Access Control

ITU-T Recommendation X.800’s definition

Access control is the prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner

Access control is a critical element in computer security because the main objective of computer security is

- to prevent unauthorized users from accessing resources
- to prevent legitimate users from accessing unauthorized resources,
- to enable users to access resources in an authorized way

RFC 2828’s definition of computer security

Measures that implement and assure security services in a computer system, particularly those that assure access control service

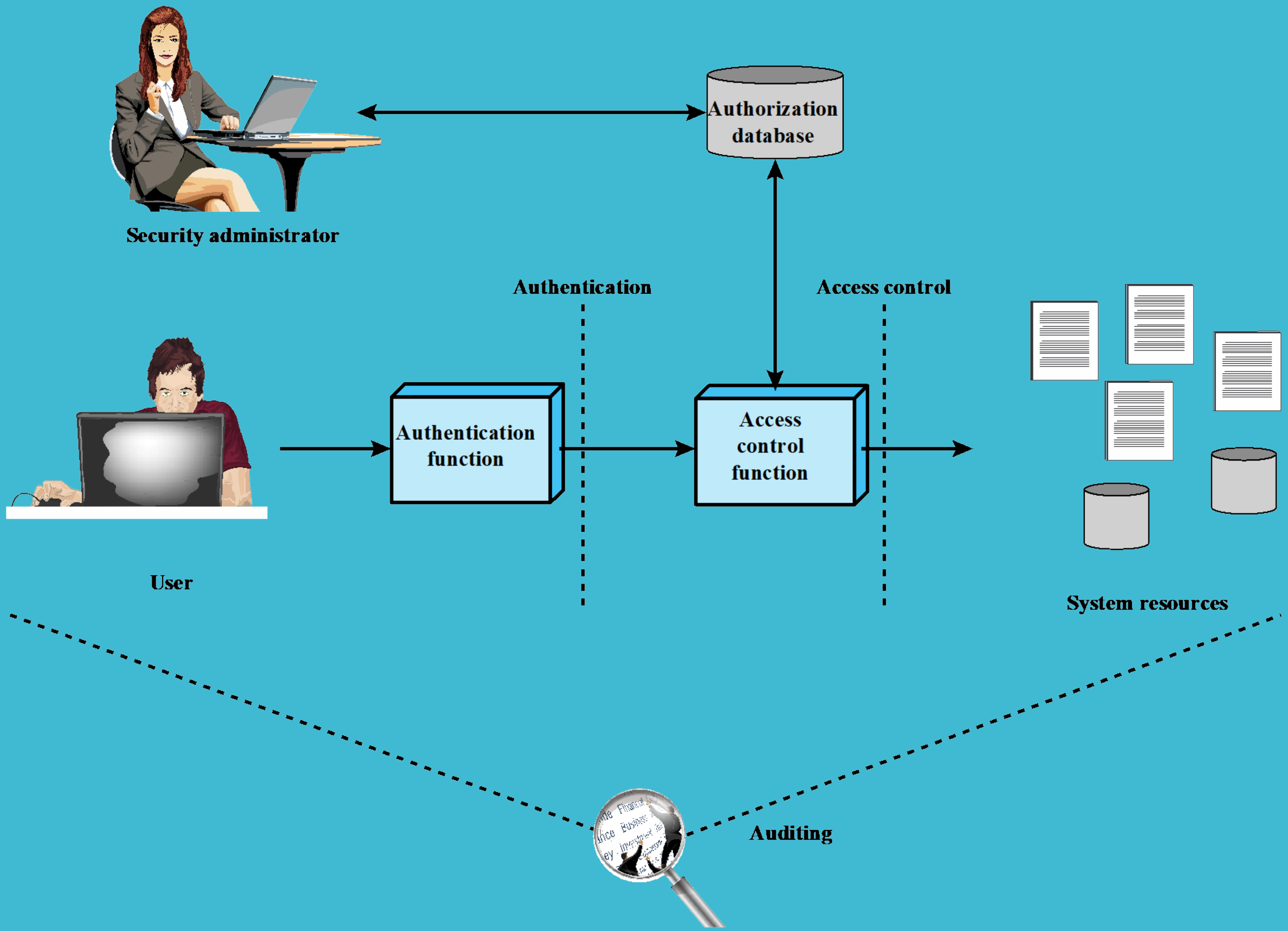


Figure 4.1 Relationship Among Access Control and Other Security Functions

Access Control Policies

Discretionary access control (DAC)

