Identification, Authentication & Authorization

Dr. Mukti Padhya

Assistant Professor, NFSU

Identification, Authentication & Authorization

- Identification: This is an action in which the user (untrusted party) declares his identity. Identification is the ability to identify uniquely a user of a system or an application that is running in the system

Authentication: This is an action(s) to prove that the user is who he claims to be. Authentication is the act of proving the identity of a computer system user

- Authorization: This action(s) is required to determine which actions a specific user can perform. Authorization is the function of specifying access rights/privileges to resources

Identification

- Here the system has to trust an untrusted party and hence may appear irrelevant.
- However for authentication, proper identification is mandatory
- Once identification is established, the system has to validate that one individual
- E.g. Automatic identification of personnel by scanning the badge at the entry gate instead of manual identification by security personnel to prevent delay and long queues at the entrance

Authentication

- This is the most attacked process and is the weakest link in the security chain
- Authentication can be performed in three different ways
  - Something you know - This is a way to identify a user using something like a PIN, a password, or a passphrase
  - Something you have - This is a way to identify a user using something like a smart card or a badge
  - Something you are - This is a way to identify a user using biometrical characteristics of the user

MFA

- The three authentication mechanisms have inherent weaknesses and hence cannot be used alone
- Each option has different type of weakness and they can be combined to create a secure system
- Hence Multi Factor Authentication is used to provide two or more verification factors to gain access to a resource
- MFA reduces likelihood of a successful cyber attack
- Multifactor authentication combines two or more independent credentials: what the user knows, such as a password; what the user has, such as a security token; and what the user is, by using biometric verification methods

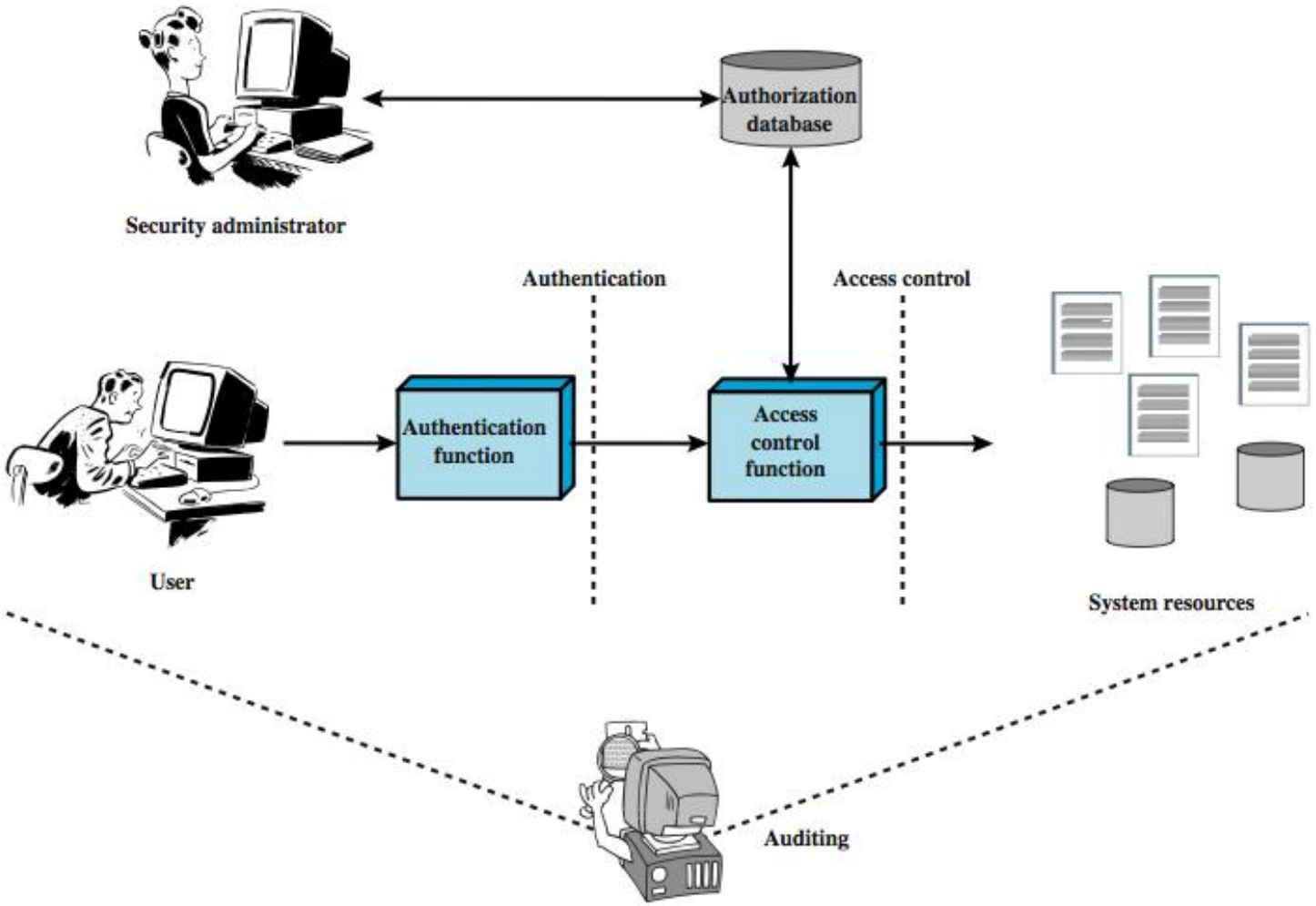
Authorization

- The goal of authorization is to be sure that the given user has clearance to do what he is asking to do
- There are multiple ways to grant privileges to a user, based on the access control model the system uses
- The primary goal of controls is to ensure the confidentiality and integrity of data by disallowing unauthorized access by authorized or unauthorized subjects
- The main access control models are
- Mandatory Access Control (MAC)
- Discretionary Access Control (DAC)
- Role-based Access Control (RBAC)
- Lattice-based Access Control (LBAC)

Access Control

- "The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner"
- Central element of computer security
- Assume have users and groups
  - authenticate to system
  - assigned access rights to certain resources on system

Access Control Principles



Access control policies

- **Discretionary** access control (DAC): based on the identity of the requestor and access rules
- **Mandatory** access control (MAC): based on comparing security labels with security clearances (mandatory: one with access to a resource cannot pass to others)
- **Role-based** access control (RBAC): based on user roles
- **Attribute-based** access control: based on the attributes of the user, the resources and the current environment

Access Control Requirements

- Reliable input: a mechanism to authenticate
- Fine and coarse specifications: regulate access at varying levels (e.g., an attribute or entire DB)
- Least privilege: min authorization to do its work
- Separation of duty: divide steps among different individuals
- Open and closed policies: accesses specifically authorized or all accesses except those prohibited
- Policy combinations and conflict resolution
- Administrative policies: who can add, delete, modify rules

