Bảo mật máy tính và mạng

Bởi:
Khoa CNTT ĐHSP KT Hưng Yên

Phiên bản trực tuyến:
< http://voer.edu.vn/content/col10207/1.1/ >

Hoc lieu Mo Vietnam - Vietnam Open Educational Resources

Tài liệu được hiệu đính bởi: August 5, 2010
Ngày tạo PDF: August 5, 2010
Để biết thông tin về đóng góp cho các module có trong tài liệu này, xem tr. 75.

Nội dung

Chương 1

Giới thiệu

1.1 Giới thiệu mục tiêu, nội dung, phương pháp học bảo mật máy tính[1]

1.1.1 Mục tiêu

Module Bảo mật máy tính và Mạng được đưa vào giảng dạy nhằm giúp người học có khả năng:

- Mô tả các nguyên lý bảo mật và các mô hình bảo mật;
- Phân tích rủi ro cho một hệ thống thông tin;
- Triển khai các kỹ thuật bảo mật bảo vệ hệ thống thông tin;
- Tư vấn về các vấn đề bảo mật cho doanh nghiệp.

Theo quan điểm năng lực, module này giúp người học phát triển các năng lực: Phân tích (4); Tư vấn (4); Thực hiện (3) và Bảo trì (3).

1.1.2 Nội dung

Module giới thiệu các vấn đề bảo mật máy tính và mạng máy tính. Các chủ đề (không hạn chế) bao gồm:

- Các kỹ thuật đảm bảo an toàn cho các hệ thống máy tính đa người dùng và các hệ thống máy tính phân tán;
- Sơ đồ nhận dạng và xác thực điện tử;
- Các mô hình bảo mật;
- Hệ mật mã: khóa bí mật, khóa công khai, chữ ký điện tử;
- Bảo mật hệ điều hành;
- Bảo mật phần mềm;
- Bảo mật thư điện tử và WWW;
- Thương mại điện tử: giao thức thanh toán, tiền điện tử;
- Phát hiện xâm nhập: virus máy tính;
- Tường lửa;
- Đánh giá rủi ro.

[1]This content is available online at <http://voer.edu.vn/content/m13179/1.1/>.

1.1.3 Phương pháp học tập

Để đăng ký học module này, trước đó người học phải tích lũy tín chỉ của các module Mạng máy tính, Kiến trúc máy tính, Cơ sở kỹ thuật lập trình, Toán chuyên ngành, và Kỹ nghệ phần mềm.

Thời lượng của module tương đương 3 tín chỉ, có kết cấu dạng lý thuyết kết hợp làm bài tập lớn, do vậy người học phải đăng ký chủ đề nghiên cứu theo nhóm (từ 2 đến 3 người) ngay từ buổi học đầu tiên. Trong quá trình học tập, sinh viên tham gia học tập trên lớp và làm việc nhóm theo các chủ đề đã đăng ký. Sau khi kết thúc 11 buổi học lý thuyết, các nhóm sinh viên báo cáo kết quả nghiên cứu trước lớp trong 4 buổi còn lại.

1.2 Một số khái niệm cơ bản trong bảo mật thông tin[2]

Bảo mật (security) là việc bảo vệ những thứ có giá trị [1]. Bảo mật thông tin (information security) là một chủ đề rộng bao gồm tất cả các vấn đề bảo mật có liên quan đến lưu trữ và xử lý thông tin. Lĩnh vực nghiên cứu chính của bảo mật thông tin gồm các vấn đề pháp lý như hệ thống chính sách, các quy định, yếu tố con người; các vấn đề thuộc tổ chức như kiểm toán xử lý dữ liệu điện tử, quản lý, nhận thức; và các vấn đề kỹ thuật như kỹ thuật mật mã, bảo mật mạng, công nghệ thẻ thông minh...

Bảo mật máy tính (computer security) là lĩnh vực liên quan đến việc xử lý ngăn ngừa và phát hiện những hành động bất hợp pháp/trái phép (đối với thông tin và tài nguyên hệ thống) của người dùng trong một hệ thống máy tính. Có nhiều định nghĩa khác nhau về bảo mật máy tính nhưng hầu hết đều đề cập đến ba khía cạnh sau đây:

- Sự bí mật (confidentiality): ngăn ngừa việc làm lộ trái phép thông tin
- Sự toàn vẹn (Integrity): ngăn ngừa việc sửa đổi trái phép đối với thông tin
- Sự sẵn sàng (Availability): ngăn ngừa việc chiếm dụng trái phép thông tin hoặc tài nguyên.

Trên thực tế, kỹ thuật mật mã được triển khai rộng rãi để đảm bảo tính bí mật và toàn vẹn của thông tin được lưu trữ hay truyền nhận nhưng kỹ thuật này không bảo đảm cho tính sẵn sàng của hệ thống.

Mạng máy tính được triển khai nhằm giúp máy tính mở rộng giao tiếp với môi trường bên ngoài đồng nghĩa việc tăng nguy cơ rủi ro. Chúng ta vì thế muốn kiểm soát cách người dùng hệ thống truy cập vào mạng, cách người dùng trên mạng truy cập vào hệ thống của chúng ta và cách thông tin được bảo vệ trên đường truyền. Do vậy, bảo mật mạng (network security) không chỉ đơn giản là mật mã mà còn đòi hỏi nhiều yêu cầu mới về kiểm soát truy xuất

1.3 Các chủ đề làm tiểu luận[3]

Sinh viên có thể chọn các chủ đề theo gợi ý (trong danh mục) hoặc chủ động lựa chọn các chủ đề nghiên cứu khác nhưng phải được sự đồng ý của giáo viên hướng dẫn. Danh mục các chủ đề bao gồm:

- Mạng riêng ảo (Virtual Private Network)
- Tường lửa (Hard and Soft-Firewall)
- Tìm hiểu kỹ thuật làm giả Email (Forged Email)
- Nghiên cứu phương pháp chống thư rác (Spam Email)
- Tìm hiểu IPSec trong bộ giao thức Ipv6
- Tìm hiểu một số công cụ (phần mềm) dùng để tấn công hệ thống từ xa
- Tìm hiểu một số công cụ (phần mềm) bảo vệ hệ thống
- Tìm hiểu một số kỹ thuật tấn công trên mạng (Vụ tấn công doanh nghiệp TMĐT Viet Co Ltd, vụ tấn công diễn đàn Hacker Việt Nam - HVA)
- Tìm hiểu kỹ thuật bảo mật trong Windows

[2]This content is available online at <http://voer.edu.vn/content/m13180/1.1/>.
[3]This content is available online at <http://voer.edu.vn/content/m13182/1.1/>.

- Tìm hiểu kỹ thuật bảo mật trong Linux
- Tìm hiểu kỹ thuật kiểm soát truy xuất trong bảo vệ mạng nội bộ
- Tìm hiểu vấn đề bảo mật trong mạng không dây và điện thoại di động
- Tìm hiểu hệ mật mã DES
- Tìm hiểu hệ mật mã IDEA
- Tìm hiểu hệ mật mã AES
- Tìm hiểu hệ mật mã RC5
- Tìm hiểu giải thuật chia MD5, SHA
- Xây dựng chương trình DEMO một số hệ mật mã cổ điển
- Xây dựng chương trình DEMO một số hệ mật mã sử dụng khóa công khai
- Bảo mật các chương trình CHAT
- Ứng dụng chữ ký điện tử cho các chương trình Email
- Truy tìm dấu vết trên mạng
- Tìm lỗ hổng của các Website
- Công cụ tấn công từ xa
- Công cụ bảo vệ hệ thống
- Tìm hiểu Spam
- Tìm hiểu Phishing
- Tìm hiểu mạng botnet
- Tìm hiểu Keyloger
- Tìm hiểu Malware
- Tìm hiểu Spyware
- Tìm hiểu Trojan horse
- Tìm hiểu Internet worm
- Tìm hiểu virus Macro
- Tìm hiểu Mobile code
- Tìm hiểu một số kỹ thuật sử dụng trong các chương trình diệt virus
- Xây dựng ngân hàng câu hỏi về Virus
- Phân tích virus
- Bảo vệ an toàn mạng LAN
- Nghiên cứu giải pháp phòng, chống tấn công DDOS
- Tính toán tin cậy
- Thủy vân số và dấu vân tay
- Bảo mật vật lý
- Tấn công truyền hình kỹ thuật số qua vệ tinh
- Xác thực người dùng trong hệ thống file mã hóa
- Giấu tin trong thư rác
- Bảo vệ phần mềm dựa trên việc thực thi
- Giấu tin trong trường TCP timestamps
- Xác thực bảo mật dựa trên danh tiếng
- Xác thực Cookie
- Phân tích cơ chế bảo mật của mạng không dây 802.11
- Các vấn đề bảo mật của Unicode
- Chữ ký điện tử cho thư tay
- Thanh toán qua điện thoại GSM
- Bảo vệ bản quyền truyền thông số
- Bảo mật các hệ thống lưu trữ trên mạng
- Kiểm tra lỗi bảo mật phần mềm
- Thiết kế một hệ thống xác thực thử nghiệm
- Các hệ thống phát hiện xâm nhập
- Bảo mật điện thoại di động

- Hệ thống kiểm tra máy tính
- Kỹ thuật bảo vệ bản quyền trong DVD + DIVx
- Các vấn đề bảo mật trong hệ thống CGI
- Kiểm soát truy xuất trên mạng
- Tiền điện tử - khả năng dung lỗi trong hệ thống ngân hàng
- Mô hình chính sách bảo mật
- Tổng quan về công nghệ sinh trắc học và ứng dụng thực tiễn
- Tìm hiểu giao thức bảo mật Secure Sockets Layer 3.0
- Lược đồ mã hóa All-or-Nothing bảo mật kênh phân phối thông tin đa người dùng
- Tác động của lý thuyết lượng tử tới mật mã
- Bảo mật công nghệ ví điện tử
- Bảo mật trò chơi điện tử Poker
- Tìm hiểu so sánh PGP và S/MIME
- Tìm hiểu SSH
- ATM: Một cái máy tin cậy?
- Khung chính sách bảo mật cho Mobile Code
- Sơ đồ thanh toán điện tử
- Tính toán trên dữ liệu mã hóa
- Bảo mật hệ thống bầu cử tự do
- Tính khả thi của tính toán lượng tử
- Bỏ phiếu điện tử.

Chương 2

Nhận dạng, xác thực và kiểm soát truy xuất

2.1 Nhận dạng và xác thực điện tử[1]

Một hệ thống bảo mật phải có khả năng lưu vết nhân dạng hay danh tính (identifier) của người dùng sử dụng dịch vụ. Xác thực (authenticate) là quá trình kiểm chứng nhân dạng của người dùng. Có hai lý do để làm việc này:

- Nhân dạng người dùng là một tham số trong quyết định kiểm soát truy xuất;
- Nhân dạng người dùng được ghi lại tại bộ phận kiểm soát dấu vết khi đăng nhập vào hệ thống.

Trong thực tế, kiểm soát truy xuất không nhất thiết chỉ dựa trên nhân dạng người dùng nhưng thông tin này được sử dụng rộng rãi trong phần kiểm soát dấu vết. Phần này sẽ trình bày về nhận dạng và xác thực vì chúng là các chuẩn mực trong các hệ thống máy tính ngày nay.

2.1.1 Giao thức xác thực

Trong phần này, chúng ta xem xét cách thức một đối tác xác thực đối tác còn lại khi hai bên thực hiện trao đổi thông tin trên mạng.

Khi thực hiện xác thực trên mạng, người trao đổi thông tin không thể dựa trên các thông tin sinh học chẳng hạn như hình dáng hay giọng nói. Thông thường, việc xác thực diễn ra tại các thành phần của mạng chẳng hạn như router hoặc các quá trình xử lý server/client. Quá trình xác thực chỉ dựa duy nhất vào những thông điệp và dữ liệu được trao đổi như một phần của giao thức xác thực (authentication protocol) [2].

Sau đây, chúng ta xem xét một số giao thức xác thực được ứng dụng trong thực tế. Các giao thức này thường được chạy trước khi người dùng thực hiện các giao thức khác.

[1]This content is available online at <http://voer.edu.vn/content/m13188/1.1/>.

2.1.1.1 Giao thức xác thực ap1.0

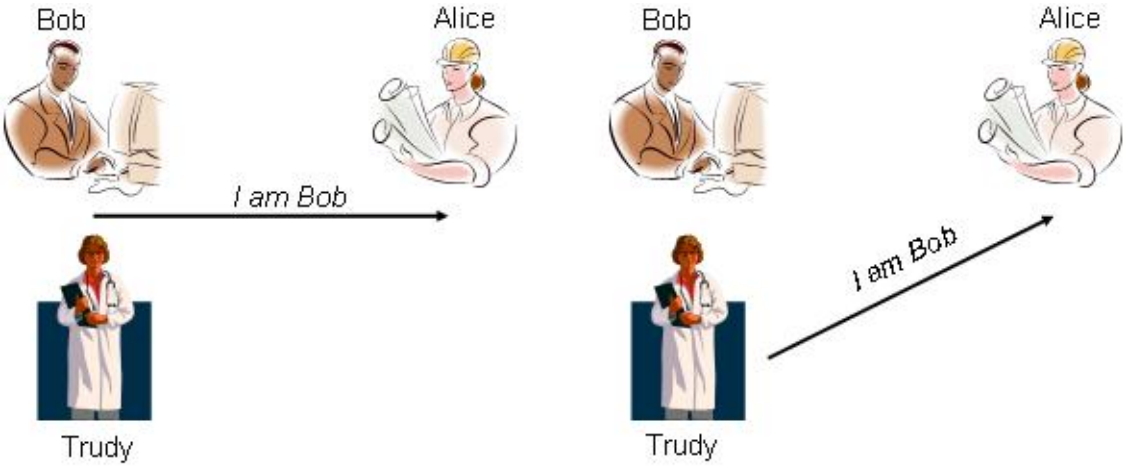


Figure 2.1: Giao thức xác thực 1.0

2.1.1.2 Giao thức xác thực ap2.0

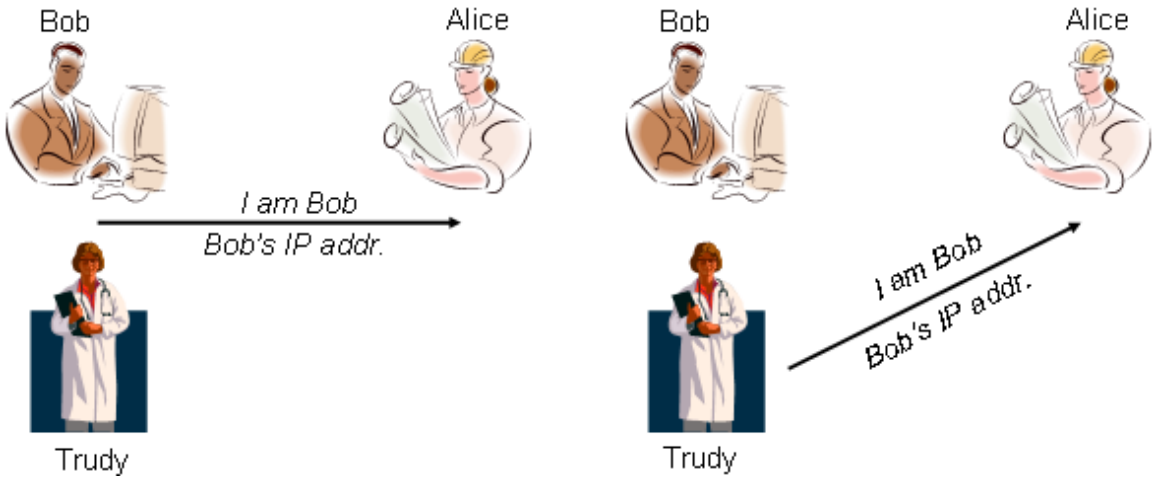


Figure 2.2: Giao thức xác thực 2.0

2.1.1.3 Giao thức xác thực ap3.0

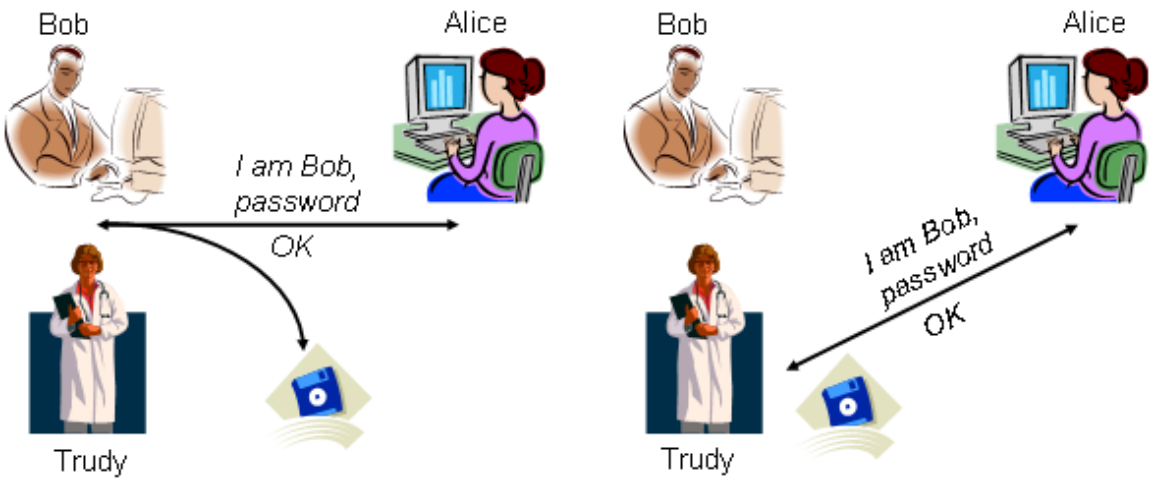


Figure 2.3: Giao thức xác thực 3.0

2.1.1.4 Giao thức xác thực ap3.1

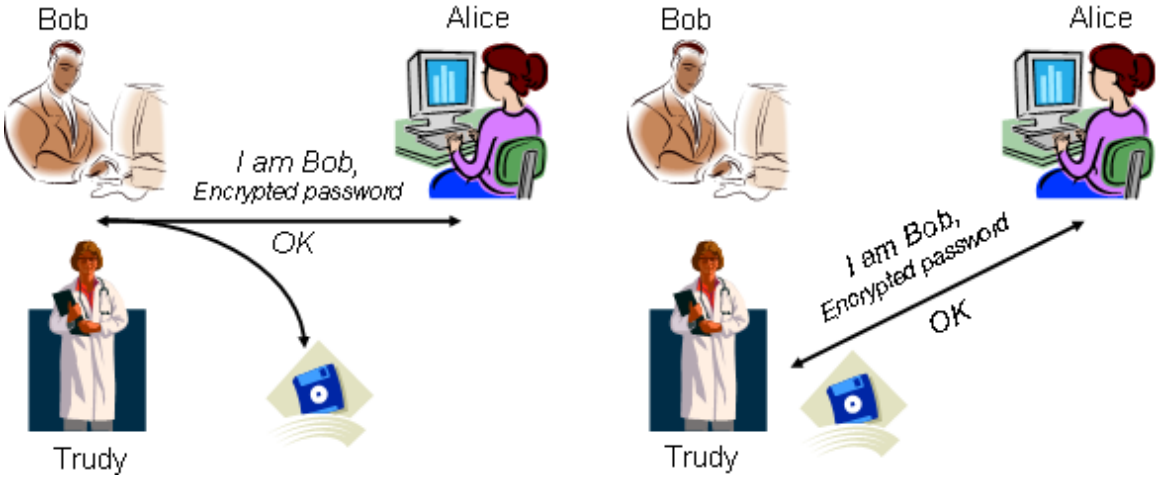


Figure 2.4: Giao thức xác thực 3.1

2.1.1.5 Giao thức xác thực ap4.0

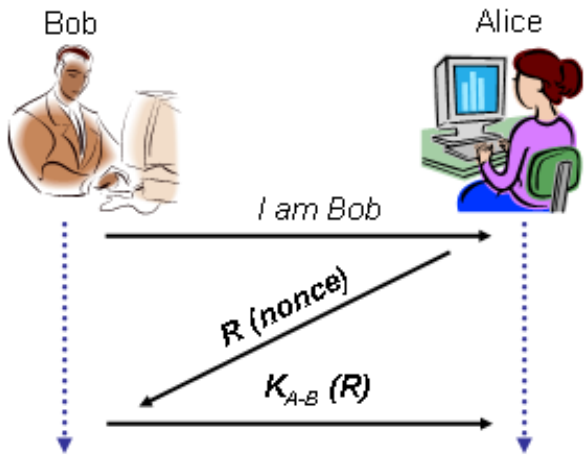


Figure 2.5: Giao thức xác thực 3.1

2.1.1.6 Giao thức xác thực ap5.0

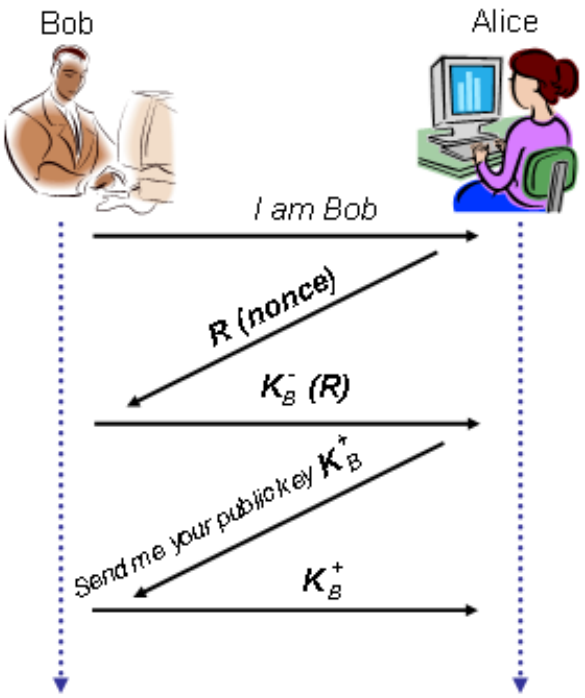


Figure 2.6: Giao thức xác thực 5.0

2.1.2 Tên truy nhập và mật khẩu

Thực tế, chúng ta đã làm quen với khái niệm bảo mật máy tính khi ta thực hiện đăng nhập vào hệ thống sử dụng tài khoản gồm tên truy nhập và mật khẩu bí mật. Bước đầu tiên là nhận dạng, khi đó bạn thông báo mình là ai. Bước thứ hai là xác thực. Bạn chứng minh những gì bạn thông báo.

2.2 Kiểm soát truy suất[2]

2.2.1 Khái niệm

Bảo mật thực chất là kiểm soát truy xuất

Mục đích của bảo mật máy tính là bảo vệ máy tính chống lại việc cố ý sử dụng sai mục đích các chương trình và dữ liệu được lưu trữ trên máy tính. Nguyên lý kỹ thuật để bảo vệ thông tin của hầu hết các hệ thống là kiểm soát truy xuất (access control) [3].

Access control có thể được hình dung như là tình huống trong đó một chủ thể chủ động (subject) truy xuất một đối tượng bị động (object) với một phép truy xuất nào đó. Trong khi một bộ điều khiển tham chiếu (reference monitor) sẽ cho phép hoặc từ chối các yêu cầu truy xuất [1]. Mô hình cơ sở của access control được đưa ra bởi Lampson như hình

[2]This content is available online at <http://voer.edu.vn/content/m13195/1.1/>.

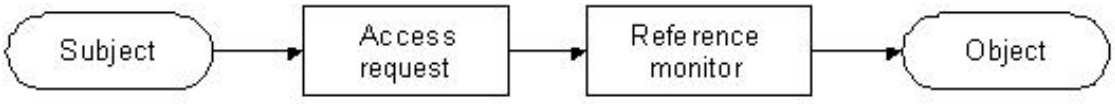


Figure 2.7: Mô hình cơ sở của kiểm soát truy xuất

Trong các hệ thống máy tính, chủ thể là người sử dụng hay các tiến trình. Đối tượng là file, bộ nhớ, các thiết bị ngoại vi, các nút mạng... Các phép truy xuất điển hình là đọc (read), ghi (write), bổ sung (append) và thực thi (execute). Quyền thực hiện một phép truy xuất nhất định trên một đối tượng được gọi là quyền truy xuất (access right). Các luật bảo mật (security policy) được định nghĩa như một bộ điều phối quyền truy xuất cho các chủ thể.

Để biểu diễn kiểm soát truy xuất, trong tài liệu này chúng ta sử dụng các quy ước sau đây:

- S là tập các chủ thể
- O là tập các đối tượng
- A là tập các thao tác

2.2.2 Cài đặt kiểm soát truy xuất

2.2.2.1 Ma trận

Nhìn chung, quyền truy xuất có thể hoàn toàn được định nghĩa đơn giản bằng một ma trận kiểm soát truy xuất.

$M = (M_{\text{so}})_{s \in S, o \in O}$ với $M_{\text{so}} \subset A$.

Điểm vào M_{so} xác định tập các phép truy xuất chủ thể s có thể thực hiện trên đối tượng o. Nhưng trong thực tế, các ma trận kiểm soát truy xuất là một khái niệm trừu tượng và không thực sự phù hợp cho việc cài đặt trực tiếp nếu số lượng chủ thể và đối tượng lớn hoặc các tập này thay đổi thường xuyên [1]. Ví dụ sau đây (lấy từ [1]) sẽ chỉ ra cách thức các ma trận kiểm soát truy xuất được triển khai trong mô hình bảo mật Bell-LaPadula.

Ví dụ : Ma trận kiểm soát truy xuất

Chúng ta sử dụng một bảng để biểu diễn ma trận, trong đó hai người dùng Bob và Alice xử lý ba file, lần lượt là bill.doc, edit.exe và fun.com. Các quyền truy xuất trên các file này có thể được mô tả như sau:

- Bob có quyền đọc hoặc ghi file bill.doc trong khi Alice không có quyền truy xuất.
- Bob và Alice chỉ có quyền thực thi file edit.exe.
- Bob và Alice có quyền thực thi và quyền đọc file fun.com nhưng chỉ có Bob có quyền ghi lên file này.

Bây giờ, chúng ta có một ma trận kiểm soát truy xuất như sau:

Users\files	bill.doc	edit.exe	fun.com
Alice	{}	{execute}	{execute, read}
Bob	{read, write}	{execute}	{execute, read, write}

Figure 2.8: Ma trận kiểm soát truy xuất

2.2.2.2 Khả năng

Phần trước, chúng ta đã chỉ ra hạn chế của việc cài đặt trực tiếp ma trận kiểm soát truy xuất. Để giải quyết vấn đề này, có nhiều giải pháp khả thi đã được đề xuất. Hai trong số các giải pháp được thảo luận trong tài liệu này là khả năng và danh sách kiểm soát truy xuất.

