

Chương 1. TỔNG QUAN VỀ AN TOÀN BẢO MẬT HỆ THÔNG THÔNG TIN

Nội dung trọng tâm sau khi học xong chương này SV có thể:

- Trình bày khái niệm về HTTT, ATTT
- Sự cần thiết của đảm bảo an ninh, an toàn cho thông tin và HTTT.
- Trình bày các yêu cầu cần để đảm bảo an toàn HTTT, các thành phần trong HTTT.
- Mô hình tổng quát an toàn HTTT

TỔNG QUAN VỀ ATBMHTTT

- Giới thiệu về An toàn HTTT
- Các yêu cầu An toàn HTTT
- Bảy vùng trong cơ sở hạ tầng CNTT và mối đe dọa ATHTTT
- Mô hình tổng quát đảm bảo ATHTTT

1. GIỚI THIỆU VỀ ATBMHTTT

- Hệ thống thông tin (IS – Information System) là một hệ thống tích hợp các thành phần nhằm phục vụ việc thu thập, lưu trữ, xử lý thông tin và chuyển giao thông tin, tri thức và các sản phẩm số (con người, hd xử lý dữ liệu, thông tin trong 1 tổ chức).

- **Bảo mật HTTT (Information Systems Security):** là bảo vệ HTTT chống lại việc truy cập, sử dụng, chỉnh sửa, phá hoại, làm lộ và làm gián đoạn thông tin và hoạt động của hệ thống một cách trái phép.

1. GIỚI THIỆU VỀ ATBMHTTT

- Các doanh nghiệp và các tổ chức sử dụng các HTTT để thực hiện và quản lý các hoạt động:
 - Tương tác với khách hàng.
 - Tương tác với các nhà cung cấp.
 - Tương tác với các cơ quan chính quyền.
 - Quản bá thương hiệu sản phẩm.
 - Cạnh tranh với các đối thủ trên thị trường.

1. GIỚI THIỆU VỀ ATBMHTTT

- HTTT gồm 4 loại theo đối tượng sử dụng:
 - Hệ thống xử lý giao dịch với người sử dụng là các nhân viên.
 - Hệ thống thông tin quản lý với người sử dụng
 - Hệ thống trợ giúp ra quyết định với người sử dụng là các quản lý cao cấp.
 - Hệ thống thông tin điều hành với người sử dụng là các Giám Đốc điều hành.

1. GIỚI THIỆU VỀ ATBMHTTT

- Một số HTTT điển hình:
 - Các kho dữ liệu.
 - Các hệ lập kế hoạch nguồn lực doanh nghiệp.
 - Các hệ thống thông tin doanh nghiệp.
 - Các hệ chuyên gia.
 - Các máy tìm kiếm.
 - Các hệ thống thông tin địa lý.
 - Các hệ thống thông tin toàn cầu.
-

1. GIỚI THIỆU VỀ ATBMHTTT

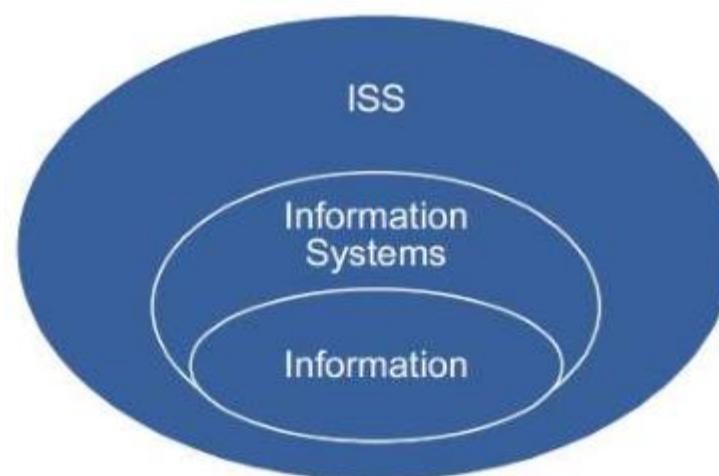
- Một HTTT dựa trên máy tính là một HTTT sử dụng công nghệ máy tính để thực thi các nhiệm vụ.
- Các thành phần của HTTT dựa trên máy tính gồm:
 - Hardware: phần cứng để thu thập, lưu trữ, xử lý và biểu diễn dữ liệu.
 - Software: các phần mềm chạy trên phần cứng để xử lý dữ liệu.
 - Databases: lưu trữ dữ liệu.
 - Networks: hệ thống truyền dẫn thông tin/dữ liệu.
 - Procedures: tập hợp các lệnh kết hợp các bộ phận nêu trên để xử lý dữ liệu, đưa ra kết quả mong muốn.

1. GIỚI THIỆU VỀ ATBMHTTT

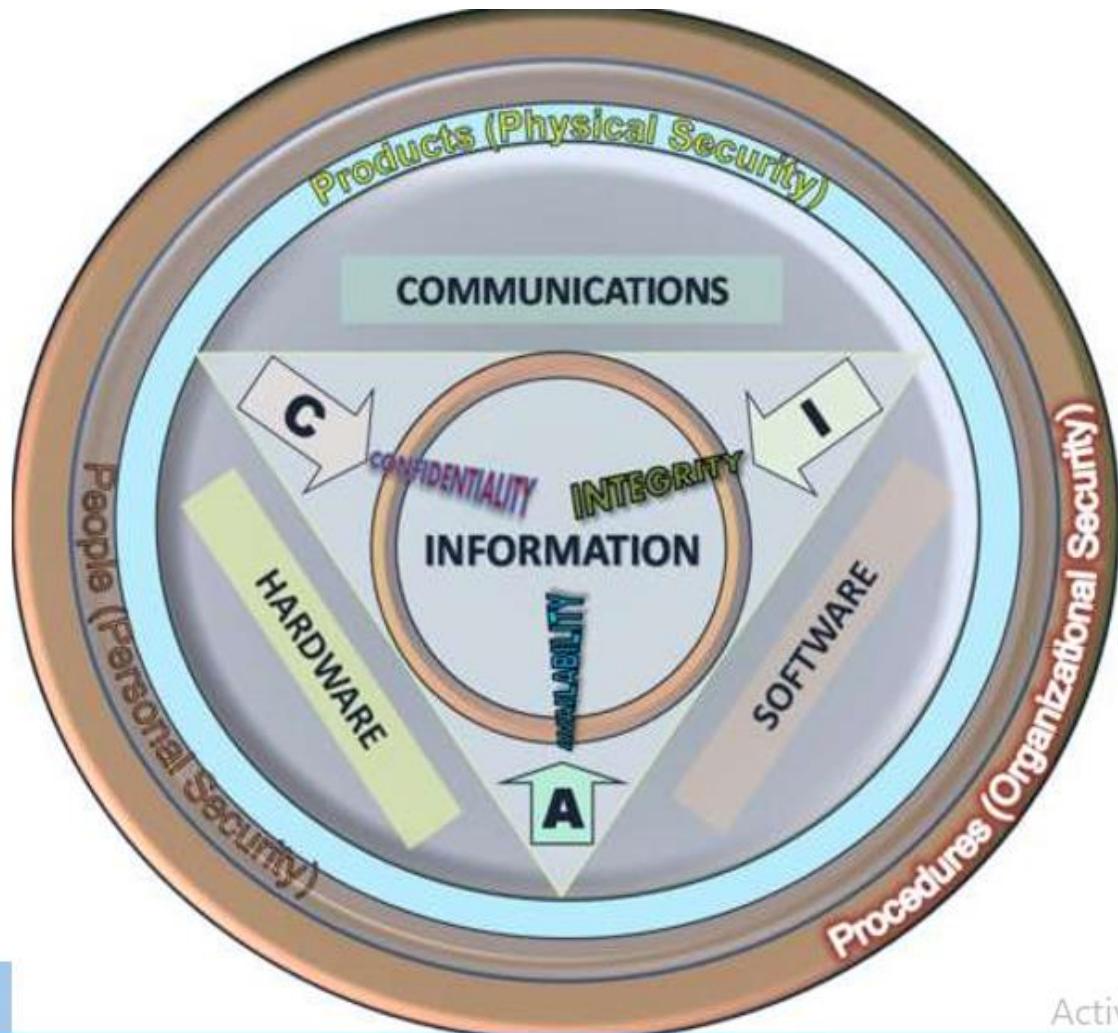
- An toàn thông tin:
 - An toàn công nghệ thông tin: là việc bảo vệ chống truy cập, sử dụng, tiết lộ, sửa đổi, hoặc phá hủy thông tin một cách trái phép.
 - Hai lĩnh vực chính của ATTT:
 - + An toàn công nghệ thông tin (an toàn máy tính): là ATTT áp dụng cho các hệ thống công nghệ.
 - + Các hệ thống thông tin của 1 tổ chức cần được đảm bảo an toàn.
 - Đảm bảo thông tin không bị mất khi xảy ra các sự cố (thiên tai, hỏng hệ thống, trộm cắp, phá hoại,...)

1. GIỚI THIỆU VỀ ATBMHTTT

- ATHTTT (ISS – Information Systems Security): đảm bảo các thuộc tính an ninh an toàn của hệ thống thông tin: bí mật (Confidentiality), Toàn vẹn (Integrity), sẵn dùng (Availability); bảo vệ HTTT chống lại việc truy cập, sử dụng, chỉnh sửa, phá hoại, làm lộ và làm gián đoạn thông tin và hoạt động của hệ thống một cách trái phép



❖ An toàn hệ thống thông tin (ISS)



2. CÁC YÊU CẦU ĐẢM BẢO ATHTTT

- Tính bí mật: bảo vệ dữ liệu không bị lộ ra ngoài một cách trái phép, chỉ người dùng có thẩm quyền mới được truy cập thông tin.
- Các thông tin bí mật gồm:
 - Dữ liệu riêng cá nhân.
 - Các thông tin thuộc quyền sở hữu trí tuệ của các doanh nghiệp hay cơ quan/tổ chức.
 - Các thông tin có liên quan đến an ninh quốc gia.

- Ví dụ: trong hệ thống ngân hàng, một khách hàng được phép xem thông tin số dư tài khoản của mình nhưng không được phép xem thông tin của khách hàng khác.

2. CÁC YÊU CẦU ĐẢM BẢO ATHTTT

- Tính toàn vẹn (Integrity): thông tin chỉ có thể được sửa đổi bởi những người dùng có thẩm quyền.
- Tính toàn vẹn liên quan đến tính hợp lệ và chính xác của dữ liệu.
 - Trong nhiều tổ chức, thông tin có giá trị rất lớn như: bản quyền pm, bản quyền phát minh, sáng chế,....
- Dữ liệu là toàn vẹn nếu:
 - Dữ liệu không bị thay đổi.
 - Dữ liệu hợp lệ.
 - Dữ liệu chính xác.

- Ví dụ: trong hệ thống ngân hàng, không cho phép khách hàng tự ý thay đổi thông tin số dư tài khoản của mình.

2. CÁC YÊU CẦU ĐẢM BẢO ATHTTT

- Tính chống thoái thác (Non-repudiation): khả năng ngăn chặn việc từ chối một hành vi đã làm.

Ví dụ: trong hệ thống ngân hàng, có khả năng cung cấp bằng chứng để minh chứng một hành vi khách hàng đã làm như rút tiền, chuyển tiền.

2. CÁC YÊU CẦU ĐẢM BẢO ATHTTT

- Tính sẵn sàng (Availability): thông tin có thể truy cập bởi người dùng hợp pháp bất cứ khi nào họ có yêu cầu.
- Tính sẵn sàng có thể được đo bằng các yếu tố:
 - Thời gian cung cấp dịch vụ (Uptime).
 - Thời gian ngừng cung cấp dịch vụ (Downtime).
 - Tỷ lệ phục vụ: = Uptime/(Uptime+Downtime)
 - Thời gian trung bình giữa các sự cố.
 - Thời gian trung bình ngừng để sửa chữa.
 - Thời gian khôi phục sau sự cố.

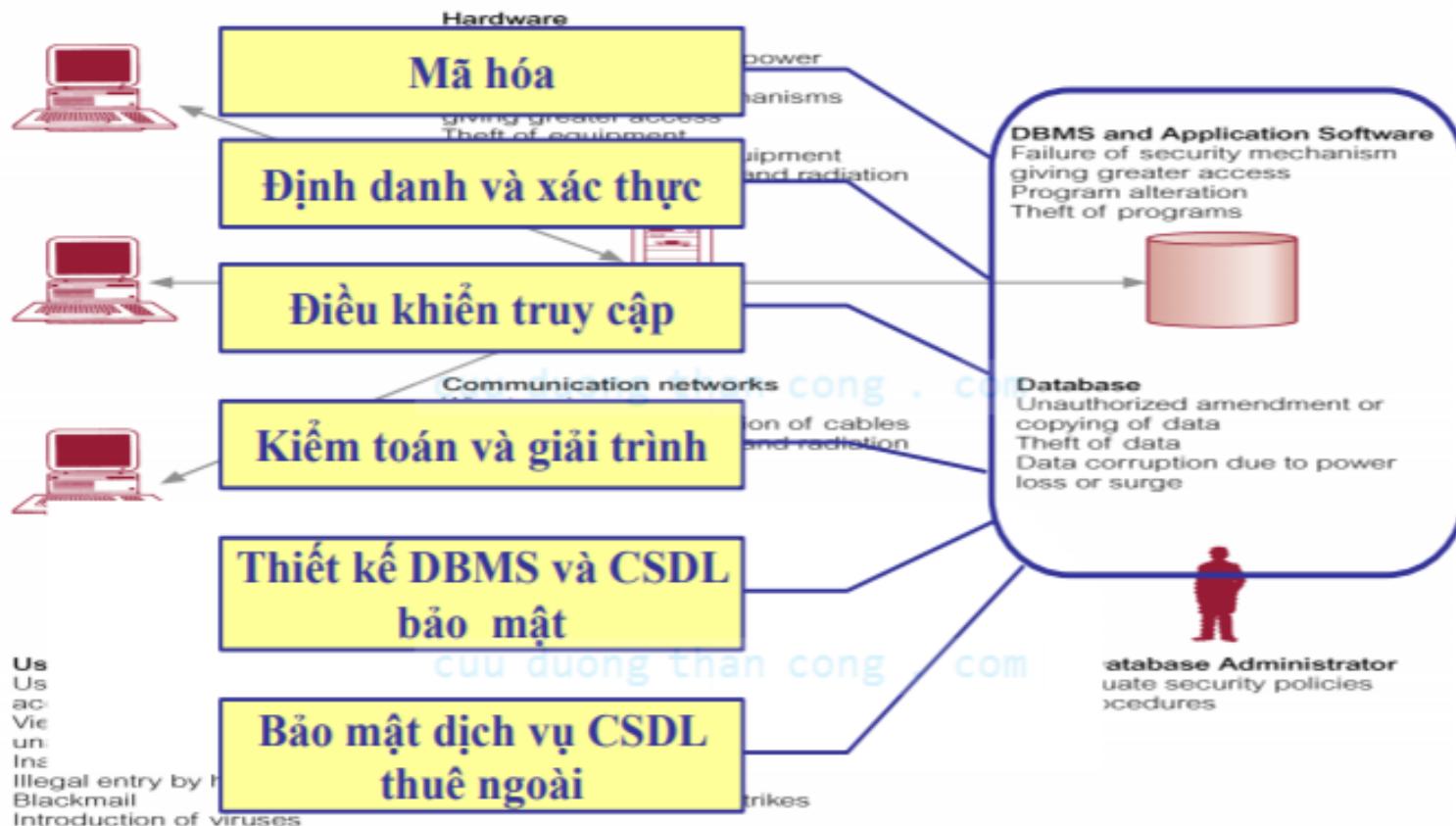
- Ví dụ: trong hệ thống ngân hàng, cần đảm bảo rằng khách hàng có thể truy vấn thông tin số dư tài khoản bất kỳ lúc nào theo như quy định.

3. CÁC THÀNH PHẦN HTTT

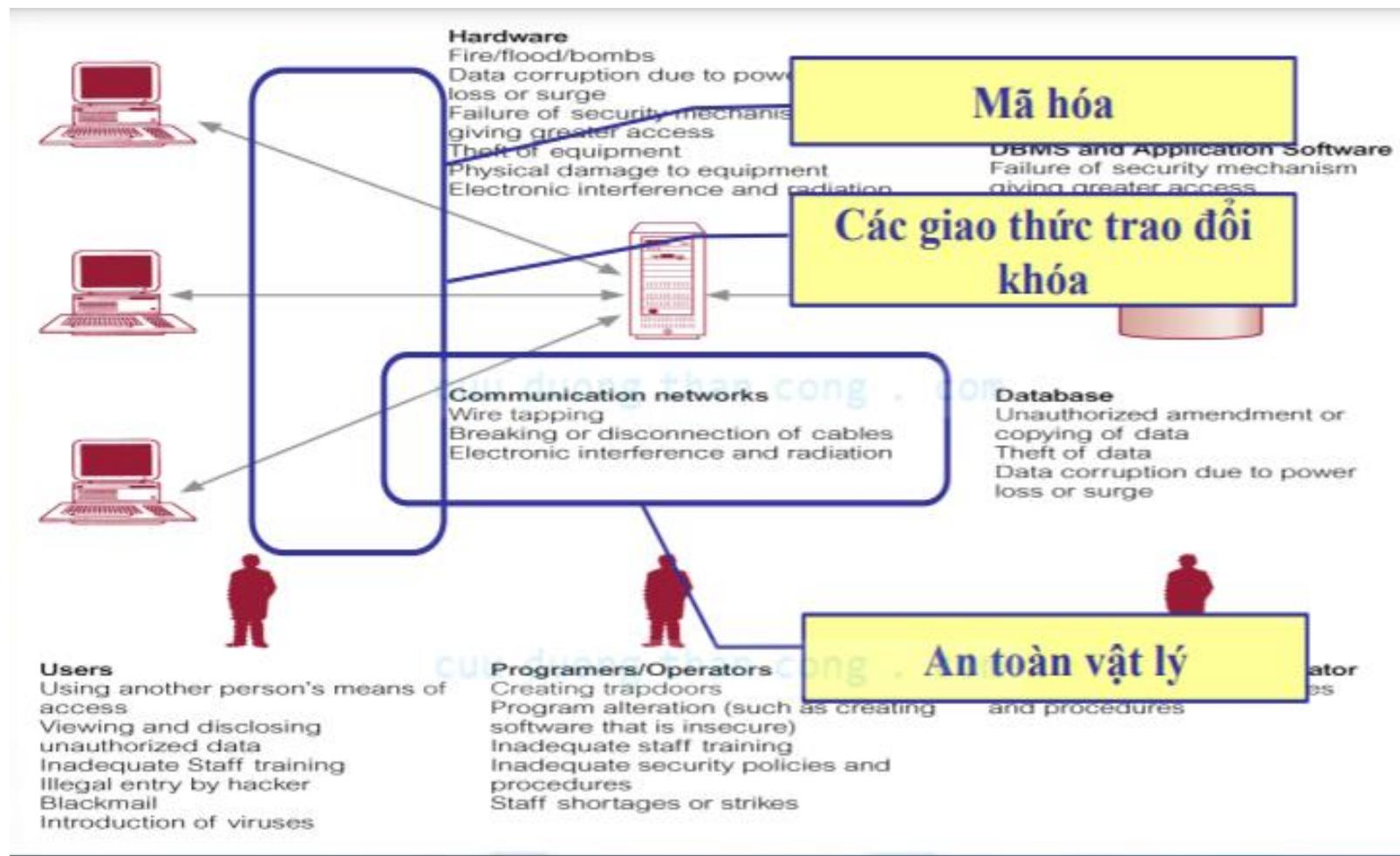
- Phần cứng
- Mạng
- Cơ sở dữ liệu (CSDL)
- Hệ QT CSDL, các ứng dụng
- Người dùng
- Người lập trình hệ thống
- Người quản trị CSDL

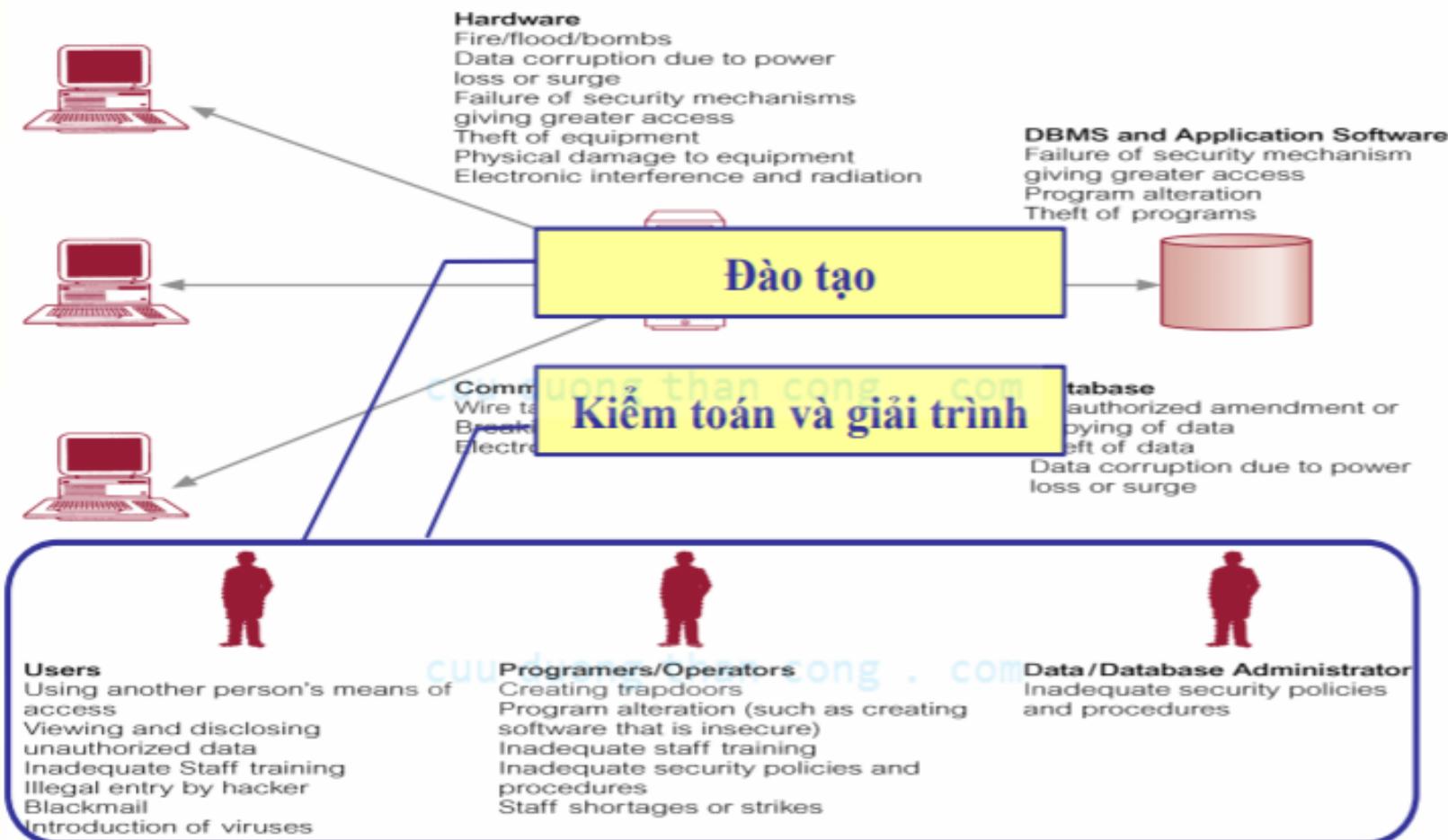
Các tp cần bảo vệ trong HTTT

Các thành phần cần bảo vệ trong một HTTT



Các tp cần bảo vệ trong HTTT





Các đe dọa (threats) với người dùng)

- Thiếu ý thức về vấn đề an ninh an toàn.
- Coi nhẹ các chính sách an ninh an toàn.
- Vi phạm chính sách an ninh an toàn.
- Đưa CD/DVD/USB với các file cá nhân vào hệ thống.
- Tải ảnh, âm nhạc, video,...
- Phá hoại dữ liệu, ứng dụng và hệ thống.
- Tấn công phá hoại từ các nhân viên bất mãn.
- Nhân viên có thể tống tiền hoặc chiếm đoạt thông tin quan trọng.

Các đe dọa (threats) với vùng máy trạm:

- Truy cập trái phép vào máy trạm.
- Truy cập trái phép vào hệ thống, ứng dụng và dữ liệu.
- Các lỗ hỏng an ninh trong các phần mềm ứng dụng máy trạm.
- Các hiểm họa từ Virus, mã độc và các phần mềm độc hại.
- Người dùng đưa CD/DVD/USB với các file cá nhân vào hệ thống.
- Người dùng tải ảnh, âm nhạc, video,...

Các đe dọa với vùng LAN:

- Truy cập trái phép vào mạng LAN vật lý.
- Truy cập trái phép vào hệ thống, ứng dụng dữ liệu.
- Các lỗ hổng an ninh trong hệ điều hành máy chủ.
- Các lỗ hổng an ninh trong các pm ứng dụng máy chủ.
- Nguy cơ từ người dùng giả mạo trong mạng WLAN.
- Tính bí mật dữ liệu trong mạng WLAN có thể bị đe dọa.
- Các hướng dẫn và chuẩn cấu hình cho máy chủ LAN chưa được tuân thủ.

Các đe dọa với vùng LAN-to-WAN:

- Thăm dò và rà quét trái phép các cổng dịch vụ.
- Truy nhập trái phép.
- Lỗ hổng an ninh trong các bộ định tuyến, tường lửa và các thiết bị mạng khác.
- Người dùng cục bộ (trong LAN) có thể tải các file không xác định nội dung từ các nguồn không xác định.

