1 KIỂM SOÁT TRUY CẬP

Hệ thống thông tin của một tổ chức, từ nay sẽ gọi tắt là hệ thông thông tin hay HTTT, là hệ thông máy tính bao gồm phần cứng, phần mềm và cơ sở dữ liệu được thiết lập phục vụ mục đích tạo lập, xử lý, lưu trữ và trao đổi thông tin giữa các người dùng với nhau.

Trong một HTTT, CSDL được xem như "quả tim", nơi cung cấp mọi dữ liệu cho các chương trình xử lý và trao đổi thông tin cho người dùng HTTT đó. Bảo mật cơ sở dữ liệu là những biện pháp, cách thức mà một tổ chức áp dụng để đảm bảo tính an toàn cho thông tin, CSDL khỏi các mối đe dọa từ những yếu tố xâm phạm, đánh cắp. Bảo mật CSDL đóng vai trò quan trọng trong việc hỗ trợ giải quyết và bảo vệ dữ liệu trong CSDL.

Hiện tại, hầu hết các hệ thống thông tin được thiết lập để phục vụ các người dùng trên mạng. Việc đảm bảo thông tin được cung cấp đúng đắn và cho đúng đối tượng sử dụng, ngoài việc phải có các thuật toán xử lý đúng đắn, thì việc đảm bảo việc đúng người truy cập và truy cập đúng nguồn dữ liệu là nhiệm vụ quan trọng, và gọi chung là kiểm-soát-truy-cập, Access Control (AC).

KHÁI NIỆM KIỂM SOÁT TRUY CẬP

Kiểm soát truy cập là một quy trình bảo vệ dữ liệu, qua đó người sở hữu CSDL quản lý được ai được ủy quyền truy cập vào dữ liệu và tài nguyên của mình. Kiểm soát truy cập thiết lập và sử dụng các chính sách hay cơ chế để xác minh người dùng và cấp quyền truy cập tương ứng với người dùng đã được xác minh.

Triển khai cơ chế kiểm soát truy cập (AC) là một thành phần quan trọng của bảo mật HTTT, nhất là các HTTT có ứng dụng web. AC đảm bảo chỉ những người dùng đã được xác minh phù hợp mới có quyền truy cập vào đúng tài nguyên mà họ được cấp quyền. Quá trình kiểm soát truy cập đúng đắn là rất quan trọng vì nó giúp tránh xâm phạm vào dữ liệu và chống lại các tấn công, như tấn công tràn bộ đệm, tấn công phát lại – Key Reinstallation Attacks (KRACK), tấn công lừa đảo .v.v.

Các thành phần chính trong kiểm soát truy cập

Cơ chế kiểm soát truy cập thường gồm xác thực, ủy quyền, truy cập, quản lý, và kiểm toán.

. **Xác thực – Authentication**. Xác thực là quá trình thường phải được thực hiện trước tiên để thiết lập danh tính người dùng. Chẳng hạn, để sử dụng email hay các dịch vụ trực tuyến của một ngân hàng, người dùng phải đăng nhập (log-in) bằng cách cung cho hệ thống các thông tin định danh của minh, thường là tên (username) và mật khẩu (password). Và người dùng đó chỉ sử dụng được các dịch vụ đó khi danh tính của họ đã được hệ thống xác minh.

. **Ủy quyền – Authorization**. Ủy quyền là cơ chế chỉ định quyền truy cập và một số đặc quyền trên tài nguyên. Và xác định xem một người dùng có được cấp quyền truy cập vào dữ liệu hay thực hiện một giao dịch cụ thể hay không. Ủy quyền là để tăng them một lớp bảo mật phụ cho hệ thống. Chẳng hạn, dịch vụ ngân hàng trực tuyến có thể yêu cầu người dùng cung cấp xác thực

hai yếu tố (2FA – Two Factors Authentication), thường là sự kết hợp yếu tố chỉ người dùng biết (như mật khẩu chẳng hạn), với yếu tố người đó sở hữu (như mã thông báo – token) hoặc một yếu tố nào khác (như đặc trưng sinh trắc – biometric). Phương thức được triển khai rất phổ biến là OTP – One Time Password, mật khẩu dùng một lần được chuyển qua điện thoại, xem điện thoại như là chính chủ.

. **Truy cập – Access**. Khi hệ thống xác định được danh tính người dùng, hệ thống sẽ cấp quyền truy cập vào tài nguyên hệ thông tương úng với danh tính mà người đó đăng ký và được cấp quyền.

. **Quản lý – Manage**. Nhiều tổ chức, bên cạnh việc quản lý CSDL, có thể xây dựng cơ chế quản lý hệ thống kiểm soát truy cập bằng cách thêm và xóa xác thực cũng như ủy quyền cho người dùng và hệ thống. Xác thực thường xuyên (Continuous Authentication) hay xác thực lại (re-authentication) là những ví dụ cho cơ chế này.

. **Kiểm toán – Audit**. Nhiều tổ chức có thể thực thi nguyên tắc đặc quyền tối thiểu thông qua quy trình kiểm tra kiểm soát truy cập. Cơ chế này cho phép tổ chức đó thu thập dữ liệu và phân tích thông tin về hoạt động của người dùng qua đó có thể phát hiện các vi phạm quyền truy cập tiềm ẩn.

Nói thêm về xác thực và ủy quyền

Xác thực và ủy quyền là rất quan trọng để kiểm soát truy cập trong bảo mật nói chung và CSDL nói riêng. Nếu như xác thực là quá trình đăng nhập vào hệ thống, chẳng hạn như đăng nhập để sử dung dịch vụ email, hay các dịch vụ trực tuyến ngân hàng hoặc và một mạng xã hội như Face Book hay Telegrame; thì ủy quyền là quá trình xác minh về danh tính người dùng. Ủy quyền cung cấp thêm một lớp bảo mật để khẳng định thêm rằng người dùng đang đăng nhập chính là người đã đăng ký.

Kiểm soát truy cập là rất quan trọng để giúp các tổ chức tuân thủ các quy định bảo mật dữ liệu khác nhau. Các lãnh vực cần một cơ chế kiểm soát truy cập đặc thù có thể kể đến:

PCI DSS cho ngành Tài chánh-Ngân hàng

Chuẩn bảo mật dữ liệu cho công nghiệp thẻ thanh toán (PCI DSS - The Payment Card Industry Data Security Standard). Là chuẩn bảo mật nhằm bảo vệ hệ sinh thái thẻ thanh toán. Hệ thống kiểm soát truy cập trong lãnh vực này là rất quan trọng để cho phép hay từ chối giao dịch và đảm bảo danh tính của người dùng.

HIPAA cho ngành chăm sóc sức khỏe

Đạo luật về trách nhiệm giải trình và cung cấp thông tin bảo hiểm y tế (HIPAA - Health Insurance Portability and Accountability Act). Là đạo luật của Mỹ nhằm để bảo vệ dữ liệu sức khỏe của bệnh nhân không bị tiết lộ mà không có sự đồng ý của bệnh nhân. Kiểm soát quyền truy cập rất quan trọng để hạn chế quyền truy cập của người dùng được ủy quyền, đảm bảo mọi người không thể truy cập dữ liệu vượt quá đặc quyền được cấp cho họ và ngăn chặn xâm phạm dữ liệu.

SOC 2 cho ngành cung cấp dịch vụ

Kiểm soát tổ chức dịch vụ 2 (SOC 2 - Service Organization Control 2). Là một quy trình kiểm tra được thiết kế dành cho các nhà cung cấp dịch vụ lưu trữ dữ liệu khách hàng trên cloud. Nó đảm bảo rằng các nhà cung cấp bảo vệ quyền riêng tư của khách hàng và yêu cầu các tổ chức triển khai và tuân theo các chính sách và quy trình nghiêm ngặt cho dữ liệu khách hàng. Hệ thống kiểm soát truy cập rất quan trọng để thực thi các quy trình bảo mật dữ liệu nghiêm ngặt này.

ISO 27001 trong lãnh vực bảo mật

Tổ chức Tiêu chuẩn hóa Quốc tế (ISO - International Organization for Standardization). Tổ chức xác định các chuẩn bảo mật mà các tổ chức trong tất cả các ngành cần tuân thủ và chứng minh cho khách hàng thấy rằng họ rất coi trọng vấn đề bảo mật. ISO 27001 là chuẩn vàng về chứng nhận tuân thủ và bảo mật thông tin của ISO. Việc triển khai các biện pháp kiểm soát truy cập là rất quan trọng để tuân thủ tiêu chuẩn bảo mật này.

Trong tài liệu này, chúng ta tập trung vào hai thành phần xác thực và ủy quyền, và gọi chung là kiểm soát truy cập – Access Control (AC).

CÁC PHƯƠNG PHÁP KIỂM SOÁT TRUY CẬP

Khái niệm và ứng dụng

Kiểm soát truy cập được sử dụng để xác minh danh tính của người dùng đang cố gắng đăng nhập vào tài nguyên kỹ thuật số, gọi là kiểm soát truy cập logic hay thông tin. Nhưng kiểm soát truy cập cũng được sử dụng để cấp quyền vào các tòa nhà vật lý và truy cập các thiết bị vật lý, gọi là kiểm soát truy cập vật lý.

Kiểm soát truy cập vật lý

Các ví dụ phổ biến về bộ điều khiển truy cập vật lý có thể minh họa:

. Quản lý an ninh phòng

Người phục vụ có thể thiết lập danh sách kiểm soát truy cập để xác minh ID và đảm bảo những người vào quán phòng đều hợp pháp.

. Cửa quay tàu điện ngầm

Kiểm soát truy cập được sử dụng tại các cửa quay tàu điện ngầm để chỉ cho phép những người đã được xác minh mới được sử dụng hệ thống tàu điện ngầm. Người dùng Subway quét thẻ để nhận dạng ngay người dùng và xác minh rằng họ có đủ tín dụng để sử dụng dịch vụ.

. Máy quét thẻ từ hoặc huy hiệu trong văn phòng công ty

Các tổ chức có thể bảo vệ văn phòng của mình bằng cách sử dụng máy quét để kiểm soát vào/ra bắt buộc. Nhân viên cần quét thẻ khóa hoặc huy hiệu để xác minh danh tính trước khi có thể vào tòa nhà.

Kiểm soát truy cập logic/thông tin

Kiểm soát truy cập logic bao gồm các công cụ và giao thức được sử dụng để xác định, xác thực và ủy quyền cho người dùng trong hệ thống máy tính. Hệ thống kiểm soát truy cập thực thi các biện pháp đối với dữ liệu, quy trình, chương trình và hệ thống.

. Đăng nhập vào máy tính xách tay bằng mật khẩu

Một dạng mất dữ liệu phổ biến là do thiết bị bị mất hoặc bị đánh cắp. Người dùng có thể giữ an toàn cho dữ liệu cá nhân và công ty của mình bằng cách sử dụng mật khẩu.

. Mở khóa điện thoại thông minh bằng quét dấu vân tay

Điện thoại thông minh cũng có thể được bảo vệ bằng các biện pháp kiểm soát truy cập chỉ cho phép chủ nhân mới mở thiết bị. Người dùng có thể bảo mật điện thoại thông minh của mình bằng cách sử dụng sinh trắc học, chẳng hạn như quét dấu vân tay, để ngăn chặn truy cập trái phép vào thiết bị của họ.

. Truy cập từ xa vào mạng nội bộ của nhà cung cấp dịch vụ bằng VPN

Điện thoại thông minh cũng có thể được bảo vệ bằng các biện pháp kiểm soát truy cập chỉ cho phép người dùng mở thiết bị. Người dùng có thể bảo mật điện thoại thông minh của mình bằng cách sử dụng sinh trắc học, chẳng hạn như quét dấu vân tay, để ngăn chặn truy cập trái phép vào thiết bị của họ.

Các phương pháp kiểm soát truy cập phổ biến

Kiểm soát truy cập dựa trên vai trò (RBAC)

RBAC - Role-Based Access Control, tạo quyền dựa trên nhóm người dùng, vai trò mà người dùng nắm giữ và hành động mà người dùng thực hiện. Người dùng có thể thực hiện bất kỳ hành động nào được kích hoạt cho vai trò của họ và không thể thay đổi cấp độ kiểm soát truy cập mà họ được chỉ định.

RBAC gán quyền cho người dùng dựa trên vai trò của họ trong tổ chức. RBAC là một tiếp cận đơn giản, dễ quản lý để quản lý quyền truy cập, thay vì phải chỉ định quyền cho từng người dùng.

Khi sử dụng RBAC, cần phân tích nhu cầu của người dùng và định nghĩa các vai trò cùng trách nhiệm. Sau đó, chỉ định một hoặc nhiều vai trò cho mỗi người dùng và các quyền tương ứng cho từng vai trò. Mối quan hệ vai trò-người dùng và vai trò-quyền giúp việc thực hiện nhiệm vụ người dùng đơn giản vì người dùng không được quản lý riêng lẻ mà thay vào đó có các đặc quyền phù hợp với các quyền được gán cho các vai trò của họ.

Ví dụ, sử dụng RBAC để kiểm soát truy cập trong ứng dụng quản lý nhân sự, có thể cấp cho người quản lý nhân sự vai trò e – edit, cho phép cập nhật thông tin chi tiết về nhân viên, trong khi các nhân viên khác chỉ có vai trò r – read, xem thông tin chi tiết của chính mình.

Mô hình

. Vai trò

Về cơ bản, vai trò là tập các quyền truy cập có thể cấp cho người dùng. Việc sử dụng vai trò giúp việc thêm – a (append), xóa – d (delete) và điều chỉnh – e (edit) quyền truy câp được dễ dàng hơn so với việc chỉ định quyền cho từng người dùng.

Vai trò cũng có thể được sử dụng để nhận các quyền từ các API khác nhau. Chẳng hạn, giả sử có mô-đun tiếp thị cho phép người dùng tạo và phân phối bản tin cho khách hàng. Chuyên viên nội dung tiếp thị tạo các bản tin để có thể phân phối chúng. Hay, giả sử có mô-đun sự kiện cho phép người dùng tạo, xuất bản và quản lý việc đăng ký sự kiện. Điều phối viên sự kiện có thể tạo ra các sự kiện. Sau khi người phụ trách tiếp thị phê duyệt bản tin và sự kiện, trợ lý của họ sẽ công bố các sự kiện và phân phối các bản tin. Trong trường hợp này, API "bản tin" có thể có quyền phân phối bản tin và API "sự kiện" có thể có quyền công bố sự kiện. Sau đó, các quyền này có thể được tập hợp trong vai trò "người công bố tiếp thị" và cấp cho trợ lý tiếp thị.

Ngoài ra, các vai trò dành riêng cho công ty hay tổ chức có thể được cấp thêm cho các thành viên của công ty để cho phép người của công ty truy cập được vào ứng dụng khi người dùng cuối đang đăng nhập vào hệ thống. Điều này hữu ích khi hỗ trợ các sản phẩm kiểu SaaS (Software as a Service) và cho thuê (multi-tenant), trong đó một người dùng nào đó có thể có vai trò đặc quyền trong tổ chức này nhưng lại không có vai trò đặc quyền đó nữa trong tổ chức khác.

. Phân công vai trò chồng chéo

RBAC là một mô hình bổ sung, vì vậy nếu các nhiệm có vụ vai trò chồng chéo nhau thì các quyền có hiệu lực của người dùng sẽ là sự kết hợp của các nhiệm vụ vai trò đã được cấp quyền.

Chẳng hạn, giả sử có API cung cấp dữ liệu cho ứng dụng sự kiện. Có thể tạo vai trò "Người tổ chức" và gán cho nó các quyền r – xem, tạo – c (create) và e – chỉnh sửa sự kiện. Cũng có thể tạo vai trò "Người đăng ký" và gán quyền r – xem và s (subscribe) – đăng ký các sự kiện. Bất kỳ người dùng nào có cả vai trò "Người tổ chức" và "Người đăng ký" đều có thể xem, tạo, chỉnh sửa và đăng ký sự kiện.

<u>Nhận xét</u>

Có hai cách phổ biến để cài đặt RBAC và có thể sử dụng riêng lẻ hay kết hợp chúng trong một hệ thống kiểm soát truy cập của API: ủy quyền lõi và gia hạn ủy quyền

Thực chất, ủy quyền lõi là API cơ bản cung cấp các chức năng cho phép triển khải RBAC. Và có thể mở rộng tập chức năng của ủy quyền lõi để có một API đầy đủ cho RBAC.

Kiểm soát truy cập dựa trên thuộc tính (ABAC)

ABAC - Attribute-based Access Control, là chính sách động, dựa trên ngữ cảnh, xác định quyền truy cập dựa trên các chính sách được cấp cho người dùng. Hệ thống này được sử dụng trong khuôn khổ quản lý danh tính và truy cập (IAM - identity and access management).

ABAC là mô hình ủy quyền đánh giá các thuộc tính, thay vì vai trò, để xác định quyền truy cập. Mục đích của ABAC là để bảo vệ các đối tượng như dữ liệu, thiết bị mạng và tài nguyên CNTT khỏi những người dùng trái phép, là những đối tượng không có đặc điểm được phê duyệt đã được định nghĩa trước trong chính sách bảo mật của tổ chức.

ABAC là một hình thức kiểm soát truy cập logic, được phát triển từ các phương pháp kiểm soát truy cập đơn giản và kiểm soát truy cập dựa trên vai trò (RBAC - role-based access control). ABAC được đề xuất làm mô hình chuẩn, áp dụng cho các tổ chức nhằm chia sẻ thông tin một cách an toàn.

<u>Các thực thể trong kiểm soát truy cập dựa trên thuộc tính</u>

Với ABAC, chính sách truy cập của tổ chức thực thi các quyết định truy cập dựa trên các thuộc tính của chủ đề, tài nguyên, hành động và môi trường liên quan đến một sự kiện truy cập.

. Chủ thể

Chủ thể là người dùng yêu cầu quyền truy cập vào tài nguyên. Các thuộc tính của chủ đề có thể gồm ID, vai trò công việc, tư cách thành viên nhóm, tư cách thành viên phòng ban và tổ chức, cấp quản lý, thông tin bảo mật và các tiêu chí nhận dạng khác. ABAC thường lấy các thông tin thuộc tính này từ CSDL nhân sự hoặc từ mã thông báo xác thực được sử dụng trong quá trình đăng nhập.

. Nguồn

Tài nguyên là nội dung (các trường trong CSDL) hoặc đối tượng (tập tin, ứng dụng, máy chủ hoặc API) mà chủ thể muốn truy cập. Thuộc tính tài nguyên thường được dùng làm các đặc điểm nhận dạng, chẳng hạn như với đối tượng tập tin có thể là ngày tạo, chủ sở hữu, tên và loại tập tin, ... Hay khi truy cập tài khoản ngân hàng trực tuyến, tài nguyên liên quan sẽ là "tài khoản ngân hang, thường là số tài khoản mà người dùng phải nhập chính xác.

. Hoạt động

Hoạt động là những gì người dùng có thể thực hiện trên tài nguyên. Thuộc tính hoạt động thường gồm: đọc – r (read), ghi – w (write), chỉnh sửa – e (edit), sao chép – c (copy) và xóa – d (delete). Trong nhiều trường hợp, một số thuộc tính có thể mô tả một hành động. Chẳng hạn, yêu cầu chuyển khoản có thể có các đặc điểm: "chuyển khoản" (loại hành động) và "số tiền" (nội dung hành động).

. Môi trường

Môi trường là bối cảnh tổng thể của mỗi yêu cầu truy cập. Mọi thuộc tính môi trường đều liên quan đến ngữ cảnh như thời gian và địa điểm, thiết bị truy cập, giao thức liên lạc. Thông tin theo ngữ cảnh cũng có thể gồm các dấu hiệu rủi ro tổ chức đã thiết lập trước, chẳng hạn như tần suất hay số lần xác thực và các mẫu hành vi thông thường của đối tượng.

<u>Cách sử dụng các thuộc tính trong kiểm soát truy cập dựa trên thuộc tính</u>

Thuộc tính là các đặc điểm hoặc giá trị của một thành phần liên quan đến một sự kiện truy cập. ABAC phân tích các thuộc tính này theo các quy tắc định trước. Các quy tắc này định nghĩa thuộc tính nào được ủy quyền để chủ thể có thể thực hiện một hành động trên một đối tượng.

Dựa trên cách các thuộc tính tương tác trong môi trường, giải pháp cho ABAC có thể đánh giá chúng trong môi trường và thực thi các quy tắc cũng như mối quan hệ. Các chính sách nhằm xác định điều kiện truy cập nào được phép hay không.

