Security Engineering

Chapter 3.
Access Control Systems

Universidad Carlos III de Madrid

PROGRAM

1. ACCESS CONTROL

1. Introduction
2. The access matrix
3. Access Control Policies
    1. Discretionary Access Control
    2. The problem of the Trojan horse
    3. Access control in operating systems

PROGRAM

2. MULTILEVEL AC

1. Introduction
2. Model confidentiality of Bell-LaPadula
3. Biba Integrity Model
4. Other MLS systems
5. Final thoughts on MLS systems

PROGRAM

3. MULTILATERAL AC

1. Introduction
2. Model of the Chinese wall
3. Multilateral and multilevel models

PROGRAM

4. EVALUATION CRITERIA AND CERTIFICATION

1. Introduction
2. American criteria
3. European criteria
4. international criteria

Security Engineering

Part I: ACCESS CONTROL

1. INTRODUCTION

ACCESS TO INFORMATION. Main assumptions

- The attacker has direct access to cryptographic techniques
- The attacker does not have direct access to the information (software layer between)
- Presence of authentication and authorization techniques (access control)

1. INTRODUCTION

"SAFE" COMPUTER SYSTEMS.
Construction main problem

The design of correct **security policies** (technical), their exact representation in **models** and its strict development **mechanisms**.

1. SECURITY POLICY

PHASES OF ASSURANCE in SYSTEMS DESIGN

1. Risk Analysis (Assets, threats, vulnerabilities and impacts)
2. Security Policy
3. Security Model
4. Security Mechanism

1. SECURITY POLICY. Terminology

- **Object:** Passive entity that contains, receives or deals with information.
  - It can be a computer (eg printer...), a physical device (eg memory card, ...) or logical (eg a file, a program ...)

- **Subject:** Active entity (user, program, etc.) that acts, or try to, on an object

1. TECHNICAL SECURITY POLICY

Set of guidelines that regulates the processing of data and the use of resources by an information system.
It is stated in natural language

Examples:
Authentication of users and resources
Authorization (access control) Control information flow
Audit Log
…

1. MODEL / MECHANISM

SECURITY MODEL

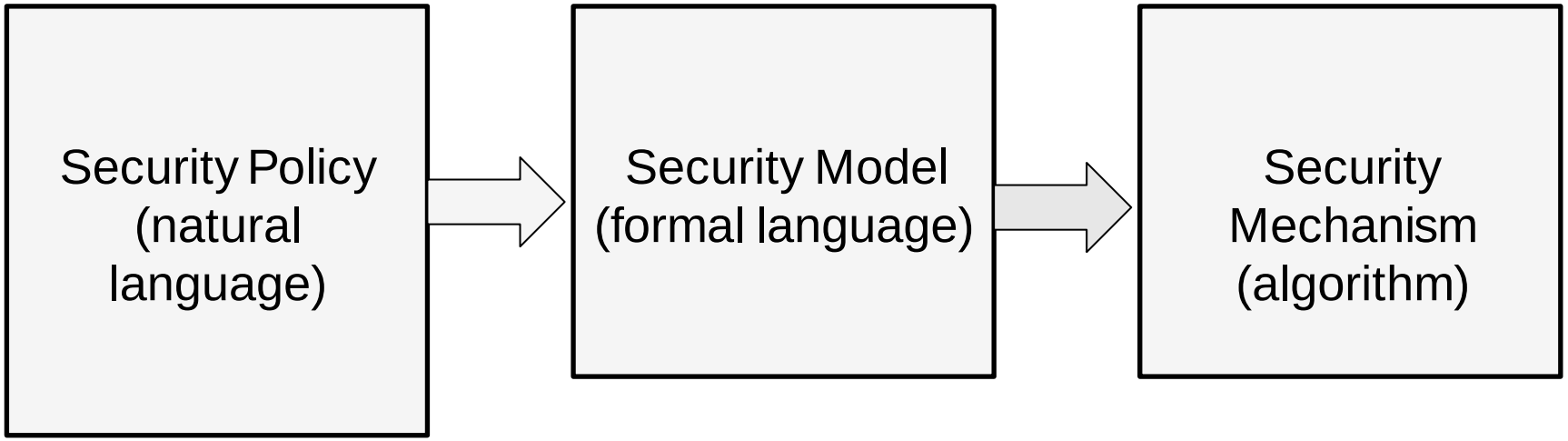
Formal theoretical expression (mathematics) of a technical security policy