Trong cách tiếp cận theo khả năng, các quyền truy xuất được kết hợp với các chủ thể hay nói cách khác mỗi chủ thể được cấp một khả năng, một thẻ nhớ xác định các quyền truy xuất [1]. Khả năng này tương ứng với các dòng của chủ thể trong ma trận kiểm soát truy xuất. Các quyền truy xuất trong Ví dụ 2.1 bây giờ có thể được biểu diễn theo quan điểm khả năng như sau:

Khả năng của Alice: edit.exe: execute; fun.com: execute, read

Khả năng của Bob: bill.doc: read, write; edit.exe: execute; fun.com: execute, read, write

2.2.2.3 Danh sách kiểm soát truy xuất

Trong danh sách kiểm soát truy xuất (Access Control List - ACL), các quyền truy xuất được lưu trữ tại từng đối tượng [1]. Danh sách kiểm soát truy xuất vì vậy tương ứng với một cột trong ma trận kiểm soát truy xuất và cho biết ai có quyền truy xuất một đối tượng nào đó. Các quyền truy xuất của Ví dụ 2.1 có thể được mô tả theo danh sách kiểm soát truy xuất như sau:

ACL cho bill.doc Bob: read, write

ACL cho edit.exe Bob: execute; Alice: execute

ACL cho fun.com Bob: execute, read, write; Alice: execute, read

2.2.3 Một số cách tiếp cận tới kiểm soát truy xuất

Tổng quát, có hai cách tiếp cận tới kiểm soát truy xuất: *tùy ý (discretionary)* và *bắt buộc (mandatory)*. Các kỹ thuật kiểm soát truy xuất tùy ý dựa trên đặc quyền của người dùng và không mịn (coarse-grained). Các kỹ thuật kiểm soát truy xuất bắt buộc dựa trên đặc tả về các thành phần của phần mềm và mịn hơn (fine-grained) [4]. Chúng ta sẽ thấy kiểm soát truy xuất dựa trên các đặc tả về người dùng dường như không phù hợp trong các môi trường tính toán phân tán vì vậy kiểm soát truy xuất dựa trên đặc tả các thành phần được đề xuất cho trường hợp này. Lý do chính là: trong các môi trường tính toán phân tán, một người dùng có thể chạy nhiều thành phần cấu thành một ứng dụng. Đương nhiên, không phải tất cả các thành phần này có cùng mức độ tin cậy, vì vậy mã được thực thi nhân danh một người dùng không thể đơn giản thừa kế các quyền của người dùng đó, nhưng thay vào đó chúng ta có thể xem xét dựa trên các tính chất của thành phần [5].

Cách tiếp cận nổi bật nhất trong các nghiên cứu hiện tại về kiểm soát truy xuất bắt buộc và mịn là kiểm soát truy xuất theo miền (domain-type enforcement - DTE) và kiểm soát truy xuất theo vai trò (role-based access control - RBAC). Xu hướng phát triển này thực sự đã chi phối các kỹ thuật xử lý các thành phần không tin cậy. Chúng ta sẽ xem xét sự thay đổi này trong bài tiếp theo.

Trong phần sau, chúng tôi giới thiệu một số cách tiếp cận; bài tiếp sẽ mô tả một số định nghĩa hình thức theo ngữ cảnh của các mô hình bảo mật.

Tùy ý

Cơ sở của kiểm soát truy xuất tùy ý (DAC) là mỗi người dùng sở hữu các quyền truy xuất tới thông tin và có thể chuyển giao các quyền này cho những người dùng khác. Điều này có nghĩa là những người sử dụng đó được phép xác định các chính sách bảo mật riêng bằng cách cấp hoặc thu hồi các kiểu truy xuất có thể có đối với thông tin. Nói chung vẫn có một chính sách chung xem xét cách thức tạo ra các chính sách địa phương. Một chính sách như vậy định nghĩa cách một người dùng cấp quyền truy xuất cho người khác, nó còn mô tả những quyền truy xuất nào một số người dùng không được sở hữu.

Có nhiều mô hình bảo mật dựa trên kiểm soát truy xuất tùy ý đã được đề xuất (Ví dụ, mô hình HRU và BLP). Một chính sách DAC cụ thể định nghĩa một tập các quyền truy xuất cho trước – ví dụ, read, write, execute, write như trong mô hình BLP – và cách người dùng được phép cấp lại quyền – ví dụ, trao quyền dựa trên khái niệm sở hữu, đó là, người dùng chỉ có thể cấp hoặc thu hồi những đặc quyền đối với những đối tượng họ sở hữu/tạo ra.

Các mô hình bảo mật DAC tìm cách trả lời câu hỏi về vấn đề an toàn. Những vấn đề này xảy ra bất cứ khi nào việc trao quyền vi phạm chính sách bảo mật chung. Đây là nguyên tắc áp dụng đối với việc trao quyền truy xuất hơn là trao đổi thông tin. Vì vậy, tính bí mật của thông tin không được xem xét [3]. Một mô hình DAC thường có một hoặc một số đặc điểm sau đây [6].

- Người sở hữu dữ liệu có thể cấp quyền sở hữu thông tin cho những người khác
- Người sở hữu dữ liệu có thể xác định kiểu truy xuất để cấp cho những người khác (read, write, copy...)
- Hệ thống cảnh báo hoặc giới hạn truy xuất của người dùng trong trường hợp yêu cầu truy xuất tới tài nguyên hoặc đối tượng không đáp ứng quá trình xác thực (thường là một số lần)
- Một phần mềm tăng cường (add-on) hoặc bổ sung (plug-in) áp dụng cho một máy khách để ngăn ngừa người dùng sao chép thông tin
- Người dùng không có quyền truy xuất thông tin không thể xác định được các đặc điểm của nó (kích thước, tên, đường dẫn của file...)
- Việc truy xuất tới thông tin được xác định dựa trên quyền hợp pháp mô tả trong danh sách kiểm soát truy xuất theo danh tính người dùng và nhóm

2.2.3.1 Bắt buộc

Kiểm soát truy xuất bắt buộc (mandatory access control - MAC) bao gồm cả các khía cạnh người dùng không thể kiểm soát (hoặc thường là không được phép kiểm soát). Trong MAC, các đối tượng được gắn nhãn mô tả sự nhạy cảm của thông tin bên trong nó. MAC giới hạn truy xuất tới các đối tượng dựa trên sự nhạy cảm của chúng. Các chủ thể cần có giấy phép chính thức (được cấp phép) mới được truy xuất tới các đối tượng [3].

Nói chung, kỹ thuật kiểm soát truy xuất bắt buộc MAC bảo mật hơn DAC và đảm bảo sự cân đối giữa hiệu năng sử dụng và sự thuận tiện đối với người dùng. Kỹ thuật MAC cấp một mức bảo mật cho tất cả các thông tin, cấp một giấy phép bảo mật cho mỗi người dùng và bảo đảm rằng tất cả người dùng chỉ có truy xuất tới dữ liệu mà họ có giấy phép. MAC thường phù hợp với những hệ thống cực mật bao gồm các ứng dụng quân sự có nhiều mức bảo mật hoặc các ứng dụng dữ liệu quan trọng. Một mô hình MAC thường có một hoặc một số đặc điểm sau đây [6].

- Chỉ có những người quản trị, không phải là người sở hữu dữ liệu, có thể thay đổi nhãn bảo mật của một tài nguyên.
- Tất cả dữ liệu được cấp/chỉ định mức bảo mật tương ứng với sự nhạy cảm, tính bí mật và giá trị của nó.
- Người dùng có thể đọc thông tin từ lớp bảo mật thấp hơn mức bảo mật họ được cấp (Một người dùng "bảo mật" có thể đọc một tài liệu không được phân loại).
- Người dùng có thể ghi lên thông tin thuộc lớp bảo mật cao hơn (Một người dùng "bảo mật" có thể xuất bản thông tin lên mức bảo mật cao nhất).

- Người dùng chỉ được cấp quyền đọc/ghi đối với những đối tượng có cùng mức bảo mật (một người dùng "bảo mật" chỉ có thể đọc/ghi một tài liệu bảo mật).
- Truy xuất tới các đối tượng được cấp phép hoặc bị giới hạn theo thời gian phụ thuộc vào nhãn được gắn với tài nguyên và giấy phép của người dùng (áp đặt bởi chính sách).
- Truy xuất tới các đối tượng được cấp phép hoặc bị giới hạn dựa trên các đặc tính bảo mật của máy khách (ví dụ, độ dài theo bit của SSL, thông tin version, địa chỉ IP gốc hoặc domain...)

2.2.3.2 Kiểm soát truy xuất theo vai trò

Trong kiểm soát truy xuất theo vai trò (role-based access control - RBAC), quyết định truy xuất được dựa trên các vai trò và trách nhiệm riêng rẽ bên trong tổ chức hoặc của cá nhân. Quá trình định nghĩa các vai trò thường dựa trên việc phân tích mục tiêu và cấu trúc của tổ chức nhưng kết nối tới các chính sách bảo mật.

Những khía cạnh sau đây thể hiện các đặc điểm của RBAC cấu thành một mô hình kiểm soát truy xuất [6].

- Các vai trò được cấp phát dựa trên cấu trúc tổ chức với sự nhấn mạnh đặc biệt về cấu trúc bảo mật.
- Các vai trò được cấp phát bởi người quản trị dựa trên các mối quan hệ nội tại của tổ chức hoặc cá nhân. Ví dụ, một người quản lý có thể có các giao dịch được cấp phép với nhân viên của anh ta. Một người quản trị có thể có các giao dịch được cấp phép trong phạm vi quản lý của mình (sao lưu, tạo tài khoản...).
- Mỗi vai trò được chỉ định rõ một hồ sơ bao gồm tất cả các câu lệnh, giao dịch và các truy xuất hợp pháp tới thông tin.
- Các vai trò được cấp quyền hạn dựa trên nguyên lý đặc quyền tối thiểu (the principle of least privilege).
- Các vai trò được xác định với các nhiệm vụ khác nhau do đó người có vai trò developer sẽ không thực hiện các nhiệm vụ của vai trò tester.
- Các vai trò được kích hoạt tĩnh hoặc động tùy thuộc vào những sự kiện kích hoạt có liên quan (hàng đợi trợ giúp, cảnh báo bảo mật, khởi tạo một project...).
- Các vai trò chỉ có thể được chuyển giao hoặc ủy quyền khi sử dụng một quy trình và thủ tục nghiêm ngặt.
- Các vai trò được quản lý tập trung bởi một người quản trị bảo mật hoặc trưởng dự án.

Chương 3

Các mô hình bảo mật[1]

Để xây dựng các chính sách bảo mật, chúng ta phải mô tả các thực thể bị chi phối bởi các chính sách và chúng ta phải phát biểu các quy tắc cấu thành nên chính sách đó. Công cụ để làm việc này là mô hình bảo mật [1]. Trong phần này, chúng ta sẽ tập trung vào ba mô hình bảo mật điển hình: mô hình bí mật, mô hình toàn vẹn và mô hình hỗn hợp.

3.1 Các định nghĩa cơ sở

Về cơ bản, bảo mật thông tin được định nghĩa dựa trên các chính sách về *bí mật* và *toàn vẹn* trong ngữ cảnh một mô hình chuyển trạng thái trừu tượng của một hệ thống bảo vệ. Bảo mật *Luồng thông tin* là hệ quả và về bản chất có liên quan tới sự bí mật

Tính bí mật liên quan đến việc che giấu thông tin và ngăn ngừa việc truy xuất trái phép tới tài nguyên. Tính toàn vẹn liên quan đến việc ngăn ngừa sửa đổi trái phép. Chúng được định nghĩa hình thức như sau (Matt, 2003, được trích dẫn trong

• Bí mật: một chính sách bí mật P_C trên một tập con C O của các đối tượng chia tập các chủ thể S thành hai tập S_C và $\overline{S_C}$. Các chủ thể trong $\overline{S_C}$ không biết về sự tồn tại của C hoặc các thông tin trong C hoặc chúng cũng không thể truy xuất tới nó sử dụng bất cứ quyền nào trong $M_{s'o}$, ∀s' $\overline{S_C}$. P_C rõ ràng xác định các quyền r mà các chủ thể s S_C có thể sử dụng để lấy thông tin xác định từ C.

• Toàn vẹn: một chính sách toàn vẹn P_I trên một tập con I O của các đối tượng chia tập chủ thể S thành hai tập S_I và $\overline{S_I}$. Các đối tượng trong $\overline{S_I}$ không được phép sửa đổi thông tin trong I. P_I rõ ràng xác định các quyền r mà các chủ thể s S_I có thể sử dụng để sửa đổi thông tin trong I. Các thay đổi có thể được thực hiện bởi bất kỳ thực thể nào trong S_I được tất cả các thực thể trong S_I tin tưởng.

Chính sách luồng thông tin là một khía cạnh khác trong bảo vệ thông tin [7]. Trong các phần tiếp theo, chúng ta sẽ xem xét các mô hình bảo mật và chỉ ra luồng thông tin của mỗi mô hình.

Ma trận truy xuất M chứa những quyền truy xuất, thao tác truy xuất được phép nếu quyền có trong hệ thống và thao tác là hợp lệ. Việc này có thể thực hiện được bởi cả chính sách về bí mật và toàn vẹn. Kiểm soát truy xuất an toàn theo ngữ cảnh của ma trận truy xuất được xác định bởi định nghĩa sau đây (sử dụng từ [7]).

Định nghĩa (Kiểm soát truy xuất an toàn).
$((rM_{s,o} \leftrightarrow (s, o, r)\ \ P)\ (allow\_access(s, o, r) \leftrightarrow r\ M_{s,o}))$

3.2 Máy trạng thái (state machine)

Một mô hình bảo mật gồm hai phần, phần thứ nhất là mô hình tổng quan của hệ thống máy tính và phần thứ hai cung cấp định nghĩa về bảo mật. Thông thường, các hệ thống được biểu diễn bằng một mô hình

[1]This content is available online at <http://voer.edu.vn/content/m13269/1.1/>.

dạng máy trạng thái [3]. Trong mô hình máy trạng thái (hay ôtô mát), mỗi trạng thái biểu diễn một trạng thái của hệ thống. Đầu ra của ôtômat phụ thuộc đầu vào và phép biến đổi trạng thái. Các phép biến đổi trạng thái có thể được định định nghĩa bằng một hàm biến đổi trạng thái. Hàm này xác định trạng thái tiếp theo phụ thuộc vào trạng thái hiện tại và đầu vào [1]. Chúng ta đang nói về các mô hình bảo mật vì vậy mối quan tâm của ta là làm thế nào bảo đảm rằng tất cả các trạng thái được sinh ra bởi ôtô mát là an toàn hay bảo mật. Trong phần tiếp theo, các mô hình bảo mật sẽ được xem xét cẩn thận vì mục đích này. Với mỗi mô hình, công việc của chúng ta là xác định các trạng thái an toàn hay bảo mật.

Máy trạng thái được định nghĩa hình thức như sau

Một máy trạng thái là bộ bốn $< \delta_0, \Delta, \Gamma, \tau >$ sao cho

- Δ là tập các trạng thái
- $\delta_0 \in \Delta$ là trạng thái bắt đầu
- Γ là tập các ký hiệu thao tác sao cho mỗi $\gamma \in \Gamma$, τ_γ là một hàm từ $\Gamma \times \Delta$ vào Δ.

Một quy tắc bảo mật là một tập $P \subseteq \Delta$.

Một trạng thái δ được gọi là đến được từ δ_0 nếu $\delta = \delta_0$ hoặc có một dãy các thao tác $\gamma_1, ..., \gamma_n$ sao cho $\delta = \tau_{\gamma_n}\left(\tau_{\gamma_{n-1}}\left(...\tau_{\gamma_1}\left(\delta_0\right)\right)\right)$.

Định nghĩa (ôtô mát bảo mật)một máy trạng thái $< \delta_0, \Delta, \Gamma, \tau >$ được gọi là bảo mật (đối với quy tắc bảo mật P) nếu với mỗi $\delta \in \Delta$, nếu δ là đến được từ δ_0 thì $\delta \in P$.

3.3 Mô hình bí mật

Một quy tắc bí mật có thể được ví như việc định nghĩa nhiều lớp thông tin khác nhau tồn tại trong hệ thống và cách thông tin được trao đổi giữa các lớp này [3].

Năm 1975, Bell và Lapadula hình thức hóa mô hình bảo mật đa cấp MAC (sau này được gọi là mô hình BLP). BLP là một mô hình máy trạng thái kiểm soát các yếu tố bí mật trong kiểm soát truy xuất. Các quyền hạn truy xuất được định nghĩa thông qua cả ma trận kiểm soát truy xuất và các mức bảo mật. Các quy tắc bảo mật ngăn ngừa thông tin rò rỉ từ mức bảo mật cao xuống mức thấp [1]. Hình 3.1 mô tả một trạng thái trong mô hình BLP.

Để biểu diễn mô hình BLP, chúng ta sử dụng các ký hiệu sau đây

- S là tập các chủ thể;
- O là tập các đối tượng;
- A={execute, read, append, write} là tập các quyền truy xuất;
- L là tập các mức bảo mật với phép quan hệ thứ tự bộ phận ;
- Một trạng thái được định nghĩa là một bộ ba (b, M, f) trong đó:

- b là một bộ ba (s, o, a), mô tả chủ thể s hiện tại đang thực hiện thao tác a trên đối tượng o.

- M là một ma trận kiểm soát truy xuất M= $(M_{\text{so}})_{s \in S, o \in O}$.

- f = (f_S, f_C, f_O), trong đó:

- $f_S : S \to L$ cho biết mức bảo mật cao nhất mỗi chủ thể có thể có.

- $f_C : S \to L$ cho biết mức bảo mật hiện tại của mỗi chủ thể, chúng ta luôn luôn có: $f_C(s) \leq f_S(s)$ hoặc viết là " $f_S$chi phối f_C".

- $f_O : O \to L$ cho biết mức bảo mật của mỗi đối tượng

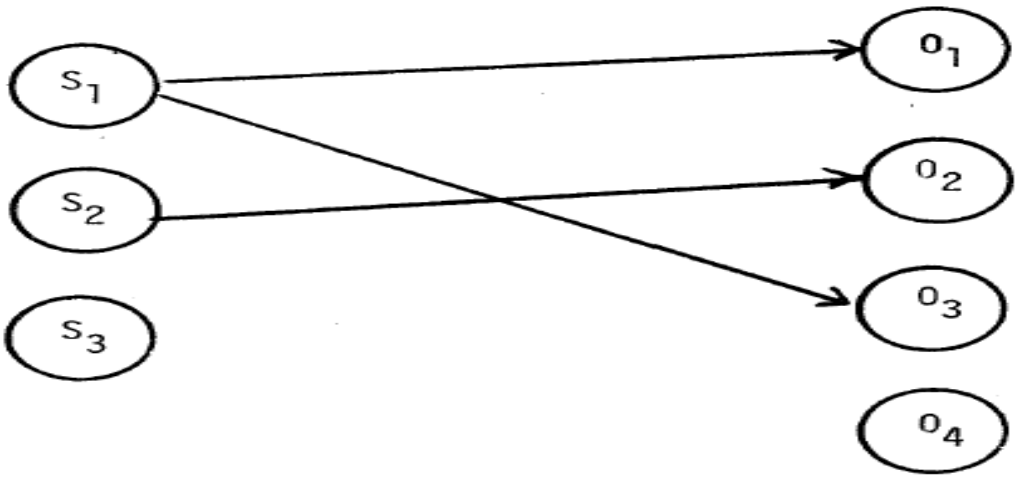


Figure 3.1: Các chủ thể truy xuất các đối tượng [9]

BLP định nghĩa bảo mật qua tính chất của các trạng thái. Tính chất thứ nhất là tính chất bảo mật đơn giản (simple security property), ký hiệu là ss-property.

ss-properties Một trạng thái (b, M, f) thỏa mãn tính chất ss-property, nếu mỗi phần tử (s, o, a) b, thao tác truy xuất a là read hoặc write, mức bảo mật của chủ thể s chi phối lớp đối tượng o, nghĩa là: $f_O(o) \leq f_S(s)$. Đặc trưng này đáp ứng được chính sách bảo mật truyền thống *no read-up*.

Tuy nhiên, tính chất ss-property không đảm bảo ngăn ngừa việc chủ thể ở mức bảo mật thấp đọc nội dung của một đối tượng có mức bảo mật cao Điều này phát sinh yêu cầu về tính chất khác, gọi là tính chất sao (star property), ký hiệu là *-property [1].

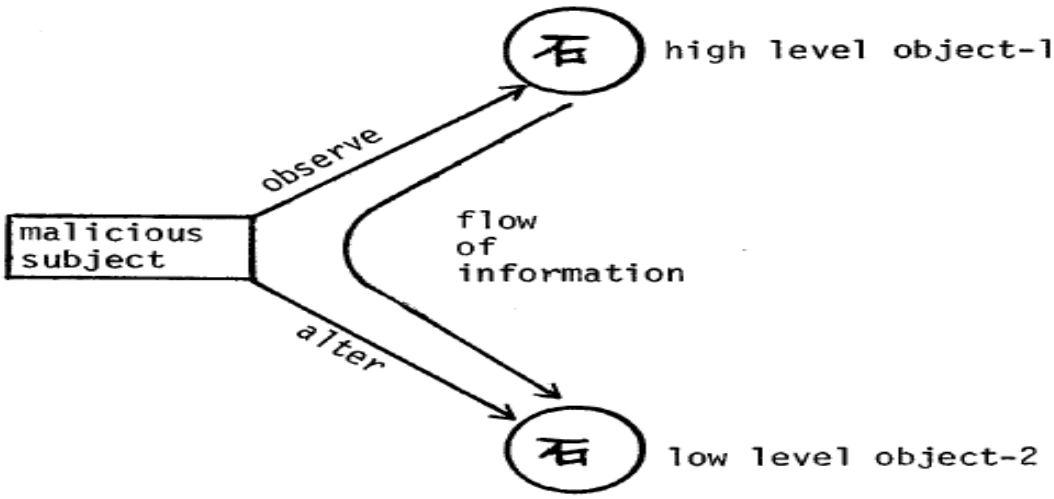


Figure 3.2: Luồng thông tin đòi hỏi *-property

***-properties** Một trạng thái (b, M, f) thỏa mãn tính chất *-property, nếu mỗi phần tử (s, o, a) b, thao tác truy xuất a là append hoặc write, mức bảo mật hiện tại của chủ thể s bị chi phối bởi lớp đối tượng o, nghĩa là: $f_C(s) \leq f_O(o)$. Đây là một chính sách no write-down. Hơn nữa, nếu có tồn tại một phần tử (s, o, a) b, a là append hoặc write, thì chúng ta phải có $f_O(o') \leq f_O(o)$ với mọi đối tượng o' mà (s, o', a') b và a' là read hoặc write.

Các chủ thể nắm giữ các quyền truy xuất có thể cấp lại các quyền này cho các chủ thể khác. Trong mô hình BLP, những chính sách như vậy có thể được đặc tả bằng một ma trận kiểm soát truy xuất và phải tuân thủ tính chất bảo mật tùy ý (discretionary security property), ký hiệu là ds-property.

ds-properties Một trạng thái (b, M, f) thỏa mãn tính chất ds-property, nếu mỗi phần tử (s, o, a) b chúng ta có $a \in M_{so}$.

Định nghĩa (trạng thái bảo mật) Một trạng thái được gọi là bảo mật nếu cả ba tính chất bảo mật đều được thỏa mãn.

Bell và LaPadula đề xuất và chứng minh định lý cơ bản về bảo mật sau đây [1].

Định lý Nếu mọi phép biến đổi trạng thái trong một hệ thống là bảo mật và trạng thái ban đầu là bảo mật thì với đầu vào tùy ý, mọi trạng thái sinh ra là bảo mật.

Luồng thông tin

Để kiểm tra một hệ thống bảo mật (theo mô hình BLP), chúng ta cần kiểm tra trạng thái mới (b', M', f') được sinh ra từ trạng thái (b, M, f) có bảo mật hay không. Ta xem xét một số khái niệm sau đây.

Ta nói rằng có một luồng thông tin hợp lý trực tiếp từ một đối tượng o tới đối tượng o' nếu có một chủ thể s sao cho hai điều kiện sau đây được thỏa.

1. (s, o, a) b và a là observe.
2. (s, o', a') b và a' là alter.

Một dãy $o_1, ..., o_n$ được gọi là luồng thông tin hợp lý nếu có một luồng thông tin hợp lý trực tiếp từ o_i tới o_{i+1} với $1 \leq i < n$.

Chúng ta nói có một luồng thông tin hợp lý từ đối tượng o tới đối tượng o' nếu có một luồng thông tin hợp lý $o_1, ..., o_n$ sao cho $o = o_1, o' = o_n$.

Định nghĩa (Luồng thông tin an toàn) Một luồng thông tin hợp $o_1, ..., o_n$ được gọi là an toàn nếu $f_O(o_1) \leq f_O(o_n)$.

3.4 Mô hình toàn vẹn

Năm 1977, Biba đề xuất một mô hình (sau này được gọi là mô hình Biba) xử lý tính toàn vẹn của hệ thống khi các chủ thể thực hiện truy xuất các đối tượng sử dụng mô hình máy bảo mật tương tự như mô hình BLP [1]. Để diễn tả mô hình Biba, chúng ta sử dụng các quy ước sau đây (được đề xuất và sử dụng trong [8]):

- S là tập các chủ thể;
- O là tập các đối tượng;
- L là tập các mức toàn vẹn với phép sắp thứ tự bộ phận ;
- $f_S : S \to L$ cho biết mức toàn vẹn của mỗi chủ thể;
- $f_O : O \to L$ cho biết mức bảo mật của mỗi đối tượng.

Hàm f_S và f_O chỉ định mức toàn vẹn cho các chủ thể và các đối tượng. Những mức bảo mật này là cơ sở để mô tả các tính chất toàn vẹn để ngăn ngừa các thao tác kiểu như việc "làm sạch" các thực thể ở mức cao bằng cách "làm bẩn" các thực thể ở mức thấp. Chúng ta có thể phát biểu các tính chất trong hai trường hợp, mức toàn vẹn cố định trong đó mức toàn vẹn không thay đổi và mức toàn vẹn biến đổi ứng với mức toàn vẹn có thể thay đổi được.

Các mức toàn vẹn cố định

Hai tính chất toàn vẹn sau đây ngăn ngừa việc làm vấy bẩn các chủ thể và các đối tượng bằng các thông tin bẩn.