Ví dụ với chính sách: "Đối tượng đảm nhận vai trò truyền thông có quyền đọc và chỉnh sửa các chiến lược truyền thông cho đơn vị kinh doanh mà đối tượng đó đại diện."

Khi có yêu cầu truy cập, ABAC sẽ phân tích các giá trị thuộc tính xem có thỏa các chính sách đã cài đặt. Khi chính sách trên được áp dụng, yêu cầu truy cập có các thuộc tính sau sẽ cấp quyền truy cập:

Đốitượng(Vai trò công việc) = "giao tiếp"

Chủthể(Đơn vị kinh doanh) = "tiếp thị"

Loại(Hành động) = "chỉnh sửa"

Loại(Tài nguyên) = "tài liệu chiến lược truyền thông"

Nguồnlực(đơn vị kinh doanh) = "tiếp thị"

Trong thực tế, ABAC cho phép quản trị viên triển khai kiểm soát quyền truy cập chi tiết, dựa trên chính sách, sử dụng các thuộc tính khác nhau để tạo điều kiện truy cập cụ thể tùy thuộc tình huống yêu cầu.

Nhận xét

ABAC cho phép xây dựng chính sách kiểm soát truy cập vừa chi tiết vừa linh hoạt, dễ dàng tương thích với người dùng mới và đảm bảo quyền riêng tư nghiêm ngặt. Tuy nhiên, như ta thấy, ABAC sẽ phức tạp khi thiết kế cũng như cài đặt.

Mặc dù việc triển khai ABAC có thể mất nhiều thời gian và nguồn lực nhưng nỗ lực nhưng một khi đã triển khai thành công, quản trị viên có thể sao chép và sử dụng lại các thuộc tính cho các thành phần và vị trí người dùng cùng vai trò. Hơn nữa, với khả năng linh động của ABAC, việc duy trì chính sách cho người dùng mới và các tình huống truy cập là một công việc tương đối "dễ dàng".

Kiểm soát truy cập tùy ý (DAC)

DAC - Discretionary Access Control, cho phép chủ sở hữu dữ liệu quyết định kiểm soát quyền truy cập bằng cách gán quyền truy cập cho các quy tắc mà người dùng chỉ định. Khi người dùng được cấp quyền truy cập vào hệ thống, họ có thể cung cấp quyền truy cập cho những người dùng khác khi họ thấy phù hợp.

DAC là một phương pháp kiểm soát truy cập nhằm bảo mật HTTT. DAC cấp cho người dùng quyền kiểm soát các quyền truy cập vào tài nguyên của mình, cho phép chủ sở hữu HTTT và sở hữu dữ liệu quyết định ai có thể truy cập vào tài nguyên tương ứng và mức độ truy cập. Tiếp cận này hỗ trợ nguyên tắc đặc quyền tối thiểu, nghĩa là cấp cho người dùng quyền truy cập cần thiết nhất để thực hiện công việc của họ. DAC dựa vào việc quyết định cấp quyền truy cập như thế nào thuộc chủ sở hữu HTTT hoặc dữ liệu, nên cần xem xét nghiêm ngặt các yêu cầu truy cập để không cấp quyền quá mức.

Mô hình DAC

Cơ bản, DAC sử dụng danh sách kiểm soát truy cập (ACL – Access Control List) để gán quyền truy cập tài nguyên. ACL gồm thông tin (nhóm) người dùng được xác định trước và cấp độ truy cập tương ứng. Các cấp độ này có thể là r – đọc, w – ghi và thực thi –R (Run), cho phép một người dùng có thể xem, sửa hoặc chạy một quy trình hoặc chương trình tương ứng quyền được cấp.

Chẳng hạn, giả sử một người dùng cần mở một tập tin được chia sẻ trên mạng công ty. Khi người dùng đó yêu cầu quyền truy cập, hệ thống DAC sẽ kiểm tra thông tin xác thực và so sánh nó với ACL được liên kết với tài nguyên mà người dùng đang thử truy cập. Nếu thông tin khớp với mục tương ứng trong ACL, người dùng sẽ được cấp quyền truy cập như được xác định trong ACL. Nếu thông tin người dùng không khớp với ACL thì yêu cầu sẽ sẽ bị từ chối.

Các bước hoạt động của DAC như sau:

(1). Tạo tài nguyên. Ví dụ, một người dùng A tạo ra tập tin F thì người đó hiện là chủ sở hữu F và có thể kiểm soát quyền truy cập.

(2). Sau đó, A cấu hình ACL cho tài nguyên F vừa tạo, chỉ định (nhóm) người dùng có thể cần quyền truy cập và với các quyền tương ứng.

(3). Người dùng B đã được cấp quyền truy cập vào tài nguyên F truy cập vào tài nguyên đó, phát sinh yêu cầu truy cập chuyển cho hệ thống DAC.

(4). DAC kiểm tra thông tin của B, là người dùng đang yêu cầu quyền truy cập F có khớp với mục đã đăng ký trong ACL hay không, để phê duyệt hoặc từ chối yêu cầu của B. Phê duyệt này phụ thuộc vào việc hệ thống có tìm thấy thông tin của B trong ACL hay không.

(5). DAC thực thi quyết định của (4) theo thời gian thực, cho phép người dùng B xem, sửa đổi hoặc chạy tài nguyên B yêu cầu miễn là ACL tương ứng cho phép hành động.

Cài đặt kiểm soát truy cập tùy ý

Để cài đặt DAC cần lập kế hoạch và xem xét các chính sách bảo mật hiện có, vì người sở hữu tài nguyên phải đảm bảo tuân thủ nội quy của công ty hoặc của bộ phận. Các bước chính triển khai DAC hiệu quả và đảm bảo đúng cách như sau.

. Xác định chính sách bảo mật

Thiết lập một chính sách kiểm soát truy cập rõ ràng. Xác định người nào có quyền truy cập vào tài nguyên nào và mức độ truy cập nào sẽ được cấp.

. Phân loại tài nguyên

Phân loại tài nguyên thường dựa trên độ nhạy cảm và tầm quan trọng của tài nguyên. Điều này là cần thiết để đảm bảo ACL được chỉ định phù hợp cho các tài nguyên, đặc biệt là những tài nguyên bí mật hoặc có độ nhạy cảm cao mà quyền truy cập phải được kiểm soát cẩn thận.

. Thiết lập quản lý (nhóm) người dùng

Tạo và quản lý các tài khoản và nhóm người dùng để gán quyền DAC. Đảm bảo người dùng và nhóm được tổ chức theo cách phù hợp với chính sách kiểm soát truy cập đã xác định.

. Định cấu hình ACL

Đối với mỗi tài nguyên, cấu hình ACL định nghĩa (nhóm) người dùng nào có thể truy cập tài nguyên và mức độ quyền mà họ phải có (r, w hoặc R).

. Thực hiện kiểm toán thường xuyên

Định kỳ xem xét và kiểm tra các chính sách của DAC để đảm bảo không có tình huống nào không tuân thủ. Nếu xác định được quyền nào không tuân thủ, phải điều tra, ghi lại và cập nhật để phù hợp với chính sách DAC.

. Đào tạo và nâng cao nhận thức người dùng

Hướng dẫn người dùng về tầm quan trọng của DAC, vai trò của họ trong việc duy trì triển khai DAC hiệu quả và hậu quả của việc không tuân thủ các chính sách kiểm soát truy cập.

Một số ứng dụng của DAC

. Quyền trên tập tin và thư mục

Ví dụ phổ biến nhất là quyền truy cập tập tin và thư mục trong hệ điều hành Windows và Unix. Khi một tập tin hoặc một thư mục được tạo ra, người tạo có thể chỉ định ai có thể truy cập và họ quyền truy cập (đọc, ghi, thực thi).

. Nền tảng lưu trữ đám mây

Tương tự với các nền tảng lưu trữ đám mây, như Microsoft OneDrive hay SharePoint hoặc Google Drive. Nếu người dùng muốn chia sẻ một tập tin hoặc một thư mục đã tồn tại trên tài khoản lưu trữ cá nhân của họ, họ phải chỉ định tập tin hoặc thư mục đó sẽ được chia sẻ cho ai và mức độ truy cập mà họ muốn cung cấp (r hoặc e). Mức độ kiểm soát này cho phép một cá nhân có thể quản lý tập tin của mình theo ý muốn, hoặc theo sự nhạy cảm thông tin và yêu cầu bảo mật.

. Hệ Quản Trị Cơ sở Dữ liệu

Hệ quản trị cơ sở dữ liệu – DBMS (Database Management Systems) là một ví dụ khác về việc dùng DAC. Để kiểm soát quyền truy cập vào các CSDL khác nhau, người quản trị CSDL xác

định người dùng có thể truy cập CSDL nào và người đó có thể làm gì trong CSDL đó. Điều này đảm bảo rằng chỉ những người được ủy quyền xem từng CSDL mới được cấp quyền truy cập vào dữ liệu họ cần (r, w, hay e).

Nhận xét

Mặc dù DAC linh hoạt theo người dùng nhưng nó cũng có một số nhược điểm và hạn chế.

. Dựa vào ý chủ quan của con người

Vì DAC dựa vào người sở hữu tài nguyên để đưa ra quyết định kiểm soát quyền truy cập nên ACL có thể bị cấu hình sai hoặc hiểu sai các yêu cầu bảo mật, dẫn đến có thể làm lộ dữ liệu và truy cập trái phép.

. Không thể mở rộng

DAC không phải là một phương pháp kiểm soát truy cập có thể mở rộng vì nó tốn thời gian và phức tạp khi phải quản lý quyền truy cập đối với nhiều tài nguyên và người dùng.

. Thiếu quản trị tập trung

DAC thiếu khả năng kiểm soát tập trung vì quyền truy cập được áp dụng ở cấp độ tài nguyên. Điều này gây khó khăn cho việc thực thi các chính sách bảo mật cụ thể trong toàn tổ chức và đánh giá các chính sách hiện có.

. Rủi ro đe dọa nội bộ gia tăng

DAC không cung cấp biện pháp bảo vệ đầy đủ trước các mối đe dọa nội bộ, trong đó người dùng được ủy quyền có thể lạm dụng đặc quyền của họ để truy cập hoặc đánh cắp dữ liệu nhạy cảm, cấp quyền cho người dùng trái phép hoặc tiết lộ thông tin cho các bên trái phép.

Kiểm soát truy cập bắt buộc (MAC)

MAC - Mandatory Access Control, đặt ra các chính sách nghiêm ngặt đối với người dùng cá nhân cũng như dữ liệu, tài nguyên và hệ thống mà họ muốn truy cập. Các chính sách được quản lý bởi quản trị viên của tổ chức. Người dùng không thể thay đổi, thu hồi hoặc đặt quyền.

MAC là một chiến lược bảo mật nhằm hạn chế khả năng mà người sở hữu tài nguyên phải cấp hoặc từ chối quyền truy cập vào tài nguyên trong hệ thống tập tin. Tiêu chí MAC được xác định bởi người quản trị hệ thống, được hệ điều hành – OS (Operating System) luôn thực thi và nhân bảo mật cũng như người dùng cuối không thể thay đổi.

MAC kiểm soát truy cập vào tài nguyên dựa trên hai yếu tố chính: sự nhạy cảm của thông tin trong tài nguyên và sự ủy quyền của người dùng đang truy cập tài nguyên đó.

Nhóm phụ trách bảo mật hoặc người quản trị định nghĩa một tài nguyên nhạy cảm theo các cấp độ bảo mật khác nhau, như "Bị hạn chế", "Bí mật", "Bí mật" hoặc "Tối mật" và chỉ định tài nguyên trong danh sách truy cập (ACL – Access control List) như "Phòng M" hoặc "Dự án X.". Cặp (mức bảo mật, người truy cập) được xem là nhãn bảo mật cho tài nguyên. Người quản trị

cũng có thể chỉ định mức độ bảo mật cho từng người để xác định tài nguyên nào họ có thể truy cập.

Sau khi gán nhãn bảo mật và hoàn thành chính sách MAC, người dùng chỉ có thể truy cập tài nguyên mà họ có quyền truy cập. Chẳng hạn, một người dùng A có thể có quyền truy cập thông tin trong tài nguyên có nhãn "Phòng M bị hạn chế", nhưng dùng B khác có thể không có quyền đó. Tương tự, B có thể có quyền truy cập vào tài nguyên được gắn nhãn "Bí mật của Dự án X", nhưng A có thể không.

Ứng dụng của bảo mật bắt buộc

MAC là một phương pháp quan trọng để kiểm soát truy cập dữ liệu. MAC thường được sử dụng để bảo vệ thông tin quan trọng. Thông tin này có thể riêng tư, nhạy cảm, bí mật hoặc bị hạn chế. Chẳng hạn:

. Bí mật thương mại.

. Bản thiết kế.

. Kế hoạch chiến lược hoặc sáp nhập và mua lại.

. Sở hữu trí tuệ.

. Thông tin cá nhân.

. Thông tin tài chính và giao dịch.

. Thông tin sức khỏe được bảo vệ.

. Thông tin khách hàng.

Những loại thông tin này có thể gây tổn hại về tài chính hoặc danh tiếng cho chủ sở hữu nếu vào tay kẻ xấu. Đó là lý do tại sao việc bảo vệ thông tin, duy trì tính bảo mật (Confident), tính toàn vẹn (Integrity) và tính khả dụng (Availability) của thông tin đó lại quan trọng – còn được gọi là bộ ba CIA, đảm bảo rằng chỉ những người dùng được ủy quyền mới có thể truy cập thông tin. Đây là nơi việc triển khai MAC có thể hữu ích.

Kiểm soát truy cập bắt buộc hoạt động bằng cách gán nhãn phân loại cho từng đối tượng hệ thống tập tin. Ngoài ra, mỗi người dùng được chỉ định một mức độ bảo mật. Người dùng chỉ có thể truy cập đối tượng hoặc tài nguyên nếu mức độ bảo mật của họ bằng hoặc lớn hơn nhãn phân loại của tài nguyên ("Bị hạn chế", "Bí mật", v.v.).

Khi một người hoặc một thiết bị truy cập vào một tài nguyên cụ thể, hệ điều hành hoặc lõi bảo mật sẽ kiểm tra thông tin xác thực của thực thể để xác định xem có quyền truy cập và được cấp hay chưa. Mặc dù MAC là cài đặt kiểm soát truy cập an toàn nhất hiện có nhưng nó đòi hỏi phải lập kế hoạch cẩn thận và giám sát liên tục để luôn cập nhật tất cả các phân loại của đối tượng tài nguyên và người dùng.

Quản trị viên đóng vai trò quan trọng trong việc thiết lập và thực thi MAC cũng như duy trì mô hình phân cấp của MAC. Người quản trị thiết lập quyền và quyền kiểm soát của người dùng. Do

quản trị tập trung và nghiêm ngặt như vậy nên người dùng không phải quản trị viên không thể đặt quyền riêng cho mình. Họ cũng không thể truy cập các tài nguyên tương ứng với mức độ bảo mật cao hơn mức độ bảo mật của họ trong hệ thống phân cấp.

Nhận xét

MAC được coi là một cách có mức an toàn cao khi dùng để kiểm soát quyền truy cập vào các tài nguyên nhạy cảm hoặc bí mật. MAC đặc biệt hữu ích cho việc bảo vệ tính bảo mật của dữ liệu. Vì quản trị viên kiểm soát người dùng nào có quyền truy cập vào tài nguyên nào nên người dùng cũng không thể thực hiện các thay đổi về quyền truy cập, điều này có thể ảnh hưởng đến tính bảo mật của tài nguyên. Những lợi ích này làm cho MAC phù hợp để bảo vệ dữ liệu nhạy cảm trong môi trường chính phủ và quân đội.

Một nhược điểm của MAC là khó quản lý vì khó cấu hình và duy trì tất cả quyền truy cập tập trung vào quản trị viên, nhất là khi số lượng hệ thống và người dùng tăng lên. Nên MAC không phù hợp với các ứng dụng có nhiều người dùng, chẳng hạn như các ứng dụng dựa trên internet.

Một nhược điểm khác của MAC là việc triển khai có thể tốn kém. Việc xóa một người dùng không cho truy cập một hoặc nhiều loại tài nguyên có thể tốn thời gian và tốn kém. Chi phí và nỗ lực sẽ tăng hơn nữa khi phải áp dụng các mức độ bảo mật hoặc các vùng bảo mật khác nhau trong cùng một hệ thống CNTT. Do đó, MAC không thường được sử dụng trong môi trường doanh nghiệp có ngân sách hạn chế.

Kiểm soát truy cập kính vỡ (BGAC)

BGAC - Break-glass Access Control, liên quan đến việc tạo một tài khoản khẩn cấp bỏ qua các quyền thông thường. Trong trường hợp khẩn cấp nghiêm trọng, người dùng được cấp quyền truy cập ngay vào hệ thống hoặc tài khoản mà thông thường họ không được phép sử dụng.

BGAC là khái niệm có nguồn gốc từ báo động khẩn cấp, theo hình ảnh chuông báo cháy được bảo vệ trong các tủ "kính vỡ". Các tủ này có cần gạt báo động hoặc nút phía sau kính để đảm bảo chỉ sử dụng trong trường hợp khẩn cấp. Điều quan trọng là không thể "tắt" cảnh báo nếu không tháo và thay thế một bộ phận trong tủ. Trong CNTT, "kính vỡ" đề cập đến kỹ thuật sử dụng truy cập vào hệ thống trong điều kiện khẩn cấp nhằm vượt qua các biện pháp bảo mật chuẩn.

Các mô hình kiểm soát truy cập thường rất cứng nhắc về các quyền dựa trên chính sách cụ thể hiếm khi thay đổi. Tuy nhiên, trong nhiều trường hợp, kiểm soát truy cập hoặc chính sách cơ bản cần cung cấp thêm tính linh hoạt. Điều này cần thiết trong các lãnh vực như chăm sóc sức khỏe hay quản lý thảm họa. Trong bối cảnh đó, kính vỡ là một chiến lược cung cấp hỗ trợ chính sách linh hoạt nhằm ngăn chặn tình trạng trì trệ của hệ thống có thể gây nguy hiểm đến tính mạng hoặc gây ra những tổn thất khác.

BGAC đề cập đến một phương cách nhanh chóng và đơn giản để ai có đặc quyền truy cập có thể truy cập dữ liệu hạn chế trong trường hợp khẩn cấp. Quy trình kính vỡ phải được tạo, ghi lại, thực hiện và thử nghiệm trong các hệ thống lưu trữ thông tin nhạy cảm. Các thủ tục này là cần thiết để trong trường hợp khẩn cấp có thể có quyền truy cập vào thông tin quan trọng. Quy trình

phải được ghi chép đầy đủ và dễ hiểu; để có chính sách rõ ràng nhằm hỗ trợ việc truy cập dữ liệu theo cách khác và/hoặc thủ công.

Mô hình trình kiểm soát truy cập kính vỡ

Có ba phần chính cần giải quyết trước khi triển khai giải pháp BGAC:

(1). *Tạo tài khoản khẩn cấp an toàn trên đám mây*

Có thể bắt đầu thao tác phá kính chỉ với hai tài khoản khẩn cấp trên đám mây. Các tài khoản khẩn cấp không được liên kết với bất kỳ hệ thống cục bộ nào và thông tin xác thực chỉ được chia sẻ với những người được phép sử dụng quyền truy cập bằng BGAC.

(2). *Thiết lập mật khẩu an toàn*

Một cách để tăng cường bảo mật mật khẩu là chia mật khẩu của tài khoản truy cập khẩn cấp thành ít nhất hai phần và lưu trữ chúng an toàn và riêng biệt. Nếu tình huống kính vỡ xảy ra, quản trị viên có thông tin xác thực có thể kết nối lại hai nửa này.

(3). *Thiết lập cấu hình ban đầu*

Nên phân bổ vai trò quản trị viên toàn cục lâu dài cho một số người đáng tin trong tổ chức. Hơn nữa, cần đảm bảo rằng tất cả các cơ quan quản lý đều sử dụng xác thực đa yếu tố (MFA – Multi-Factor Authentication). Không cần phải có MFA đối với tài khoản kính vỡ nếu nhân viên có đặc quyền kính vỡ chỉ có quyền truy cập vào thiết bị của riêng họ.

Ứng dụng của BGAC

BGAC có thể ứng dụng trong các tình huống kiểm soát truy cập điển hình sau.

. Tài khoản bắt buộc chúng thực đa yếu tố

Quản trị viên MFA có quyền truy cập vào một nhóm đặc quyền. Việc xác minh có thể được thực hiện qua điện thoại hoặc tin nhắn; tuy nhiên, do mạng di động bị cúp nên điều này không thể thực hiện được. Nếu công ty có cơ chế ghi đè xác thực, quản trị viên hệ thống có thể bắt đầu "phá kính" và kích hoạt các vai trò cần thiết.