Access Control Elements

- Subject: entity that can access objects
  - a process representing user/application
  - often have 3 classes: **owner**, **group**, **world**
- Object: access controlled resource
  - e.g. files, directories, records, programs etc
  - number/type depend on environment
- Access right: way in which subject accesses an object
  - e.g. read, write, execute, delete, create, search

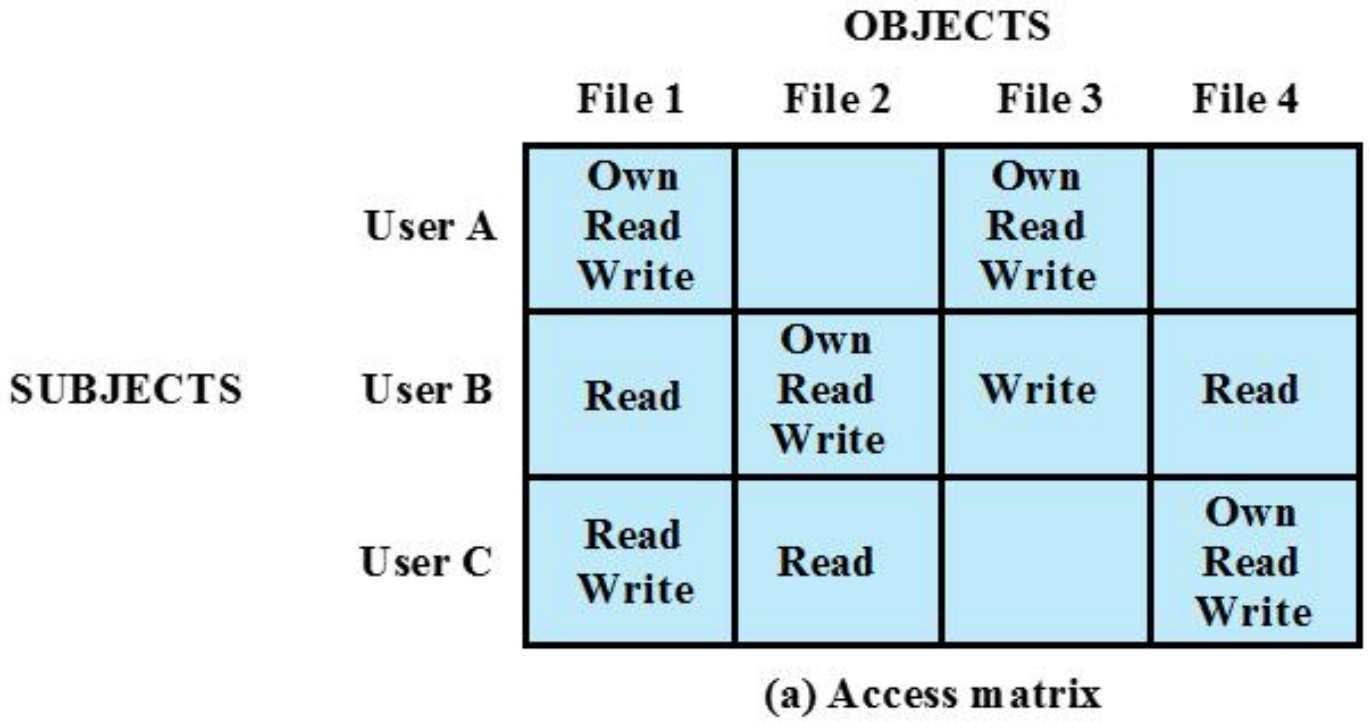
Discretionary Access Control

- Often provided using an access matrix
  - lists subjects in one dimension (rows)
  - lists objects in the other dimension (columns)
  - each entry specifies access rights of the specified subject to that object
- Access matrix is often sparse
- Can decompose by either row or column

Access Control Structures

- Access control lists (decomposed by column)
- Capability tickets (decomposed by row)

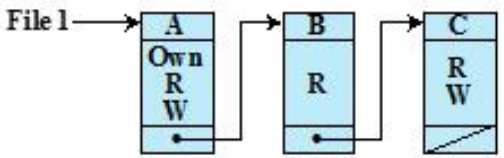
An access matrix

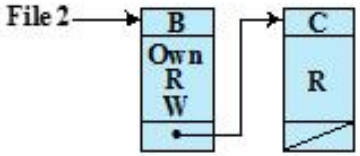


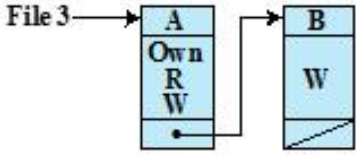
SUBJECTS	OBJECTS: File 1	File 2	File 3	File 4
User A	Own Read Write		Own Read Write	
User B	Read	Own Read Write	Write	Read
User C	Read Write	Read		Own Read Write

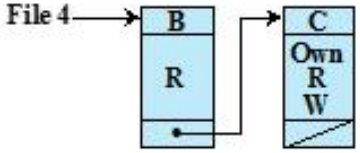
(a) Access matrix

Access matrix data structures

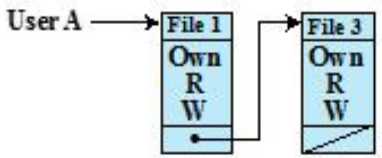


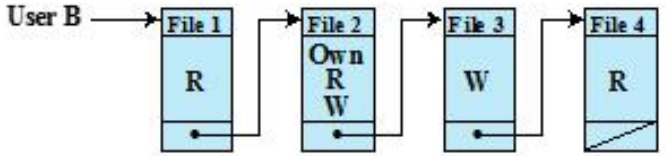


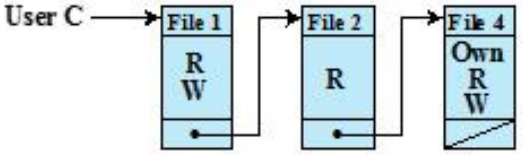




(b) Access control lists for files of part (a)







(c) Capability lists for files of part (a)

Alternate authorization table

Subject	Access Mode	Object
A	Own	File 1
A	Read	File 1
A	Write	File 1
A	Own	File 3
A	Read	File 3
A	Write	File 3
B	Read	File 1
B	Own	File 2
B	Read	File 2
B	Write	File 2
B	Write	File 3
B	Read	File 4
C	Read	File 1
C	Write	File 1
C	Read	File 2
C	Own	File 4
C	Read	File 4
C	Write	File 4