Simple integrity property Nếu chủ thể s có thể sửa đổi (biến đổi) đối tượng o, thì $f_S(s) \geq f_O(o)$. Đây là quy tắc không ghi-lên (*no write-up).*

Integrity *-property Nếu chủ thể s có thể đọc (quan sát) đối tượng o, thì s chỉ có thể ghi lên đối tượng p nếu $f_O(p) \leq f_O(o)$.

Luồng thông tin

Chúng ta nói có một luồng thông tin hợp lý trực tiếp từ đối tượng o tới đối tượng o' nếu có một chủ thể s sao cho s có thể quan sát o và biến đổi o'.

Một dãy $o_1, ..., o_n$ được gọi là một luồng thông tin hợp lý nếu có một luồng thông tin hợp lý trực tiếp từ o_i tới o_{i+1} với mỗi $1 \leq i < n$.

Chúng ta nói có một luồng thông tin hợp lý từ đối tượng o tới đối tượng o' nếu có một luồng thông tin hợp lý $o_1, ..., o_n$ sao cho $o = o_1, o' = o_n$.

Định nghĩa Một luồng thông tin $o_1, ..., o_n$ được gọi là an toàn nếu $f_O(o_1) \geq f_O(o_n)$.

Các mức toàn vẹn biến đổi

Hai tính chất sau đây giúp tự động điều chỉnh mức toàn vẹn của một thực thể nếu nó thực hiện tương tác với thông tin ở mức thấp hơn.

Subject low watermark property chủ thể s có thể đọc (quan sát) một đối tượng o tại bất kỳ mức toàn vẹn nào. Mức toàn vẹn mới của chủ thể là $\inf(f_S(s), f_O(o))$, trong đó $f_S(s)$ và $f_O(o)$ là các mức toàn vẹn trước khi thao tác được thực hiện.

Object low watermark property chủ thể s có thể sửa (biến đổi) đối tượng o tại bất kỳ mức toàn vẹn nào. Mức toàn vẹn mới của đối tượng là $\inf(f_S(s), f_O(o))$, trong đó $f_S(s)$ và $f_O(o)$ là các mức toàn vẹn trước khi thao tác được thực hiện.

Luồng thông tin

Chúng ta nói rằng có một luồng thông tin trực tiếp từ đối tượng o tới đối tượng o', ký hiệu bởi (s, o, o') nếu có một chủ thể trước hết quan sát o và sau đó biến đổi o'.

Một dãy $o_1, s_1, o_2, ..., o_{n-1}, s_{n-1}, o_n$ được gọi là một luồng thông tin nếu (o_i, s_i, o_{i+1}) là luồng thông trực tiếp với mọi $1 \leq i < n$.

Chúng ta nói có một luồng thông tin từ một đối tượng o tới đối tượng o' nếu có tồn tại một luồng thông tin $o_1, s_1, o_2, ..., o_{n-1}, s_{n-1}, o_n$ sao cho $o = o_1, o' = o_n$.

Cho $a \in \text{read}, \text{write}$. Theo các quy tắc toàn vẹn biến đổi, khi một chủ thể thực hiện một thao tác truy xuất a, f_S hoặc f_O có thể bị thay đổi. Sự thay đổi này được biểu diễn bởi ánh xạ α: $\alpha(s, o, \text{read}, f_S, f_O) = f'_S$

trong đó f'_S được định nghĩa như sau: với mỗi chủ thể r:

$f'_S(r) = f_S(r)$ nếu $r \neq s$

$f'_S(r) = \inf(f_S(s), f_O(o))$ nếu $r = s$
Tương tự chúng ta có thể định nghĩa:

$$\alpha(s, o, \text{write}, f_S, f_O) = f'_O \tag{3.1}$$

trong đó f'_O được định nghĩa như sau: với mỗi đối tượng o':
$f'_O(o') = f_O(o')$ nếu $o' \neq o$
$f'_O(o') = \inf(f_S(s), f_O(o))$ nếu $o' = o$
Gọi s là chủ thể đã thực hiện đọc từ o và ghi vào o'. Định nghĩa:

$$\Omega(s, o, o', f_S, f_O) = (f'_S, f'_O) \tag{3.2}$$

trong đó $f'_S = \alpha(s, o, \text{read}, f_S, f_O)$ và $f'_O = \alpha(s, o', \text{read}, f'_S, f_O)$.

Định nghĩa Một luồng thông tin $o_1, s_1, o_2, ..., o_{n-1}, s_{n-1}, o_n$ được gọi là an toàn nếu $f_{O,n}(o_1) \geq f_{O,n}(o_n)$. Trong đó:

$$(f_{S,1}, f_{O,1}) = \Omega(s, o_1, o_2, f_S, f_O) \tag{3.3}$$

$$(f_{S,i+1}, f_{O,i+1}) = \Omega(s_i, o_i, o_{i+1}, f_{S,i}, f_{O,i}) \tag{3.4}$$

Thực tế, một chủ thể có thể triệu gọi thực thể khác, ví dụ, một tiến trình gọi một tiến trình khác trong lúc đang thực thi. Mô hình Biba có thể được mở rộng để xử lý thao tác dạng này [1]. Chúng ta xét hai tính chất sau.

Invoke property chủ thể s_1 chỉ có thể triệu gọi chủ thể s_2 nếu $f_S(s_2) \leq f_S(s_1)$.

Ring property một chủ thể s_1 có thể đọc các đối tượng ở tất cả các mức toàn vẹn. Nó chỉ có thể sửa đổi đối tượng o với $f_O(o) \leq f_S(s)$ và nó chỉ có thể triệu gọi chủ thể s_2 nếu $f_S(s_1) \leq f_S(s_2)$.

3.5 Mô hình bảo mật hỗn hợp và kiểm soát truy xuất theo vai trò

Như chúng ta đã thảo luận, các mô hình có khả năng định nghĩa thuần túy hoặc cho vấn đề toàn vẹn hoặc bí mật. Hầu hết các hệ thống bảo mật thông tin yêu cầu kết hợp cả hai loại quy tắc bảo mật này. Vì vậy, các nhà nghiên cứu đã đề xuất các mô hình hỗn hợp có khả năng kiểm soát cả vấn đề toàn vẹn và bí mật trong cùng một ngữ cảnh. Hai mô hình nổi tiếng nhất trong số này là mô hình Chinese-Wall và mô hình kiểm soát truy xuất theo vai trò.

Mô hình Chinese-Wall phát triển khái niệm xung đột giữa các nhóm lợi ích và phân chia các chủ thể và các đối tượng thành các nhóm tương ứng khác nhau. Nó còn định nghĩa một cơ cấu để kiểm soát ý niệm hành vi trong quá khứ (hay lịch sử) tác động tới việc truy xuất thông tin trong tương lai như thế nào trong cơ cấu này [3].

Để diễn tả mô hình Chinese-Wall, chúng ta sử dụng các quy ước sau đây (được đề xuất và sử dụng trong [8]):

- C là một tập các tổ chức;
- O là tập các đối tượng;
- $y : O \to C$ kết hợp mỗi tài liệu với chủ sở hữu.
- $x : O \to 2^C$ là một ánh xạ kết hợp mỗi đối tượng với tập các tổ chức không được phép biết thông tin về nó.
- Nhãn bảo mật của mỗi đối tượng là (x(o), y(o))
- Một ma trận $N = (N_{\text{so}})_{s \in C, o \in O}$ sao cho $N_{\text{so}} = \text{true}$ nếu và chỉ nếu s đã từng truy xuất tới o, ngược lại $N_{\text{so}} = \text{false}$.

Các quy tắc bảo mật trong mô hình Chinese-Wall được định nghĩa như sau.

ss-property một chủ thể s sẽ chỉ được phép truy xuất tới một đối tượng o nếu với mọi đối tượng o' có $N_{s,o'}$= true, y(o) x(o') hoặc y(o)=y(o').

weak *-property một chủ thể s có quyền ghi lên một đối tượng o và có quyền đọc một đối tượng o' thì: y(o) = y(o') hoặc x(o') = θ.

strong *-property một chủ thể s được cấp quyền ghi lên một đối tượng o và quyền đọc một đối tượng o' thì: (y(o) = y(o') và (x(o) θ nếu x(o') θ)) hoặc x(o') = θ.

perfect *-property một chủ thể s được cấp quyền ghi lên một đối tượng o và quyền đọc tới một đối tượng o' thì: (y (o) = y(o') và (x(o') x(o) θ)) hoặc x(o') = θ.

Luồng thông tin

Chúng ta nói có một luồng thông tin nhạy cảm hợp lý trực tiếp từ một đối tượng o tới đối tượng o' nếu có một chủ thể s sao cho s có quan sát o và biến đổi o'.

Một dãy $o_1, ..., o_n$ được gọi là một luồng thông tin nhạy cảm hợp lý nếu có một luồng thông tin hợp lý trực tiếp từ o_i tới o_{i+1} với mỗi $1 \leq i < n$.

Chúng ta nói có một luồng thông tin nhạy cảm hợp lý từ một đối tượng o tới đối tượng o' nếu có một luồng thông tin hợp lý $o_1, ..., o_n$ sao cho $o = o_1, o' = o_n$.

Định nghĩa Một luồng thông tin nhạy cảm hợp lý $o_1, ..., o_n$ được gọi là an toàn nếu $y(o_n) \notin x(o_1)$.

Kiểm soát truy xuất trên cơ sở vai trò (RBAC) được xem là mô hình hỗn hợp phổ biến nhất trong công nghiệp. Khái niệm cốt lõi trong RBAC là vai trò – đại diện cho một nhóm người dùng. Mỗi vai trò được kết hợp với một tập các quyền hạn là các quyền thao tác trên các đối tượng. Những vai trò này có thể được tổ chức theo cấu trúc phân cấp để phản ánh sự phân cấp của người sử dụng trong một hệ thống. RBAC duy trì hai ánh xạ: cấp phát người dùng (user assignment - UA) và cấp phát quyền hạn (permission assignment - PA). Hai ánh xạ này có thể được cập nhật độc lập và sự linh hoạt này cung cấp cho người quản trị một kỹ thuật hiệu quả để quản lý và quản trị những quy tắc kiểm soát truy xuất [7].

Công việc quản trị an ninh trong những hệ thống lớn là phức tạp nhưng có thể đơn giản hóa bằng cách áp dụng mô hình RBAC [10]. Để chỉ ra cách làm việc của RBAC, ROO đã đề xuất một họ bốn mô hình khái niệm Hình biểu diễn mô hình quan hệ còn Hình 3.3(b) miêu tả những đặc điểm thiết yếu của nó. $RBAC_0$, xem như mô hình cơ sở nằm bên dưới, là yêu cầu tối thiểu đối với một hệ thống RBAC. Mô hình tiên tiến $RBAC_1$ và $RBAC_2$ bao trùm $RBAC_0$, nhưng chúng lần lượt có thêm sự phân cấp vai trò (các tình huống trong đó các vai trò có thể thừa kế các quyền hạn từ các vai trò khác) và các ràng buộc (giúp áp đặt các giới hạn trên các cấu hình có thể có của các thành phần khác nhau thuộc RBAC). Mô hình hợp nhất, $RBAC_3$, hàm chứa cả $RBAC_1$, $RBAC_2$ và $RBAC_0$ (theo tính chất bắc cầu).

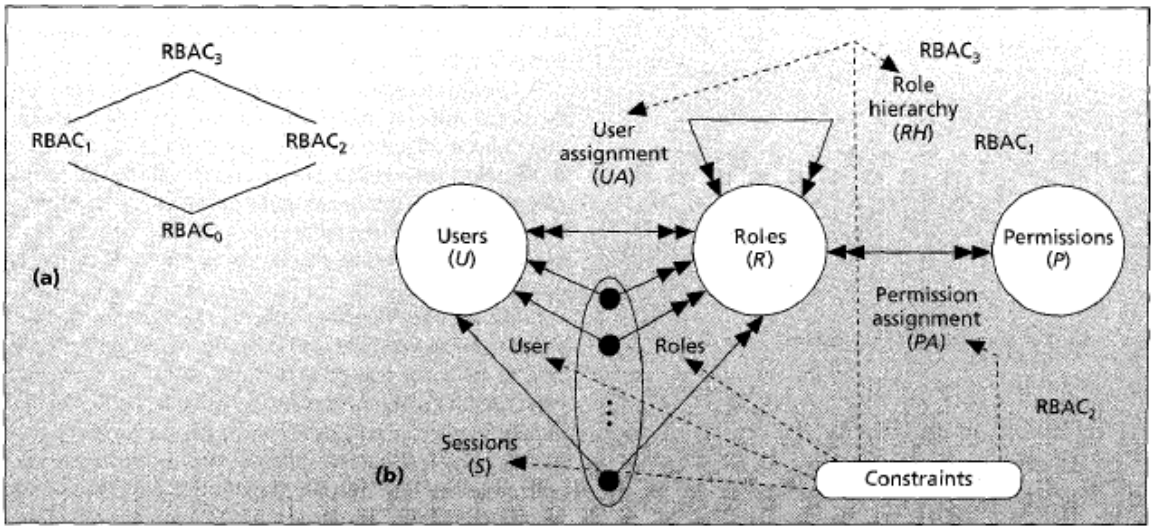


Figure 3.3: Một họ các mô hình kiểm soát truy xuất theo vai trò

Để rõ thêm (chi tiết xem [10]), chúng ta xem xét định nghĩa hình thức sau đây về RBAC [10].
Định nghĩa
Mô hình RBAC có các thành phần sau:

- U, R, P, và S (người sử dụng, vai trò, quyền hạn và phiên làm việc);
- PA $\subseteq P \times R$, một quan hệ nhiều-nhiều cấp phát quyền hạn cho vai trò;
- UA $\subseteq U \times R$, một quan hệ nhiều-nhiều cấp phát người sử dụng cho vai trò;
- RH $\subseteq R \times R$, một quan hệ thứ tự bộ phận trên R được gọi là quan hệ phân cấp hay quan hệ chi phối vai trò, còn được viết là ;
- user : $S \rightarrow U$, một hàm ánh xạ mỗi phiên làm việc s_i tới một người sử dụng đơn lẻ user (s_i)(không đổi trong suốt phiên làm việc); và
- roles : $S \rightarrow 2^R$ được biến đổi từ RBAC_0 để yêu cầu roles $(s_i) \subseteq \{r \mid (\exists r' \geq r)\,[(\text{user}\,(s_i)\,, r') \in \text{UA}]\}$ (có thể thay đổi theo thời gian) và phiên s_i có quyền hạn $_{r \in \text{roles}(s_i)}\{p \mid (\exists r'' \leq r)\,[(p, r'') \in \text{PA}]\}$.

Chương 4

Kỹ thuật mật mã

4.1 Định nghĩa hệ thống mật mã[1]

- 4000 năm trước ở Ai cập, con người đã biết sử dụng chữ tượng hình.
- Thượng cổ ở Hy Lạp, mã hiệu đã được sử dụng để đánh dấu nô lệ.
- 2000 năm trước, Julius Caesar là người đầu tiên sử dụng hệ thống mật mã trong trao đổi thông tin.

Tồn tại nhiều hệ mật mã khác nhau, tuy nhiên có thể biểu diễn bằng mô hình tổng quát như sau:

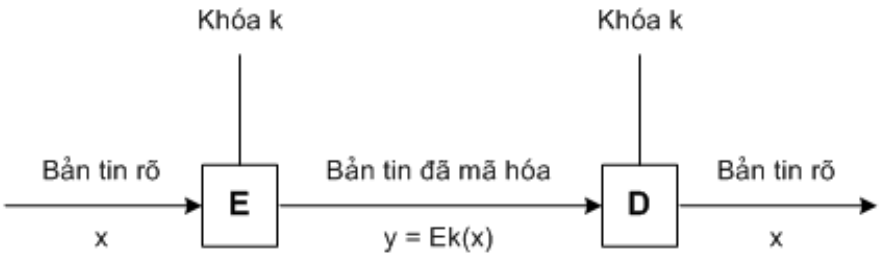


Figure 4.1

Mô hình hệ mật mã tổng quát
Có hai kỹ thuật cơ bản sử dụng trong các hệ mã cổ điển, đó là thay thế và dịch chuyển.

4.1.1 Định nghĩa 1

Một hệ mật mã là một bộ 5 (P, C, K,ϵ, D), thỏa mãn các điều kiện sau đây:

1. P là tập hữu hạn các bản tin rõ
2. C là một tập hữu hạn các bản tin đã mã hóa
3. K là không gian khóa, là tập hữu hạn các khóa

[1]This content is available online at <http://voer.edu.vn/content/m13277/1.1/>.

4. Với mỗi K K, tồn tại một giải thuật mã hóa $e_K \in \epsilon$và một giải thuật giải mã $d_K \in D$. *Trong đó:* $e_K : P \to C$ và $d_K : C \to P$ là các hàm sao cho $d_K(e_K(x)) = x$ với mọi x P.

Một số quy ước: plain text (chữ thường), cipher text (chữ hoa)

Ghi chú:

Z_m được định nghĩa là tập {0, ..., m-1} và hai phép toán + và *. Hai phép toán này tương tự như trên trường số nguyên cộng với phép toán rút gọn kết quả module theo m.

4.1.2 Định nghĩa 2

Cho a, b là các số nguyên, m là số nguyên dương. Ta viết $a \equiv b \pmod{m}$ nếu (b-a) chia hết cho m. Cụm từ $a \equiv b \pmod{m}$ được đọc là "a, b đồng dư module m".

4.2 Một số hệ mật mã đơn giản[2]

4.2.1 Mã dịch chuyển (shift cipher)

Đặt $P=C=K=Z_{26}$. Với $0 \leq K \leq 25$, định nghĩa:

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

và

$d_K(y) = y - K \bmod 26$

(x, y Z_{26}).

Trường hợp đặc biệt K=3 ứng với hệ mật mã Caesar.

Ví dụ:

Plain: meet me after the toga party

Cipher: PHHW PH DIWHU WKH WRJD SDUWB

4.2.2 Mã thay thế (substitution cipher)

Đặt $P=C=Z_{26}$. Với K gồm tất cả các hoán vị có thể của 26 ký hiệu 0, 1, ..., 25. Với mỗi K K định nghĩa:

$e_K(x) = K(x) \bmod 26$

và

$d_K(y) = K^{-1}(y) \bmod 26$

Trong đó K^{-1} là hoán vị ngược của K.

[2]This content is available online at <http://voer.edu.vn/content/m13302/1.1/>.

Ví dụ:

Khóa K:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
X	N	Y	A	H	P	O	G	Z	Q	W	B	T	S	F	L	R	C	V	M	U	E	K	J	D	I

Khóa K^{-1}

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
d	l	r	y	v	o	h	e	z	x	w	p	t	b	g	f	j	q	n	m	u	s	k	a	c	i

Cipher: MGZVYZLGHCMHJMXSSFMNHAHYCDLMHA

Figure 4.2

4.2.3 Hệ mật mã Affine

Đặt P=C= Z_{26} và đặt

K={(a, b) Z_{26}X Z_{26}: gcd(a, 26)=1}.

Với K=(a, b) K, định nghĩa:

$e_K(x) = \text{ax} + b \bmod 26$

và

$d_K(y) = a^{-1}(y - b) \bmod 26$

(x, y Z_{26}).

Trong đó $a^{-1} Z_{26}$, sao cho $\text{aa}^{-1} \equiv a^{-1}a \equiv 1 \,(\text{mod}26)$.

Ví dụ:

K=(7, 3)

Giải thuật Euclid mở rộng:

Tính phần tử nghịch đảo: a^{-1}

1. $n_0 = n$
2. $a_0 = a$
3. $t_0 = 0$
4. t = 1
5. $q = \lfloor \frac{n_0}{a_0} \rfloor$
6. $r = n_0 - q \text{ x } a_0$
7. while r > 0 do
8. $\text{temp} = t_0 - q \text{ x } t$
9. If temp 0 then temp = temp mod n
10. If temp < 0 then temp = n – ((-temp) mod n)
11. $t_0 = t$
12. t = temp
13. $n_0 = a_0$
14. $a_0 = r$
15. $q = \lfloor \frac{n_0}{a_0} \rfloor$
16. $r = n_0 - q \text{ x } a_0$
17. if a_0 1 then

a không có nghịch đảo
else
$a^{-1} = t \bmod n$

4.2.4 Hệ mật mã Vigenere

Đặt m là một số nguyên dương. Định nghĩa P=C=K= $(Z_{26})^m$. Với một khóa K=($k_1, k_2, ..., k_m$), chúng ta định nghĩa:

$$e_K(x_1, x_2, ..., x_m) = (x_1 + k_1, x_2 + k_2, ..., x_m + k_m) \tag{4.1}$$

và

$$d_K(y_1, y_2, ..., y_m) = (y_1 - k_1, y_2 - k_2, ..., y_m - k_m) \tag{4.2}$$

Trong đó các phép +, - được thực hiện trên trường Z_{26}.

4.2.5 Hệ mật mã Hill

Đặt m là một số nguyên dương. Đặt P=C=(Z_{26})$^{\mathrm{m}}$ và đặt
K={m x m là ma trận khả nghịch trên Z_{26}}.
Với K K, định nghĩa:
$e_K(x) = \mathrm{xK} \bmod 26$
và
$d_K(y) = \mathrm{yK}^{-1} \bmod 26$
(x, y Z_{26}).
Trong đó: $\mathrm{KK}^{-1} = \mathrm{I_m}$ với $\mathrm{I_m}$ là ma trận đơn vị.
Ví dụ: Với m=2; K= $\begin{pmatrix}11 & 8\\ 3 & 7\end{pmatrix}$, K^{-1}= $\begin{pmatrix}7 & 18\\ 23 & 11\end{pmatrix}$
có x=(9, 20), xK=(3, 4); có x=(11, 24), xK=(11, 22);

4.2.6 Mã hoán vị (permutation cipher)

Đặt m là một số nguyên dương. Đặt P=C=(Z_{26})$^{\mathrm{m}}$ và đặt K là tập tất cả các hoán vị của tập {1, ..., m}.
Với K K, định nghĩa:
$e_K(x_1, ..., x_m) = \left(x_{K(1)}, ..., x_{K(m)}\right) \bmod 26$
và
$d_K(y_1, ..., y_m) = \left(y_{K^{-1}(1)}, ..., y_{K^{-1}(m)}\right) \bmod 26$
Trong đó K^{-1} là hoán vị ngược của K.

4.2.7 Mã dòng (stream cipher)

Định nghĩa
Một hệ mã dòng là một bộ 7 (P, C, K, L, F,ϵ,D), thỏa mãn các điều kiện sau đây:

1. P là tập hữu hạn các bản tin rõ
2. C là một tập hữu hạn các bản tin đã mã hóa
3. K là không gian khóa, là tập hữu hạn các khóa
4. L là tập các dòng khóa
5. *F =(f 1 , f 2, ….) là bộ sinh. Với i>=1: f i : KxP i-1 -> L*
6. Với mỗi z <L, tồn tại một giải thuật mã hóa e_z $\in\epsilon$và một giải thuật giải mã d_z $\in D$. *Trong đó:* e_z :$P\rightarrow C$ và d_z :$C\rightarrow P$ là các hàm sao cho $d_z(e_z(x)) = x$ với mọi x P.

4.3 Một số phương pháp thám mã[3]

4.3.1 Vấn đề thám mã

4.3.1.1 Khái niệm:

Thám mã là công việc phân tích bản tin mã hóa để nhận được bản tin rõ trong điều kiện không biết trước khóa mã.

Trong thực tế, công việc thám mã gặp nhiều khó khăn hơn khi không biết rõ hệ mật mã nào được sử dụng. Tuy nhiên, để đơn giản hóa, chúng ta giả sử người thám mã đã biết rõ hệ mật mã được sử dụng khi tiến hành phân tích mã (*nguyên lý Kerckhoff*). Mục đích là thiết kế được một hệ mật mã an toàn bảo mật.

Trước hết chúng ta cần phân loại mức độ tấn công vào các hệ mật mã. Mức độ này tùy thuộc vào hiểu biết của người thám mã đối với hệ mật mã được sử dụng. Theo đó, chúng ta có thể chia thành các loại tấn công sau:

- **Tấn công chỉ biết bản mã (ciphertext-only):** người thám mã chỉ có bản tin mã hóa.
- **Tấn công biết bản tin rõ (known plaintext):** người thám mã có bản tin rõ và bản mã.
- **Tấn công chọn bản tin rõ (chosen plaintext):** người thám mã tạm thời có quyền truy xuất tới Bộ mã hóa, do đó anh ta có khả năng chọn bản tin rõ và xây dựng bản mã tương ứng.
- **Tấn công chọn bản mã (chosen ciphertext):** người thám mã tạm thời có quyền truy xuất tới Bộ giải mã, do đó anh ta có khả năng chọn bản mã và xây dựng lại bản tin rõ tương ứng.

Trong mọi trường hợp, mục đích là tìm ra khóa mã được sử dụng. Kiểu tấn công chọn bản mã được thực hiện với hệ mật mã khóa công khai mà chúng ta sẽ xem xét trong chương kế tiếp. Trong phần này chúng ta chỉ thảo luận về kiểu tấn công được xem là "yếu nhất" - Tấn công chỉ biết bản mã.

Nhiều kỹ thuật thám mã sử dụng đặc điểm thống kê của tiếng Anh, trong đó dựa vào tần suất xuất hiện của 26 chữ cái trong văn bản thông thường để tiến hành phân tích mã. Becker và Piper đã chia 26 chữ cái thành năm nhóm và chỉ ra xác suất của mỗi nhóm như sau:

1. E, có xác suất khoảng 0.120
2. T, A, O, I, N, S, H, R, mỗi chữ cái có xác xuất nằm trong khoảng từ 0.06 đến 0.09
3. D, L, mỗi chữ cái có xác xuất xấp xỉ 0.04
4. C, U, M, W, F, G, Y, P, B, mỗi chữ cái có xác xuất nằm trong khoảng từ 0.015 đến 0.023
5. V, K, J, X, Q, Z, mỗi chữ cái có xác xuất nhỏ hơn 0.01

Ngoài ra, tần suất xuất hiện của dãy hai hay ba chữ cái liên tiếp được sắp theo thứ tự giảm dần như sau [11]: TH, HE, IN, ER ... THE, ING, AND, HER...

4.3.1.2 Thám mã tích cực:

Thám mã tích cực là việc thám mã sau đó tìm cách làm sai lạc các dữ liệu truyền, nhận hoặc các dữ liệu lưu trữ phục vụ mục đích của người thám mã.

Thám mã thụ động:

Thám mã thụ động là việc thám mã để có được thông tin về bản tin rõ phục vụ mục đích của người thám mã.

4.3.2 Thám mã Affine

Giả sử Trudy đã lấy được bản mã sau đây:

FMXVEDKAPHFERBNDKRXRSREFMORUDSDKDVSHVUFEDKAPRKDLYEVLRHHRH.