. Quản lý tài khoản đặc quyền (PAM - Privileged Account Management)

Thông thường, thông tin xác thực của quản trị viên đặc quyền được lưu trữ được mã hóa một vùng đặc biệt trong hệ thống PAM. Tuy nhiên, trong tình huống mất quyền truy cập vào vùng đặc biệt đó khiến việc truy xuất thông tin xác thực cho tài khoản quản trị trở nên khó khăn. Các giao thức kính vỡ khẩn cấp được áp dụng nếu chỉ duy nhất quản trị viên hệ thống có thể truy cập vào kho mật khẩu để ở nơi đặc biết hoặc một cuộc tấn công từ chối dịch vụ phân tán - DDoS (Distributed Denial of Services) ngăn cản bất kỳ ai đăng nhập.

. Truy cập khẩn cấp vào hồ sơ sức khỏe điện tử - ePHI (electronic Personal Health Information)

Thông tin xác thực cho tài khoản ePHI bị mất hoặc bị đánh cắp có thể trì hoãn hoặc thậm chí ngăn cản việc điều trị khẩn cấp. Phương pháp kính vỡ cho phép người chăm sóc không có đặc

quyền hoặc người chăm sóc truy cập vào tài khoản bị hạn chế và cung cấp các phương pháp điều trị cần thiết.

<u>Các giải pháp kiểm soát truy cập kính vỡ</u>

Để tận dụng khả năng GBAC, các giải pháp sau có thể thực hiện.

(1). *Yêu cầu MFA trong hầu hết các trường hợp*

Triển khai MFA cho mọi người dùng để hạn chế khả năng bị tấn công do mật khẩu bị xâm phạm. Tuy nhiên, cần có các phần ghi đè GBAC không phụ thuộc vào MFA điện thoại hoặc tin nhắn.

(2) *Có ít nhất một tài khoản truy cập không có điều kiện*

Có ít nhất một tài khoản có quyền truy cập khẩn cấp được miễn tất cả các quy tắc truy cập có điều kiện. Điều này sẽ đảm bảo rằng có ít nhất một điểm truy cập trong trường hợp khẩn cấp.

(3) *Giữ thông tin xác thực tài khoản an toàn*

Đảm bảo chỉ những nhân viên được ủy quyền mới có thể truy cập thông tin đăng nhập tài khoản truy cập khẩn cấp.

Như đã đề cập trước, mật khẩu cho tài khoản truy cập khẩn cấp nên được chia thành nhiều phần và được bảo quản an toàn ở những nơi an toàn tách biệt với nhau.

Nếu chọn triển khai bảo vệ bằng mật khẩu, phải đảm bảo sử dụng mật khẩu mạnh, không hết hạn được tạo ngẫu nhiên và dài ít nhất 16 ký tự.

(4) *Theo dõi hồ sơ kiểm toán và hoạt động đăng nhập*

Trong trường hợp khẩn cấp, nên theo dõi ai đã đăng nhập vào hệ thống và đã làm gì sau khi đăng nhập, đồng thời thông báo cho người phụ trách. Có thể theo dõi các tài khoản kính vỡ để đảm bảo chúng chỉ được sử dụng cho mục đích thử nghiệm và cho tình huống thực tế.

<u>Lưu ý</u>

Điều quan trọng là phải giữ quyền truy cập khẩn cấp vào hệ thống tại chỗ và quyền truy cập khẩn cấp vào các dịch vụ đám mây riêng biệt và độc lập. Trong trường hợp hệ thống ngừng hoạt động, rủi ro tăng thêm do việc kiểm soát hoặc tìm nguồn cung ứng xác thực cho các tài khoản có đặc quyền truy cập khẩn cấp từ các hệ thống khác là không nên.

Kiểm soát truy cập dựa trên quy tắc (RuBAC)

RuBAC - Rule-based Access Control, cách tiếp cận dựa trên quy tắc cho phép quản trị viên hệ thống xác định các quy tắc chi phối quyền truy cập vào tài nguyên của công ty. Các quy tắc này thường được xây dựng dựa trên các điều kiện, chẳng hạn như vị trí hoặc thời gian trong ngày mà người dùng truy cập tài nguyên.

RuBAC thiết lập quyền truy cập theo một bộ quy tắc xác định trước cho phép hoặc từ chối quyền truy cập của người dùng trong hệ thống.

Trong RuBAC, quản trị viên đặt ra các quy tắc xác định cách thức, thời gian và địa điểm một người có thể truy cập vào các khu vực, không gian và tài nguyên.

Danh sách kiểm soát – ACL (Access Control List) được thiết lập cho từng không gian hoặc tài nguyên và khi nhân viên yêu cầu quyền truy cập, danh sách các yêu cầu sẽ được hệ thống kiểm soát truy cập kiểm tra và quyền truy cập sẽ được cấp hoặc bị từ chối.

Không giống như kiểm soát quyền truy cập dựa trên vai trò – RBAC (Role-based Access Control), trong RuBAC, quyền truy cập không liên quan đến vai trò cụ thể hoặc hệ thống phân cấp trong tổ chức và có thể được sử dụng để ghi đè các quyền khác mà người dùng có thể đang giữ.

Chẳng hạn, người dùng trong bộ phận nhân sự có quyền truy cập dựa trên vai trò để truy cập vào khu vực trong nơi lưu giữ hồ sơ nhân sự có thể không được phép truy cập vào cũng nơi đó vào cuối tuần nếu nơi đó tuân theo quy tắc quy định rằng khu vực đó không thể truy cập ngoài giờ hành chánh.

RuBAC hầu thường được sử dụng kết hợp với các mô hình kiểm soát truy cập khác, nhất là các RBAC. Hệ thống kết hợp này cho phép quản trị viên cung cấp các mức bảo mật bổ sung để đáp ứng các rủi ro cụ thể.

Mô hình kiểm soát truy cập theo quy tắc

RuBAC hoạt động nhằm hạn chế quyền truy cập của người dùng trái phép trong khi cấp quyền truy cập cho những người được ủy quyền.

Bộ quy tắc truy cập được quản trị viên tạo ra và được tích hợp trong toàn bộ hệ thống kiểm soát truy cập. Khi người dùng cung cấp thông tin xác thực (thẻ truy cập, mã truy cập, khóa thông minh, điện thoại di động hoặc sinh trắc) cho hệ thống, thông tin sẽ được kiểm tra theo bộ quy tắc truy cập và người dùng được cấp hoặc từ chối quyền truy cập.

Một số bước quan trọng khi cài đặt hệ thống RuBAC gồm:

(1). Xem xét các quy tắc áp dụng cho một số điểm truy cập nhất định cũng như các quy tắc chung áp dụng cho tất cả các điểm truy cập. Các khu vực có rủi ro cao mà không có bất kỳ quy tắc cụ thể nào phải được xem xét thường xuyên để kịp vá các lỗ hổng bảo mật thay đổi liên tục.

(2). Xác định và phân tích các tình huống tiềm ẩn có thể yêu cầu các quy tắc bổ sung nhằm giảm thiểu rủi ro.

(3). Đặt ra các quy tắc mới hoặc cập nhật bộ quy tắc hiện có dựa trên kết quả phân tích đánh giá để tăng cường mức độ bảo mật.

(4). So sánh các quy tắc với các quyền truy cập được thiết lập bằng các mô hình kiểm soát truy cập khác, chẳng hạn như kiểm soát truy cập dựa trên vai trò - RBAC, để đảm bảo không có xung đột.

(5). Lập tài liệu và công bố bộ quy tắc để mọi người dùng biết về quyền truy cập của họ. Có thể bỏ qua các chi tiết nhưng điều quan trọng là người dùng phải hiểu các quy tắc và thay đổi chính sách có thể ảnh hưởng như thế nào đến công việc của họ.

(6). Thực hiện đánh giá thường xuyên, tiến hành kiểm tra một cách có hệ thống để xác định bất kỳ vấn đề hoặc lỗ hổng nào có thể có trong hệ thống và cập nhật bộ quy tắc nếu cần.

MỘT SỐ GIAO THỨC CHỨNG THỰC

Giao thức chứng thực dựa vào mật khẩu

Chứng thực người dùng

Thông thường, một người muốn sử dụng các dịch vụ do một hệ thống cung cấp, người dùng phải trải qua thủ tục đăng ký, ở đó, người dùng cung cấp một số thông tin định danh – Id (Identification) để phân biệt với các người dùng khác. Thông tin định danh phải là duy nhất trong hệ thống. Và để hệ thống có thể xác minh định danh của một người đúng là người có định danh người đó đã đăng ký với hệ thống hay không, hệ thống cần lưu giữ một số thông tin bí mật, đơn gian nhất là mật khẩu – password, của người dùng có định danh đó. Ta có thể gọi (Id, password) là tài khoản của người dùng trên hệ thống.

Thông tin mật khẩu là thông tin nhạy cảm của người dùng, vì thế, nó phải được lưu giũ bảo mật trên máy chủ hệ thống, để ngay cả khi dữ liệu tài khoản người dùng có bị đánh cắp thì kẻ xấu cùng không thể sử dụng được các thông tin trong các tài khoản lưu trên máy chủ. Kỹ thuật đơn giản nhất là lưu trị băm – hash value của mật khẩu thay vì lưu mật khẩu rõ. Như vậy, CSDL các tài khoản trên máy chủ có cấu trúc kiểu như: Account = {(Id, h(password)}, với h là một hàm băm mật mã nào đó, chẳng hạn như MD5 là hàm băm được dùng phổ biến trong các hệ quản trị CSDL.

Quá trình xác thực được mô tả trong Protyocol_1.

Protocol_1 #chứng thực người dùng

(1). Người dùng cung cấp thông tin tài khoản (Id, password) cho hệ thống.

(2). Hệ thống tìm trong CSDL các tài khoản người dùng tài khoản Acc = Account(Id,hp).

(3). Nếu không tồn tại tài khoản Acc trong hệ thống thì tiến hành thủ tục dăng ký; ngược lại, nó sẽ so sánh h với giá trị băm của password nó nhận được: hp == h(password) và tủy thuộc kết quả so sánh mà chấp nhận (accept) hay từ chố (reject) cấp quyền truy cấp cho người dùng đó.

Trong trường hợp dịch vụ được cung cấp trên mạng, như ngân hàng trực tuyến chẳng hạn, thống tin tài khoản (Id, password) phải được gửi đến máy chủ hệ thống. Để tránh rủi ro bị đánh cắp trên đường truyền, thông tin kênh truyền phải được bảo vệ. Có thể sử dụng công nghệ SSL – Secure Sockets Layer, một giao thức bảo mật tạo kênh mã hóa giữa máy chủ web và trình duyệt web, hoặc có thể sử dụng hệ mã khóa công khai để trao đổi thông tin cho máy chủ.

Chứng thực nhiều yếu tố

Trong nhiều ứng dụng quan trọng, nhất là những dịch vụ cung cấp trên mạng, như ngân hang trực tuyến hay thương mại điện tử, việc chứng thực người dùng bằng mật khổi thôi là chưa đủ, nhiều hệ thông trang bị chứng thêm một hoặc nhiều yếu tố định danh khác nữa, gọi là chứng thực nhiều yếu tố - Multi-Factors Authentication (MFA), chẳng hạn có thể sử dụng thêm thông tin sinh trắc học hay bất kỳ thông tin nào được cho là chỉ mỗi người cần chứng thực có như điện thoại hay email. Protocol_2 là giao thực sử dụng điện thoại thông minh để chứng minh them bằng cách hệ thống gửi một tin nhắn cho số điện thoại người dùng đã đăng ký cho hệ thông. Người đăng nhập phải nhập tin nhắn họ nhận được vào màn hình đăng nhập để xác minh định danh là chính họ. Tin nhắn phải nhập cho đúng này được gọi là mật khẩu dùng một lần – OTP (One Time Password).

Protocol_2 #mật khẩu 1 lần – OTP.

(1). Người dùng U cung cấp thông tin tài khoản (Id, password) cho hệ thống S.

(2). Hệ thống băm và đối sánh h(password) với trị băm mật khẩu tương ứng với định danh Id mà nó đã lưu trữ.

(3). Nếu đối sánh thành công, hệ thông gửi cho U một tin nhắn “nonce” (ngược lại, kết thúc tiến trình xác minh).

(4). Trong khoảng thời gian quy định, nếu người dùng nhập đúng nonce cho hệ thống, thì S sẽ cấp quyền truy cập cho U (ngược lại, chứng thực thất bại).

Trong trường hợp đơn giản, hệ thống không lưu trữ thông tin mật khẩu của người sử dụng, bước (2) có thể bỏ qua.

Chứng thực lẫn nhau

Hai giao thức chứng thực người dùng trên là các giao thức chứng thực 1 chiều, chỉ hệ thống xác minh người sử dụng. Người dùng giao thức này có thể bị đánh lừa đưa thông tin cho những hệ thống giả mạo tinh vi. Nhiều hệ thông cung cấp dịch vụ quan trọng trang bị giao thức chứng thực 2 chiều, ở đó, không chỉ hệ thống xác minh người dùng của nó mà phía ngược lại, người dùng cũng thực hiện xác minh máy chủ họ sắp làm việc có chính chủ. Những hệ thống chứng thực này được gọi chung là chứng thực lẫn nhau – Mutual Authentication.

Nếu như trong chứng thực 1 chiều, người sử dụng chia sẻ cho (máy chủ) hệ thống thông tin riêng của mình, thì trong chứng thực lẫn nhau, hệ thống cũng cần chia sẻ thông tin riêng của nó cho người sử dụng.

<u>Chứng thực lẫn nhau dựa trên mật mã khóa công khai</u>

Giao thức sau, Protocol_3, sử dụng mật mã đối xứng cho chứng thực lẫn nhau. Trong giao thức này, giả sử người sử dụng U và hệ thống S chia sẻ chung với nhau khóa bí mật k của một hệ mã đối xứng E.

Protocol_3 #chứng thực lẫn nhau

(1). Để đăng nhập vào hệ thống, người sử dụng U chuyển cho hệ thống S tên định danh Id_u của mình cùng với bản mã $c = E_k(nonce)$ của một số ngẫu nhiên nonce sử dụng hệ mã E với khóa bí mật k.

(2). Hệ thống S sử dụng định danh Id_u để lấy khóa bí mật k của U mà nó đã lưu trữ và gửi lại cho U bản mã hóa $c' = E_k(D(nonce) + sk)$ của giá trị nonce + sk, với sk là số ngẫu nhiên do hệ thống chọn. Trong đó D là trình giải mã tương ứng với trình mã hóa E.

(3). Người dùng U kiểm tra, nếu $E_k(D(c')) == nonce + sk$ thì gửi lại cho hệ thống giá trị nonce; nếu không thì kết thúc và từ chối giao dịch với hệ thống.

(4). Nếu hệ thống S nhận được nonce thì nó cấp quyền truy cập hệ thông cho U, ngược lại, chứng thực không thành công và kết thúc.

Giá trị sk do hệ thống sinh ra, có thể được sử dụng làm khóa bí mật để trao đổi thông tin mật trong phiên giao dịch này, gọi là khóa phiên – session key.

Vấn đề khó khăn khi cài đặt Protocol_3 là làm sao để hệ thống quản lý an toàn khóa bí mật của các người sử dụng. Không thể lưu trữ giá trị băm của khóa bí mật người dùng vì khi ấy, hệ thống sẽ không thể thực hiện bước (2). Vấn đề này được gọi chung là quản lý khóa – key management, và có thể có nhiều cách giải khác nhau.

<u>Chứng thực lẫn nhau dựa trên mật mã khóa công khai</u>

Cách đơn giản cho quản lý khóa là không giữ khóa bí mật của nhau. OTP là ví dụ cho chứng thực 1 chiều. Trong trường hợp chứng thực lẫn nhau, có thể sử dụng thông tin khóa công khai của nhau để chứng thực. Giả sử hệ thống vẫn lưu trị băm của mật khẩu người dùng. Giao thức sau chứng thực lẫn nhau dựa trên khóa công khai, es của máy chủ và eu của người dùng, đã được công bố trước. Ký hiệu E và D lần lượt là hàm mã hóa và giải mã hệ mã khóa công khai; và h là hàm băm mật mã.

Protocol_4 #chứng thực sử dụng khóa công khai

(1). Người dùng U gửi thông tin tài khoản $Inf_u = (Id, c) = (Id_u, E_{es}(password))$ với mật khẩu được mã hóa dùng khóa công khai của hệ thống S.

(2). Hệ thống giải mã c và thực hiện hăm để so sánh với trị băm hp tương ứng với Id_u: $h(D_{ds}(c)) == hp$. Nếu thành công, S gửi cặp thông tin ($hp' = h(password||sk)$, $c' = E_{eu}(sk)$) cho U (nếu không thành công, tiến trình xác minh kết thúc).

(3). U kiểm tra $h(password||D_{du}(sk)) == hp'$, nếu thành công thì sử dụng gửi $h'' = h(sk)$ cho S; ngược lại, kết thúc xác minh.

(4). S kiểm tra $h'' == h(sk)$ thì cấp quyền truy cập cho U, và cả 2 có thể sử dụng sk như khóa phiên; ngược lại, xác minh thất bại.

MỘT SỐ KỸ THUẬT BẢO MẬT VÀ QUẢN LÝ KHÓA

Chứng thực lẫn nhau dùng mã công khai không lưu trữ thông tin khóa cá nhân

Ta thấy, sử dụng hàm băm mật mã là kỹ thuật đơn giản nhất để bảo mật mật khẩu. Giao thực chứng thực lẫn nhau kết hợp giữa bàm băm mật mã (để bảo mật mật khẩu trên mấy chủ hệ thống), và khóa công khai (để chứng thực lẫn nhau) có thể tránh quản lý khóa bí mật trên máy chủ. Tuy nhiên, vẫn còn rủi ro khi khóa cá nhân của hệ thống lưu giữ trên máy chủ (nếu không, không thể tự động hoàn toàn được quy trình chứng thực vì phải cung cấp khóa cá nhân của máy chủ cách tủ công). Giải pháp thiết lập khóa bằng hệ mã Diffie-Hellman có thể giải quyết vấn đề này. Ta thử mở rộng Protocol_4 với giao thức thiết lập khóa của Diffie-Hellman.

Protocol_5 #chứng thực và thiết lập khóa phiên

(0). Hệ thống công khai số nguyên g, p với, g là phần tử sinh của $\mathbb{Z}_p$ và p là một số nguyên tố lớn.

(1). Người dùng U gửi cho hệ thống S thông tin Inf_u = (Id, u) = (Id_u, g^x mod p) với x là số ngẫu nhiên trong $\mathbb{Z}_p$.

(2). Hệ thống S cũng gửi cho U thông tin giá trị s = g^y mod p, w = hp*s^y mod p, với y là số ngẫu nhiên trong $\mathbb{Z}_p$, và hp là trị băm của mật khẩu password người dùng U đã đăng ký với S.

(3). Người dùng U kiểm tra, nếu hp == h(password)*u^x mod p thì gửi cho S giá trị c = password*s^x mod p, với h là hàm băm mật mã; ngược lại, kết thúc tiến trình.

(4). Hệ thống S kiểm tra, nếu h(c*$(s^y)^{-1}$) == hp thì cấp quyền truy cập cho U; ngược lại, tiến trình kết thúc.

Trong Protocol_5, $sk \equiv s^x \equiv u^y (mod\ p)$ có thể được dùng làm khóa phiên.

Chia sẻ bí mật chung

Mô hình ngưỡng (k, n)

Mô hình ngưỡng – **(k, n)-threshold**, được phát biểu như sau.

(k, n)-threshold là phương pháp chia sẻ bí mật S giữa n thành viên P_1, …, P_n, thỏa các tính chất sau:

. k < n. k được gọi là ngưỡng.

. Mỗi thành viên P_i giữ một bí mật thành phần I_i, i = 1, 2, …, n.

. Cần tối thiểu k thành viên cung cấp tối thiểu k bí mật trong tập I = {I_1, I_2, …, I_n} để phục hồi lại S.

. Ít hơn k thành viên cung cấp bí mật thì không thể phụ hồi được S.

Phương pháp nội suy Lagrange

Có nhiều cách để cài đặt mô hình ngưỡng (k, n)-threshold, ở đây chúng ta sử dụng kỹ thuật nội suy Lagrange.

Đa thức

Ta biết rằng, trong mặt phẳng Oxy,

. Qua 2 điểm (x_1, y_1), (x_2, y_2), xác định duy nhất đường thẳng $y = ax + b$.