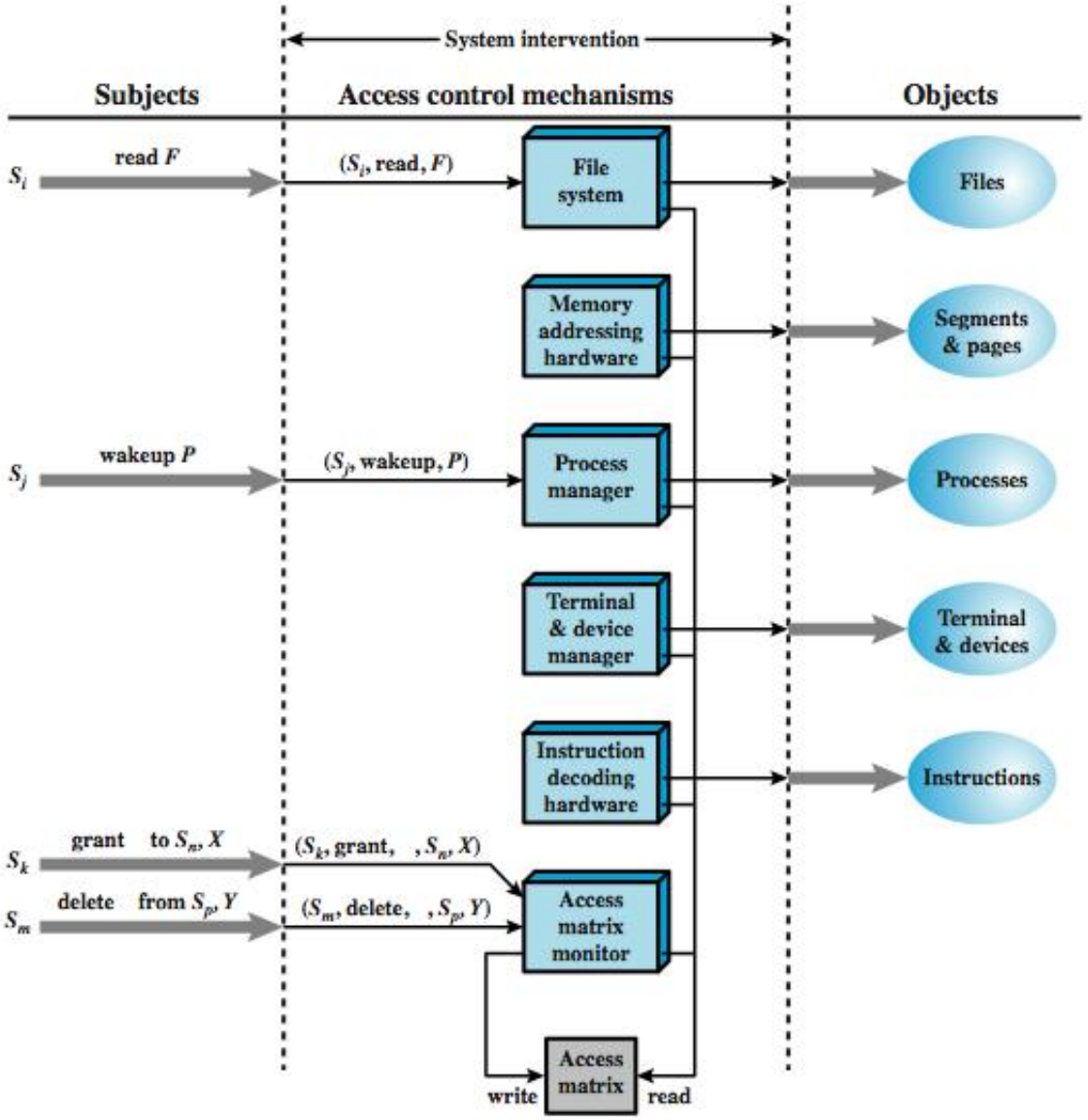
An Access Control Model

- Extend the universe of objects to include processes, devices, memory locations, subjects

		OBJECTS								
		subjects			files		processes		disk drives	
		S_1	S_2	S_3	F_1	F_1	P_1	P_2	D_1	D_2
SUBJECTS	S_1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
	S_2		control		write *	execute			owner	seek *
	S_3			control		write	stop			