Trudy thống kê tần suất xuất hiện của 26 chữ cái như trong bảng sau:

[3]This content is available online at <http://voer.edu.vn/content/m13303/1.1/>.

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

Figure 4.3

Chỉ có 57 chữ cái trong bản mã nhưng phương pháp này tỏ ra hiệu quả để thám mã Affine. Ta thấy tần suất xuất hiện các chữ cái theo thứ tự là: R(8), D(6), E, H, K(5) và F, S, V(4). Vì vậy dự đoán đầu tiên của ta có thể là: R là mã của e, D là mã của t. Theo đó, $e_K(4) = 17$ và $e_K(19) = 3$. Mà $e_K(x) = ax + b$ với a, b là các biến. Để tìm K=(a, b) ta giải hệ phương trình:

4a+b=17

19a+b=3

Suy ra, a = 6, b=19. Đây không phải là khóa vì gcd(a, 26) = 2 > 1. Ta lại tiếp tục phỏng đoán: R là mã của e, E là mã của t. Ta nhận được a = 13, chưa thỏa mãn. Tiếp tục với H, ta có a=8. Cuối cùng, với K ta tìm được K = (3, 5). Sử dụng khóa mã này ta có được bản tin rõ như sau:

algorithmsrequiregeneraldefinitionsofarithmeticprocesses

4.3.3 Thám mã Vigenere

Để thám mã Vigenere, trước hết cần xác định độ dài từ khóa, ký hiệu là m. Sau đó mới xác định từ khóa. Có hai kỹ thuật để xác định độ dài từ khóa đó là phương pháp Kasiski và phương pháp chỉ số trùng hợp (index of coincidence).

Phương pháp Kasiski được đưa ra bởi Friedrich Kasiski năm 1863. Phương pháp này làm việc như sau:

Tìm trên bản mã các cặp xâu kí tự giống nhau có độ dài ít nhất là 3, ghi lại khoảng cách giữa vị trí chữ cái đầu tiên trong các xâu và xâu đầu tiên. Giả sử nhận được d 1 , d 2 ... Tiếp theo ta phỏng đoán m là số sao cho ước số chung lớn nhất của các d i chia hết cho m.

Ví dụ:

Plaintext: conghoa|danchun|handant|runghoa|sapsuat|hanghoa

Keyword: abcdefg

Ciphertext: CPPJLTG DBPFLZT HBPGESZ RVPJLTG SBRVYFZ HBPJLTG

Vị trí xuất hiện của dãy PJL lần lượt là: *3, 24, 38.* Do vậy, dãy *d1, d2 ... là 21, 35; gcd(d1, d2 ...) = 7*

Phương pháp chỉ số trùng hợp sẽ cho biết các bằng chứng để nhận được giá trị m. Phương pháp này được đưa ra bởi Wolfe Friedman năm 1920 như sau:

Giả sử x=x 1 x 2... x n là xâu có n ký tự. Chỉ số trùng hợp của x, ký hiệu là I c (x), được định nghĩa là xác suất mà hai phần tử ngẫu nhiên của x là giống nhau. Giả sử chúng ta ký hiệu tần suất của A, B, C, ..., Z trong x lần lượt là f 0 , f 1 , ..., f 25 . Chúng ta có thể chọn hai phần tử của x theo $(2^n) = n!/(2!(n-2)!)$ *cách. Với mỗi* $0 \leq i \leq 25$ *, có* $\left(2^{f_i}\right)$ *cách chọn các phần tử là i. Vì vậy, chúng ta có công thức:*

I c (x) = $\frac{\sum_{i=0}^{25} f_i(f_i-1)}{n(n-1)}$

Bây giờ, giả sử x là xâu văn bản tiếng Anh. Ta có $I_c(x)$ $\sum_{i=0}^{25} p_i^2 = 0.065$

Ví dụ:

Cho bản mã trong hệ mật mã Vigenere

CHREEV	OAHMAE	RATBIA	XXWTNX	BEEOPH	BSBQMQ	EQERBW
RVXUOA	XXAOSXX	...				
LXFPSK						
VRVPRT						...**CHR**
ZBWELE						
AMRVLO	...		...	W**CHR**QH	...	
PEEWEV	KAKOE	WADREM	XMTBHHC	HRTKDN	VRZ**CHR**	CLQOHP
WQAIIW	XNRMGW	OIIFKBE				

Figure 4.4

- Theo phương pháp Kasiski, đầu tiên xâu CHR xuất hiện ở 4 vị trí trong bản mã, lần lượt là: 1, 166, 236 và 286. Khoảng cách giữa các xâu là 165, 235 và 285. Ước số chung lớn nhất của các số này là 5. Vậy ta có m =5.
- Theo phương pháp chỉ số trùng hợp, với m=1 thì chỉ số trùng hợp là $I_c(x) = 0.045$; m=2, $I_c(x)$=0.046 và 0.041; m=3, $I_c(x)$=0.043, 0.050, 0.047; m=4, $I_c(x)$=0.042, 0.039, 0.046, 0.040; m=5, $I_c(x)$=0.063, 0.068, 0.069, 0.072; Ta dừng và nhận được m = 5.

Để xác định khóa mã, ta sử dụng phương pháp thống kê sau đây:

Giả sử x=x 1 x 2... x n và y=y 1 y 2... y n' là hai xâu có n và n' ký tự . Chỉ số trùng hợp tương quan của x và y, ký hiệu là MI c (x,y), được định nghĩa là xác suất mà một phần tử ngẫu nhiên của x bằng một phần tử ngẫu nhiên của y. Nếu chúng ta ký hiệu tần suất của A, B, C, ..., Z trong x và y lần lượt là f 0 , f 1 , ..., f 25 . và f' 0 , f' 1 , ..., f' 25 . Thì:

MI c (x,y) = $\frac{\sum_{i=0}^{25} f_i f'_i}{nn'}$

Bây giờ, giả sử x,y là xâu văn bản tiếng Anh. Ta có $MI_c(x_i,y_j)$ 0.065

Ví dụ:

Giả sử m=5 như ta đã thực hiện ở trên. Theo phương pháp thống kê [11] ta tìm được khóa mã là: JANET. Vậy bản tin rõ sẽ là: *the almond tree was in ...*

4.4 Lý thuyết Shannon về mật mã[4]

Năm 1949, Claude Shannon đưa ra lý thuyết về các hệ thống mật. Lý thuyết này có ảnh hưởng lớn tới việc nghiên cứu mật mã. Theo Shannon, có hai cách tiếp cận về tính bảo mật của một hệ mật mã là: bảo mật tính toán được (computational security) và bảo mật không điều kiện (unconditional security).

Các hệ mật mã được xếp vào nhóm "Bảo mật tính toán được" nếu tồn tại một giải thuật phá vỡ hệ mật mã với không quá N phép tính toán (N là số đủ lớn). Tuy vậy, trên thực tế chưa có hệ mật mã nào, về mặt thực hành, được chứng minh là thỏa tiêu chí này. Vì vậy, khi đưa một hệ mật mã vào nhóm này, ta cần chỉ ra sự tồn tại một giải thuật tốt nhất để phá vỡ hệ mã nhưng với khối lượng tính toán cực lớn vượt quá khả năng tính toán của máy tính hoặc thời gian tính toán cho phép. Về lý thuyết, người ta thường chỉ đưa ra một bằng chứng về độ bảo mật bằng cách chuyển tương đương về một bài toán khó trong toán học.

Một hệ mật mã được xếp vào nhóm "Bảo mật không điều kiện" nếu với mọi nguồn lực tính toán cần thiết chúng ta cũng không thể phá vỡ hệ mã.

[4]This content is available online at <http://voer.edu.vn/content/m13304/1.1/>.

Chương 5

Giới thiệu lý thuyết Số-Mã[1]

5.1 Hệ mật mã khóa công khai

Hệ mật mã khoá công khai (public-key cryptography) là một bước tiến lớn nhất và là cuộc cách mạng thực sự trong lĩnh vực mật mã. Sau hàng thiên niên kỷ làm việc với các giải thuật mã hoá và giải mã thủ công, sự ra đời của các thiết bị mã hoá và giải mã tự động (rotor) đánh dấu một bước ngoặt lớn. Tuy nhiên, các giải thuật mã hoá mặc dù rất phức tạp chẳng hạn như DES của IBM, cũng vẫn dựa trên hai kỹ thuật cơ bản là dịch chuyển và thay thế.

Hệ mã khoá công khai triệt để thay đổi những gì đã tồn tại trước đây. Ví dụ như các giải thuật mã công khai dựa trên các hàm toán học chứ không dựa trên kỹ thuật thay thế và dịch chuyển. Quan trọng hơn nữa, hệ mã công khai là hệ mã không đối xứng, nghĩa là sử dụng hai khoá liên đới cho việc mã hoá và giải mã thay vì một khoá duy nhất như trong các hệ mã cổ điển (hay còn gọi là hệ mã đối xứng). Việc này đáp ứng được các yêu cầu trong các ứng dụng về bảo mật, phân phối khoá, và xác thực điện tử.

Trước khi đi vào chi tiết, để tránh các quan điểm sai lạc, chúng ta cùng xem xét một số khái niệm về hệ mã khoá công khai [2]:

- Quan điểm cho rằng hệ mật mã khoá công khai bảo mật hơn hệ mã cổ điển là sai lầm vì mức độ bảo mật của bất kỳ giải thuật mã hoá nào cũng đều phụ thuộc vào độ dài khoá và khối lượng công việc phải đầu tư để phá vỡ mã.
- Quan điểm cho rằng hệ mật mã khoá công khai là một kỹ thuật tổng quát và đã làm cho hệ mã cổ điển trở nên lỗi thời là sai vì khối lượng công việc yêu cầu phải tính toán trước của các mô hình mã công khai tỏ ra là một hạn chế, do đó không có khả năng nào chỉ ra rằng hệ mã cổ điển sẽ bị loại bỏ. Theo các nhà phát minh ra hệ mã công khai thì việc giới hạn áp dụng hệ mã công khai cho công tác quản lý khoá mã và các ứng dụng sử dụng chữ ký điện tử là thích hợp nhất.
- Có quan điểm cho rằng việc phân phối khoá mã là công việc nặng nề đối với hệ mã cổ điển (sử dụng các trung tâm phân phối) trong khi lại là đơn giản với hệ mật mã khoá công khai. Trên thực tế thì không phải như vậy. Một số giao thức phân phối khoá sử dụng hệ mật mã khoá công khai vẫn cần có trung tâm phân phối và thủ tục không hề đơn giản chút nào.

Bài này cung cấp các kiến thức tổng quan về hệ mật mã khoá công khai. Phần lớn lý thuyết của hệ mật mã khoá công khai dựa trên lý thuyết số nên chúng ta sẽ cùng nhắc lại một số vấn đề có liên quan đến lý thuyết số. Tiếp theo, ta sẽ tập trung nghiên cứu một số hệ mật mã khoá công khai điển hình là hệ mật mã El Gamal và hệ mật mã RSA.

[1]This content is available online at <http://voer.edu.vn/content/m13305/1.1/>.

5.2 Một số khái niệm trong lý thuyết số

5.2.1 Phép chia (divisors)

b<>0, chúng ta nói a chia hết cho b nếu a=mb với m bất kỳ. a,b,m là các số nguyên. Nếu a chia hết cho b, chúng ta ký hiệu b|a. Nếu b|a ta nói b là ước số của a.

Chúng ta xem xét một số tính chất sau đây:

- Nếu a|1 thì a= ± 1
- nếu a|b và b|a thì a= $\pm b$
- b<>0=> b|0
- b|g và b|h thì b|(mg+nh) với m, n là các số nguyên tùy ý.

Số nguyên tố (prime)

Một số nguyên p>1 là số nguyên tố khi các ước số của nó là ± 1 và $\pm p$.

Tính chất quan trọng:

Bất kỳ số nguyên a >1 có thể phân tích thành dạng duy nhất tích luỹ thừa của các số nguyên tố.

a= $p_1^{\alpha_1} p_2^{\alpha_2} \dots p_t^{\alpha_t}$

Trong đó, p_i là các số nguyên tố và α_i>0.

Ví dụ: 91=7X13;11011=7X11^2X13; 3600=$2^4 3^2 5^2$

Ước số chung lớn nhất (greatest common divisor):

gcd(a, b)=max[k, sao cho k|a và k|a]

(xem thêm [12])

Giải thuật Euclid:

gcd(a, b)=gcd(b, a mod b)

Số nguyên tố cùng nhau (relatively prime)

a, b gọi là nguyên tố cùng nhau nếu gcd(a, b)=1

5.2.2 Phép toán Modulo

a=qn+r (0<=r<n, q=[n|a])

a, b là đồng dư modulo n nếu (a mod n)=(b mod n)

(xem thêm [12])

Định lý Fermat và định lý Euler

Định lý Fermat:

Nếu p là số nguyên tố và a là số nguyên dương không chia hết cho p, thì:

a^{p-1} 1 mod p

Định lý Euler:

Với mọi a và n nguyên tố cùng nhau, ta có:

$a^{\phi(n)}$ 1 mod n

Trong đó, trường hợp đặc biệt nếu n là số nguyên tố: $\phi(n) = n - 1$

Bài toán logarit rời rạc

Bài toán logarit rời rạc trong Z_n:

Cho I = (n, α, β), trong đó n là số nguyên tố, $\alpha \in Z_n$ là phần tử nguyên thủy và $\beta \in Z_n$. Tìm một số nguyên a, $0 \le a \le n-2$, sao cho:

$$\alpha^a \equiv \beta \,(\mathrm{mod} n) \tag{5.1}$$

Chúng ta có: $a = \log_\alpha \beta$

5.3 Hệ mật mã và sơ đồ chữ ký El Gamal

Hệ mật mã El Gamal

Chọn n là số nguyên tố sao cho bài toán logarit rời rạc trong Z_n không có lời giải và chọn một phần tử nguyên thủy $\alpha \in Z_n$. Đặt $P = Z_n$, $C = Z_n$ x Z_n và định nghĩa:

$K = \{ (n, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{n} \}$

Công khai giá trị (n, α, β), giữ bí mật a.

Với K = (n, α, a, β) và một giá trị ngẫu nhiên (bí mật) $k \in Z_{n-1}$, định nghĩa:

$$e_K(x, k) = (y_1, y_2) \tag{5.2}$$

Trong đó: $y_1 = \alpha^k \bmod n, y_2 = \mathrm{x}\beta^k \bmod n$

Với $y_1, y_2 \in Z_n$, định nghĩa:

$$d_K(y_1, y_2) = y_2 (y_1^a)^{-1} \bmod n \tag{5.3}$$

5.3.1 Định nghĩa sơ đồ chữ ký điện tử

Như chúng ta thấy, trong mô hình truyền thống [1], xác thực mã cung cấp một phương tiện để phát hiện ra các đoạn tin lừa dối đưa vào bởi tin tặc trung gian. Tuy nhiên, xác thực mã không đủ cung cấp bằng chứng cho đối tượng thứ 3 (trọng tài) đưa ra quyết định về việc A hay B đã gửi một bản tin nào đó khi có xảy ra tranh chấp. Vì vậy, xác thực mã được sử dụng rất hạn chế trong lĩnh vực thương mại điện tử [1] khi khách hàng cần được bảo đảm rằng các nhà buôn không thể làm giả các đơn hàng và các nhà buôn cần sự ràng buộc về mặt trách nhiệm khi khách hàng thực hiện các đơn hàng. Trong các trường hợp như vậy, một chữ ký điện tử là cần thiết.

Một sơ đồ chữ ký điện tử bao gồm một giải thuật chữ ký và một giải thuật kiểm chứng. Một chữ ký của một tài liệu là một giá trị phụ thuộc vào nội dung của tài liệu và bí mật lưu giữ bởi người ký (nghĩa là một khoá bí mật) kết hợp tài liệu với một thực thể (nghĩa là một khoá công khai để kiểm chứng). Giải thuật kiểm chứng thường sử dụng tài liệu và khoá kiểm chứng như đầu vào, tuy nhiên trong các trường hợp ngoại lệ, tài liệu hoặc các phần của tài liệu có thể được khôi phục từ chữ ký. Các tính chất của chữ ký điện tử được trình bày trong [12].

Chúng ta thấy rõ ràng mô hình mật mã khoá công khai là tự nhiên với sơ đồ chữ ký điện tử [11]. Định nghĩa hình thức sơ đồ chữ ký điện tử được giới thiệu trong [11] như sau:

Định nghĩa : Một sơ đồ chữ ký là một bộ 5 (*P, A, K, S, V*), thỏa mãn các điều kiện sau đây:

1. *P* là tập hữu hạn các bản tin
2. *A* là một tập hữu hạn các chữ ký
3. *K* là không gian khóa, là tập hữu hạn các khóa
4. Với mỗi K *K*, tồn tại một giải thuật ký $\mathrm{sig}_K \in S$và một giải thuật kiểm chứng liên đới $\mathrm{ver}_K \in V$. *Mỗi* $\mathbf{sig}_K$:*P*→*A* và ver_K :*P*x*A*→*{true, false}* là các hàm sao cho công thức sau thỏa mãn với mọi mọi x *P* vàvới mọiy *A:*

true nếu y=$\mathrm{sig_K}$(x)false nếu y $\mathrm{sig_K}$(x) $\mathrm{ver}_K(x, y) =$

5.3.2 Sơ đồ chữ ký El Gamal

Chọn n là số nguyên tố sao cho bài toán logarit rời rạc trong Z_n không có lời giải và chọn một phần tử nguyên thủy $\alpha \in Z_n$. Đặt $P = Z_n$, $A = Z_n$ x Z_{n-1} và định nghĩa:

$K = \{ (n, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{n} \}$

Công khai giá trị (n, α, β), giữ bí mật a.

Với K = (n, α, a, β) và một giá trị ngẫu nhiên (bí mật) $k \in Z_{n-1}$, định nghĩa:

$$\mathrm{sig}_K(x, k) = (\gamma, \delta) \tag{5.4}$$

Trong đó: $\gamma = \alpha^k \mathrm{mod} n, \delta = (x - \mathrm{a}\gamma)\, k^{-1} \mathrm{mod}\,(n-1)$

Với $x, \gamma \in Z_n$ và $\delta \in Z_{n-1}$, định nghĩa:

$$\mathrm{ver}_K(x, \gamma, \delta) = \mathrm{true} \Leftrightarrow \beta^\gamma \gamma^\delta \equiv \alpha^x (\mathrm{mod} n) \tag{5.5}$$

Chương 6

Hệ mật mã và sơ đồ chữ ký RSA[1]

6.1 Bài toán phân tích thừa số

Bài toán phân tích n thành các thừa số:
Input: n và β
1. a=2
2. for j=2 to β do
a=a^j mod n
3. d=gcd(a-1, n)
4. if 1<d<n then
d is a factor of n (success)
else
no factor of n is found (failure)

6.2 Hàm một chiều và ứng dụng

Định nghĩa : Một hàm băm h là hàm một chiều nếu với giá trị băm z, chúng ta không có khả năng tìm ra thông điệp x sao cho h(x) =z.

Định nghĩa : hé lộ ứng dụng của hàm băm một chiều trong việc xây dựng chữ ký điện tử sử dụng các hệ mật mã khóa công khai vốn được xem là có tốc độ xử lý chậm. Phần sau chúng ta sẽ xem xét ứng dụng của hàm băm một chiều trong lĩnh vực này.

6.3 Hệ mật mã và sơ đồ chữ ký RSA

6.3.1 Hệ mật mã RSA

Giải thuật RSA là một giải thuật mật mã công khai được phát triển bởi Ron Rivest, Adi Shamir và Leonard Adleman có thể được sử dụng trong công tác mã hoá và công nghệ chữ ký điện tử. Trước hết chúng ta sẽ xem xét cách làm việc của giải thuật này khi áp dụng trong mã hoá. Giả sử rằng Bob muốn nhận được các bản tin được mã hoá như trong hình sau :

[1]This content is available online at <http://voer.edu.vn/content/m13307/1.1/>.

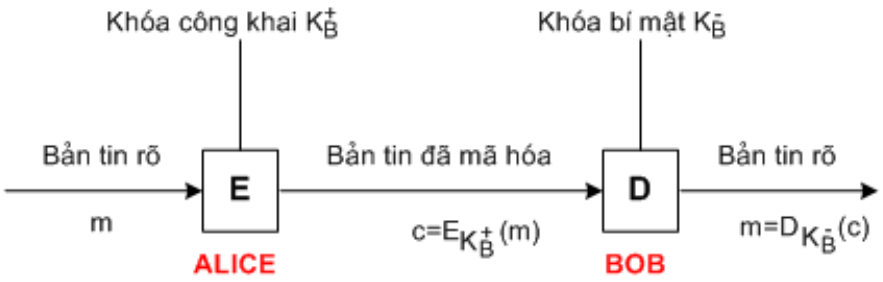


Figure 6.1: Mô hình hệ mật mã khóa công khai

Có hai thành phần liên đới trong RSA:

- Việc lựa chọn khoá công khai và khoá bí mật
- Giải thuật mã hoá và giải mã

Để lựa chọn các khoá công khai và khoá bí mật, Bob phải thực hiện các bước sau đây:

1. Chọn hai số nguyên tố đủ lớn, p và q. Chọn p và q thế nào là đủ lớn? Nếu p, q càng lớn thì càng khó phá vỡ RSA, tuy nhiên thời gian để mã hoá và giải mã sẽ tăng lên đáng kể. Phòng nghiên cứu RSA đề xuất rằng tích của p và q nên là 1024 bit, ngoài ra, tích có thể là 768bit đối với các thông tin ít có giá trị hơn.
2. Tính toán n=pq và z=(p-1)(q-1)
3. Chọn một số, e, nhỏ hơn n sao cho gcd(e, z)=1. e sẽ được sử dụng trong mã hoá.
4. Tìm một số d sao cho ed-1 chia hết cho z. d sẽ được sử dụng để giải mã.
5. Bob công khai khoá K^+_B, khoá này là cặp số (n,e) và giữ bí mật khoá K^-_B, khoá này là cặp số (n,d)

Việc mã hoá thực hiện bởi Alice và giải mã bởi Bob được hoàn thiện như sau:

Giả sử Alice muốn gửi cho Bob một mẫu bit, hoặc một số m sao cho m<n. Để mã hoá, Alice thực hiện việc luỹ thừa, m^e, sau đó tính toán số dư khi đem chia modulo m^e cho n. Vì vậy, giá trị được mã hoá (c), của bản tin m là:

$c=m^e \bmod n$

Để giải mã đoạn tin mã hoá nhận được (c), Bob tính toán:

$m=c^d \bmod n$

Việc này đòi hỏi việc sử dụng khoá bí mật (n, d)

Ví dụ:

p=5, q=7 => n=pq=35, z=24.

Bob chọn e=5 vì gcd(5, 24)=1; cuối cùng Bob chọn d=29 vì ed-1 =29.5-1 chia hết cho 24.

Chọn m = 0, 1, 2 ... sinh viên thực hiện quá trình mã hóa và giải mã.

Tại sao giải thuật RSA khả thi?

Để hiểu công việc của giải thuật RSA chúng ta cần thực hiện phép chia modulo n, kết quả của mỗi phép toán được thay thế bởi số dư khi chia cho n. Chúng ta lấy n=pq, trong đó p và q là các số nguyên tố lớn được sử dụng trong giải thuật RSA [2].

Nhắc lại rằng trong mã hoá RSA, một đoạn thông tin (được biểu diễn bởi một số nguyên), m, được luỹ thừa e sử dụng phép toán modulo-n để mã hoá. Giải mã được thực hiện bởi tăng thêm giá trị này lên luỹ thừa d, một lần nữa sử dụng phép toán modulo-n. Kết quả của một bước mã hoá theo sau bởi một bước giải mã là: $(m^e)^d$. Trong đó:

$(m^e)^d \bmod n = m^{ed} \bmod n$

Theo giả thiết z=(p-1)(q-1), z|(ed-1), nghĩa là ed = tz+1 (*với t1*). Do đó:

$(m^e)^d\ m^{t(p-1)(q-1)+1} \bmod n$

$(m^{t(p-1)(q-1)})m \bmod n$

$(m^{(p-1)(q-1)})^t m \bmod n$

$(m^{\phi(n)})^t m \bmod n$

$(1)^t m \bmod n$ {*Theo định lý Euler*}

$m \bmod n$

m.

Bảo mật trong giải thuật RSA

Bảo mật của RSA dựa trên việc chưa tồn tại các giải thuật đủ nhanh để khai triển luỹ thừa một số, trong trường hợp này là khai triển số công khai n thành các số nguyên tố p và q. Nếu ta biết p và q sau đó cộng với việc hiểu biết giá trị e thì chúng ta có thể dễ dàng tính ra d. Do đó bảo mật của RSA tương đương với việc tìm ra giải thuật đủ nhanh để triển khai luỹ thừa một số [2].

6.3.2 Sơ đồ chữ ký RSA

Tương tự như mô hình RSA chúng ta đã xem xét trong phần trên, tuy nhiên Bob sẽ ký vào tài liệu sử dụng khoá bí mật và người nhận sẽ sử dụng khoá công khai để kiểm chứng chữ ký. Nếu tài liệu quá dài chúng ta có thể áp dụng phương pháp chia nhỏ tài liệu, sau đó ký từng đoạn

Chương 7

Phân phối khóa và thỏa thuận khóa[1]

7.1 Quản trị khóa trong các mạng truyền tin

Trong các bài trước, chúng ta đã làm quen với các kỹ thuật mật mã và các bài toán quan trọng khác liên quan đến việc truyền tin bảo mật trên các mạng truyền tin công cộng nói chung. Ta cũng đã thấy rằng các hệ mật mã khoá công khai có nhiều ưu việt hơn các hệ mật mã khoá đối xứng trong việc làm nền tảng cho các giải pháp an toàn thông tin, và đặc biệt nếu đối với các hệ mật mã khoá đối xứng việc triển khai đòi hỏi những kênh bí mật để chuyển khoá hoặc trao đổi khoá giữa các đối tác, thì về nguyên tắc, đối với các hệ mật mã khoá công khai, không cần có những kênh bí mật như vậy, vì các khoá công khai có thể được truyền hoặc trao đổi cho nhau một cách công khai qua các kênh truyền tin công cộng. Tuy nhiên, trên thực tế, để bảo đảm cho các hoạt động thông tin được thật sự an toàn, không phải bất cứ thông tin nào về các khoá công khai của một hệ mật mã, của một thuật toán kiểm chứngchữ ký, của một giao thức xác thực thông báo hay xác thực danh tính... cũng phát công khai một cách tràn lan trên mạng công cộng, mà dẫu là công khai nhưng người ta cũng mong muốn là những ai cần biết thì mới nên biết mà thôi. Do đó, dẫu là dùng các hệ có khoá công khai, người ta cũng muốn có những giao thức thực hiện việc trao đổi khoá giữa những đối tác thực sự có nhu cầu giao lưu thông tin với nhau, kể cả trao đổi khoá công khai. Việc trao đổi khoá giữa các chủ thể trong một cộng đồng nào đó có thể được thiết lập một cách tự do giữa bất cứ hai người nào khi có nhu cầu trao đổi thông tin, hoặc có thể được thiết lập một cách tương đối lâu dài trong một thời hạn nào đó trong cả cộng đồng với sự điều phối của một cơ quan được uỷ thác (mà ta ký hiệu là TA-trusted authority). Việc trao đổi khoá trong trường hợp thứ nhất ta gọi đơn giản là thoả thuận khoá, còn trong trường hợp thứ hai ta gọi là phân phối khoá, TA là nơi thực hiện việc phân phối, cũng tức là nơi quản trị khoá. Việc thoả thuận khoá nói chung không cần có sự tham gia của một TA nào và chỉ có thể xẩy ra khi các hệ bảo mật mà ta sử dụng là hệ có khoá công khai, còn việc phân phối khoá thì có thể xẩy ra đối với các trường hợp sử dụng các hệ khoá đối xứng cũng như các hệ có khoá công khai. Việc phân phối khoá với vai trò quản trị khoá của một TA là một việc bình thường, đã tồn tại từ rất lâu trước khi có các hệ mật mã khoá công khai. Trong phần này, chúng ta sẽ chỉ tập trung giới thiệu sơ đồ trao đổi khóa và giao thức trao đổi khóa Diffie-Hellman (xem [13]).

