Phát triển ứng dụng Web

NodeJs – part 3
Xác thực tài khoản và phân quyền

Nội dung

- ❏ Authentication
- ❏ Authorization
- ❏ Accounting

Nội dung

- **Authentication**
- Authorization
- Accounting

Quản lý người dùng hệ thống

- Trong ứng dụng, việc thay đổi thông tin dữ liệu hợp lý và nhất quán có ý nghĩa rất quan trọng. Vì vậy việc xác định người dùng nào được phép làm những gì đối với hệ thống là bắt buộc.
- Ngoài ra để dễ dàng kiểm soát hệ thống thì việc xây dựng kiến trúc giúp lưu vết hoạt động cũng là yêu cầu cần thiết và bắt buộc đặc biệt với hệ thống lớn.

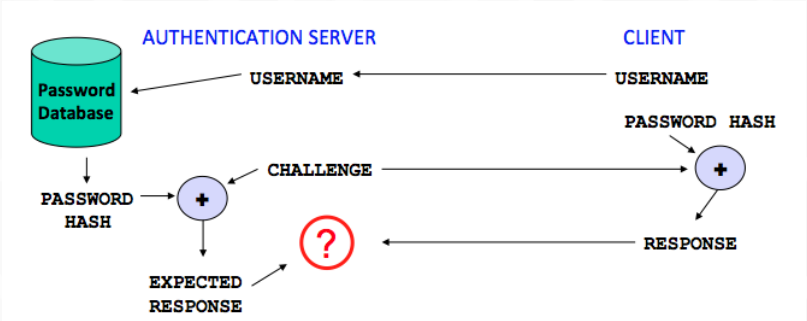
Authentication

- **Authentication** là xác thực, là quá trình kiểm tra danh tính của người dùng hoặc một hệ thống khác đến hệ thống hiện tại thông qua một hệ thống xác thực.
- Đây là bước ban đầu của mọi hệ thống có yếu tố người dùng. Nếu không có bước xác thực này, hệ thống sẽ không biết được người đang truy cập vào hệ thống là ai để có các phản hồi phù hợp.
- Một số phương thức xác thực thông dụng
  - Mật khẩu
  - Khóa
  - Sinh trắc học

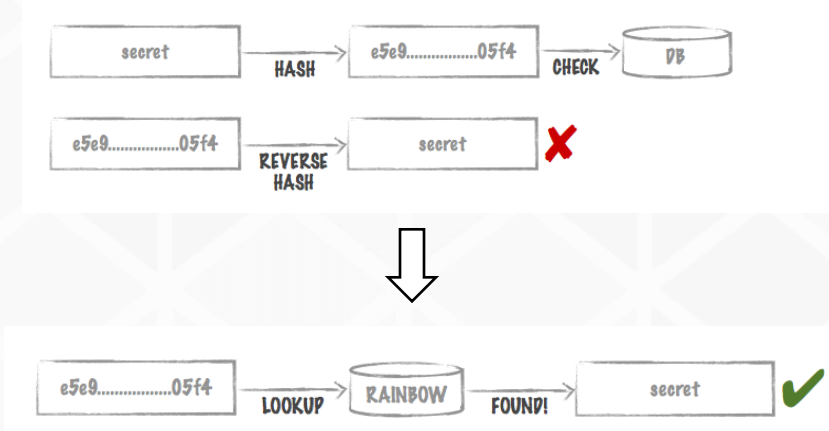
Xác thực bằng mật khẩu (Password & Pin)

- Mật khẩu là một trong những phương pháp đơn giãn và dễ triển khai nhất. Thường mỗi hệ thống sẽ lưu lại mật khẩu ở dạng đã được mã hóa một chiều (***md5, sha1***, …) để đảm bảo mật khẩu có bị lộ cũng không thể khôi phục thành chuỗi gốc.
- Phương pháp này còn có nhiều biến thể như thiết kế dạng ***Swipe Pattern PIN*** (trong các điện thoại android) hoặc mật khẩu dùng một lần (dùng cho các chức năng quan trọng).