Các đe dọa với vùng WAN:

- Rủi ro từ việc dữ liệu có thể được truy nhập trong môi trường công cộng và mở.
- Hầu hết dữ liệu được truyền dưới dạng rõ.
- Dễ bị nghe trộm.
- Dễ bị tấn công phá hoại.
- Kẻ tấn công có thể tự do, dễ dàng gửi email có đính kèm Virus, sâu và các phần mềm độc hại.

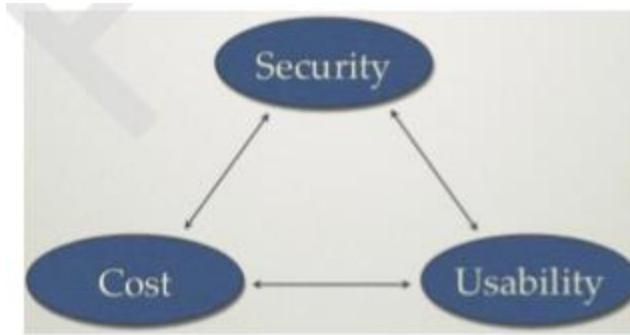
Đe dọa với vùng truy cập từ xa:

- Tấn công kiểu vét cạn vào tên người dùng và mật khẩu.
- Tấn công vào hệ thống đăng nhập và điều khiển truy cập.
- Truy cập trái phép vào hệ thống CNTT, ứng dụng và dữ liệu.
- Thông tin bí mật có thể bị đánh cắp từ xa.
- Rò rỉ dữ liệu do vi phạm các tiêu chuẩn phân loại dữ liệu.

Các đe dọa với vùng hệ thống/ứng dụng:

- Truy cập trái phép đến trung tâm dữ liệu, phòng máy hoặc tủ cáp.
- Khó khăn trong quản lý các máy chủ yêu cầu tính sẵn sàng cao.
- Lỗ hổng trong quản lý các phần mềm ứng dụng của hệ điều hành máy chủ.
- Các vấn đề an ninh trong các môi trường ảo của điện toán đám mây.
- Vấn đề hỏng hóc hoặc mất dữ liệu.

4. MÔ HÌNH TỔNG QUÁT ĐẢM BẢO ATHTTT



Các lớp bảo vệ cần cân bằng giữa Tính hữu dụng (Usability), Chi phí (Cost) và An toàn (Security)

Nguyên tắc đảm bảo an toàn thông tin, hệ thống và mạng:

- Phòng vệ nhiều lớp có chiều sâu: tạo ra nhiều lớp bảo vệ, kết hợp tính năng tác dụng của mỗi lớp để đảm bảo an toàn tối đa cho thông tin, hệ thống và mạng.
- Một lớp, một công cụ phòng vệ thường không đảm bảo an toàn.
- Không tồn tại HTTT an toàn tuyệt đối.
 - Thường HTTT an toàn tuyệt đối là hệ thống kín và không hoặc ít có giá trị sử dụng.
 - Cần cân bằng giữa an toàn, giá trị sử dụng và chi phí.

Các lớp phòng vệ điển hình:

- Lớp an ninh cơ quan/tổ chức.
 - Lớp bảo vệ vật lý
 - Lớp chính sách và thủ tục đảm bảo ATTT.
- Lớp an ninh mạng:
 - Lớp an ninh cho từng thành phần mạng.
 - Tường lửa, mạng riêng ảo.
- Lớp an ninh hệ thống
 - Lớp tăng cường an ninh hệ thống.
 - Lớp quản trị tài khoản và phân quyền người dùng.
 - Lớp quản lý các bản vá và cập nhật phần mềm.
 - Lớp phát hiện và ngăn chặn phần mềm độc hại.

Chương 2. CÁC DẠNG TẤN CÔNG VÀ CÁC PHẦN MỀM ĐỘC HẠI

- Khái quát về mối đe dọa, điểm yếu và tấn công
 - Các công cụ hỗ trợ tấn công.
 - Các dạng tấn công phá hoại
 - Các dạng phần mềm độc hại

1. Khái quát về mối đe dọa, lỗ hổng và tấn công

- Mối đe dọa: mối đe dọa một hành động nào đó có thể gây hư hại đến các tài nguyên hệ thống (phần cứng, phần mềm, CSDL, các File, dữ liệu,...)
- Lỗ hổng: là bất kỳ điểm yếu nào trong hệ thống cho phép mối đe dọa có thể gây tác hại.
- Quan hệ giữa mối đe dọa và lỗ hổng:
 - Cá mối đe dọa thường khai thác 1 hoặc 1 số lỗ hổng đã biết để thực hiện các cuộc tấn công phá hoại.
 - Nếu tồn tại một lỗ hổng trong hệ thống, sẽ có khả năng một mối đe dọa trở thành hiện thực.
 - Không thể triệt tiêu được hết các mối đe dọa, nhưng có thể giảm thiểu các lỗ hổng qua đó giảm thiểu khả năng bị tận dụng để tấn công.

Mối đe dọa, lỗ hổng và tấn công

Các mối đe dọa thường gặp:

- Phần mềm độc hại.
 - Hư hỏng phần cứng hoặc phần mềm.
 - Kẻ tấn công ở bên trong.
 - Mất trộm các thiết bị.
 - Kẻ tấn công ở bên ngoài.
 - Tai họa thiên nhiên.
 - Gián điệp công nghiệp.
 - Khủng bố phá hoại.
- Không phải các mối đe dọa đều độc hại, một số là do cô ý, một số có thể là do ngẫu nhiên/vô tình.

Mối đe dọa, lỗ hổng và tấn công

Các lỗ hổng tồn tại trong cả 7 vùng của nền tảng CNTT:

- Người dùng.
- Máy trạm.
- Mạng LAN.
- Mạng LAN-to-WAN.
- Mạng WAN.
- Truy cập từ xa.
- Hệ thống/ứng dụng.

Mối đe dọa, lỗ hổng và tấn công

Các lỗ hổng tồn tại trong HĐH và các phần mềm ứng dụng:

- Lỗi tràn bộ đệm.
- Không kiểm tra đầu vào.
- Các vấn đề với điều khiển truy cập.
- Các điểm yếu trong xác thực, trao quyền.
- Các điểm yếu trong các hệ mật mã.

Mối đe dọa, lỗ hổng và tấn công

- Tấn công độc hại: một cuộc tấn công vào hệ thống máy tính hoặc các tài nguyên mạng được thực hiện bằng cách khai thác các lỗ hổng trong hệ thống.
- Các loại tấn công: 4 loại chính
 - Giả mạo: giả mạo thông tin người dùng để đánh lừa người dùng thông thường.
 - Chặn bắt: liên quan đến việc nghe trộm trên đường truyền và chuyển hướng thông tin để sử dụng trái phép.
 - Gây ngắt quãng: gây ngắt kênh truyền thông ngăn cản việc truyền dữ liệu.
 - Sửa đổi: liên quan đến việc sửa đổi thông tin trên đường truyền hoặc sửa đổi dữ liệu File.

Mối đe dọa, lỗ hổng và tấn công

Hai kiểu tấn công:

- Tấn công chủ động:
 - Sửa đổi dữ liệu tấn công trên đường truyền.
 - Sửa đổi dữ liệu trong File.
 - Giành quyền truy cập trái phép vào máy tính hoặc hệ thống mạng.
 - Tấn công chủ động là một đột nhập về mặt vật lý.
- Tấn công thụ động: không gây ra thay đổi trên hệ thống.
 - Nghe trộm.
 - Giám sát lưu lượng trên đường truyền.

Mối đe dọa, lỗ hổng và tấn công

Một số dạng tấn công điển hình:

- Tấn công bằng mã độc.
- Tấn công vào mật khẩu.
- Tấn công từ chối dịch vụ.
- Tấn công giả mạo địa chỉ, nghe trộm.
- Tấn công kiểu phát lại và người đứng giữa.
- Tấn công bằng bom thư và thư rác.
- Tấn công sử dụng cửa hậu
-

2. Các công cụ hỗ trợ tấn công

- Công cụ tấn công là các công cụ phần cứng, phần mềm, hoặc các kỹ thuật hỗ trợ kẻ tấn công tấn công vào các hệ thống máy tính hoặc các tài nguyên mạng.
- Một số công cụ và kỹ thuật hỗ trợ tấn công:
 - Công cụ quét lỗ hổng.
 - Công cụ quét cổng dịch vụ.
 - Công cụ nghe lén.
 - Công cụ ghi phím gõ.

Công cụ quét lỗ hổng

Một số công cụ quét lỗ hổng:

- Thu thập thông tin về các điểm yếu/lỗ hổng đã biết của hệ thống máy tính hoặc mạng.
- Gửi những thông điệp được tạo đặc biệt để kiểm tra điểm yếu/lỗ hổng đến hệ thống máy tính cần rà quét. Nếu hệ thống có phản hồi → điểm yếu vẫn tồn tại.
- Kẻ tấn công sử dụng kết quả rà quét điểm yếu/lỗ hổng để quyết định dạng tấn công có khả năng thành công cao nhất.

Một số công cụ quét lỗ hổng:

- Nmap Network Scanner
- Wireshark
- SecurityBox
- PuTTY
- SQLMap

Công cụ quét lỗ hổng

Công cụ quét cổng dịch vụ:

- Các cổng TCP/IP, UDP nằm trong khoảng từ 0 – 65535
 - Các cổng 0-1024 là cổng chuẩn.
 - Cổng lớn hơn 1024 là các cổng tùy gán.
- Kẻ tấn công thường sử dụng công cụ quét cổng để nhận dạng các điểm yếu trong hệ thống.
- Công cụ quét cổng kết nối đến máy tính để xác định cổng nào được mở và có thể truy nhập vào máy tính. Từ đó xác định được dịch vụ/ứng dụng nào đang chạy trên hệ thống:
 - Cổng 80/443 mở → dịch vụ web đang chạy.
 - Cổng 25 mở → dịch vụ email SMTP đang chạy.
 - Cổng 1433 mở → Máy chủ CSDL MS SQL Server đang chạy.
 - Cổng 53 mở → dịch vụ DNS đang chạy,...

Công cụ quét lỗ hổng

Nguyên tắc tối thiểu các cổng được mở:

- Đóng tất cả các cổng không sử dụng.
- Chỉ mở những cổng có dịch vụ cần thiết cho người dùng.

Một số công cụ quét cổng:

- Nmap
- Portsweep
- Advanced Port Scanner
- Angry IP Scanner

Công cụ nghe trộm (Sniffers):

- Công cụ nghe trộm cho phép bắt các gói tin khi chúng được truyền trên mạng.
- Công cụ nghe trộm có thể là module phần cứng, phần mềm hoặc kết hợp.
- Các thông tin nhạy cảm như mật khẩu nếu không được mã hóa thì có thể bị kẻ tấn công nghe trộm khi được truyền từ máy trạm đến máy chủ.

Một số công cụ cho phép bắt gói tin tuyề

- Tcpdump
- Pcap / Wincap (packet capture)
- IP Tools (<http://www.softpedia.com>)
- Wireshark

Công cụ ghi phím gõ (Keyloggers):

- Công cụ ghi phím gõ là một dạng công cụ giám sát có thể bằng phần cứng hoặc phần mềm có khả năng ghi lại mọi phím người dùng gõ và lưu vào một file;
- Sau đó file đã ghi có thể được gửi cho kẻ tấn công theo địa chỉ định trước hoặc sao chép trực tiếp.
- Người quản lý có thể cài đặt Keyloggers vào máy tính của nhân viên để theo dõi hoạt động của nhân viên.

Cài đặt Keyloggers:

- Bằng phần cứng: thường được cài như 1 khớp nối kéo dài giữa máy tính và dây bàn phím.
- Bằng phần mềm: kẻ tấn công có thể tích hợp công cụ Keyloggers vào một phần mềm thông thường và lừa người dùng cài đặt vào máy tính.

3. Các dạng tấn công phá hoại thường gặp:

- Tấn công vào mật khẩu.
- Tấn công bằng mã độc.
- Tấn công từ chối dịch vụ.
- Tấn công giả mạo địa chỉ.
- Tấn công nghe trộm.
- Tấn công kiểu người đứng giữa.
- Tấn công bằng bom thư và thư rác.
- Tấn công sử dụng cửa hậu.
- Tấn công kiểu Social Engineering
- Tấn công phishing, pharming

Tấn công vào mật khẩu

Tấn công vào mật khẩu: là dạng tấn công nhằm đánh cắp mật khẩu và thông tin tài khoản để lạm dụng.

Nếu kẻ tấn công có tên người dùng và mật khẩu → có thể đăng nhập vào tài khoản và thực hiện các thao tác như người dùng bình thường.

Các dạng tấn công vào mật khẩu:

- Tấn công dựa trên từ điển: người dùng có xu hướng chọn mật khẩu là các từ đơn giản có trong từ điển cho dễ nhớ.
- Tấn công vét cạn: sử dụng tổ hợp các ký tự và thủ tự động. Phương pháp này thường sử dụng với các mật khẩu đã được mã hóa.

Tấn công vào mật khẩu

Phòng chống:

- Chọn mật khẩu đủ mạnh: độ dài ≥ 8 ký tự gồm chữ cái hoa, thường, chữ số và ký tự đặc biệt (?#\$...)
- Không nên đặt MK là: ngày sinh, số cmnd,...
- Thay đổi mật khẩu thường xuyên.

Một số công cụ khôi phục mật khẩu:

- Password Cracker (<http://www.softpedia.com>)
- Ophcrack
- Offline NT Password & Registry Editor
- PC Login Now

Tấn công bằng mã độc

Tấn công bằng mã độc gồm 1 số dạng:

- Lợi dụng các lỗ hổng về lập trình, lỗ hổng cấu hình hệ thống để chèn và thực hiện mã độc trên hệ thống.
 - Tấn công lợi dụng lỗi tràn bộ đệm.
 - Tấn công lợi dụng lỗi không kiểm tra đầu vào: tấn công chèn mã SQL, tấn công Script kiểu XSS, CSRF.
- Lừa người sử dụng tải, cài đặt và thực hiện các phần mềm độc hại: Virus, Trojan,....

Tràn bộ đệm

- Lỗi tràn bộ đệm xảy ra khi một ứng dụng cố gắng ghi dữ liệu vượt khỏi phạm vi bộ đệm.
- Lỗi tràn bộ đệm có thể khiến ứng dụng ngừng hoạt động, gây mất dữ liệu hoặc thậm chí giúp kẻ tấn công kiểm soát hệ thống.
- Lỗi tràn bộ đệm chiếm 1 tỉ lệ lớn cho số các lỗi gây lỗ hổng bảo mật.
- Không phải tất cả các lỗi tràn bộ đệm có thể bị khai thác bởi kẻ tấn công.

Tràn bộ đệm

Các vùng nhớ chứa bộ đệm của các ứng dụng:

- Ngăn sếp (Stack): vùng nhớ lưu các tham số gọi hàm, phương thức và dữ liệu cục bộ của chúng: các biến cục bộ được cấp phát tĩnh.
- Vùng nhớ heap: là vùng nhớ chung lưu dữ liệu cho ứng dụng: bộ nhớ heap thường được cấp phát động theo yêu cầu.

Tràn bộ đệm

Các biện pháp phòng chống lỗi tràn bộ đệm:

- Kiểm tra mã nguồn bằng tay để tìm và vá các điểm có khả năng xảy ra lỗi tràn bộ đệm.
- Sử dụng các công cụ phân tích mã tự động tìm các điểm có khả năng xảy ra lỗi tràn bộ đệm.
- Đặt cơ chế không cho thực hiện mã trong Stack;
- Sử dụng các cơ chế bảo vệ Stack: thêm số ngẫu nhiên trước địa chỉ trả về, kiểm tra số ngẫu nhiên này trước khi trả về chương trình gọi để xác định khả năng bị thay đổi địa chỉ trả về.

Tràn bộ đệm

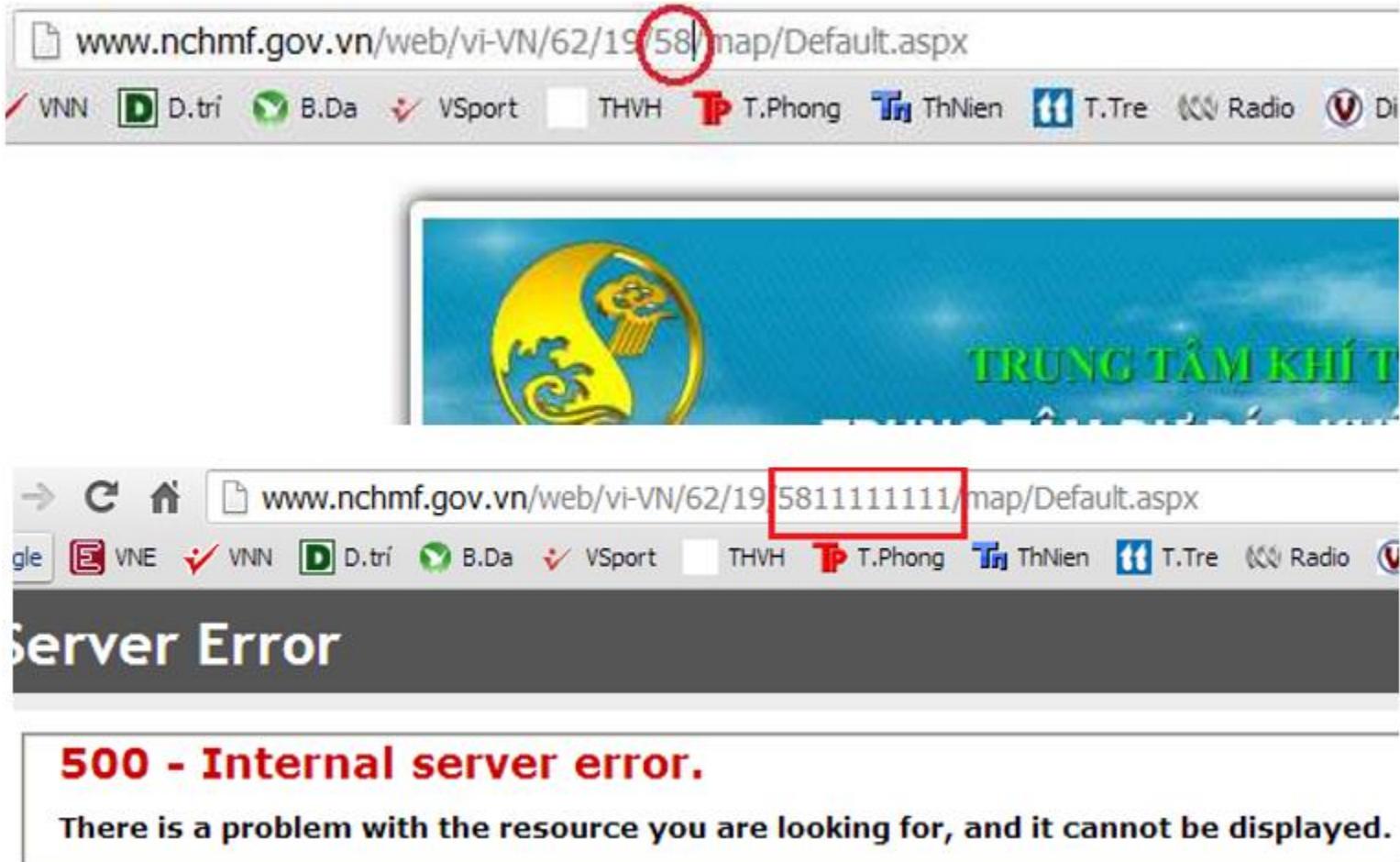
Các dữ liệu đầu vào cần được kiểm tra để đảm bảo đạt y/c về định dạng và kích thước

- Các dạng dữ liệu nhập điền hình cần kiểm tra.
- Các trường dữ liệu Text.
- Các lệnh được truyền qua URL để kích hoạt chương trình.
- Các File âm thanh, hình ảnh, hoặc đồ họa do người dùng hoặc các tiến trình khác cung cấp.
- Các đối số đầu vào trong dòng lệnh.
- Các dữ liệu từ mạng hoặc các nguồn không tin cậy.

→ Kẻ tấn công có thể kiểm tra các dữ liệu đầu vào và thử tất cả các khả năng để khai thác.

Một số dạng tấn công lợi dụng lỗ không kiểm tra đầu vào:

- Cố tình nhập dữ liệu quá lớn hoặc sai định dạng gây lỗi cho ứng dụng → có thể làm ứng dụng ngừng hoạt động.
- Chèn mã độc SQL để thực hiện (SQL Injection).



Chèn mã độc

- SQL Injection (chèn mã độc SQL) là một kỹ thuật cho phép kẻ tấn công chèn mã SQL vào dữ liệu gửi đến máy chủ và được thực hiện trên máy chủ CSDL.
- Tùy mức độ tinh vi, SQL Injection có thể cho phép kẻ tấn công:
 - Vượt qua các khâu xác thực người dùng.
 - Đánh cắp thông tin trong CSDL.
 - Chèn, xóa, sửa đổi dữ liệu.
 - Chiếm quyền điều khiển hệ thống.
- Nguyên nhân: do dữ liệu đầu vào từ người dùng hoặc từ các nguồn khác không được kiểm tra hoặc kiểm tra không kỹ.

- public static DataTable ThongNguoidung(string ten, string mk)
 - {
 - string sql = "Select * From NGUOIDUNG
 where TenDangNhap=" + ten + " and MatKhau="
 + mk + "";
- DataTable dt = new DataTable();
dt = KetNoiCSDL.ExcuteQuery(sql);
return dt;
}

VD: form đăng nhập:

```
• private void btDangNhap_Click(object sender, EventArgs e)
  • {
    •     DataTable dt = new DataTable();
    •     dt = ThongNguoidung(txtTenDN.Text, txtMatKhau.Text);
    •     if (dt.Rows.Count == 0)
    •     {
    •         MessageBox.Show("Tên đăng nhập và mật khẩu chưa đúng!", "Thông báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);
    •     }
    •     else
    •     {
    •         MessageBox.Show("Đăng nhập thành công!", "Thông báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);
    •         frm_Main fr = new frm_Main();
    •         fr.Show();
    •     }
  }
```

- Nếu người dùng nhập username: **admin**, passwd: **admin** thì mã code hoạt động đúng.
- Nếu người dùng nhập username: **aaaaa' OR 1=1--** , passwd: **123abc** (bất kỳ) thì from vẫn đăng nhập mặc dù tài khoản này không có trong CSDL.

→ *select * from NGUOIDUNG*

where [TenDangNhap]='aaaaa' OR 1=1--' and MatKhau='123abc'

Vượt qua các khâu xác thực người dùng:

- Phòng chống/sửa chữa:
 - Kiểm soát kích thước và định dạng của dữ liệu đầu vào, lọc bỏ các ký tự đặc biệt, các từ khóa SQL.
 - Tránh sử dụng câu truy vấn trực tiếp nên dùng:
 - Stored Procedure là dạng các câu lệnh SQL dưới dạng các thủ tục và được lưu trong CSDL
 - Sử dụng các cơ chế truyền tham số, tạo câu truy vấn của ngôn ngữ.

Chèn mã độc

- Nếu người dùng nhập vào:

abc';Delete From NGUOIDUNG;--

Khi đó chuỗi sql:

Select * From NGUOIDUNG where

TenDangNhap='abc';**Delete From**
NGUOIDUNG;--' and MatKhau='";

→ Dữ liệu Table NGUOIDUNG bị xóa

- Nếu người dùng nhập vào:
*abc'; Insert into NGUOIDUNG
Values('2','linh','linh');--*
→ Select * From NGUOIDUNG
where TenDangNhap='abc';*Insert into
NGUOIDUNG Values('2','linh','linh');--'*
and MatKhau=";

Kẻ tấn công có thể thay lệnh Delete bằng Insert, Update, Drop để thêm, sửa hoặc xóa cả bảng dữ liệu.

Chèn mã độc

Các biện pháp phòng chống:

- Kiểm tra tất cả các dữ liệu đầu vào, đặc biệt dữ liệu nhập từ người dùng và từ các nguồn không tin cậy.
- Kiểm tra định dạng và kích thước dữ liệu đầu vào.
- Kiểm tra sự hợp lý của nội dung dữ liệu.
- Tạo các bộ lọc để lọc bỏ các ký tự đặc biệt và các từ khóa của các ngôn ngữ trong các trường hợp cần thiết mà kẻ tấn công có thể sử dụng: các ký tự đặc biệt: *, ', =, --,...; các từ khóa: **SELECT, INSERT, UPDATE, DELETE, DROP**,....
- Procedure
- Function

Tấn công từ chối dịch vụ

- Tấn công từ chối dịch vụ: (DoS - Denial of Service Attacks): là dạng tấn công cản trở người dùng hợp pháp truy cập các tài nguyên hệ thống.
- Hai loại tấn công DoS:
 - Tấn công Logic: tấn công dựa vào các lỗi phần mềm làm dịch vụ ngừng hoạt động hoặc làm giảm hiệu năng hệ thống. → cần cài đặt bản cập nhật thường xuyên để phòng chống.
 - Tấn công gây ngập lụt: (SYN floods, Smurf) kẻ tấn công gửi một lượng lớn yêu cầu gây cạn kiệt tài nguyên hệ thống hoặc băng thông đường truyền mạng.

Tấn công DoS - SYN floods

Tấn công DoS - SYN floods là kỹ thuật gây ngập lụt các gói tin TCP.

- SYN là bít điều khiển của TCP dùng để đồng bộ số trình tự gói.

Kịch bản tấn công SYN floods:

- Kẻ tấn công gửi 1 lượng lớn gói tin y/c mở kết nối (SYN-REQ) đến máy tính nạn nhân.
- Máy tính nạn nhân ghi nhận y/c kết nối và dành 1 chỗ trong bảng lưu kết nối trong bộ nhớ cho mỗi y/c kết nối.
- Máy tính nạn nhân sau đó gửi gói tin xác nhận kết nối đến kẻ tấn công.
- Do kẻ tấn công không bao giờ trả lời xác nhận kết nối nên máy tính nạn nhân vẫn phải lưu tất cả các yêu cầu kết nối chưa được xác nhận trong bảng kết nối → bảng kết nối đầy và người dùng hợp pháp không thể truy cập.
- Máy tính nạn nhân chỉ có thể xóa yêu cầu kết nối khi nó timed-out.

