Lecture 01: Access Control

Dr. Abdennacer KHELAIFA

Outlines

- Access Control Principles.
- Access Control Basic Elements.
- Access Control Policies.
  - Discretionary Access Control.
  - Mandatory Access Control.
  - Role-based Access Control.

Access Control Principles.

ITU-T Recommendation X.800 defines access control as follows:

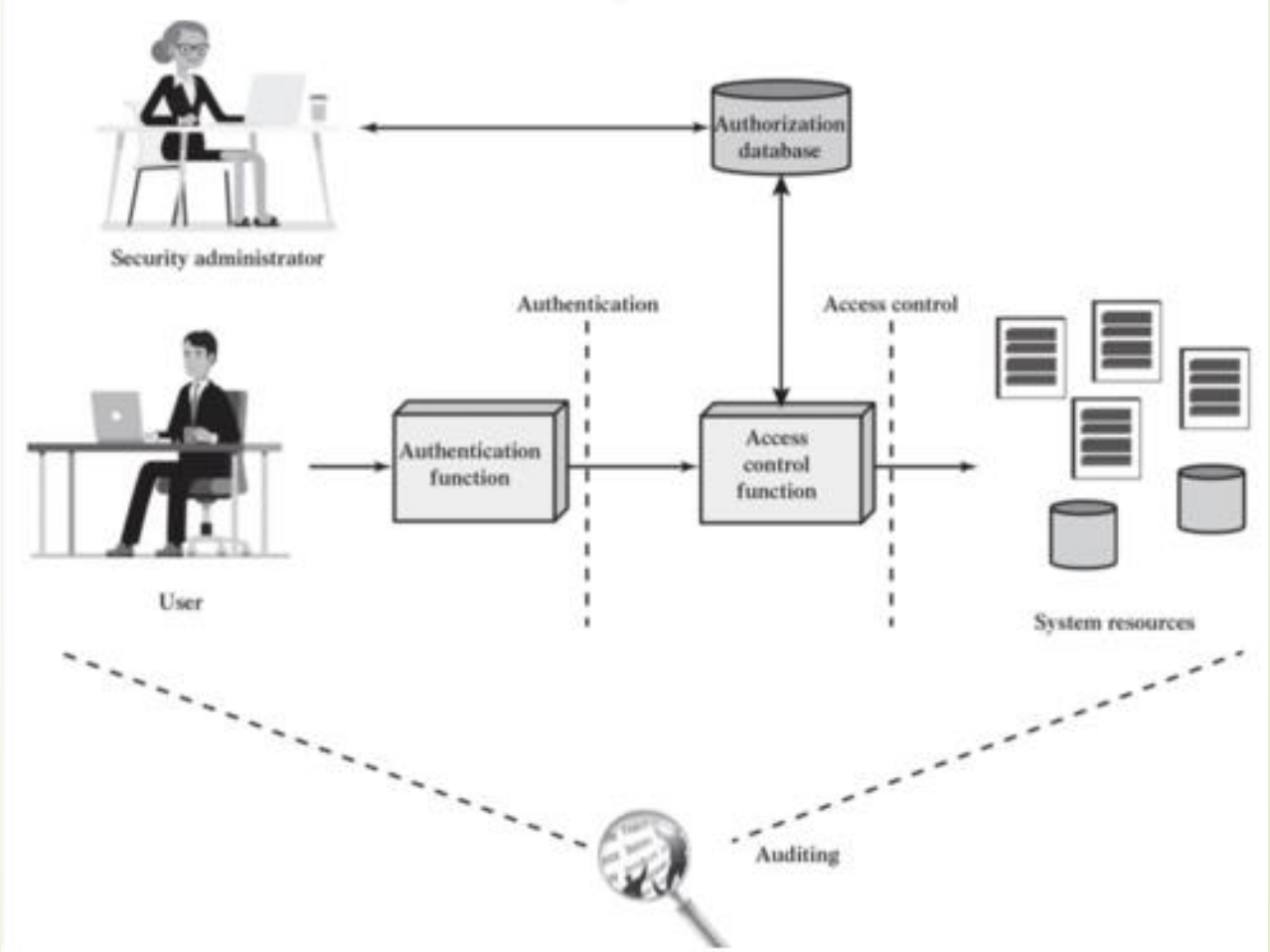
"The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner."