Controls access based on the identity of the requestor and on access rules (authorizations) stating what requestors are (or are not) allowed to do. This policy is termed discretionary because an entity might have access rights that permit the entity, by its own volition, to enable another entity to access some resource.

Mandatory access control (MAC): Controls access based on comparing

Security labels (which indicate how sensitive or critical system resources are) with security clearances (which indicate system entities are eligible to access certain resources). This policy is termed *mandatory because an entity that has* clearance to access a resource may not, just by its own volition, enable another entity to access that resource.

Role-based access control (RBAC): Controls access based on the roles that

users have within the system and on rules stating what accesses are allowed to users in given roles.

Access Control Requirements

- **Reliable Input**

An access control system assumes that a user is authentic; thus, an authentication mechanism is needed as a front end to an access control system. Other inputs to the access control system must also be reliable. For example, some access control restrictions may depend on an address, such as a source IP address or medium access control address. The overall system must have a means of determining the validity of the source for such restrictions to operate effectively.

- **Support for fine and coarse specifications**

The access control system should support fine-grained specifications, allowing access to be regulated at the level of individual records in files, and individual fields within records. The system should also support fine-grained specification in the sense of controlling each individual access by a user rather than a sequence of access requests. System administrators should also be able to choose coarse-grained specification for some classes of resource access, to reduce administrative and system processing burden.

Access Control Requirements

- Reliable Input
- Support for fine and coarse specifications
- Least privilege
- Separation of duty
- Open and Closed policy
- Policy combination and conflict resolution
- Administrative policy
- Dual control

Elements of Access Control

The basic elements of access control are

- **Subject: an entity that accesses object**

  A subject is an application or a user that is represented by a process in the system that takes on the user's attribute, e.g., access right