* - copy flag set

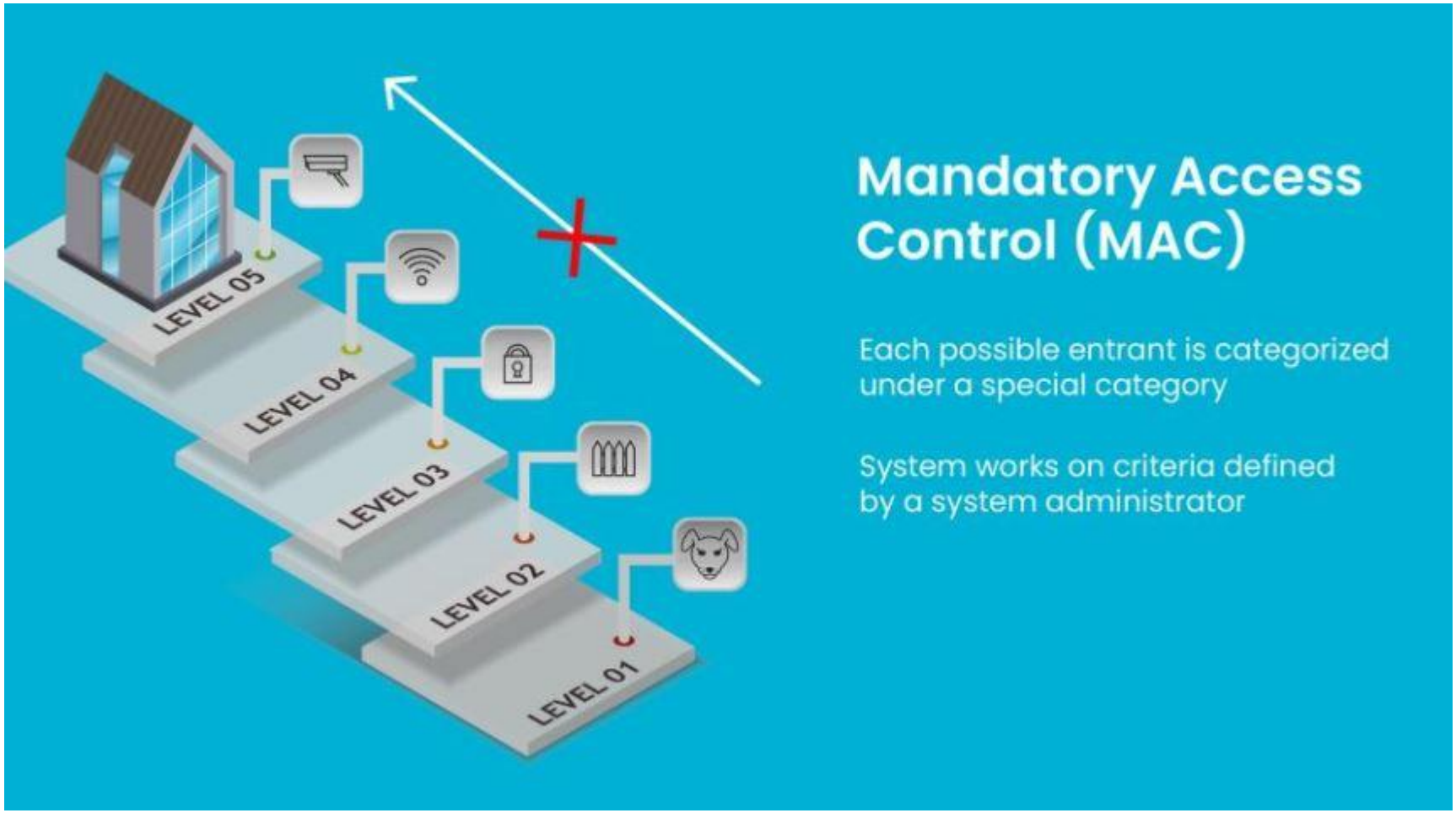
Access Control Function



Access control system commands

Rule	Command (by S_0)	Authorization	Operation
R1	**transfer** $\begin{Bmatrix}\alpha^* \\ \alpha\end{Bmatrix}$ **to** S, X	'α^*' in $A[S_0, X]$	store $\begin{Bmatrix}\alpha^* \\ \alpha\end{Bmatrix}$ in $A[S, X]$
R2	**grant** $\begin{Bmatrix}\alpha^* \\ \alpha\end{Bmatrix}$ **to** S, X	'owner' in $A[S_0, X]$	store $\begin{Bmatrix}\alpha^* \\ \alpha\end{Bmatrix}$ in $A[S, X]$
R3	**delete** α **from** S, X	'control' in $A[S_0, S]$ or 'owner' in $A[S_0, X]$	delete α from $A[S, X]$
R4	$w \leftarrow$ **read** S, X	'control' in $A[S_0, S]$ or 'owner' in $A[S_0, X]$	copy $A[S, X]$ into w
R5	**create object** X	None	add column for X to A; store 'owner' in $A[S_0, X]$
R6	**destroy object** X	'owner' in $A[S_0, X]$	delete column for X from A
R7	**create subject** S	none	add row for S to A; execute **create object** S; store 'control' in $A[S, S]$
R8	**destroy subject** S	'owner' in $A[S_0, S]$	delete row for S from A; execute **destroy object** S

MAC –Mandatory Access Control



Advantages: Mandatory Access Control (MAC)

High-level data protection

Centralized Information

Privacy

Disadvantages: Mandatory Access Control (MAC)

Careful Setting-Up Process

Regular Update Required:

Lack of Flexibility

Bell-LaPadula Model, Step 1

- Security levels arranged in linear ordering
  - Top Secret: highest
  - Secret
  - Confidential
  - Unclassified: lowest
- Levels consist of *security clearance* $L(s)$
  - Objects have *security classification* $L(o)$

Example

security level	*subject*	*object*
Top Secret	Tamara	Personnel Files
Secret	Samuel	E-Mail Files
Confidential	Claire	Activity Logs
Unclassified	Ulaley	Telephone Lists

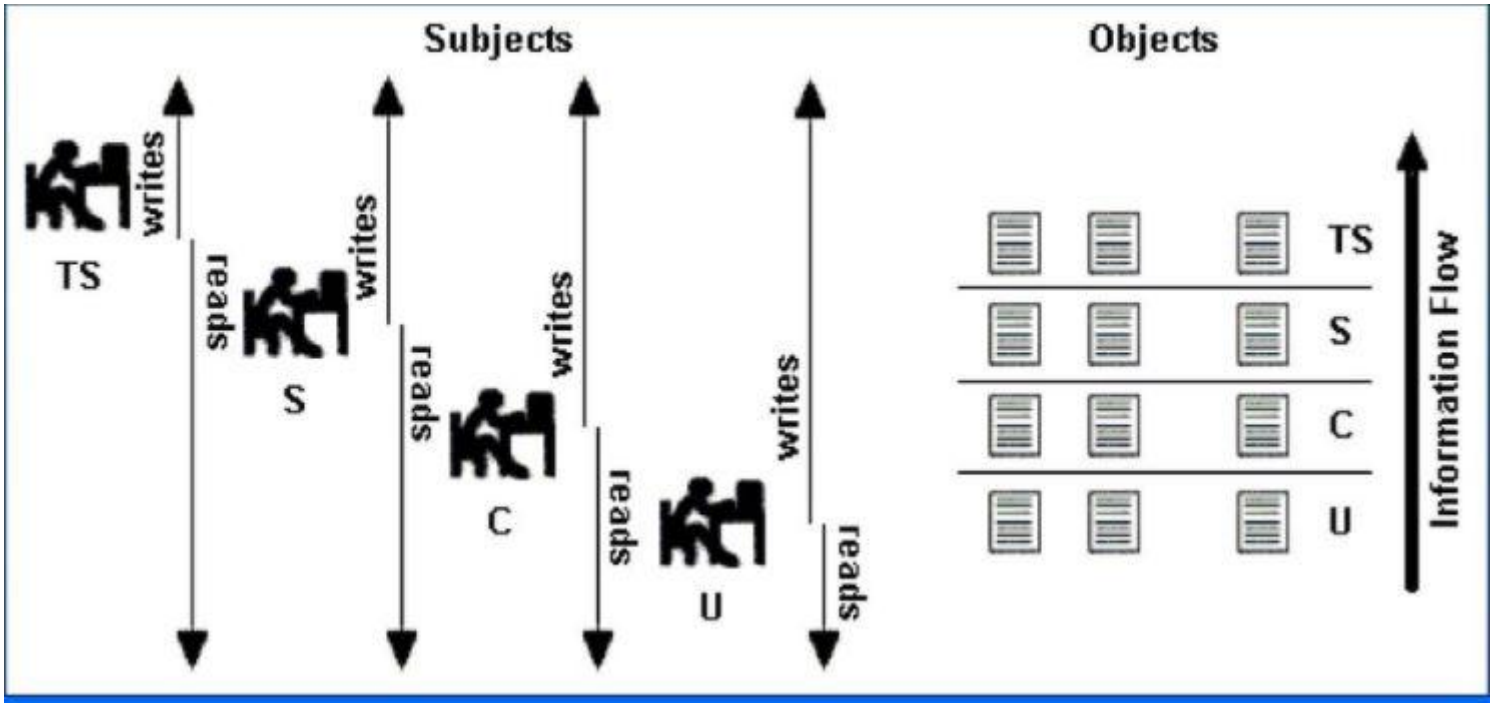
- Tamara can read all files
- Claire cannot read Personnel or E-Mail Files
- Ulaley can only read Telephone Lists