RFC 2828 defines computer security as:

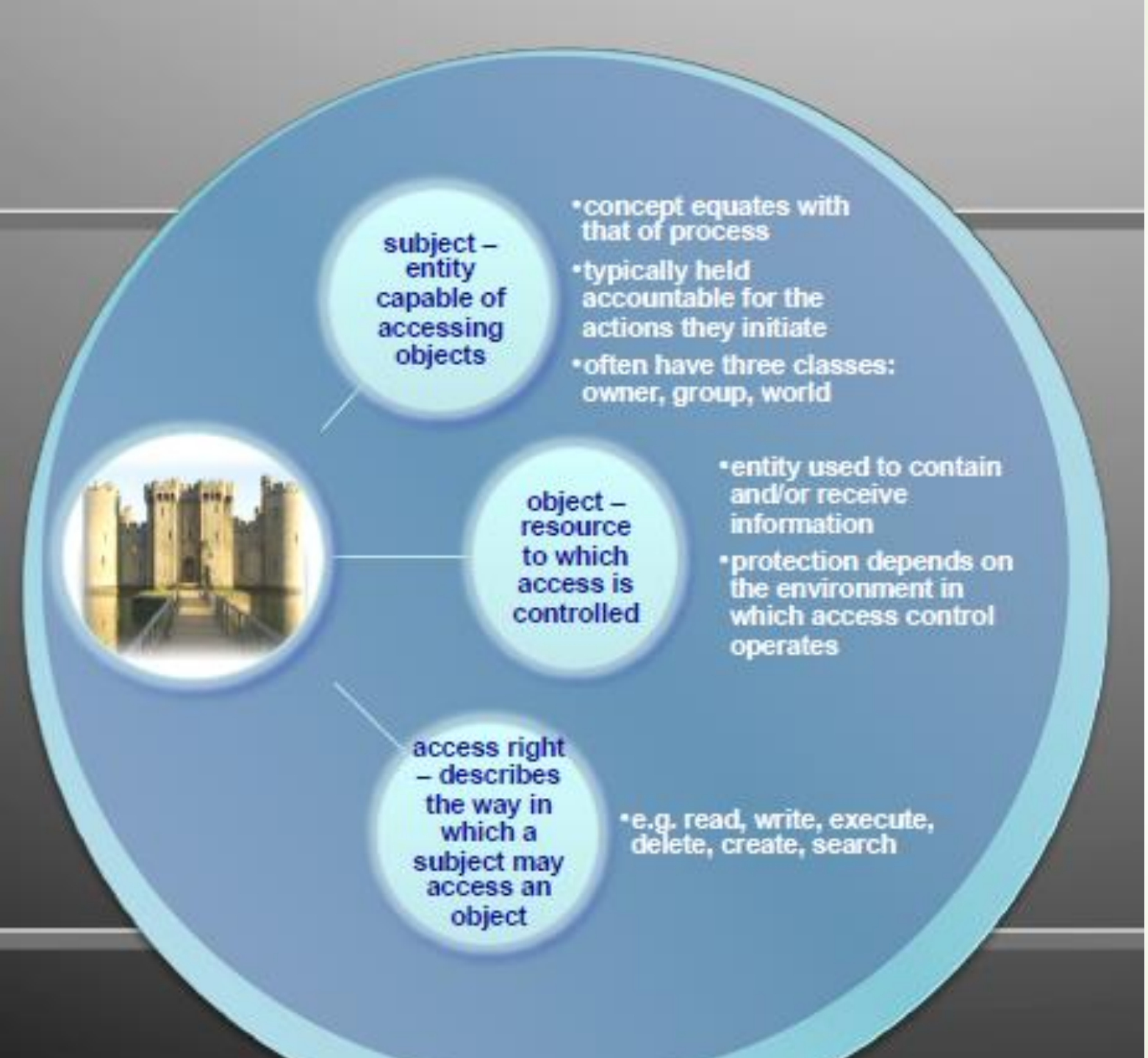
"Measures that implement and assure security services in a computer system, particularly those that assure Access control service".

