Chương 5:

Xây dựng các cơ chế bảo vệ dữ liệu - Thiết lập hệ thống tập tin chuyên biệt

5.1. XÂY DỰNG CÁC CƠ CHẾ BẢO VỆ DỮ LIỆU

5.1.1. Sử dụng các hỗ trợ có sẵn

- Các hỗ trợ đăng nhập hệ thống (trên các cấp hệ điều hành /firmware, máy tính, đĩa cứng,…)
- Các hỗ trợ mã hóa dữ liệu (chuyển dữ liệu vào các file có mã hóa và kiểm tra mật khẩu)
- Các tường lửa và phần mềm chống virus

5.1.2. Mã hóa với khóa đối xứng /bất đối xứng

* Mã hóa với khóa đối xứng (không công khai):

Phương pháp mã hóa là đối xứng khi các khóa để mã hóa và giải mã có quan hệ rõ ràng với nhau (có thể dễ dàng tìm được một khóa nếu biết khóa kia). Loại khóa này gọi là khóa không công khai. Khóa dùng để mã hóa có liên hệ một cách rõ ràng với khóa dùng để giải mã có nghĩa chúng có thể hoàn toàn giống nhau, hoặc chỉ khác nhau nhờ một biến đổi đơn giản giữa hai khóa.

Phương pháp này rất thích hợp khi mã hóa dữ liệu của chính mình và không có nhu cầu chia sẻ cho người khác, hoặc chia sẻ với người mà mình có thể đưa khóa giải mã qua một kênh an toàn. Trường hợp không có sự đảm bảo an toàn khi trao đổi khóa, và /hoặc khi muốn xác định bên nào đã làm lộ khóa – thì phải dùng phương pháp mã hóa khóa bất đối xứng (khóa công khai)

* Mã hóa với khóa bất đối xứng (khóa công khai):

Phương pháp mã hóa công khai dùng một cặp khóa k1 & k2, trong đó k1 dùng để mã hóa thì k2 sẽ dùng để giải mã và ngược lại. Một khóa phải giữ bí mật (được gọi là khóa Private), còn một khóa có thể công khai cho mọi người biết (còn gọi là khóa Public). Độ phức tạp và khối lượng xử lý của phương pháp này lớn hơn phương pháp khóa đối xứng rất nhiều lần nhưng rất hiệu quả khi cần trao đổi các dữ liệu mật.

5.1.3. Xác thực với mật khẩu động, khóa mềm và khóa cứng

Để tránh bị lộ khóa, tăng khả năng đề kháng các hành động dò tìm khóa, có thể thiết kế hệ thống có mật khẩu liên tục thay đổi. Cơ chế xác minh mật khẩu có thể dựa trên đồng hồ hệ thống hoặc các ngữ cảnh đặc trưng. Các giao dịch chứng khoán, thương mại điện tử, chuyển khoản tiền, … cũng hay dùng hình thức xác thực 2 lớp na ná : ngoài mật khẩu bình thường phải nhập thêm mã OTP được gửi qua SMS tới số điện thoại đã đăng ký hoặc lấy từ thiết bị Token card.

Khóa mềm là phương thức bảo vệ dữ liệu qua các công cụ phần mềm (còn được dùng vào việc bảo vệ bản quyền phần mềm chứ không chỉ là các dữ liệu bình thường). Ngược lại, khóa cứng (secure dongle, hardware key, hardlock, usb dongle,…) là phương thức bảo vệ dữ liệu thông qua thiết bị phần cứng.

Khóa mềm có nhiều ưu điểm nhưng cũng có những nhược điểm nhất định, khi đó khóa cứng là giải pháp có thể khắc phục. Khóa mềm là phương thức được nghĩ đến đầu tiên bởi các nhà lập trình, tuy nhiên khóa cứng mới là phương thức được phát minh và ứng dụng đầu tiên. Khóa cứng thường được chế tạo sẵn và khá dễ dàng sử dụng, tuy nhiên ở mức đơn giản thì các lập trình viên cũng có thể dùng các USB flash làm khóa cứng bằng cách tổ chức thiết kế các chương trình /cấu trúc riêng biệt trên đó

5.1.4. Thiết lập các hệ thống chuyên biệt

Nếu phải lưu trữ nhiều dữ liệu cần bảo mật thì có thể xây dựng hẳn một hệ thống tập tin chuyên biệt, có cấu trúc khác với các hệ thống hiện hành – và không công bố công cụ truy xuất chúng. Như vậy chỉ có thể truy xuất được các tập tin khi đang ở trong hệ thống, và ngay cả khi toàn bộ volume hoặc thiết bị lưu trữ bị chép trộm và mang đi nơi khác thì người trộm cũng không thể xem / giải mã được các tập tin – vì trước mắt là không xác định được danh sách tập tin và cũng không định vị được nội dung của từng tập tin.