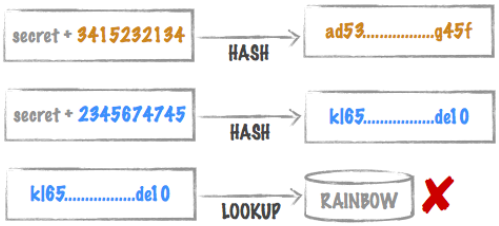
Mô hình cơ bản

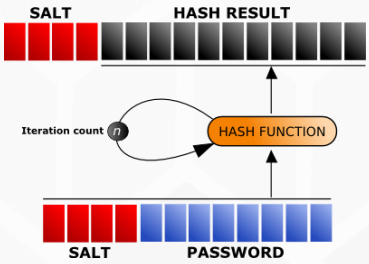


Vấn đề phá mã



Giải pháp thông dụng

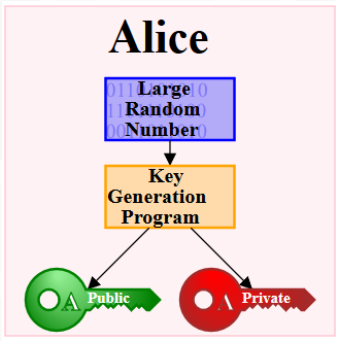




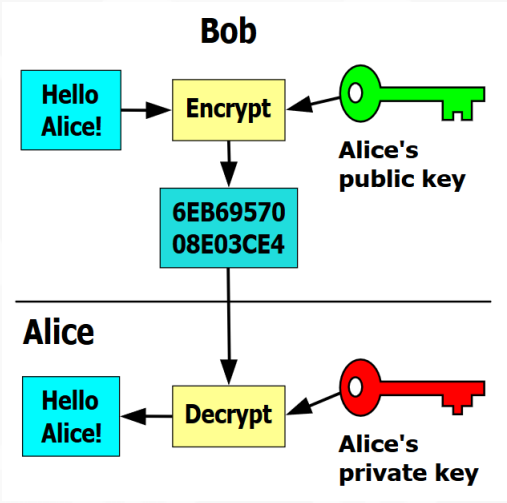
Xác thực bằng khóa công khai

- Phương pháp này dựa trên thuật toán mã hóa khóa công cộng (***public key***) và khóa cá nhân (***private key***). Phương pháp này giúp cho người đăng nhập không cần nhớ thông tin gì về đăng nhập như phương pháp mật khẩu. Để đăng nhập vào hệ thống người dùng chỉ cần có khóa cá nhân (***private key***) trên máy và đăng nhập vào hệ thống (nếu đã khai báo với khóa công cộng của người dùng). Cách này thường được áp dụng và bật với các hệ thống quản trị ***server***.