Tấn công DoS - SYN floods

- Kẻ tấn công thường dùng địa chỉ IP giả mạo hoặc địa chỉ không có thực làm Source IP trong gói tin IP nên thông điệp gửi gói tin xác nhận đến kết nối của máy tính nạn nhân không bao giờ đến đích.
- Kẻ tấn công cố tình tạo một lượng rất lớn yêu cầu mở kết nối dở dang để:
 - Để các y/c mở kết nối điền đầy đủ bảng kết nối → máy tính nạn nhân không thể chấp nhận y/c kết nối của những người dùng khác.
 - Làm cạn kiệt tài nguyên bộ nhớ của máy tính nạn nhân → có thể làm máy nạn nhân ngừng hoạt động.
 - Gây nghẽn đường truyền mạng.

Phòng chống

- Sử dụng kỹ thuật lọc: cần sửa đổi giao thức TCP không cho phép kẻ tấn công giả mạo địa chỉ.
- Tăng kích thước Backlog: tăng kích thước bảng Baklog lưu các yêu cầu → tăng khả năng phục vụ yêu cầu.
- Giảm thời gian chờ: các kết nối chưa được xác nhận sẽ bị xóa khi hết thời gian chờ.
- Yêu cầu kết nối chỉ được cấp phát không gian nhớ đầy đủ khi nó được xác nhận.
- Sử dụng Firewalls và Proxies: Có khả năng nhận dạng các địa chỉ IP nguồn là địa chỉ không có thực; Có khả năng tiếp nhận kết nối, chờ đến khi có xác nhận mới chuyển lại cho máy chủ đích.

Tấn công DoS Smurf

- Tấn công DoS Smurf: sử dụng kiểu phát quảng bá có định hướng để gây ngập lụt đường truyền mạng của máy tính nạn nhân.
- Kịch bản: kẻ tấn công gửi 1 lượng lớn gói tin ICMP (Ping) với địa chỉ IP nguồn là địa chỉ của máy nạn nhân đến một mạng sử dụng một địa chỉ quảng bá (IP Broadcast address);
 - Các máy tính trong mạng nhận được thông điệp ICMP sẽ gửi trả lời đến máy tính có địa chỉ nguồn IP (máy nạn nhân), nếu lượng máy trong mạng quá lớn → máy tính nạn nhân sẽ bị ngập lụt đường truyền.

Phòng chống

- Cấu hình các máy và Router không trả lời các yêu cầu ICMP hoặc các yêu cầu phát quảng bá.
- Cấu hình các Router không chuyển tiếp yêu cầu gửi đến các địa chỉ quảng bá.

Tấn công DDoS

- Tấn công DDoS (Distributed Denial of Service Attacks) là một loại tấn công DoS:
 - Liên quan đến gây ngập lụt các máy nạn nhân với một lượng rất lớn các yêu cầu kết nối giả mạo.
 - DDoS khác DoS ở phạm vi tấn công.
- Kịch bản tấn công DDoS:
 - Kẻ tấn công chiếm quyền điều khiển hàng trăm thậm chí hàng nghìn máy tính trên mạng Internet, sau đó cài các chương trình tấn công tự động lên các máy tính này.
 - Sau đó, kẻ tấn công ra lệnh cho các chương trình tấn công tự động đồng loạt tạo ra các yêu cầu giả mạo gửi đến các máy nạn nhân.
 - Lượng yêu cầu giả mạo có thể rất lớn và đến từ nhiều nguồn khác nhau nên rất khó đối phó và lần vết để tìm ra kẻ tấn công.

Tấn công Reflective DDoS

Tấn công Reflective DDoS là một loại tấn công DDoS với một số điểm khác biệt:

- Các máy tính do kẻ tấn công điều khiển không trực tiếp tấn công máy nạn nhân.
- Gửi một lượng lớn các yêu cầu giả mạo với địa chỉ nguồn là địa chỉ máy nạn nhân đến một số lượng lớn các máy khác trên mạng Internet.
- Các máy tính này gửi Reply đến máy nạn nhân do địa chỉ nguồn của máy nạn nhân được đặt vào yêu cầu giả mạo.
- Nếu số lượng máy tính được gửi y/c có số lượng lớn, số Reply sẽ rất lớn và gây ngập lụt máy tính nạn nhân.

Tấn công Reflective DDoS khó lẩn vết và phòng chống hơn tấn công DDoS thông thường do có thể qua nhiều cấp

Tấn công giả mạo địa chỉ IP (IP Spoofing)

- Là kiểu tấn công trong đó kẻ tấn công sử dụng địa chỉ IP giả, thường để đánh lừa máy nạn nhân để vượt qua các hàng rào kiểm soát an ninh.
- Nếu kẻ tấn công giả địa chỉ IP là địa chỉ cục bộ của mạng LAN, có thể có nhiều cơ hội đột nhập vào các máy khác trong LAN do chính sách kiểm soát an ninh với các máy tính trong mạng LAN thường được giảm nhẹ.
- Nếu Router hoặc Firewall của mạng không được cấu hình để nhận ra IP giả mạo của mạng LAN nội bộ → kẻ tấn công có thể thực hiện.

Tấn công nghe trộm

- Tấn công nghe trộm: là dạng tấn công sử dụng thiết bị phần cứng hoặc phần mềm, lắng nghe trên card mạng, hub hoặc Router để bắt các gói tin dùng cho phân tích về sau.

Tấn công người đứng giữa

- Lợi dụng quá trình chuyển gói tin đi qua nhiều trạm thuộc các mạng khác nhau.
- Kẻ tấn công chặn bắt các thông điệp giữa 2 bên tham gia truyền thông và chuyển thông điệp lại cho bên kia.
- Thường được sử dụng để đánh cắp thông tin.

Tấn công bằng bom thư và thư rác

- Tấn công bằng bom thư (Mail bombing) là dạng tấn công DoS khi kẻ tấn công chuyển một lượng lớn mail đến nạn nhân.
 - Có thể thực hiện được bằng kỹ thuật Social Engineering.
 - Hoặc khai thác lỗi trong hệ thống gửi nhận email SMTP.
 - Kẻ tấn công có thể lợi dụng các máy chủ email không được cấu hình tốt để gửi email.
- Tấn công bằng thư rác (Spam emails)
 - Spams là email không mong muốn, thường là các email quảng cáo.
 - Spams gây lãng phí tài nguyên tính toán và thời gian của người dùng (phải lọc, xóa).
 - Spams cũng có thể dùng để chuyển các phần mềm độc hại.

Tấn công sử dụng cửa hậu

Tấn công sử dụng cửa hậu (Back doors hoặc Trap doors)

- Cổng hậu thường được các lập trình viên tạo ra, dùng để gõ rối và test chương trình.
- Cổng hậu thường cho phép truy nhập trực tiếp vào hệ thống mà không qua các thủ tục kiểm tra an ninh thông thường.
- Khi cổng hậu được lập trình viên tạo ra để truy nhập hệ thống bất hợp pháp, nó trở thành một mối đe doạ đe dọa an ninh hệ thống.
- Rất khó phát hiện ra cổng hậu vì nó thường được thiết kế và cài đặt khéo léo: cổng hậu chỉ được kích hoạt trong một ngũ cảnh nào đó.

Tấn công kiểu Social Engineering

Tấn công kiểu Social Engineering là dạng tấn công sử dụng các kỹ thuật xã hội đã thuyết phục người dùng tiết lộ thông tin truy nhập hoặc các thông tin có giá trị cho kẻ tấn công.

- Kẻ tấn công có thể giả danh làm người có vị trí cao hơn so với nạn nhân để có được sự tin tưởng.
- Kẻ tấn công có thể mạo nhận là người được ủy quyền của người có thẩm quyền để yêu cầu các nhân viên tiết lộ thông tin về cá nhân/tổ chức.
- Kẻ tấn công có thể lập trang web giả để đánh lừa người dùng cung cấp các thông tin cá nhân và thông tin tài khoản, thẻ tín dụng,...

Trò lừa đảo Nigeria 4-1-9

Trò lừa đảo Nigeria 4-1-9: lợi dụng sự ngây thơ và lòng tham của nhiều người.

- Kẻ lừa đảo gửi thư tay hoặc email đến nhiều người nhận, mô tả về việc có 1 khoản tiền lớn (thùa kẽ, lợi tức,...) cần chuyển ra nước ngoài, nhờ người nhận giúp đỡ để hoàn thành giao dịch. Khoản tiền có thể lên đến hàng chục hoặc trăm triệu USD. Kẻ tấn công hứa sẽ trả cho người tham gia một phần số tiền (20-30%).
- Nếu người nhận có phản hồi và đồng ý tham gia, kẻ tấn công sẽ gửi tiếp thư/email khác, yêu cầu chuyển cho hắn 1 khoản phí giao dịch (từ vài ngàn đến hàng chục ngàn USD).
- Nếu người nhận gửi tiền cho kẻ tấn công → người đó mất tiền, do giao dịch mà kẻ tấn công hứa là giả mạo.

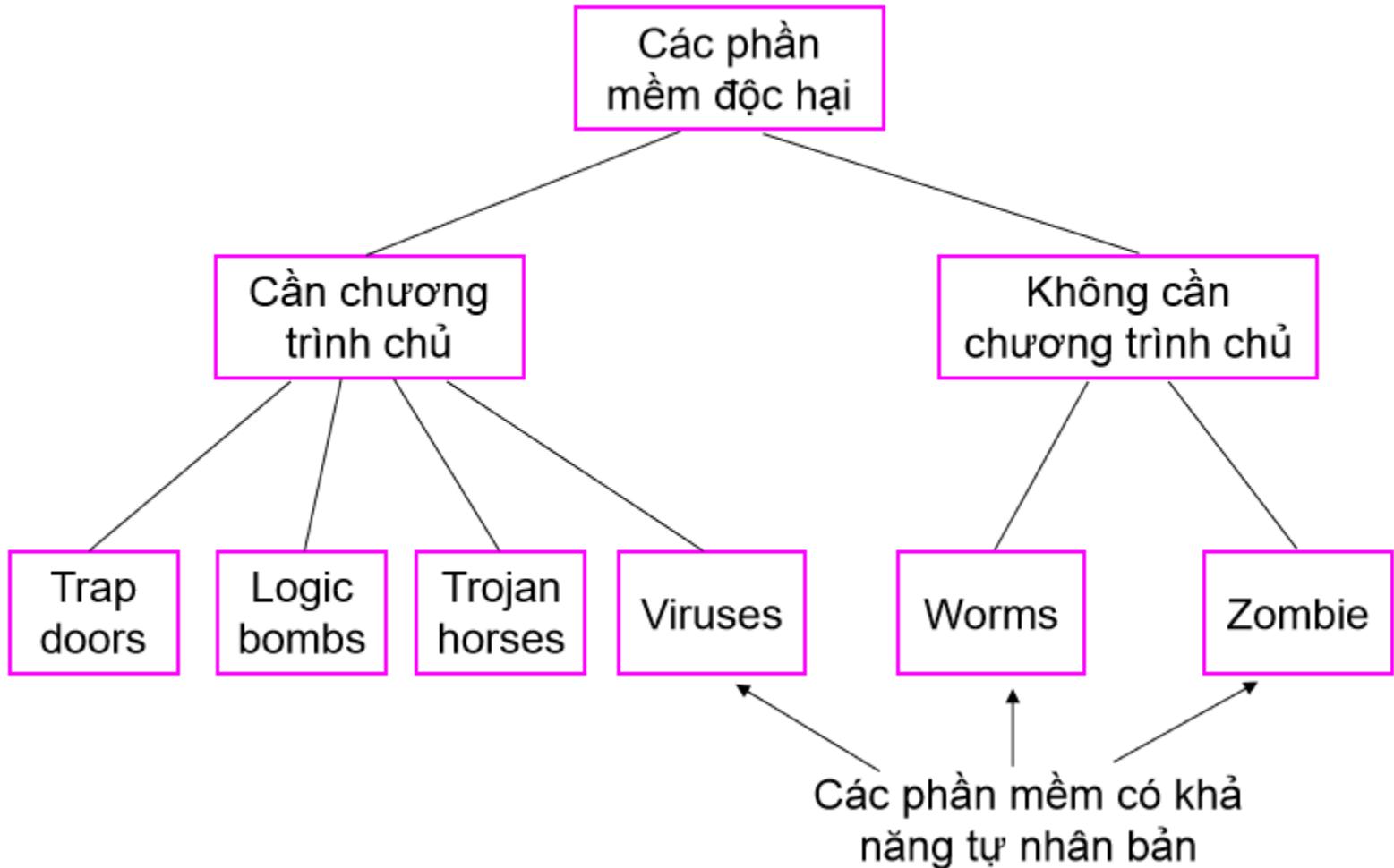
Phishing là một dạng của tấn công Social Engineering, lừa người dùng để lấy thông tin cá nhân, thông tin tài khoản, thẻ tín dụng,...

- Kẻ tấn công có thể giả mạo trang web của các tổ chức tài chính, ngân hàng.
- Chúng gửi email cho người dùng (địa chỉ email thu thập trên mạng), yêu cầu xác thực thông tin.
- Nếu người dùng làm theo hướng dẫn cung cấp các thông tin cá nhân, thông tin tài khoản, thẻ tín dụng cho kẻ tấn công.

Pharming là kiểu tấn công vào trình duyệt người dùng:

- Người dùng gõ địa chỉ 1 website, trình duyệt lại yêu cầu 1 website khác (độc hại).
- Kẻ tấn công thường sử dụng sâu, virus hoặc các phần mềm độc hại cài vào hệ thống để điều khiển trình duyệt của người dùng.
- Kẻ tấn công cũng có thể tấn công vào hệ thống DNS để thay đổi kết quả truy vấn: thay địa chỉ IP của website hợp pháp thành IP của website độc hại.

Các dạng phần mềm độc hại



Bom logic (Logic bombs)

- Bom logic (Logic bombs) thường được “nhúng” vào các chương trình bình thường và thường hẹn giờ để “phát nổ” trong một số điều kiện cụ thể.
- Điều kiện để bom “phát nổ” có thể là:
 - Sự xuất hiện hoặc biến mất của các files cụ thể;
 - Một ngày nào đó, hoặc một ngày trong tuần.
- Khi “phát nổ” bom logic có thể xoá dữ liệu, files, tắt cả hệ thống...
- Ví dụ: Quả bom logic do Tim Lloyd cài lại đã “phát nổ” tại công ty Omega Engieering vào ngày 30/7/1996, 20 ngày sau khi Tim Lloyd bị sa thải. Bom logic này đã xoá sạch các bản thiết kế và các chương trình, gây thiệt hại 10 triệu USD. Tim Lloyd bị phạt 2 triệu USD và 41 tháng tù.

Trojan horses

- Trojan horses chứa mã độc, thường giả danh những chương trình có ích, nhằm lừa người dùng kích hoạt chúng.
- Trojan horses thường được sử dụng để thực thi gián tiếp các tác vụ, mà tác giả của chúng không thể thực hiện trực tiếp do không có quyền truy nhập.
- VD: trong một hệ thống nhiều users, một user có thể tạo ra một trojan đội lốt một chương trình hữu ích đặt ở thư mục chung. Khi trojan này được thực thi bởi một user khác, nó sẽ cho phép tất cả các users truy nhập vào các files của user đó.

Zombie

- Zombie là một chương trình được thiết kế để giành quyền kiểm soát một máy tính có kết nối Internet, và sử dụng máy tính bị kiểm soát để tấn công các hệ thống khác.
- Các zombies thường được dùng để tấn công DDoS các máy chủ/website lớn.
- Rất khó để lẩn vết và phát hiện ra tác giả tạo ra và điều khiển các zombies.

Virus

- Virus và một chương trình có thể “nhiễm” vào các chương trình khác, bằng cách sửa đổi các chương trình này.
- Nếu các chương trình đã bị sửa đổi chứa virus được kích hoạt thì virus sẽ tiếp tục “lây nhiễm” sang các chương trình khác.
- Giống như virus sinh học, virus máy tính cũng có khả năng tự nhân bản, tự lây nhiễm sang các chương trình khác mà nó tiếp xúc.
- Có nhiều con đường lây nhiễm virus: sao chép file, gọi các ứng dụng và dịch vụ qua mạng, email...
- Virus có thể thực hiện được mọi việc mà một chương trình thông thường có thể thực hiện. Khi đã lây nhiễm vào một chương trình, virus tự động được thực hiện khi chương trình này chạy.

4 giai đoạn của vòng đời virus

- Giai đoạn “nằm im”: Virus trong giai đoạn không được kích hoạt. Trong giai đoạn này virus có thể được kích hoạt nhờ một sự kiện nào đó.
- Giai đoạn phát tán: Virus “cài” một bản sao của nó vào các chương trình khác.
- Giai đoạn kích hoạt: virus được kích hoạt để thực thi các tác vụ đã thiết kế định sẵn. Virus cũng thường được kích hoạt dựa trên một sự kiện nào đó.
- Giai đoạn thực hiện: thực thi các tác vụ. Một số viruses có thể vô hại, nhưng một số khác có thể xoá dữ liệu, chương trình...

Cơ chế chèn mã virus vào chương trình chủ:

- Virus có thể chèn mã của nó vào đầu hoặc cuối của chương trình bị lây nhiễm.
- Khi chương trình nhiễm virus được thực hiện, mã virus được thực hiện trước, sau đó mã chương trình mới được thực hiện.

Macro viruses

- Macro viruses thường lây nhiễm vào các files tài liệu của MS-Word và ứng dụng office khác.
- Macro viruses hoạt động được nhờ tính năng cho phép tạo và thực hiện các đoạn mã macro trong các tài liệu của bộ ứng dụng MS Office. Các đoạn mã macro thường được dùng để tự động hóa 1 số việc và được viết bằng ngôn ngữ Visual Basic for Applications.
- Macro viruses thường lây nhiễm vào các files định dạng chuẩn và từ đó lây nhiễm vào tất cả các files tài liệu được mở.
- Macro viruses cũng có thể được tự động kích hoạt nhờ các auto-executed macros: AutoExecute, Automacro và Command macro.
- Theo thống kê, macro viruses chiếm khoảng 2/3 tổng lượng viruses đã được phát hiện.

E-mail viruses

- E-mail viruses lây nhiễm bằng cách tự động gửi một bản copy của nó như 1 file đính kèm đến tất cả các địa chỉ email trong sổ địa chỉ của user trên máy bị lây nhiễm.
- Nếu user mở email hoặc file đính kèm, virus được kích hoạt.
- E-mail viruses có thể lây nhiễm rất nhanh chóng, lan tràn trên khắp thế giới trong một thời gian ngắn.

Sâu (Worms)

- Sâu (Worms) có khả năng tự lây nhiễm từ máy này sang máy khác mà không cần sự trợ giúp của người dùng (khác email viruses).
- Khi sâu lây nhiễm vào một máy, nó sử dụng máy này làm “bàn đạp” để tiếp tục tấn công các máy khác.
- Các sâu trên mạng sử dụng kết nối mạng để lây lan từ máy này sang máy khác.
- Khi sâu hoạt động, nó tương tự virus.

Các phương pháp lây lan của sâu:

- Lây lan qua thư điện tử: sử dụng email để gửi bản copy của sâu đến các máy khác.
- Lây lan thông qua khả năng thực thi từ xa: Sâu thực thi một bản copy của nó trên một máy khác nhờ lợi dụng các lỗ hổng an ninh của hệ điều hành, các dịch vụ hoặc phần mềm ứng dụng.
- Lây lan thông qua khả năng log-in (đăng nhập) từ xa: sâu đăng nhập vào hệ thống ở xa như một user và sử dụng lệnh để copy bản thân từ máy này sang máy khác.

Phòng chống:

- Ngăn chặn viruses lây nhiễm vào hệ thống:
 - Luôn cập nhật hệ thống để hạn chế các lỗi phần mềm.
 - Sử dụng các biện pháp kiểm soát truy nhập.
- Khi hệ thống đã bị nhiễm virus:
 - Phát hiện virus.
 - Nhận dạng virus.
 - Loại bỏ virus.

5. MỘT SỐ KỸ THUẬT VÀ CÔNG CỤ ĐẨM BẢO AN TOÀN HTTT

1. Tường lửa – Giới thiệu

- Tường lửa có thể dùng để bảo vệ hệ thống và mạng cục bộ tránh các đe doạ từ bên ngoài.
- Tường lửa thường được đặt ở vị trí cảng vào của mạng nội bộ công ty hoặc tổ chức
- Tất cả các gói tin từ trong ra và từ ngoài vào đều phải đi qua tường lửa.
- Chỉ các gói tin hợp lệ được phép đi qua tường lửa (xác định bởi chính sách an ninh).
- Bản thân tường lửa phải miễn dịch với các loại tấn công.

Tường lửa – Các loại tường lửa

- Lọc gói tin (Packet-Filtering): Áp dụng một tập các luật cho mỗi gói tin đi/đến để quyết định chuyển tiếp hay loại bỏ gói tin.
- Các cổng ứng dụng (Application-level gateway): Còn gọi là proxy server, thường dùng để phát lại (relay) traffic của mức ứng dụng.
- Cổng chuyển mạch (Circuit-level gateway): Hoạt động tương tự các bộ chuyển mạch.

Tường lửa – Lọc có trạng thái và không trạng thái:

- Tường lửa có trạng thái (Stateful firewall):
 - Có khả năng lưu trạng thái của các kết nối mạng đi qua nó;
 - Nó được lập trình để phân biệt các gói tin thuộc về các kết nối mạng khác nhau;
 - Chỉ những gói tin thuộc một kết nối mạng đang hoạt động mới được đi qua tường lửa, còn các gói tin khác (không thuộc kết nối đang hoạt động) sẽ bị chặn lại.
- Tường lửa không trạng thái (Stateless firewall):
 - Lọc các gói tin riêng lẻ mà không quan tâm đến mỗi gói tin thuộc về kết nối mạng nào;
 - Dễ bị tấn công bởi kỹ thuật giả mạo địa chỉ, giả mạo nội dung gói tin do tường lửa không có khả năng nhớ các gói tin đi trước thuộc cùng một kết nối mạng.

Tường lửa – Kỹ thuật kiểm soát truy nhập:

- Kiểm soát dịch vụ: Xác định dịch vụ nào có thể được truy nhập, hướng đi ra hay đi vào.
- Kiểm soát hướng: Điều khiển hướng được phép đi của các gói tin của mỗi dịch vụ.
- Kiểm soát người dùng:
 - Xác định người dùng nào được quyền truy nhập;
 - Thường áp dụng cho người dùng mạng nội bộ.
- Kiểm soát hành vi: Kiểm soát việc sử dụng các dịch vụ cụ thể. Ví dụ: tường lửa có thể lọc để loại bỏ các thư rác và hạn chế truy nhập đến một bộ phận thông tin của máy chủ web.

Tường lửa – Các hạn chế:

- Không thể chống lại các tấn công không đi qua nó.
- Không thể chống lại các tấn công hướng dữ liệu, hoặc tấn công vào các lỗ hổng an ninh của các phần mềm.
- Không thể chống lại các hiểm họa từ bên trong (mạng nội bộ).
- Không thể ngăn chặn việc vận chuyển các chương trình hoặc các file bị nhiễm virus hoặc các phần mềm độc hại.

2. Các công cụ rà quét và diệt phần mềm độc hại:

Một số phần mềm diệt virus và phần mềm độc hại:

- + Microsoft Security Essentials (Windows 7 trở lên)
- + Semantec Norton Antivirus
- + Kaspersky Antivirus
- + BitDefender Antivirus
- + AVG Antivirus
- + McAfee VirusScan
- + Trend Micro Antivirus
- + F-secure
- + BKAV

3. Các công cụ rà quét lỗ hổng, điểm yếu an ninh:

Công cụ quét lỗ hổng (Vulnerability scanners):

- Thu thập các thông tin về các điểm yếu/lỗ hổng đã biết của hệ thống máy tính hoặc mạng;
- Gửi những thông điệp được tạo đặc biệt để kiểm tra điểm yếu/lỗ hổng đến hệ thống máy tính cần rà quét. Nếu hệ thống có phản hồi, điểm yếu vẫn tồn tại;
- Kẻ tấn công sử dụng kết quả rà quét điểm yếu/lỗ hổng để quyết định dạng tấn công có khả năng thành công cao nhất.

Một số công cụ quét lỗ hổng cho người quản trị:

- Microsoft Baseline Security Analyzer: rà quét các lỗ hổng an ninh trong hệ điều hành Windows và các phần mềm của Microsoft;
- Nessus vulnerability scanner;
- Acunetix Web Vulnerability Scanner.

Chương 3:

ĐIỀU KHIỂN TRUY CẬP VÀ XÁC THỰC NGƯỜI DÙNG

1. Khái niệm

- Điều khiển truy cập là quá trình mà trong đó người dùng được nhận dạng và trao quyền truy cập đến các thông tin, các hệ thống và tài nguyên.
- Một hệ thống điều khiển truy nhập có thể được cấu thành từ 3 dịch vụ:
 - **Xác thực:** là quá trình xác minh tính chân thật của các thông tin nhận dạng mà người dùng cung cấp.
 - **Trao quyền:** là xác định các tài nguyên mà người dùng được phép truy cập sau khi người dùng đã được xác thực.
 - **Quản trị:** cung cấp khả năng thêm, bớt và sửa đổi các thông tin tài khoản người dùng, cũng như quyền truy cập của người dùng.