  Three classes of subject: owner, group, world

- **Object: the resource which access is to be controlled**

  Example: records, files, mailbox, program, messages

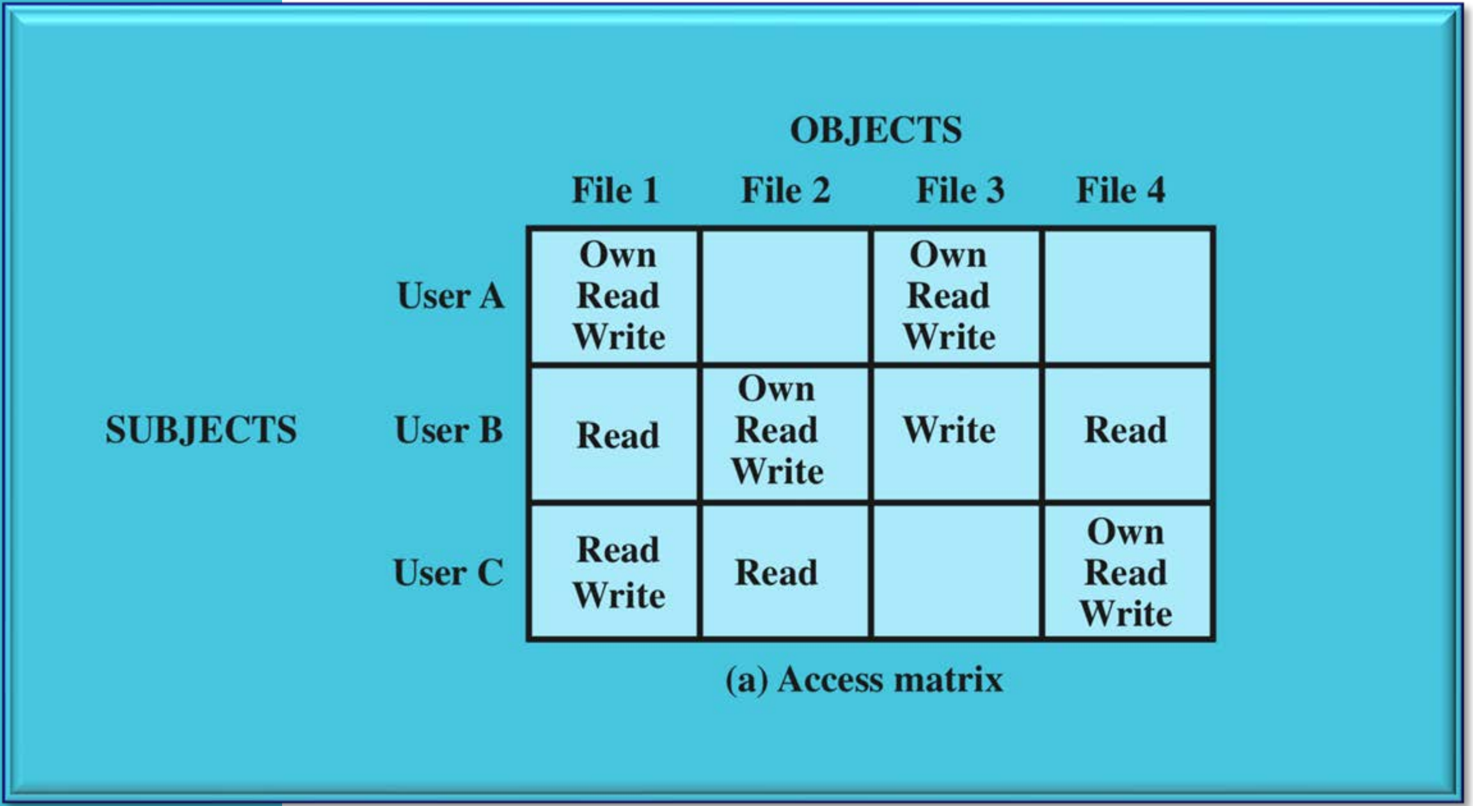
- **Access Right: the way a subject may access an object**

  Access right includes read, write, execute, delete, create, search

Discretionary Access Control

- General access control in OS uses an access matrix
  - One dimension (column) consists of subjects that need to access objects
  - The other dimension (row) lists the objects that can be accessed
  - Each entry in the matrix contains access rights of the subject in that row for the object in that column
- Access matrix is usually sparse, and implemented by decomposing it into one or two ways:
  - By columns resulting in Access Control Lists (ACLs) for all objects
  - By rows resulting in capability list/tickets for all subjects/users
- Each list for an object in ACL lists users and their access rights to access the object
  - ACL may contain a default or public entry to allow users that are not explicitly listed to have a default access right
  - Access rights should follow the least privilege or read-only access
  - Elements in the list can be an individual or group users

From Stallings & Brown textbook



		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

ACL

- Each list for an object in ACL lists users and their access rights to access the object
  - ACL may contain a default or public entry to allow users that are not explicitly listed to have a default access right
  - Access rights should follow the least privilege or read-only access
  - Elements in the list can be an individual or group users

- ACL is efficient when we want to know which subjects have what access rights to a particular object
  - However, it it harder to determine what access rights a specific user has on which objects

- Each capability ticket what access rights a particular user has on the objects in the list
  - Each user has a number of tickets and they can loan or give the tickets to other users
- Tickets may be spread around the system
  - The tickets cause a greater security problem than ACL
- The OS must protect and guarantee the integrity of each ticket; the ticket must be unforgeable
  - OS keeps all tickets for the user in a memory region inaccessible by users
  - Users must use a system call to request for their tickets
- For distributed system, the ticket is in a form of a token
  - A token can be a large random password, or a cryptographic message authentication code whose value is verified by the corresponding resource when requesting for access

Role-based Access Control (RBAC)

- RBAC defines the access rights based on the roles the users assume in the system rather than the user’s identity like in DAC
  - Role is a job function in the organization
  - RBAC assign rights to the roles, not the users
  - Users are assigned roles either statically or dynamically
  - The relationship between users and roles are many-to-many

- Access matrix representation can be used to describe the key elements of RBAC; see Fig. 4.8.