. Qua 3 điểm (x_1, y_1), (x_2, y_2), (x_3, y_3), xác định duy nhất một parabol $y = ax^2 + bx + c$.

. Qua 4 điểm (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) xác định duy nhất một đa thức bậc 3 $y = ax^3 + bx^2 + cx + d$

Một cách tổng quát, qua n + 1 điểm $\{(x_0, y_0), (x_1, y_1),\ldots, (x_n, y_n)\}$xác định duy nhất một đa thức bậc n: $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$

<u>Công thức nội suy Lagrange</u>

Xét đa thức bậc n có dạng

$f(x) = A_0 \prod_{i=0,i\neq 0}^{n}(x - x_i) + A_1 \prod_{i=0,i\neq 1}^{n}(x - x_i) + \cdots A_j \prod_{i=0,i\neq j}^{n}(x - x_i) + \cdots +$
$A_{n-1} \prod_{i=0,i\neq n}^{n}(x - x_i)$.

. $x = x_0, f(x_0) = y_0 = A_0(x_0 - x_1)(x_0 - x_2) \ldots (x_0 - x_n)$

$\rightarrow A_0 = \frac{y_0}{(x_0-x_1)(x_0-x_2)\ldots(x_0-x_n)}$.

. $x = x_1, f(x_1) = y_1 = A_1(x_1 - x_0)(x_1 - x_2) \ldots (x_1 - x_n)$

$\rightarrow A_1 = \frac{y_1}{(x_1-x_0)(x_1-x_2)\ldots(x_1-x_n)}$.

…

. $x = x_n, f(x_n) = y_n = A_n(x_n - x_0)(x_n - x_1) \ldots (x_n - x_{n-1})$

$\rightarrow A_{n-1} = \frac{y_n}{(x_n-x_0)(x_n-x_1)\ldots(x_n-x_{n-1})}$.

Vậy,

$$f(x) = \frac{(x-x_1)(x-x_2)\ldots(x-x_n)}{(x_0-x_1)(x_0-x_2)\ldots(x_0-x_n)}y_0 + \frac{(x-x_0)(x-x_2)\ldots(x-x_n)}{(x_1-x_0)(x_1-x_2)\ldots(x_1-x_n)}y_1 + \cdots + \frac{(x-x_1)(x-x_2)\ldots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\ldots(x_n-x_{n-1})}y_n,$$

hay

$f(x) = A_0L_0(x_0, y_0) + A_1L_1(x_1, y_1) + \cdots + A_{n-1}(x_n, y_n)$, (*)

với

$A_i = \frac{y_i}{(x_i-x_0)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_1-x_n)}, i = 0, 1, \ldots, n$, (**)

và

$L_i(x_i, y_i) = (x - x_0) \ldots (x - x_{i-1})(x - x_{i+1}) \ldots (x - x_n), i = 0,1, \ldots, n.$ (***)

Giao thức (k, n)-threshold

Thiết lập bí mật riêng phần

Để tạo bí mật riêng phần cho bí mật chung S, quản trị hệ thống thực hiện:

(1) Trước hết, chọn 1 số nguyên tố lớn p.

(2) Chọn ngẫu nhiên k – 1 số $a_i \in \mathbb{Z}_p, i = 1, 2, \dots, k-1$.

(3) Tính và cấp cho thành viên P_i cặp $(x_i, y_i), i = 1, \dots, n$, với các x_i phân biệt và $y_i \equiv S + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1} \pmod p$.

Phục hồi bí mật chung

Để phục hồi bí mật chung, cần ít nhất k thành viên cung cấp các bí mật riêng $\{(x_1, y_1), \dots, (x_K, y_K)\}, K \geq k$, và thực hiện

(1) Tính K hệ số A_i theo công thức (*)

(2) Tính K đa thức $L_i(x_i, y_i)$ theo công thức (**)

(3) Xác định đa thức $f(x)$ theo công thức (*) và tính $S = f(0)$.

2 BẢO MẬT CƠ SỞ DỮ LIỆU

Như ta đã biết, Cơ sở Dữ liệu (CSDL) là trái tim của một Hệ thống Thông tin (HTTT). Dữ liệu trong CSDL không chỉ cung cấp nguồn cho các tính toán khai thác thông tin của HTTT mà trong nhiều trường hợp, dữ liệu còn là tài sản của đơn vị hay tổ chức sở hữu nó.

Kiểm soát truy cập – AC (Access Control) cung cấp các cơ chế bảo vệ HTTT nói chung vào dữ liệu trong CSDL bằng cách xác minh và chỉ cấp quyền truy cập và dữ liệu nào được truy cập cho các thực thể đã được xác minh. Triển khai AC giống như thiết lập "tường rào" bảo vệ CSDL không bị khai thác bất hợp pháp. Tuy nhiên, có khả năng kẻ xấu có thể vượt qua được tường rào bảo vệ do AC thiết lập và khai thác dữ liệu bắt hợp pháp. Mã hóa nếu được sử dụng thì ngay cả khi kẻ xấu thâm nhập được vào CSDL và lấy đi dữ liệu thì cũng chỉ sử dụng được dữ liệu đã đánh cắp nếu giải mã được dữ liệu đã mã hóa.

Vấn đề là CSDL được tạo cho mục đích tìm kiếm, vì thế, nếu dữ liệu trong CSDL được mã hóa thì nhiệm vụ tìm kiếm không thể hoàn thành. Thực vậy, nếu có thể tìm kiếm trên dữ liệu đã mã hóa như tìm kiếm trên dữ liệu không mã hóa thì hệ mã xem như không đáng tin. Đây chính là điểm gút của bảo mật nội dung CSDL. Giải pháp của ta là dung hòa giữa hiệu năng tìm kiếm và tính riêng tư của dữ liệu. Giải pháp nằm ở thiết kế CSDL cho mục đích tìm kiếm và thiết kế các giao thức thao tác trên CSDL được mã hóa – Encrypted Database.

CÁC THÀNH TỐ MẬT MÃ CƠ BẢN

Trước khi học cách bảo mật CSDL, chúng tôi giới thiệu các công cụ mã hóa hiện đại có thể áp dụng cho tìm kiếm dữ liệu trong CSDL được mã hóa, gọi chung là mã hóa cho phép tìm kiếm – searchable encryption (SE).

Số dư Trung Hoa

Gọi D là tập hợp

D = $\{f_1, f_2, \ldots, f_n\}$

n thành phần dữ liệu có cùng cấu trúc. Nếu

$p_1, p_2, \ldots, p_n$

là n số nguyên tố phân biệt, và nếu

$p_1 > f_1, p_2 > f_2, \ldots, p_n > f_n$.

Đặt

$P \equiv p_1 p_2 \ldots p_n$,

thì

$P_i = \frac{P}{p_i}, \forall i = 1,2, \dots, n,$

là n số nguyên dương. Hơn nữa M_i, m_i nguyên tố cùng nhau, $\gcd(P_i, p_i) = 1$ nên tồn tại N_i sao cho $P_i N_i \ mod \ p_i = 1$, với mọi i = 1, 2, …, n.

Đặt

$e_i = P_i N_i, \forall i = 1, 2, \dots, n,$

và tính

$C \equiv \sum_{i=1}^{n} e_i f_i \ \ (mod \ M).$

Thì $C \ mod \ p_i = f_i$

Thực vậy,

$C \ mod \ p_i = e_1 f_1 \ mod \ p_i + \cdots + e_i f_i \ mod \ p_i + \cdots + e_n f_n \ mod \ p_i$

$= P_1 N_1 f_1 \ mod \ p_i + \cdots + P_i N_i f_i \ mod \ p_i + \cdots P_n N_n f_n \ mod \ p_i$

$= \ p_1 \dots p_i \dots p_n N_i f_1 \ mod \ p_i + \cdots + 1 f_i + \cdots + p_1 \dots p_i \dots p_{n-1} N_i f_{n-1} \ mod \ p_i$

$= 0 + \cdots + f_i + \cdots + 0$

$= f_i.$

Đây chính là hệ quả suy trực tiếp từ định lý số dư Trung Hoa – CRT (Chinese Remainder Theorem) và có thể áp dụng bảo mật CSDL cho n (nhóm) người dùng khác nhau. Mỗi (nhóm) người được cấp cho khóa p_i để chỉ đọc được chính dữ liệu f_i của (nhóm) mình. Trong thực tế, f_i, i = 1, 2, …, n, các mục tin (item) dữ liệu nhạy cảm cần bảo mật. Chẳng hạn, tiền lương trong một đơn vị là dữ liệu nhạy cảm có thể được mã hóa để chỉ lương ai nhân viên đó mới đọc được. Nếu số nhân viên không nhiều, có thể sử dụng phương pháp này để bảo mật bảng lương. Trong trường hợp số nhân viên nhiều, áp dụng CRT trực tiếp có thể không hiệu quả, giải pháp có thể là phân cấp hay/và gom nhóm các đối tượng. Chẳng hạn $C_1, \dots C_p$ là mã hóa bảng lương của phòng ban $j, j = 1,2, \dots, p$.

Hàm băm đồng cấu

Hàm băm mật mã (cryptographic hash function) $H: \{0,1\}^* \to \{0,1\}$ được gọi là đồng cấu – homomorphic, với phép toán 2-ngôi $\oplus$: $X \oplus Y$, nếu với 2 dữ liệu phân biệt có biểu diễn nhị phân $X = x_0 x_1 \dots x_{(n-1)}, Y = y_0 y_1 \dots y_{(n-1)}$, và

$h_X = H(X)$

$h_Y = H(Y),$

lần lượt là trị băm của D_1 và D_2, thì

$h_{XY} = h_X \circ h_Y = H(X \oplus Y)$

là giá trị băm của dữ liệu tổng hợp X và Y theo phép $\oplus$, và $\circ$ là phép toán 2-ngôi trên các giá trị băm.

Ví dụ, cho

$H: \{0,1\}^n \to \{0,1, \dots, p\}$

được định nghĩa bởi

$H(D) = H(d_0 d_1 \dots d_{n-1}) = g^{\Sigma_{i=0}^{n-1} 2^i d_i} \ mod \ p,$

với g là phần tử sinh của số nguyên tố p có *n* bit nhị phân, nghĩa là mọi số z trong tập $\mathbb{Z}_p^* = \{1, \dots, p-1\}$ luôn có một số $v \in \mathbb{Z}_p^*$ sao cho $z = g^v \ mod \ p$, thì H đồng cấu với phép cộng số học của 2 số nguyên nhị phân:

$X \oplus Y = \left(\Sigma_{i=0}^{n-1} 2^i x_i\right) + \left(\Sigma_{i=0}^{n-1} 2^i y_i\right).$

Thực vậy, nếu

$x = \Sigma_{i=0}^{n-1} 2^i x_i$

và

$y = \Sigma_{i=0}^{n-1} 2^i y_i$,

lần lượt là giá trị của 2 số nguyên nhị phân $X = x_0 x_1 \dots x_{n-1}$ và $Y = y_0 y_1 \dots y_{n-1}$, thì

$H(X) \circ H(Y) \ mod \ p = H(x) H(y) \ mod \ p$

$= g^x g^y \ mod \ p$

$= g^{x+y} \ mod \ p$

$= g^{X \oplus Y} mod \ p$

$= H(X \oplus Y).$

Hàm băm mật mã có tính đồng cấu có thể được sử dụng trong bảo mật CSDL. Chẳng hạn, trong giao thức kiểm soát truy cập bằng username-password, sử dụng hàm băm đồng cấu để lưu $password$ ở (máy chủ) hệ thống, $H(password)$, thì thay vì chuyển cho hệ thống $password$, người dùng có thể đăng nhập bằng $H(-password)$ và kiểm tra bằng cách so sánh với $H(0)$, với $0 = X \oplus -X$.

Mã đồng cấu

Nếu như hàm băm đồng cấu là hàm một chiều, chỉ mã hóa và tính toán trên trị băm đã mã hóa theo phép đồng cấu tương ứng mà không thể giải mã, thì mã đồng cấu ngoài việc cho phép tính toán trên các bản mã, còn cho phép giải mã.

Mã đồng cấu cho phép tạo ra bản mã kết hợp của hai số, chỉ dựa vào bản mã của từng bản rõ mà không cần giải mã. Gọi bản mã – cipher text, của bản rõ – plaintext u là

[[u]] = E(u).

Trong đó E là hàm mã hóa và phép đồng cấu ⊕, thỏa: với mọi bản rõ u và v ta có:

[[u]] ∘ [[v]] = [[u ⊕ v]].

Ví dụ, gọi (e, g, p) là khóa công khai của hệ mã Elgmal, và $c_1, c_2 \in \mathbb{Z}_n^2$ lần lượt là 2 bản mã của 2 bản rõ $m_1, m_2 \in \mathbb{Z}_n$:

$[[m_1]] = c_1 = (x_1, y_1) = (m_1 e^{r_1} \bmod p, g^{r_1} \bmod p)$,

$[[m_2]] = c_2 = (x_2, y_2) = (m_2 e^{r_2} \bmod p, g^{r_2} \bmod p)$.

Thì $c_1 \circ c_2 = (x_1 * x_2, y_1 * y_2) = (m_1 m_2 e^{r_1 + r_2} \bmod p, g^{r_1 + r_2} \bmod p) = [[m_1 \oplus m_2]]$.

Hệ mã đồng cấu có thể sử dụng để tìm kiếm trong CSDL trên các thuộc tính đã mã hóa. Ví dụ, nếu lưu trữ $[[X]] = (x, y) = (Xe^r \bmod p, g^r \bmod p)$ bằng hệ mã ElGamal chẳng hạn, thì có thể để tìm được X cần gửi $[[X^{-1}]] = (x', y') = X^{-1} e^u \bmod p, g^u \bmod p)$ cho (máy chủ) hệ thống kiểm tra$x * x' = Xe^r * X^{-1} e^u = (X * X^{-1}) e^{r+u} == y * y'$.

RSA và Paillier cũng là các hệ thống mã đồng cấu phổ biến. Chúng tôi giới thiệu đây hệ mã Paillier vì trước hết, nó rất thích hợp với yêu cầu bảo mật CSDL, sau nữa, nó còn có những tính chất tựa-đồng cấu – homomorphic-like.

Mã đồng cấu cộng

Sơ đồ mã hóa Paillier là một hệ mật mã đồng cấu cộng gồm ba thuật toán ngẫu nhiên sau.

KeyGen()

(1) Chọn 2 số nguyên tố thỏa: $p \neq 1$, $\gcd(pq, (p-1)(q-1)) = 1$.

(2) Tính $n = pq$.

(3) $\lambda = lcm(p-1, q-1)$.

(4) Chọn $g \in \mathbb{Z}_{n^2}$: $\gcd(gmn^2) = 1$.

(5) Tính $\mu = L\left(g^\lambda \bmod n^2\right)^{-1} \bmod n$. Trong đó, $L(x) = (x-1)/n$.

(6) Trả về khóa công khai $e = (n, g)$; và khóa cá nhân $d = \lambda$.

Encryption(m)

(1) Chon ngẫu nhiên $r \in \mathbb{Z}_n$.

(2) Tính $c = g^m r^n \bmod n^2$.

(3) Trả về bản mã c.

Decryption(λ, c)

(1) Tính $\mu = L\left(g^{\lambda} mod\ n^2\right)^{-1}\ mod\ n$.

(2) Tính $m = \mu L\left(c^{\lambda}\ mod\ n^2\right)\ mod\ n$.

(3) Trả về bản rõ m.

Ví dụ, với

$p = 13, q = 17$, thì $n = 13 \times 17 = 221$ và $\lambda = lcm(12,17) = 48$.

Với bản rõ $m_1 = 123$.

Giả sử $g = 4886, r_1 = 666$, thì $c_1 = 4888^{123} 666^{221}\ mod\ 221^2 = 25889$.

Tính đồng cấu của Paillier

<u>Đồng cấu cộng</u>

Nếu

$c_1 = \left[[m_1]\right] = g^{m_1} r_1^n\ mod\ n^2$,

và

$c_2 = \left[[m_2]\right] = g^{m_2} r_2^n\ mod\ n^2$,

thì

$c_1 * c_2 = g^{m_1+m_2}(r_1 r_2)^n\ mod\ n^2 = [[m_1 + m_2]]$.

Ví dụ, với

$m_2 = 87, r_2 = 999$,

thì cùng cách tính như ví dụ trên, $c_2 = 30692$.

Đặt

$c_{12} = c_1 \times c_2 = 25889 \times 30692 = 39800\ mod\ 221\text{^}2$.

Giải mã ta được

$m_{12} = 160 = 123 + 87 = (m_1 + m_2)\ mod\ 221$.

<u>Tựa đồng cấu cộng</u>

Bây giờ, nếu

$c = \left[[m]\right] = g^m r^n\ mod\ n^2$,

và

$c^k = (g^m r^n)^k\ mod\ n = g^{km}(r^k)^n\ mod\ n^2 = \left[[k * m]\right] = [[m + \cdots + m]]$.

Đẳng thức cuối cùng cho thấy ta có thể tính được bản mã tich km, ký hiệu

$k[[m]]$,

bằng cách nâng lũy thừa k bản mã $[[m]]$:

$k[[m]] = [[m]]^k \ mod \ n^2$.

Ví dụ, với

$k = 25$,

tính

$c = (m_1)^k \ mod \ n^2 = 25899^{25} \ mod \ 221^2 = 15723$.

Giải mã ta được

$m = 202 = 25 \times 123 \ mod \ 221 = k \cdot m_3 \ mod \ n$.

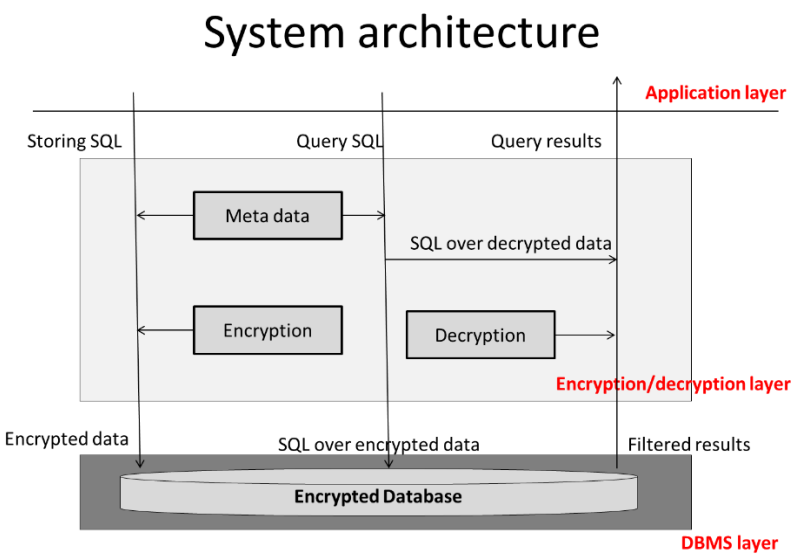
Rõ ràng, sử dụng Paillier, việc tìm kiếm trong CSDL mã hóa trở nên đơn giản, dễ hiểu hơn với ElGamal ta đã ví dụ trên. Thực vậy, để tìm X đã được lưu trong (máy chủ) hệ thống dạng $[[X]]$, chỉ đơn giản $[[X \oplus -X]] == 1$ khi người dùng gửi $[[-X]]$, với $-X = X + n$.

CHỈ MỤC AN TOÀN

Có thể sử dụng các thành tố mật mã để bảo mật dữ liệu trong CSDL và thiết kế các giáo thức tìm kiếm trên các dữ liệu đã mã hóa này. Tuy nhiên, việc tìm kiếm như thế sẽ phải trả giá về hiệu năng tìm kiếm. Hơn nữa, kết quả tìm duy nhất, mặc dù trả về dữ liệu đã mã hóa cũng gặp nhiều rủi ro trước những tấn công. Ví thế, thay vì mã hóa và tìm trực tiếp trên dữ liệu đã mã hóa, ta tạo thêm dữ liệu phụ phục vụ tìm kiếm, gọi là meta-data, dữ liệu quản lý dữ liệu. Kỹ thuật đó được gọi là chỉ mục an toàn – secure index, trong mô hình CSDL bảo mật 2-pha, như minh họa trong hình dưới.

Mô hình tìm kiếm 2-pha

Kiến trúc hệ thống



Trong mô hình 2-pha, hệ thống được kiến trúc gồm 3 lớp

(1). Lớp hạ tầng hay lớp quản trị CSDL – DBMS Layer, là này triển khai một hệ quản trị CSDL ở hạ tầng hệ thống để lưu trữ và thao tác trên dữ liệu trong CSDL.