Reading Information

- Information flows *up*, not *down*
  - "Reads up" disallowed, "reads down" allowed
- Simple Security Condition (Step 1)
  - Subject *s* can read object *o* iff, $L(o) \leq L(s)$ and *s* has permission to read *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no reads up" rule

Writing Information

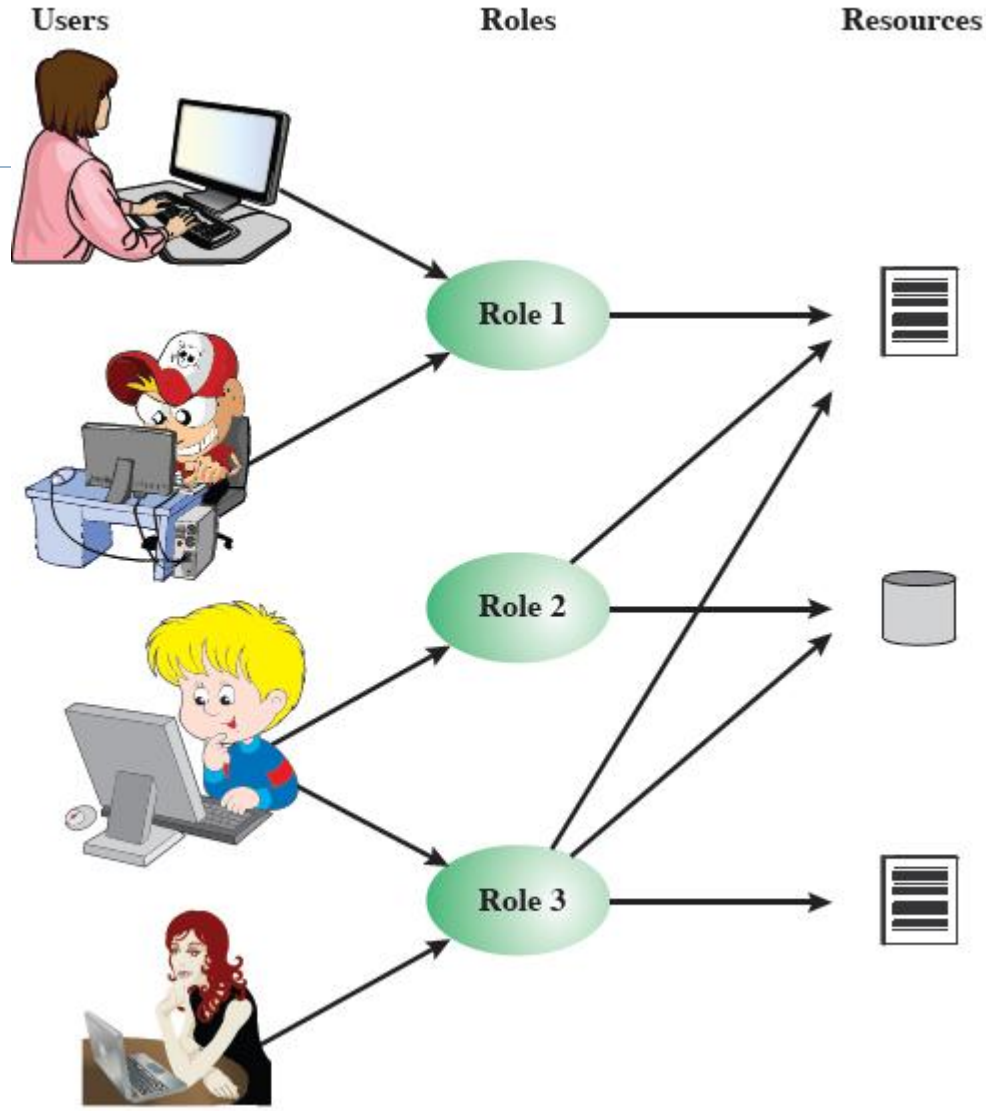
- Information flows up, not down
  - "Writes up" allowed, "writes down" disallowed
- *-Property (Step 1)
  - Subject *s* can write object *o* iff $L(s) \leq L(o)$ and *s* has permission to write *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no writes down" rule


Subjects
Objects
TS
S
C
U
writes
reads
Information Flow

Role-Based Access Control

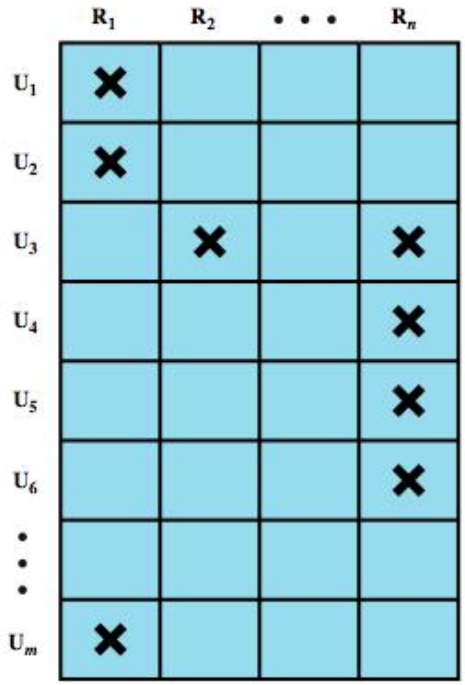
Access based on 'role', not identity

Many-to-many relationship between users and roles



Role-Based Access Control

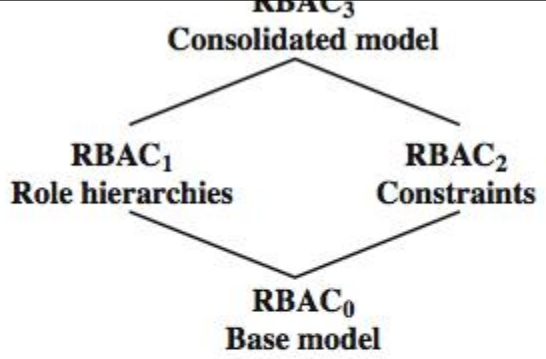
Role-users and roles-object access matrix