SECURITY MECHANISM

Algorithm, implementable in hardware or software, which represents a security model

1. POLICY / MODEL / MECHANISM

Relationship between **policies**, **models** and security **mechanisms**



1. Access control Security Policy

Guidelines that establish under which conditions an identified (and authenticated) subject inside a system can access to a particular object

- Least privilege granularity
- Open versus closed
- Control based on risk (contextual)
- Administration of privileges
- Segregation of duties

…

1. ACCESS CONTROL MODEL (ACM)

It formally expresses (usually on matrix representation) a policy of access control.

The main ones are:

the model of Harrison, Ruzzo and Ullman and

the model of Graham and Denning

1. ACCESS CONTROL MODEL (ACM)

It is formalized by **states** (represented by a matrix, Q) and **transitions between** them (controlled by a monitor)

Q: Defined by the triple (S, O, A)

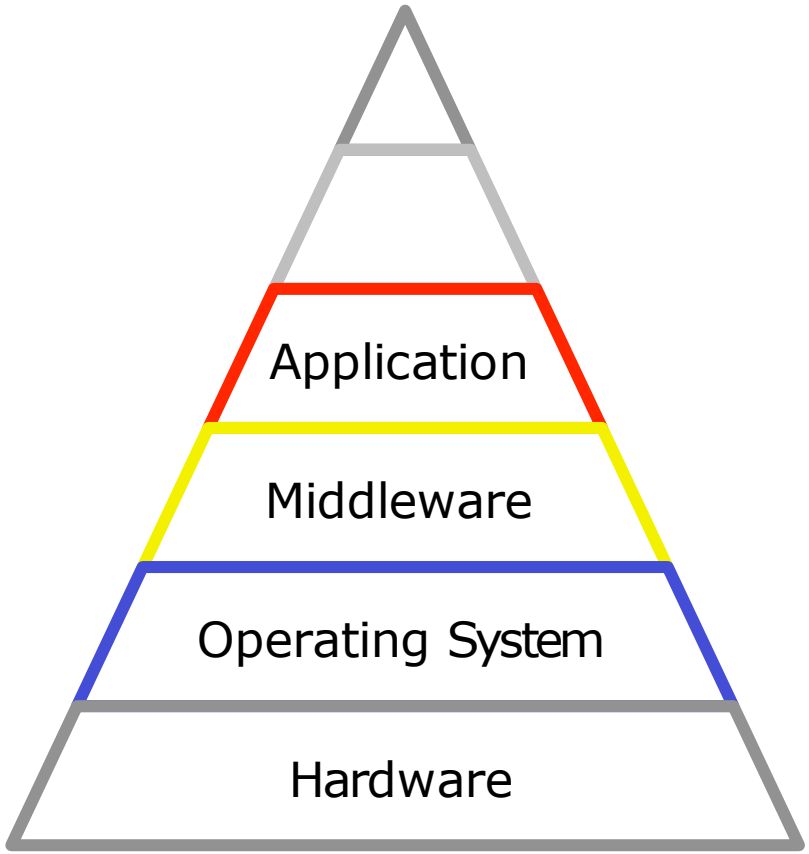
S: Set of subjects: S $\{si\}$

O: Set of objects: O $\{oj\}$

A: Access records: A $[s, o] \subseteq R$

R: Finite set of access rights

ACCESS CONTROL (CA). Implementation



- web servers
- banking systems
- applications management...

- CORBA
- DBMS
- Bank notes Systems...

- CA to ports
- CA to files
- ...

- CA to memory
- ...