(2). Lớp bảo mật – Encryption/Decryption Layer, là lớp sẽ được người làm cung cấp thêm các thành tố mật mã để người thiết kế các HTTT có thể triển khai các giao thức bảo mật CSDL.

(3). Lớp ứng dung – Application Layer, là lớp giao tiếp trực tiếp với người dùng hoặc các ứng dụng khai thác CSDL bảo mật.

Lưu trữ CSDL cho mô hình 2-pha

Mô hình 2-pha – 2-phrase Scheme, nói đến 2 pha chính của tìm kiếm trên CSDL mã hóa. Tuy nhiên, để có thể tìm kiếm hiệu quả trên CSDL mã hóa, ngoài thiết kế hợp lý, việc tổ chức dữ liệu đảm bảo tính bảo mật và có thể tìm kiếm là quan trọng. Trước hết, ta sẽ sử dụng khái niệm CSDL quan hệ – Relationship Database, theo đó, CSDL là tập các bản ghi – Record lưu trữ các thuộc tính – Attribute của đối tượng cần được quản lý. Và ta quy ước một số ký hiệu sau.

<u>Ký hiệu</u>

(.) R, ký hiệu CSDL rõ, không mã hóa.

(.) A^E, ký hiệu thuộc tính được mã hóa.

(.) I^A, ký hiệu dữ liệu được tạo thêm phục vụ cho việc tìm kiếm trên thuộc tính dữ liệu đã được mã hóa A.

(.) $R^E = R(A_1, \dots, A_s^E, \dots, A_n, I^{A_s})$, ký hiệu CSDL đã được mã hóa thuộc tính A_s.

Dữ liệu tạo thêm I^A, gọi là chỉ mục an toàn – Secure Index, và được thực hiện trên giá trị thuộc tính cần bảo mật A trong lúc mã hóa A thành A^E. Việc tại chỉ mục an toàn tùy thuộc kiểu dữ liệu

của thuộc tính nhạy cảm A cần mã hóa. Ta tập trung vào 2 kiểu dữ liệu chính là dữ liệu dạng chuỗi ký tự – Char, và kiểu dữ liệu số – Numeric.

Chỉ mục trên dữ liệu chuỗi

Đặt $S = s_1 s_2 \dots s_{i-1} s_i s_{i+1} \dots s_n$, với $s_i, i = 1,2, \dots, n$ là các ký tự. Ta định nghĩa hàm mã cặp – pair code, ký hiệu PC, dựa trên hàm băm mật mã

$H: \{0,1\}^* \to \{0,1, \dots, p-1\}$,

như sau.

Đinh nghĩa. Hàm mã cặp là hàm mã hóa chuỗi ký tự thành chuỗi p bit, là hàm

$PC: D \to \{0,1\}^p$,

với D là tập các ký tự có thể có. PC ánh xạ chuỗi n ký tự $S = s_1 s_2 \dots s_{j-1} s_j s_{j+1} \dots s_n$, thành chuỗi p bit $I = i_0 i_1 \dots i_{p-1}$có khởi tạo gồm tất cả các bit 0, và sau đó, bit $i_j, j = 0, \dots, p-1$, được bật thành 1 nếu tồn tại cặp ký tự $s_{k-1} s_k, k = 2, \dots, n$, sao cho $H(s_{k-1} s_k) = j$.

Chỉ mục trên dữ liệu số

Trước tiên, ta sẽ làm việc trên số nguyên nhị phân w bit, là chuỗi w bit biểu diễn các giá trị trong tập $\{0,1, \dots, 2^w - 1\}$. Các số thực chấm động là cặp 2 số nguyên (và thêm bit dấu nếu cần) gồm số và số mũ. Chẳng hạn, xấp xỉ số π được biểu diễn là (31416, -4) tương ứng với giá trị $31416 * 10^{-4} = 3.1416$ (cơ số 10 là quy ước nên không cần biểu diễn).

Định nghĩa (k-prefix). Một chuỗi w bit được gọi là $(k, w) - prefix(p)$ nếu k bit đầu là các giá trị cố định p cho trước, gọi là $prefix$, và $w - k$ bit còn lại nhận giá trị tự do trong $\{0,1\}$.

Ví dụ, $(2,5) - prefix(11)$, là chuỗi $\mathbf{11} ***$. Chuỗi này biểu diễn tập

$\{\mathbf{11}000, \mathbf{11}001, \mathbf{11}010, \mathbf{11}011, \mathbf{11}100, \mathbf{11}101, \mathbf{11}110, \mathbf{11}111\}$.

Định nghĩa (họ prefix). Họ prefix $F(x), x \in \mathbb{N}$, là tập các prefix, xác định bởi

$F(x) = F(x_1 x_2 \dots x_w) = \{(k, w) - prefix(x_1 \dots x_k)\}$.

Ví dụ, F(12) = F(1100) = {1100, 110*, 11**, 1***, ****}.

Hệ luận 1. Cho số nguyên x và $prefix\ P$, ta có $x \in P \Leftrightarrow P \in F(x)$.

Để biểu diễn đoạn $\{a, a+1, \dots, b\}$, ta định nghĩa khái niệm hàm khoảng mã $S([a, b])$ như sau.

Định nghĩa (khoảng mã). Khoảng mã $S([a, b])$ của tập các số nguyên liên tiếp $[b - a] = \{a, a+1, \dots, b\}$ là tập nhỏ nhất các $prefix\ P_i$ sao cho $U_i P_i$ biểu diễn được mọi phần tử trong $[b - a]$.

Ví dụ, $S([11,15]) = \{1101,11 **\}$; hay

$S([10, 14]) = \{1010,1011, 1100, 1101, 1110\} = \{10 **, 1100,1101,1110\}$.

Hệ luận 2. Số nguyên $x \in [b-a] \Leftrightarrow F(x) \cap S([a,b]) \neq \emptyset$.

Định nghĩa (số hóa). Số hóa $prefix\ P$, ký hiệu $N(P)$ là chuỗi bit sao cho mọi cặp $prefix\ P, P'$, nếu $P = P'$ thì $N(P) = N(P')$, và ngược lại.

Có nhiều cách để cài đặt hàm số hóa. Sau đây là một cách.

Hàm số hóa. Cho P là $(k,w) - prefix(b_1 \dots b_k) = b_1 \dots b_k * \dots *$, hàm $N(P)$ là chuỗi $w+1$ bit với *prefix* là $k+1$ bit $b_1 \dots b_k 1$, hay $N(P) = (k+1, w+1) - prefix(b_1 \dots b_k 1)$.

Ví dụ, $N(10\ **) = 111\ **$, $N(101\ ***) = 1011\ ***$.

Hệ luận 3. Số nguyên $x \in [b-a] \Leftrightarrow N(F(x)) \cap N(S[a,b]) \neq \emptyset$.

<u>Lưu trữ</u>

Trong quan hệ bảo mật

$R^E = (A_1, \dots, A_s, \dots, A_n, I^{A_s})$,

khi thêm 1 bản ghi rõ – plain-record

$r = (a_1, \dots, a_s, \dots, a_n)$,

vào quan hệ R^E, r được biến đổi thành

$r^E = (a_1, \dots, E(a_s), \dots, a_n, I^{a_s})$,

trước khi thêm (append hay insert) vào R^E, trong đó E là một hàm mã, thường là mã đối xứng, và I^{a_s} là chỉ mục an toàn – secure index, cho phép tìm kiếm trên dữ liệu đã mã hóa $E(a_s)$. Chỉ mục I^{a_s} được tính tùy thuộc kiểu dữ liệu của thuộc tính nhạy cảm A_s là chuỗi hay số, như đã trình bày ở trên.

Tìm kiếm 2-pha

Với câu truy vấn Q, tìm kiếm 2-pha trên CSDL quan hệ $R^E = (A_1, \dots, A_s, \dots, A_n, I^{A_s})$ theo thuộc tính nhạy cảm A_s là tiến trình gồm 2 pha:

Pha 1 (tìm kiếm trên CSDL mã hóa). Trong pha này,

(1). Trước hết, truy vấn Q được biến đổi thành truy vấn mã hóa Q^E bằng cách chuyển điều kiện trong truy vấn gốc Q trên thuộc tính A_s thành điều kiện tìm kiếm trên Q^E trên chỉ mục an toàn I^{A_s}.

(2). Sử dụng các công cụ hệ quản trị CSDL hạ tầng hỗ trợ, thực hiện truy vấn Q^E như truy vấn thông thường. Kết quả sẽ là $r^E \subset R^E$ tập tất cả các bản ghi trong CSDL mã hóa thỏa truy vấn mã Q^E.

(3). Chuyên r^E cho pha 2.

Pha 2 (tìm kiếm trên tập con CSDL rõ). Pha này thực hiện,

(4). Giải mã kết quả r^E nhận được từ pha 1 thành CSDL con các bản ghi rõ r.

(5). Thực hiện truy vấn rõ Q trên r để được kết quả cuối cùng r''.

(6). Trả kết quả r'' cho tầng ứng dụng.

Hàm biến đổi truy vấn

Việc biến đổi truy vấn rõ Q thành truy vấn mã Q^E trong bước (1) ở pha 1, tùy thuộc kiểu dữ liệu nhạy cảm A_s là chuỗi hay số.

<u>Truy vấn bảo mật trên dữ liệu chuỗi</u>

Để tìm kiếm trên dữ liệu đã mã hóa, điều kiện truy vấn Q trên dữ liệu rõ $a_s \in A_s$ được biến đổi sang điều kiện truy vấn Q^E trên chỉ mục an toàn I^{A_s}. Ta định nghĩa hàm biến đổi, ký hiệu Trans(Q), cho 3 trường hợp cơ bản.

(I). Q ≡ $R.A_s = S$ (So sánh "=")

Trans($R.A_s = S$):

. $S^E = PC(S)$.

. return S^E.

Và áp dụng truy vấn $Q^E \equiv R^E.I^{A_s} = S^E$

(II). Q ≡ $not(R.A_s = S)$ (So sánh "≠")

. Tính $S^E = Trans(R.A_s = S)$ theo thủ tục (I).

. Và áp dụng truy vấn $Q^E \equiv not(R^E.I^{A_s} = S^E)$.

(III). Q ≡ $S\ like\ R.A_s$ (Tìm chuỗi con trong chuỗi lớn)

Trans($S\ like\ R.A_s$):

. $C = PC(S)$.

. $r^E = \{\}$

. $\forall r'' \in R^E$:

. $t = True$

. $for\ i\ in\ range(len(C))$:

. $if\ C[i] == 1\ and\ R^E.I^{A\_s} == 0$:

. $t = False$

. break

$if\ t: r^E.append(r'')$

. $return\ r^E$

Bài tập: cài đặt các truy vấn với biểu thức luận lý And, Or, XoR.

Truy vấn bảo mật trên dữ liệu số

Ví dụ mở đầu

Để dễ hình dung, ta bắt đàu bằng ví dụ sau. Truy vấn tìm giá trị 12 trong CSDL đã mã hóa. Trước hết, ta tìm đoạn tương ứng đã chia trước mà có chứa 12, giả sử đó là đoạn $[11, 15]$.

. Trong giai đoạn lưu trữ bảo mật giá trị a $\in [11,15]$, đoạn $[11,15]$ được lưu theo tiến trình.

(.) Tính S($[11,15]$) được $S = \{1011, 1***\}$.

(.) Số hóa các phần tử trong S, được NS $= \{10111, 11000\}$.

(.) Lưu NS trong CSDL bảo mật: $R^E.I^{A_S} = NS$.

. Dịch và thi hành câu truy vấn tìm giá trị 12.

(.) Trước tiên, biểu diễn nhị phân của $12 \equiv 1100$.

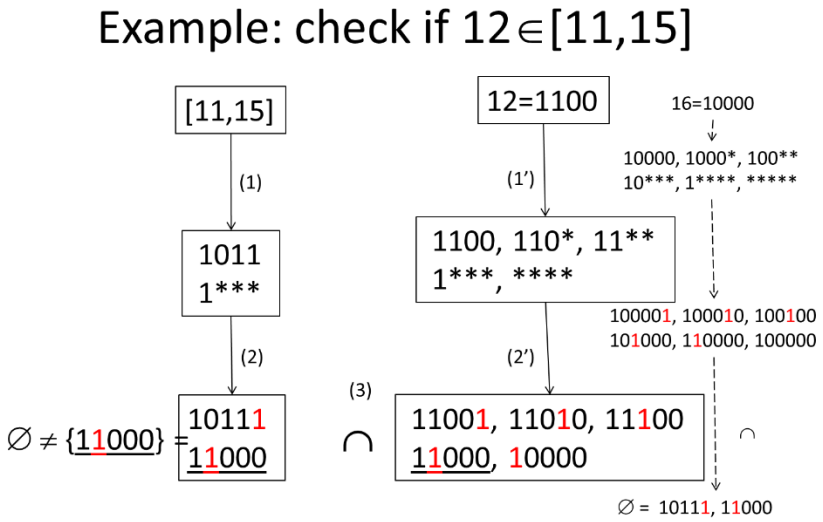
(.) Tính họ *prefix* cho 12: F $=$ F$(1100) = \{1100, 110*, 11**, 1***, ****\}$.

(.) Số hóa các phần tử trong F, được $NF = \{11001, 11010, 11100, 11000, 10000\}$

. Tìm các bản ghi trong CSDL đã mã hóa R^E.

(.) Với mọi bản ghi $r^E \in R^E$, nếu $r^E.I^{A_S} \cap NF \neq \emptyset$, thì chọn bản ghi này.

Hình sau minh hóa tiến trình lưu trữ và tìm kiếm của ví dụ trên.



Như vậy, để lưu trữ miền dữ liệu số $D = [a, b]$ cho thuộc tính A_s,

(.) Trước hết, ta phân hoạch D thành n đoạn $[d_i, d_{(i+1)}], i = 0,1, \dots, n$, với $d_0 = a$ và $d_{n+1} = b$.

(.) Sau đó, với mỗi đoạn $\left[d_i, d_{(i+1)}\right], i = 0,1, \dots, n$, ta gán nhãn t_i, chính là giá trị số hóa các khoảng mã này, $t_i = N\left(S\left[d_i, d_{(i+1)}\right]\right), i = 0,1, \dots, n$.

(.) Và lưu trữ dữ liệu $d \in \left[d_i, d_{(i+1)}\right], i = 0,1, \dots, n$ bằng cặp $A_s^E.a = E(d)$ và $I^{A_s} = t_i$.

Tuy nhiên, thực chất t_i cũng là tập rõ dạng nhị phân. Không khó để có thể suy ngược lại đoạn $\left[d_i, d_{(i+1)}\right], i = 0,1, \dots, n$. Nên ta cần bảo mật nội dung nhãn t_i này bằng mã chứng thực thông điệp – MAC (Message Authentication Code).

Mã chứng thực thông điệp và lọc Bloom

Mã chứng thực thông điệp $MAC_k(M)$ là một chuỗi nhị q bit định danh duy nhất cho thông điệp rõ $M \in \{0,1\}^*$ sao cho chỉ người có khóa bí mật k mới tạo ra được chuỗi $MAC_k(M)$ tương đương.

Mã chứng thực thông điệp có thể được cài đặt dễ dàng bằng hàm cách sử dụng một băm mật mã, chẳng hạn như MD5 hay SHA2, bằng cách ghép khóa bí mật vào thông điệp rõ. Kỹ thuật này còn được gọi là hàm-băm-có-khóa – Keyed hash function. Ta sẽ ký hiệu hàm băm có khóa k là H_k. Ví dụ, sử dụng hàm băm SHA2 ta có thể tạo ra mã chứng thực MAC cho thông điệp M với khóa bí mật k là: $MAC = SHA2(M||k)$, với $||$ là hàm nối chuỗi.

Ta cũng thấy, tìm kiếm trên chỉ mục số là thực hiện các pháp toán trên tập hợp. Với bài toán đảm bảo tính riêng tư cho dữ liệu tìm kiến, ta sẽ thiết kế hàm lọc Bloom theo kiểu hàm mã cặp – PC (Pairing Code) như đã dùng trong tạo chỉ mục an toàn cho chuỗi, như sau:

(.) Gọi $H_k: \{0,1\}^* \to \{0,1\}^p$ là hàm băm mật mã có khóa k, tạo ra một giá trị ngẫu nhiên trong tập $\{0, 1, \dots, p-1\}$.

(.) Chuỗi Bloom của tập hợp $S = \{s_1, \dots, s_q\}$ là chuỗi nhị phân p bit $B = b_0 b_1 \dots b_{p-1}$ trong đó $b_i = 1$ nếu $H_k(s_i) = i$. B ban đầu được khởi tạo là chuỗi các bit 0.

Như vậy, phần tử $s \in S$ nếu $B' = H_k(\{s\}) = b'_0 b'_1 \dots b'_{p-1}$ và nếu $b'_i = 1$ thì $b_i = 1, i = 0,1, \dots, p-1$.

Lưu trữ khoảng trên CSDL mã hóa

Khác với dữ liệu chữ, dữ liệu số được lưu theo từng tập, mỗi tập là một đoạn (range) $\left[d_i, d_{(i+1)}\right], i = 0,1, \dots, n$, với $a = d_0 < d_i < d_{i+1} < d_{n+1} = b$. Như vậy, với thuộc tính A_s, người dùng xác định miền dữ liệu $D = [a, b], a < b$, cho các giá trị thuộc tính $a \in A_s$ có thể nhận. Mỗi khoảng $[d_i, d_{i+1}), i = 0, \dots, n$, được gán một nhãn t_i, được tính như sau.

(1). Mã hóa các khoảng $[d_i, d_{i+1}]$: $\left\{S\left[d_i, d_{(i+1)}\right], i = 0,1, \dots, n\right\}$.

(2). Số hóa các khoảng mã đã mã: $\left\{N\left(S\left[d_i, d_{(i+1)}\right]\right), i = 0,1, \dots, n\right\}$.

(3). Tính tập giá trị băm của từng khoảng đã số hóa: $\left\{t_i = H_k\left(N\left(S\left[d_i, d_{(i+1)}\right]\right)\right), i = 0,1, \dots, n\right\}$.

(4). Để lưu a, giá trị thuộc tính A_s vào CSDL $R^E = (A_1, \dots, A_s^E, \dots, I^{A_s})$: trước hết, chọn khoảng $D_i = [d_i, d_{i+1})$ mà $a \in D_i$; mã hóa $c = E(a)$, và gửi cho (máy chủ) hệ thống bản ghi $r^E = (a_1, \dots, c, \dots, t_i)$, với $t_i = H\big(N(S[D_i])\big)$ là nhãn của khoảng $\big[d_i, d_{(i+1)}\big]$ này.

Các bước từ (1) đến (3) có thể được tính sẵn.

Hàm biên đổi cho dữ liệu số

Với cách mã hóa và lưu trữ này, các truy vấn rõ – plain querry, Q trên CSDL rõ – plain database R, trước hết được biến đổi thành các truy vấn mã – cipher querry, Q^E dưới dạng các truy vấn khoảng và thực hiện trên CSDL mã – cipher database, R^E. Kết quả của pha 1, $R'^E \subset R^E$, được chuyển cho pha 2, ở đó, nó sẽ được giải mã thành R' và truy vấn rõ Q được thực hiện trên R' này để được kết quả cuối cùng.

Thủ tục biến đổi truy vấn khoảng Q trên CSDL rõ thành Q^E trên CSDL mã gồm các bước sau.

Trans($Q \equiv x \in [\mathrm{a}, \mathrm{b}]$)

(1). Xác định 2 họ *prefix*: $F(a), F(b)$.

(2). Số hóa 2 họ này: $N\big(F(a)\big), N\big(F(b)\big)$.

(3). Tính nhãn $t = H_k\big(\{N\big(F(a)\big), N\big(F(b)\big)\}\big)$.

(4). Trả về t.

Và ta áp dụng minh họa một số dạng truy vấn sau

(I). $Q \equiv R.A_s = a$ (Truy vấn =)

. $t = Trans(R.A_s \in [a, a+1])$.

. Trả về các bản ghi có $R^E.I^{A_s} \cap t \neq \emptyset$.

(II). $Q \equiv a \leq R.A_s \leq b$ (Truy vấn khoảng)

. $t = Trans(R.A_s \in [a, b])$.

. Trả về các bản ghi có $R^E.I^{A_s} \cap t \neq \emptyset$.

(III). **$Q \equiv R.A_s \neq a$ (Truy vấn khác)**

. $t = Trans(R.A_s \in [a, a+1])$.

. Trả về các bản ghi có $R^E.I^{A_s} \cap t = \emptyset$.

Bài tập: cài đặt các truy vấn với biểu thức luận lý And, Or, XoR.