7.2 Sơ đồ trao đổi khóa Diffie-Hellman

7.2.1 Hệ phân phối khoá Diffie-Hellman

Hệ phân phối khoá Diffie-Hellman không đòi hỏi TA phải biết và chuyển bất kỳ thông tin bí mật nào về khoá của các người tham gia trong mạng để họ thiết lập được khoá chung bí mật cho việc truyền tin với nhau.

[1]This content is available online at <http://voer.edu.vn/content/m13308/1.1/>.

Trong một hệ phân phối khoá Diffie-Hellman, TA chỉ việc chọn một số nguyên tố lớn p và một phần tử nguyên thuỷ α theo mod p, sao cho bài toán tính $\log_\alpha$ trong p $Z_p{}^*$ là rất khó. Các số p và α được công bố công khai cho mọi người tham gia trong mạng. Ngoài ra, TA có một sơ đồ chữ ký với thuật toán ký (bí mật) sig_{TA} và thuật toán kiểm chứng (công khai) ver_{TA}.

Một thành viên bất kỳ A với danh tính ID(A) tuỳ ý chọn một số $a_A\,(0 \le a_A \le p-2)$ và tính $b_A = \alpha^{a_A} \text{mod} p$. A giữ bí mật a_A và đăng ký các thông tin (ID(A), b_A) với TA. TA cấp cho A chứng chỉ:

C(A) = (ID(A), b_A, sig_{TA}(ID(A), b_A)).

Các chứng chỉ của các thành viên trong mạng có thể được lưu giữ trong một cơ sở dữ liệu công khai hoặc uỷ thác cho TA lưu giữ và cung cấp công khai cho các thành viên mỗi khi cần đến.

Khi hai thành viên A và B trong mạng cần có một khoá bí mật chung để truyền tin bảo mật cho nhau thì A dùng thông tin công khai b_B có trong C(B) kết hợp với số bí mật của mình là a_A để tạo nên khoá:

$$K_{A,B} = b_B^{a_A} \text{mod} p = \alpha^{a_B a_A} \text{mod} p \tag{7.1}$$

Khoá chung đó B cũng tạo ra được từ các thông tin công khai b_Acủa A và số bí mật của mình:

$$K_{A,B} = b_A^{a_B} \text{mod} p = \alpha^{a_A a_B} \text{mod} p \tag{7.2}$$

Để bảo đảm được các thông tin về và b_B và b_A là chính xác, A và B có thể dùng thuật toán ver_{TA} để kiểm chứng chữ ký xác thực của TA trong các chứng chỉ C(B) và C(A) tương ứng.

Độ an toàn của hệ phân phối khoá Diffie-Hellman được bảo đảm bởi yếu tố sau đây: Biết b_A và b_B để tính $\text{K}_{\text{A,B}}$ chính là bài toán Diffie-Hellman tương đương: biết $\alpha^a \text{mod} p$ và $\alpha^b \text{mod} p$, tính $\alpha^{\text{ab}} \text{mod} p$. Đây là một bài toán khó tương đương bài toán tính lôgarit rời rạc hay bài toán phá mật mã ElGamal.

7.2.2 Giao thức trao đổi khoá Diffie-Hellman

Hệ phân phối khoá Diffie-Hellman nói trong mục trước có thể dễ dàng biến đổi thành một giao thức trao đổi (hay thoả thuận) khoá trực tiếp giữa những người sử dụng mà không cần có sự can thiệp của một TA làm nhiệm vụ điều hành hoặc phân phối khoá. Một nhóm bất kỳ người sử dụng có thể thoả thuận cùng dùng chung một số nguyên tố lớn p và một phần tử nguyên thuỷ α theo mod p, hai người bất kỳ trong nhóm A và B mỗi khi muốn truyền tin bảo mật cho nhau có thể cùng thực hiện giao thức sau đây để trao đổi khoá:

1. A chọn ngẫu nhiên số a_A (0 a_A p -2), giữ bí mật a_A, tính $b_A = \alpha^{a_A} \text{mod} p$
và gửi b_A cho B.
2. Tương tự, B chọn ngẫu nhiên số a**B** (0 a_B p -2), giữ bí mật a_B, tính $b_B = \alpha^{a_B} \text{mod} p$ và gửi b_B cho A.
3. A và B cùng tính được khoá chung:

$$K_{A,B} = b_B^{a_A} \text{mod} p = b_A^{a_B} \text{mod} p\,(= \alpha^{a_A a_B} \text{mod} p) \tag{7.3}$$

Giao thức trao đổi khoá Diffie-Hellman có các tính chất sau:

1. *Giao thức là an toàn đối với việc tấn công thụ động*, nghĩa là một người thứ ba, dù biết b_A và b_B sẽ khó mà biết được $\text{K}_{\text{A,B}}$.

Ta biết rằng bài toán "biết b_A và b_B tìm $\text{K}_{\text{A,B}}$" chính là bài toán Diffie-Hellman và trong mục 7.2.1 ta có nói rằng bài toán đó tương đương với bài toán phá mật mã El Gamal. Bây giờ ta chứng minh điều này. Phép mật mã El Gamal với khoá $\text{K} = (p, \alpha, a, \beta)$, trong đó $\beta = \alpha^a \text{mod} p$, cho ta từ một bản rõ x và một số ngẫu nhiên $k \in Z_{p-1}1$ lập được mật mã:

$$e_K\,(x, k) = (y_1, y_2) \tag{7.4}$$

Trong đó: $y_1 = \alpha^k \text{mod} p, y_2 = \text{x}\beta^k \text{mod} p$

Và phép giải mã được cho bởi:

$$d_K\,(y_1, y_2) = y_1\,(y_2^a)^{-1} \,\text{mod} p \tag{7.5}$$

Giả sử ta có thuật toán A giải bài toán Diffie-Hellman. Ta sẽ dùng A để phá mã El Gamal như sau: Cho mật mã (y_1, y_2). Trước hết, dùng A cho $y_1 = \alpha^k \text{mod} p$ và $\beta = \alpha^a \text{mod} p$, ta được:

A(y_1, β) = $\alpha^{ka} = \beta^k \mathrm{mod} p$
và sau đó ta thu được bản rõ x từ β^k và y_2 như sau:

$$x = y_2 \left(\beta^k\right)^{-1} \mathrm{mod} p \tag{7.6}$$

Ngược lại, giả sử có thuật toán B phá mã El Gamal, tức là:
B $(p, \alpha, a, \beta, y_1, y_2) = x = y_2 \left(y_1^a\right)^{-1} \mathrm{mod} p$
Áp dụng B cho $\beta = b_A, y_1 = b_B, y_2 = 1$, ta được
B $(p, \alpha, b_A, b_B, 1)^{-1} = \left(1.\left(b_B^{a_A}\right)^{-1}\right)^{-1} = \alpha^{a_A a_B} \mathrm{mod} p$ tức là giải được bài toán Diffie-Hellman.

2. *Giao thức là không an toàn đối với việc tấn công chủ động bằng cách đánh tráo giữa đường*, nghĩa là một người thứ ba C có thể đánh tráo các thông tin trao đổi giữa A và B, chẳng hạn, C thay α^{a_A} mà A định gửi cho B bởi $\alpha^{a'_A}$,và thay α^{a_B} mà B định gửi cho A bởi $\alpha^{a'_B}$, như vậy, sau khi thực hiện giao thức trao đổi khoá, A đã lập một khoá chung $\alpha^{a_A a'_B}$ với C mà vẫn tưởng là với B, đồng thời B đã lập một khoá chung $\alpha^{a'_A a_B}$với C mà vẫn tưởng là với A; C có thể giải mã mọi thông báo mà A tưởng nhầm là mình gửi đến B, cũng như mọi thông báo mà B tưởng nhầm là mình gửi đến A!

Một cách khắc phục kiểu tấn công chủ động nói trên là làm sao để A và B có thể kiểm chứng để xác thực tính đúng đắn của các khoá công khai b_A và b_B. Đưa vào giao thức trao đổi khoá Diffie-Hellman thêm vai trò điều phối của một TA để được một hệ phân phối khoá Diffie-Hellman như ở mục 7.2.1 là một cách khắc phục như vậy. Trong hệ phân phối khoá Diffie-Hellman, sự can thiệp của TA là rất yếu, thực ra TA chỉ làm mỗi một việc là cấp chứng chỉ xác nhực khoá công khai cho từng người dùng chứ không đòi hỏi biết thêm bất cứ một bí mật nào của người dùng. Tuy nhiên, nếu chưa thoả mãn với vai trò hạn chế đó của TA, thì có thể cho TA một vai trò xác nhận yếu hơn, không liên quan gì đến khoá, chẳng hạn như xác nhận thuật toán kiểm thử chữ ký của người dùng, còn bản thân các thông tin về khoá (cả bí mật và công khai) thì do những người dùng trao đổi trực tiếp với nhau. Với cách khắc phục có vai trò rất hạn chế đó của TA, ta có được giao thức ở phần sau.

7.2.3 Giao thức trao đổi khoá D-H có chứng chỉ xác thực

Mỗi người dùng A có một danh tính ID(A) và một sơ đồ chữ ký với thuật toán ký $\mathrm{sig_A}$ và thuật toán kiểm chứng $\mathrm{ver_A}$. TA cũng có một vai trò xác thực, nhưng không phải xác thực bất kỳ thông tin nào liên quan đến việc tạo khoá mật mã của người dùng (dù là khoá bí mật hay là khoá công khai), mà chỉ là xác thực một thông tin ít quan hệ khác như thuật toán kiểm chứng chữ ký của người dùng. Còn bản thân các thông tin liên quan đến việc tạo khoá mật mã thì các người dùng sẽ trao đổi trực tiếp với nhau. TA cũng có một sơ đồ chữ ký của mình, gồm một thuật toán ký $\mathrm{sig_{TA}}$ và một thuật toán kiểm chứng (công khai) $\mathrm{ver_{TA}}$. Chứng chỉ mà TA cấp cho mỗi người dùng A sẽ là:
C(A) = (ID(A), $\mathrm{ver_A}$, $\mathrm{sig_{TA}}$(ID(A), $\mathrm{ver_A}$)).
Rõ ràng trong chứng chỉ đó TA không xác thực bất kỳ điều gì liên quan đến việc tạo khoá của A cả. Việc trao đổi khoá giữa hai người dùng A và B được thực hiện theo giao thức sau đây:
1. A chọn ngẫu nhiên số $\mathrm{a_A}$ (0 $\mathrm{a_A}$ p -2), tính $b_A = \alpha^{a_A} \mathrm{mod} p$
và gửi $\mathrm{b_A}$ cho B.
2. B chọn ngẫu nhiên số a**B** (0 $\mathrm{a_B}$ p -2), giữ bí mật $\mathrm{a_B}$, tính $b_B = \alpha^{a_B} \mathrm{mod} p$, tính tiếp

$$K = b_B^{a_B} \mathrm{mod} p \tag{7.7}$$

$y_B = \mathrm{sig}_B\left(b_B, b_A\right)$ và gửi (C(B), $\mathrm{b_B}$, $\mathrm{y_B}$) cho A.
3. A tính: $K = b_B^{a_B} \mathrm{mod} p$ dùng $\mathrm{ver_B}$ để kiểm chứng $\mathrm{y_B}$, dùng $\mathrm{ver_{TA}}$ để kiểm chứng C(B), sau đó tính $y_A = \mathrm{sig}_A\left(b_A, b_B\right)$, và gửi (C(A), $\mathrm{y_A}$) cho B.
4. B dùng $\mathrm{ver_A}$ để kiểm chứng $\mathrm{y_A}$,và dùng $\mathrm{ver_{TA}}$ để kiểm chứng C(A).

Nếu tất cả các bước đó được thực hiện và các phép kiểm chứng đều cho kết quả đúng đắn, thì giao thức kết thúc, cả A và B đều có được khoá chung K. Do việc dùng các thuật toán kiểm chứng nên A biết chắc giá trị $\mathrm{b_B}$ là của B và B biết chắc giá trị $\mathrm{b_A}$ là của A, loại trừ khả năng một người C nào khác đánh tráo các giá trị đó giữa đường.

Chương 8

Bảo mật dịch vụ thương mại điện tử[1]

8.1 Đặt vấn đề

8.1.1 An toàn thông tin trong hệ thống thương mại điện tử

Mối đe dọa và hậu quả tiềm ẩn đối với thông tin trong hệ thống mạng phục vụ hoạt động TMĐT là rất lớn, và được đánh giá trên nhiều khía cạnh khác, như: kiến trúc hệ thống công nghệ thông tin, từ chính sách bảo mật thông tin, các công cụ quản lý và kiểm tra, quy trình phản ứng, v.v.

Nguy cơ rủi ro tiềm ẩn trong kiến trúc hệ thống công nghệ thông tin, đó là: tổ chức hệ thống kỹ thuật không có cấu trúc bảo vệ an toàn thông tin; tổ chức và khai thác CSDL, quản lý, lưu trữ và sắp đặt thông tin đã phân loại; cơ cấu tiếp cận từ xa; sử dụng phần mềm ứng dụng; chương trình kiểm tra, kiểm soát người sử dụng, phát hiện và xử lý sự cố, v.v;

Nguy cơ mất an toàn thông tin tiềm ẩn trong chính sách bảo-mật/an-toàn thông tin, đó là: sự chấp hành các chuẩn an toàn, tức là sự xác định rõ ràng cái được phép và không được phép trong khi vận hành hệ thống thông tin; thiết lập trách nhiệm bảo vệ thông tin không rõ ràng; việc chấp hành sử dụng các chuẩn bảo mật thông tin được phân cấp, chuẩn an toàn mạng, truy cập từ bên ngoài, chuẩn an toàn bức tường lửa; chính sách an toàn Internet,v.v;

Thông tin trong hệ thống TMĐT cũng sẽ dễ bị tổn thất nếu công cụ quản lý và kiểm tra của các tổ chức quản lý điều khiển hệ thống không được thiết lập như: các quy định mang tính hành chính như duy trì kiểm tra tiêu chuẩn bảo mật thường xuyên; các công cụ phát hiện âm mưu xâm nhập nhằm báo trước các ý đồ tiếp cận trái phép và giúp đỡ phục hồi những sự cố vốn không tránh khỏi; các công cụ kiểm tra tính toàn vẹn dữ liệu và thông tin tránh bị cá nhân bất hợp pháp và phương tiện khác thay đổi; công cụ chống virus,v.v;

Nguy cơ mất thông tin trong hệ thống TMĐT còn tiềm ẩn ngay trong cấu trúc phần cứng của các thiết bị tin học (tất nhiên không phải là tất cả) và trong phần mềm hệ thống và ứng dụng (kể cả phần mềm mật mã thương mại) do hãng sản xuất cài sẵn các loại "rệp" điện tử theo ý đồ định trước-thường gọi là "bom điện tử". Khi cần thiết, thông qua kênh viễn thông, người ta có thể điều khiển cho "nổ" tung thiết bị đang lưu trữ thông tin, hoặc tự động rẽ nhánh thông tin vào một địa chỉ đã định trước mỗi khi có sự truyền và xử lý thông tin của thiết bị (hay đang sử dụng phần mềm chương trình đó) đó trên mạng, thậm chí có thể điều khiển làm tê liệt hoặc làm tắc nghẽn hoạt động trao đổi thông tin của cả hệ thống mạng nếu cần, v.v.

Ngày nay, các chuyên gia đánh giá nguy cơ tiềm tàng nguy hiểm nhất đối với mạng máy tính mở là đạo tặc tin học, xuất hiện từ phía bọn tội phạm và giới tình báo. Nguy hiểm bởi: nó xuất phát từ phía những kẻ có chuyên môn cao và sử dụng kỹ thuật tinh vi (như đoán mật khẩu, khai thác các điểm yếu của hệ thống và các chương trình hệ thống, giả mạo địa chỉ IP, khai thác nguồn trên gói IP, đón lõng các trạm đầu cuối hoặc truy cập đang hoạt động, cài rệp điện tử, bơm virus máy tính phá hoại CSDL, sửa nội dung thông tin theo ý đồ đen tối của chúng, thậm chí nếu cần còn có thể làm tắc nghẽn kênh truyền, v.v); hoạt động của

[1]This content is available online at <http://voer.edu.vn/content/m13310/1.1/>.

chúng là có chủ đích và trên phạm vi rộng (như không những đối với từng cơ quan, doanh nghiệp mà còn đối với cả Chính phủ). Những tác hại mà chúng gây ra ảnh hưởng tới không chỉ riêng trong lĩnh vực kinh tế mà cả đối với lĩnh vực chính trị, an ninh-quốc phòng. Bởi vậy, vấn đề bảo mật/an toàn thông tin trong hệ thống TMĐT phải là cả một kế hoạch tổng thể của Chính phủ, không đơn thuần chỉ có lĩnh vực sử dụng mật mã.

8.1.1.1 Yêu cầu bảo mật thông tin cho các chủ thể tham gia TMĐT.

Việc phân tích các nguy cơ tiềm ẩn trên cho chúng ta thấy rằng vấn đề bảo mật thông tin của các chủ thể tham gia TMĐT là rất quan trọng trong việc hoạch định các phương án bảo mật thông tin trong hệ thống TMĐT. Các chủ thể tham gia TMĐT là người tiêu dùng, các doanh nghiệp (nhà nước và tư nhân) và Chính phủ, nhưng mối quan tâm bảo mật thông tin của các chủ thể tuy có mục đích giống nhau song yêu cầu thì hoàn toàn khác nhau và sự khác nhau về yêu cầu bảo mật thông tin có thể còn có ngay trong cùng một chủ thể do bởi thông tin giao dịch với các chủ thể khác nhau có nguy cơ đe doạ mất an toàn thông tin là khác nhau. Ví dụ như: Người tiêu dùng chỉ cần quan tâm đến bảo mật riêng tư và quá trình thanh toán ngân hàng; các doanh nghiệp quan tâm chủ yếu đến việc bảo mật thông tin mang tính cạnh tranh; đối với Chính phủ thì mối quan tâm bảo mật thông tin sẽ cao hơn, để một mặt chống được sự xâm hại của bọn đạo tặc kinh tế, mặt khác còn chống được việc ăn cắp thông tin của giới tình báo nước ngoài và bọn chống đối chế độ. Do đó, việc bảo mật thông tin cho các chủ thể tham gia TMĐT cần tính đến tính chất và yêu cầu của các chủ thể để xây dựng các phương án bảo mật thông tin tiết kiệm và hiệu quả. Không thể có một phương án chung cho mọi đối tượng.

Một khía cạnh khác rất quan trọng là cần phải tiên liệu trong quá trình xây dựng phương án bảo mật thông tin trong hệ thống TMĐT, đó là các luồng thông tin và các dạng thức thông tin giao tiếp trong hệ thống TMĐT để xác định phương thức tổ chức hệ thống kỹ thuật mật mã (KTMM) phù hợp nhằm bảo mật thông tin có hiệu quả. Có 4 loại giao tiếp thông tin trong hệ thống TMĐT: (1) người với người, (2) người với máy tính, (3) máy tính điện tử với người, và (4) máy tính điện tử với máy tính điện tử. Với 5 dạng thức chủ yếu: (1) thư điện tử, (2) thanh toán điện tử, (3) trao đổi EDI, (4) giao gửi số hoá các dung liệu-tức là việc trao đổi, mua bán hàng hoá thực hiện trực tuyến trên mạng bởi nội dung hàng hóa (còn gọi là hàng hóa mềm), (5) bán lẻ hàng hóa hữu hình. Trong các dạng thức trên, EDI dưới dạng có cấu trúc phải được đặc biệt chú ý bởi nó đã được Uỷ ban Liên hợp quốc về luật thương mại quốc tế (UNCITRAL) định nghĩa pháp lý sử dụng trên bình diện toàn cầu. Với các luồng thông tin và các dạng thức thông tin trao đổi phức tạp trong hệ thống TMĐT, có thể hình dung được những khó khăn trong việc bố trí KTMM và quản lý phân phối sử dụng khóa mật mã.

8.1.1.2 Các mâu thuẫn nảy sinh từ vấn đề sử dụng mật mã để bảo mật thông tin trong hệ thống TMĐT.

Như trên ta đã phân tích cho thấy lợi ích và tầm quan trọng của việc bảo mật thông tin trong hệ thống thông tin máy tính (và cũng là của TMĐT) là cần thiết. Song vấn đề sử dụng mật mã để bảo mật thông tin trong hệ thống TMĐT lại nảy sinh những vấn đề tranh cãi (không chỉ là vấn đề kỹ thuật, mà cả vấn đề pháp lý, thậm chí là cả vấn đề văn hóa đạo đức), không những trong khuôn khổ quốc gia, mà còn cả trên diễn đàn tổ chức quốc tế. Các mâu thuẫn gây tranh cãi nảy sinh từ yếu tố khách quan liên quan đến việc bảo vệ lợi ích chính đáng của mỗi chủ thể tham gia TMĐT và bảo mật thông tin cho TMĐT.

Trước hết, việc vừa bảo đảm khả năng tiếp cận rộng rãi của các chủ thể, vừa cung cấp mức độ hợp lý sản phẩm bảo mật thông tin đặc thù và các tài sản tính toán. Đây quả là một bài toán khó đối với cơ quan quản lý, cung cấp dịch vụ mật mã cho các chủ thể tham gia TMĐT. Sao cho làm hài hòa các lợi ích của 4 lực lượng: Người sử dụng, Người nghiên cứu, Người sản xuất, Nhà nước vì:

Người sử dụng (cá nhân, tổ chức) mong muốn thông tin của mình được bảo mật ở mức cao nhất, đồng thời lại không muốn sự kiểm soát của Chính phủ (nhất là các cơ quan an ninh) đối với tài sản thông tin riêng của mình;

Người nghiên cứu (nhà nước, tư nhân) mong muốn không có sự kiểm soát và hạn chế nào trong nghiên cứu và sử dụng mật mã để họ có một diễn đàn rộng rãi nhằm triển khai, chia sẻ và công bố kết quả

nghiên cứu;

Người sản xuất các sản phẩm mật mã muốn càng ít bị kiểm soát càng tốt để cho họ có được một thị trường rộng đối với các loại sản phẩm của họ sản xuất ra;

Nhà nước: vì lợi ích của cả cộng đồng và sự an ninh quốc gia, Nhà nước muốn việc triển khai mật mã trong hệ thống TMĐT không được làm ảnh hưởng đến khả năng phát hiện truy bắt tội phạm và làm phương hại đến lợi ích quốc gia.

Một nguyên nhân khác tác động từ bên ngoài cũng làm tăng mức độ mâu thuẫn trong các lực lượng nói trên, đó là việc tuyên truyền quảng cáo và thậm chí cung cấp miễn phí một số sản phẩm mật mã cho TMĐT của một số "cường quốc" về CNTT vì những mục đích riêng của họ, nên đã làm cho nhiều người lầm tưởng đây là một cơ hội cần phải tận dụng, bỏ qua lời khuyên của Oliver Lau: "Đừng quá dựa vào các sản phẩm của Hoa Kỳ bởi vì ngay từ đầu đã có nhiều mong muốn không an toàn về mặt chính trị, do đó các "cửa sau" của hệ thống bảo mật/an toàn bị bỏ ngỏ, v.v Hãy thận trọng, không hề có cái gọi là công khai đâu"

Để làm cân bằng các yêu cầu bảo mật thông tin của các chủ thể tham gia TMĐT và bảo vệ quyền lợi của quốc gia mình trước sự "xâm lược" thông tin của các cường quốc công nghệ cao, nên chính sách mật mã và cơ chế kiểm soát mật mã cho TMĐT của Chính phủ các nước là không giống nhau. Hiện nay, nội dung của các chủ đề trên đang gây tranh luận sôi nổi, không những trong nội bộ quốc gia, mà ngay trong diễn đàn quốc tế. Mặc dù vậy, Chính phủ các nước đều căn cứ vào trình độ phát triển và đặc thù tổ chức xã hội của nước mình để đề ra chính sách và các cơ chế kiểm soát việc sử dụng mật mã hữu hiệu để bảo vệ quyền lợi của quốc gia mình.