1. ACCESS CONTROL MECHANISM

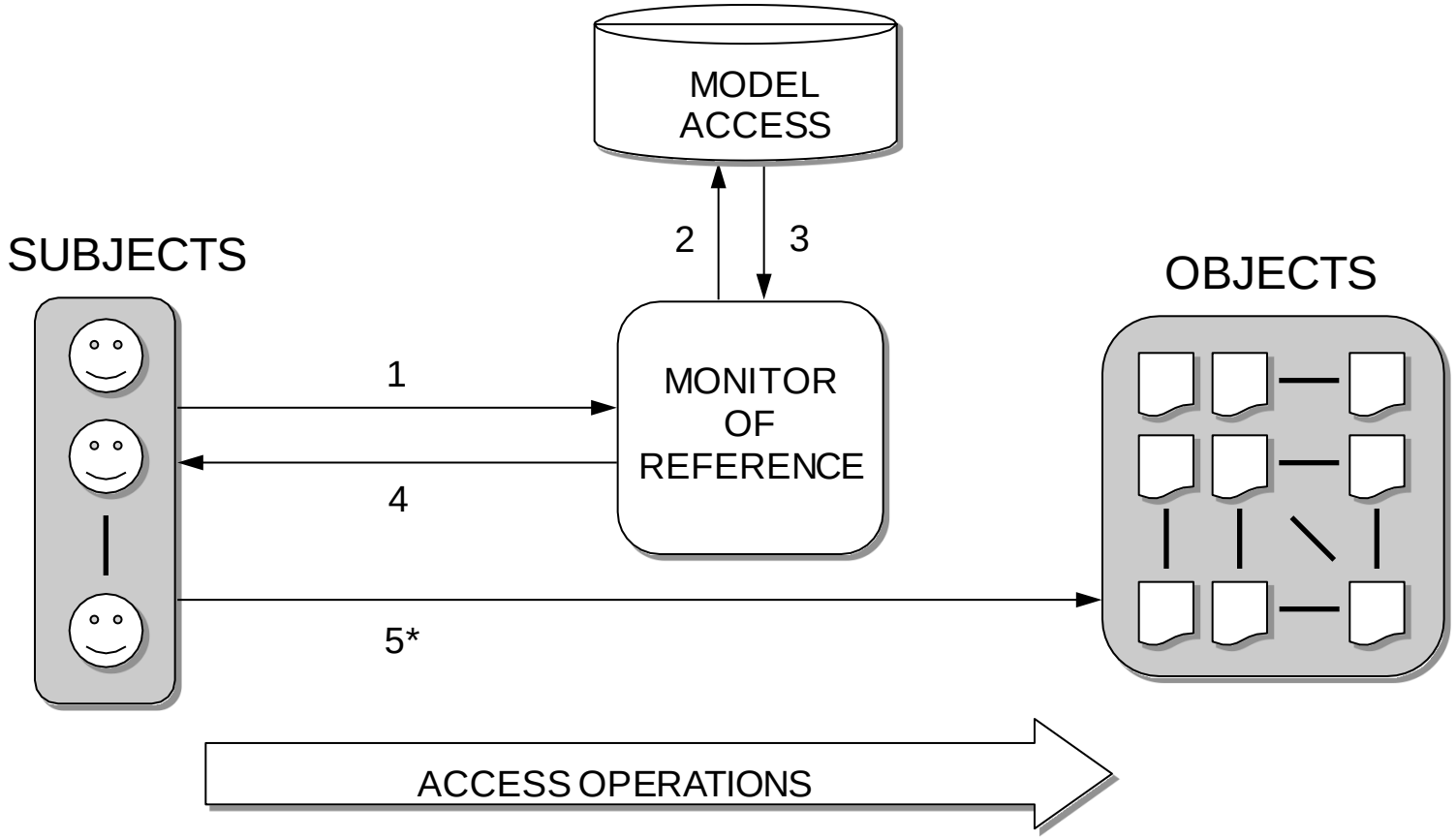
ACCESS CONTROL MECHANISM

- It controls the attempts to access objects from subjects, and
- monitors the commands that grant, transfer or revoke the rights of subjects to objects

REFERENCE MONITOR

- Abstract machine that represents an access control mechanism

1. REFERENCE MONITOR



2. THE ACCESS MATRIX (ACM)

ACM [s_i,o_j] = access operations that sj is authorized to make on oj.