3 THIẾT KẾ CƠ SỞ DỮ LIỆU CHO BẢO MẬT

THIẾT KẾ CƠ SỞ DỮ LIỆU

Như ta đã biết, CSDL là "trái tim" của một hệ thông thống tin (HTTT). Một cách không hình thức, đó là tập tất cả đối tượng dữ liệu được tập hợp và tổ chức cho mục đích tìm kiếm phục vụ các nghiệp vụ mà HTTT phải đáp ứng. Chẳng hạn, HTTT về quản lý học sinh, tập hợp các thông tin về đối tượng học sinh mà nhà trường cần quản lý kiểu như "sổ học bạ" mà ta đã quen; hay HTTT bán hàng mô phỏng cách cách chủ cửa hàng làm và tra cứu "sổ cái" của cửa hàng để biết doanh số, lợi nhuận, hay lên kế hoạch nhập hang …; hay HTTT nhân sự giúp hoạch định kế hoạt phát triển và sử dụng nguồn nhân lực của công ty; .v.v.

Qua các ví dụ trên, ta thấy, để xây dựng thành công một HTTT, 2 câu hỏi quan trong phải được trả lời là:

. HTTT sẽ làm hay hỗ trợ làm những gì? Trả lời câu hỏi này được gọi là phân tích nghiệp vụ – BA (Business Analysis). Mục tiêu là xác định các nhiệm vụ mà hệ thống phải hoàn thành để hỗ trợ quản lý.

. Cần những thông tin hay dữ liệu gì? Trả lời câu hỏi này được gọi là thiết kế mô hình dữ liệu – Data Model. Mục tiêu là xác định những dữ liệu gì phải cung cấp để hệ thống có thể thực hiện những nhiệm vụ đó.

Sơ đồ luồng dữ liệu

Nghiệp vụ thường đi liền với tác nhân – agent, bao gồm người hay máy móc, thực hiện và kết quả thực hiện, trong nhiều trường hợp, nó còn gắn liền với nơi thực hiện nữa. Chẳng hạn, lương thưởng do nhân viên phòng kế toán-tài chánh thực hiện trên cơ sở hiệu suất làm việc, kết quả là bảng chi tiết về lương hay thưởng; hay kế hoạch tuyển dụng do phòng nhân sự hoạch định dựa trên nhu cầu nhân lực của các dự án đang hay sẽ thực hiện, cùng với quỹ lương có thể đáp ứng…

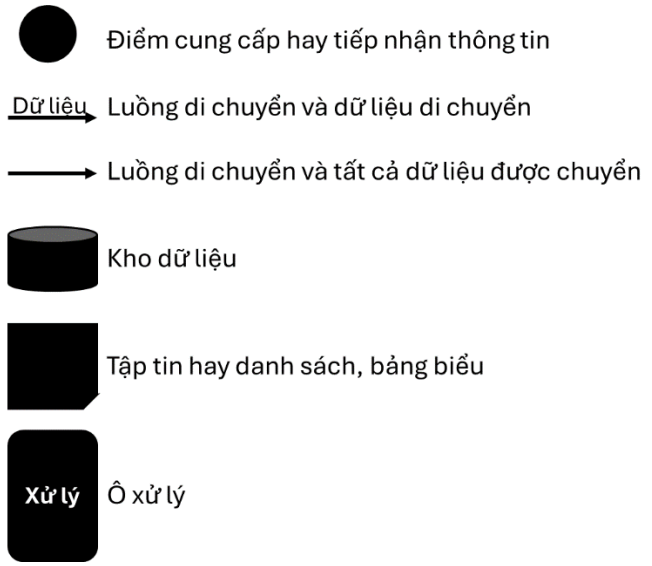
Thông tin hay dữ liệu sẽ di chuyển theo một luồng, gọi là luồng dữ liệu – Data Flow. Dữ liệu được biến đổi bởi những tác nhân tại những vị trí khác nhau để được kết quả mong muốn cuối cùng. Quá trình đó biểu diễn hình ảnh bằng sơ đồ luồng dữ liệu – DFD (Data Flow Diagrams). Một cách tổng quát, luồng dữ liệu có thể được phác thảo bằng cách, trước hết, xác định đối tượng sử dụng và các thông tin hay dữ liệu cần cung cấp và muốn nhận. Sau đó, hình thành luồng dữ liệu qua các nghiệp vụ biến đổi dữ liệu tương ứng tại các vị trí – locations, sử dụng vị trí thay vì tác nhân ở mức phác thảo để không sa vào chi tiết.

Ví dụ, ta thử phác họa luồng dữ liệu cho bán hàng của một đại lý.

Từ phác thảo để có các nhìn tổng thể, người thiết kế có thể phân rã luồng dữ liệu theo sơ đồ luồng dữ liệu.

Các ký hiệu quy ước

Sơ đồ luồng dữ liệu (DFD) là các biểu diễn đồ họa của một hệ thống minh họa luồng dữ liệu trong hệ thống. Chúng tôi sử dụng các ký hiệu cho DFD như trong hình.



Các mức phân rã

DFD có thể được chia thành các cấp độ khác nhau, cung cấp các mức độ chi tiết khác nhau về hệ thống. Sau đây là bốn cấp độ của DFD.

(.) **DFD cấp 0**. DFD cấp độ 0, kí hiệu DFD-0, cấp tổng quát nhất. Nó cung cấp cái nhìn tổng quan về toàn bộ hệ thống. Nó đơn giản chỉ có một xử lý với các luồng dữ liệu và kho lưu trữ dữ liệu trong hệ thống mà không cung cấp bất kỳ chi tiết nào về hoạt động nội bộ của các quy trình này.

DFD-0 có thể được thiết lập bằng cách đặt tên hệ thống và xác định các dữ liệu đầu vào (input) và đầu ra (output) của hệ thống.

(.) **DFD cấp 1**. DFD cấp độ 1, ký hiệu DFD-1, cung cấp cái nhìn chi tiết hơn về hệ thống bằng cách chia nhỏ các quy trình chính được xác định trong DFD-0 thành các quy trình con. Mỗi quy trình con được mô tả như một quy trình riêng biệt trên DFD-1. Các luồng dữ liệu và kho lưu trữ dữ liệu liên quan đến từng quy trình phụ cũng được hiển thị.

DFD-1 có thể được thiết lập bằng cách xác định các đối tượng sử dụng hệ thống, và các dữ liệu nhập/xuất cho tác nhân đó.

(.) **DFD cấp i**. DFD cấp độ i, i =2,3,…, ký hiệu DFD-i, cung cấp cái nhìn chi tiết hơn về hệ thống bằng cách chia nhỏ các quy trình con được xác định trong DFD-(i – 1) thành các quy trình con tiếp theo. Mỗi quy trình con được mô tả như một quy trình riêng biệt trên DFD-i. Các luồng dữ liệu và kho lưu trữ dữ liệu liên quan đến từng quy trình phụ cũng được hiển thị.

DFD-i có thể được thiết lập bằng cách phân rã các xử lý phức tạp thành các xử lý đơn giản hơn. Một xử lý là "phức tạp" nếu hoặc các nguồn dữ liệu chồng chéo, khi ấy có thể tách theo nguồn dữ liệu; hoặc xử lý tuần tự bởi nhiều đối tượng, khi ấy có thể tách theo đối tượng.

DFD-i có thể được tiếp tục phân rã cho đến khi đọc tên ô xử lý và các dữ liệu vào ra của ô xử lý có thể hiểu và có thể thực hiện.

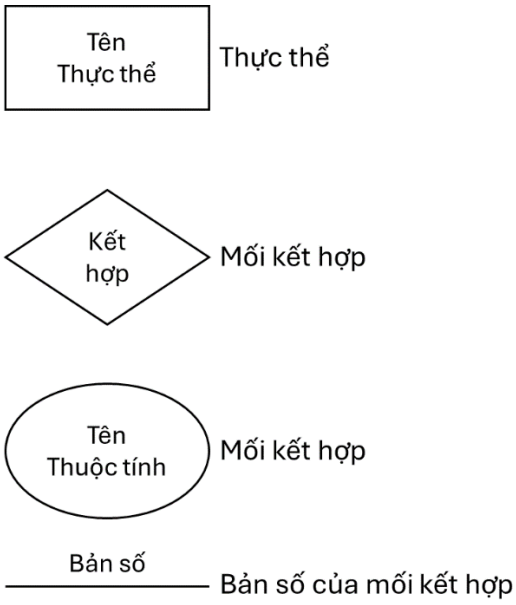
Lược đồ thực thể kế hợp

Trong quá trình phân tích nghiệp vụ, ta luôn phải làm rõ: dữ liệu được dùng để làm gì? Và phải lưu trữ dữ liệu như thế nào? Nhiệm vụ này được gọi là thiết kế CSDL. Ta sẽ tiếp cận bằng lược đồ thực thể kết hợp cho nhiệm vụ thiết kế này

Khái niệm và ký hiệu

Lược đồ thực thể kết hợp – ER (Entity Relationship) là mô hình thường được dùng để thiết kế CSDL ở mức quan niệm giúp biểu diễn một cách trừu tượng cấu trúc của CSDL. Lược đồ ER giúp người thiết kế mường tượng rõ hơn về CSDL họ phải xây dựng để đáp ứng các yêu cầu nghiệp vụ mà hệ thống phải thực hiện. ER được trình bày dạng đồ họa gồm các thành phần chính như trong hình vẽ.

<u>Ký hiệu</u>



<u>Khái niệm</u>

(.) **Thực thể**. Thực thể – Entity, là 1 đối tượng cần quản lý. Đó có thể là đối tượng cụ thể hoặc trừu tượng, được quản lý bởi các thuộc tính xác định. Ví dụ, Sinh_viên hay Mặt_hàng là các thực thể trong thế giới thực cần được mô hình hóa để quản lý bằng máy tính. Chẳng hạn, Mặt_hàng được quản lý qua Mã_MH, Tên_MH, Số_tồn, Giá_mua, Giá_bán, thường được viết theo kiểu quan hệ - relation, kiểu Mặt_hàng(<u>Mã_MH</u>, Tên_MH, ĐV_tính Số_tồn, Giá_mua, Giá_bán).

(.) **Thuộc tính**. Thuộc tính – Attributes, là các đặc trưng mô tả thực thể. Mỗi thuộc tính sẽ nhận giá trị trong miền thuộc tính của nó do kiểu dữ liệu quy định. Đó có thể là dữ kiểu số (Numeric), kiểu chuỗi ký tự (Char), kiểu luận lý (Boolean), hay các kiểu đặc biệt khác. Ví dụ, Họ_tên hay Tên_mặt_hàng có kiểu dữ liệu là Char(30) với 30 ký tự tối đa; Năm_sinh hay Số_lượng kiểu Numeric; Phái có thể dùng kiểu Boolean.

(.) **Thể hiện**. Thể hiện – instance, là giá trị cụ thể của một thực thể. Chẳng hạn mặt hàng xà-bông có thể là Mặt_hàng(0015, "xà-bông", "hộp", 700, 4000, 5000).

(.) **Thuộc tính khóa**. Thuộc tính khóa – key, là (tập) thuộc tính để phân biệt các thể hiện của cùng một thực thể. Các giá trị của thuộc tính khóa các thể hiện khác nhau trên 2 thực thể phải khác nhau. Ví dụ mỗi sinh viên có 1 mã sinh viên riêng biệt; hay mỗi mặt hang phải có 1 mã mặt hàng.

(.) **Mối kết hợp**. Mối kết hợp hay quan hệ – Relationship, là một "thực thể" biểu diễn quan hệ giữa 2 hoặc nhiều thực thể. Ví dụ Hoc_sinh có mối kết hợp HS_MH thể hiện mối quan hệ giữa thực thể Hoc_sinh và thực thể Môn_học; hay Mặt_hàng có quan hệ với Nhà_cung_cấp cũng sẽ phát sinh một mối kế hợp. Mối kết hợp cũng có thuộc tính, ngoài thuộc tính khóa của nó (thường là sự kết hợp các thuộc tính khóa của các thực thể trong quan hệ). Chẳng hạn, mối kết hợp HS_MH viết là HS_MH(MHS, MMH, Điểm, HK, Niên_khóa), trong đó khóa là cặp 2 thuộc tính MHS và MMH.

(.) **Bản số kết hợp**. Bản số - quantity, là giá trị số cho biết 1 thể hiện của thực thể này có thể kết hợp với tối đa bao nhiêu thể hiện của thực thể kia trong cùng một mối kết hợp. Các kiểu liên kết phổ biến gồm 1:1 (liên kết 1-1), 1:n (liên kết 1-nhiều), n:1 (liên kết nhiều-1), và m:n (liên kết nhiều-nhiều).

Ràng buộc trên bản số

Ràng buộc trên bản số là những quy định để giới hạn số các tổ hợp có thể của các thực thể tham gia trong một mối kết hợp nhằm phản ánh đúng ràng buộc của các thực thể trong thế giới thực. Có 2 loại ràng buộc là ràng buộc tỉ số và ràng buộc tham gia.

(.) **Ràng buộc min-max**. Là cặp (min, max) trên thực thể, ký hiệu (min, max)-Thực_thể để xác con số định tối thiểu và tối đa số thể hiện một thực thể có thể có. Số max thường được ghi là m hay n. Ví dụ (1, n)-Phòng_ban diễn tả ràng buộc thực tế là "một phòng ban phải có ít nhất 1 nhân viên, và có thể có nhiều nhân viên"; hay (10,40)-Lớp_học – "chỉ mở lớp nếu có đủ 10 học sinh và sĩ số mỗi lớp không quá 40 học sinh".

(.) **Ràng buộc tham gia**. Là ràng buộc khi thực thể không có thuộc tính khóa, chỉ tham gia khi có 1 thực thể chủ.

Các bước thiết kế

Để thiết kế CSDL cho một hệ thống thông tin, người thiết kế có thể thực hiện các bước sau.

(.) **Bước 1**. Liệt kê tất cả các thực thể có thể phải quản lý.

(.) **Bước 2**: Với mỗi cặp thực thể, xác định xem có thể có quan hệ với nhau không, nếu có thì là gì. Đồng thời xác định bản số cho các liên kết.

(.) **Bước 3**. Với mỗi thực thể và mối kết hợp, xác định và gắn các thuộc tính cho tập thực thể và mối quan hệ.

(.) **Bước 4**. Quyết định miền giá trị cho từng thuộc tính.

(.) **Bước 5**: Đồng thời cũng xác định (các) thuộc tính khóa cho mỗi thực thể và mối kết hợp.

(.) **Bước 6**: Xác định ràng buộc (min-max, tham gia) cho mối quan hệ và thể hiện chúng.

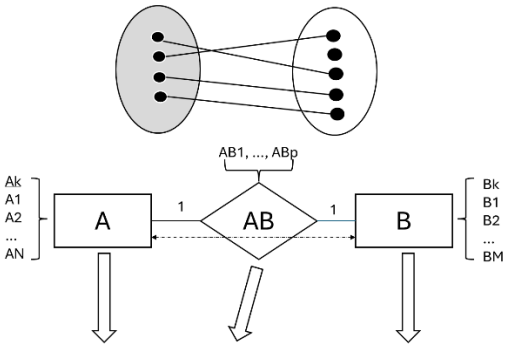
Mô hình vật lý

Mô hình vật lý – Physical Model, của 1 hệ CSDL, là tập hợp cả (bảng) quan hệ được lưu trên thiết bị vật lý, nghĩa là hệ thống các tập tin lưu trữ các thể hiện của các thực thể, bao gồm cả thực thể biểu diễn một đối tượng cụ thể lẫn thực thể biểu diễn mối kết hợp. Mô hình CSDL vật lý có thể dễ dàng được thiết lập từ mô hình CSDL ý niệm, mô hình ER.

Nhu ta đã thấy trong mô hình ER có 2 "đối tượng" là thực thể và mối kết hợp. Sự liên kết giữa 2 thực thể lại có 3 dạng chính: liên kết 1-1, liên kết 1-n hay liên kết n-1, và liên kết m-n.

Liên kết 1-1

Hai thực thể A và B có liên kết 1-1 như hình vẽ dưới, nếu 1 thể hiện $a \in A$ có liên kết với tối đa 1 thể hiện $b \in B$, và ngược lại. Khi ấy, thiết bị vật lý sẽ lưu giữ 2 quan hệ A và B, là các tập tin gồm các bản ghi (record) với các trường (field) gồm các thuộc tính của nó kết hợp với thuộc tính khóa của thực thể đối ngẫu. Tập thuộc tính của mối kết hợp AB sẽ được lưu giữ trong các trường mới của 1 trong 2 thực thể A hay B.



A(Bk, Ak, A1, A2, ..., AN, AB1, ..., ABp) B(Ak, Bk, A1, A2, ..., AM)

Ví dụ. Giả sử thực thể A có các thuộc tính $Ak, A1,\ A2, \dots, AN$, với Ak là thuộc tính khóa, và thục thể B có các thuộc tính $Bk, B1, B2, \dots, BM$, với Bk là thuộc tính khóa. A và B có liên kết 1-1 qua mối kết hợp AB có các thuộc tính $AB1, \dots, ABp$. Mô hình ER này được chuyển thành 2 quan hệ, và sẽ lưu trên thiết bị phần cứng thành tập tin với các cấu trúc tương ứng sau:

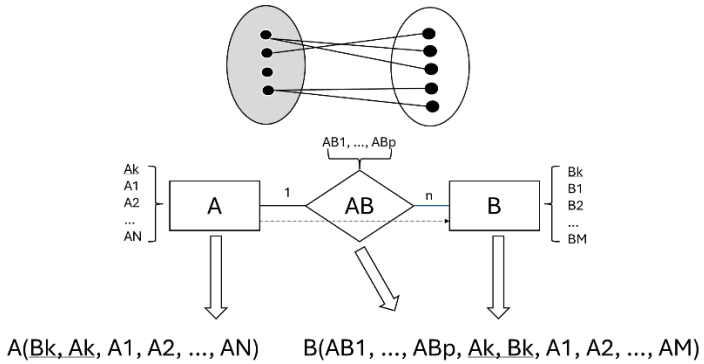
(.) Mối kết hợp AB không phải lưu trữ vật lý. Các thuộc tính của nó $AB1, \dots, ABp$ có thể chuyển cho 1 trong 2 quan hệ sau (chỉ cần 1, không phải cả 2).

(.) Quan hệ A sẽ nhận thuộc khóa BK của thực thể B làm khóa ngoại (foreign) của nó, và nhận các thuộc tính của mối kết hợp và thành các trường (field) dữ liệu của cấu trúc bản ghi (record): $A(Bk, Ak, A1, A2, \dots, AN, AB1, \dots, ABp)$.

(.) Quan hệ B sẽ nhận thuộc tính khóa ngoại Ak, và bản ghi của nó có cấu trúc $B(Ak, Bk, B1, B2, \dots, BM)$.

Liên kết 1-n hay liên kết n-1.

Hai thực thể A và B có liên kết 1-n như hình vẽ dưới, nếu như 1 thể hiện $a \in A$ có thể liên kết với $n \geq 0$ thể hiện $b \in B$; trong khi đó, 1 thể hiện $b \in B$ chỉ có thể liên kết tối đa 1 thể hiện $a \in A$ (hoặc 0 liên kết với thể hiện nào của A). Khi ấy, thiết bị vật lý cũng sẽ chỉ lưu giữ 2 quan hệ A và B. Trong đó, quan hệ A là các tập tin gồm các bản ghi (record) chỉ với các trường (field) gồm các thuộc tính của chính A. Và quan hệ B gồm các trường lưu giữ các thuộc tính của B và thuộc tính khóa của A, cùng với tất cả các thuộc tính của mối kết hợp AB.



Ví dụ. Giả sử thực thể A có các thuộc tính $Ak, A1, \ A2, \dots, AN$, với Ak là thuộc tính khóa, và thục thể B có các thuộc tính $Bk, B1, B2, \dots, BM$, với Bk là thuộc tính khóa. A và B có liên kết 1-n qua mối kết hợp AB có các thuộc tính $AB1, \dots, ABp$. Mô hình ER này được chuyển thành 2 quan hệ, và sẽ lưu trên thiết bị phần cứng thành tập tin với các cấu trúc tương ứng sau:

(.) Mối kết hợp AB không phải lưu trữ vật lý. Các thuộc tính của nó $AB1, \dots, ABp$ chuyển cho quan hệ B.

(.) Quan hệ A sẽ có các trường (field) dữ liệu của cấu trúc bản ghi (record) $A(Ak, A1, A2, \dots, AN)$.

