI cho ngân hàng

Không bên nào biết chi tiết của người khác. Nhưng cần phải biết là họ được kết nối với nhau. Sử dụng chữ ký kép cho mục đích này

- Ký trên bản ghép của OI và PI

6.3.3 Yêu cầu trả tiền

Trao đổi yêu cầu trả tiền gồm 4 mẫu tin sau

Khởi tạo yêu cầu - nhận chứng nhận

Khởi tạo trả lời – ký trả lời

Yêu cầu trả tiền - của OI và PI

Trả lời trả tiền – đơn phúc đáp

6.3.4 Giấy phép cổng trả tiền

Kiểm chứng mọi chứng nhận

Giải mã phong bì điện tử của khối giấy phép và nhận được khoá đối xứng, sau đó giải mã khối giấy phép

Kiểm tra chữ ký của người bán trên khối giấy phép

Giải mã phong bì điện tử khối trả tiền, nhận được khoá đối xứng, sau đó giải mã khối trả tiền

Kiểm tra chữ ký kép trên khối trả tiền

Kiểm tra rằng, thanh toán ID nhận được từ người bán phù hợp với danh tính trong PI nhận được (không trực tiếp) từ người bán

Yêu cầu và nhận được giấy phép từ nơi phát hành

Gửi trả lời giấy phép cho người bán

6.3.5 Nhận trả tiền

Người bán gửi cho cổng trả tiền yêu cầu nhận trả tiền. Cổng kiểm tra yêu cầu đó. Sau đó yêu cầu chuyển tiền đến tài khoản người bán. Thông báo cho người bán và chờ trả lời việc nhận.

6.4 An toàn thư điện tử

Thư điện tử là một trong những dịch vụ mạng được coi trọng và ứng dụng rộng rãi nhất. Đồng thời nội dung của các mẫu tin không an toàn. Có thể bị quan sát trên đường truyền hoặc bởi những người có thẩm quyền thích hợp ở hệ thống đầu cuối.

Nâng cao an toàn thư điện tử là mục đích quan trọng của mọi hệ thống trao đổi thư. Ở đây phải đảm bảo các yêu cầu sau: tính bảo mật nội dung tin gửi, xác thực người gửi mẫu tin, tính toàn vẹn của mẫu tin, hơn nữa bảo vệ khỏi bị sửa, tính chống từ chối gốc, chống từ chối của người gửi.

6.4.1 Dịch vụ PGP.

PGP (Pretty Good Privacy) là một dịch vụ về bảo mật và xác thực được sử dụng rộng rãi cho chuẩn an toàn thư điện tử. PGP được phát triển bởi Phil Zimmermann. Ở đây lựa chọn các thuật toán mã hoá tốt nhất để dùng, tích hợp thành một chương trình thống nhất, có thể chạy trên Unix, PC, Macintosh và các hệ thống khác. Ban đầu là miĩen phí, bây giờ có các phiên bản thương mại. Sau đây chúng ta xem xét hoạt động của PGP

Thao tác PGP – xác thực

Người gửi tạo mẫu tin, sử dụng SHA-1 để sinh Hash 160 bit của mẫu tin, ký hash với RSA sử dụng khoá riêng của người gửi và đính kèm vào mẫu tin.

Người nhận sử dụng RSA với khoá công khai của người gửi để giải mã và khôi phục bản hash. Người nhận kiểm tra mẫu tin nhận sử dụng bản hash của nó và so sánh với bản hash đã được giải mã.

Thao tác PGP – bảo mật

Người gửi tạo mẫu tin và số ngẫu nhiên 128 bit như khoá phiên cho nó, mã hoá mẫu tin sử dụng CAST-128/IDEA /3DES trong chế độ CBC với khoá phiien đó. Khoá phiên được mã sử dụng RSA với khoá công khai người nhận và đính kèm với mẫu tin.

Người nhận sử dụng RSA với khoá riêng để giải mã và khôi phục khoá phiên. Khoá phiên được sử dụng để giải mã mẫu tin.

Thao tác PGP - Bảo mật và xác thực

Có thể sử dụng cả hai dịch vụ trên cùng một mẫu tin. Tạo chữ ký và đính vào mẫu tin, sau đó mã cả mẫu tin và chữ ký. Đính khoá phiên đã được mã hoá RSA/ElGamal.

Thao tác PGP – nén

Theo mặc định PGP nén mẫu tin sau khi ký nhưng trước khi mã. Như vậy cần lưu mẫu tin chưa nén và chữ ký để kiểm chứng về sau. Vì rằng nén là không duy nhất. Ở đây sử dụng thuật toán nén ZIP.