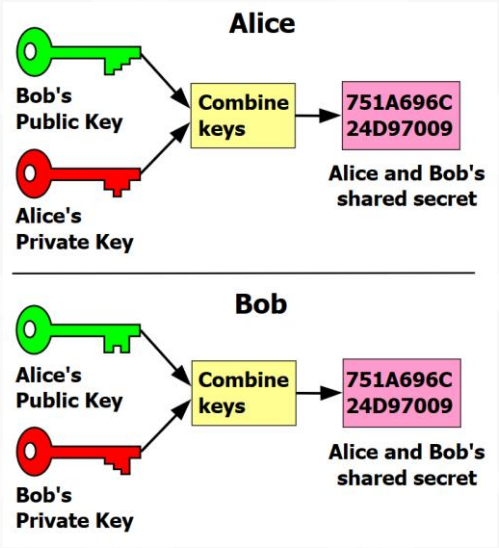
Tạo khóa



Áp dụng



Kết hợp



Mã hóa bất đối xứng

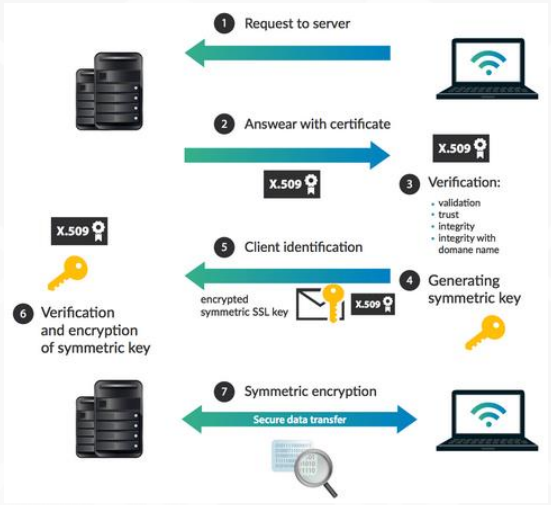
- Thuật toán thông dụng RSA
  - Encryption: $m^e \mod n = c$
  - Decryption: $c^d \mod n = m$
- Ví dụ: e = 17, n = 3233 và d = 2753
  - Mã hóa với m = 42

$$42^{17} \mod 3233 = 2557$$

  - Giải mã

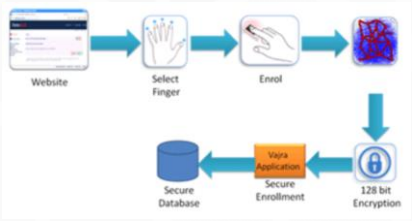
$$2557^{2753} \mod 3233 = 42$$

SSL



Xác thực bằng Sinh học (Biometrics)

- Đây là phương pháp dựa trên các yếu tố đặc trưng bởi người dùng như dấu vân tay, tròng mắt hoặc khuôn mặt. Phương pháp này có cái lợi là người dung không cần nhớ và chỉ dùng nó mỗi khi cần đăng nhập vào hệ thống.



Nội dung

- *Authentication*
- **Authorization**
- Accounting

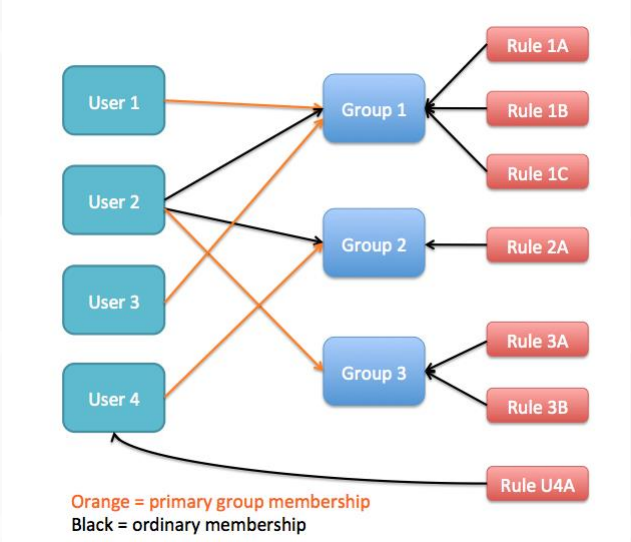
Authorization

- **Authorization** là quá trình xác định xem một người dùng có quyền truy cập một tài nguyên cụ thể hoặc để thực hiện một số hành động hay không.
- Các hình thức phân quyền thường gặp là:
  - ***Role-based authorization***
  - ***Object-based authorization***