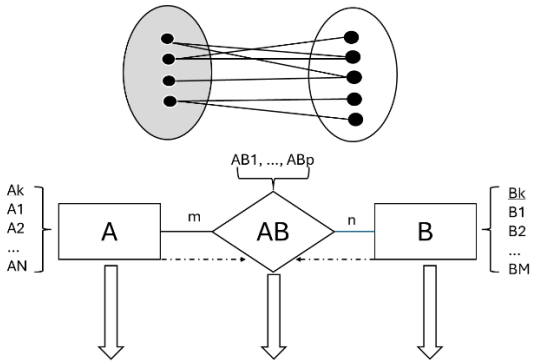
(.) Quan hệ B sẽ nhận thuộc tính khóa ngoại Ak cùng tất cả thuộc tính của mối kết hợp AB, và bản ghi của nó có cấu trúc $B(Ak, Bk, B1, B2, \dots, BM, AB1, \dots, ABp)$.

Mối liên kết n-1 thực hiện tương tụ theo chiều ngược lại của vai trò A và B.

Mối liên kết m-n

Hai thực thể A và B có liên kết m-n như hình vẽ dưới, nếu như 1 thể hiện $a \in A$ có thể liên kết với $n \geq 0$ thể hiện $b \in B$; và 1 thể hiện $b \in B$ có thể liên kết với $m \geq 0$ thể hiện $b \in B$. Khi ấy, thiết bị vật lý cũng sẽ 3 quan hệ A, B và AB. Trong đó, quan hệ A và B là các tập tin gồm các bản

ghi (record) chỉ với các trường (field) gồm các thuộc tính của chính A hay B. Và quan hệ AB sẽ giữ khóa của 2 $Ak \in A$ và $Bk \in B$ và các trường lưu giữ các thuộc tính của mối kết hợp AB.



A(Ak, A1, A2, ..., AN) AB(Ak, Bk, AB1, ..., ABp)B(Bk, A1, A2, ..., AM)

Ví dụ. Giả sử thực thể A có các thuộc tính $Ak, A1,\ A2, \dots, AN$, với Ak là thuộc tính khóa, và thục thể B có các thuộc tính $Bk, B1, B2, \dots, BM$, với Bk là thuộc tính khóa. A và B có liên kết m-n qua mối kết hợp AB có các thuộc tính $AB1, \dots, ABp$. Mô hình ER này được chuyển thành 2 quan hệ, và sẽ lưu trên thiết bị phần cứng thành tập tin với các cấu trúc tương ứng sau:

(.) Quan hệ AB sẽ có các trường lưu giữ các thuộc tính $AB1, \dots, ABp$ đồng thời sẽ lưu giữ them 2 thuộc tính khóa $Ak \in A$ và $Bk \in B$.

(.) Quan hệ A sẽ có các trường (field) dữ liệu của cấu trúc bản ghi (record) $A(Ak, A1, A2, \dots, AN)$.

(.) Quan hệ B sẽ có các trường (field) dữ liệu của cấu trúc bản ghi (record) $B(Bk, B1, B2, \dots, BM)$.

Qui trình thiết kế mô hình vật lý

Ta tóm tắt quy trình thiết kế mô hình vật lý của CSDL từ mô hình ý niệm dạng thực thể - kết hợp gồm các quy tắc sau:

(1). Thực thể sẽ được chuyển sang quan hệ với các trường dữ liệu mở rộng tùy thuộc các bản số quan hệ là 1:1, 1:n, hay m:n.

(2). Chuyển các mối kết hợp 3-ngôi thành quan hệ có khóa chung là tổ hợp của 3 khóa thành phần.

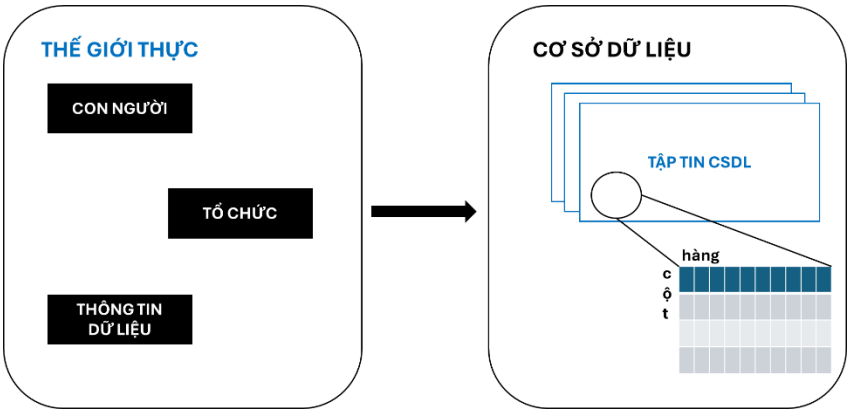
(3). Mối kết hợp có liên kết 1-1 thì chuyển thành 2 quan hệ và quan hệ này sẽ giữ khóa của quan hệ kia, và ngược lại. Các thuộc tính của mối kết hợp, nếu có, sẽ chuyển thành trường dữ liệu của chỉ 1 trong 2 quan hệ thành phần.

(4). Mối kết hợp có liên kết 1-n, sẽ phát sinh 2 quan hệ. Quan hệ phía 1 có cùng cấu trúc với thực thể. Quan hệ phía n sẽ sẽ chép khóa phía 1 và mọi thuộc tính của mối kết hợp, nếu có, cho quan hệ phía n.

(5). Mối kết hợp có liên kết m-n sẽ phát sinh 3 quan hệ. Hai quan hệ tương ứng với 2 thực thể có cùng cấu trúc với thực thể. Quan hệ thứ 3 lưu giữ thông tin mối kết hợp cùng với 2 khóa của các quan hệ trong mối kết hợp.

THIẾT KẾ CƠ SỞ DỮ LIỆU CHO BẢO MẬT

Như trên ta đã thấy, thiết kế hệ thống thông tin là quá trình mô hình hóa thế giới thực như hình vẽ minh họa.



Quá trình thiết kế gồm 2 bước chính:

(.) **Phân tích nghiệp vụ**. Xác định quy trình luồng thông tin/dữ liệu di chuyển và được biến đổi như thế nào qua việc phân tích các nghiệp vụ phải thực hiện trong thế giới thực.

(.) **Xây dựng mô hình dữ liệu**. Tập hợp và tổ chức các thông tin/dữ liệu sao cho có thể tự động hóa các hoạt động nghiệp vụ một cách hợp lý và hiệu quả. Kết quả của giai đoạn này là hệ thống các tập tin lưu trữ các quan hệ.

Trong quá trình phân tích nghiệp vụ, việc xác định được các thao tác nào được thực hiện và cường độ thực hiện các thao tác đó.

Phân rã các quan hệ

Cơ sở dữ liệu, như ta đã biết, được thiết lập cho mục đích tìm kiếm. Như ta cũng biết, mã hóa là biến đổi dữ liệu dạng rõ – plain data, thành dữ liệu dạng mã – cipher data, sao cho chỉ người có khóa mã hóa mới có thể giải mã để phục hồi lại được bản rõ. Các thuật toán mật mã hiện đại đều đảm bảo tính ngẫu nhiên theo nghĩa chỉ thay đổi một phần nhỏ thông tin của dữ liệu rõ thì dữ liệu mã cũng thay đổi gần như khắc hẳn. Như vậy, hầu như không thể đinh nghĩa được một độ đo để có thể sắp xếp hay so sánh các bản mã với nhau.

Mã hóa và tìm kiếm

Mục tiêu tìm kiếm và mục tiêu bảo mật CSDL bằng mã hóa là 2 mục tiêu hầu như không thể cùng tồn tại trong một hệ thống thông tin. Điều này dẫn đến nhu cầu phân ra một quan hệ sao cho có thể bảo mật dữ liệu nhưng vẫn hiệu quả cho mục tiêu tìm kiếm. Để bảo mật CSDL, ta sẽ sử dụng các nguyên tắc phân rã sau.

1*: không mã hóa thuộc tính có cường độ tìm kiếm cao*

Giả sử có quan hệ $R(A_k, A_1, \dots, A_s, \dots, A_n)$ với A_k là thuộc tính khóa và A_s là thuộc tính chứa dữ liệu nhạy cảm cần bảo mật nhưng tần suất truy cập trên A_s cao. Việc tách thuộc tính nhạy cảm A_s sang một quan hệ R_s phái sinh từ R là cần thiết. Quan hệ R được gọi là quan hệ cha và quan hệ R_s được gọi là quan hệ con.

2: thiết lập mối kết

Mối kết hợp RR giữa quan hệ R và quan hệ R_s được thiết lập là mối kết hợp m-n, với R_{s_k} là thuộc tính khóa của quan hệ phái sinh R_s.

Theo cách này, giả sử không có bảng dữ liệu RR, nếu việc phục hồi lại quan hệ R từ R_s, dữ liệu trên R_s không phải mã hóa. Ngược lại, cần xem xét them về thiết kế.

3: xây dựng các chỉ mục an toàn

Sử dụng các phương pháp chỉ mục an toàn để lưu giữ quan hệ mối kết hợp RR.

4 ỨNG DỤNG

CÔNG NGHỆ SỔ CÁI PHI TẬP TRUNG

Sổ cái.

Thu nhỏ lại hoạt động của xã hội như những giao dịch (*transaction*) như trao đổi, vay mượn, buôn bán,… trực tiếp giữa các cá nhân trong công đồng với nhau. Để đảm bảo có vay thì phải trả, có bán thì mới mua,… các hoạt động cần có người làm chứng, nói cách khác, mọi giao dịch cần minh bạch, không có gì khuất tất. Mô hình giao dịch đơn giản này có thể được triển khai với khái niệm sổ-cái (*ledger*). Theo đó, ta hình dung, giữa làng đặt một sổ cái mà mọi người đều có thể đọc và gi lên đó công khai. Trong làng, ai vay của ai hay ai trả ai cái gì đều tự nguyện ghi lên sổ cái đó. Mọi tẩy xóa là không chấp nhận. Hình ảnh sổ cái công cộng (*public ledger*) với các giao dịch công khai như hình dưới, cho ta ý niệm ban đầu về việc vận hành ổn định xã hội một cách minh bạch và công khai. Chẳng hạn, sổ cái ghi

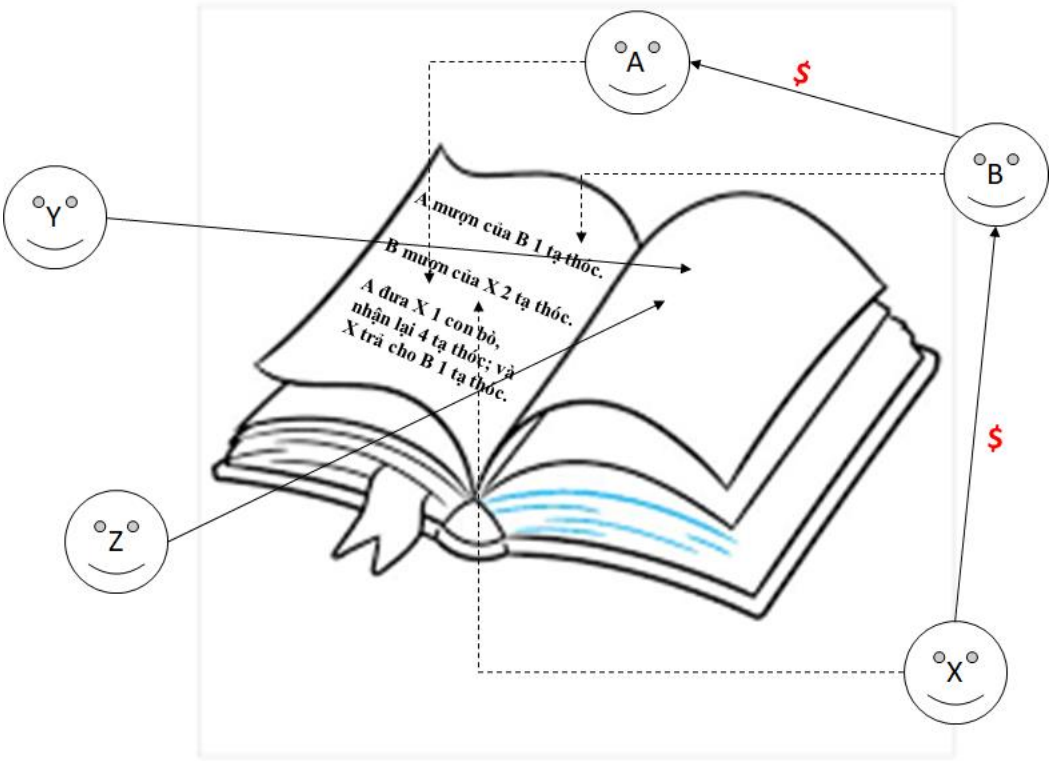
- “A mượn của B 1 tạ thóc.”

- “B mượn của X 2 tạ thóc.”

- “A đưa X 1 con bò, nhận lại 4 tạ thóc; và X trả cho B 1 tạ thóc.”

- …

Bằng cách quy đổi mọi thứ về một thứ ta gọi là “tiền tệ”, ký hiệu $, ta có mô hình kinh tế tiền tệ đơn giản như hình vẽ.



Sổ cái phi tập trung

Để đảm bảo sổ cái không bị phá hủy, vì một lý do nào đó như "chúa chổm" có thể hủy sổ cái để từ đó không còn bằng chứng nào chứng minh hắn nợ-chúa-chổm. Vấn đề này có thể giải được bằng cách "phi tập trung sổ cái – **Decentralized Ledger**", theo đó, sổ cái sẽ được giao cho n người giữ, mỗi người tự sao chép và cập nhật sổ cái mỗi khi có một giao dịch công khai nào xảy ra. Bằng cách đó, các cá nhân trao đổi tài sản với nhau chỉ phải công bố giao dịch của mình cho mọi người thấy. Những người đang giữ bản sao luận án sẽ xác minh (validate) các giao dịch đó, và nếu hợp lệ, sẽ ghi giao dịch hợp lệ (validated transcaction) vào sổ cái mình đang giữ. Một sổ cái phi tập trung như thế đảm bảo 3 tính chất sau cho mọi giao dịch công khai:

(•) Minh bạch: mọi giao dịch được thực hiện công khai với các nhân chứng là những cá nhân đạng giữ sổ cái.

(•) Toàn vẹn: không ai có thể sửa đổi nội dung của giao dịch đã được ghi xuống vì không thể thông đồng được với mọi người giữ sổ cái.

(•) An toàn: sổ cái không bị phá hủy (nếu còn lại một vài người đang giữ sổ cái).

Công nghệ hiện thực hóa khái niệm sổ cái thỏa 3 tính chất trên gọi là **công-nghệ-sổ-cái-phi-tập-trung**.

Blockchain và sổ cái phi tập trung

Blockchain viết liền gồm hai thành phần chính là:

(**1**) **block**: tựa như một trang (*page*) của sổ cái (*ledger*) với nội dung là tất cả các giao dịch (*transactions*) được ghi lên trang đó.

(**2**) **chain**: tựa như một mắt xích (*chain*) kết chặt một trang mới với trang trước đó. Các trang chỉ được thêm vào cuối chuỗi và chuỗi dài dần ra.

Blockchain và công nghệ thông tin

Trong khoa học máy tính, khái niệm blockchain hoàn toàn có thể được cài đặt bằng một cấu trúc dữ liệu đặc biệt kiểu danh sách liên kết (*list*) trong đó mỗi trang gồm các bản ghi (*record*) lưu trữ thông tin của một giao dịch (*transaction*), và một số thông tin quan lý khác.

Blockchain là **list** (*danh sách*) đặc biệt ở chỗ với **list**, để liên kết thì phân tử sau giữ địa chỉ (*address*) của phần tử trước (trong nhiều ngôn ngữ liên kết này được thể hiện qua kiểu con trỏ (pointer)); nhưng trong **blockchain**, trang sau giữ mã-chứng-thực (**MAC** – *Message Authentication Code*) của trang trước (chính là trị băm mật mã (*cryptographic-hash value*) của thông điệp (message) là toàn bộ nội dung của trang trước.).

Như vậy, blockchain gần với khái niệm tập hợp (*set*) hơn là khái niệm mảng (*array*) hay danh sách liên kết (*list*) theo nghĩa phải duyệt tuần tự (sequence search) để tìm trang hay Tx cần tìm. Điều này phù hợp với lưu trữ dạng tập tin (*file*) trên thiết bị phần cứng.

Blockchain và khoa học máy tính

Bên cạnh việc tổ chức sổ cái như xâu các trang (*blockchain*), mỗi trang là bản ghi (*record*) chứa tập hợp các giao dịch (*Tx*), và được lưu trên thiết bị phần cứng như một CSDL phi tập trung (*decentralized database*), theo nghĩa CSDL được lưu trữ trên nhiều máy một cách độc lập với nhau nhưng vẫn đảm bảo tính nhất quán của CSDL chung. Như vậy, để duy trì sổ cái cần (**i**) một hạ tầng kiến trúc mạng và các thuật toán truyền thông mạng hiệu quả và (**ii**) các thuật toán đồng bộ dữ liệu giữa các CSDL được lưu trữ trên các điểm khác nhau.

Hạ tầng mạng cho blockchain

Vấn đề truyền thông mạng được giải quyết và hiện thực trên kiến trúc mạng ngang hàng (**P2P network** – *peer-to-peer network*), ở đó các nút mạng liên kết vào trao đổi thông tin trực tiếp, không qua trung gian. Các nút mạng được trang bị các thuật toán để quét mạng liên tục và sẵn sàng cập nhật CSDL khi có một trang mới xuất hiện.

Đồng bộ dữ liệu

Tính nhất quán của CSDL chung được đảm bảo qua các thuật toán được biết với tên thuật-giải-đồng-thuận (*consensus algorithm*) mà được xây dựng dựa trên nguyên tắc đồng-thuận-số-đông, theo đó, block mới sẽ được thêm vào chain dài nhất hiện tại.

Một cách tổng quát, blockchain là một CSDL phi-tập trung (decentralized database).

Công nghệ sổ cái blockchain

Công nghệ lockchain (*blockchain technology*) là một nền tảng (*platform*) cung cấp các cơ chế cũng như công cụ cho phép hình thành và duy trì một mạng blockchain (*blockchain network*).

Mạng blockchain

Mạng blockchain hình thành từ các nút (*node*) mạng hoạt động độc lập, giao tiếp với các nút khác theo giao thức mạng ngang hành. Mỗi nút tham gia mạng với một định danh (*identification*) duy nhất và cũng chính là tài khoản của người sở hữu nút đó, vì thế, nhìn dưới góc nhìn tài chánh-ngân hàng, định danh này chính là số tài khoản (*account number*) của ngươi đó.

Có 2 loại nút trong mạng blockchain là nút sử dụng (*user node*) và nút duy trì (*miner node*) dữ liệu của CSDL blockchain.

Nút sử dụng dữ liệu – user node

Là bất kỳ thiết bị nào có thể kết nối mạng để thực hiện các giao dịch liên quan đến thông tin được lưu giữ trong CSDL blockchain. Để giao dịch các dữ liệu của riêng mình, user node phải có khóa sử dụng tài khoản, như mã nhận dạng cá nhân (*Pin Code – Personal Identification Number*) được dùng để sử dụng thẻ ATM của ngân hàng. Khóa sử dụng tài khoản và số tài khoản có một quan hệ toán học chặt chẽ. Thực chất chúng là cặp khóa gồm khóa cá nhân (*private key*) và khóa công khai (*public key*) của một hệ mã công khai hiện đại (advanced public key cryptosystem), và địa chỉ (*address*) được phái sinh từ khóa công khai (nhiều mạng chuỗi khối, còn cho phép sử dụng trực tiếp khóa công khai làm địa chỉ tài khoản).

Nút duy trì mạng – miner nodes

Là những máy chủ (*server*) đủ mạnh để có thể lưu giữ toàn bộ CSDL blockchain, từ block đầu tiên đến block mới nhất và còn tiếp tục cập nhật thêm các block tiếp nữa. Do tính chất bất biến của dữ liệu lưu trữ trong mạng blockchain, các khối chỉ được thêm vào chuỗi chứ không thể xóa hay sủa đổi. Hơn nữa, chỉ các khối hợp lệ (*validated block*) mới được gắn thêm (*chain*) vào chuỗi. Một Các miner nodes có nhiệm vụ kiểm tra (*verify*) các giao dịch hợp lệ tập hợp trong khối hợp lệ và tranh đua cùng các miner nodes khác trong một trò chơi (*game*) được thiết kế riêng cho từng chuỗi

khối để trở thành nút tạo được khối mới nhanh nhất. Khối mới này sẽ được gắn vào chuỗi khối và cuộc đua tạo khối mới lại tiếp tục.