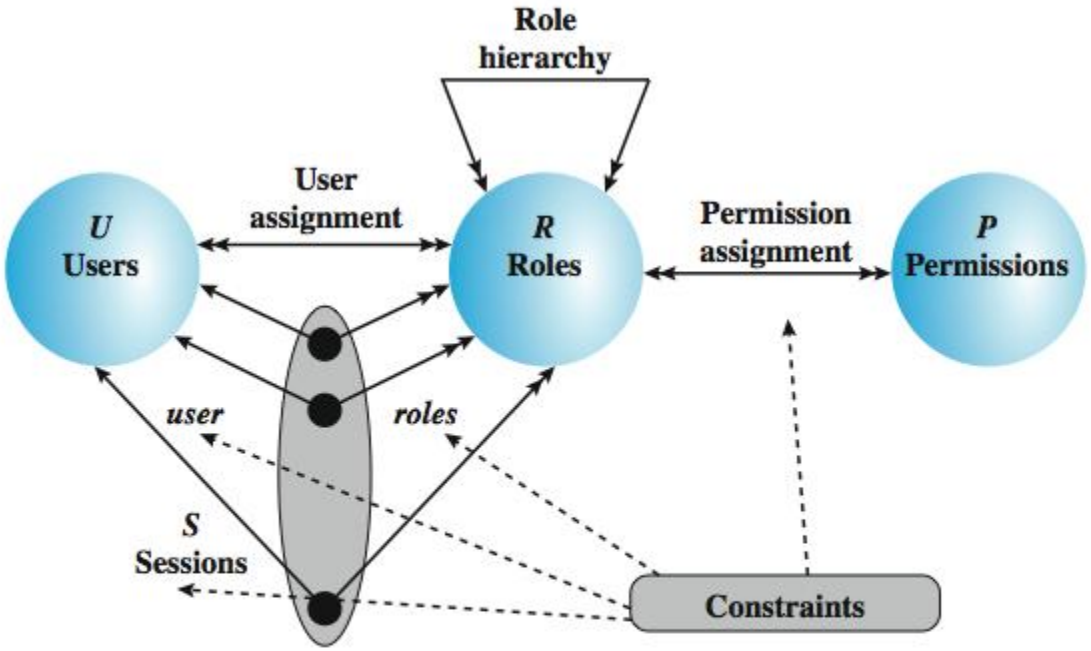
	OBJECTS								
ROLES	R1	R2	Rn	F1	F1	P1	P2	D1	D2
R1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
R2		control		write *	execute			owner	seek *
• • •									
Rn			control		write	stop			

Role-Based Access Control

Double arrow: 'many' relationship
Single arrow: 'one' relationship



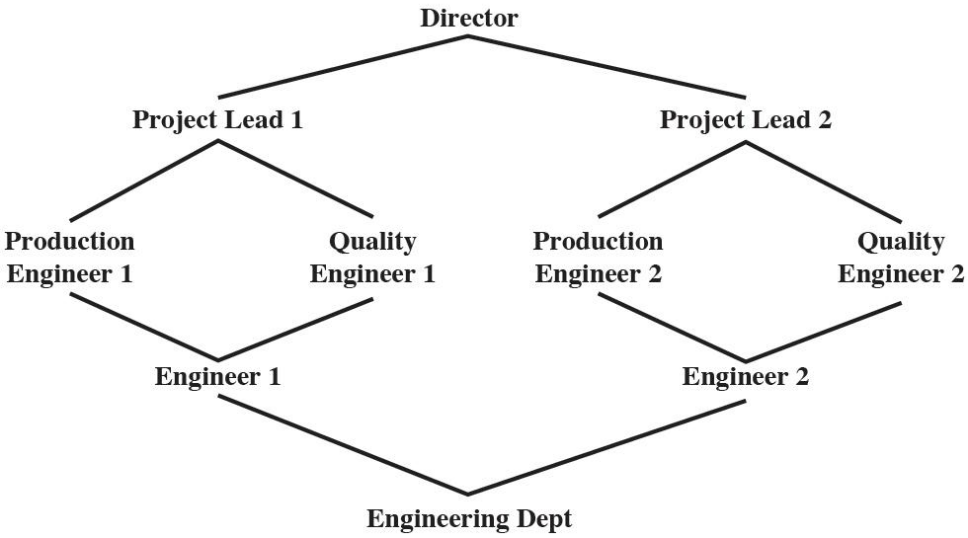
(a) Relationship among RBAC models



(b) RBAC models

Example of role hierarchy

- Director has most privileges
- Each role inherits all privileges from lower roles
- A role can inherit from multiple roles
- Additional privileges can be assigned to a role



Constraints

- A condition (restriction) on a role or between roles
  - **Mutually exclusive**
    - role sets such that a user can be assigned to only one of the role in the set
    - Any permission can be granted to only one role in the set
  - **Cardinality**: set a maximum number (of users) wrt a role (e.g., a department chair role)
  - **Prerequisite role**: a user can be assigned a role only if that user already has been assigned to some other role

Case study: RBAC system for a bank

Role	Function	Official Position
A	financial analyst	Clerk
B	financial analyst	Group Manager
C	financial analyst	Head of Division
D	financial analyst	Junior
E	financial analyst	Senior
F	financial analyst	Specialist
G	financial analyst	Assistant
•••	•••	•••
X	share technician	Clerk
Y	support e-commerce	Junior
Z	office banking	Head of Division