5.2. THIẾT LẬP HỆ THỐNG TẬP TIN CHUYÊN BIỆT

5.2.1. Giới thiệu

Khi thiết lập một hệ thống tập tin chuyên biệt có cấu trúc do mình tự đưa ra (khác với các cấu trúc hiện hữu) thì dĩ nhiên các công cụ truy xuất tập tin hiện có sẽ không hiểu được và xem như không ai có thể lấy được các dữ liệu lưu trữ bên trong (nếu đã có mã hóa và thiết kế là không quá đơn giản /quá giống với một thiết kế có sẵn)

Ta không nhất thiết phải làm ra một hệ thống tập tin thực sự như hệ điều hành – phải dùng tới các hỗ trợ truy xuất trực tiếp trên thiết bị lưu trữ (thường là truy xuất sector), mà có thể xây dựng hệ thống tập tin nằm trong một tập tin – như các file .ZIP, .RAR, .ISO,… Trường hợp phức tạp hơn, nếu muốn gây khó khăn nhiều hơn cho những người muốn lấy trộm các tập tin dữ liệu, ta có thể tổ chức một hệ thống tập tin trên nhiều thiết bị hoặc ẩn nhúng trong các tập tin tín hiệu.

Để có thể lưu trữ được thông tin /dữ liệu vào hệ thống lưu trữ & truy xuất, sử dụng được các dữ liệu trên hệ thống lưu trữ một cách hiệu quả thì cần phải tuần tự thực hiện các công đoạn:

- Đưa ra khái niệm tập tin, thư mục, volume.
- Xây dựng mô hình thuộc tính & chức năng trên tập tin & thư mục.
- Tổ chức được hình thức lưu trữ tập tin /thư mục & các hình thức quản lý cần thiết khác trên vol.
- Lập thuật toán & chương trình thực hiện các chức năng cần thiết với các tập tin /thư mục trên vol.

5.2.2. Mô hình tổ chức và các chức năng thiết yếu

Mô hình thuộc tính

* Nội dung tập tin

* Tên tập tin

* Các thuộc tính Trạng thái

* Các thuộc tính Thời điểm

* Mật khẩu

Các chức năng

* Liệt kê danh sách tên & thuộc tính của các tập tin: ở gốc /ở thư mục con / cây thư mục, liệt kê theo thứ tự của thuộc tính nào, các thuộc tính nào cần xuất kèm, hình thức hiển thị như thế nào, …

* Đổi tên: phải kiểm tra người dùng hiện tại có được quyền thực hiện hay không, tên mới có hợp lệ không, có bị trùng với tập tin khác hay không.

* Đặt /đổi mật khẩu: phải hỏi mật khẩu cũ (nếu có), nếu đúng thì yêu cầu nhập mật khẩu mới và chỉ chấp nhận khi thỏa mãn đầy đủ các ràng buộc cần thiết (nội dung nhập ở 2 lần hỏi giống nhau, đáp ứng được qui định về chiều dài tối thiểu & tối đa, không phải là những chuỗi đặc biệt dễ đoán như trùng với tên tập tin /tên người dùng,…); nội dung tập tin phải được mã hóa tương ứng với mật khẩu; phải có cơ chế đề kháng với các hệ thống dò mật khẩu tự động;…

* Đặt /đổi thuộc tính: phải xác định thuộc tính tương ứng có được phép thay đổi hay không & tính hợp lệ của giá trị mới.

* Xóa tập tin: xóa bình thường /xóa nhưng không mất /xóa mất hẳn /xóa rác.

* Chép vào chép ra (Import /Export): chép tập tin /thư mục ra khỏi hệ thống và lưu lại theo định dạng đang có trên thiết bị được chỉ định hoặc ngược lại. Với mục đích bảo mật thì chức năng Export phải có sự kiểm soát chặt chẽ để tránh tình trạng xuất dữ liệu ra cho kẻ gian, thậm chí có thể không hỗ trợ tính năng này.

5.2.3. Tổ chức các thông số và thông tin quản lý

Có thể tham khảo cấu trúc định dạng cơ bản FAT và triển khai tương tự, nếu hệ thống tập tin không quá lớn. Cũng có thể tham khảo và thiết kế như các định dạng .ISO, .ZIP, .RAR (nhưng không cần phải nén).