8.1.2 Các giải pháp an toàn thông tin trong thương mại điện tử

Vấn đề bảo mật thông tin trong hệ thống TMĐT được nhìn nhận một cách toàn diện như trên thực sự là một vấn đề phức tạp và bao hàm nhiều khía cạnh, nó không đơn giản như lời khuyên của một số chuyên gia nghiệp dư về CNTT là "*muốn tiếp cận với Internet thì hãy trang bị bức tường lửa, nếu cần sự bảo vệ thì hãy mã hóa và mật khẩu là đủ để xác thực*". Thực tế quy trình bảo mật thông tin trong hệ thống TMĐT muốn đạt hiệu quả thiết thực và tiết kiệm cần phải được hiểu theo khái niệm như là '*biết cách bảo vệ để chống lại sự tấn công tiềm ẩn*' ". Bởi vậy, nó phải là tổng hòa các giải pháp của hạ tầng cơ sở bảo mật. Đó là:

Pháp lý và tổ chức: trước hết là phải xây dựng chính sách mật mã cho TMĐT rõ ràng và có thể tiên liệu được, phản ánh được sự cân bằng quyền lợi của các chủ thể tham gia TMĐT, quan tâm tính riêng tư và an toàn xã hội, bảo đảm sự thi hành pháp luật và lợi ích an ninh quốc gia; ban hành các luật chứng cứ đối với "hồ sơ điện tử", tiêu chuẩn mật mã và chữ ký điện tử sử dụng trong TMĐT, giải quyết khiếu nại và tố cáo khi có sự tranh chấp liên quan đến sử dụng mật mã; tổ chức các cơ quan chứng nhận, cấp phép, quản lý và phân phối sản phẩm mật mã, phản ứng giải quyết sự cố, thanh tra và kiểm tra, vấn đề lưu trữ và phục hồi khoá, v.v;

Về kỹ thuật: Kết hợp chặt chẽ với hạ tầng công nghệ, quy định thống nhất tiêu chuẩn cấu trúc thiết lập hệ thống mạng và sử dụng công nghệ, ngôn ngữ giao tiếp và phần mềm ứng dụng, tổ chức hệ thống chứng thực và phân phối khóa mã, các công cụ nghiệp vụ kỹ thuật kiểm tra và phát hiện xâm nhập, dự phòng, khắc phục sự cố xẩy ra đối với KTMM sử dụng trong hệ thống TMĐT, v.v;

Về phía người sử dụng (tổ chức, cá nhân): trước hết phải được "giác ngộ" về bảo mật thông tin trong hệ thống TMĐT, họ cần biết phải bảo vệ cái gì trong hệ thống của họ, ước định mức rủi ro và các nguy cơ tiềm tàng khi kết nối mạng của mình với các đối tượng khác, việc mở rộng mạng của mình trong tương lai, v.v để họ có ý thức đầu tư bảo mật cho hệ thống của họ ngay từ khi bắt đầu xây dựng; chấp nhận và chấp hành chính sách, các quy định pháp luật về sử dụng mật mã, và phải chịu trách nhiệm trước pháp luật về bảo vệ bí mật quốc gia trong quá trình xử lý và truyền tải thông tin trong hệ thống TMĐT, v.v.

Với hệ thống thông tin mở, sử dụng công nghệ đa phương tiện như hiện nay thì về mặt lý thuyết không thể có vấn đề bảo mật thông tin 100%, điều cốt yếu là chúng ta phải biết tiên liệu được các nguy cơ tấn công tiềm ẩn đối với cái cần phải bảo vệ và cần bảo vệ như thế nào cho hiệu quả đối với hệ thống của mình. Cuối cùng, yếu tố con người vẫn là quyết định. Con người không được đào tạo kỹ năng và không có ý thức bảo mật cũng là kẽ hở cho những kẻ bất lương khai thác, và nếu con người trong hệ thống phản bội lại lợi ích của cơ quan, xí nghiệp và rộng hơn là của quốc gia thì không có giải pháp kỹ thuật bảo mật nào có hiệu

quả. Nói cách khác, bảo mật thông tin trong hệ thống TMĐT cần phải được bổ sung giải pháp an toàn nội bộ đặc biệt chống lại những đe doạ từ bên trong.

8.2 Ứng dụng kỹ thuật bảo mật trong các giao dịch điện tử

Có 5 hình thức thanh toán điện tử được sử dụng rộng rãi trong các giao dịch thương mại điện tử bao gồm [14]:

- Thanh toán qua chuyển khoản tại điểm bán bao gồm các giao dịch bằng thẻ tín dụng
- Thanh toán qua chỉ dẫn ngân hàng bao gồm việc trao đổi dữ liệu điện tử tài chính, điện thoại và hệ thống trả lời tự động
- Thanh toán bằng séc điện tử
- Thanh toán bằng thẻ thông minh
- Thanh toán bằng tiền mặt điện tử

Trong phần này chúng ta chỉ xem xét việc ứng dụng các kỹ thuật bảo mật đã học để bảo vệ thông tin séc điện tử và quá trình giao dịch sử dụng séc này. Hình 8.1 là quy trình mua hàng và thanh toán trên Internet sử dụng séc điện tử.

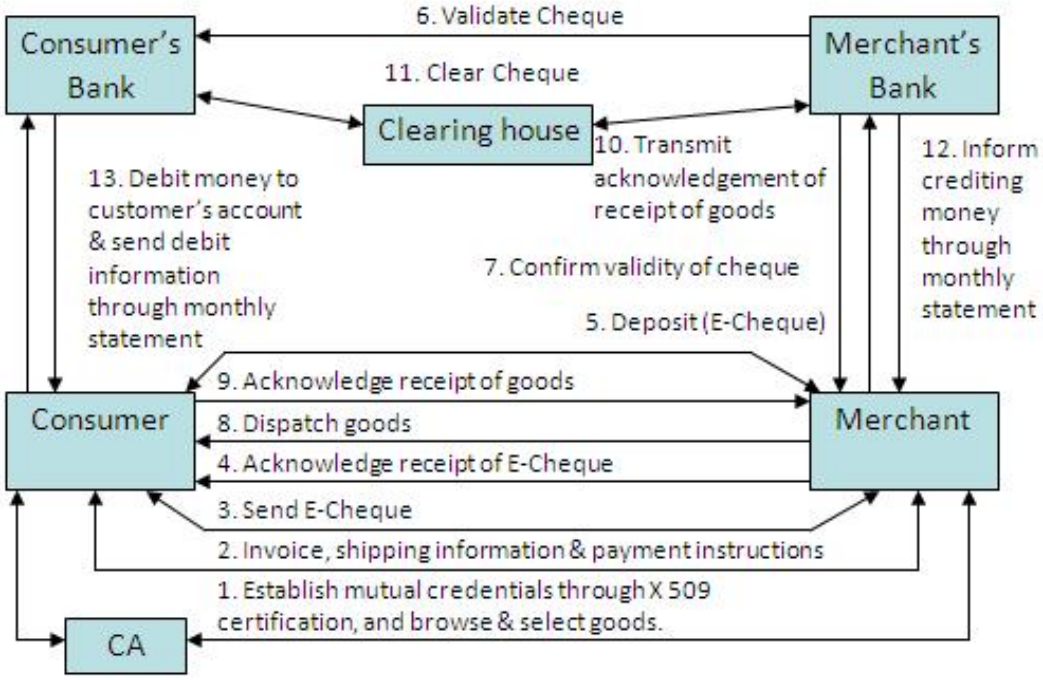
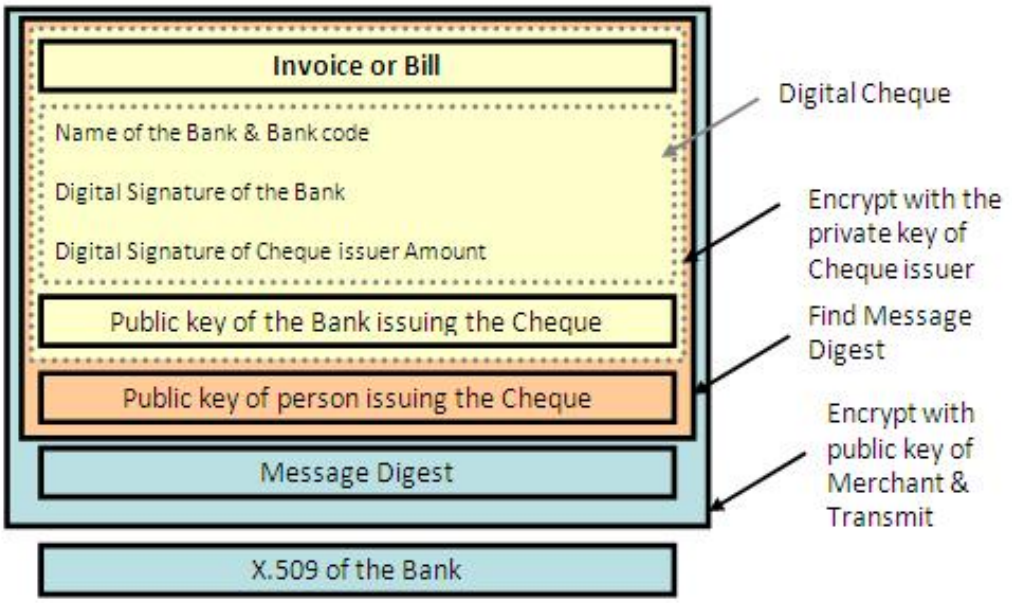


Figure 8.1

Quy trình mua bán sử dụng séc điện tử
mô tả cấu trúc của séc điện tử sử dụng kỹ thuật mã hóa, kỹ thuật băm và xác thực điện tử.



Cấu trúc séc điện tử

Figure 8.2

mô tả quy trình xử lý séc điện tử tại phía người bán hàng.

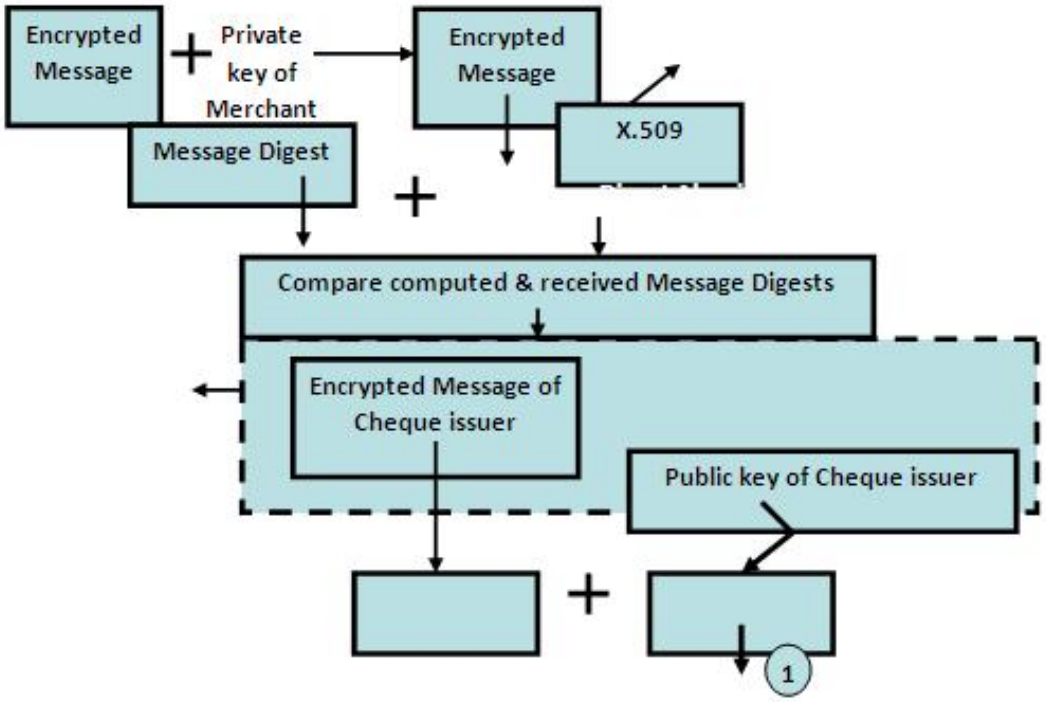

Encrypted Message
Private key of Merchant
Encrypted Message
X.509
Message Digest
Compare computed & received Message Digests
Encrypted Message of Cheque issuer
Public key of Cheque issuer
1

Figure 8.3

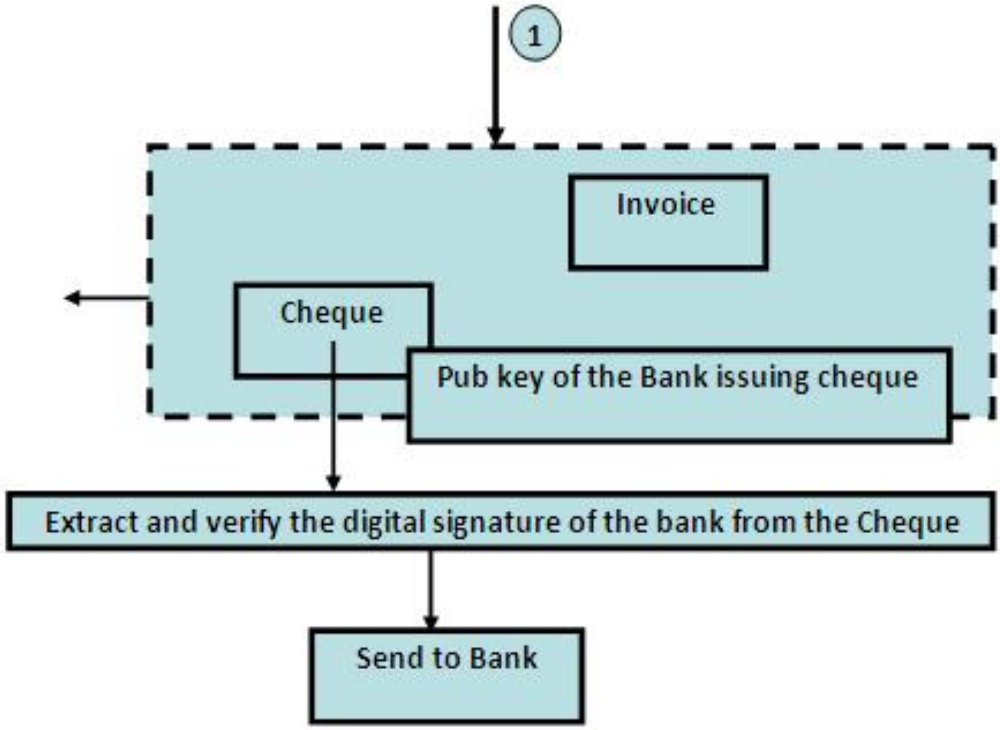


Figure 8.4: Quy trình xử lý séc điện tử

Chương 9

Virus máy tính[1]

9.1 Giới thiệu

9.1.1 Khái niệm virus máy tính

Virus là một thuật ngữ chung để chỉ các loại chương trình máy tính có một số đặc tính chung như khả năng xâm nhập, lây lan, phá hoại... Do đó khi nhắc tới virus máy tính chúng ta cần hiểu từ này theo nghĩa rộng để nó bao hàm các loại chương trình như virus, trojan horse, worm...Vì virus thường được tin tặc sử dụng như các phương tiện để tấn công các hệ thống máy tính của các tổ chức và cá nhân nên virus thường được hiểu với nghĩa tiêu cực. Trên thực tế, virus có thể được sử dụng trong các công việc hữu ích như chống sao chép phần mềm, chống lại các virus phá hoại khác...

Chương trình virus có cấu trúc như sau [28]:
Program V : =
{1234567;
Subroutine infect-executable:=
{loop: file = random-executable;
if (first-line of file = 1234567)
then goto loop;
else prepend V to file;}
Subroutine do-damage:=
{Bất cứ thứ gì bạn có thể lập trình}
Subroutine trigger-pulled:=
{Bất cứ thứ gì bạn muốn kích hoạt}
Main-program-of-virus:=
{infect-executable;
If (trigger-pulled) then do-damage;
Goto next;}
Next:
}
Ví dụ : Virus tấn công từ chối dịch vụ
Trigger-pulled:=
{if the date is after Jan 1, 1999;}
Do-damage:=
{loop: goto loop;}
Ví dụ : Virus nén file
Program V : =

[1]This content is available online at <http://voer.edu.vn/content/m13311/1.1/>.

{01234567;
Subroutine infect-exec:=
{loop: file = random-executable;
if first-line of file = 01234567
then goto loop;
(1) compress file;
(2) prepend CV to file;
}
Main-program:=
{if ask-permission
then infect-exec;
(3) uncompress rest-of-file;
(4) run uncompressed file;
}
}
Virus nén file

9.1.2 Lịch sử phát triển

Một số cột mốc đáng lưu ý trong lịch sử hình thành và phát triển của virus máy tính như sau [15]:

- Năm 1949 mô hình của một chương trình máy tính tự nhân bản ra đời
- Năm 1959 trò chơi Core War được phát triển
- Năm 1983 Thompson đưa ra ý tưởng về phương pháp làm virus
- Năm 1987 virus Lehigh xuất hiện lần đầu tiên trên máy IBM PC

9.1.3 Các tính chất cơ bản của virus

Các tính chất cơ bản, điển hình của virus bao gồm

1. Tính lây lan
2. Tính chất phá hoại
3. Tính nhỏ gọn
4. Tính tương thích
5. Tính phát triển kế thừa.

Bài tập: Xác định các tính chất điển hình của sâu Internet (Internet Worm).

9.2 Phân loại virus

Dựa vào một số đặc tính đặc trưng của virus chúng ta có thể phân loại chúng. Sau đây là một số cách phân loại điển hình. Mục đích là đưa ra cách phòng chống phù hợp với từng loại [16].

Phân loại theo đối tượng lây nhiễm và môi trường hoạt động
1. Virus boot
2. Virus file (a. DOS virus, b. Windows virus, c. macro virus)

Phân loại theo phương pháp tìm đối tượng lây nhiễm
1. Virus thường trú
2. Virus không thường trú

Phân loại theo phương pháp lây nhiễm
1. Ghi đè
2. Ghi đè bảo toàn

3. Dịch chuyển
4. Song hành
5. Nối thêm
6. Chèn giữa
7. Định hướng lại lệnh nhảy
8. Điền khoảng trống

Phân loại theo mức độ phá hoại
1. Virus thông thường
2. Virus huỷ diệt

Phân loại theo họ virus
1. Virus thời gian
2. Virus sự kiện ...

9.3 Nhắc lại Assembly

Cấu trúc chung một chương trình Assembly: (xem [17])
Segment_Name SEGMENT
...
Segment_Name ENDS
END
Một số lệnh cơ bản:

1. Lệnh truyền (di chuyển) dữ liệu

MOV *đích, nguồn*
Ví dụ: MOV AX, BX
MOV AX, [100]

1. Lệnh lôgic và số học

ADD *toán hạng đích, toán hạng nguồn*
ADC *toán hạng đích, toán hạng nguồn (Cờ nhớ CF)*
DEC AX ; giảm AX đi 1 đơn vị
CMP *toán hạng 1, toán hạng 2 (cờ nhớ ZF, CF)*
T1<T2 `[U+F0F3]` ZF=0 CF=0
T1=T2 `[U+F0F3]` ZF=1 CF=0
T1>T2 `[U+F0F3]` ZF=0 CF=1
Ví dụ: ADD AX, BX
ADC CX, DX
ADD BX, 100
ADD 2000, 10
CMP AX, BX
CMP SI, 120 ; so sánh SI với hằng số

1. Lênh dùng cho chuỗi kí tự

MOVSB, REP

1. Lệnh để quản lý đơn vị trung tâm

STD, CLD, NOP

1. Lệnh xuất/nhập hay lệnh vào/ra

IN AX, DX ; Dữ liệu ở cổng lưu trong thanh ghi DX sẽ được đọc vào AX
IN AL, 1A ; Cổng được chỉ định là các hằng số
OUT DX, AL ; Lệnh xuất dữ liệu ra cổng lưu trong DX
OUT 1A, AL ; Cổng được chỉ định là hằng số

1. Lệnh nhảy

JMP đích ; lệnh nhảy không điều kiện
JC 802 A ; nhảy khi CF=1
JNC 2308 ; CF=0
JZ 5508 ; ZF=1
JNZ 149B ; ZF=0
CALL *đích* ; gọi chương trình con

1. Lệnh ngắt

INT <số hiệu ngắt>

1. Lệnh xử lý ngăn xếp

PUSH <thanh ghi, từ> ; 16 bit
POP <thanh ghi, từ> ; 16 bit
Ví dụ: PUSH BX ; cất dữ liệu của BX vào ngăn xếp
POP DI ; tạo lại một thanh ghi chỉ số
Biên dịch chương trình assembly:

- Chương trình MASM
- Chương trình TASM

Bài tập: Viết chương trình Assembly đọc vào ký tự từ bàn phím và hiển thị lên màn hình, sau đó quay trở về DOS.

Đọc thêm: Các dấu hiệu máy tính bị nhiễm virus.

Chương 10

Một số mô hình bảo mật xử lí virus[1]

.

10.1 Khái niệm mã ngoại lai

Trong các môi trường trao đổi thông tin số, điển hình là Internet, các đoạn mã máy tính, gọi tắt là mã (*code*) được di cư từ máy này sang máy khác. Những đoạn mã di cư này được biết đến với tên gọi là mobile code, khi được xem xét để thực thi trên máy trạm chúng được coi là các mã ngoại lai (foreign code). Trong tài liệu này chúng ta thống nhất định nghĩa: *Mã ngoại lai là bất kỳ mã nào không phải sinh ra tại máy trạm nhưng bằng cách này hay cách khác tới được máy trạm và chạy trên đó.* Các loại Applets, ActiveX, các file đính kèm với thư điện tử, TclScript, JavaScript, PostScript, Word macros, và Excel macros là các ví dụ cho foreign code.

Khi nói về hệ thống kiểm chứng phân tán, Phillip [19] đã đề cập đến khái niệm về nơi tiêu dùng mã (*code consumer*) và nơi cung cấp mã (*code producer*). Trong giáo trình này chúng ta sẽ thống nhất dùng khái niệm code consumer như một môi trường cho phép chạy foreign code và nơi cung cấp mã là nơi foreign code được xuất bản. Ví dụ, trong kiến trúc web: Web server được xem xét như code producer. Web browser được xem như code consumer, và applet là foreign code.

10.2 Các vấn đề về bảo mật khi thực thi mã ngoại lai

Bản chất tự nhiên của foreign code là di trú, foreign code thường đến từ phía bên ngoài hệ thống máy trạm, đi qua nhiều môi trường tính toán khác nhau và thường là từ các môi trường không an toàn chẳng hạn như Internet. Chúng di cư đến máy trạm và thực thi tương tự như một chương trình máy tính thông thường. Việc người dùng không biết được xuất xứ, tác giả và hành vi của foreign code đã tiềm ẩn nhiều nguy cơ đối với hệ thống máy tính một khi chấp nhận chạy forein code. Dễ dàng hình dung virus máy tính là một loại foreign code độc hại. Xử lý foreign code cũng chính là xử lý virus máy tính.

Để bảo vệ một hệ thống máy tính, trước hết chúng ta cần phải kiểm soát việc truy xuất tới hệ thống. Sau đó là kiểm soát truy xuất bên trong hệ thống với các luật bảo mật [1]. Giải pháp là có cách nào đó giới hạn các truy xuất của foreign code tới các dữ liệu và tài nguyên của hệ thống. Điều chúng ta muốn làm thực sự có liên quan đến kiểm soát truy xuất. Chúng ta biết rằng mọi tiến trình muốn thực thi phải nhận được một môi trường nhất định. Ví dụ để thực thi, một tiến trình cần truy xuất bộ nhớ, truy xuất các tài nguyên của hệ thống, v.v. Đương nhiên một chương trình không bao giờ thực thi sẽ chẳng bao giờ làm hư hại đến hệ thống. Chương trình càng bị giới hạn truy xuất tới hệ thống thì hệ thống càng ít nguy cơ rủi ro. Do vậy nguyên tắc chung của chúng ta là nghiêm khắc kiểm soát việc truy xuất của các chương trình không bảo mật tới hệ thống. Để đạt được điều này chúng ta cần phát triển các luật bảo mật và thực thi các luật này

[1]This content is available online at <http://voer.edu.vn/content/m14612/1.1/>.

để bảo vệ hệ thống. Theo đó, có rất nhiều mô hình an toàn, bảo mật được đưa ra chẳng hạn Bell-LaPadula, các mô hình khác như Biba, Clack-Wilson dành cho sự toàn vẹn của hệ thống.

Để xử lý foreign code, chúng ta cũng đã được biết tới nhiều mô hình. Trước hết cần giả thiết rằng môi trường thực thi các luật bảo mật là tin cậy (*trusted*) và tất cả các loại foreign code là không tin cậy (*untrusted*). Trong thực tế chúng ta xử lý foreign code theo cách đầy mâu thuẫn, một mặt chúng ta muốn thực thi chúng một cách an toàn. Vì vậy sự truy xuất của foreign code tới các dữ liệu và tài nguyên của hệ thống phải bị nghiêm khắc giới hạn. Mặt khác chúng ta lại muốn chúng có nhiều sức mạnh hơn. Do đó, sự truy xuất của foreign code cần được nới rộng. Một vành đai bảo mật tưởng tượng đã được áp đặt lên các hệ thống máy tính. Trong đó mã cục bộ sinh ra tại máy trạm (*local code*) được xem xét là tin cậy và foreign code được xem là không tin cậy. Trường hợp lý tưởng nhất có lẽ là foreign code được xử lý như local code. Đây có lẽ là mục đích của hầu hết các giải pháp cho foreign code. Tuy nhiên tin cậy (*trust*) không có nghĩa là an toàn (*safe*). Tiếp theo chúng ta sẽ bàn về bốn mô hình bảo mật điển hình để xử lý foreign code.

10.3 Các mô hình bảo mật xử lý foreign code

10.3.1 Mô hình Sandboxing

Thuật ngữ "sandboxing" được đưa ra lần đầu tiên bởi nhóm của Wahbe (*được viện dẫn trong* [20]) để chỉ ra sự giam giữ một tiến trình trong miền sai (*fault domain*) của nó với mục đích là bảo đảm an toàn của bộ nhớ. Để hiểu rõ về mô hình này trong khía cạnh thực hành, chúng ta cùng xem xét mô hình sandbox nổi tiếng được phát triển cho Java.

Java sử dụng thuật ngữ "sandboxing" theo nghĩa rộng hơn để chỉ sự giới hạn truy xuất tới bất kỳ tài nguyên nào của hệ thống mà không chỉ đơn thuần là bộ nhớ. Mô hình bảo mật của Java dựa trên mô hình truy xuất tuỳ ý DAC với nhiều mức quyền truy xuất tới vùng được bảo vệ [5].

Mô hình bảo mật của Java gồm ba thành phần: bộ kiểm duyệt (*byte code verifier*), tải (*class loader*) và bộ quản lý bảo mật (*security manager*). Chúng ta có thể xem chi tiết hơn trong [21]. Theo mô hình cơ sở của access control, ba thành phần này làm việc như một bộ điều khiển tham chiếu. Mô hình bảo mật của Java (*JDK phiên bản 1.0*) nghiêm khắc giới hạn quyền truy xuất của tất cả các applet trong một cái sandbox (*hình 10.2*). Tuy nhiên việc này sẽ làm hạn chế các khả năng của applet. Do vậy phiên bản JDK 1.1 đã mở rộng hơn bằng cách đưa vào khái niệm mới signed applet. Đây là một khái niệm mới sử dụng công nghệ chữ ký điện tử chúng ta sẽ xem xét kỹ hơn ở phần sau.