- Mục đích chính của điều khiển truy cập là để đảm bảo tính bí mật, toàn vẹn và sẵn dùng của thông tin, hệ thống và các tài nguyên:
 - Tính bí mật: đảm bảo chỉ những người có thẩm quyền mới có khả năng truy cập vào dữ liệu và hệ thống.
 - Tính toàn vẹn: đảm bảo dữ liệu không bị sửa đổi bởi các bên không có đủ thẩm quyền.
 - Tính sẵn dùng: đảm bảo tính sẵn sàng của dịch vụ cung cấp cho người dùng thực sự.

2. Điều khiển truy cập tùy quyền DAC

2.1. Giới thiệu:

- Điều khiển truy cập tuỳ quyền – Discretionary Access Control (DAC): là các cơ chế hạn chế truy cập các đối tượng dựa trên thông tin nhận dạng của các chủ thể hoặc nhóm của các chủ thể.
- **Thông tin nhận dạng có thể gồm:**
 - Bạn là ai? (CMND, bằng lái xe, vân tay,...)
 - Những cái bạn biết (tên truy cập, mật khẩu, số pin,...)
 - Bạn có gì? (thẻ ATM, thẻ tín dụng,...)
- DAC cho phép người dùng có thể có thể cấp hoặc ủy quyền truy cập cho các người dùng khác đến các đối tượng thuộc quyền điều khiển của họ.
- Chủ sở hữu của các đối tượng là người có toàn quyền điều khiển các đối tượng này.

2. Điều khiển truy cập tùy quyền DAC

2.1. Giới thiệu:

- Ví dụ:

- Người dùng được cấp 1 thư mục riêng và là chủ sở hữu của thư mục này.
- Người dùng có quyền tạo, sửa đổi và xoá các files trong thư mục của riêng mình (home directory) và cũng có khả năng trao hoặc huỷ quyền truy cập vào các tập tin của mình cho các người dùng khác.

2. Điều khiển truy cập tùy quyền DAC

2.1. Giới thiệu:

Cách thức cơ bản của điều khiển truy cập DAC trong một HCSQL là dựa vào 2 thao tác cơ bản:

- Gán quyền: cho phép người dùng khác được quyền truy cập lên đối tượng do mình làm chủ.
- Thu hồi quyền: thu hồi lại quyền đã gán cho người dùng khác.

2. Điều khiển truy cập tùy quyền DAC

2.1. Giới thiệu:

Các loại quyền trong DAC

- Quyền ở cấp tài khoản/hệ thống (Account/System level): là những quyền này độc lập với các đối tượng trong hệ CSDL, những quyền này do người quản trị hệ thống định nghĩa và gán cho mỗi người dùng.

- Quyền ở cấp đối tượng (Object level): là những quyền trên mỗi đối tượng trong hệ CSDL, người dùng tạo ra đối tượng nào thì sẽ có tất cả các quyền trên đối tượng đó.

2. Điều khiển truy cập tùy quyền DAC

- Quyền ở cấp tài khoản/hệ thống gồm các quyền:
 - Tạo lược đồ CSDL.
 - Tạo bảng dữ liệu/quan hệ (Relation).
 - Tạo View.
 - Chỉnh sửa các Schema/relation.
 - Xóa relation/view.
 - Thêm/xóa/sửa các dòng dữ liệu.
 - Thực thi câu truy vấn thông tin trong CSDL.

2. Điều khiển truy cập tùy quyền DAC

- Quyền ở cấp đối tượng: gồm các đối tượng dữ liệu và các loại truy cập mà người dùng được phép thực hiện trên đối tượng đó.
 - Các đối tượng dữ liệu này gồm: các **Relation** hoặc **View**
 - Các thao tác gồm:
 - + Thêm dữ liệu vào Relation.
 - + Cập nhật/chỉnh sửa dữ liệu trong Relation.
 - + Xóa dữ liệu trong Relation.
 - + Tham khảo đến dữ liệu trong Relation.

2. Điều khiển truy cập tùy quyền DAC

2.2. Mô hình điều khiển truy cập tùy quyền

- Mô hình bảo mật (Security model).
- Mô hình ma trận truy cập (Access matrix model).
- Mô hình Take – Grant (Take – Grant model)

Mô hình bảo mật

Mô hình bảo mật: cung cấp một cách biểu diễn giàu ngữ nghĩa cho các thuộc tính cấu trúc và thuộc tính chức năng của một hệ thống bảo mật.

- Mô hình bảo mật giúp biểu diễn được các đặc tả yêu cầu về bảo mật cho hệ thống.
- Mô hình bảo mật là mô hình ý niệm cấp cao và độc lập với các phần mềm.
- Mô hình bảo mật có thể dùng để chứng minh các tính chất cần có của bảo mật HTTT.

Mô hình ma trận truy cập

- Mô hình được đề nghị bởi Lampson (1971), và được Graham và Denning mở rộng (1972).
- 1976, Harrison và các cộng sự đã phát triển mô hình ma trận truy cập một cách có hệ thống.
- **Access Control Matrix (ACM)** là một công cụ cơ bản để thể hiện trạng thái bảo vệ hệ thống một cách chi tiết và chính xác
- ACM là mô hình bảo mật được dùng cho cả cấp hệ điều hành và cấp cơ sở dữ liệu.

Mô hình ma trận truy cập

- Ma trận điều khiển truy cập ACM là ma trận giữa các chủ thể **S(subject)**, các đối tượng **O(object)** và các quyền tương ứng giữa của chủ thể với đối tượng.

	O_1	\dots	O_i	\dots	O_m
S_1	$A[s_1, o_1]$		$A[s_1, o_i]$		$A[s_1, o_m]$
\dots					
S_i	$A[s_i, o_1]$		$A[s_i, o_i]$		$A[s_i, o_m]$
\dots					
S_n	$A[s_n, o_1]$		$A[s_n, o_i]$		$A[s_n, o_m]$

Trạng thái định quyền (Authorization state)

$$Q = (S, O, A)$$

- **S(Subjects):** là tập các chủ thể - các thực thể chủ động (active entity) sử dụng các nguồn tài nguyên của hệ thống.
- **Ví dụ:** người dùng, nhóm các người dùng (group), quá trình (process), chương trình (programs)

Trạng thái định quyền:

$$Q = (S, O, A)$$

- **O(Objects):** là tập các đối tượng - các thực thể cần được bảo vệ, bao gồm các thực thể bị động (passive object) như tài nguyên hệ thống và các chủ thể
- **Ví dụ:** ở cấp hệ điều hành: file, bộ nhớ, segments, quá trình ở cấp CSDL: CSDL, quan hệ, thuộc tính, hàng, trường dữ liệu của hàng

Trạng thái định quyền

$Q = (S, O, A)$

- **A**(Access matrix): là ma trận truy cập.
 - Hàng: các chủ thẻ
 - Cột: các đối tượng
 - Mỗi ô **A[s,o]** chứa các quyền truy cập mà chủ thẻ **s** được quyền làm trên đối tượng **o**.
 - Các quyền truy cập: thêm, xóa, sửa, đọc, thực thi,...

	O_1	...	O_i	...	O_m
S_1	$A[s_1, o_1]$		$A[s_1, o_i]$		$A[s_1, o_m]$
...					
S_i	$A[s_i, o_1]$		$A[s_i, o_i]$		$A[s_i, o_m]$
...					
S_n	$A[s_n, o_1]$		$A[s_n, o_i]$		$A[s_n, o_m]$

Ví dụ:

S O	File 1	File 2	File 3	Program I
Lan	Read		Read	
Minh		Execute		Read, Write
Tuấn		Read	Execute	
Vương	Execute		Read Write	

Mô hình Take-Grant

- Johns và các cộng sự đề nghị mô hình Take-Grant năm 1976
- Sử dụng các cấu trúc hình học để biểu diễn mối quan hệ về quyền giữa các chủ thể với đối tượng, giữa chủ thể với chủ thể và giữa đối tượng với đối tượng.
- Mục đích chính là làm rõ vấn đề về một chủ thể của hệ thống nhận các quyền truy nhập tới một đối tượng tại trạng thái được mô tả bằng một giản đồ truy nhập
- Có thể được xem là một dạng mở rộng của mô hình ma trận truy cập

Mô hình Take-Grant

- Trạng thái định quyền:

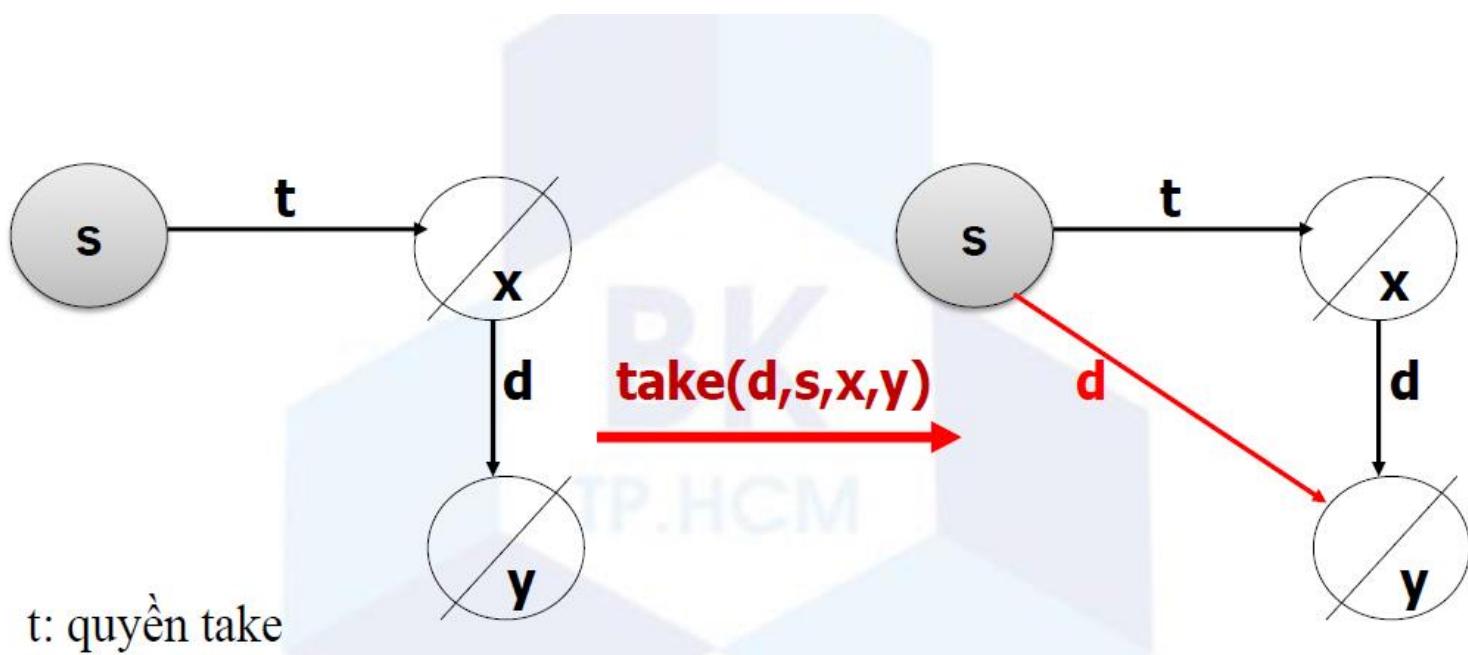
$$G = (S, O, E)$$

- S: tập các chủ thể (người dùng, quá trình, chương trình)
- O: tập các đối tượng bị động (file, bộ nhớ, CSDL, bảng, hàng, trường dữ liệu)
- $V = S \cup O$: tập các đỉnh, $S \cap O = \emptyset$
- E: tập các cung được đánh nhãn

Mô hình Take-Grant

Thao tác Take và Grant

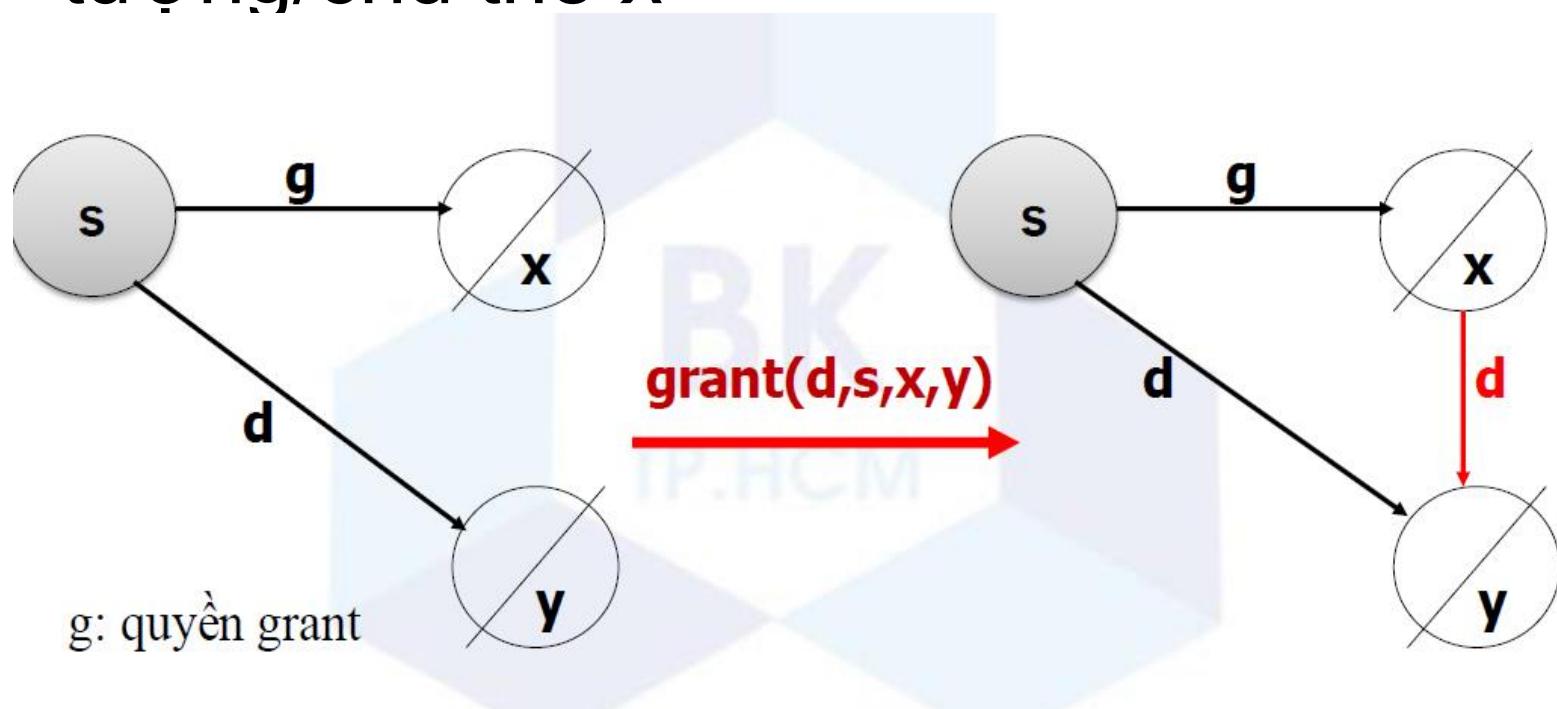
- **take(d, s, x, y)**: chủ thẻ s lấy quyền d trên đối tượng/chủ thẻ y từ đối tượng/chủ thẻ x



Mô hình Take-Grant

Thao tác Take và Grant

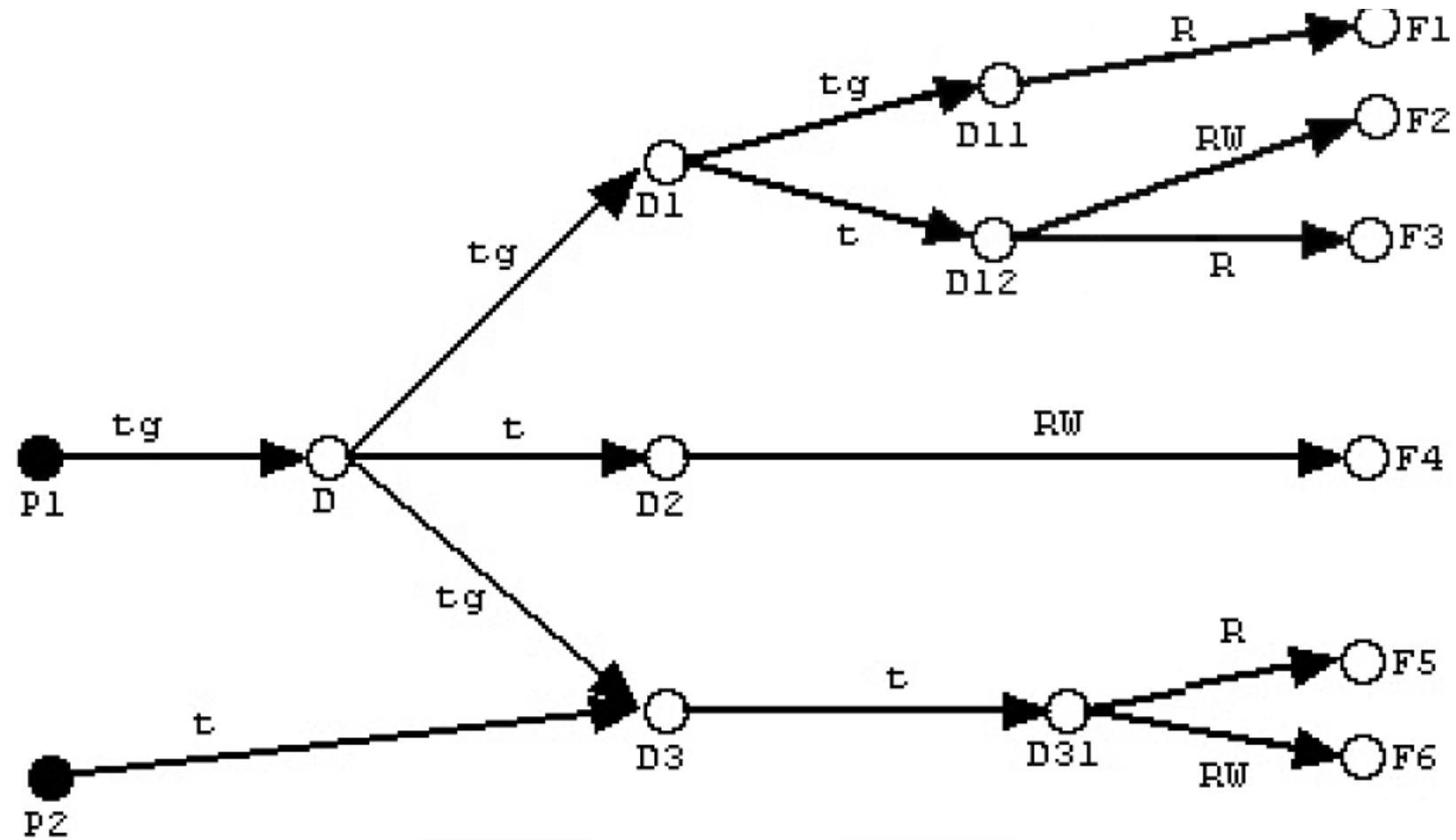
- **grant(d, s, x, y)**: chủ thẻ s gán quyền d trên đối tượng/chủ thẻ y cho đối tượng/chủ thẻ x



Mô hình Take-Grant

- Các loại quyền truy cập: read, write, take, grant
 - read, write: không làm thay đổi trạng thái định quyền
 - take, grant: làm thay đổi trạng thái định quyền
- Các loại thao tác truyền quyền: take, grant, create, remove
 - take, grant: lấy và gán quyền
 - create(s, x): chủ thể s tạo đối tượng/chủ thể x. Khi đó cung nối giữa s và x sẽ được đánh nhãn *p* (possess: sở hữu)
 - Remove p(s, x): chủ thể s bị thu hồi quyền *p* trên đối tượng/chủ thể x

Mô hình Take-Grant



Mô hình Take-Grant

- Khuyết điểm của mô hình Take-Grant:
- Không có tính chọn lọc của các quyền quản lý:
- Tất cả các quyền của s đều có thể bị truyền đi nếu s sở hữu quyền GRANT
- Tất cả các quyền của o/s đều có thể bị lấy đi(truyền đi) nếu có một quyền TAKE trên nó.
- Không quản lý được sự lan truyền quyền
- Tính không cục bộ: nếu s có quyền GRANT trên o thì s có thể truyền bất kỳ quyền gì của mình cho o. Như vậy không kiểm soát được tập quyền có thể có trên o.
- Khả năng lan truyền ngược của dòng di chuyển quyền

3. Điều khiển truy cập bắt buộc – Mandatory Access Control (MAC):

3.1. Giới thiệu:

- Điều khiển truy cập bắt buộc: Là một chính sách truy cập không do cá nhân sở hữu tài nguyên quyết định mà do hệ thống quyết định. MAC được dùng trong các hệ thống đa cấp, là những hệ thống sử lý các loại dữ liệu nhạy cảm như các thông tin được phân loại theo mức độ bảo mật trong cơ quan chính phủ và trong quân đội. Một hệ thống đa cấp là một hệ thống máy tính duy nhất chịu trách nhiệm xử lý nhiều cấp thông tin nhạy cảm giữa các chủ thể và các đối tượng trong hệ thống.
 - Tính nhạy cảm (sensitivity) của thông tin (thường được gán nhãn) chứa trong các đối tượng và sự trao quyền chính thức cho các chủ thể truy cập các thông tin nhạy cảm này.

Điều khiển truy cập bắt buộc

Các mức nhạy cảm:

- **Tối mật** (Top Secret - T): được áp dụng với thông tin mà nếu bị lộ có thể dẫn đến những thiệt hại trầm trọng đối với an ninh quốc gia.
- **Tuyệt mật** (Secret - S): được áp dụng với thông tin mà nếu bị lộ có thể dẫn đến một loạt thiệt hại đối với an ninh quốc gia.
- **Mật** (Confidential - C): được áp dụng với thông tin mà nếu bị lộ có thể dẫn đến thiệt hại với an ninh quốc gia.
- **Không phân loại** (Unclassified - U): những thông tin không gây thiệt hại đối với an ninh quốc gia nếu bị tiết lộ.

Điều khiển truy cập bắt buộc

- MAC không cho phép người tạo ra các đối tượng (thông tin/tài nguyên) có toàn quyền truy cập các đối tượng này.
- Quyền truy cập đến các đối tượng (thông tin/tài nguyên) do người quản trị hệ thống định ra trước trên cơ sở chính sách an toàn thông tin của tổ chức đó.
- MAC thường được sử dụng phổ biến trong các cơ quan an ninh, quân đội và ngân hàng.

Điều khiển truy cập bắt buộc

Ví dụ: một tài liệu được tạo ra và được đóng dấu “Mật”:

- Chỉ những người có trách nhiệm trong tổ chức mới được quyền xem và phổ biến cho người khác;
- Tác giả của tài liệu không được quyền phổ biến đến người khác.

3.2. Mô hình điều khiển truy cập bắt buộc:

Mô hình Bell-LaPadula:

- Mô hình Bell-La Padula là mô hình bảo mật đa cấp thường được sử dụng trong quân sự, nhưng nó cũng có thể áp dụng cho các lĩnh vực khác.
- Trong quân sự, các tài liệu được gán một mức độ bảo mật, chẳng hạn như không phân loại, mật, bí mật và tối mật. Người dùng cũng được ấn định các cấp độ bảo mật, tùy thuộc vào những tài liệu mà họ được phép xem.
 - Một vị tướng quân đội có thể được phép xem tất cả các tài liệu, trong khi một trung úy có thể bị hạn chế chỉ được xem các tài liệu mật và thấp hơn.
 - Một tiến trình chạy nhân danh một người sử dụng có được mức độ bảo mật của người dùng đó.

Nguyên tắc bảo mật tài nguyên của mô hình Bell-La Padula:

- Nguyên tắc đọc xuống: một người dùng ở mức độ bảo mật k có thể đọc các đối tượng cùng mức hoặc thấp hơn.
- VD: Một vị tướng có thể đọc các tài liệu của một trung úy, nhưng một trung úy không thể đọc các tài liệu của vị tướng đó.

- Nguyên tắc ghi lên: một người dùng ở mức bảo mật k chỉ có thể ghi các đối tượng ở cùng cấp độ hoặc cao hơn.
- VD: Một trung úy có thể nối thêm một tin nhắn vào hộp thư của chung về tất cả mọi thứ ông biết, Nhưng một vị tướng không thể ghi thêm một tin nhắn vào hộp thư của trung úy với tất cả mọi thứ ông ấy biết vì vị tướng có thể đã nhìn thấy tài liệu bí mật trên thư mà không thể được tiết lộ cho một trung úy.

Mô hình Biba

- Do Biba đề nghị năm 1977
- Mô hình Biba tập trung vào việc **bảo vệ tính toàn vẹn** của dữ liệu
- Mô hình Biba phân loại chủ thể, đối tượng theo mức toàn vẹn (*integrity level*)
- Các nhóm phân loại gồm:
 - Crucial (C)
 - Very Important (VI)
 - Important (I)

C > VI > I

- **Quyền truy xuất (access mode):** truy xuất đối tượng
- **Chỉnh sửa (modify):** ghi thông tin lên đối tượng
- **Liên hệ (invoke):** quyền giữa 2 chủ thể, cho phép 2 chủ thể liên lạc với nhau
- **Quan sát (observe):** đọc thông tin của đối tượng
- **Thực thi (execute):** thực thi chương trình

Chính sách toàn vẹn: **mức toàn vẹn cố định**

- **Tính chất toàn vẹn đơn giản (Simple integrity property)**: một chủ thể s có thể quan sát được đối tượng o nếu và chỉ nếu:

$$i(s) \leq i(o)$$

Không đọc xuống (No read down)

- **Tính chất toàn vẹn sao (Integrity star property)**: một chủ thể s có thể chỉnh sửa được đối tượng o nếu và chỉ nếu:

$$i(o) \leq i(s)$$

Không ghi lên (No write up)

Chính sách toàn vẹn: mức toàn vẹn cố định

- **Tính chất liên hệ (Invocation property)**: một chủ thẻ s1 có thể *liên hệ* với chủ thẻ s2 nếu và chỉ nếu:

$$i(s2) \leq i(s1)$$

- Một luồng thông tin o_1, \dots, o_n được gọi là an toàn nếu $f_O(o_1) \geq f_O(o_n)$

- Mô hình Biba bảo vệ tính **toàn vẹn** và không cung cấp tính **mật** nên cần kết hợp với những mô hình khác.
- Mô hình Lipner là mô hình kết hợp giữa mô hình Bell-LaPadula và mô hình Biba.