Access Control Principles.

Relationship Among Access Control and Other Security Functions



Access Control Basic Elements



Access Control Policies.

- **Discretionary access controls (DAC)** – the access of objects (or subjects) can be propagated from one subject to another. Possession of an access right by a subject is sufficient to allow access to the object.
- **Mandatory access controls (MAC)** – the access of subjects to objects is based on a system-wide policies (based on security labels) that can be changed only by the administrator.
- **Role-Based Access Control (RBAC)**–can be configured as both MAC or DAC, access to objects is based on roles.

Discretionary Access Control (DAC)

Scheme in which an entity may enable another entity to access some resource

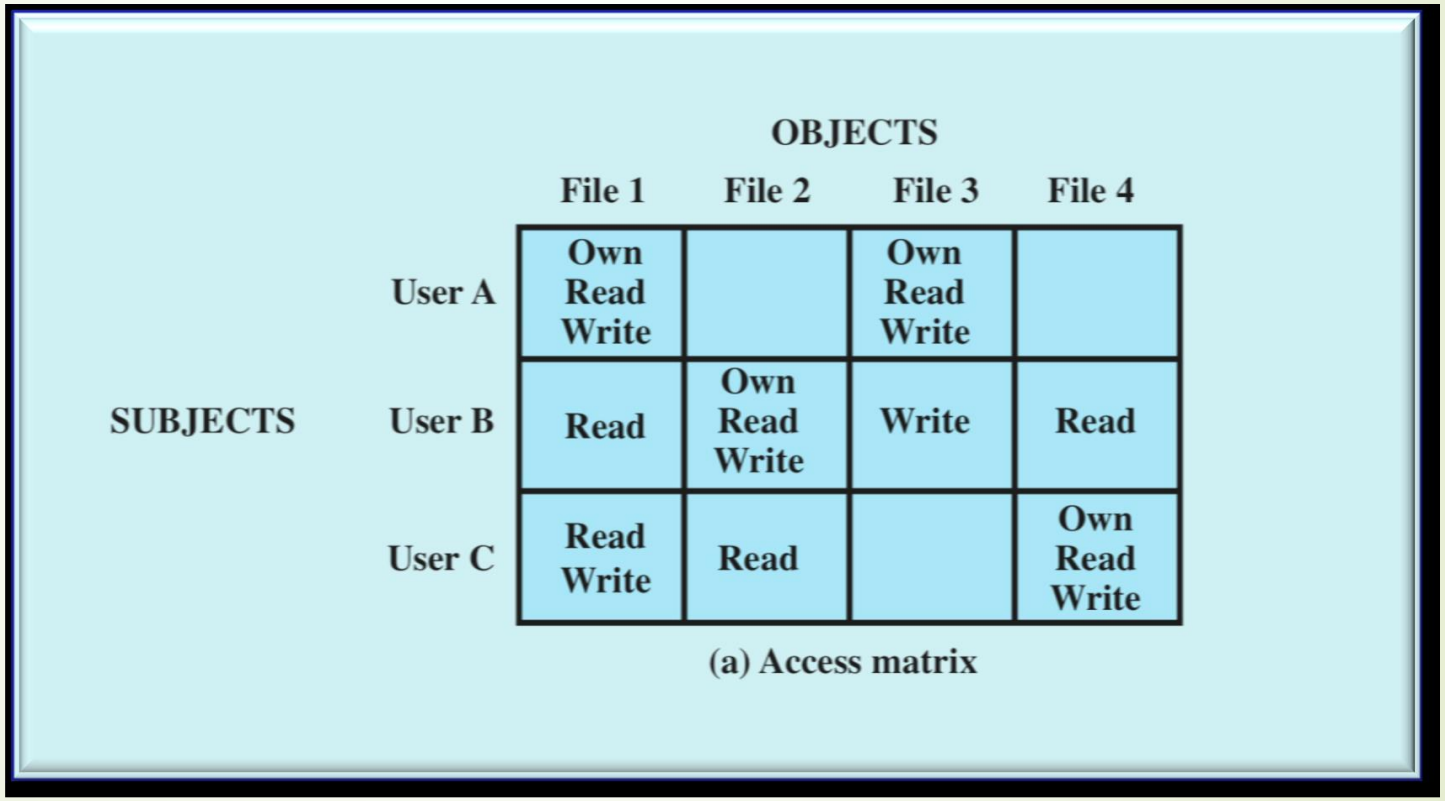
- Often provided using an access matrix

  – one dimension consists of identified subjects that may attempt data access to the resources.

  – the other dimension lists the objects that may be accessed.

- Each entry in the matrix indicates the access rights of a particular subject for a particular object