Miner node cũng có thể có thông tin riêng của nó. Nghĩa là miner node hàm ý cũng là user node khi tham gia các giao dịch trao đổi thông tin của nó với các nút mạng khác trong cùng một mạng chuỗi khối.

Các công nghệ cốt lõi

Như ta đã thấy, blockchain thực chất là một CSDL phi-tập trung được lưu trữ trên nhiều nút mạng trong mạng blockchain (*blockchain network*). CSDL này chỉ được phép gắn thêm vào, mọi thao tác có thể làm thay đổi dữ liệu đã lưu như sửa đổi (*edit*) hay xóa (*delete*) đều không được phép. Nguyên do là khi một miner node thay đổi dữ liệu nó đã lưu đồng nghĩa nó tự loại mình ra khỏi mạng các nút duy trì chuỗi khối và chỉ có thể là nút sử dụng.

Dữ liệu trong CSDL blockchain phải đảm bảo tính toàn vẹn (*integrity*).

Tính toàn vẹn của dữ liệu trong CSDL chuỗi khối được đảm bảo bởi công nghệ mật mã hiện đại bao gồm mã chứng thực thông điệp - và khóa sử dụng thông tin tài khoản.

Mọi dữ liệu được băm bằng một hàm băm mật mã (*cryptographic hash function*) và được ký (*sign*) bằng khóa cá nhân của người tạo ra dữ liệu đó.

CSDL blockchain chỉ cho phép xem (*view*) và thêm các khối hợp lệ vào.

Một khối hợp lệ tạo là khối được bởi một miner node chiên thắng (*winner*) trong một trò chơi (*game*) được thiết kế riêng cho từng mạng blockchain.

Như vậy, 3 lý thuyết cũng như công nghệ nền tảng làm trụ cột cho blockchain gồm: lý thuyết trò chơi – mật mã hiện đại – mạng ngang hàng.

Lý thuyết trò chơi

Lý thuyết trò chơi là khoa học nghiên cứu các mô hình toán học về tương tác chiến lược giữa các tác nhân. Lý thuyết trò chơi có ứng dụng trong mọi lĩnh vực khoa học xã hội, cũng như logic, khoa học hệ thống và khoa học máy tính. Các khái niệm về lý thuyết trò chơi cũng được sử dụng rộng rãi trong kinh tế học. Lý thuyết trò chơi tiên tiến được áp dụng cho nhiều mối quan hệ hành vi và hiện tại được xem như một thuật ngữ chung cho khoa học về việc ra quyết định hợp lý ở con người, động vật cũng như máy tính.

Lý thuyết trò chơi được phát triển rộng rãi vào những năm 1950 và đã được áp dụng cho các quá trình tiến hóa vào những năm 1970, mặc dù những khái niệm tương tự đã có từ những năm 1930. Lý thuyết trò chơi đã được thừa nhận rộng rãi như một công cụ quan trọng trong nhiều lĩnh vực.

Tài liệu này không nghiên cứu nhiều về lý thuyết trò chơi cũng như công nghệ mạng ngang hàng mà tập trung chủ yếu vào công nghệ mật mã hiện đại.

Mạng ngang hàng

Mạng ngang hang (**P2P**) là mạng các máy tính trong đó hai hoặc nhiều máy tính có thể chia thông tin mà không yêu cầu máy chủ hoặc phần mềm máy chủ riêng biệt.

Ở dạng đơn giản nhất, mạng **P2P** được tạo ra khi hai hay nhiều máy tính được kết nối và chia thông tin mà không cần thông qua một máy chủ riêng. Mạng **P2P** có thể là một kết nối đặc biệt — một số máy tính được kết nối qua Universal Serial Bus để truyền tập tin (*file*). Mạng **P2P** cũng có thể là một cơ sở hạ tầng cố định liên kết nhiều máy tính trong một văn phòng nhỏ qua dây cáp. Hoặc mạng **P2P** có thể là mạng ở quy mô lớn, trong đó các giao thức và các ứng dụng đặc biệt thiết lập mối quan hệ trực tiếp giữa những người dùng qua Internet.

Việc sử dụng mạng **P2P** ban đầu trong kinh doanh diễn ra sau khi triển khai các máy tính cá nhân độc lập vào đầu những năm 1980. Các máy tính cá nhân lúc bấy giờ có ổ cứng độc lập và CPU tích hợp. Các thiết bị mạng thông minh cũng có các ứng dụng tích hợp, nghĩa là chúng có thể được triển khai trên máy tính để bàn và hữu ích mà không cần dây nối chúng với máy tính lớn.

Cũng như lý thuyết trò chơi như lưu ý trên, tài liệu này không tập trung nhiều vào công nghệ mạng **P2P** mà sử dụng các giao thức tổng quát của mạng **P2P** cho mục đích chia sẻ thống tin.

Mật mã hiện đại

Mật mã học là kỹ thuật bảo mật thông tin và liên lạc thông qua việc sử dụng mã để chỉ những người có quyền sử dụng thông tin mới có thể hiểu và xử lý nó. Do đó ngăn chặn truy cập trái phép vào thông tin. **Crypt**o**graph**y được cấu thành từ tiền tố "*crypt*" có nghĩa là "*ẩn*" và hậu tố "*graph*" có nghĩa là "*viết*" – viết ẩn (nghĩa). Các kỹ thuật được sử dụng để bảo vệ thông tin được khai thác từ các khái niệm toán học và tập các phép tính theo thuật toán (*algorithm*) để chuyển đổi thông điệp (*message*) theo những cách khiến việc giải mã nó trở nên khó khăn. Các thuật toán này được sử dụng để tạo khóa mã hóa (*encryption key*), chữ ký số (*digital signature*), xác minh (*authentication*) để bảo vệ quyền riêng tư của dữ liệu (*data privacy*), và để bảo vệ các giao dịch bí mật như giao dịch thẻ tín dụng và thẻ ghi nợ.

Quá trình chuyển đổi một văn bản rõ (*plain text*) thành văn bản mật mã (*cipher text*), là văn bản được tạo ra sao cho chỉ người nhận văn bản hợp pháp có thể giải mã nó, được gọi là mã hóa (*encryption*). Quá trình chuyển đổi văn bản mã (*cipher text*) thành văn bản rõ (*plain text*) được gọi là giải mã (*decryption*).

BẢO MẬT DỮ LIỆU NHẠY CẢM

Chúng tô minh họa bằng một ứng dụng đặc biệt liên quan đến các dịch vụ trên dữ liệu DNA, bao gồm kiểm định định danh – mà có thể sử dụng trong các dịch vụ pháp y, kiểm định huyết thống – mà có thể sử dụng trong đời song dân sự. Trong các dịch vụ này, thông tin về DNA là các thông tin liên quan đến các nhiễm sắc thể, theo đó, thông tin DNA của một cá nhân là bộ

$DNA(S) = \{(s_{i,1}, s_{i,2}), (s_i), i = 1,2, \dots, N\}$.

Kiểm định định danh

Đặc tả

Bài toán kiểm định định danh là kiểm tra xem một cá thể S với thông tin nhiễm sắc thể

$DNA(S) = \{(s_{i,1}, s_{i,2});\ i = 1,2, \dots, N\}$,

có phải là cá nhân T có thông tin DNA đã được lưu

$DNA(T) = \{(t_{i,1}, t_{i,2});\ i = 1,2, \dots, N\}$.

Về lý thuyết, cá thể S chính là T nều và chỉ nếu

$(\wedge_{i=1}^{N}[\{s_{i,1}, s_{i,2}\} = \{t_{i,1}, t_{(i,2)}\}]) == TRUE$. (1)

Biểu diễn hình thức

Đặt

$Z_I = \Sigma_{i=1}^{N}((s_{i1} - t_{i1}) + (s_{i2} - t_{i2}))$, (2)

Công thức (1) tương đương

$(\wedge_{i=1}^{N}[\{s_{i,1}, s_{i,2}\} = \{t_{i,1}, t_{(i,2)}\}]) == TRUE \Leftrightarrow Z_I = 0.$

Cho

$Z_I = 0$

$\Leftrightarrow \Sigma_{i=1}^{N}((s_{i1} - t_{i1}) + (s_{i2} - t_{i2})) = 0$

$\Leftrightarrow \Sigma_{i=1}^{N}(s_{i1} + s_{i2}) - \Sigma_{i=1}^{N}(t_{i1} + t_{i2}) = 0$. (3)

Giả sử

$s_{ij}, t_{ij} \in \mathbb{Z}_p$,

với p là số nguyên tố và

$g \in \mathbb{Z}_p$ là phần tử sinh của $\mathbb{Z}_p$.

$(3) \Leftrightarrow g^{\Sigma_{i=1}^{N}(s_{i1}+s_{i2}) - \Sigma_{i=1}^{N}(t_{i1}+t_{i2})} \equiv g^0 \equiv 1\ (mod\ p)$

$\Leftrightarrow g^{\Sigma_{i=1}^{N}(s_{i1}+s_{i2})} g^{-\Sigma_{i=1}^{N}(t_{i1}+t_{i2})} \equiv 1\ (mod\ p)$. (4)

Đẳng thức (4) chính là điều kiện tìm kiếm nhưng không tiết lộ thông tin chi tiết của các DNA.

Kiểm định cha-con hay mẹ-con

Đặc tả

Hai cá nhân T và S có thông tin nhiễm sắc thể lần lượt là

$DNA(T) = \{(t_{i,1}, t_{i,2}); i = 1,2, \dots, N\}$,

$DNA(S) = \{(s_{i,1}, s_{i,2}); i = 1,2, \dots, N\}$,

Có quan hệ cha-co hay mẹ-con nếu và chỉ nếu

$\wedge_{i=1}^{N}\left[\{s_{i,1}, s_{(i,2)}\} \cap \{t_{i,1}, t_{(i,2)}\} \neq \emptyset\right] == TRUE$ (5).

Biểu diễn hình thức

Đặt

$z_i = (s_{i,1} - t_{i,1})(s_{i,1} - t_{i,2})(s_{i,2} - t_{i,1})(s_{i,2} - t_{i,2})$

$= \Big(\big(s_{i,1} - t_{i,1}\big)\big(s_{i,1} - t_{i,2}\big)\Big)\Big(\big(s_{i,2} - t_{i,1}\big)\big(s_{i,2} - t_{i,2}\big)\Big)$.

và

$Z_p = \Sigma_{i=1}^{N} z_i$.

Thì

$(5) \Leftrightarrow Z_p = 0$.

Hay

$\Sigma_{i=1}^{N}\Big(\big(s_{i,1} - t_{i,1}\big)\big(s_{i,1} - t_{i,2}\big)\Big)\Big(\big(s_{i,2} - t_{i,1}\big)\big(s_{i,2} - t_{i,2}\big)\Big) = 0$ (6).

Đẳng thức (6) chính là điều kiện tìm kiếm nhưng không tiết lộ thông tin chi tiết của các DNA.

Cài đặt giao thức cơ bản

Các kiểm định trên, đều cần kiểm tra điều kiện cơ bản

$Z = (s_1 - t_1)(s_2 - t_2) = s_1 s_2 - s_1 t_2 - t_1 s_2 + t_1 t_2$. (7)

Do các giá trị $s_i, t_i; i = 1,2$ thường nhỏ. Để chống vét cạn, ta sẽ sử dụng giao thức ElGamal hay Weil-Pairing để xây dựng giao thức kiểm tra đẳng thức (7).

Giao thức dựa trên ElGamal

Giả sử cặp khóa (khóa công khai, khóa cá nhân) tương ứng là:

$[PK, pk] \equiv [(g, g^{\alpha}), (\alpha, r)]$.

Giao thức thực hiện các tương tác sau:

(**1**) T gửi cho máy chủ S thông tin

$(y_1, y_2, y_3) = (g^{t_1} g^{\alpha r}, g^{t_2} g^{\alpha r}, g^{t_1 t_2} g^{\alpha r})$.

(**2**) S tính

$(z_1, z_2, z_3) = \big(y_1^{-s_2}, y_2^{-s_1}, g^{s_1 s_2}\big)$,

$(v_1, v_2) = (z_1 z_2 z_3 y_3, g^{-s_1 - s_2 + 1}) = \big(g^{Z} g^{\alpha r(-s_1 - s_2 + 1)}, g^{-s_1 - s_2 + 1}\big)$.

Gửi (v_1, v_2) lại cho T.

(**3**) T kiểm tra

$v_1(v_2)^{-1} == g^{\alpha r}$.

Giao thức dựa trên Weil-Pairing trên đường cong elliptic

Giả sử cặp khóa (khóa công khai, khóa cá nhân) tương ứng là:

$[PK, pk] \equiv [(P, \alpha P), (\alpha, r)]$.

Trước hết, từ đẳng thức (7)

$Z = s_1 s_2 - s_1 t_2 - s_2 t_1 + t_1 t_2$,

ta có

$Z \equiv 0 \Leftrightarrow Z' = \frac{s_1 s_2}{t_1 t_2} - \frac{s_1}{t_1} - \frac{s_2}{t_2} + 1 \equiv 0 \ (mod\ p)$,

với p là số nguyên tố.

Đặt

$\mu_i = \frac{s_i}{t_i}$,

Ta có,

$\mu_1 + \mu_2 - \mu_1 \mu_2 \equiv 1 \ (mod\ p)$.

Giao thức thực hiện các tương tác sau:

(**1**) T tính và gửi cho máy chủ S

$(y_1, y_2, y_3) = (g^{-rt_1}, g^{-rt_2}, g^{-rt_1 t_2})$,

(**2**) S tính

$v_1 = y_1^{s_1} = g^{-rs_1 t_1} = g^{r\mu_1}$,

$v_2 = y_2^{s_2} = g^{-rs_2 t_2} = g^{r\mu_2}$,

$v_3 = y_3^{-s_1 s_2} = g^{-r\mu_1 \mu_2}$,

$v = v_1 v_2 v_3 = g^{(Z')r}$.

Gửi cho T giá trị v.

(**3**) T kiểm tra đẳng thức sau:

$v == g^r$.

A1 KIỂM ĐỊNH CHI-BÌNH PHƯƠNG

PHÂN PHỐI CHI-BÌNH PHƯƠNG

Hàm Gamma

Hàm Gamma, ký hiệu Γ, là hàm được xác định theo

$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}e^{-x}dx.$

Với mọi $\alpha > 0$, $\Gamma(\alpha)$ luôn cho giá trị hữu hạn và có các tính chất sau:

(1) $\Gamma(\alpha) = (\alpha-1)\Gamma(\alpha-1)$,

(2) $\Gamma(n) = (n-1)!, \forall n > 0.$

Phân phối χ^2

Phân phối $\chi^2(r)$, đọc là chi-bình phương $r \in \mathbb{R}$ bậc tự do ($\boldsymbol{\chi - square(r)}$), là phối phối của biến ngẫu nhiên χ^2 được định nghĩa bởi

$\chi^2 = Z_1^2 + \cdots + Z_r^2,$

với $Z_1, \dots, Z_r$ là r biến ngẫu nhiên có phân phối chuẩn tắc $N(0,1)$.

Phân phối χ^2 thường được áp dụng để ước lượng phương sai của một phân phối chuẩn, đồng thời đóng vai trò quan trọng trong kiểm định giả thuyết về dáng điệu các phân phối xác suất, kiểm định χ^2 (***Chi-square test***).

Định lý (Pearson). Giả sử $X_1, \dots, X_n$ là dãy các biến ngẫu nhiên độc lập và có cùng phân phối xác suất với biến ngẫu nhiên X nhận giá trị trong tập $\{x_1, \dots, x_s\}$ với xác suất $\Pr(X = x_i) = p_i > 0$ và $\Sigma_{i=1}^s p_i = 1$. Với mỗi n, gọi $v_i = v_{i,n}$ là biến ngẫu nhiên: v_i là số lần xuất hiện x_i trong dãy $X_1, \dots, X_n$, nghĩa là $\Sigma_{i=1}^s v_i = n$. Khi đó

$\sum_{i=1}^s \frac{(v_i - np_i)^2}{np_i} \xrightarrow{d} \chi_{s-1}^2.$

(biến ngẫu nhiên $\sum_{i=1}^s \frac{(v_i-np_i)^2}{np_i}$ *hội tụ theo phân phối* về χ_{s-1}^2, khi n tiến tới vô cùng.)

Định lý. Phân phối $\chi^2(r), r > 0$ có mật độ xác suất là

$$f(x) = \begin{cases} \frac{1}{2^{\frac{r}{2}}\Gamma(r/2)} x^{\frac{r}{2}-1}e^{-\left(\frac{x}{2}\right)}, & x > 0 \\ 0 & x \le 0 \end{cases}.$$

KIỂM ĐỊNH CHI-BÌNH PHƯƠNG

Kiểm định $\boldsymbol{\chi - square}$

Khi muốn kiểm định xem hai biến ngẫu nhiên có độc lập với nhau không, có thể sử dụng χ^2.

Giả sử biến ngẫu nhiên X nhận m giá trị $x_1, \dots, x_m$, và biến ngẫu nhiên Y nhận n giá trị $y_1, \dots, y_n$. Kiểm định giả thuyết H_0: X, Y độc lập, nghĩa là

$$P(i,j) = P\big(X = x_i, Y = y_j\big) = P(X = x_i)P\big(Y = y_j\big), \forall i, j.$$

Không gian xác suất trong trường hợp này có mn phần tử $(X = x_i, Y = y_j)$. Và mô hình xác suất có $m + n - 2$ tham số, nghĩa là nếu ước lượng được $m + n - 2$ giá trị $P(X = x_1), \dots, P(X = x_{m-1}, P(Y = y_1), \dots, P(Y = y_{n-1})$, thì biết được toàn bộ phân phối xác suất của không gian xác suất nếu chấp nhận giả thuyết $P\big(X = x_i, Y = y_j\big) = P(X = x_i)P\big(Y = y_j\big)$, với mọi i, j. Do đó, số bậc tự do của phân phối χ^2 cần dùng trong kiểm định giả thuyết H_0 là $mn - (m + n - 2) - 1 = (m - 1)(n - 1)$.

Ví dụ

Khảo sát 50 học sinh về một hoạt động ngoại khóa của lớp, thu được kết quả như bảng sau

	Thích	Không thích	Tổng
Trai	18 (A)	12 (B) 12	30
Gái	7 (C)	13 (D) 7	20 14
Tổng	25	25 19	50 44

H_0: có quan hệ giữa phái tính và sở thích này hay không.

Kiểm định χ^2so sánh tần số đếm được trong mỗi ô của bảng với kỳ vọng tần số đếm được cho mỗi ô (giá trị phải ước tính). Giá trị χ^2cho mỗi ô trong bảng được tính theo công thức

$$\chi^2 = \sum \frac{(O-E)^2}{E},$$

trong đó O là tần suất quan sát được trong mỗi ô, và E là kỳ vọng. Các bước thủ tục kiểm định χ^2 ($\chi^2 - test$) như sau:

(**Bước 1**) Tính kỳ vọng cho các ô (từ (A) đến (D))

$$E_A = \frac{25\times30}{50} = 15, E_A = \frac{25\times30}{44} = 17.04$$

$$E_B = \frac{25\times30}{50} = 15, E_B = \frac{19\times30}{44} = 12.95$$

$$E_C = \frac{25\times20}{50} = 10, E_C = \frac{25\times14}{44} = 7.95$$

$$E_D = \frac{25\times20}{50} = 10. E_D = \frac{19\times14}{44} = 6.04$$

(**Bước 2**) Tính giá trị χ^2 cho mỗi ô

$$\chi_A^2 = \frac{(18-15)^2}{15} = 0.6, \chi_A^2 = \frac{(18-17.04)^2}{17.04} = 0.05$$

$$\chi_B^2 = \frac{(12-15)^2}{15} = 0.6, \chi_B^2 = \frac{(12-12.95)^2}{12.95} = 0.07$$

$\chi_C^2 = \frac{(7-10)^2}{10} = 0.9$, $\chi_C^2 = \frac{(7-7.9)^2}{7.9} = 0.1$

$\chi_D^2 = \frac{(13-10)^2}{10} = 0.9$. $\chi_D^2 = \frac{(7-6.04)^2}{6.04} = 0.15$

(**Bước 3**) Tính tổng các biến χ^2

$\chi^2 = 0.6 + 0.6 + 0.9 + 0.9 = 3.0$. $\chi^2 = 0.05 + 0.07 + 0.1 + 0.15 = 0.37$

(**Bước 4**) Suy luận

Bậc tự do trong ví dụ này là $r = 1$. Với giá trị $p \leq 0.05$, tra bảng phân phối χ^2 được $3.841 > 3.0 = \chi^2$, bác bỏ giả thuyết H_0.