Một số cách điều khiển truy cập khác:

Điều khiển truy nhập dựa trên vai trò – Role-Based Access Control (RBAC)

VD: pm quản lý trường học chia người dùng thành các nhóm gán sẵn quyền truy cập vào hệ thống:

- Nhóm quản lý được quyền truy cập vào tất cả các thông tin trong hệ thống.
- Nhóm GV được truy cập vào CSDL các môn học, cập nhật điểm GV phụ trách.
- Nhóm SV chỉ được quyền xem nội dung các môn học, tải tài liệu học tập và xem điểm của mình.

Điều khiển truy nhập dựa trên luật – Rule-Based Access Control:

- Điều khiển truy cập dựa trên luật cho phép người dùng truy cập vào hệ thống thông tin dựa trên các luật (rules) đã được định nghĩa trước.
- Các luật có thể được thiết lập để hệ thống cho phép truy cập đến các tài nguyên của mình cho người dùng thuộc một tên miền, một dạng hay một dãy địa chỉ IP.

Firewalls/Proxies

Firewalls/Proxies là ví dụ điển hình về điều khiển truy cập dựa trên luật:

- Dựa trên địa chỉ IP nguồn và đích của các gói tin.
- Dựa trên phần mở rộng các Files để lọc các mã độc hại.
- Dựa trên địa chỉ IP hoặc các tên miền để lọc/chặn các Website bị cấm.
- Dựa trên tập các từ khóa để lọc các nội dung bị cấm.

Điều khiển truy cập trong Windows

- Các HĐH Microsoft Windows NT, 2000, XP, 7, 8, 2003, 2008, 2012 server.
- Quản lý người dùng:
 - Các thông tin về người dùng (users) được lưu trong 1 file C:\WINDOWS\system32\config\SAM
 - Thông tin chính về người dùng gồm có:
 - + Tên truy cập (username)
 - + Mật khẩu được lưu dưới dạng hash
 - + Họ tên người dùng
 - + Mô tả người dùng
 - + Thuộc nhóm
 - + Tên thư mục riêng (home directory)
 - + Đường dẫn đến profile

Điều khiển truy cập trong Windows

Hệ điều hành Microsoft Windows thực hiện các cơ chế điều khiển truy cập rất chi tiết. Trong việc quản trị hệ thống, các quản trị viên thường làm việc với người dùng, nhóm và các đối tượng. Các quyền cơ bản trong Windows:

- **Full control** (Toàn quyền): Cho phép thay đổi quyền, chủ sở hữu và xóa thư mục con, file.
- **Modify** (Sửa): Có quyền sửa chữa như tạo, xoá, sửa folder.
- **Read & Execute** (Đọc và thực thi): Quyền đọc (bao gồm cả việc gọi các phương thức, các file ứng dụng chạy ngầm).
- **List Folder Contents** (Liệt kê nội dung thư mục): Cho phép xem tên file và subdomain trong thư mục.
- **Special permissions**: đọc/ghi thuộc tín, chuyển quyền sở hữu...

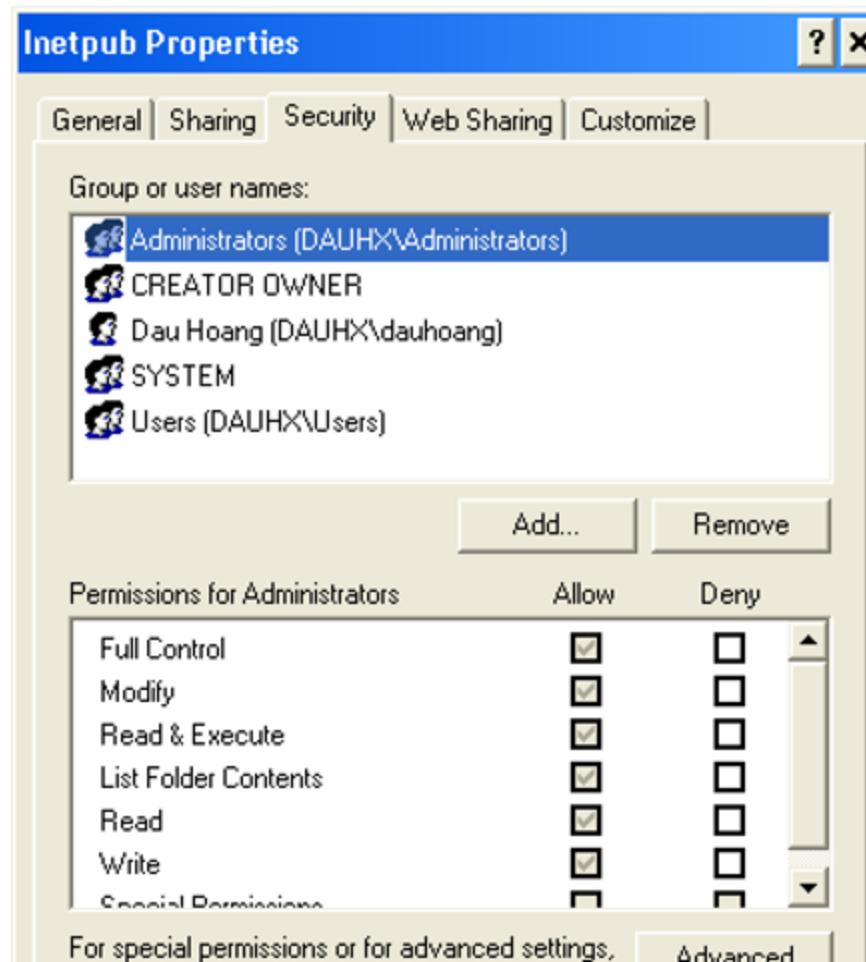
Điều khiển truy cập trong Windows

- Người dùng được tổ chức thành các nhóm (groups), mỗi nhóm có các quyền truy cập khác nhau vào các tài nguyên hệ thống. Một người dùng có thể thuộc nhiều nhóm và một nhóm có thể có nhiều người dùng.
 - Nếu một người dùng thuộc nhiều nhóm, thì quyền truy cập là hợp của quyền truy cập các nhóm mà người dùng là thành viên.
- Các nhóm ngầm định: Administrators, Power Users, Backup Operators, Users, Guests.
- Các người dùng ngầm định: Administrator, everyone, Guest,...

Quản lý quyền truy cập: sử dụng kết hợp 2 phương pháp DAC + Role-Based AC:

- Quyền truy cập được tổ chức theo mô hình phân cấp của các miền được quản lý: giống tổ chức cây tên miền.
- Quyền truy cập tại mỗi miền được tổ chức thành các nhóm “vai trò” và đến từng người dùng.
- Mỗi đối tượng (file, thư mục, tiến trình, ...) trong hệ thống đều có một (hoặc nhiều) chủ sở hữu, thường là người tạo ra đối tượng. Chủ sở hữu có thể được chuyển đổi.
- Quyền truy cập các đối tượng con được thừa hưởng từ quyền truy cập các đối tượng cha, mẹ.

Giao diện quản lý quyền truy cập của Microsoft Windows:



Advanced Security Settings for Inetpub



Permissions | Auditing | Owner | Effective Permissions |

To view more information about Special permissions, select a permission entry, and then click Edit.

Permission entries:

Type	Name	Permission	Inherited From	Apply To
Allow	Administrators (DAUH...)	Full Control	C:\	This folder, subfolders...
Allow	SYSTEM	Full Control	C:\	This folder, subfolders...
Allow	Dau Hoang (DAUHX...)	Full Control	C:\	This folder only
Allow	CREATOR OWNER	Full Control	C:\	Subfolders and files only
Allow	Users (DAUHX\Users)	Read & Execute	C:\	This folder, subfolders...
Allow	Users (DAUHX\Users)	Special	C:\	This folder and subfol...

Add...

Edit...

Remove

Inherit from parent the permission entries that apply to child objects. Include these with entries explicitly defined here.

Replace permission entries on all child objects with entries shown here that apply to child objects

OK

Cancel

Apply

Điều khiển truy cập dựa trên mật khẩu:

Thông thường mỗi người dùng được cấp 1 tài khoản (account). Để truy cập tài khoản, thường cần có:

- Tên người dùng (username)
- Mật khẩu (Password)
 - Mật khẩu có thể ở dạng nguyên bản (plain text)
 - Mật khẩu có thể ở dạng mã hoá (encrypted text)

Các thuật toán thường dùng để mã hoá mật khẩu: MD4, MD5, SHA-1, SHA256,...

- Mật khẩu có thể được dùng nhiều lần hoặc 1 lần (one time password).

Tính bảo mật của kỹ thuật điều khiển truy cập sử dụng mật khẩu dựa trên:

- Độ khó đoán của mật khẩu: Dùng nhiều loại ký tự (Chữ thường, hoa, chữ số, ký tự đặc biệt). VD: abc1234, aBc*1#24.
- Độ dài của mật khẩu: Mật khẩu tốt có chiều dài ≥ 8 ký tự.

Điều khiển truy cập dựa trên thẻ thông minh

- Thẻ thông minh (Smartcard) là các thẻ nhựa có gắn các chip điện tử.
- Có khả năng tính toán và các thông tin lưu trong thẻ được mã hoá.
- Smartcard sử dụng hai yếu tố (two-factors) để xác thực và nhận dạng chủ thẻ:
 - Cái bạn có (what you have): thẻ
 - Cái bạn biết (what you know): số PIN

CÁC KỸ THUẬT MÃ HÓA THÔNG TIN

1. Giới thiệu:

- Thông tin chưa được mã hóa: là thông tin ở dạng có thể hiểu được (bản rõ).
- Thông tin đã được mã hóa: là thông tin ở dạng đã bị xáo trộn (bản mã).
- Mã hóa (Encryption) là hành động xáo trộn bản rõ để chuyển thành bản mã.
- Giải mã (Decryption) là hành động xáo trộn bản mã để chuyển thành bản rõ.

- Mã hóa: sử dụng một thuật toán (Algorithm) để mã hóa thông tin.
- Một bộ mã hóa (Cipher) là một thuật giải để mã hóa và giải mã.
- Khóa (Key): là một chuỗi được sử dụng trong giải thuật mã hóa và giải mã.
- Mã hóa bí mật: một khóa được sử dụng trong giải thuật mã hóa và giải mã.
- Mã hóa công khai: một cặp khóa được sử dụng, trong đó khóa công khai để mã hóa, khóa bí mật để giải mã.

- Không gian khóa: là tổng số khóa có thể có của một hệ mã hóa.
- Hàm băm: là một ánh xạ chuyển các dữ liệu có kích thước thay đổi về dữ liệu có kích thước cố định.
- Phá mã: là quá trình giải mã thông điệp đã bị mã hóa mà không cần có trước thông tin về giải thuật mã hóa và khóa mã (key).

Mục đích:

- Đảm bảo sự bí mật của dữ liệu.
- Dữ liệu được mã hóa được trình bày ở 1 dạng khác nhau so với dữ liệu chưa được mã hóa.
- Trong trường hợp kẻ tấn công có được dữ liệu đã được mã hóa thì vẫn không thể suy luận ra dữ liệu trước khi mã hóa.
- Chỉ có người có mật mã mới có thể đọc được dữ liệu đã được mã hóa.

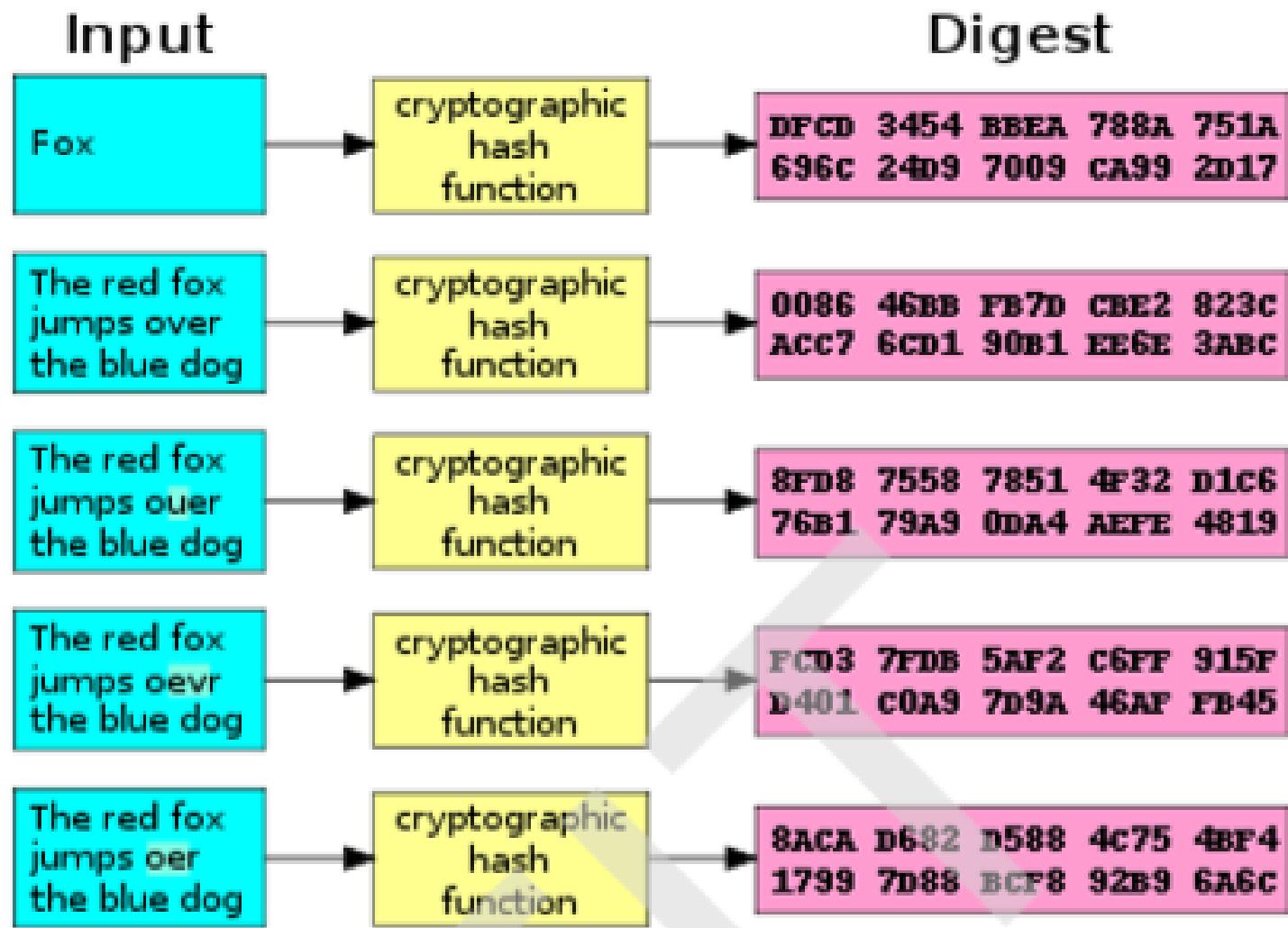
Vai trò của mã hóa trong ATTT:

Mã hóa thông tin có thể được sử dụng để đảm bảo an toàn thông tin trên đường truyền với các thuộc tính:

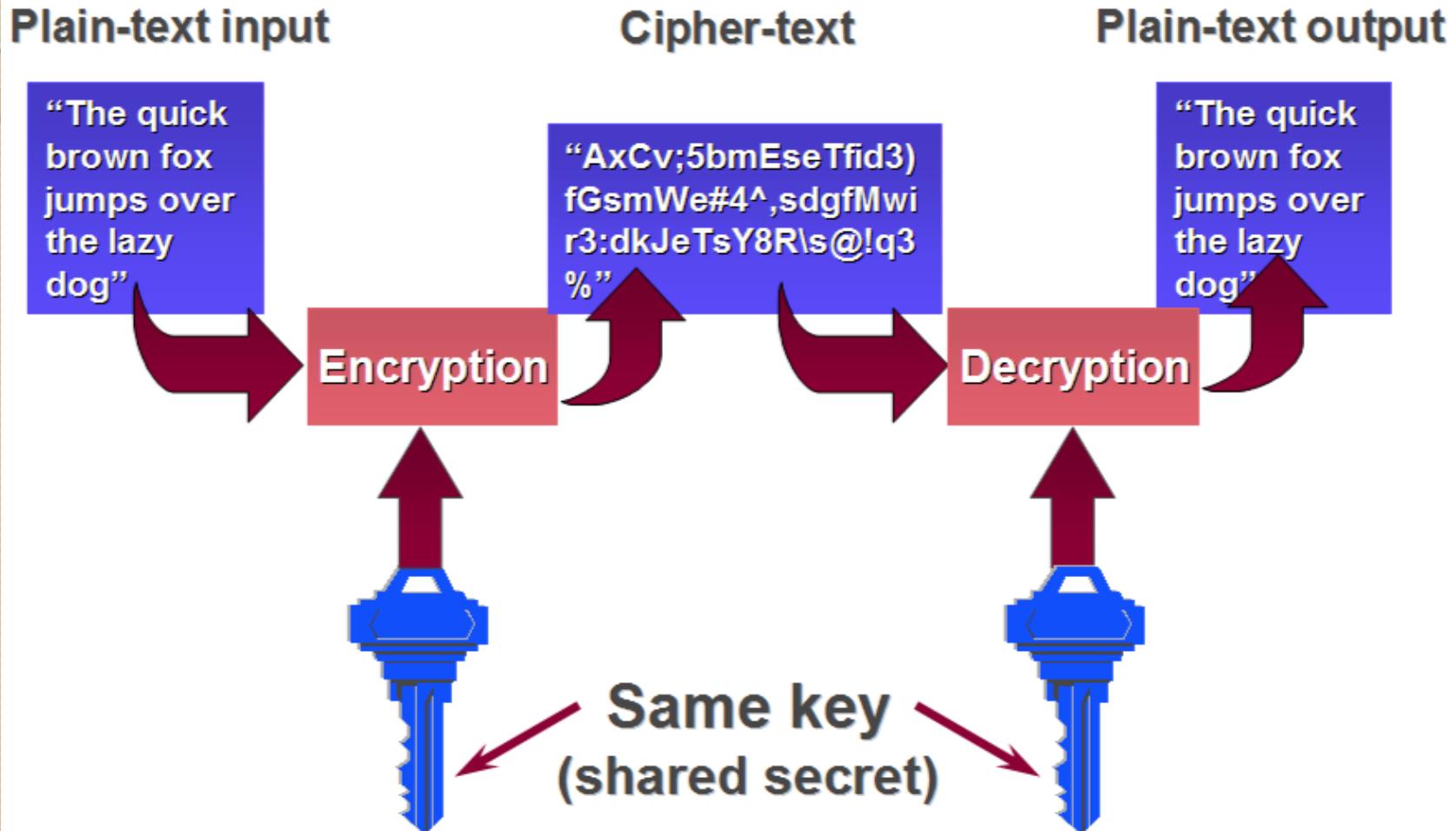
- Bí mật (Confidentiality): đảm bảo chỉ những người có thẩm quyền mới có khả năng truy cập vào thông tin.
- Toàn vẹn (integrity): đảm bảo dữ liệu không bị sửa đổi bởi các bên có đủ thẩm quyền.
- Sẵn dùng (Availability): dữ liệu phải luôn ở trạng thái sẵn sàng cho người dùng có quyền truy cập.

Phân loại:

- Mã hóa đối xứng: là dạng mã hóa trong đó một khóa (bí mật) được sử dụng cho cả giải thuật mã hóa và giải mã.
- Mã hóa bất đối xứng là dạng mã hóa trong đó một cặp khóa được sử dụng: khóa công khai dùng để mã hóa, khóa riêng dùng để giải mã.
- Hàm băm: là một ánh xạ chuyển các dữ liệu có kích thước thay đổi về dữ liệu có kích thước cố định.



Mã hóa đối xứng:



Mã hóa đối xứng: MH Ceasar

Là phương pháp thay thế mỗi chữ trong bản tin bằng chữ đứng sau nó k vị trí trong bảng chữ cái.

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	80	90

Vd: dữ liệu ban đầu: hello kitty

→ với k=3, dữ liệu sau khi mã hóa là:

KHOOR NLWWB

Mã hóa đối xứng: MH Ceasar

- Gọi k là khóa, P là ký tự cần mã, C là kết quả mã hóa:
- Mã hóa:
$$C = (P+k) \text{mod } 26$$
- Giải mã:
$$P = (C-k) \text{ mod } 26$$

Mã hóa đx: Thay thế đơn bảng

- B1: Hoán vị bảng chữ cái: bảng chữ cái đã được hoán vị gọi là khóa.
- B2: sd bảng chữ cái đã hoán vị để mã hóa.

VD:

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
Y	T	Z	L	M	D	A	E	F	G	B	H	I	J	K	C	N	O	P	Q	R	S	U	V	W	X

BD: ABCDEFGHIJKLMNOPQRSTUVWXYZ

HV: YTZL MDAEFGBHJKCNOOPQRS UVWX

Dãy ban đầu: HOW AREYOU

→ Mã hóa: EKUYOMWKR

Mã hóa đx: Thay thế đa bảng

Ke y	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
A	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
B	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	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	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

CPPJRLNLBQYYSGWC và Key là: **ABCDEFGHIJKLMOP**.

P= HOWAREYOU

Key= HELLO (Được viết lặp lại cho có độ dài bằng với độ dài chuỗi cần mã hóa)

→ key=HELLOHELL

Chuỗi sau khi mã hóa: OSHLFLCZF

Mã hóa đx: One-Time Pad

- Sử dụng lại ma trận bảng ký tự của mã hóa thay thế đa bảng.
- Khóa là 1 chuỗi ký tự ngẫu nhiên có độ dài bằng với độ dài chuỗi dữ liệu cần mã hóa.
- Mỗi khóa chỉ sử dụng 1 lần.
- Một chuỗi dữ liệu được mã hóa có thể được giải mã ra nhiều thông điệp có nghĩa (nếu dùng nhiều khóa giải mã khác nhau) nhưng không biết thông điệp nào là thật

Mã hóa đx: MH hoán vị

Giả sử chuỗi ký tự cần mã hóa có độ dài n ký tự

- $m=\sqrt{n}$
- Viết chuỗi ký tự cần mã hóa vào 1 ma trận m dòng và m cột (viết theo từng dòng cho đến khi đủ $m \times m$ ký tự)
- Thực hiện hoán đổi vị trí của dòng, cột.
- Viết kết quả là cách liệt kê ký tự theo cột (từ cột 1 đến cột cuối cùng).

Mã hóa đx: MH RC4

- RC4 được dùng trong giao thức SSL để bảo mật dữ liệu trong quá trình truyền dữ liệu giữa WebServer và trình duyệt Web, ngoài ra RC4 còn được sử dụng trong mã hóa WEP của mạng Wireless LAN.
- Để đơn giản chúng ta xem xét một mô hình thu nhỏ của RC4 gọi là TinyRC4.
- TinyRC4 dùng 2 mảng S và T mỗi mảng gồm 8 số nguyên 3 bit (từ 0 đến 7). Khóa là một dãy gồm N số nguyên 3 bit với N có thể lấy giá trị từ 1 đến 8. Bộ sinh số mỗi lần sinh ra 3 bit để sử dụng trong phép XOR. Quá trình sinh số của TinyRC4 gồm hai giai đoạn:

- Để minh họa MH RC4, dùng bảng mã 3 bit để biểu diễn 8 chữ cái A, B, C, D, E, F, G, H:

Chữ cái	Nhi phân
A	000
B	001
C	010
D	011
E	100
F	101
G	110
H	111

Mã hóa đx: MH RC4

- Giai đoạn khởi tạo:

```
/* Khoi tao day so S va T */
for i = 0 to 7 do
    S[i] = i;
    T[i] = K[i mod N];
next i
/* Hoan vi day S */
j = 0;
for i = 0 to 7 do
    j = (j + S[i] + T[i]) mod 8;
    Swap(S[i], S[j]);
next i
```

- Dãy S gồm các số nguyên 3 bit từ 0 đến 7 được sắp theo thứ tự tăng dần.
- N là độ dài của khóa K.
- Các phần tử của S được hoán vị lẫn nhau đến một mức độ ngẫu nhiên nào đó.