Thao tác PGP – tương thích thư điện tử

Khi sử dụng PGP sẽ có dữ liệu nhị phân để gửi (mẫu tin được mã). Tuy nhiên thư điện tử có thể thiết kế chỉ cho văn bản. Vì vậy PGP cần mã dữ liệu nhị phân thô vào các ký tự ASCII in được. Sau đó sử dụng thuật toán Radix 64, ánh xạ 3 byte vào 4 ký tự in được và bổ sung kiểm tra thừa quay vòng CRC để phát hiện lỗi khi truyền. PGP sẽ chia đoạn mẫu tin nếu nó quá lớn.

Tóm lại, cần có khoá phiên cho mỗi mẫu tin, có kích thước khác nhau: 56 bit – DES, 128 bit CAST hoặc IDEA, 168 bit Triple – DES, được sinh ra sử dụng dữ liệu đầu vào ngẫu nhiên lấy từ sử dụng trước và thời gian gõ bàn phím của người sử dụng

Khoá riêng và công khai của PGP

Vì có nhiều khoá riêng và khoá công khai có thể được sử dụng, nên cần phải xác định rõ cái nào được dùng để mã khoá phiên trong mẫu tin. Có thể gửi khoá công khai đầy đủ với từng mẫu tin. Nhưng đều đó là không đủ, vì cần phải nêu rõ danh tính của người gửi. Do đó có thể sử dụng định danh

khoá để xác định người gửi. Có ít nhất 64 bit có ý nghĩa của khoá và là duy nhất, có thể sử dụng định danh của khoá trong chữ ký.

PGP Message Format

Các chùm khoá PGP

Mỗi người sử dụng PGP có một cặp chùm khoá. Chùm khoá công khai chứa mọi khoá công khai của các người sử dụng PGP khác được người đó biết và được đánh số bằng định danh khoá (ID key). Chùm khoá riêng chứa các cặp khoá công khai/riêng của người đó được đánh số bởi định danh khoá và mã của khoá lấy từ giai đoạn duyệt hash. An toàn của khoá công khai như vậy phụ thuộc vào độ an toàn của giai đoạn duyệt.

Sinh mẫu tin PGP

Sơ đồ sau mô tả qui trình sinh mẫu tin PGP để gửi cho người nhận.

Nhận mẫu tin PGP

Sơ đồ sau nêu cách người nhận giải mã, kiểm chứng thông tin để đọc mẫu tin.

Quản lý khoá PGP

Tốt hơn hết dựa vào chủ quyền chứng nhận. Trong PGP mỗi người sử dụng có một CA của mình. Có thể ký khoá cho người sử dụng mà anh ta biết trực tiếp. Tạo thành "Web của niềm tin". Cần tin cậy khóa đã được ký, và tin cậy các khoá mà các người khác ký khi dùng một dây chuyền các chữ ký đến nó.

Chùm khoá chưá cả các chỉ dẫn tin cậy. Người sử dụng có thể thu hồi khoá của họ

6.4.2 Mở rộng thư Internet đa mục đích/an toàn S/MIME

Tăng cường an toàn cho thư điện tử đa mục đích mở rộng MIME (Multipurpose Internet Mail Extension). Thư điện tử Internet RFC822 gốc chỉ có văn bản, MIME cung cấp hỗ trợ cho nhiều kiểu nội dung và mẫu tin có nhiều phần với mã hoá dữ liệu nhị phân thành dạng văn bản.

S/MIME tăng cường tính an toàn, có hỗ trợ của S/MIME trong nhiều tác nhân thư điện tử như MS Outlook, Mozilla, Mac Mail, …

Các chức năng S/MIME

Dữ liệu đóng phong bì, nội dung được mã hoá và liên kết khoá, dữ liệu được ký, mẫu tin được mã và ký sau nén, dữ liệu rõ ràng được ký, mẫu tin tường minh và mã hoá ch ữ ký trên bản nén, dữ liệu đóng phong bì và ký, lồng nhau các th ực th ể ký và mã.

Các thuật toán mã hoá S/MIME

Các chữ ký điện tử DSS và RSA, các hàm hash: SHA-1 và MD5, mã khoá phiên: Elgamal & RSA, mã mẫu tin: AES, Triple-DES, RC2/40, …;MAC: HMAC với SHA-1.

Có quá trình để đối thoại quyết định sử dụng thuật toán nào.

Các mẫu tin S/MIME

S/MIME bảo vệ các thực thể MIME với chữ ký, mã hoặc cả hai tạo thành các đối tượng đóng gói MIME. Có phạm vi các kiểu nội dung khác nhau: dữ liệu đóng phong bì, dữ liệu được ký, dữ liệu rõ ràng được ký, yêu cầu đăng ký, chứng nhận mẫu tin.