Case study: RBAC system for a bank

- b has more access than A (strict ordering)
- Inheritance makes tables simpler

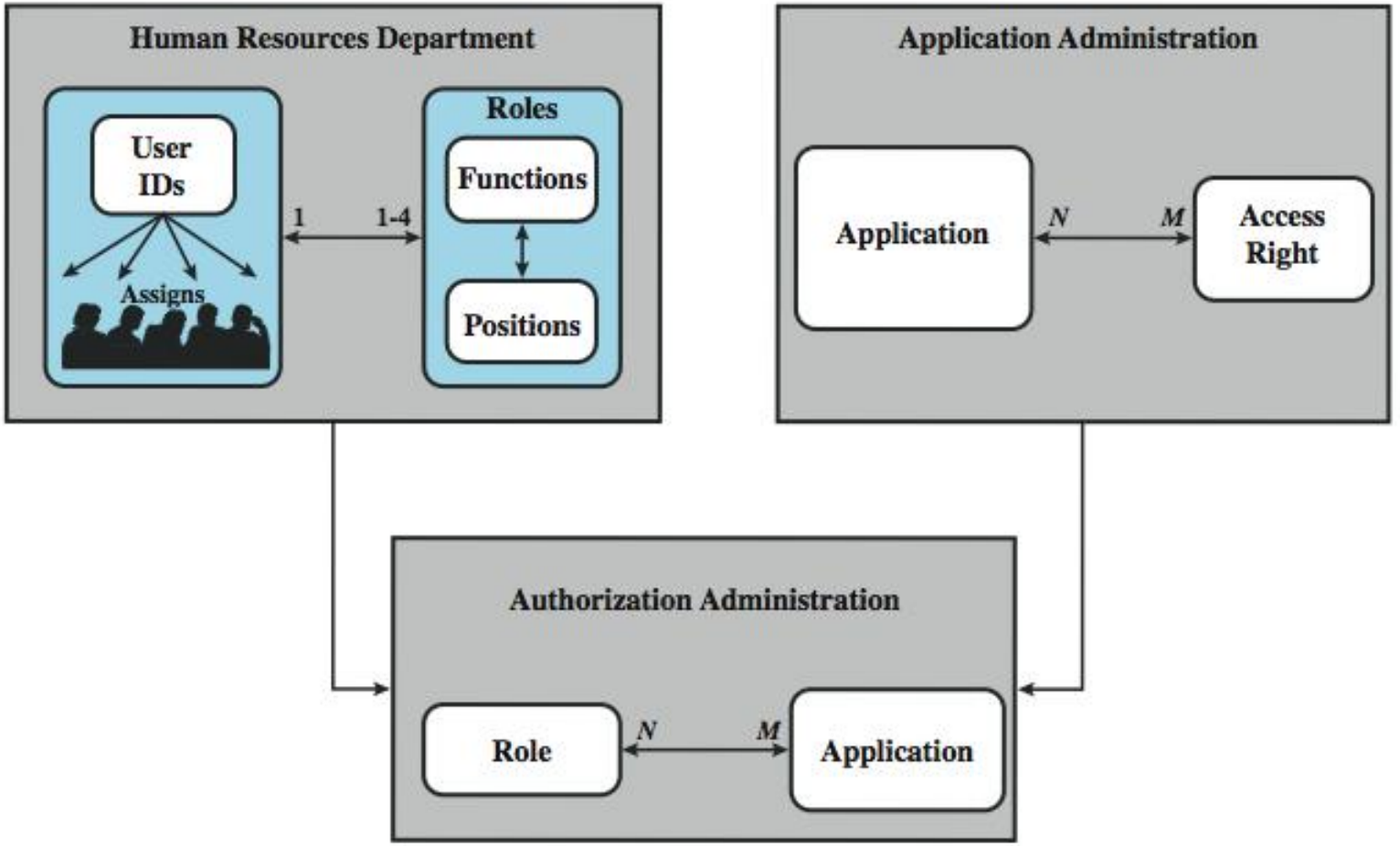
(b) Permission Assignments

Role	Application	Access Right
A	money market instruments	1, 2, 3, 4
	derivatives trading	1, 2, 3, 7, 10, 12
	interest instruments	1, 4, 8, 12, 14, 16
B	money market instruments	1, 2, 3, 4, 7
	derivatives trading	1, 2, 3, 7, 10, 12, 14
	interest instruments	1, 4, 8, 12, 14, 16
	private consumer instruments	1, 2, 4, 7
•••	•••	•••

(c) PA with Inheritance

Role	Application	Access Right
A	money market instruments	1, 2, 3, 4
	derivatives trading	1, 2, 3, 7, 10, 12
	interest instruments	1, 4, 8, 12, 14, 16
B	money market instruments	7
	derivatives trading	14
	private consumer instruments	1, 2, 4, 7
•••	•••	•••

Case study: RBAC system for a bank



Attribute-based access control

- Fairly recent
- Define authorizations that express conditions on properties of both the resource and the subject
  - Each resource has an attribute (e.g., the subject that created it)
  - A single rule states ownership privileges for the creators
- Strength: its flexibility and expressive power
- Considerable interest in applying the model to cloud services

Types of attributes

- Subject attributes
- Object attributes
- Environment attributes

Subject attributes

- A subject is an active entity that causes information to flow among objects or changes the system state
- Attributes define the identity and characteristics of the subject
  - Name
  - Organization
  - Job title

Object attribute

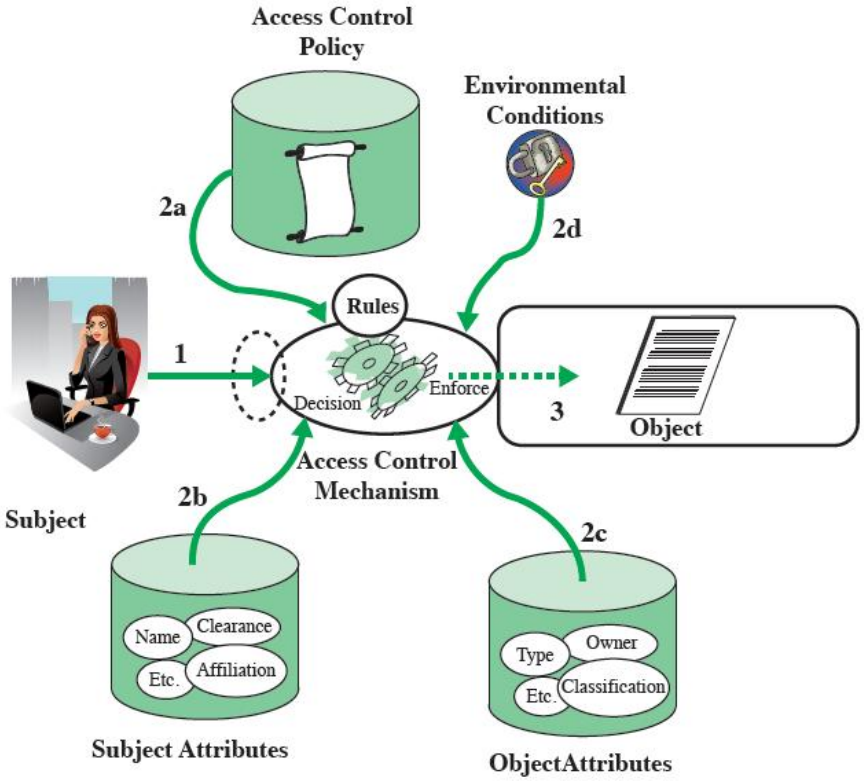
- An object (or resource) is a passive information system-related entity containing or receiving information
- Objects have attributes that can be leveraged to make access control decisions
  - Title
  - Author
  - Date

Environment attributes

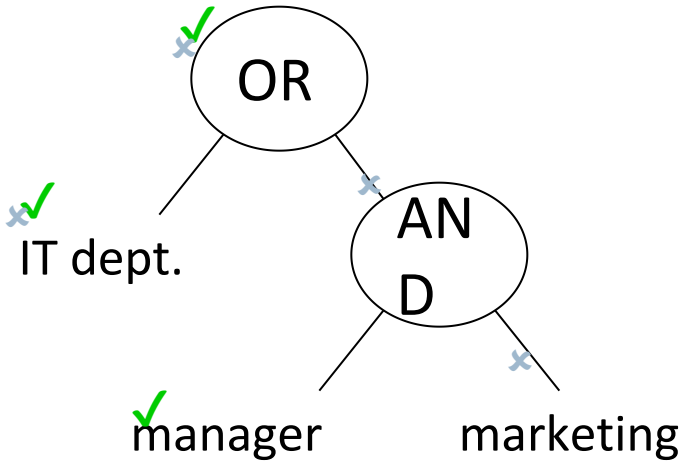
- Describe the operational, technical, and even situational environment or context in which the information access occurs
  - Current date
  - Current virus/hacker activities
  - Network security level
  - *Not associated with a resource or subject*
- These attributes have so far been largely ignored in most access control policies