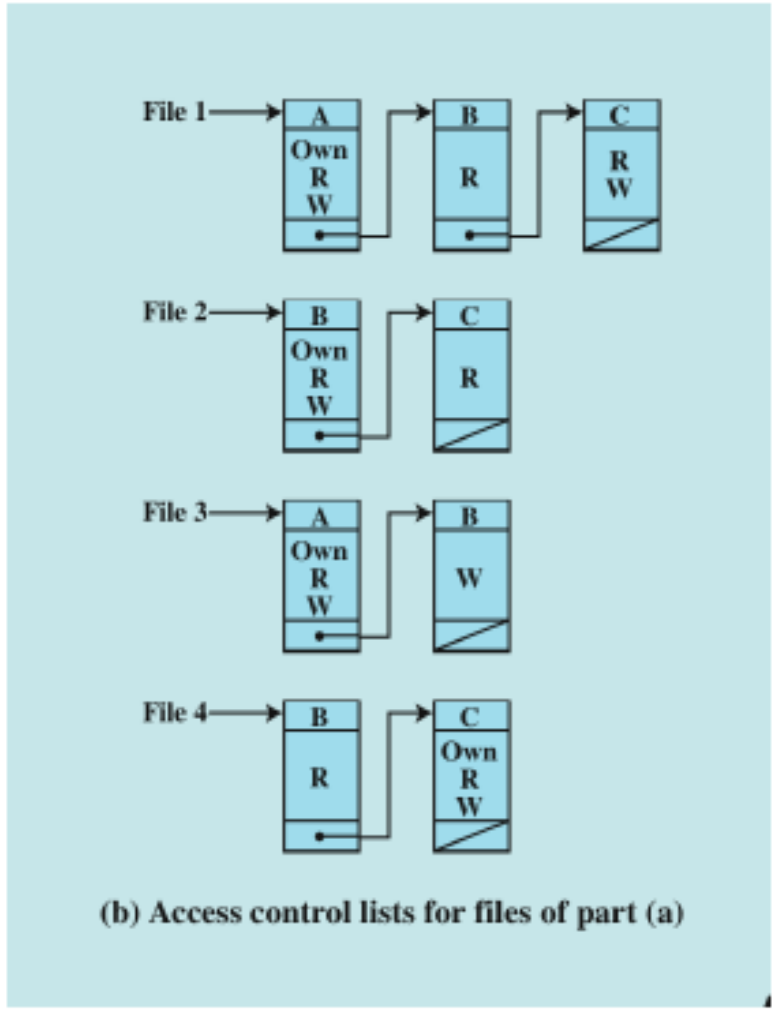
Access Matrix



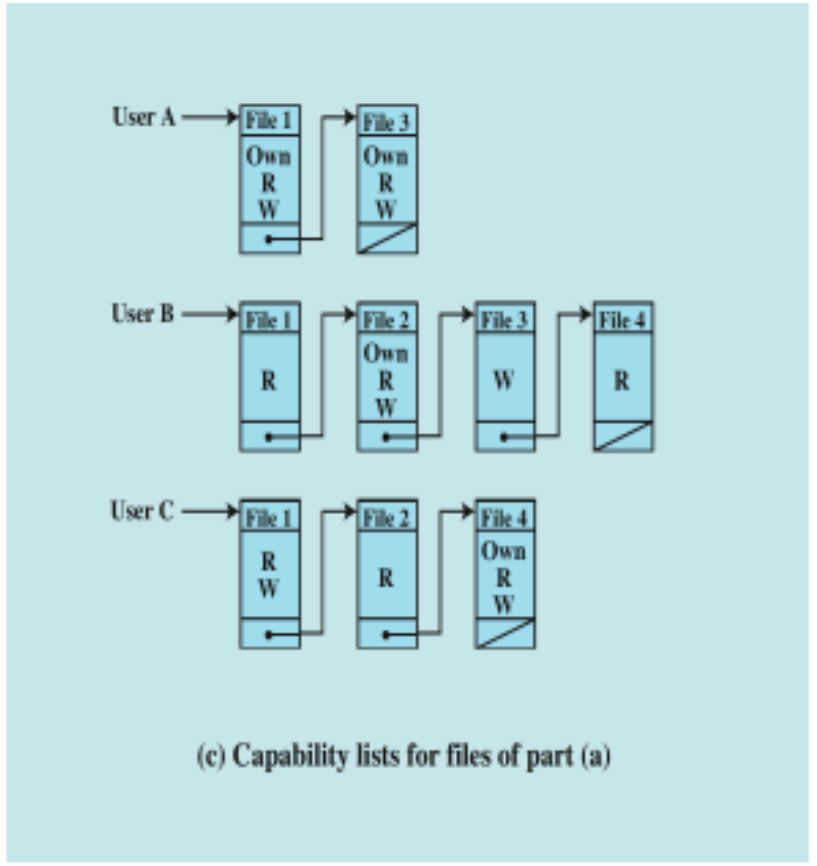
SUBJECTS	OBJECTS: File 1	File 2	File 3	File 4
User A	Own Read Write		Own Read Write	
User B	Read	Own Read Write	Write	Read
User C	Read Write	Read		Own Read Write

(a) Access matrix

Access Control Lists (ACL) & Capabilities



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



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

Authorization Table for Files

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

DAC Problems

There is a difference, though, between trusting a person and trusting a program. E.g., A gives B a program that A trusts, and since B trusts A, B trusts the program, while neither of them is aware that the program is buggy. Suppose a subject S has access to some highly secret object O. Moreover, suppose that another subject S' does not have access to O, but would like to. What can S' do to gain access? S' can write a program that does two things, the first of which is the following sequence of commands:

- Create a new object O'.
- Grant S write access to O'.
- Grant S' read access to O'.
- Copy O to O'.

The second thing the program does is to act like a video game. If the program is run by S, then S' will get access to the contents of O (now in O'). This type of program is referred to as a *Trojan horse*. DAC mechanisms are typically insufficient to protect against Trojan horse attacks.

Mandatory Access Control

The philosophy underlying these policies is that information belongs to an organization (rather than individual members of it), and it is that organization which should control the security policy. MAC policies strive to defend against **Trojan horse attacks**.

Multi-level security (MLS) documents are labeled according to:

- Their ***sensitivity* levels**:

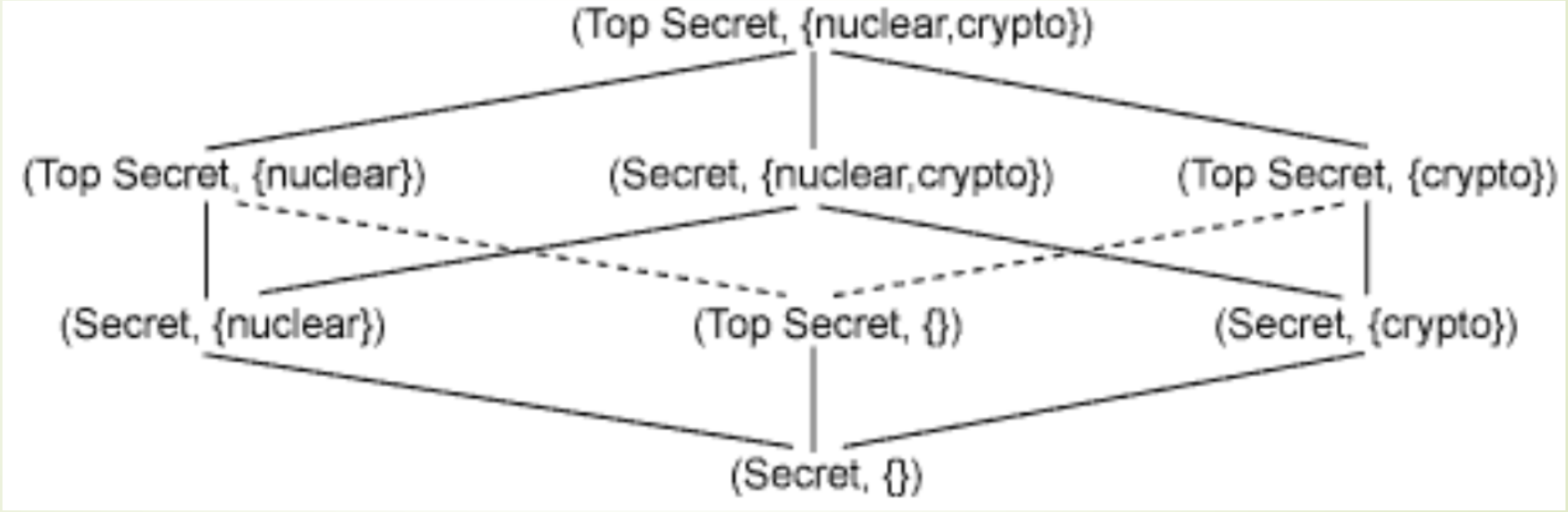
  $$\text{Unclassified} \leq \text{Confidential} \leq \text{Secret} \leq \text{Top Secret}.$$

- **Set of categories** consists of the data environment and the application area (e.g. crypto, nuclear, biological, reconnaissance, etc.).

A document label L = (S, C), S: sensitivity, C: Category (Compartment).

Given two labels L1 = (S1, C1) and L2 = (S2, C2), we write that $L1 \leq L2$---meaning that L1 is no more restrictive than L2---when $S1 \leq S2$, and $C1 \subseteq C2$.

Dominance relationship diagram



Bell-LaPadula(BLP)Model[1973]

Let L(X) denote the label of an *entity* X, where an entity is either a subject or an object. The *BLP security conditions* are:

- A subject S may read object O only if $L(O) \leq L(S)$. In other words, subjects are not allowed to "read up."
- A subject S may write object O only if $L(S) \leq L(O)$. In other words, a subject may not "write down."

BLP Problems

It is possible that the security level for an entity could be changed in mid-operation. This change could violate the information-flow constraints we wish to preserve. Consider a system with subjects s1, s2, and objects o1, o2