Quá trình chứng nhận S/MIME

S/MIME sử dụng chứng nhận X.509 phiên bản 3. Quản trị việc sử dụng kết hợp sơ đồ phân cấp CA của X.509 và Web niềm tin của PGP. Mỗi client có một danh sách các giấy chứng nhận cho CA tin cậy và có các giấy chứng nhận và cặp khoá công khai/riêng của mình. Chứng nhận cần được ký bởi các CA tin cậy.

Chủ quyền chứng nhận CA (Certificate Authorities)

Có một số CA mọi người đều biết. Verisign là một CA được sử dụng rộng rãi. Verisign xuất bản một số kiểu định danh điện tử. Tăng mức kiểm tra và kéo theo độ tin cậy.

6.4.3 Bài tập

Bài 1. Nêu mục đích IPSec, các tham số, AH và ESP

Bài 2. Nêu mục đích SSL và TLS. Trình bày kiến trúc và nhiệm vụ của các thành phần của chúng.

Bài 3. Thế nào là thanh toán điện tử an toàn

Bài 4. Nêu yêu cầu của chữ ký kép và chứng tỏ chữ ký kép trong thanh toán điện tử an toàn đáp ứng các yêu cầu đó.

Bài 5. Nêu qui trình thanh toán điện tử an toàn, chứng tỏ nó đáp ứng được các yêu cầu an toàn đề ra.

Bài 6. Nêu các yêu cầu bảo mật, xác thực, chữ ký điện tử của hệ thống thư điện tử.

Bài 7. Trình bày giải pháp đề xuất của PGP cho hệ thống thư điện tử.

Bài 8. Tìm hiểu xác thực cở bản HTTP trong Internet Explorer.

CÁC THUẬT NGỮ CHUYÊN MÔN

Thuật ngữ	Giải thích
Database	Cơ sở dữ liệu

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1]. Ths. Ngô Bá Hùng-Ks. Phạm Thế Phi, *Giáo trình mạng máy tính*, Đại học Cần Thơ, năm 2005

[2]. Đặng Xuân Hà, *An toàn mạng máy tính*, NXB Giáo dục, năm 2005

[3]. Nguyễn Anh Tuấn, *Bài giảng Kỹ thuật an toàn mạng*, Trung tâm TH-NN Trí Đức, 2010

[4]. Angus Wong, Alan Yeung, *Network Infrastructure Security*, Springer Science+Business Media, 2009.

DANH SÁCH BAN BIÊN SOẠN GIÁO TRÌNH DẠY NGHỀ

TRÌNH ĐỘ TRUNG CẤP, CAO ĐẲNG

Tên giáo trình: QUẢN TRỊ CƠ SỞ DỮ LIỆU NÂNG CAO
Tên nghề: QUẢN TRỊ MẠNG

1. Ông (bà).....	Chủ nhiệm
2. Ông (bà).....	Phó chủ nhiệm
3. Ông (bà).....	Thư ký
4. Ông (bà).....	Thành viên
5. Ông(bà).....	Thành viên
6. Ông(bà).....	Thành viên
7. Ông(bà).....	Thành viên
8. Ông(bà).....	Thành viên
9. Ông(bà).....	Thành viên

DANH SÁCH HỘI ĐỒNG NGHIỆM THU

GIÁO TRÌNH DẠY NGHỀ TRÌNH ĐỘ TRUNG CẤP, CAO ĐẲNG

1. Ông (bà).....	Chủ tịch
2. Ông (bà).....	Phó chủ tịch
3. Ông (bà).....	Thư ký
4. Ông (bà).....	Thành viên
5. Ông(bà).....	Thành viên
6. Ông(bà).....	Thành viên
7. Ông(bà).....	Thành viên
8. Ông(bà).....	Thành viên
9. Ông(bà).....	Thành viên