	$Object_1$	$Object_2$	$Object_3$	...	$Object_M$
$User_1$	rwx	rw	rwx	...	rw
$User_2$	x	r	x	...	rw
$User_3$	x	rw	rwx	...	r
...	...	...	...	...	...
$User_N$	x	rw	x	...	w

r = read authorization

w = write authorization

x = execute authorization

2. THE ACCESS MATRIX (ACM)

PROBLEM: Low scalability

- On large systems: huge access matrix;
- Most of the cells are empty

COMMON IMPLEMENTATIONS

- In columns - Access Control Lists (ACL)
- In rows - Habilitation
- Hybrid approaches - relationships of authorization

2. ACCESS MATRIX. Access Control Lists (ACL)

ACCESS MATRIX (Eg)

	$Object_1$	$Object_2$	$Object_3$	...	$Object_M$
$User_1$	rwx	rw	rwx	...	rw
$User_2$	x	r	x	...	rw
$User_3$	x	rw	rwx	...	r
...	...	...	...	...	...
$User_N$	x	rw	x	...	w

ACL (Eg)

ACL	$Usuer_1$	$User_2$	$User_3$	...	$User_N$
$Object_2$	rw	r	rw	...	rw

2. ACCESS MATRIX. Access Control Lists (ACL)

ADVANTAGES:

- Easy to identify and revoke user permissions on an object
- Easy to revoke all permissions on an object

DISADVANTAGES:

- Expensive to determine, and to revoke if necessary, all the permissions of a user

2. ACCESS MATRIX. Habilitations

ACCESS MATRIX (Eg)

	$Object_1$	$Object_2$	$Object_3$	...	$Object_M$
$User_1$	rwx	rw	rwx	...	rw
$User_2$	x	r	x	...	rw
$User_3$	x	rw	rwx	...	r
...	...	...	...	...	...
$User_N$	x	rw	x	...	w

Habilitations (Eg)

$User_3$	$Object_1$	$Object_2$	$Object_3$	...	$Object_N$
	x	rw	rwx	...	r

2. ACCESS MATRIX. Habilitations

COMPARED TO ACL

- Dual advantages and disadvantages
- Less adopted
- More appropriate in distributed systems:

  →→do not require multiple authentications SSO (Single Sign-On)

2. MATRIX OF ACCESS. Authorization relations

Representation of tuples (U_i, O_j, a_ij) not empty

	$Objeto_1$	$Objeto_2$	$Objeto_3$	...	$Objeto_M$
$Usuario_1$	rwx	rw		...	
$Usuario_2$	x		r	...	
$Usuario_3$		rw		...	
...	...	...	...	...	...
$Usuario_N$	r		rw	...	w

Sujeto	**Objeto**	**Acceso**
$Usuario_1$	$Objeto_1$	rwx
$Usuario_1$	$Objeto_2$	rw
$Usuario_2$	$Objeto_1$	x
$Usuario_2$	$Objeto_3$	r
$Usuario_3$	$Objeto_2$	rw
$Usuario_N$	$Objeto_1$	r
$Usuario_N$	$Objeto_3$	rw
...	...	...
$Usuario_N$	$Objeto_M$	w

- Share the advantages of ACL and habilitations
- In relational DBMS - very efficient queries

2. ACCESS MATRIX. Role Based Access Control (RBAC)

Facilitates the design and management

Is adaptable to the structure and dynamics of the organization

- **Group**: set of users.
- **Role**: set of permissions for one or more subjects over a period of time.
  - Examples: soldier / guard on duty, engineer, project manager, ...
  - Relationship many (subjects) -to-many (permissions)

2. MATRIX OF ACCESS. Role Based Access Control (RBAC)

FEATURES

- The political and the security model incorporate groups and roles, also subjects
- Many OS and applications provide support for groups
- Not much available at commercial products

3. ACCESS CONTROL POLICIES

- **DISCRETIONARY POLICY**
  - The owner of an object grants or not its access, discretionally, to other subjects

- **NON-DISCRETIONARY POLICY (MANDATORY)**
  - Subjects and objects are compartmentalized.
  - Subjects can only access to objects if they both belong to the same compartment.
  - Subjects S and objects O are structured in levels to limit the access of an S to an O