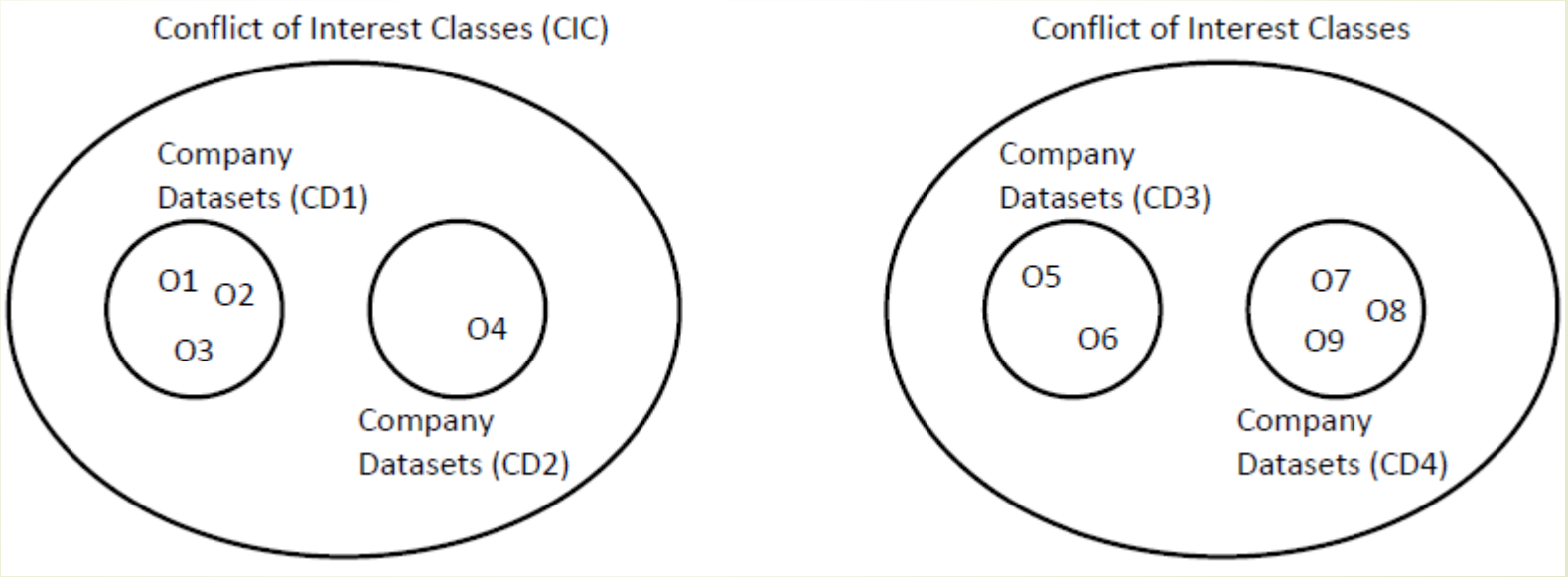
L(s1) = L(o1) = high. L(s2) = L(o2) = low

And the following execution s1 gets access to o1, reads something, releases access, then changes current level to low, gets write access to o2, writes to o2

- The model is concerned with only confidentiality, not integrity. For example, subjects can "write up". Thus, a subject that cannot read an object is permitted to make changes to that object; this is called a "blind write".
- Yet it makes little sense to trust a subject to modify the information contained in an object, if that subject is not trusted to read the information contained in the object.
- A subject S that wants to write object O is not allowed to do so if $L(O) < L(S)$. The subject must login at a lower level than his or her clearance. It is annoying for principals to be forced to decide, at the time they login, what rights they will need. But it is also a nice application of the Principle of Least Privilege.

Chinese Wall

In the Chinese Wall policy, we (as usual) have have objects, subjects, and users. However, objects are now grouped into *company datasets* (CDs). For example, an object might be a file, and a company dataset would then be all of the files related to a single company. Company datasets are themselves grouped into *conflict of interest classes* (COIs). For example, one COI might be the set of all companies in the banking industry, and another COI might be all the companies in the oil industry.



Chinese Wall

S can read O only if:

- O is in the same company dataset as some object previously read by S (i.e., O is within the wall)
- Or O belongs to a conflict of interest class within which S has not read any object (i.e., O is in the open)

S can write O only if

- S can read O by the simple security rule.
- S has never read an object O' such that CD(O) ≠ CD(O').

Chinese Wall

Suppose that S1 has previously read from CD1, and S2 has previously read from CD2. Consider the following sequence of operations, based on the figure above.

- S1 reads information from an object in CD1.
- S1 writes that information to object O6 in CD3.
- S2 reads that information from O6.

At the end of this sequence, S2 would have read information pertaining to both CD1 and CD2, which would violate the Chinese Wall policy since both CDs are in the same CIC.