Phụ lục 3.2

CÁC TIÊU CHÍ VÀ TIÊU CHUẨN ĐÁNH GIÁ CHẤT LƯỢNG GIÁO TRÌNH DẠY NGHỀ TRÌNH ĐỘ TRUNG CẤP VÀ CAO ĐẲNG

Số TT	Các tiêu chí đánh giá	Mức độ đánh giá			Ghi chú
		Đạt yêu cầu đề nghị ban hành ngay	Đạt yêu cầu nhưng phải chỉnh sửa	Chưa đạt yêu cầu phải xây dựng lại	
A	Sự tương ứng với chương trình				
1	Giáo trình có đủ các đề mục và thể hiện nội dung theo đúng mẫu định dạng				
2*	Giáo trình có đầy đủ các nội dung theo chương trình chi tiết các môn học/mô đun trong chương trình đào tạo				
3*	Nội dung các chương/bài đảm bảo mục tiêu kiến thức, kỹ năng đã đề ra không?				
4*	Khối lượng các thông tin trong các môn học/mô đun có phù hợp với thời lượng của chương trình không?				
B	Tính logíc				
5*	Nội dung từng chương/bài có được trình bày một cách logic với quá trình nhân thức không? (tức là: Mức độ từ dễ đến khó, tính trình tự cho các khái niệm từ đơn giản đến phức tạp)				
6*	Các bước hình thành kỹ năng có hợp lý và vừa phải không? (tức là quan sát mẫu - bắt trước - làm được - làm độc lập - làm thuần thục hoặc theo đường xoắn ốc để hình thành các kỹ xảo)				

7*	Mối quan hệ giữa lý thuyết và thực hành có hợp lý để bảo đảm được sự nhận thức và kiến thức, sự hình thành kỹ năng không?				
8*	Hình thức học tập và các giải pháp sư phạm cho từng chủ đề có thích hợp so với mục tiêu đã đề ra không?				
C	Mức đầy đủ/bao quát đối với mục tiêu				
9*	Nội dung có đầy đủ để đảm bảo đào tạo có kết quả theo các mục tiêu thực hiện không				
10*	Nội dung có được nhấn mạnh để rèn luyện, hình thành các kỹ năng cần thiết không? (Tức là có các quy trình rèn luyện/thực hành bao gồm cả các khía cạnh khác như: tinh thần trách nhiệm, tuân thủ kỷ luật, ý thức an toàn, ứng xử trong nhóm, tác phong công nghiệp…)				
11*	Các cấu phần tạo sự chủ động và học tích cực có đầy đủ không? (tức là đủ các mục: Giới thiệu, hướng dẫn, tự đánh giá, giải thích thuật ngữ, tài liệu tham khảo…)				
12	Có vận dụng được sự hỗ trợ của các trang thiết bị, nguồn học liệu, nguồn l;ực khác cho quá trình học tập của học viên không?				
13	Các hành ảnh minh họa, bảng biểu, bản vẽ, quy trình thực hiện…có đủ ở mức cần thiết, rõ ràng và ăn nhập với đoạn viết không?				
D	Tính chuẩn xác				
14	Nội dung khoa học của thông tin có chính xác không? (về bản chất vấn đề, về các số liệu, về các sự kiện và đường nét…được đề cập trên các đoạn viêt, các bảng biểu và các hình minh họa, bản vẽ..)				

15	Các thuật ngũ có đảm bảo tính phổ thông và nhất quán không?				
E	Phong cách biên soạn				
16	Ý tứ trình bày rõ ràng, sáng sủa, đơn giản và dễ hiểu không?				
17	Cân đối và phù hợp giữa kênh hình và kênh chữ				
18	Có vi phạm gì về văn hóa tập quán của các dân tộc Việt Nam không?				
19	Có sai phạm gì đối với Luật bản quyền không?				
20	Phong cách trình bày có thể hiện tính gợi mở, lôi kéo người học thực hiện công việc không?				
F	Cấu trúc và các chuyên mục				
21	Bố cục có nhất quán trong toàn bộ tài liệu không?				
22	Mối liên hệ giữa các chuyên mục có chặt chẽ và tương ứng với nhau không? (đặc biệt là mục tiêu, kiểm tra đánh giá và các hướng dẫn trả lời)				
23	Mã các chuyên mục, hình vẽ, bảng biểu, bản vẽ…có nhất quán và chính xác và tạo điều kiện thuận lợi cho việc tìm kiếm và liên hệ				

Ghi chú:

1. Các tiêu chí có đánh dấu * có ý nghĩa rất quan trọng đối với chất lượng giáo trình đã biên soạn
2. Các mức độ đánh giá:

- Đạt yêu cầu: Không phải sửa chữa gì hoặc chỉ cần sửa chữa vài lỗi nhỏ về biên tập;
- Đạt yêu cầu nhưng phải chỉnh sửa: Phải sửa chữa một số lỗi về nội dung chuyên môn và biên tập, chỉnh lý, bổ sung; sau đó trình chủ tịch, phó chủ tịch và thư ký hội đồng xem xét, nếu thông qua được thì đạt yêu cầu đề nghị phê duyệt;
- Không đạt yêu cầu: Có nhiều lỗi về nội dung chuyên môn và biên tập, phải biên soạn lại để trình Hội đồng thẩm định lại.