Tuy nhiên, với tiêu chí cần có sự bảo mật & an toàn cao, các thành phần quan trọng dĩ nhiên cần có bản lưu. Các thành phần quan trọng cần được mã hóa trước khi lưu trữ và các bản lưu cần được mã theo kiểu khác (trước mắt để không bị phát hiện sự tồn tại của nhiều nội dung giống nhau – những nội dung mà các cracker dễ đoán là cái gì đó quan trọng rồi từ đó có cơ sở để lần dò ra các thông tin cần bảo mật)

5.2.4. Cài đặt các chức năng thiết yếu

- Các chức năng định dạng

- Các chức năng mã hóa /nén
- Các chức năng đọc dữ liệu
- Các chức năng lưu dữ liệu
- Các chức năng sửa, xóa và hủy dữ liệu

Định dạng Volume

Khái niệm

Để vol có thể sử dụng được thì thao tác đầu tiên phải tiến hành chính là định dạng (format) vol. Chức năng này có thể do người sử dụng thực hiện, cũng có thể do nhà sản xuất hoặc người phân phối làm giùm tùy theo loại thiết bị. Bởi vì khi chưa thi hành chức năng định dạng thì vol chỉ là một dãy sector có nội dung rác (những giá trị ngẫu nhiên không đúng với những giá trị cần thiết theo qui định), do đó không thể thực hiện được các thao tác truy xuất tập tin trên vol vì không biết trên vol đang có những tập tin nào, nằm tại đâu, chỗ nào còn trống, kích thước cluster là bao nhiêu,...

Như vậy việc định dạng vol chính là xác định các thông số của từng thành phần trên vol (vị trí, kích thước của cluster, bảng quản lý cluster, bảng thư mục,..) và đưa các giá trị thích hợp vào những thành phần đó. HĐH phải căn cứ vào những thông số này mới "hiểu" được tổ chức tập tin trên vol, từ đó mới thực hiện được các chức năng chép, xóa, xem, sửa,...

Muốn có một vol trắng để có thể sử dụng bình thường thì sau khi xác định vị trí & kích thước của các thành phần quản lý, ta cần phải lưu các thông số quan trọng đó vào BootSector, sau đó lưu vào các entry trên bảng thư mục các giá trị tương ứng với trạng thái trống, các phần tử trên bảng quản lý cluster cũng vậy – trừ các phần tử tương ứng với các cluster bị hư (nếu hệ thống có quản lý đến trạng thái cluster hư).

Vấn đề có vẻ lớn nhất trong chức năng định dạng chính là việc xác định kích thước bảng quản lý cluster. Khi thực hiện thao tác format thì ban đầu chỉ có kích thước vol & kích thước sector - các thông số còn lại phải tự xác định. Các thông số khác có thể tự phán quyết không được chính xác lắm cũng không gây ảnh hưởng lớn, nhưng kích thước của bảng quản lý cluster phải được tính chính xác.

Với một vol đã được định dạng ta cũng có thể định dạng lại, khi này có 2 trường hợp:

- Định dạng hoàn toàn (full format): để tạo ra những dạng thức mới phù hợp hơn cho vol, các thông số của từng thành phần trên vol sẽ được xác định lại. Chức năng này dĩ nhiên cũng được dùng cho những vol chưa được định dạng.
- Định dạng nhanh (quick format): chấp nhận giữ lại các thông số cũ của vol, chỉ cập nhật lại trạng thái các cluster đang chứa dữ liệu thành trống và cho tất cả entry trên bảng thư mục gốc về trạng thái trống. Chức năng này tương đương với việc xóa tất cả mọi tập tin & thư mục đang tồn tại trên vol, nhưng thời gian thi hành rất nhanh – có thể nhanh hơn thời gian xóa một tập tin!

Đọc nội dung tập tin trên Volume

Đây là thao tác truy xuất vol được thực hiện nhiều nhất, cũng là thao tác thường xuyên của hệ thống máy tính. Mà tốc độ truy xuất bộ nhớ ngoài (nơi chứa tập tin) chậm hơn nhiều so với bộ nhớ trong, do đó để tăng tốc độ hoạt động của máy tính – đồng thời để có thể bảo mật dữ liệu,

kiểm chứng sự hợp lý, đề kháng với các sự cố có thể gây hư hỏng, ... – khi truy xuất tập tin người ta thường dùng tới nhiều hệ thống Cache, nhiều kỹ thuật tối ưu & các xử lý khác. Tuy nhiên để có thể dễ dàng hơn cho việc nắm được một cách cơ bản tổ chức lưu trữ tập tin & cơ chế hoạt động của hệ thống quản lý tập tin, các thuật giải được trình bày sau đây chỉ nêu cách giải quyết cơ bản, chân phương nhất - không quan tâm nhiều đến việc tối ưu & các xử lý nâng cao khác.