Role-Based Access Control (RBAC)

From Stallings & Brown textbook

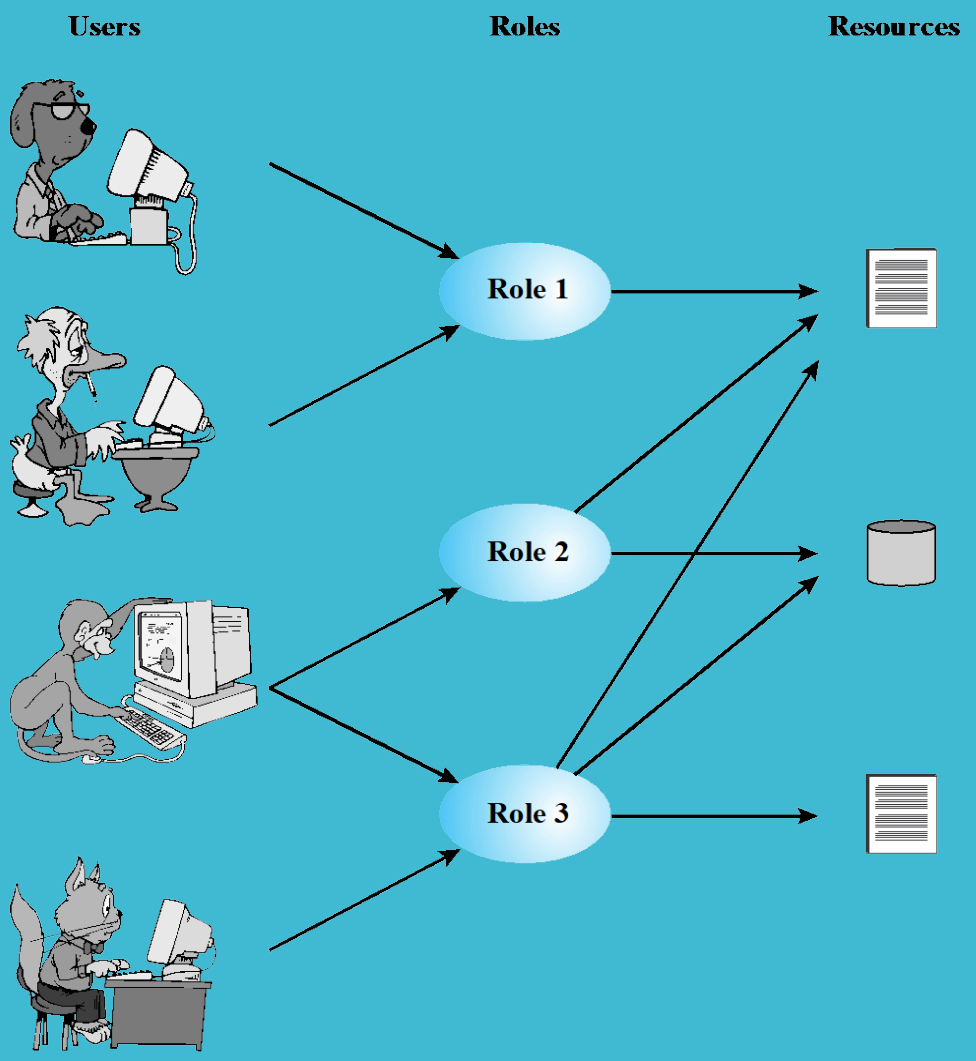
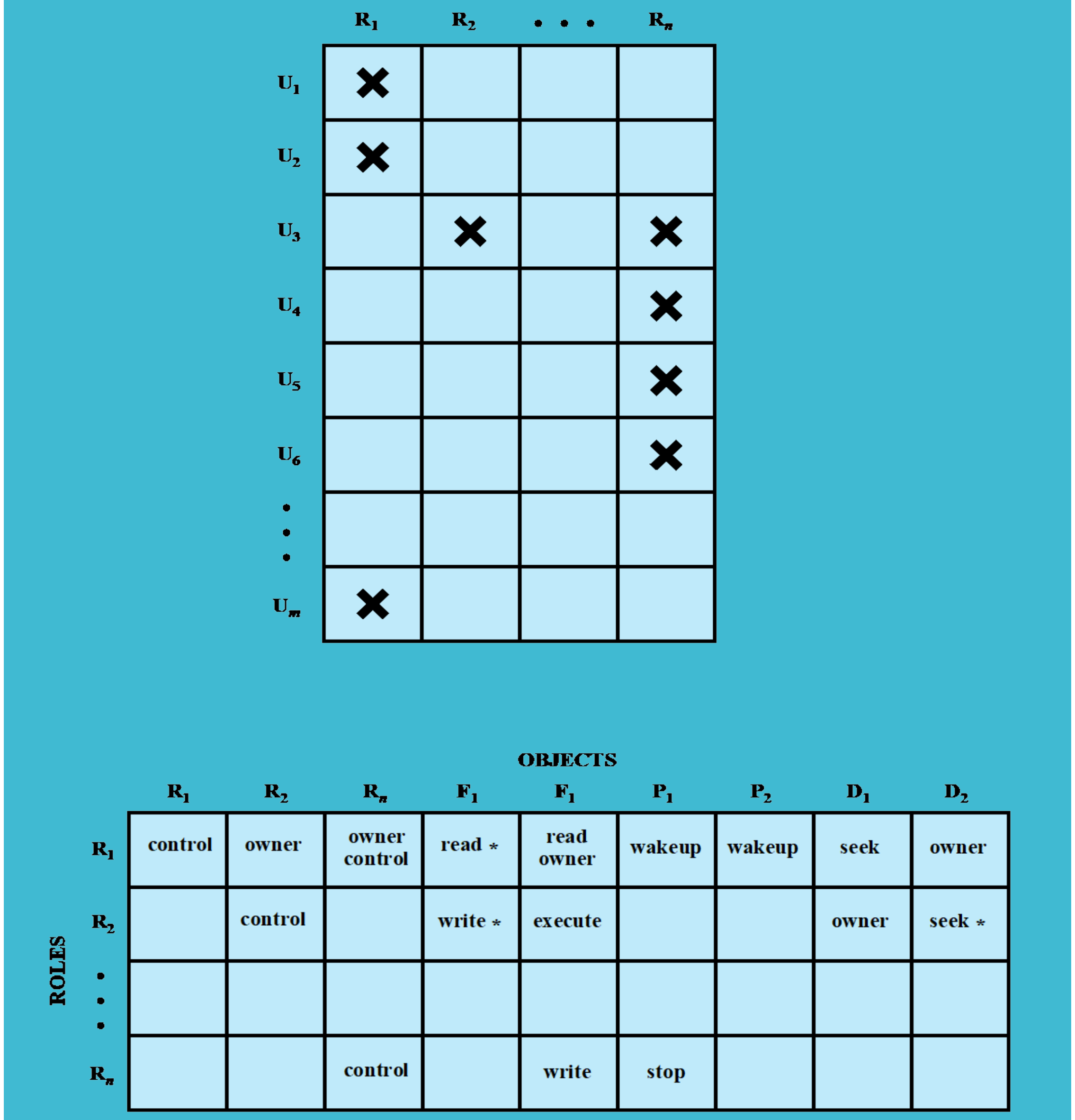


Figure 4.7 Users, Roles, and Resources



	R_1	R_2	• • •	R_n
U_1	✖			
U_2	✖			
U_3		✖		✖
U_4				✖
U_5				✖
U_6				✖
• • •				
U_m	✖			

		OBJECTS								
		R_1	R_2	R_n	F_1	F_1	P_1	P_2	D_1	D_2
ROLES	R_1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
	R_2		control		write *	execute			owner	seek *
	• • •									
	R_n			control		write	stop			

Figure 4.8 Access Control Matrix Representation of RBAC

From Stallings & Brown textbook