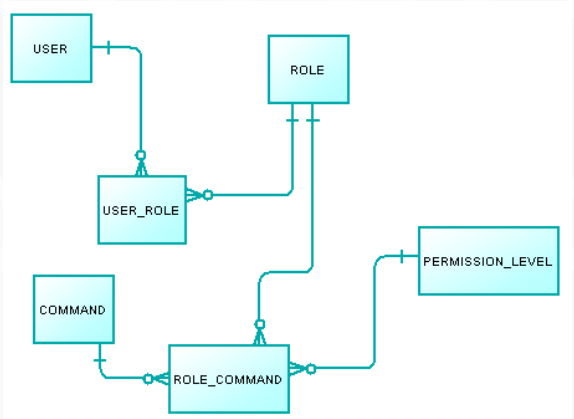
Role-based authorization

- Phân quyền dựa trên vai trò của người dùng. Ví dụ trong WordPress có các role như là Subscriber, Contributor, Author, Editor, Administrator và mỗi một ***role*** sẽ có những quyền khác nhau và mỗi người dùng sẽ được phân role có quyền tương ứng. Đối với những hệ thống có nhiều người dùng thì ***role-based*** là cách tiếp cận tốt nhất để tiết kiệm thời gian trong việc phân quyền.

Cách vận dụng 1

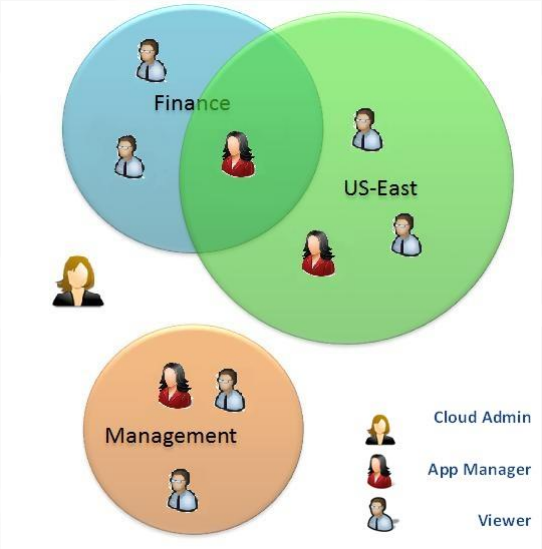


Cách vận dụng 2



Object-based authorization

- Phân quyền theo đối tượng. Kiến trúc phân quyền này sẽ giúp phân quyền được đến từng tài khoản cụ thể hoặc giải quyết các bài toán phân quyền như những người trong nhóm A,B có thể vào edit sản phẩm nhưng người nhóm A chỉ được edit sản phẩm thuộc danh mục X, còn người nhóm B chỉ được edit sản phẩm thuộc danh mục Y…
- Kiến trúc ***object-based*** lý tưởng phải được khai báo động trong database và có thêm lớp ***cache*** để giải quyết vấn đề ***performance*** khi ***check*** quyền.



Nội dung

- ❏ *Authentication*
- ❏ *Authorization*
- ❏ **Accounting**

Accounting (hay Auditing)

- Quá trình cuối cùng của hệ thống phân quyền gọi là ***Accounting*** (hay còn gọi là ***Auditing***), tức là kiểm tra hay ghi ***log***. Quá trình kiểm tra là công đoạn ghi lại các hành động của người dùng sau khi đã thực hiện một chức năng nào đó trong hệ thống.
- Tùy theo nhu cầu kiểm tra (ghi ***log***) mà quyết định nên ghi lại những hành động nào của ***user*** hoặc có thể ghi lại hết nếu hệ thống yêu cầu.
- Việc ghi log có tác dụng rõ ràng là đánh giá, theo dõi hoạt động của user trên hệ thống và kiểm soát khi có sự cố mất mát, sai lệch, rò rĩ thông tin. Nếu được thiết kế tốt, hệ thống log sẽ giúp cho mọi hoạt động của hệ thống được rõ ràng, minh bạch và an toàn.

Nội dung

- Authentication
- Authorization
- Accounting
- Bài tập

Bài tập

- Xây dựng chức năng tạo tài khoản, đăng nhập, đăng xuất, ghi nhớ đăng nhập dựa vào database QLBH sử dụng Postgre.

Bài tập

- Xây dựng chức năng quản lý Category (them, xóa, sửa), xem danh sách các Product thuộc Category với database QLBH sử dụng Postgre.