Trong mô hình JDK 1.1, applet được chia thành hai loại. Signed applet được xử lý như mã cục bộ và unsigned applet được đưa vào xử lý trong sandbox.

Mô hình bảo mật hiện nay của Java (*JDK phiên bản 1.2*) phân loại foreign code ngay tại đầu vào thành các lớp khác nhau. Cấp đặc quyền cho mỗi lớp và xử lý từng lớp theo các đặc quyền đã cấp.

10.3.2 Mô hình Code Signing

Mô hình này có cách tiếp cận khác với mô hình chúng ta đã xét. Để bảo đảm sự an toàn cho hệ thống, nơi tiêu dùng mã (*code consumer*) phải nhận được sự cam đoan rằng nguồn sản xuất foreign code là đáng tin cậy và bản thân foreign code không bị làm giả. Yêu cầu này có thể được đáp ứng bằng cách sử dụng công nghệ chữ ký điện tử

Trong trường hợp tổng quát, quá trình mã hoá mã (*hay ký mã*) được hoàn thiện bởi một giải thuật chữ ký công khai/bí mật chẳng hạn như RSA. Như chỉ ra trong Hình 10.5. Bất cứ khi nào các tác giả muốn xuất bản mã (*foreign code*), họ phải sử dụng khoá bí mật riêng để ký mã. Tuy nhiên do bất tiện của việc xử lý với những đoạn mã dài. Hơn nữa còn cần bảo đảm tính toàn vẹn của mã. Đoạn mã dài cần được chia nhỏ và mỗi đoạn chia này sau đó sẽ được băm (hash) và ký riêng rẽ. Cuối cùng mã và các đoạn chia đã ký được xuất bản.

Tại phía code consumer, một quá trình so sánh được hoàn tất để đảm bảo các giá trị hash tính toán được tại đó và giá trị hash được gửi từ code producer là giống nhau. Khi đó foreign code sẽ được phép thực

thi, bằng không nó sẽ bị loại bỏ. Foreign code thoả mãn quá trình kiểm tra và được thực thi sẽ được xử lý như mã cục bộ nghĩa là không bị giới hạn truy xuất tới hệ thống.

10.3.3 Mô hình Firewalling

Hiện nay rất nhiều tổ chức đã và đang triển khai tường lửa (*firewall*) để bảo vệ mạng cục bộ trước những nguy cơ tấn công từ bên ngoài. Tường lửa có thể được triển khai dưới dạng phần mềm hoặc phần cứng. Nó nằm giữa mạng cục bộ hoặc máy trạm và phần còn lại của mạng Internet. Chức năng chính của tường lửa là điều khiển luồng dữ liệu đi vào hệ thống được bảo vệ. Nó sẽ lọc bỏ các chương trình không tin cậy và chỉ cho phép những chương trình tin cậy đi qua.

Cách tiếp cận này được biết đến như một cách kiểm tra sự an toàn của foreign code, các luật bảo mật được công thức hoá như các đặc tính dùng để phân tích [19]. Các đơn vị foreign code phải đi qua một chương trình phân tích để kiểm tra sự an toàn trước khi tới được code consumer. Chương trình phân tích này thường được gọi là bộ kiểm tra sẽ lọc ra các chương trình bị nó coi là không an toàn. Những đơn vị mã vượt qua được bộ kiểm tra này sẽ đến được máy trạm và được đối xử như mã cục bộ.

10.3.4 Mô hình Proof-Carrying Code

Để giải quyết vấn đề bảo mật cho foreign code. Các luật bảo mật phải được thống nhất từ trước giữa hai phía code consumer và code producer, theo đó code producer sẽ tạo ra một bằng chứng chứng minh sự an toàn của foreign code.

Trong tất cả các mô hình đã xem xét, cách tiếp cận mới này có vẻ như ít khả thi nhất vì sự mâu thuẫn xảy ra ngay trong bản thân các yêu cầu kỹ thuật. Một mặt chúng ta muốn mọi thứ thật đơn giản để không làm ảnh hưởng tới hiệu năng làm việc của hệ thống. Mặt khác chúng ta muốn bằng chứng (*proof*) phải bảo đảm. Sự thật là chẳng thể có bằng chứng gì đảm bảo với một công nghệ đơn giản.

Như mô tả trong hình, trước hết một thoả thuận về các luật bảo mật (*safe policy*) cần phải đạt được giữa code consumer và code producer. Sau đó code producer sẽ điền thêm các chú thích vào mã được biên dịch. Các luật bảo mật sẽ được áp dụng cho các mã được chú thích này để sinh ra một điều kiện kiểm chứng (*verification condition*) cho tính tương thích của bằng chứng. Tiếp theo, các luật bảo mật được tiếp tục sử dụng để mã hoá điều kiện kiểm chứng. Cuối cùng mã được chú thích và bằng chứng được sản sinh sẽ được gửi tới code consumer.

Một quá trình xử lý tương tự sẽ được thực hiện tại phía code consumer. Tuy nhiên thay vì sản sinh ra bằng chứng, quá trình sẽ là kiểm tra nó. Trong trường hợp bằng chứng là tương thích, mã sẽ được tích hợp vào hệ thống máy trạm.

10.3.5 Một số kết quả nghiên cứu khác

Đối với virus (*hay chính là các đoạn foreign code có hại*), theo Cohen [24,25,28], trong các môi trường không bảo mật chúng ta không thể ngăn ngừa được việc lây nhiễm. Vì không thể ngăn được việc lây nhiễm nên chúng ta chỉ có thể hy vọng phát hiện được chúng và giới hạn sự lây nhiễm xa hơn. Theo đó giải pháp hiệu quả nhất mà Cohen đưa ra hiện nay là Integrity Shell. Giải pháp này sử dụng công nghệ mã hoá để phát hiện ra sự thay đổi của thông tin. Nhờ đó phát hiện ra virus và ngăn ngừa các lây lan tiếp theo.

Một số xu hướng nghiên cứu mới theo quan điểm hạn chế tối đa ảnh hưởng của foreign code tới hệ thống bằng cách xây dựng các môi trường cách ly như Janus [26] và Deeds [27]. Đây là các môi trường trung gian giữa foreign code và hệ thống người dùng. Mọi yêu cầu truy xuất của foreign code tới hệ điều hành đều được điều khiển bởi các môi trường này do vậy bảo đảm sự an toàn cho hệ thống.

Chương 11

Một số loại virus máy tính điển hình

11.1 B-virus[1]

11.1.1 Cấu trúc đĩa cứng

11.1.1.1 Cấu trúc vật lý

- Track
- Side
- Cylinder
- Sector

11.1.1.2 Cấu trúc logic

- Boot sector
- FAT
- Root directory
- Bảng Partition

11.1.1.3 Dịch vụ truy nhập đĩa

- Mức BIOS (basic Input/Output System)
- Mức DOS

11.1.2 Phân tích B-virus

11.1.2.1 Đặc điểm

- B-virus triển khai kẽ hở của hệ thống để chiếm quyền điều khiển.
- Nạp trước hệ điều hành
- Không phụ thuộc vào môi trường

Cấu trúc của một b-virus thường bao gồm hai phần: phần cài đặt và phần thân
Phân loại
sb-virus:

[1]This content is available online at <http://voer.edu.vn/content/m14613/1.1/>.

- Chỉ dùng một sector thay chỗ boot record
- Cất boot record vào các sector cuối trong Root Directory hoặc trong đĩa mềm hoặc lưu trong các sector của track 0 trong đĩa cứng.

db-virus

- Chương trình chia làm hai phần, dùng nhiều sector

11.1.2.2 Các yêu cầu của một B-virus

- Tính tồn tại duy nhất
- Tính lưu trú
- Tính lây lan
- Tính phá hoại
- Tính gây nhiễm và ngụy trang
- Tính tương thích

11.1.2.3 Nguyên tắc hoạt động

Do chỉ được trao quyền điều khiển một lần khi boot máy, do đó b-virus phải tìm mọi cách để tồn tại và hoạt động giống như một chương trình thường trú. Chương trình thường gồm hai phần, một phần nằm tại boot record, phần còn lại nằm trên đĩa và được tải vào bộ nhớ khi virus được kích hoạt.

Phần install
Đã tồn tại trong bộ nhớ chưa →
↓
Đọc phần thân (db-virus)
↓
Nạp chương trình và lưu trú
↓
Chiếm các ngắt cứng (13, 8, 9)
↓
Trả boot sector cũ
↓
JMP FAR 0:07C00
Phần thân

- Phần lây lan
- Phần phá hoại
- Phần dữ liệu
- Phần boot record

11.1.2.4 Kỹ thuật lây lan

Đọc/Ghi →
↓
Đọc boot sector
↓
Đã nhiễm →
↓
Ghi boot sector của virus
↓
Ghi phần thân vào một vùng xác định →
(chi tiết xem [16])

11.1.3 Phòng chống và diệt B-virus

11.1.3.1 Phòng

Chúng ta nên cài đặt và sử dụng các chương trình phòng chống virus, đặc biệt cần nâng cao ý thức cảnh giác trong quá trình sử dụng máy tính chẳng hạn như luôn thực hiện việc sao lưu dữ liệu, kiểm tra đĩa mềm trước khi đưa vào máy, bật nấc chống ghi trên đĩa mềm, ...

11.1.3.2 Phát hiện

Việc phát hiện b-virus có thể tiến hành theo hai cách dựa vào đặc điểm của b-virus đó là kiểm tra virus trong vùng nhớ và trên đĩa.

11.1.3.3 Trong vùng nhớ

B-virus tồn tại trong vùng nhớ cao, việc phát hiện có thể qua các bước sau:

- So sánh tổng số vùng nhớ
- Dò tìm đoạn mã xác định của chương trình virus
- Có thể vô hiệu hoá virus bằng cách dành lại ngắt 013 cũ.
- Vô hiệu hoá virus và khởi động lại máy là phương pháp tốt nhất hiện nay

11.1.3.4 Trên đĩa

Việc dò tìm trên đĩa phải thực hiện sau khi kiểm tra vùng nhớ không phát hiện được virus. Việc phát hiện virus trên đĩa có thể tiến hành bằng nhiều cách:

- Dò tìm đoạn mã
- Kiểm tra key value

11.1.3.5 Gỡ bỏ B-virus

Sửa lại boot record theo các bước:

- Tìm nơi virus cất dấu boot sector
- Đọc và kiểm tra boot sector/partition trên cơ sở bảng tham số BPB (Bios Parameter Block) và dấu hiệu nhận dạng đĩa
- Khôi phục boot sector

11.2 Virus lây nhiễm trên file thi hành[2]

11.2.1 Nguyên tắc

- Virus phải chiếm quyền điều khiển trước khi trả lại cho file
- Dữ liệu của file phải được bảo toàn sau khi file nhận được điều khiển

[2]This content is available online at <http://voer.edu.vn/content/m14615/1.1/>.

11.2.2 Kỹ thuật lây nhiễm

Cách thức:
1. Chèn đầu
2. Chèn đuôi
3. Chèn giữa

Thuật toán:
mở file
Ghi lại thuộc tính
Lưu trữ các byte đầu tiên
Tính toán lệnh nhảy mới
Đặt lệnh nhảy
Chèn thân virus chính vào
Khôi phục thuộc tính
Đóng file

11.2.3 Phân tích một ví dụ virus lây nhiễm file COM

Toàn bộ chương trình và dữ liệu gói gọn trong một phân đoạn. Sau khi được nhận dạng, file COM được tải vào ngay sau PSP (Program Segment Prefix) mà không cần định vị lại (xem bài tập [15] trang 146).

11.2.4 Phân tích một ví dụ virus lây nhiễm trên file EXE

Chương trình EXE có thể nằm trong nhiều phân đoạn khác nhau do đó file EXE cần được định vị lại khi được nạp vào bộ nhớ bằng cách sử dụng các tham số trong một cấu trúc đầu file foi là exe header (xem bài tập [15] trang 152).

11.2.5 Cách phòng chống

Phát hiện

- Trong vùng nhớ
- Trên file

Chữa trị

- Chèn đuôi: định vị lại file
- Chèn đầu: tải toàn bộ file vào bộ nhớ rồi ghi lại lên đĩa

11.3 Virus macro[3]

11.3.1 Các macro

Macro là chương trình được viết bằng ngôn ngữ VBA.

11.3.2 Đặc điểm virus macro

Cần được kích hoạt bởi chương trình chủ.

[3]This content is available online at <http://voer.edu.vn/content/m14617/1.1/>.

11.3.3 Phân tích một ví dụ virus macro lây nhiễm tệp DOC

Ví dụ : Virus Melissa.

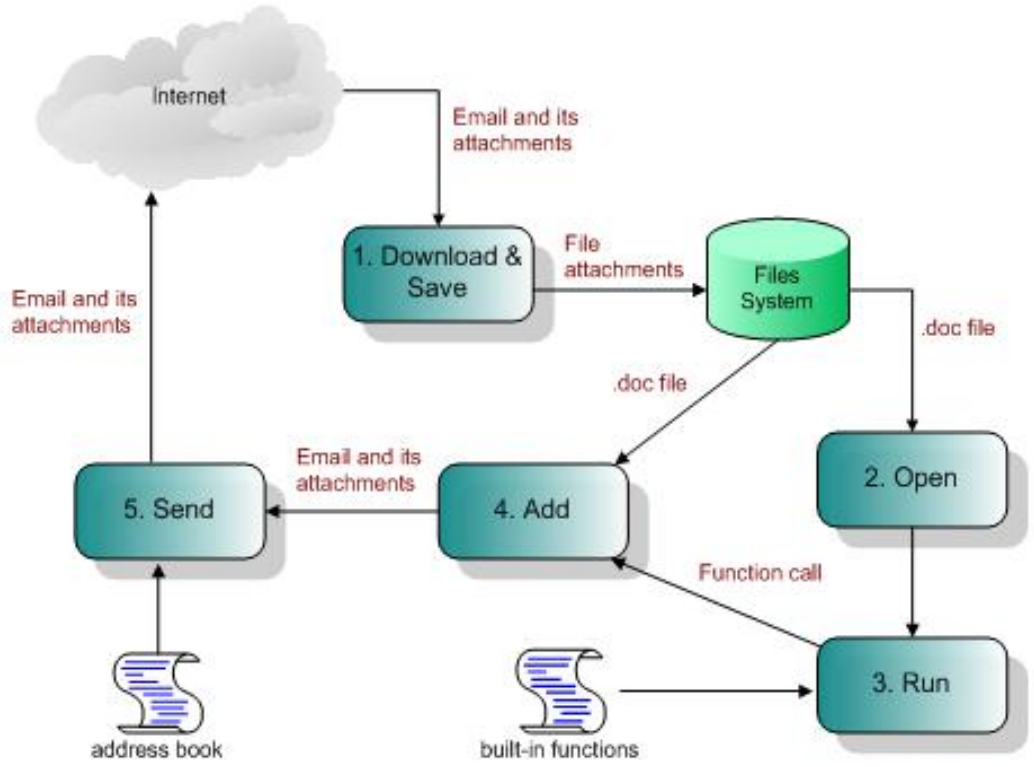


Figure 11.1: Các bước hoạt động của virus Melissa

11.3.4 Phòng chống virus macro

- Cài đặt các chương trình anti-virus
- Không đặt chế độ chạy tự động các macro
- Không mở email lạ...

11.4 Virus lây nhiễm qua thư điện tử[4]

Theo kết quả từ các đợt khảo sát quy mô lớn [29], các nguy cơ tiềm ẩn đối với người dùng email bao gồm:

- Máy tính bị nhiễm virus hoặc sâu – loại mã độc được phát tán thông qua file đính kèm hoặc nhúng trong nội dung email.

[4]This content is available online at <http://voer.edu.vn/content/m14618/1.1/>.

- Spam – thư rác.
- Web beaconing – quá trình kiểm tra địa chỉ email được kích hoạt khi người nhận mở email.

Cơ chế hoạt động

Virus được viết bằng ngôn ngữ nào đó và sử dụng thư điện tử như phương tiện lây lan. Chúng ta xem xét tổng thể hệ thống như sau:

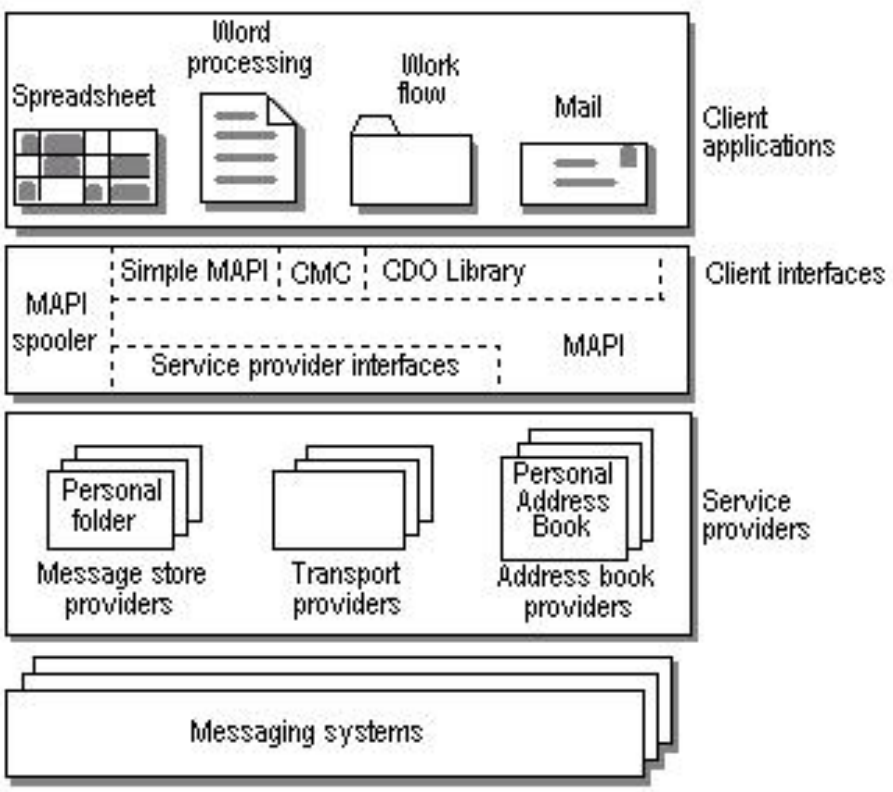


Figure 11.2

Mô hình MAPI

Ngoài việc triệu gọi trực tiếp các hàm API, các chương trình virus có thể sử dụng các dịch vụ trong mô hình đối tượng của Outlook để lây lan.

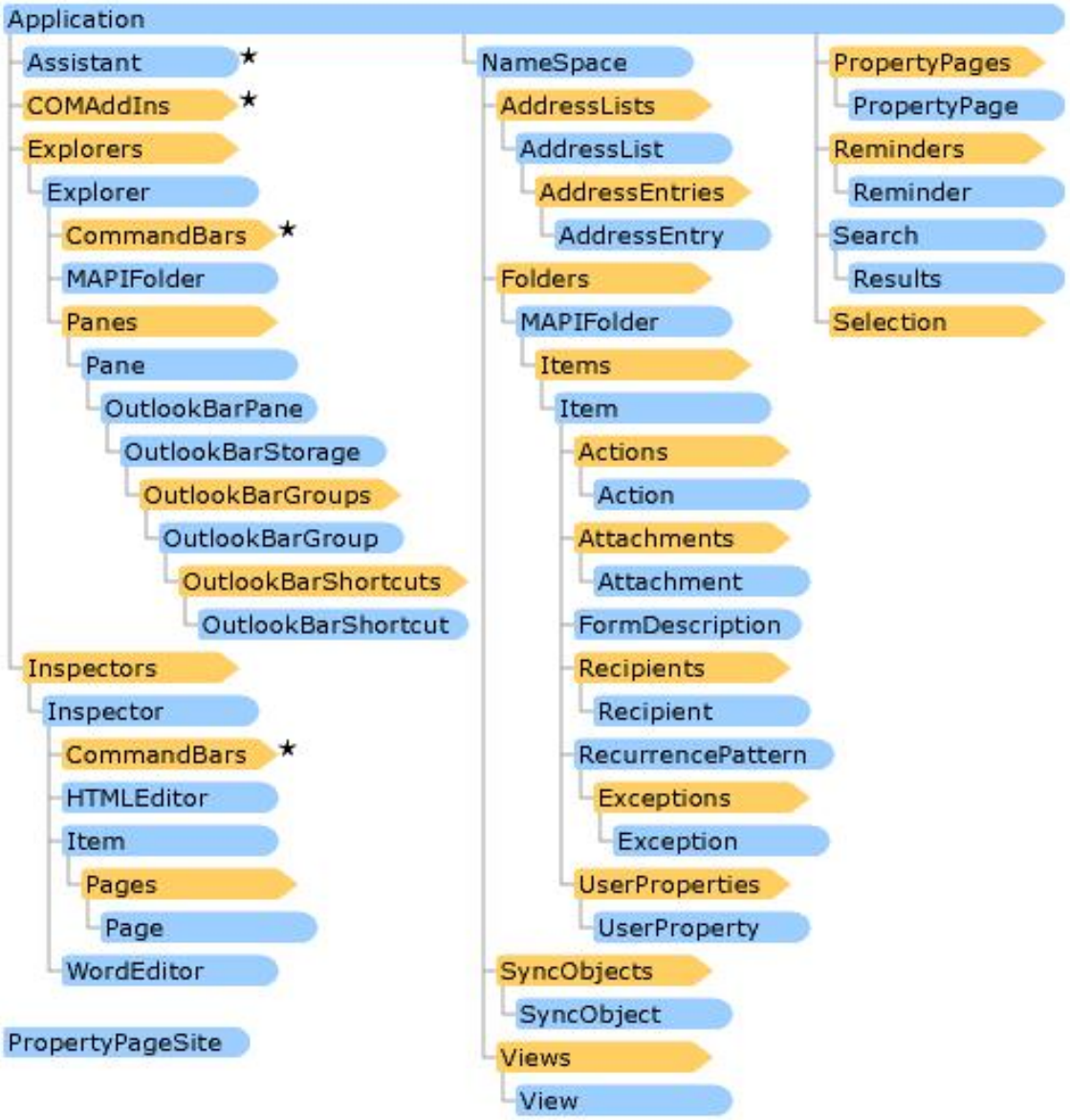


Figure 11.3

Mô hình đối tượng của Outlook

Ví dụ về virus thư điện tử

Phần này cung cấp thông tin về một số loại virus gần đây (trích dẫn từ [29]) được phát tán qua Microsoft Outlook hoặc được nhúng trong các email HTML nhận bởi Outlook. Hầu hết các virus lợi dụng Outlook theo một số cách thức giống nhau, vì vậy chúng ta chỉ đưa ra đại diện của mỗi loại.

11.4.1 Ví dụ : Virus BleBla

Virus này còn được biết đến với tên Romeo & Juliet, Verona hoặc TROJ_BLEBLA.A. Virus BleBla khai thác một số lỗ hổng trong quá trình xử lý email HTML của Outlook cho phép một virus tự động kích hoạt phần thân payload.exe được gửi kèm khi người dùng mở email. Thực tế nó sử dụng các thành phần Iframe bên trong thông điệp HTML để thực hiện tải hai file, MYJULIET.CHM và MYROMEO.EXE về thư mục TEMP của Windows. Một đoạn mã kịch bản nhỏ sẽ mở file MYJULIET.CHM, sau đó file này kích hoạt file MYROMEO.EXE – đoạn thân thực sự của virus. Quá trình này xảy ra được là nhờ khai thác một lỗi trong Windows.

Biện pháp phòng chống:

Cấu hình để Outlook không chạy các đoạn mã kịch bản trong các email HTML là biện pháp cốt lõi chống lại các loại virus này. Thực hiện các bước sau đây để đưa Outlook vào miền giới hạn và vô hiệu hóa các đoạn mã kịch bản trong miền giới hạn đó:

1. Use **Tools | Options | Security** to set the security zone for Outlook HTML mail to Restricted Sites.
2. Click the **Zone Settings** button, then **OK**.
3. Select **Custom**, and then click the **Settings** button.
4. On the **Security Settings** dialog box, choose **Disable** for all options under these headings:

- ActiveX Controls and plugins
- Scripting

1. Click **OK** three times to save the updated security settings.

11.4.2 Ví dụ : Virus LoveLetter

Windows Script Host (WSH) là một dịch vụ cho phép ta tạo và chạy các đoạn mã kịch bản rất hữu ích (.vbs – tương tự như các file batch của DOS, nhưng tốt hơn). Virus LoveLetter, còn có tên là Lovebug, I-Worm.Loveletter, ILOVEYOU, là một sâu viết bằng VBScript. Virus này sử dụng các thông điệp của Outlook để phát tán file VBScript.vbs (Tên đầy đủ của file là LOVE-LETTER-FOR-YOU.TXT.vbs). Để được kích hoạt, nó yêu cầu người dùng mở file đính kèm. Khi file được thực thi, nó vận hành các tiến trình nhằm ăn cắp thông tin nhạy cảm (download và sử dụng chương trình WIN-BUGSFIX.exe) và gửi tiếp email virus tới các nạn nhân tiếp theo có trong sổ địa chỉ của máy tính bị nhiễm.

Biện pháp phòng chống:

Để chống lại các loại sâu VBScript, cách hiệu quả nhất là không mở file đính kèm. Tuy nhiên, chúng ta còn có thể hoặc vô hiệu hóa Windows Script Host or hoặc cấm thực thi tự động các file VBS.

Chúng ta làm theo các chỉ dẫn dưới đây để loại bỏ các file có phần mở rộng là VBS trong danh sách các file đã được nhận biết của Windows 2000.

1. Open “My Computer”
2. Select “Tools/Folder Options”
3. Find “VBScript Script File” from the “File Types” tab
4. Select “Delete”
5. In confirmation dialog, select “Yes”.

11.4.3 Ví dụ : Stages Virus

Virus/sâu Stages tự lây lan giống như virus Loveletter, ngoại trừ việc sử dụng một loại file không phổ biến, đó là Shell Scrap Object [29]. Một scrap là một file (có phần mở rộng là .shs hoặc .shb) được tạo ra khi chúng ta thực hiện kéo thả một phần tài liệu ra ngoài màn hình desktop. Các đối tượng Scrap là các file MS Windows OLE được đóng gói chứa đựng hầu hết mọi thứ [29]. Các file này có thể chứa các đoạn mã có khả năng xóa file, thư mục hoặc chạy chương trình.

Biện pháp phòng chống:
Vô hiệu hóa các đối tượng scrap bằng một hoặc cả hai biện pháp sau:

- Sửa đổi hoặc xóa bỏ cả hai loại file (.shs và .shb) trong hộp thoại File Types.
- Xóa bỏ hoặc đổi tên file shscrap.dll trong thư mục Windows system.