3.1. DISCRETIONARY ACCESS CONTROL (DAC)

- **Objective:**
  - Access to information controlled by the creator (owner) of the object
- **Model:**
  - Each object has an **owner** subject
  - Only this subject assigns permissions to other subjects
- **Extensions:**
  - It is possible to **transfer the ownership** of one object to another subject (Amazon S3)
  - Delegation of the right to assign permissions (Amazon S3)

3.1. DISCRETIONARY ACCESS CONTROL (DAC)

DAC incorporates "p" (owner) to access properties

	$Object_1$	$Object_2$	$Object_3$	...	$Object_M$
$User_1$	prwx	rw	prwx	...	rw
$User_2$	x	prwx	X	...	rw
$User_3$		rw	rwx	...	pwr
...	...	...	...	...	...
$User_N$	x	rw	x	...	w

3.2. THE PROBLEM OF THE TROJAN HORSE

Eg: S $\{s_i\}$/ i=1,2,3; O $\{o_j\}$/ j=1,2.

- **s_1** owns the object **o_1**
- **s_1** grants read permission on **o_1** to **s_2**, but not to **s_3**

	o_1
s_1	prw
s_2	r
s_3	–

s_3 lee **o_1**

	o_1	o_2
s_1	prw	–
s_2	r	pr
s_3	–	r

Sequence:

- **s_2** reads **o_1**
- **s_2** creates **o_2** and copies **o_1** on it
- **s_2** grants read permission of **o_2** to **s_3**, but not to **s_1**

3.2. THE PROBLEM OF THE TROJAN HORSE.

Discretionary policies and ACM are insufficient

Solution: Control of **information flow**

- Imposes other restrictions
- Still not much integrated in commercial systems
- Types:
  - multilevel security
  - multilateral security

3.3. ACCESS CONTROL. Operating Systems

- Subjects are identified by authentication mechanisms (passwords, Kerberos, smart cards, ...)
- In addition to the files, are considered "special" objects (devices, network ports, etc.)
- Common Operations: read, write, execute, owner change, delete, ...

Security Engineering

Part II: Multilevel security

PROGRAM

2. MULTINIVEL SECURITY

1. Introduction
2. Confidentiality Model of Bell-LaPadula
3. Biba Integrity Model
4. Other MLS systems
5. Final thoughts on MLS systems

MULTILEVEL SYSTEMS (MLS)

OS and DDBB containing objects with different levels of sensitivity (confidentiality and integrity) and subjects recorded with different ratings

Access to objects is granted or not, after comparing the **sensitivity level of the objects** with the **ratings of the subjects**