Mã hóa đx: MH RC4

- VD: Mã hóa bản rõ P = 001000110 (BAG) với khóa K gồm 3 số 2, 1, 3 (N=3)
- Khởi tạo S và T:

S	0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---	---

T	2	1	3	2	1	3	2	1
K								

Mã hóa đx: MH RC4

i=0

T	2	1	3	2	1	3	2	1
---	---	---	---	---	---	---	---	---

$$\begin{aligned}j &= (j + S[i] + T[i]) \bmod 8 \\&= (0+0+2) \bmod 8 = 2\end{aligned}$$

S	0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---	---

Swap(S[0],S[2])

i=2

T	2	1	3	2	1	3	2	1
---	---	---	---	---	---	---	---	---

$$\begin{aligned}j &= (j + S[i] + T[i]) \bmod 8 \\&= (4+3+0) \bmod 8 = 7\end{aligned}$$

S	2	4	0	3	1	5	6	7
---	---	---	---	---	---	---	---	---

Swap(S[2],S[7])

i=1

T	2	1	3	2	1	3	2	1
---	---	---	---	---	---	---	---	---

$$\begin{aligned}j &= (j + S[i] + T[i]) \bmod 8 \\&= (2+1+1) \bmod 8 = 4\end{aligned}$$

S	2	1	0	3	4	5	6	7
---	---	---	---	---	---	---	---	---

Swap(S[1],S[4])

i=3

T	2	1	3	2	1	3	2	1
---	---	---	---	---	---	---	---	---

$$\begin{aligned}j &= (j + S[i] + T[i]) \bmod 8 \\&= (7+2+3) \bmod 8 = 4\end{aligned}$$

S	2	4	7	3	1	5	6	0
---	---	---	---	---	---	---	---	---

Swap(S[3],S[4])

Mã hóa đx: MH RC4

i=4

T	2	1	3	2	1	3	2	1
---	---	---	---	---	---	---	---	---

$$\begin{aligned} j &= (j + S[i] + T[i]) \bmod 8 \\ &= (4+1+3) \bmod 8 = 0 \end{aligned}$$

S	2	4	7	1	3	5	6	0
---	---	---	---	---	---	---	---	---

↑ ↑
Swap(S[4],S[0])

i=6

T	2	1	3	2	1	3	2	1
---	---	---	---	---	---	---	---	---

$$\begin{aligned} j &= (j + S[i] + T[i]) \bmod 8 \\ &= (0+2+6) \bmod 8 = 0 \end{aligned}$$

S	5	4	7	1	2	3	6	0
---	---	---	---	---	---	---	---	---

↑ ↑
Swap(S[6],S[0])

i=5

T	2	1	3	2	1	3	2	1
---	---	---	---	---	---	---	---	---

$$\begin{aligned} j &= (j + S[i] + T[i]) \bmod 8 \\ &= (0+3+5) \bmod 8 = 0 \end{aligned}$$

S	3	4	7	1	2	5	6	0
---	---	---	---	---	---	---	---	---

↑ ↑
Swap(S[5],S[0])

i=7

T	2	1	3	2	1	3	2	1
---	---	---	---	---	---	---	---	---

$$\begin{aligned} j &= (j + S[i] + T[i]) \bmod 8 \\ &= (0+1+0) \bmod 8 = 1 \end{aligned}$$

S	6	4	7	1	2	3	5	0
---	---	---	---	---	---	---	---	---

↑ ↑
Swap(S[7],S[1])

→ KQ cuối cùng

S	6	0	7	1	2	3	5	4
---	---	---	---	---	---	---	---	---

Mã hóa đx: MH RC4

Mã hóa/giải mã:

```
i, j = 0;  
while (true)  
    i = (i + 1) mod 8;  
    j = (j + S[i]) mod 8;  
    Swap (S[i], S[j]);  
    t = (S[i] + S[j]) mod 8;  
    k = S[t];  
end while;
```

Mã hóa đx: MH RC4

VD: Mã hóa bản rõ: P = 001000110 (BAG)

Bước 0: i, j = 0

S	6	0	7	1	2	3	5	4
---	---	---	---	---	---	---	---	---

$$i = (i+1) \bmod 8 = 1$$

$$j = (j+S[i]) \bmod 8 = 0$$

Swap(S[1], S[0])

S	0	6	7	1	2	3	5	4
---	---	---	---	---	---	---	---	---

$$t = (S[1] + S[0]) \bmod 8 = 6$$

$$k = S[6] = 5 \rightarrow 101$$

Bước 1: i=1, j=0

S	0	6	7	1	2	3	5	4
---	---	---	---	---	---	---	---	---

$$i = (i+1) \bmod 8 = 2$$

$$j = (j+S[2]) \bmod 8 = 7$$

Swap(S[2], S[7])

S	0	6	4	1	2	3	5	7
---	---	---	---	---	---	---	---	---

$$t = (S[2] + S[7]) \bmod 8 = 3$$

$$k = S[3] = 1 \rightarrow 001$$

Bước 2: i = 2, j = 7

S	0	6	4	1	2	3	5	7
---	---	---	---	---	---	---	---	---

$$i = (i+1) \bmod 8 = 3$$

$$j = (j + S[1]) \bmod 8 = 1$$

Swap(S[3], S[1])

S	0	1	4	6	2	3	5	7
---	---	---	---	---	---	---	---	---

$$t = (S[3] + S[1]) \bmod 8 = 7$$

$$k = S[7] = 7 \rightarrow 111$$

Vậy sau khi mã hóa: 001 000 110 XOR 101 001 111

→ 100 001 001 (EBB)

Mã hóa đx: MH RC4

Cơ chế hoạt động:

- Đơn vị mã hóa của CR4 là 1 byte 8 bit.
- Mảng S và T gồm 256 số nguyên 8 bit.
- Khóa K là một dãy gồm N số nguyên 8 bit với N có thể lấy giá trị từ 1 → 256.
- Bộ sinh số mỗi lần sinh ra 1 byte để sử dụng trong phép XOR.

Khởi tạo S, T:

```
for i=0 to 255 do  
    S[i]=i;  
    T[i]=K[i mod keylen];  
end for
```

Mã hóa/giải mã:

```
i=0; j=0;  
while (true)  
    i=(i+1) mod 256  
    j=(j+S[i]) mod 255;  
    Swap(S[i],S[j]);  
    t=(S[i]+S[j]) mod 256  
    k=S[t]  
end while
```

Hoán vị dây S:

```
j=0;  
for i=0 to 255 do  
    j=(j+S[i]+T[i]) mod 255;  
    Swap(S[i],S[j]);  
end for
```

MÃ KHỐI (Block Cipher)

1. Giới thiệu:

Trong thuật toán MH có sd phép toán XOR chỉ cần biết 1 cặp khối bản rõ và bản mã có thể dễ dàng tìm được khóa và dùng khóa này để giải các khối mã khác. VD:

<i>bản rõ:</i>	1111	0000	0011
<i>khóa:</i>	0101	0101	0101
<i>bản mã:</i>	1010	0101	0110

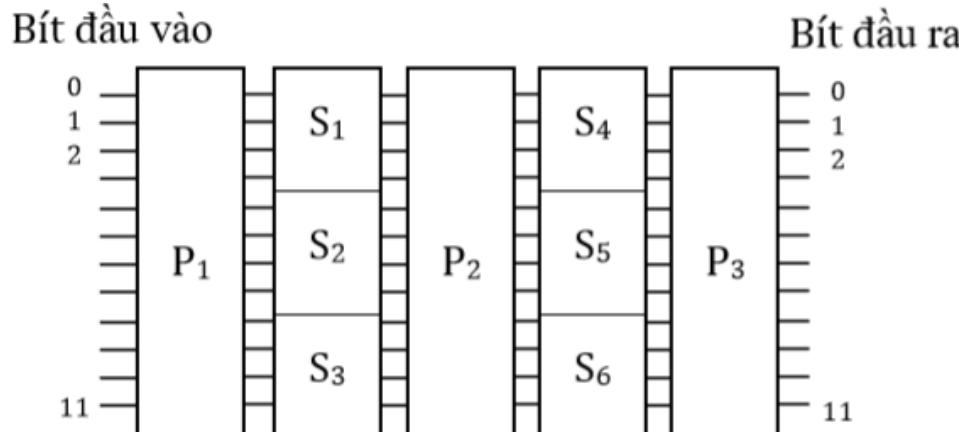
→ Bản rõ và bản mã không có mối liên hệ “toán học” → Lập 1 bảng tra cứu ngẫu nhiên giữa bản rõ và bản mã:

Bản rõ	Bản mã
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

- Tuy nhiên, khi kích thước khối lớn thì số dòng của bảng khóa cũng lớn và gây trở ngại cho việc lưu trữ cũng như trao đổi khóa giữa người gửi và người nhận. Bảng khóa có 264 dòng mỗi dòng 64 bít do đó kích thước khóa sẽ là $64 \times 264 = 270 \approx 1021$ bít. Do đó mã khối an toàn lý tưởng là không khả thi trong thực tế

2. Mạng SPN:

- Trong thực tế, người ta chỉ tìm cách để chỉ cần dùng một khóa có kích thước ngắn để giả lập một bảng tra cứu có độ an toàn xấp xỉ độ an toàn của mã khối lý tưởng.
- Kết hợp hai hay nhiều mã hóa đơn giản lại với nhau để tạo thành một mã hóa tổng (product cipher), trong đó mã hóa tổng an toàn hơn rất nhiều so với các mã hóa thành phần.
- Các mã hóa đơn giản thường là phép thay thế (substitution, S-box) và hoán vị (Permutation, P-box). Do đó, gọi mã hóa tổng là Substitution-Permutation Network (mạng SPN).



Việc kết hợp các S-box và P-box tạo ra hai tính chất quan trọng của mã hóa là tính khuếch tán (diffusion) và tính gây lẫn (confusion). Hai tính chất này do Claude Shannon giới thiệu vào năm 1946, và là cơ sở của tất cả các mã khối hiện nay.

- Tính khuếch tán: một bít của bản rõ tác động đến tất cả các bít của bản mã, hay nói cách khác, một bít của bản mã chịu tác động của tất cả các bít trong bản rõ. Việc làm như vậy nhằm làm giảm tối đa mối liên quan giữa bản rõ và bản mã, ngăn chặn việc suy ra lại khóa. Tính chất này có được dựa vào sử dụng P-box kết hợp S-box.
- Tính gây lẫn: làm phức tạp hóa mối liên quan giữa bản mã và khóa. Do đó cũng ngăn chặn việc suy ra lại khóa. Tính chất này có được dựa vào sử dụng S-box.

Mô hình mã Feistel

Mô hình mã Feistel là một dạng tiếp cận khác so với mạng SP. Mô hình do Horst Feistel đề xuất, cũng là sự kết hợp các phép thay thế và hoán vị. Trong hệ mã Feistel, bản rõ sẽ được biến đổi qua một số vòng để cho ra bản mã cuối cùng.

$$P \xrightarrow{K_1} C_1 \xrightarrow{K_2} C_2 \xrightarrow{K_3} \dots \xrightarrow{K_{n-1}} C_n$$

Bản rõ P và các bản mã C_i được chia thành nửa trái và nửa phải:

$$P = (L_0, R_0)$$

$$C_i = (L_i, R_i) \quad i = 1, 2, \dots, n$$

Quy tắc biến đổi các nửa trái và nửa phải qua các vòng được thực hiện như sau:

$$L_i = R_{i-1}$$

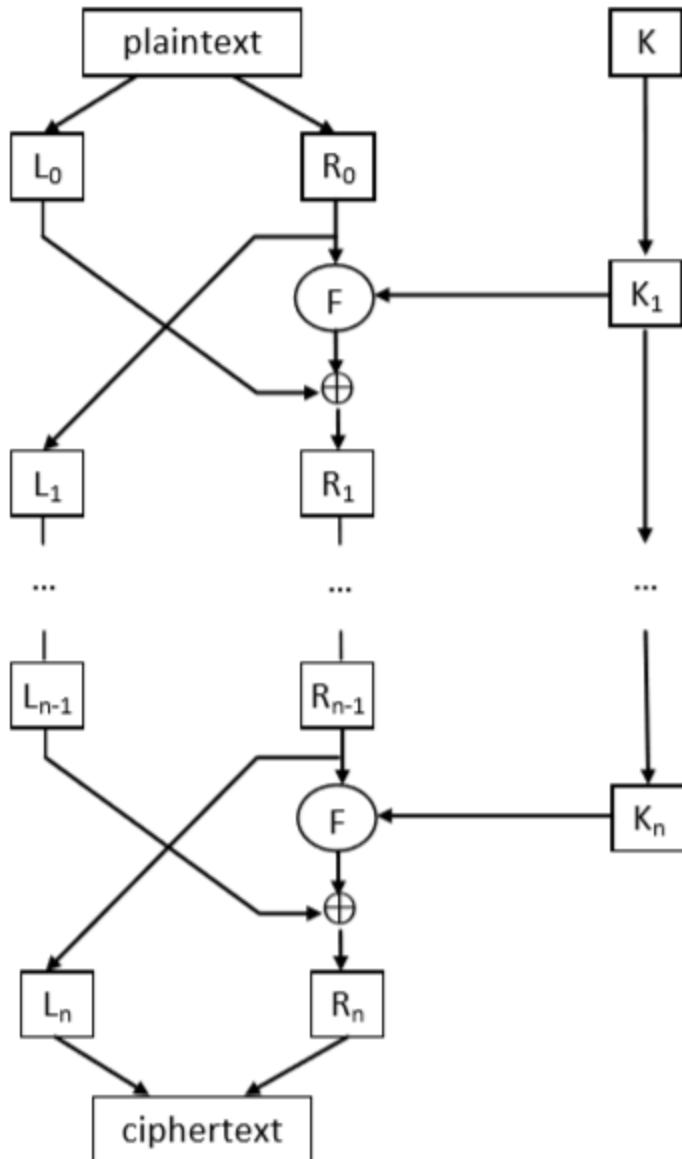
$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

Mô hình mã Feistel

- K_i là 1 khóa con cho vòng thứ i . Khóa này được sinh ra từ khóa K ban đầu theo một thuật toán sinh khóa con: $K \rightarrow K_1 \rightarrow K_2 \rightarrow \dots \rightarrow K_n$
- F là 1 hàm mã hóa dùng chung cho tất cả các vòng. Hàm F đóng vai trò như là phép thay thế còn việc hoán đổi các nǔa trái phải có vai trò hoán vị.
- Bản mã C được tính từ kết xuất của vòng cuối cùng:

$$C = C_n = (L_n, R_n)$$

Sơ đồ tính toán của hệ mã Feistel được thể hiện:



Để giải mã quá trình được thực hiện qua các vòng theo thứ tự ngược lại:

$$C \rightarrow L_n, R_n$$

$$R_{i-1} = L_i \quad (\text{theo mã hóa } L_i = R_{i-1})$$

$$L_{i-1} = R_i \oplus F(R_{i-1}, K_i) \quad (\text{theo mã hóa } R_i = L_{i-1} \oplus F(R_{i-1}, K_i))$$

Và cuối cùng bản rõ là: **P = (L₀, R₀)**

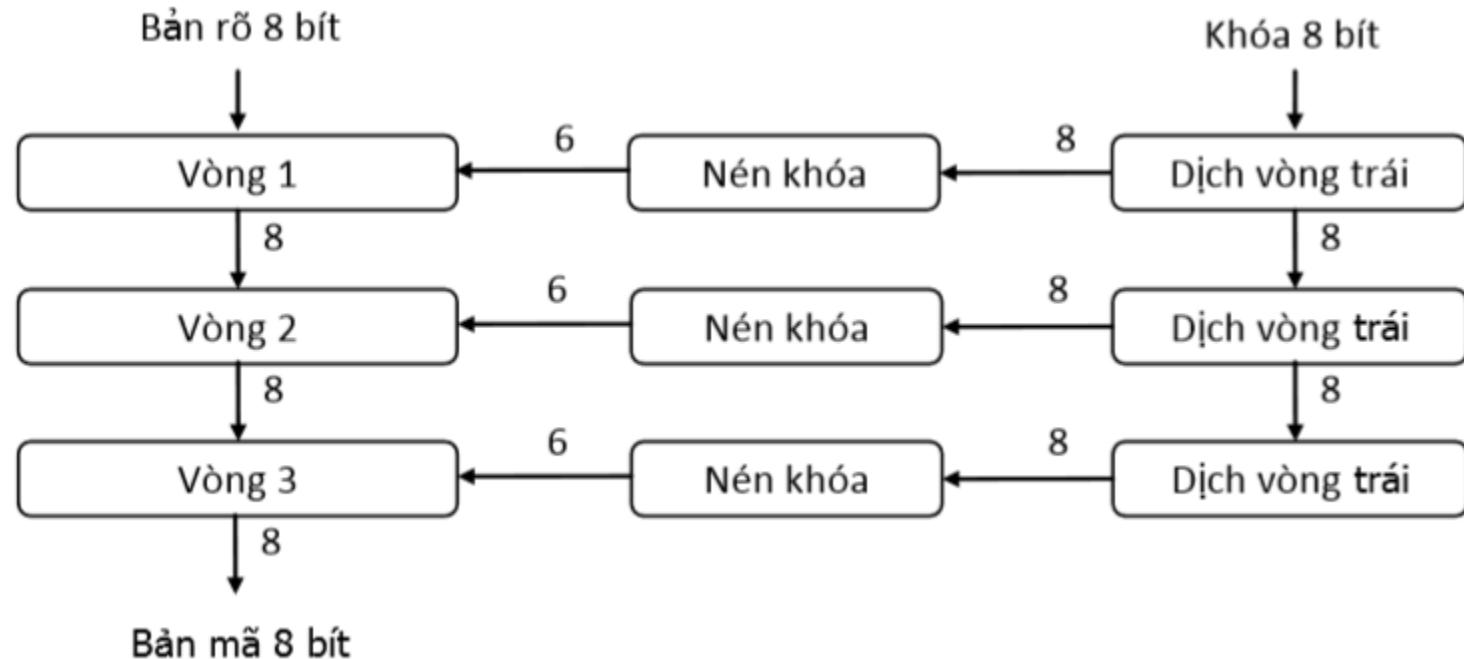
Ứng với các hàm F và thuật toán sinh khóa con khác nhau thì sẽ có các phương pháp mã hóa khác nhau, phần tiếp theo sẽ trình bày mã hóa **DES**, là một phương pháp mã hóa dựa trên nguyên tắc của hệ mã Feistel.

MH: DES (Data Encryption Standard)

Xét thuật toán TinyDES là thuật toán thu nhỏ của DES. Mã TinyDES có các t/c sau:

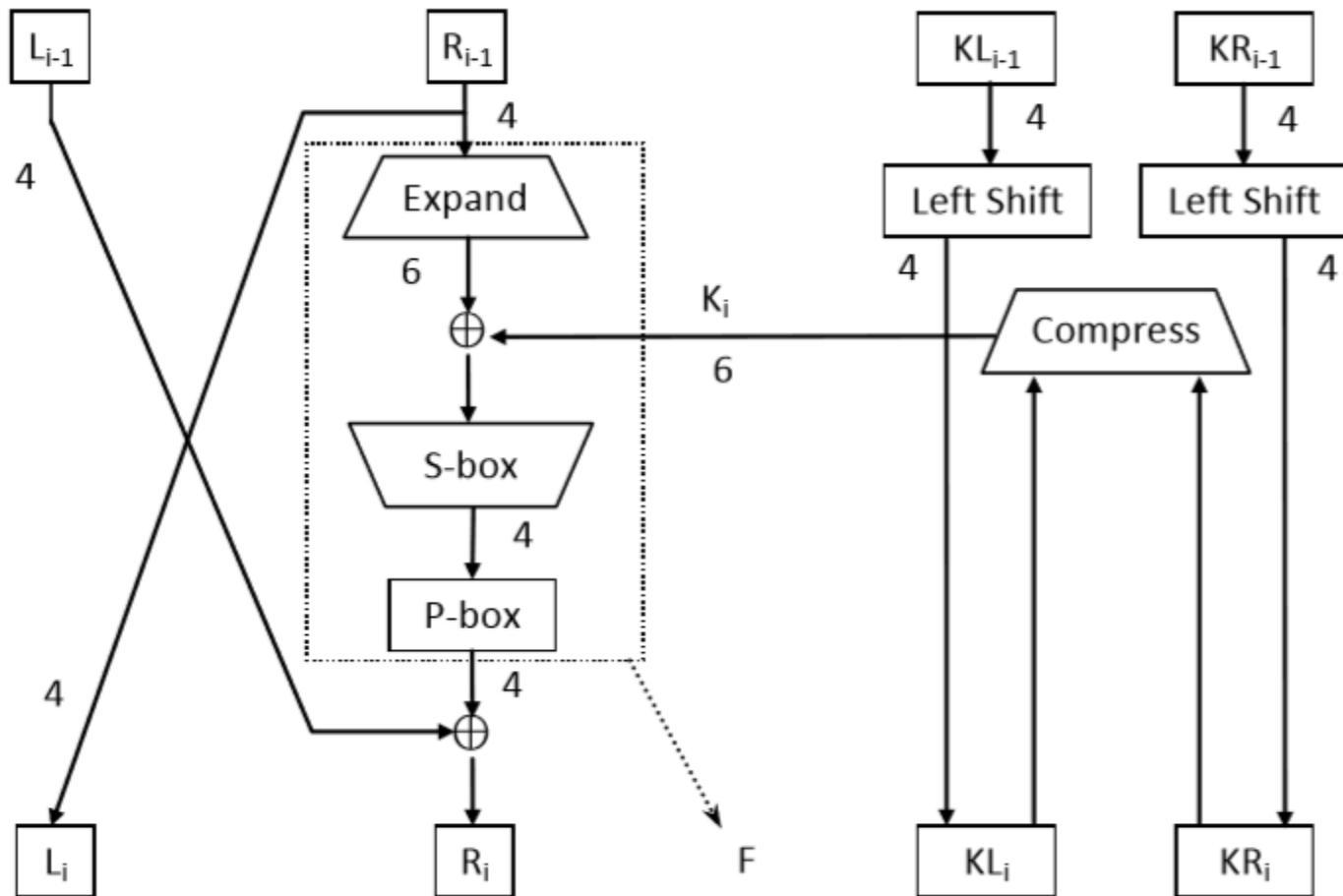
- Là mã thuộc hệ mã Feistel gồm 3 vòng.
- Kích thước tối đa của khối là 8 bit.
- Kích thước khóa là 8 bit.
- Mỗi vòng của TinyDES dùng khóa con có kích thước 6 bit được trích ra từ khóa chính.

- Các vòng lặp của mã TinyDES:



Sơ đồ mã TinyDES gồm 2 phần: các vòng Feistel và thuật toán sinh khóa con.

Các vòng của TinyDES:



$$F(R_{i-1}, K_i) = P\text{-box}(S\text{-box}(Expand(R_{i-1}) \oplus K_i))$$

Hàm **Expand** vừa mở rộng vừa hoán vị R_{i-1} từ 4 bit lên 6 bit. Hàm **S-boxes** biến đổi một số 6 bit đầu vào thành một số 4 bit đầu ra. Hàm **P-box** là một hoán vị 4 bit. Mô tả của các hàm trên là như sau:

- **Expand**: gọi 4 bit của R_{i-1} là $b_0 b_1 b_2 b_3$. Hàm Expand hoán vị và mở rộng 4 bit thành 6 bit cho kq: $b_2 b_3 b_1 b_2 b_1 b_0$.

VD: $R_0 = 0110 \rightarrow \text{Expand}(R_0) = 101110$

- **S-box**: gọi $b_0 b_1 b_2 b_3 b_4 b_5$ là 6 bit đầu vào S-box, ứng với mỗi trường hợp 6 bit đầu vào sẽ cho 4 bit đầu ra. Việc tính các bit đầu ra dựa trên bảng sau:

		$b_1 b_2 b_3 b_4$															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
$b_0 b_5$	00	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111
	01	0000	1111	0111	0100	1110	0010	1101	0001	1010	0110	1100	1011	1001	0101	0011	1000
	10	0100	0001	1110	1000	1101	0110	0010	1011	1111	1100	1001	0111	0011	1010	0101	0000
	11	1111	1100	1000	0010	0100	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101

Hai bit b_0b_1 xác định thứ tự hàng, 4 bit $b_1b_2b_3b_4$ xác định thứ tự cột của bảng, từ đó tính được 4 bit đầu ra. Viết lại bảng trên dưới dạng số thập lục phân:

	$b_1b_2b_3b_4$																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7	
b_0b_5	1	0	F	7	4	E	2	D	1	A	6	C	B	9	5	3	8
2	4	1	E	8	D	6	2	B	F	C	9	7	3	A	5	0	
3	F	C	8	2	4	9	1	7	5	B	3	E	A	0	6	D	

VD: $X = 101010 \rightarrow S\text{-box}(X) = 0110$

- **P-box**: thực hiện hoán vị 4 bit đầu $b_0b_1b_2b_3$ cho ra kết quả $b_2b_0b_3b_1$.

- Thuật toán sinh khóa con của TinyDES:
Khóa K 8 bit ban đầu được chia thành 2 nửa trái phải KL_0 và KR_0 , mỗi nửa có kích thước 4 bit.
 - **Vòng 1:** KL_0 và KR_0 được dịch vòng trái **1 bit** để có được KL_1 và KR_1 .
 - **Vòng 2:** KL_1 và KR_1 được dịch vòng trái **2 bit** để có được KL_2 và KR_2 .
 - **Vòng 3:** KL_2 và KR_2 được dịch vòng trái **1 bit** để có KL_3 và KR_3 .
 - Cuối cùng khóa K_i của mỗi vòng được tạo ra bằng cách hoán vị và nén (compress) 8 bit của KL_i và KR_i ($k_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7$) thành kết quả gồm 6 bit : **$k_5 k_1 k_3 k_2 k_7 k_0$** .