Sample ABAC scenario

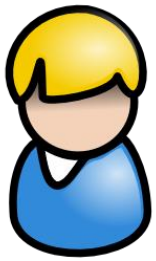
1. A subject requests access to an object
2. AC is governed by a set of rules (2a): assesses the attr of subject (2b), object (2c) and env (2d)
3. AC grants subject access to object if authorized



ABAC

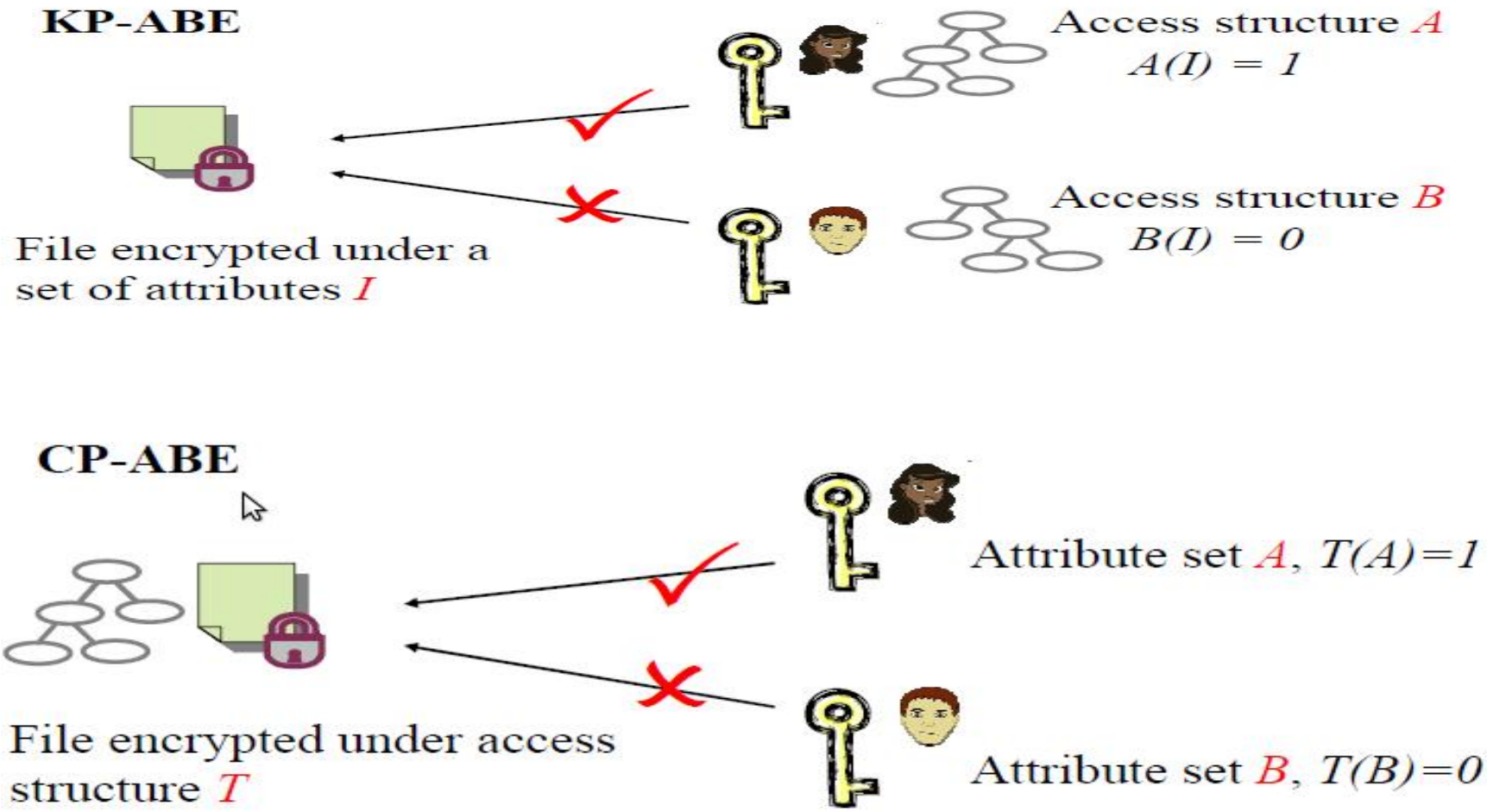


SK_{Sarah}:
"manager"
"IT dept."



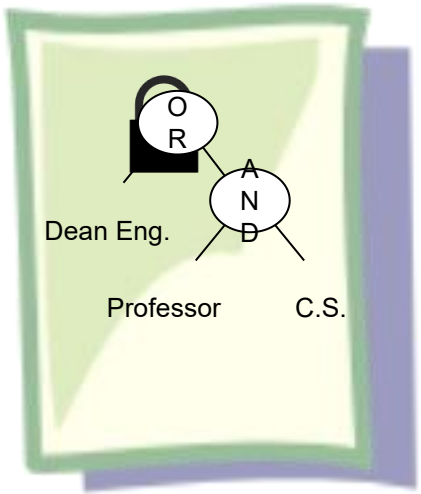
SK_{Kevin}:
"manager"
"sales"

Attribute Based Encryption : Two Types

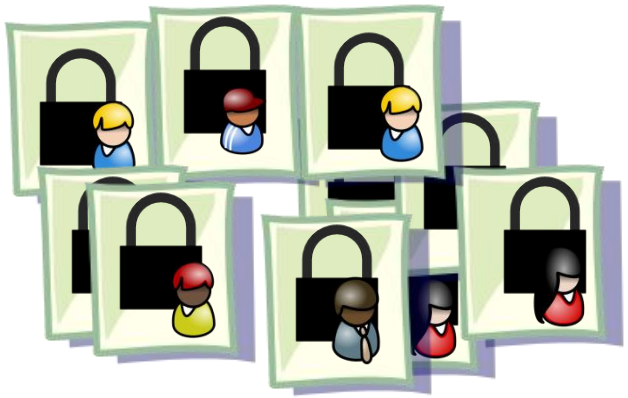


Why CP-ABE?

Efficiency:



vs.



Summary

- introduced access control principles
  - subjects, objects, access rights
- discretionary access controls
  - access matrix, access control lists (ACLs), capability tickets
  - UNIX traditional and ACL mechanisms
- role-based access control
- case study