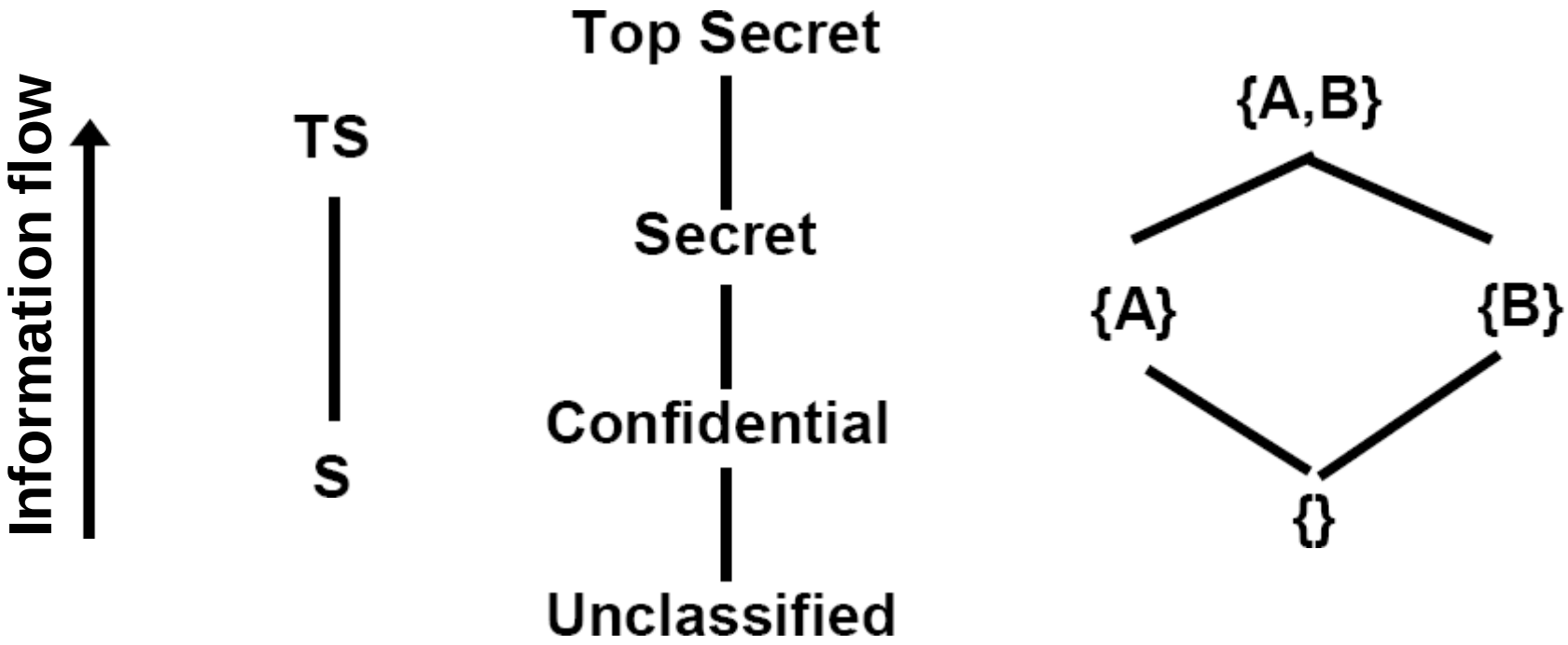
MULTILEVEL SYSTEMS (MLS)

- **Objective:** Strict control of information flow
- **Source:** Centers for military research or funded by the US DoD
- **Products: Military** OS and DDBB. Some of them were commercialized afterwards in Civil markets

CLASSIFICATION AND SECURITY CLEARANCE

- **Source:** NATO Governments
- **Purpose:**
  - Labelling of items: sensitivity of the information they contain or could deal with
  - Labelling of subjects: security clearance
  - Tags (example): Unclassified, Confidential, Secret, Top Secret
- **Hierarchy** of subjects and objects

EXAMPLES OF HIERARCHY



2. FORMAL MODELS OF CONFIDENTIALITY

- MAIN APPLICATIONS

  Military, diplomatic and intelligence sectors

- EXAMPLES

  Bell-LaPadula Model

2. BELL-LAPADULA MODEL (BLP)

Controls the flow of information to preserve the **confidentiality** of data, establishing precise rules for access control.

To determine if a subject can access (for reading or writing) an object,

- the level of **qualification (or authority) of the subject** is compared with the level of **confidentiality of the object**

2. BELL-LAPADULA MODEL (BLP)

- **Authors:** David Bell and Len LaPadula, 1973.
- **Motivation:** Complex SS OO working on timeshare
- **Objective:** To formalize the model of control of the information flow (confidentiality)
- **Emphasis:** Notice
- **Security Policy:** Mandatory Access Control

2. BELL-LAPADULA MODEL. Formalization

$\{o_1, o_2, \ldots, o_n\}$ $\forall i / 1 \leq i \leq n$, o_i: Object

$\{s_1, s_2, \ldots, s_m\}$ $\forall i / 1 \leq j \leq m$, s_j: Subject

Confidentiality level of o_i: $C(o_i)$

Level of qualification of s_j: $A(s_j)$

$\forall$ $A(s_j)$ y $C(o_i)$ $\exists$ a relation order ($\leq$) / $C(o_i) \leq A(s_j) \Rightarrow$ confidentiality of o_i is less than the qualification of s_j

2. BELL-LAPADULA MODEL. Formalization

1.- **Simple Security** Property:

s_i can read $\mathbf{o_i} \Leftrightarrow C(\boldsymbol{o_i}) \leq A(\boldsymbol{s_j})$

2.- **Star** property (property *):