Ví dụ : Virus Worm.ExploreZip
Theo [29], ExploreZip là một Trojan horse, bởi vì trước tiên nó yêu cầu nạn nhân mở hoặc chạy file đính kèm để tự cài đặt virus vào máy tính, tiếp theo thực hiện quá trình lây nhiễm ngầm mà nạn nhân không hay biết. Sau đó chương trình hoạt động giống như sâu Internet.
Trojan ExploreZip thực hiện việc lây lan giữa những người sử dụng qua con đường gửi email có đính kèm file zipped_files.exe. Mở file zipped_files.exe file sẽ kích hoạt chương trình virus. Chương trình thực hiện lây lan bằng cách tự động trả lời các thư mới nhận của máy bị lây nhiễm. Nội dung thư trả lời tương tự như email gốc đã mô tả ở trên.
Biện pháp phòng chống:
Không mở file đính kèm là cách hiệu quả nhất phòng chống loại virus này. Một trong các giải pháp khác được đề xuất trong [29] là không chia sẻ file WIN.INI.
Phòng và chống virus thư điện tử

- Cài đặt các chương trình anti-virus
- Update liên tục SP, livedate
- Không mở email lạ
- Mở thư dưới dạng plain text
- Cryptographic

File bị block by Outlook?
Một số loại virus và mã độc khác

- Phishing
- Spam
- Trojan
- Active content

11.5 Chiến lược phòng chống virus[5]

Trên thực tế có rất nhiều giải pháp bảo mật đã được triển khai để chống lại các nguy cơ từ virus máy tính. Tuy nhiên chưa có giải pháp nào thực sự là phương thuốc trị bách bệnh. Thiệt hại do virus máy tính gây ra ngày càng lớn đòi hỏi cần có các giải pháp hữu hiệu hơn để bảo vệ các hệ thống máy tính, đặc biệt là các hệ thống máy tính dịch vụ và thương mại.

Chúng ta hãy hình dung một kịch bản trong đó virus cố gắng bằng mọi cách xâm nhập vào hệ thống máy tính của chúng ta. Còn chúng ta thì nghĩ cách chống lại chúng. Để có thể phá hoại được một hệ thống, trước hết virus cần xâm nhập được vào hệ thống, rồi sau đó mới tìm các lỗ hổng bảo mật hoặc điểm yếu của hệ thống để triển khai tấn công. Rõ ràng kịch bản tấn công của virus mang đến cho ta hai chiến lược chính để phòng chống. Đó là ngăn ngừa virus xâm nhập hệ thống và giảm thiểu thiệt hại của hệ thống trong trường hợp bị xâm nhập. Nói theo khía cạnh về kiểm soát truy xuất thì có hai yếu tố cần được đảm bảo đó là kiểm soát việc truy xuất tới hệ thống và kiểm soát sự truy xuất bên trong chính hệ thống đó [1].

Hai kỹ thuật tương ứng để triển khai các chiến lược lần lượt là đặt các chốt chặn và triển khai mô hình sandbox. Với chiến lược thứ nhất, chúng ta phải đặt các chốt chặn tại các điểm vào của hệ thống cần bảo vệ. Các chốt chặn này thực chất là các chương trình kiểm tra, phân tích để phát hiện virus. Với mô hình sandbox, chúng ta không tập trung vào phát hiện virus mà tập trung nghiêm khắc giới hạn sự truy xuất của

[5]This content is available online at <http://voer.edu.vn/content/m14639/1.2/>.

các chương trình không tin cậy tới các dữ liệu và tài nguyên của hệ thống. Sự truy xuất tới hệ điều hành được kiểm soát chặt chẽ theo nguyên lý ít đặc quyền nhất (The Least Privilege). Nghĩa là chương trình chỉ được cấp quyền truy xuất tối thiểu để đủ thực hiện công việc. Việc giới hạn truy xuất vì thế sẽ giảm thiểu rủi ro cho hệ thống khi bị xâm hại .

11.5.1 Chốt chặn

Trước hết cần xác định ra các con đường (hay các điểm vào) từ đó virus có thể xâm nhập vào máy tính. Tại các điểm vào đó chúng ta đặt các chương trình kiểm tra (Hình 11.4). Ví dụ các chương trình firewall được cài đặt để kiểm soát luồng thông tin vào ra một hệ thống. Kỹ thuật ký mã (code signing) hoặc mã mang theo bằng chứng (proof-carrying code) được triển khai để kiểm tra tính tin cậy và toàn vẹn của thông tin đảm bảo thông tin đến từ các nguồn tin cậy và bản thân thông tin không bị làm giả trong quá trình di trú [29].

Tuy vậy chúng ta cần lưu ý rằng việc kết hợp sử dụng kỹ thuật firewall và code signing là không khả thi vì kỹ thuật firewall yêu cầu duyệt từng gói tin và có thể thêm vào đó các sửa đổi cần thiết. Trong khi kỹ thuật code signing không cho phép thông tin bị sửa đổi để đảm bảo tính toàn vẹn.

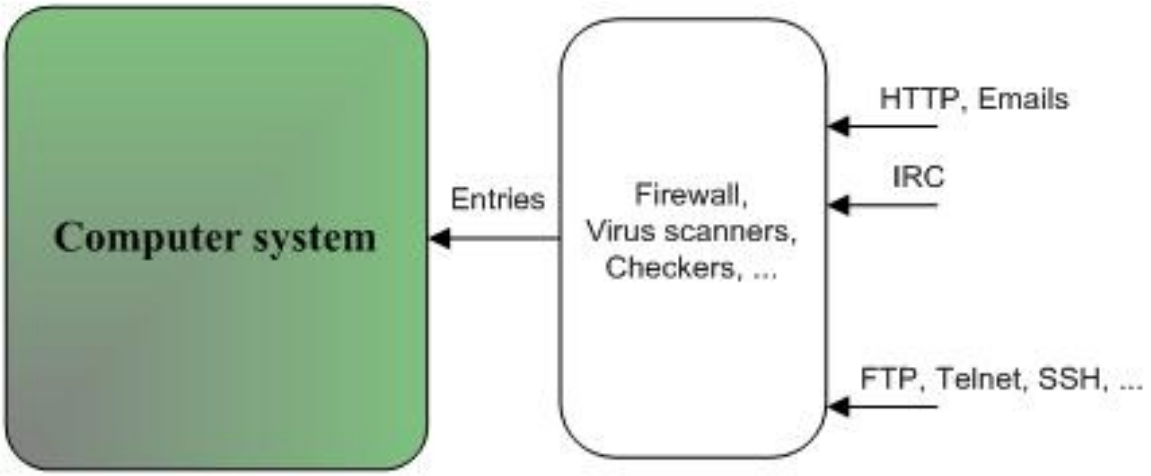


Figure 11.4: Mô hình chốt chặn

Mặc dù các kỹ thuật mới được triển khai rầm rộ song các phần mềm anti-virus ngày càng đóng vai trò quan trọng. Hàng chục ngàn virus đã tồn tại và virus mới xuất hiện hàng ngày hàng giờ. Nhận biết những virus này để bảo vệ máy tính là công việc vô cùng khó khăn nếu không có các chương trình anti-virus. Thực tế hầu hết các chương trình anti-virus đều triển khai kỹ thuật tìm kiếm kinh nghiệm (heuristics) để phát hiện virus. Kỹ thuật tìm kiếm kinh nghiệm bao gồm hai nhóm là tìm kiếm tĩnh (static) và tìm kiếm động (dynamic). Tìm kiếm tĩnh duyệt toàn bộ file, phân tích cấu trúc của nó và tìm kiếm các mẫu điển hình của virus. Sau đó sử dụng các thông tin này để đưa ra quyết định file có nhiễm virus hay không. Trong khi đó, tìm kiếm động thiết lập một môi trường kiểm soát ảo trong đó các file cần kiểm tra được mở ra (các file tài liệu) hoặc thực thi. Các hành vi lúc run-time của quá trình thử nghiệm sẽ được ghi lại, và dựa vào các luật bảo mật đã được thiết lập sẵn hệ thống sẽ đưa ra quyết định [30].

Chiến lược cơ bản để sử dụng kỹ thuật tìm kiếm kinh nghiệm là duy trì một cơ sở dữ liệu chứa các thông tin về virus. Tuy nhiên, nếu như thông tin về virus mới không được cập nhật đầy đủ (hoặc các luật bảo mật không đầy đủ trong trường hợp tìm kiếm động) thì các chương trình anti-virus trở thành vô dụng một khi

có virus hoàn toàn mới xâm nhập hệ thống.

11.5.2 Mô hình Sandbox

Như đã đề cập, các chương trình anti-virus không thể phát hiện được virus mới, do vậy không bảo vệ được máy tính trước những virus này. Giải thuật cho việc phát hiện mọi virus là khó và không khả thi [24]. Trong trường hợp này, chúng ta chỉ mong giảm thiểu được thiệt hại bằng việc giới hạn sự truy xuất của virus tới các dữ liệu và tài nguyên của hệ thống. Mô hình áp đặt các giới hạn này được biết đến với cái tên "Sandbox" (Hình 11.5).

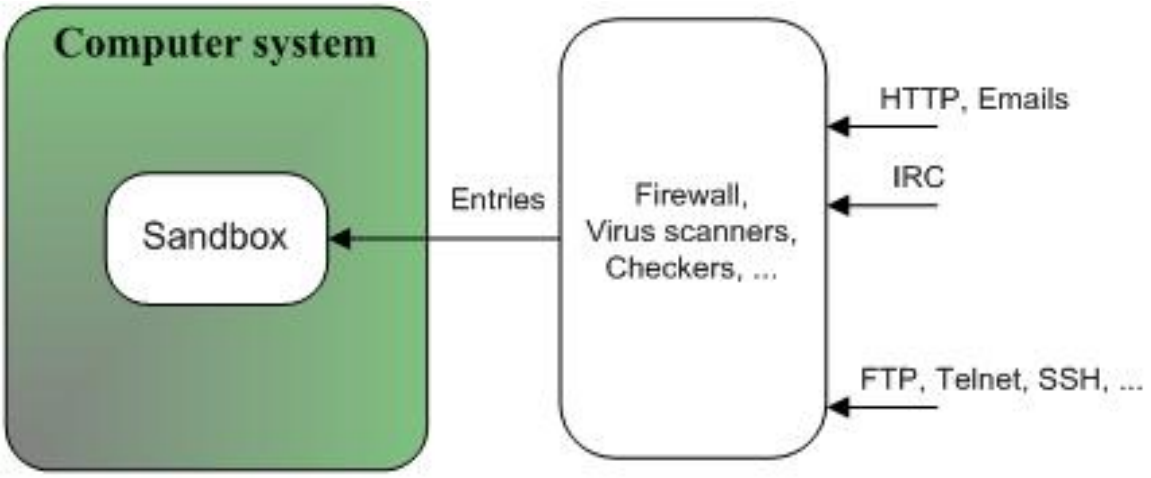


Figure 11.5: Mô hình sandbox

Có nhiều người cho rằng máy tính hay bị nhiễm virus là do hay kết nối vào internet. Nếu bạn thường xuyên duyệt Web có lẽ bạn cũng đồng ý như vậy. Nhưng không, nếu chúng ta chỉ duyệt web không thôi thì khả năng máy tính bị nhiễm virus là rất thấp. Ngay cả khi bạn bị tái điều hướng (redirect) tới một trang web bất thường và chạy một chương trình nào trên đó. Điều này có được là do hầu hết các trình duyệt web đều đã triển khai mô hình sandbox. Các loại mã ngoại lai hay các chương trình không tin cậy sẽ phải chạy dưới sự kiểm soát của mô hình này. Tiếp theo chúng ta cùng xem xét hai ví dụ điển hình của việc triển khai mô hình sandbox trên thực tế.

Ví dụ : Java applet

Java là một ngôn ngữ mạnh mẽ, bảo mật. Vấn đề về bảo mật với Java chỉ thực sự liên quan đến applet. Tận dụng khả năng của Java, tin tặc có thể viết các applet để triển khai các kiểu tấn công như đã xét. Mô hình sandbox cho Java có lẽ nổi tiếng nhất. Bất cứ khi nào người dùng duyệt trang web có chứa applet. Applet được trình duyệt tự động tải về và chạy dưới sự kiểm soát chặt chẽ của mô hình này.

Ví dụ : Công nghệ Active Content và các chương trình trợ giúp (helper applications)

Để xử lý nhiều loại dữ liệu khác nhau, trình duyệt cần triệu gọi các chương trình trợ giúp tương ứng với chúng. Bất cứ khi nào một tài liệu bất kỳ được gửi từ Web Server cho trình duyệt web, trình duyệt sẽ trình diễn nội dung của tài liệu đó hoặc trực tiếp thông qua các chương trình có sẵn (built-in) hay các chương trình kết hợp, hoặc gián tiếp qua việc triệu gọi các chương trình trợ giúp chẳng hạn như Word, Excel, Ghostscript. Không may, các chương trình trợ giúp này thường cồng kềnh và không được thiết kế với các tiêu chí bảo mật.

Các công nghệ mã ẩn (Active Content) cho phép các đoạn mã máy tính (gọi tắt là mã) dưới dạng các script, macro hoặc các loại mã mô tả tài liệu (postscript) nhúng trong các tài liệu được thực thi khi một tài liệu được mở. Trong lúc người sử dụng hiển thị tài liệu một cách bị động thì nội dung của tài liệu lại được thông dịch và trở nên được kích hoạt (active) vì vậy công nghệ này có tên gọi là Active Content [31]. Giống như bất kỳ một công nghệ nào khác, công nghệ mã ẩn có thể cung cấp những khả năng hữu ích cho việc thực hiện các dịch vụ thiết yếu nhưng nó đồng thời cũng trở thành một điểm yếu bảo mật, tạo điều kiện cho các cuộc tấn công của tin tặc. Ví dụ, chúng ta có một file Word được đính kèm với thư điện tử (file này có chứa một virus macro). Trong khi file này chưa được mở (tức là nó ở trạng thái chưa được kích hoạt) chẳng có gì hư hại đến bảo mật của hệ thống. Tuy nhiên tại thời điểm Microsoft Word mở file và cho phép macro thực thi. File Word ở trạng thái kích hoạt và đe doạ tới bảo mật của hệ thống.

Đối với trình duyệt, mô hình sandbox đã được triển khai để khắc phục các nguy cơ đến từ công nghệ mã ẩn. Chúng ta dễ thấy điều này khi mở một file Word, Excel hay Postscript trong trình duyệt. Khi đó trình duyệt chỉ đóng vai trò như một công cụ hiển thị, các chức năng thông thường của các chương trình trợ giúp bị loại bỏ.

11.5.3 Các công việc liên quan

Để tăng cường phát hiện virus, chúng ta đã xây dựng các phần mềm tự kiểm tra (built-in self-test) dùng kỹ thuật mã hoá. Integrity Shell được xem là kỹ thuật tối ưu nhất trong cách tiếp cận này [24]. Theo đó, nó có thể phát hiện ra sự lây nhiễm đầu tiên để ngăn ngừa các lây nhiễm thứ cấp.

Kỹ thuật sandbox cũng bao hàm nhiều cách thức triển khai khác nhau. Mô hình sandbox cho Java applet gồm máy ảo (Virtual Machine) được nhúng trong trình duyệt và các thư viện lớp hỗ trợ thực thi. Mô hình sandbox cho các chương trình trợ giúp là một môi trường cách ly được tạo ra bởi phần mềm, trong đó triển khai các luật bảo mật để kiểm soát hành vi của các chương trình. Ngay bản thân trình duyệt cũng là đối tượng kiểm soát của nó. Mặc dù vậy phương pháp này có hạn chế là đòi hỏi sự hỗ trợ của hệ điều hành để lưu vết và lọc các lời gọi hệ thống.

11.5.4 Kết luận

Chưa bao giờ thế giới lại dành nhiều quan tâm đến bảo mật như lúc này. Các cuộc tấn công gần đây gây hoang mang lớn trong cộng đồng người dùng. Trong khi một số người có đôi chút hiểu biết về các cuộc tấn công thì úp mở về chúng. Đối với người dùng, virus dường như thật đáng sợ và hiện hữu mọi lúc mọi nơi. Thế nhưng sự thật không hẳn chỉ có duy nhất bức tranh mầu tối đó. Virus chỉ có thể xâm nhập vào máy tính của người dùng bằng những con đường nhất định và sẽ là vô hại nếu bạn có thể kiểm soát được các con đường này.

Mặc dù các chiến lược chống virus mang lại những kết quả ngăn chặn nhất định nhưng bản thân chúng vẫn còn nhiều hạn chế. Chẳng hạn chiến lược chốt chặn có thể không phát hiện được virus, hoặc virus có thể làm giả các bằng chứng để qua mặt các chương trình kiểm tra...Ở mô hình sandbox, sự truy xuất của các chương trình bị giới hạn do vậy các tính năng của chúng sẽ bị hạn chế. Sự hội tụ của các chiến lược này có thể đưa đến một giải pháp toàn vẹn hơn, mang đến câu trả lời chính xác hơn cho vấn đề virus. Quả thực hiện nay chúng ta triển khai cùng một lúc rất nhiều các kỹ thuật khác nhau để bảo vệ máy tính. Đó cũng chính là xu hướng hiện tại để phòng chống virus.

Chương 12

Tài liệu tham khảo-Bảo mật máy tính[1]

1.	Dieter Gollmann. (1999). Computer Security. John Wiley & Sons, Inc.
2.	James, F.K. and Keith, W.R. (2003). Computer Networking: A Top-Down Approach Featuring the Internet. Pearson Education, Inc.
3.	Pierre Bieber and Fre'de'ric Cuppens. (2001). Expression of Confidentiality Policies with Deontic Logic. John Wiley & Sons, Inc.
4.	Zhiqing Liu. (2001). Manage Component-Specific Access Control with Differentiation and Composition. Retrieved April 2005, from Indiana University, website: http://www.cs.iupui.edu/∼zliu/doc/policy.pdf.
5.	Matthew J.Herholtz. (2001). Java's Evolving Security Model: Beyond the Sandbox for Better Assurance or Murkier Brew?. Retrieved May 2005, from Sans Institute, website:http://www.giac.org/practical/gsec/**Matthew Herholtz** GSEC.pdf.

continued on next page

[1]This content is available online at <http://voer.edu.vn/content/m16539/1.1/>.

6.	OWASP Project. (2001). A Guide to Building Secure Web Applications. Retrieved may 2005, from http://www.cgisecurity.com/owasp/html/ch08s03.html
7.	Prasad G. Naldurg. (2004). Modelling insecurity: Enabling recovery-oriented security with dynamic policies. Retrieved May 2005, from University of Illinois at Urbana-Champaign, website:http://www.cs.uiuc.edu/Dienst/Repository/2.0/Body/ncstrl.R-2004-2378/postscript.pdf
8.	Dung. P.M. (2005). Computer Security. (Lecture notes, Course No AT70.13, School of Advanced Technology). Asean Institute of Technology.
9.	Bell D.E. and LaPadula L.J. (1976). Secure computer system: unified exposition and Multics interpretation. The Mitre Corporation.
10.	Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein and Charles E. Youman. (1996). Role-based access control models. IEEE Computer, 29(2), 38-47.
11.	Stinson, D.R. (1995). Cryptography: Theory and Practice. CRC Press, Inc.
12.	William Stallings. (1999). Cryptography and Network Security: Principles and Practice. Prentice Hall, Inc.
13.	Phan Đình Diệu. (2002). Lý thuyết mật mã và an toàn thông tin. NXB ĐHQG Hà Nội.
14.	Aptech Limited. (2000). Concepts of E-commerce. Aptech certified E-commerce program. India: Author.
15.	Ngô Anh Vũ. (1999). Virus tin học – Huyền thoại và thực tế. NXB Thành phố Hồ Chí Minh.
16.	Nguyễn Viết Linh và Đậu Quang Tuấn. (2001). Hướng dẫn phòng chống Virus trong tin học một cách hiệu quả. NXB Trẻ.
17.	Nguyễn Mạnh Giang. (2005). Lập trình bằng ngôn ngữ Assembly cho máy tính IBM-PC. (Tái bản lần 3). NXB Giáo dục.

continued on next page

18.	Nguyễn Đình Hân. (2005). Bảo mật: Chung sống với mã ngoại lai. Thế giới vi tính – PC World Viet Nam, Sê-ri A - Công nghệ máy tính và mạng, 147, 102-104.
19.	Philip W. L. Fong.(2004). Proof Linking: A Modular Verification Architecture for Mobile Code Systems. (PhD Dissertation No V5A 1S6, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada). Retrieved May 2005, from http://www2.cs.uregina.ca/∼pwlfong/Pub/dissertation.pdf
20.	Valentin Razmov. (2002). Security in Untrusted Code Environments: Missing Pieces of the Puzzle. Retrieved May 2005, from University of Washington, website:http://www.cs.washington.edu/homes/valentin/papers/Secur UntrustedCode.pdf
21.	Sean Boran. (2002). The IT Security Cookbook. Retrieved May 2005, fromhttp://www.secinf.net/misc/the_it_security_cookbook/the_it_
22.	Satish R., Venkata R., Balaji T., Govindakrishnan K., Rajneesh M., Nilest P. and Sravan K. (2002). Security of Mobile Code. Retrieved April 2005, from University of Kentucky, website:http://cs.engr.uky.edu/∼singhal/term-papers/Mobilecode.pdf
23.	Tobias Fuchs. (2004). Proof-Carrying Code. Retrieved April 2005, from University of Muenchen, website:http://wwwbrauer.in.tum.de/seminare/security/SS04/fuchs-paper.pdf
24.	Frederick B. Cohen. (1991). A Cryptographic Checksum for Integrity Protection. **Computers and Security, 6(6), 505-510**.
25.	Frederick B. Cohen. (1991). Trends In Computer Virus Research. Retrieved May 2005, from http://all.net/books/integ/japan.html
26.	Ian Goldberg, David Wagner, Randi Thomas and Eric A. Brewer. (1996). A Secure Environment for Untrusted Helper Applications: Confining the Wily Hacker. Retrieved May 2005, from University of California, Bereley, website:http://www.cs.berkeley.edu/∼daw/papers/janus-usenix96.ps.

continued on next page

27.	Guy Edjlali, Anurag Acharya and Vipin Chaudhary. (1998). History-based Access Control for Mobile Code. Retrieved April 2005, from:http://www.pdcl.eng.wayne.edu/∼vipin/papers/deeds.ps.
28.	Frederick B. Cohen. (1994). A Short Course on Computer Viruses. (Second Edition). John Wiley & Sons, Inc.
29.	Nguyễn Đình Hân. (2005). Các chiến lược chống Virus. Thế giới vi tính – PC World Viet Nam, Sê-ri A - Công nghệ máy tính và mạng, 150, 108-110.
30.	Pavan Verma. (2005). Virus Protection. Retrieved May 2005, from EECS Department, University of Michigan, website:http://vx.netlux.org/lib/pdf/Virus Protection.pdf
31.	Wayner A. Jansen. (2001). Guidelines on Active Content and Mobile Code. Retrieved May 2005, from National Institute of Standard and Technology, U.S. Department of Commerce, website: http://csrc.nist.gov/publications/nistpubs/800-28-ver2/SP800-28v2.pdf

Table 12.1

Tham gia đóng góp

Tài liệu: *Bảo mật máy tính và mạng*
Biên soạn bởi: Khoa CNTT ĐHSP KT Hưng Yên
URL: http://voer.edu.vn/content/col10207/1.1/
Giấy phép: http://creativecommons.org/licenses/by/3.0/

Module: "Giới thiệu mục tiêu, nội dung, phương pháp học bảo mật máy tính"
Tác giả: Khoa CNTT ĐHSP KT Hưng Yên
URL: http://voer.edu.vn/content/m13179/1.1/
Trang: 1-2
Bản quyền: Khoa CNTT ĐHSP KT Hưng Yên
Giấy phép: http://creativecommons.org/licenses/by/3.0/

Module: "Một số khái niệm cơ bản trong bảo mật thông tin"
Tác giả: Khoa CNTT ĐHSP KT Hưng Yên
URL: http://voer.edu.vn/content/m13180/1.1/
Trang: 2
Bản quyền: Khoa CNTT ĐHSP KT Hưng Yên
Giấy phép: http://creativecommons.org/licenses/by/3.0/

Module: "Bảo mật thông tin: Các chủ đề làm tiểu luận"
Được sử dụng như là: "Các chủ đề làm tiểu luận"
Tác giả: Khoa CNTT ĐHSP KT Hưng Yên
URL: http://voer.edu.vn/content/m13182/1.1/
Trang: 2-4
Bản quyền: Khoa CNTT ĐHSP KT Hưng Yên
Giấy phép: http://creativecommons.org/licenses/by/3.0/

Module: "Nhận dạng và xác thực điện tử"
Tác giả: Khoa CNTT ĐHSP KT Hưng Yên
URL: http://voer.edu.vn/content/m13188/1.1/
Trang: 5-9
Bản quyền: Khoa CNTT ĐHSP KT Hưng Yên
Giấy phép: http://creativecommons.org/licenses/by/3.0/

Module: "Kiểm soát truy suất"
Tác giả: Khoa CNTT ĐHSP KT Hưng Yên
URL: http://voer.edu.vn/content/m13195/1.1/
Trang: 9-13
Bản quyền: Khoa CNTT ĐHSP KT Hưng Yên
Giấy phép: http://creativecommons.org/licenses/by/3.0/

Module: "Các mô hình bảo mật"
Tác giả: Khoa CNTT ĐHSP KT Hưng Yên
URL: http://voer.edu.vn/content/m13269/1.1/
Trang: 15-22
Bản quyền: Khoa CNTT ĐHSP KT Hưng Yên
Giấy phép: http://creativecommons.org/licenses/by/3.0/

Module: "Chiến lược phòng chống virus"
Tác giả: Khoa CNTT ĐHSP KT Hưng Yên
URL: http://voer.edu.vn/content/m14639/1.2/
Trang: 67-70
Bản quyền: Khoa CNTT ĐHSP KT Hưng Yên
Giấy phép: http://creativecommons.org/licenses/by/3.0/

Module: "Tài liệu tham khảo-Bảo mật máy tính"
Tác giả: Khoa CNTT ĐHSP KT Hưng Yên
URL: http://voer.edu.vn/content/m16539/1.1/
Trang: 71-74
Bản quyền: Khoa CNTT ĐHSP KT Hưng Yên
Giấy phép: http://creativecommons.org/licenses/by/3.0/

Hoc lieu Mo Vietnam - Vietnam Open Educational Resources

Học liệu mở Việt Nam là hỗ trợ việc quản lý, tạo, lưu trữ tài liệu giáo dục hiệu quả.