VD: mã hóa bản rõ $P=0101.1100$ (5C) với khóa $K=1001.1010$

$$L_0 = 0101, R_0 = 1100, KL_0 = 1001, KR_0 = 1010$$

- Vòng 1:

$$L_1 = R_0 = 1100, \text{Expand}(R_0) = 001011$$

$$KL_1 = KL_0 \ll 1 = 0011, KR_1 = KR_0 \ll 1 = 0101$$

$$K_1 = \text{Compress}(KL_1KR_1) = 101110$$

$$\text{Expand}(R_0) \oplus K_1 = 100101$$

$$S\text{-box}(100101) = 1000$$

$$F_1 = P\text{-box}(1000) = 0100$$

$$R_1 = L_0 \oplus F_1 = 0001$$

- Vòng 2:

$$L_2 = R_1 = 0001, \text{Expand}(R_1) = 010000$$

$$KL_2 = KL_1 \ll 2 = 1100, KR_2 = KR_1 \ll 2 = 0101$$

$$K_2 = \text{Compress}(KL_2KR_2) = 110011$$

$$\text{Expand}(R_1) \oplus K_2 = 100011$$

$$S\text{-box}(100011) = 1100$$

$$F_2 = P\text{-box}(1100) = 0101$$

$$R_2 = L_1 \oplus F_2 = 1001$$

- Vòng 3:

$$L_3 = R_2 = 1001, \text{Expand}(R_2) = 010001$$

$$KL_3 = KL_2 \ll 1 = 1001, KR_3 = KR_2 \ll 1 = 1010$$

$$K_3 = \text{Compress}(KL_3KR_3) = 001001$$

$$\text{Expand}(R_2) \oplus K_3 = 011000$$

$$S\text{-box}(011000) = 0101$$

$$F_3 = P\text{-box}(0101) = 0011$$

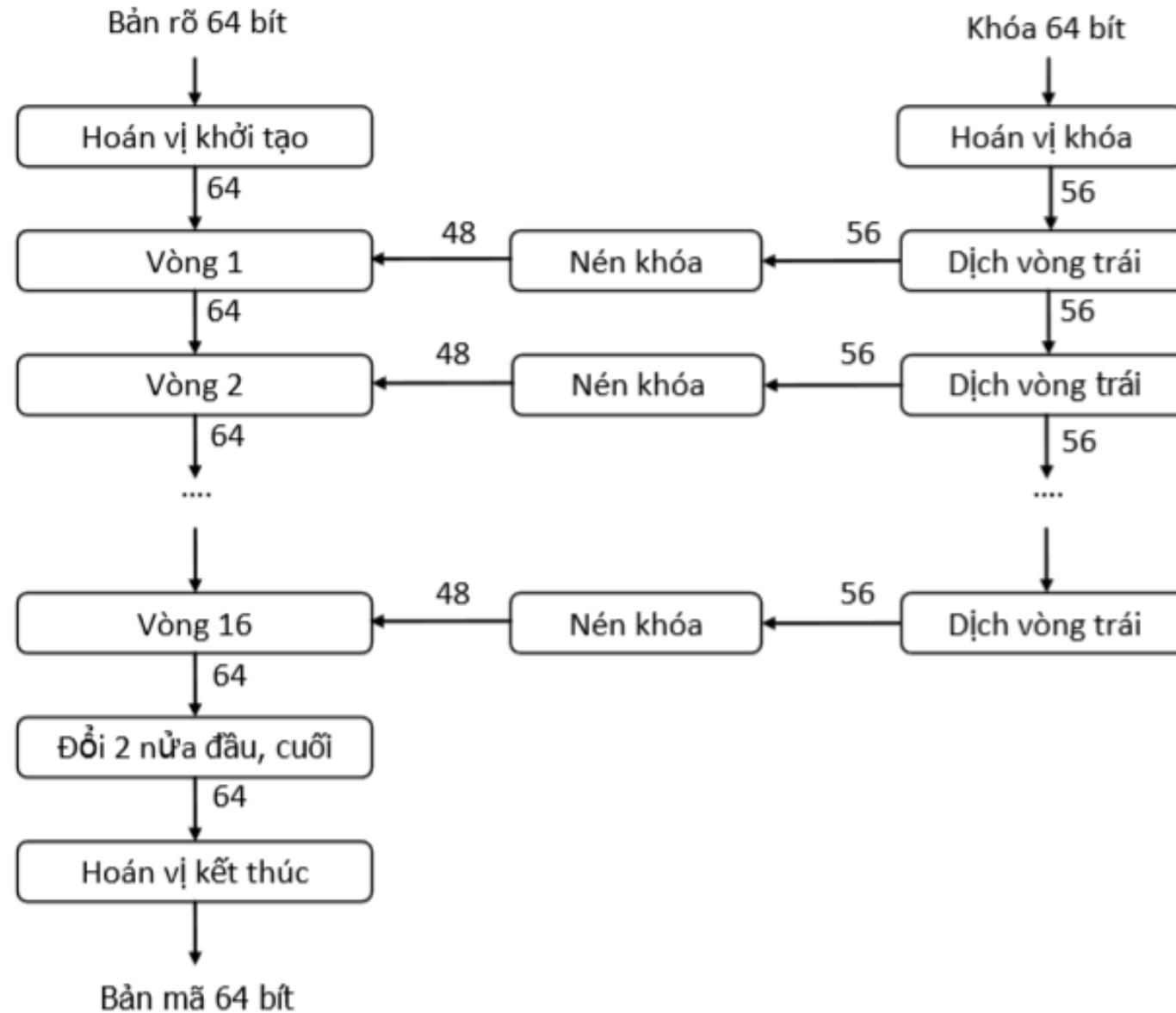
$$R_3 = L_2 \oplus F_3 = 0010$$

→ Kết quả $C = L_3R_3 = 1001.0010$ (hệ thập lục phân: 92)

Mã DES

- Là mã thuộc hệ mã Feistel gồm 16 vòng, ngoài ra DES có thêm một hoán vị khởi tạo trước khi vào vòng 1 và một hoán vị khởi tạo sau vòng 16.
- Kích thước của khối là 64 bit.
- Kích thước khóa là 56 bit.
- Mỗi vòng của DES dùng khóa con có kích thước 48 bit được trích ra từ khóa chính.

Minh họa các vòng của mã DES:



Sơ đồ mã DES trên gồm ba phần:

- Phần thứ nhất là các hoán vị khởi tạo và hoán vị kết thúc.
- Phần thứ hai là các vòng Feistel.
- Phần thứ ba là thuật toán sinh khóa con.

1. Hoán vị khởi tạo và hoán vị kết thúc:

- Đánh số các bit của khối 64 bit theo thứ tự:

0, 1, 2, 3,...63: **b₀b₁b₂b₃...b₆₃**

- Hoán vị khởi tạo sẽ hoán vị các bit theo nguyên tắc sau:

57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7
56	48	40	32	24	16	8	0
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6

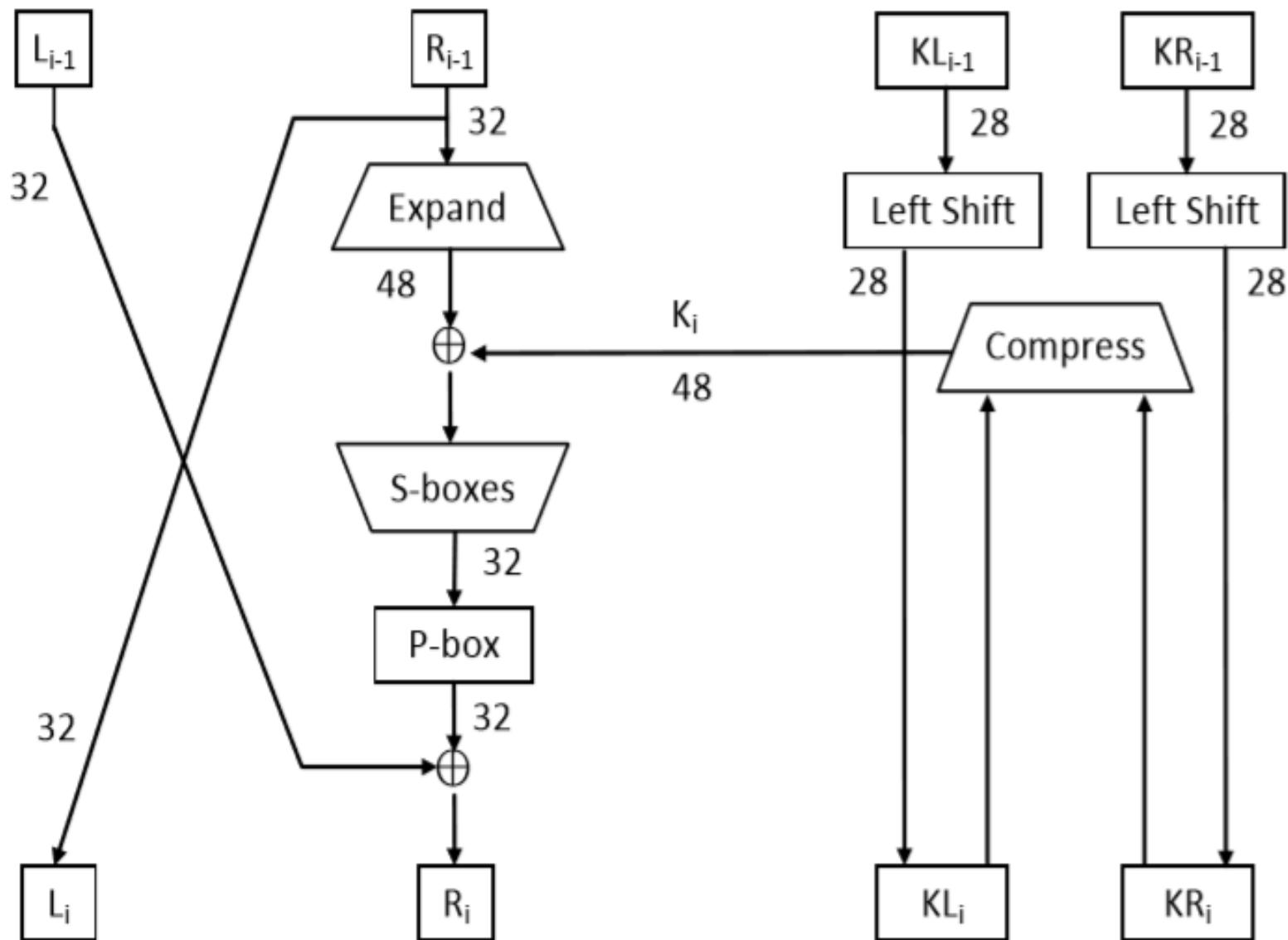
(b₀b₁b₂ ... b₆₂b₆₃ → b₅₇b₄₉b₄₁ ... b₁₄b₆)

Hoán vị kết thúc hoán đổi các bit theo nguyên tắc sau:

39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25
32	0	40	8	48	16	56	24

Hoán vị kết thúc chính là hoán vị nghịch đảo của hoán vị khởi tạo.

2. Các vòng của DES:

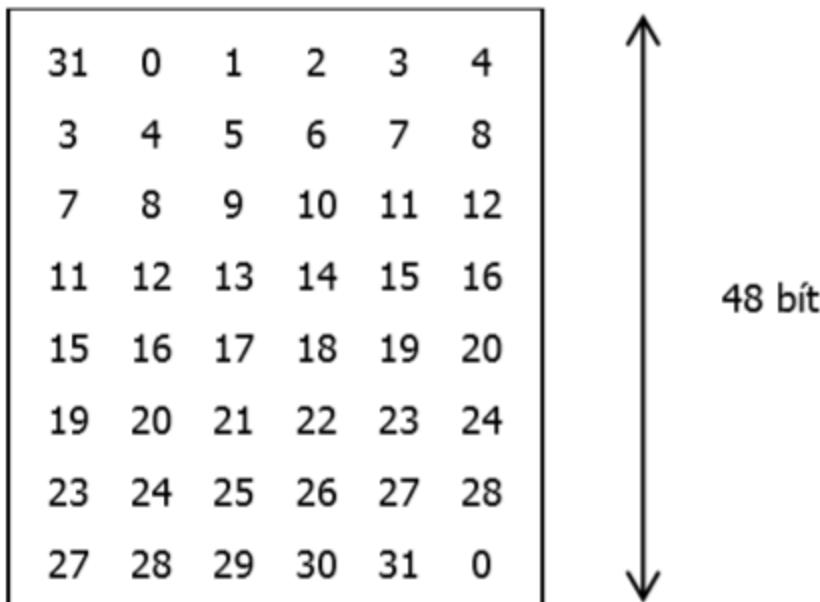


Trong DES, hàm F của Feistel là:

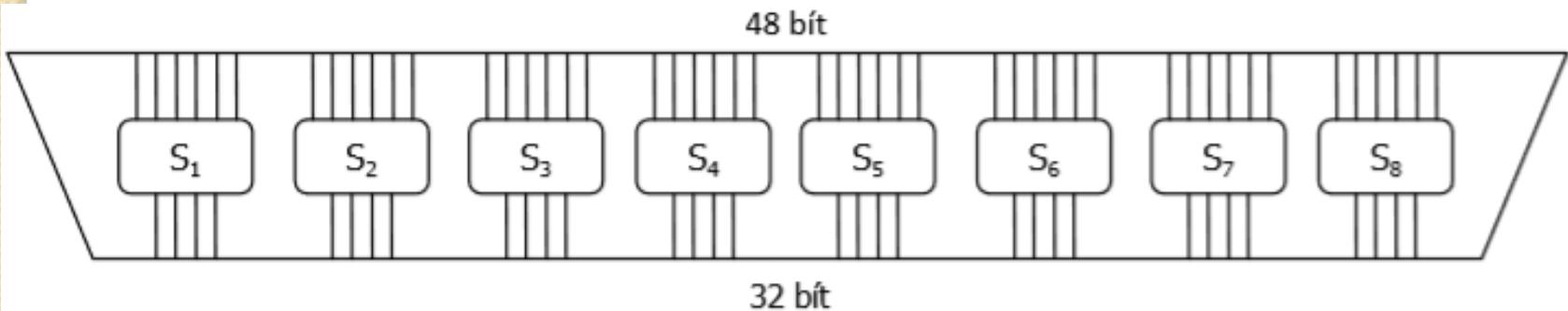
$$F(R_{i-1}, K_i) = P\text{-box}(S\text{-boxes}(Expand(R_{i-1}) \oplus K_i))$$

- Expand vừa mở rộng vừa hoán vị R_{i-1} từ 32 bit lên 48 bit. Hàm Sboxes nén 48 bit lại còn 32 bit.
- P-box là một hoán vị 32 bit.

Expand: đánh số các bít của R_{i-1} theo thứ tự từ trái sang phải là 0, 1, 2, ..., 31. Hàm Expand thực hiện vừa hoán vị vừa mở rộng 32 bit thành 48 bit theo quy tắc:



- Hàm S-boxes của DES biến đổi một số 48 bít thành một số 32 bít. Tuy nhiên, nếu chỉ lập một bảng tra cứu như ở TinyDES thì bảng này phải có 216 dòng và 232 cột, dẫn đến số phần tử của bảng rất lớn. Để giảm kích thước của bảng tra cứu, chia hàm S-boxes thành 8 hàm S-box con, mỗi hàm biến đổi số 6 bít thành số 4 bít



$b_1 b_2 b_3 b_4$

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
		0	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7
$b_0 b_5$	0	0	F	7	4	E	2	D	1	A	6	C	B	9	5	3	8	
	1	4	1	E	8	D	6	2	B	F	C	9	7	3	A	5	0	
	2	F	C	8	2	4	9	1	7	5	B	3	E	A	0	6	D	
	3																	

DES S-box 1

 $b_1 b_2 b_3 b_4$

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
		0	F	1	8	E	6	B	3	4	9	7	2	D	C	0	5	A
$b_0 b_5$	0	3	D	4	7	F	2	8	E	C	0	1	A	6	9	B	5	
	1	0	E	7	B	A	4	D	1	5	8	C	6	9	3	2	F	
	2	D	8	A	1	3	F	4	2	B	6	7	C	0	5	E	9	
	3																	

DES S-box 2

$b_1 b_2 b_3 b_4$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	A	0	9	E	6	3	F	5	1	D	C	7	B	4	2	8	
$b_0 b_5$	1	D	7	0	9	3	4	6	A	2	8	5	E	C	B	F	1
2	D	6	4	9	8	F	3	0	B	1	2	C	5	A	E	7	
3	1	A	D	0	6	9	8	7	4	F	E	3	B	5	2	C	

DES S-box 3

 $b_1 b_2 b_3 b_4$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	7	D	E	3	0	6	9	A	1	2	8	5	B	C	4	F	
$b_0 b_5$	1	D	8	B	5	6	F	0	3	4	7	2	C	1	A	E	9
2	A	6	9	0	C	B	7	D	F	1	3	E	5	2	8	4	
3	3	F	0	6	A	1	D	8	9	4	5	B	C	7	2	E	

DES S-box 4

		$b_1 b_2 b_3 b_4$															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$b_0 b_5$	0	2	C	4	1	7	A	B	6	8	5	3	F	D	0	E	9
	1	E	B	2	C	4	7	D	1	5	0	F	A	3	9	8	6
	2	4	2	1	B	A	D	7	8	F	9	C	5	6	3	0	E
	3	B	8	C	7	1	E	2	D	6	F	0	9	A	4	5	3

DES S-box 5

		$b_1 b_2 b_3 b_4$															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$b_0 b_5$	0	C	1	A	F	9	2	6	8	0	D	3	4	E	7	5	B
	1	A	F	4	2	7	C	9	5	6	1	D	E	0	B	3	8
	2	9	E	F	5	2	8	C	3	7	0	4	A	1	D	B	6
	3	4	3	2	C	9	5	F	A	B	E	1	7	6	0	8	D

DES S-box 6

	$b_1 b_2 b_3 b_4$																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
$b_0 b_5$	0	4	B	2	E	F	0	8	D	3	C	9	7	5	A	6	1
	1	D	0	B	7	4	9	1	A	E	3	5	C	2	F	8	6
	2	1	4	B	D	C	3	7	E	A	F	6	8	0	5	9	2
	3	6	B	D	8	1	4	A	7	9	5	0	F	E	2	3	C

DES S-box 7

	$b_1 b_2 b_3 b_4$																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
$b_0 b_5$	0	D	2	8	4	6	F	B	1	A	9	3	E	5	0	C	7
	1	1	F	D	8	A	3	7	4	C	5	6	B	0	E	9	2
	2	7	B	4	1	9	C	E	2	0	6	A	D	F	3	5	8
	3	2	1	E	7	4	A	8	D	F	C	9	0	3	5	6	B

DES S-box 8

P-box: hàm P-box cũng thực hiện hoán vị 32 bit đầu vào theo quy tắc:

15	6	19	20	28	11	27	16
0	14	22	25	4	17	30	9
1	7	23	13	31	26	2	8
18	12	29	5	21	10	3	24

Khóa K 64 bit ban đầu được rút trích và hoán vị thành một khóa 56 bit (tức chỉ sử dụng 56 bít) theo quy tắc:

56	48	40	32	24	16	8
0	57	49	41	33	25	17
9	1	58	50	42	34	26
18	10	2	59	51	43	35
62	54	46	38	30	22	14
6	61	53	45	37	29	21
13	5	60	52	44	36	28
20	12	4	27	19	11	3

56 bít

- Khóa 56 bit này được chia thành 2 nửa trái phải KL_0 và KR_0 , mỗi nửa có kích thước 28 bit. Tại vòng thứ i ($i = 1, 2, 3, \dots, 16$), KL_{i-1} và KR_{i-1} được dịch vòng trái r_i bit để có được KL_i và KR_i , với r_i được định nghĩa:

$$r_i = \begin{cases} 1 & \text{nếu } i \in \{1, 2, 9, 16\} \\ 2 & \text{với những } i \text{ khác} \end{cases}$$

- Cuối cùng khóa K_i của mỗi vòng được tạo ra bằng cách hoán vị và nén 56 bit của KL_i và KR_i thành 48 bit theo quy tắc:

13	16	10	23	0	4	2	27
14	5	20	9	22	18	11	3
25	7	15	6	26	19	12	1
40	51	30	36	46	54	29	39
50	44	32	47	43	48	38	55
33	52	45	41	49	35	28	31

48 bit

MH: DES (Data Encryption Standard)

- Mã hóa từng khối dữ liệu có độ dài 64bit với khóa có độ dài 56bit.
- Đầu vào gồm thông điệp cần mã hóa và khóa dùng để mã hóa.
- Mỗi khối dữ liệu 64bit được thực hiện qua 16 vòng lặp để cho ra 1 khối 64bit đã được mã hóa.
- Thuật toán DES được sử dụng trong một thời gian dài và được xem như một thuật toán mã hóa đảm bảo an toàn.

Giải mã:

- Có thể sử dụng thuật toán mã hóa DES để giải mã.
- Các bước thực hiện giống quá trình mã hóa.
- Các khóa phụ trong các vòng lặp được sử dụng theo trật tự ngược lại: khóa phụ 16, 15,..., 2, 1 cho các vòng 1, 2,..., 15, 16 tương ứng.

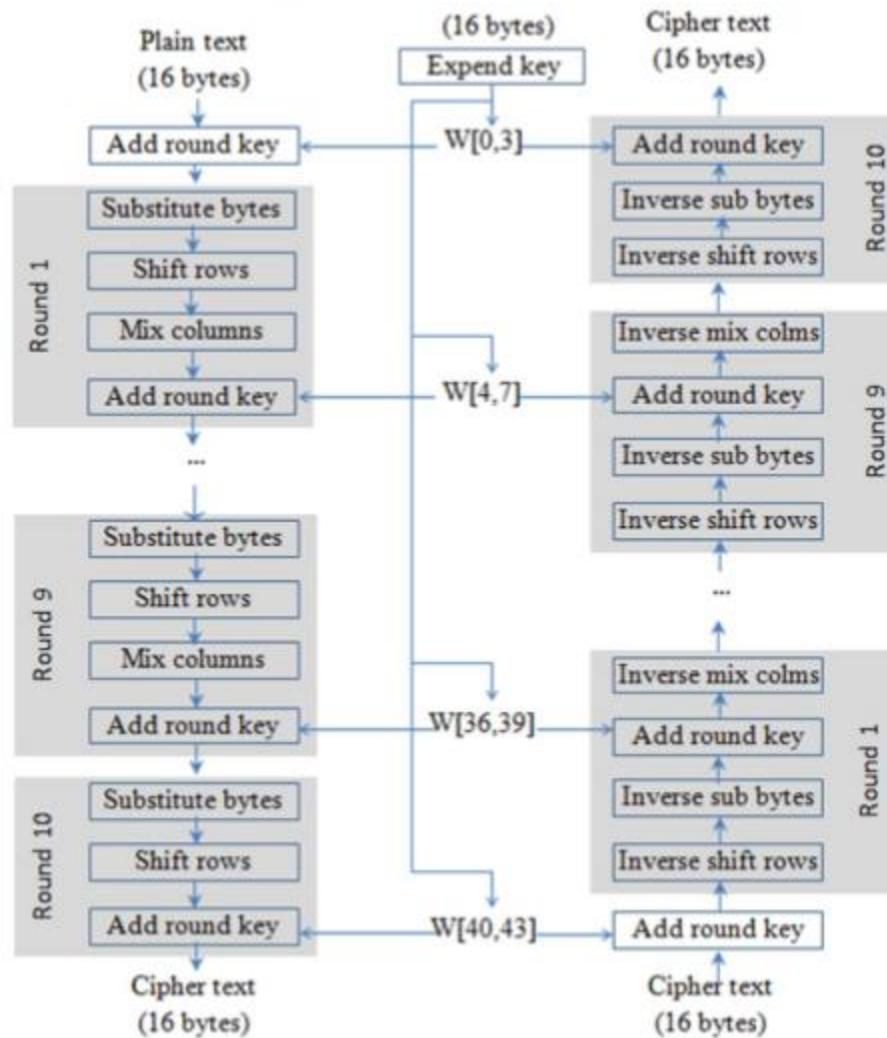
MH: AES

- Là thuật toán mã hóa nâng cao được thiết kế cho các ứng dụng thương mại, có nhiều cấu trúc khác nhau.
- MH mỗi lần 1 khối dl 128 bit qua 10 vòng MH được gọi là AES-128.
- MH mỗi lần 1 khối dữ liệu 192 bit qua 12 vòng MH được gọi là AES-192.
- MH mỗi lần 1 khối dl 256bit qua 14vòng MH được gọi là AES-256.

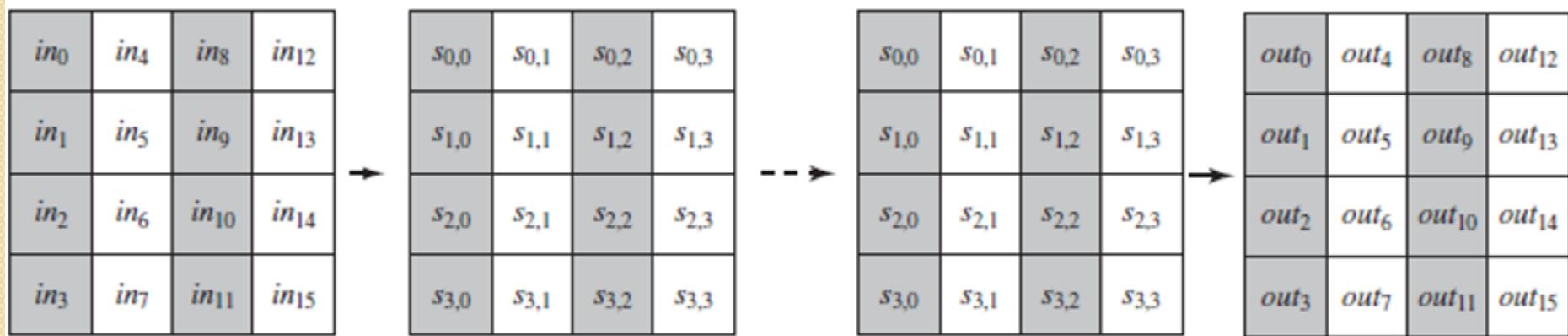
Trường hợp: AES-128

- Mỗi khối 128bit được xem như một ma trận 4×4 của các bytes.
- Thuật toán mã hóa AES thực hiện 4 phép toán:
 - Thay thế giá trị khóa.
 - Dịch bit.
 - Đảo cột.
 - Cộng khóa cho từng vòng mã hóa.

- Mã hóa mỗi lần một khối dữ liệu 128bit qua 10 vòng mã hóa.
- Chiều dài của khóa là 16byte.



- 16 byte input được copy vào ma trận State
- Tất cả các thao tác đều được thực hiện trên ma trận State.
- Cuối cùng copy nội dung của ma trận State ra output.