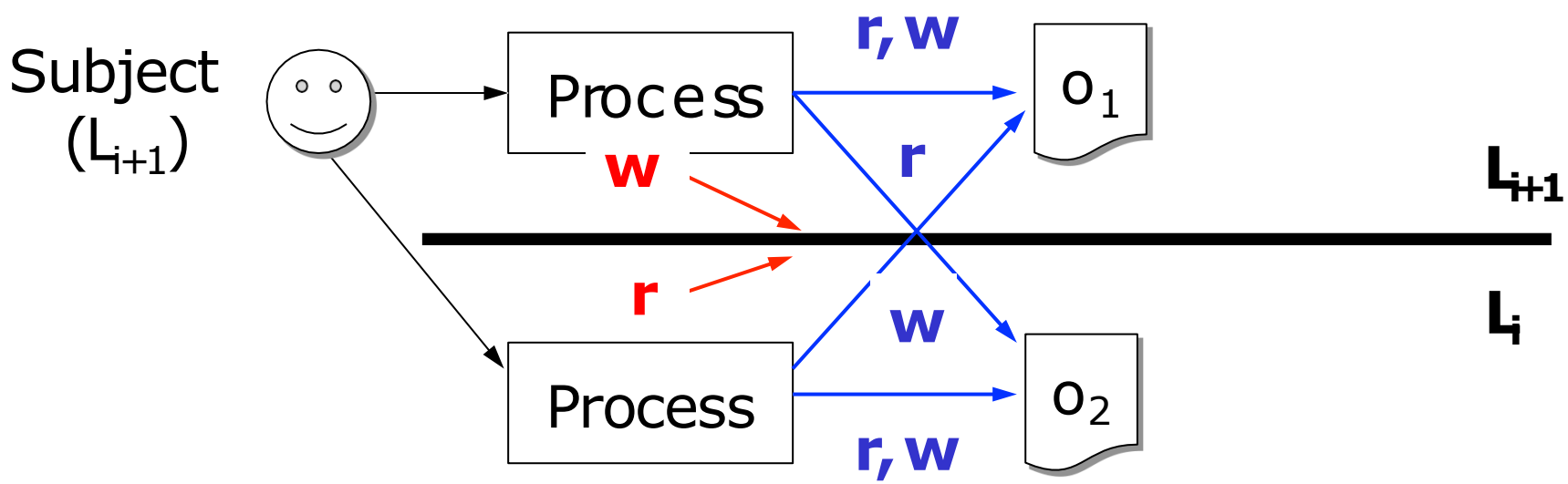
If $\boldsymbol{s_i}$ can read $\mathbf{o_i}$ and can write on $\boldsymbol{o_j} \Rightarrow C(\boldsymbol{o_i}) \leq C(\boldsymbol{o_j})$

3.- T**ranquility** property:

The levels of confidentiality and ability cannot be changed

2. BELL-LAPADULA MODEL. Formalization

1. **Simple Security Property:** No process can read information from a higher level
2. **Star Property** : No process can write information on a lower level



2. BELL-LAPADULA MODEL. Reviews

- **Innovation:** Star property (solves the problem of the Trojan horse)
- **Criticism:** focused only on confidentiality
- **Assumption:** unfair users
- **Reliable subject:** one subject authorized to declassify objects

3. FORMAL MODELS OF INTEGRITY

- Applicable in commercial systems
- Integrable with BLP in a single hierarchy
- Integrity models:
  - Biba
  - Goguen-Meseguer
  - Sutherland
  - Clark-Wilson
  - Brewer-Nash

3. BIBA MODEL

Controls the flow of information to preserve data **integrity**, establishing precise rules of access control

To determine if a subject could access (for read or write) to an object,

- the level of **qualification** (or **authority**) of the subject is compared with the level of **integrity** of the object

3. BIBA MODEL

- **Author:** Ken Biba, 1977.
- **Objective:** To formalize the control model of information flow (integrity)
- **Dual** to BLP
- **Emphasis:** Integrity
- **Security Policy:** Mandatory Access Control