Ta đã biết về cơ bản có hai loại tập tin khác nhau: tập tin bình thường và tập tin thư mục (còn gọi là thư mục con). Cho nên trước mắt có thể thấy có 2 thao tác khác nhau: đọc nội dung tập tin bình thường & đọc nội dung bảng thư mục con (tức nội dung tập tin thư mục - SDET). Nhưng tập tin bình thường và thư mục đều có thể nằm trong một SDET nào đó, vì vậy có thể phân ra tới 4 thao tác tương đối riêng biệt: đọc nội dung một tập tin bình thường ở RDET, đọc nội dung một thư mục con (SDET) ở RDET, đọc nội dung một tập tin bình thường ở SDET, và đọc nội dung một thư mục con nằm trong một SDET nào đó của vol.

Lưu giữ tập tin vào Volume

Sau thao tác đầu tiên là định dạng vol, chức năng kế tiếp có ảnh hưởng đến nội dung lưu trữ trên vol là đưa tập tin vào vol, chức năng này sẽ được thực hiện nhiều lần trong quá trình sử dụng vol. Việc đưa một tập tin vào vol cụ thể là chép một tập tin từ nơi khác vào vol, tạo một tập tin trên vol, hoặc tạo một thư mục con trên vol. Khi vol đang ở trạng thái trống thì các thao tác đó chỉ có thể thực hiện trên thư mục gốc của vol, nhưng khi vol đã có thư mục con thì những thao tác trên có thể thực hiện trong một thư mục con nào đó của vol.

Ta có thể phân ra 4 thao tác tương đối riêng biệt: đưa một tập tin bình thường vào thư mục gốc của vol, tạo một thư mục con ở thư mục gốc của vol, đưa một tập tin bình thường vào một thư mục con nào đó của vol, tạo một thư mục con trong một thư mục con nào đó của vol.

Xóa tập tin

Ta có nhiều dạng xóa tập tin: xóa bình thường (có thể phục hồi lại được – nhưng khả năng thành công càng thấp dần theo thời gian), xóa chắn chắn phục hồi được, xóa sao cho không thể phục hồi được, xóa thư mục, …

5.2.5. Tổ chức một hệ thống tập tin trong một tập tin

5.2.6. Tổ chức một hệ thống tập tin trên nhiều thiết bị

5.2.7. Tổ chức hệ thống tập tin nhúng trong các tập tin tín hiệu

Các Thực nghiệm

TN#5.1	*Khảo sát hình thức mật khẩu « động »*
TN#5.2	*Khảo sát hình thức đưa một phần dữ liệu ra thiết bị Removeable*
TN#5.3	*Khảo sát sơ lược các file .ZIP, .ISO*
TN#5.4	*Khảo sát hình thức đưa một phần dữ liệu ra thiết bị Removeable*

Các Bài tập :

BTVN#5.1	***Viết chương trình đưa một phần dữ liệu ra thiết bị Removeable và lấy vào trở lại***
BTVN#5.2	*Khảo sát và trình bày về các secure* dongle (khóa cứng)
BTVN#5.3	*Thiết lập một mô hình hệ thống tập tin đơn giản theo mô tả về nhu cầu và cấu hình cho trước*

Các Đồ án:

ĐAMH#5	***Viết chương trình cho phép tạo một hệ thống tập tin trong một tập tin (như các file .ZIP, .ISO) và các chức năng cho phép lấy ra /đưa vào thêm các tập tin bên trong.***

Các Bài tập / Đồ án mở rộng

MR#5.1	*Nghiên cứu trình bày các hình thức đánh cắp tài khoản người dùng trên mạng Internet (email, facebook,…) và cách nhận biết, ngăn ngừa & khắc phục tương ứng.*
MR#5.2	*Nghiên cứu trình bày các hình thức bảo vệ dữ liệu cá nhân trên PC hoặc SmartPhone (dưới góc độ người dùng).*
MR#5.3	*Nghiên cứu trình bày các hình thức bảo vệ an toàn dữ liệu trong điện toán đám mây (dưới góc độ người dùng).*

MR#5.4	*Nghiên cứu trình bày các hình thức bảo vệ dữ liệu cá nhân trong thương mại điện tử (dưới góc độ người dùng).*
MR#5.5	*Xây dựng công cụ USB logger theo dõi và ghi nhận lại danh sách các dữ liệu được đưa vào thiết bị lưu trữ qua cổng usb, vô hiệu hóa các đĩa USB flash không nằm trong danh sách cho phép sử dụng.*