- $\text{State}[r,c] = \text{input}[r+4c]$ for $0 \leq r < 4, 0 \leq c < 4$
- $\text{output}[r+4c] = \text{State}[r,c]$ for $0 \leq r < 4, 0 \leq c < 4$

- Trong trường hợp nội dung thông điệp cần mã hóa dài hơn chiều dài của khối, thông điệp sẽ chia thành m khối, các khối này sẽ được mã hóa với cùng một khóa.
- Trong thực tế, khi áp dụng thuật toán AES vào các chương trình ứng dụng được cài đặt theo 5 kiểu khác nhau: Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), Counter(CTR)

MH BẤT ĐỐI XỨNG: RSA

- RSA là thuật toán MH với khóa công khai được đặt tên theo 3 tác giả: Ron Rivest, Adi Shamir, Leonard Adleman.
- RSA là pp mã hóa theo khối. Trong đó bản rõ M và bản mã C là các số nguyên từ 0 đến 2^i với i là số bit của khối (kích thước thường dùng của i là 1024bit).
- RSA sử dụng hàm 1 chiều là vấn đề phân tích 1 số thành thừa số nguyên tố.

Để thực hiện MH và giải mã, RSA dùng phép lũy thừa Modulo của lý thuyết số. Các bước thực hiện như sau:

1. Chọn 2 số nguyên tố lớn p và q, tính $N=pq$. Cần chọn p và q sao cho: $M < 2^{i-1} < N < 2^i$.

Với $i = 1024$ thì N là số nguyên dài 309 chữ số.

2. Tính $n=(p-1)(q-1)$

3. Tìm 1 số e sao cho e là nguyên tố cùng nhau với n.

4. Tìm 1 số d sao cho $e \cdot d \equiv 1 \pmod{n}$ (d là nghịch đảo của e trong phép modulo n)

5. Hủy bỏ n, p và q. Chọn khóa công khai K_u là cặp (e, N) , khóa riêng K_R là cặp (d, N)

6. Việc mã hóa thực hiện theo công thức:

- PP1, MH bảo mật: $C = E(M, K_U) = M^e \text{ mod } N$
- PP2, MH chứng thực:
 $C = E(M, K_R) = M^d \text{ mod } N$

7. Việc giải mã thực hiện theo công thức:

- PP1, MH bảo mật: $\bar{M} = D(C, K_R) = C^d \text{ mod } N$
- PP2, MH chứng thực: $\bar{M} = D(C, K_U) = C^e \text{ mod } N$

Bản rõ M có kích thước $i-1$ bit, bản mã C có kích thước i bit.

VD MH RSA với kích thước khóa là 6 bit

1. Chọn $p = 11$ và $q = 3$, do đó $N = pq = 33$
 $(25 = 32 < 33 < 64 = 26)$
2. $n = (p-1)(q-1) = 20$
3. Chọn $e = 3$ nguyên tố cùng nhau với n
4. Tính nghịch đảo của e trong phép modulo n được $d = 7$ ($3 \times 7 = 21$)
5. Khóa công khai $K_U = (e, N) = (3, 33)$.
Khóa bí mật $K_R = (d, N) = (7, 33)$

Mã hóa theo PP1: MH bảo mật

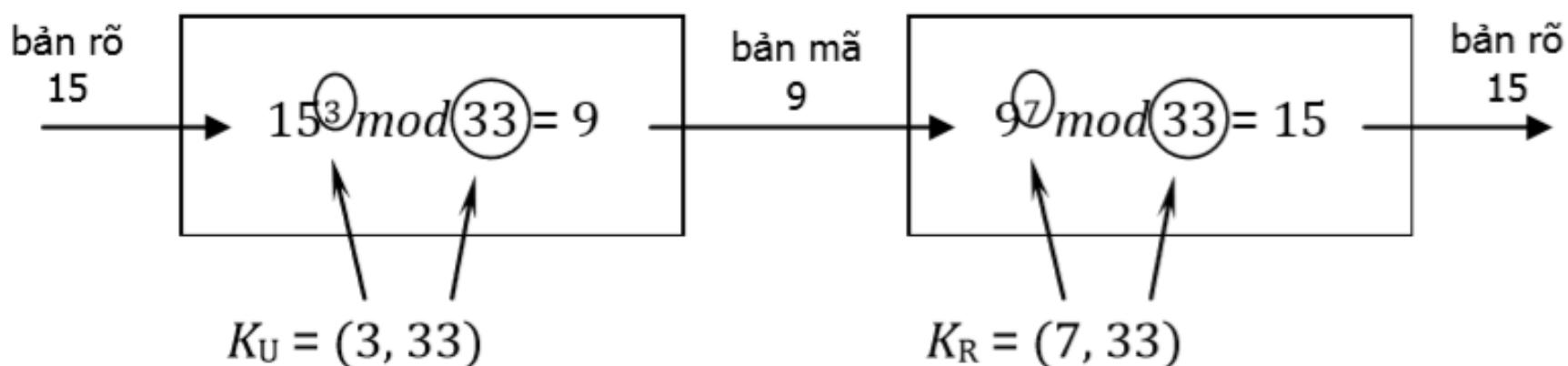
6. MH bản rõ $M = 15$:

$$C = M^e \bmod N = 15^3 \bmod 33 = 9$$

$$(vì 15^3 = 3375 = 102 \times 33 + 9)$$

7. Giải mã với bản mã là 9:

$$\bar{M} = C^d \bmod N = 9^7 \bmod 33 = 15 = M \quad (vì 9^7 = 4.782.696 = 144.938 \times 33 + 15)$$



MH theo PP2: MH chứng thực

6. Mã hóa bản rõ $M = 15$:

$$C = M^d \bmod N = 15^7 \bmod 33 = 27$$

(vì $15^7 = 170.859.375 = 5177.556 \times 33 + 27$)

7. Giải mã bản mã: $C=9$

$$\bar{M} = C^e \bmod N = 27^3 \bmod 33 = 15 = M$$

(vì $27^3 = 19.683 = 596 \times 33 + 15$)

CÁC HÀM BĂM

Hàm băm (hash function) là một hàm toán học h có tối thiểu 2 thuộc tính.

- Nén (compression): h là một ánh xạ từ chuỗi đầu vào x có chiều dài bất kỳ sang một chuỗi đầu ra $h(x)$ có chiều dài cố định n bit.
- Dễ tính toán (ease of computation): cho trước hàm h và đầu vào x, việc tính toán $h(x)$ là dễ dàng.

Phân loại hàm băm theo khóa sử dụng:

- Hàm băm không khóa (unkeyed): đầu vào chỉ là thông điệp.
- Hàm băm có khóa (keyed): đầu vào gồm thông điệp và khóa.

Phân loại hàm băm theo tính năng:

- Mã xác thực thông điệp (MAC - Message authentication codes)
 - MAC cũng được dùng để đảm bảo tính toàn vẹn của thông điệp mà không cần một biện pháp bổ sung khác;
 - MAC là loại hàm băm có khóa: đầu vào là thông điệp và một khóa.

Một số giải thuật hàm băm điển hình:

- CRC (Cyclic redundancy checks)
- Checksums
- MD2, MD4, MD5
- MD6
- SHA0, SHA1
- SHA2, SHA3

Hàm băm: MD5

- MD5 (Message Digest) là hàm băm không khóa được Ronald Rivest thiết kế năm 1991 để thay thế MD4;
- Chuỗi đầu ra (giá trị băm) của MD5 là 128 bít (16 bytes) và thường được biểu diễn thành 32 số hexa;
- MD5 được sử dụng khá rộng rãi trong nhiều ứng dụng:
 - Chuỗi đảm bảo tính toàn vẹn thông điệp;
 - Tạo chuỗi kiểm tra lỗi – Checksum;
 - Mã hóa mật khẩu.

Quá trình xử lý thông điệp của MD5:

- Thông điệp được chia thành các khối 512 bít. Nếu kích thước thông điệp không là bội số của 512 → nối thêm số bít thiêu;
- Phần xử lý chính của MD5 làm việc trên state 128 bít, chia thành 4 từ 32 bít (A, B, C, D);
 - Các từ A, B, C, D được khởi trị bằng một hằng cố định;
 - Từng phần 32 bít của khối đầu vào 512 bít được đưa dần vào để thay đổi state;
- Quá trình xử lý gồm 4 vòng, mỗi vòng gồm 16 thao tác tương tự nhau;
- Mỗi thao tác gồm:
 - Hàm F (4 hàm khác nhau cho mỗi vòng);
 - Cộng modulo;
 - Quay trái.

Lưu đồ xử lý một thao tác của MD5:

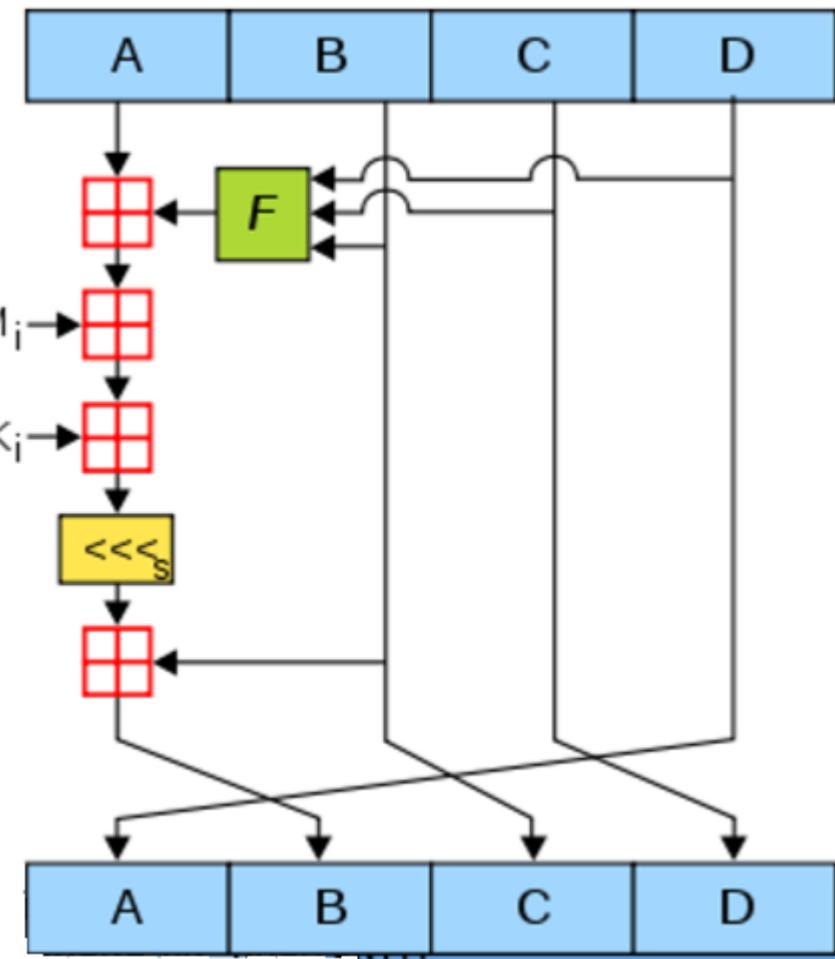
- A, B, C, D: các từ 32 bit
- M_i : khối 32 bit thông điệp đầu vào;
- K_i : 32 bit hằng. Mỗi sử dụng một hằng khác nhau;
- $<<<_S$: thao tác dịch trái S bit
-  biểu diễn cộng modulo 32 bít;
- F: hàm phi tuyến tính, gồm 4 loại:

$$F(B, C, D) = (B \wedge C) \vee (\neg B \wedge D)$$

$$G(B, C, D) = (B \wedge D) \vee (C \wedge \neg D)$$

$$H(B, C, D) = B \oplus C \oplus D$$

$$I(B, C, D) = C \oplus (B \vee \neg D)$$



Hàm băm: SHA1

- SHA1 (Secure Hash Function) được NSA (Mỹ) thiết kế năm 1995 để thay thế cho SHA0;
- Chuỗi đầu ra của SHA1 có kích thước 160 bít và thường được biểu diễn thành 40 số hexa;
- Họ hàm băm SHA: SHA-0, SHA-1, SHA-2, SHA-3:
 - SHA0 ít được sử dụng trên thực tế;
 - SHA1 tương tự SHA0, nhưng đã khắc phục một số lỗi;
 - SHA2 ra đời năm 2001 khắc phục lỗi của SHA1 và có nhiều thay đổi. Kích thước chuỗi đầu ra có thể là 224, 256, 384 và 512 bít;
 - SHA3 ra đời năm 2012, cho phép chuỗi đầu ra có kích thước không cố định.
- SHA1 được sử dụng rộng rãi để đảm bảo tính xác thực và toàn vẹn thông điệp.

- Quá trình xử lý thông điệp của SHA1:
 - SHA1 sử dụng thủ tục xử lý thông điệp tương tự MD5
 - Thông điệp được chia thành các khối 512 bít. Nếu kích thước thông điệp không là bội số của 512 → nối thêm số bít thiêu;
 - Phần xử lý chính của SHA1 làm việc trên state 160 bít, chia thành 5 từ 32 bít (A, B, C, D, E);
 - Các từ A, B, C, D, E được khởi trị bằng một hằng cố định;
 - Từng phần 32 bít của khối đầu vào 512 bít được đưa dần vào để thay đổi state;
 - Quá trình xử lý gồm 80 vòng, mỗi vòng gồm các thao tác: add, and, or, xor, rotate, mod.

CHỮ KÝ SỐ

- Khái niệm
- Quá trình ký và kiểm tra chữ ký số
- Thuật toán chữ ký số RSA
- Thuật toán chữ ký số DSA

CHỮ KÝ SỐ

- Chữ ký số (Digital Signature) là một chuỗi dữ liệu liên kết với một thông điệp (message) và thực thể tạo ra thông điệp;
- Giải thuật tạo chữ ký số (Digital Signature generation algorithm) là một phương pháp sinh chữ ký số;
- Giải thuật kiểm tra chữ ký số (Digital Signature verification algorithm) là một phương pháp xác minh tính xác thực của chữ ký số, có nghĩa là nó thực sự được tạo ra bởi 1 bên chỉ định;
- Một hệ chữ ký số (Digital Signature Scheme) bao gồm giải thuật tạo chữ ký số và giải thuật kiểm tra chữ ký số.

- Quá trình tạo chữ ký số (Digital signature signing process) bao gồm:
 - Giải thuật tạo chữ ký số
 - Phương pháp chuyển dữ liệu thông điệp thành dạng có thể ký được.
- Quá trình kiểm tra chữ ký số (Digital signature verification process) bao gồm:
 - Giải thuật kiểm tra chữ ký số
 - Phương pháp khôi phục dữ liệu từ thông điệp.

CHỮ KÝ SỐ - QUÁ TRÌNH KÝ

Các bước của quá trình ký một thông điệp (bên người gửi):

- Tính toán chuỗi đại diện (message digest/hash value) của thông điệp sử dụng một giải thuật băm (Hashing algorithm);
- Chuỗi đại diện được ký sử dụng khóa riêng (Private key) của người gửi và một giải thuật tạo chữ ký (Signature/Encryption algorithm). Kết quả là chữ ký số (Digital signature) của thông điệp hay còn gọi là chuỗi đại diện được mã hóa (Encrypted message digest);
- Thông điệp ban đầu (message) được ghép với chữ ký số (Digital signature) tạo thành thông điệp đã được ký (Signed message);
- Thông điệp đã được ký (Signed message) được gửi cho người nhận.

CHỮ KÝ SỐ - QUÁ TRÌNH KIỂM TRA

Các bước của quá trình kiểm tra chữ ký (bên người nhận):

- Tách chữ ký số và thông điệp gốc khỏi thông điệp đã ký để xử lý riêng;
- Tính toán chuỗi đại diện MD1 (message digest) của thông điệp gốc sử dụng giải thuật băm (là giải thuật sử dụng trong quá trình ký);
- Sử dụng khóa công khai (Public key) của người gửi để giải mã chữ ký số ☐ chuỗi đại diện thông điệp MD2;
- So sánh MD1 và MD2:
 - Nếu $MD1 = MD2$ ☐ chữ ký kiểm tra thành công. Thông điệp đảm bảo tính toàn vẹn và thực sự xuất phát từ người gửi (do khóa công khai được chứng thực).
 - Nếu $MD1 \neq MD2$ ☐ chữ ký không hợp lệ. Thông điệp có thể đã bị sửa đổi hoặc không thực sự xuất phát từ người gửi.

Chữ ký số - Giải thuật chữ ký số RSA

RSA là giải thuật cho phép thực hiện 2 tính năng:

- Mã hóa thông điệp:
 - Người gửi mã hóa thông điệp sử dụng khóa công khai của người nhận;
 - Người nhận giải mã thông điệp sử dụng khóa riêng của mình.
- Tạo chữ ký số:
 - Người gửi tạo chữ ký số sử dụng khóa bí mật của mình;
 - Người nhận kiểm tra chữ ký sử dụng khóa công khai của người gửi.

Chữ ký số - Giải thuật chữ ký số DSA

- DSA (Digital Signature Algorithm) là chuẩn chữ ký số được phát triển bởi NIST (Mỹ) năm 1991;
- DSA được phát triển từ giải thuật Digital Signature Standard (DSS);
- Các thành phần của DSA:
 - Sinh khóa: sinh cặp khóa. Gồm 2 giai đoạn:
 - Lựa chọn tham số của giải thuật;
 - Sinh cặp khóa cho người dùng.
 - Quá trình ký: ký thông điệp
 - Quá trình kiểm tra chữ ký: kiểm tra chữ ký.

Chữ ký số - Giải thuật chữ ký số DSA

Sinh khóa:

- Lựa chọn tham số:
 - Lựa chọn giải thuật băm chuẩn H. Giải thuật băm có thể được lựa chọn là SHA-1 hoặc SHA-2;
 - Chọn kích thước cho các khóa L và N.
 - L có thể là 1024, 2048, 3072;
 - N có thể là 160, 224, 256. N phải nhỏ hơn hoặc bằng kích thước chuỗi băm đầu ra của hàm H đã chọn;
 - Chọn số nguyên tố q N bít;
 - Chọn modulo p L bít sao cho $p-1$ là bội số của q;
 - Chọn g là hệ số nhân sao cho $(g^*q) \bmod p = 1$;
 - Các tham số (q, p và g) được chia sẻ giữa các người dùng.

Chữ ký số - Giải thuật chữ ký số DSA

Sinh khóa:

- Sinh khóa cho một người dùng:
 - Chọn số ngẫu nhiên x sao cho $0 < x < q$;
 - Tính $y = g^x \text{ mod } p$;
 - Khóa công khai là (q, p, g, y) ;
 - Khóa riêng là x .

Chữ ký số - Giải thuật chữ ký số DSA

Ký thông điệp:

- H là hàm băm sử dụng và m là thông điệp gốc.
- Tính $H(m)$ từ thông điệp gốc
- Tạo số ngẫu nhiên k cho mỗi thông điệp, $0 < k < q$;
- Tính $r = (g^k \text{ mod } p) \text{ mod } q$;
- Nếu $r = 0$, chọn một k mới và tính lại r ;
- Tính $s = k^{-1}(H(m) + xr) \text{ mod } q$;
- Nếu $s = 0$, chọn một k mới và tính lại r và s ;
- Chữ ký là cặp (r, s) .

Kiểm tra chữ ký của thông điệp:

- Loại bỏ chữ ký nếu r và s không thỏa mãn $0 < r, s < q$;
- Tính $H(m)$ từ thông điệp nhận được;
- Tính $w = s^{-1} \text{ mod } q$;
- Tính $u_1 = H(m) * w \text{ mod } q$;
- Tính $u_2 = r * w \text{ mod } q$;
- Tính $v = ((g^{u_1} * y^{u_2}) \text{ mod } p) \text{ mod } q$;
- Chữ ký là xác thực nếu $v = r$.

4. Kiểm toán và giải trình

1. Giới thiệu

Kiểm toán (Audit): giám sát và ghi lại những hoạt động đã và đang xảy ra trong hệ thống một cách có chọn lọc.

Giải trình (Accountability): trách nhiệm tìm ra và chứng minh nguồn gốc các hoạt động xảy ra trong hệ thống.

Hoạt động kiểm toán nhằm phục vụ cho hoạt động giải trình.

4. Kiểm toán và giải trình

Tại sao phải kiểm toán?

- **Trách nhiệm giải trình** từ những hoạt động xảy ra lên dữ liệu (schema, bảng, dòng,...)
- **Kiểm tra hành động đáng ngờ:** ví dụ xóa dữ liệu từ một bảng.
- Thông báo nếu có nếu người dùng không được ủy quyền nhưng lại thao tác trên dữ liệu mà đòi hỏi phải có đủ quyền truy cập (truy cập vượt quyền).

4. Kiểm toán và giải trình

Tại sao phải kiểm toán?

- Giám sát và ghi lại các hoạt động xảy ra nhằm **phát hiện các vấn đề** trong quá trình định quyền và điều khiển truy cập.
- Thống kê tình hình truy xuất tài nguyên để có biện pháp **cải thiện hiệu xuất**.

Ví dụ: dựa vào các trường, bảng thường hay được truy cập → chọn cách đánh chỉ mục thích hợp để tăng hiệu suất.

- Kiểm toán để thỏa các yêu cầu chính sách pháp lý: thể hiện trách nhiệm với dữ liệu của khách hàng.

4. Kiểm toán và giải trình

Khi nào cần kiểm toán?

Kiểm toán tại mọi thời điểm khi hệ thống bắt đầu hoạt động.

Nội dung kiểm toán:

Việc kiểm toán có thể làm giảm hiệu suất của hệ thống → chỉ nên kiểm toán những gì cần thiết.

4. Kiểm toán và giải trình

Những hoạt động cần kiểm toán:

- Hoạt động của những người dùng có quyền.
- Đăng nhập và đăng xuất.
- Những thay đổi trong các Application Trigger và Data Trigger.
- Thay đổi quyền và mô tả thông tin người dùng.
- Cấu trúc dữ liệu bị thay đổi.
- Các truy cập đọc và ghi trên những dữ liệu nhạy cảm.
- Những lỗi và ngoại lệ.
- Nguồn gốc của những hoạt động truy cập dữ liệu.
- Thời gian, tên chương trình, kích thước dữ liệu, câu lệnh,...

4. Kiểm toán và giải trình

Quy trình kiểm toán:

1. Phân tích các yêu cầu bảo mật của ứng dụng.
2. Chọn các sự kiện/hoạt động/đối tượng sẽ kiểm toán.
3. Giám sát và ghi nhận.
4. Lưu trữ Audit log (nhật ký kiểm toán)
5. Kiểm tra và phân tích Audit log.
6. Phản hồi.

4. Kiểm toán và giải trình

Các vấn đề với kiểm toán:

- Kiểm toán là công cụ, không phải mục tiêu.
- Nên sử dụng kết hợp giữa kiểm toán bên trong và kiểm toán bên ngoài.
- Lưu trữ và bảo mật thông tin Audit log.
- Tự động hóa và giám sát hoạt động kiểm toán.
- Kích thước của các Audit log lớn, cần sử dụng các công cụ kho dữ liệu, và khai phá dữ liệu để quản lý và phân tích dữ liệu Audit log.
- Vấn đề tính riêng tư trong Audit log.

4. Kiểm toán và giải trình

2. Kỹ thuật kiểm toán và giải trình trong CSDL

**Các yêu cầu của kiểm toán trong
CSDL:**

- Có khả năng hoạt động độc lập, cho phép theo dõi và ghi nhận lại tất cả các hoạt động trong hệ thống kể cả những hoạt động của người quản trị hệ thống.
- Có khả năng lưu trữ Audit log một cách an toàn bên ngoài CSDL.
- Có khả năng thu thập và kết hợp các hoạt động xảy ra ở nhiều loại hệ quản trị CSDL khác nhau.

4. Kiểm toán và giải trình

Các yêu cầu của kiểm toán trong CSDL:

- Có khả năng ngăn chặn người quản trị hệ thống chỉnh sửa hoặc xóa dữ liệu trong Audit log.
- Có khả năng đưa ra những cảnh báo kịp thời cho người quản trị hệ thống khi có những bất thường xảy ra trong hệ thống.

Các phương pháp kiểm toán:

- Kiểm toán bằng Application Server log.
- Kiểm toán mức độ ứng dụng (Application Audit).
- Kiểm toán bằng Trigger.
- Kiểm toán bằng câu lệnh (Command).

Các đối tượng cần kiểm toán:

- Các hoạt động đăng nhập/đăng xuất CSDL.
 - + Username.
 - + Client IP mà đăng nhập không thành công.
 - + Chương trình (Source Program).
 - + Thời gian đăng nhập và đăng xuất.
- Kiểm toán nguồn gốc truy cập CSDL.
 - + Địa chỉ IP và host name được dùng để kết nối CSDL.
 - + Chương trình nào kết nối CSDL.

Các đối tượng cần kiểm toán:

- Kiểm toán các hoạt động truy cập CSDL ngoài giờ làm việc:
 - + Các truy cập CSDL ngoài giờ làm việc đáng nghi ngờ.
 - + Cần ghi lại mọi truy cập CSDL ngoài giờ làm việc trừ những thao tác đã được định giờ.
- Kiểm toán các thao tác thuộc ngôn ngữ định nghĩa dữ liệu.

Các đối tượng cần kiểm toán:

- Các lỗi thao tác với CSDL:
 - + Các lỗi: đăng nhập thất bại, SQL Injection → dấu hiệu của sự tấn công.
 - + Các ứng dụng có thể chứa lỗi và gây ra lỗi. VD: sinh ra những câu SQL sai cú pháp.
- Kiểm toán trên sự thay đổi mã nguồn của Trigger và Stored Procedure.
 - + Kẻ tấn công có thể giấu những đoạn mã độc hại vào Trigger hoặc Stored Procedure.

Các đối tượng cần kiểm toán:

- Kiểm toán sự thay đổi của các dữ liệu nhạy cảm.
- Kiểm toán sự thay đổi của các Audit log.
 - + Audit log cần được bảo vệ và không cho phép thay đổi.
 - + Phương pháp: sử dụng các chức năng Built-in của CSDL hoặc một hệ thống bên ngoài khác.