3. BIBA MODEL. Formalization

$\{o_1, o_2, \ldots, o_n\}$ $\forall i / 1 \leq i \leq n$, o_i: Object

$\{s_1, s_2, \ldots, s_m\}$ $\forall i / 1 \leq j \leq m$, s_j: Subject

Integrity Level of o_i: $I(o_i)$

Level of ability of s_i: $A(s_j)$

$\forall$ $A(s_j)$ and $I(o_i)$ $\exists$ an order relation $(\leq)$ / $I(o_i) \leq A(s_j)$

$\Rightarrow$ integrity of o_i is less than the ability of s_j

3. BIBA MODEL. Formalization

Simple Security Property

s_i can write $\boldsymbol{o_i} \Leftrightarrow I(\boldsymbol{o_i}) \leq A(\boldsymbol{s_j})$

Star Property

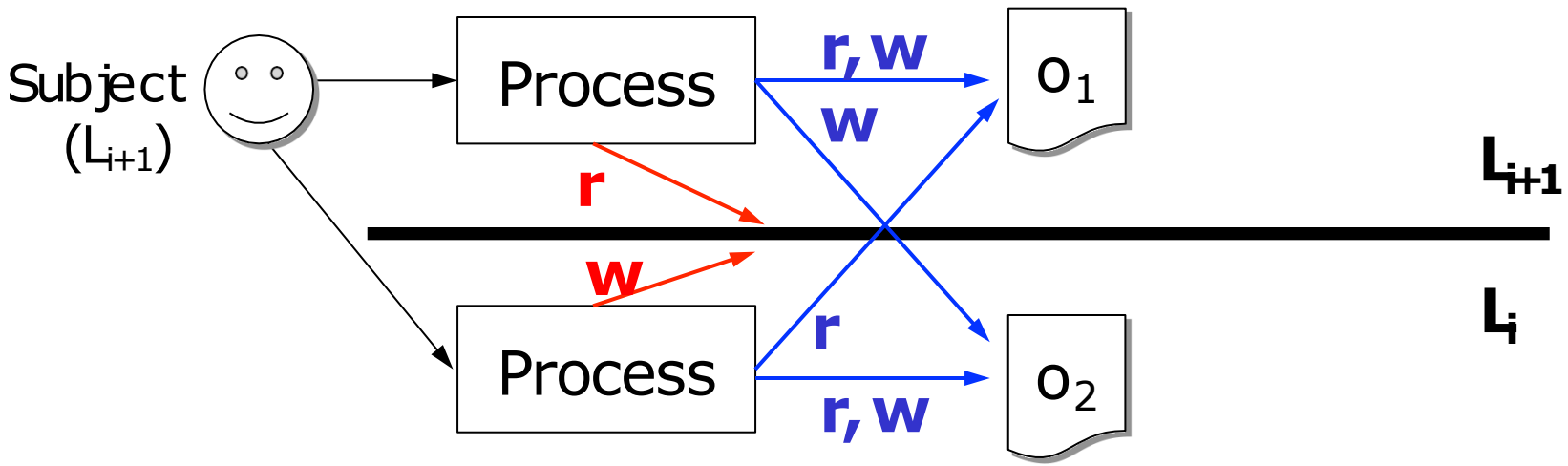
If s can write $\boldsymbol{o_i}$ and read $\boldsymbol{o_j} \Rightarrow I(\boldsymbol{o_i}) \leq I(\boldsymbol{o_j})$

Tranquility property :

The levels of integrity and ability cannot be changed

3. BIBA MODEL. Formalization

1. **Simple Security** Property: No process can read information from a lower level
2. **Property star** (* -property): No process can write information on a higher level



3. BIBA MODEL

Higher level of label $\Rightarrow$ **higher confidence** (in terms of integrity) in the subject / object

- Higher level on the label of a program $\Rightarrow$ more confidence in their correct implementation
- Higher level on the label of a data file $\Rightarrow$ greater confidence in its accuracy or reliability

CLARK-WILSON MODEL

Formalizes the integrity requirements of a security policy.

Serves **integrity**, so it is better suited to business needs than the BLP model

5. MULTILEVEL MODELS. Problems

Very **expensive** (small market)

Complex administration

- Multitude of objects and subjects
- Frequent changes of level classification
- Laborious declassification

Applications need to be **rewritten**

Prevent unauthorized access, but also authorized