CS306: Introduction to IT Security

Fall 2018

Lecture 8: Access control

Instructor: **Nikos Triandopoulos**

October 30, 2018

STEVENS
INSTITUTE *of* TECHNOLOGY
THE INNOVATION UNIVERSITY®

Last lecture

- Revision
- User authentication

Today

- Access control

8.0 Announcements

CS306: Tentative Syllabus

Week	Date	Topics	Reading	Assignment
1	Aug 28	Introduction	Ch. 1	-
2	Sep 4	Symmetric encryption	Ch. 2 & 12	Lab 1
3	Sep 11	Symmetric encryption II	Ch. 2 & 12	Lab 2, HW 1
4	Sep 18	Message authentication	Ch. 2 & 12	Lab 3, HW 1
5	Sep 25	Hash functions	Ch. 2 & 12	Lab 4
6	Oct 2	Public-key cryptography	Ch. 2 & 12	Lab 5
–	Oct 9	No class (Monday schedule)		Help session
7	Oct 16	Midterm (closed books)	All materials covered	No labs

CS306: Tentative Syllabus (continued)

Week	Date	Topics	Reading	Assignment
8	Oct 23	User authentication	Ch. 2	No labs
9	**Oct 30**	**Access Control**	**Ch. 2**	**Lab 6**
10	Nov 6	Software, Web & Network security		
11	Nov 13	Database & cloud security		
12	Nov 20	Privacy		
13	Nov 27	Economics		
14	Dec 4	Legal & ethical issues		
15	Dec 11 (or later)	**Final** (closed books)	All materials covered*	

* w/ focus on what covered after midterm

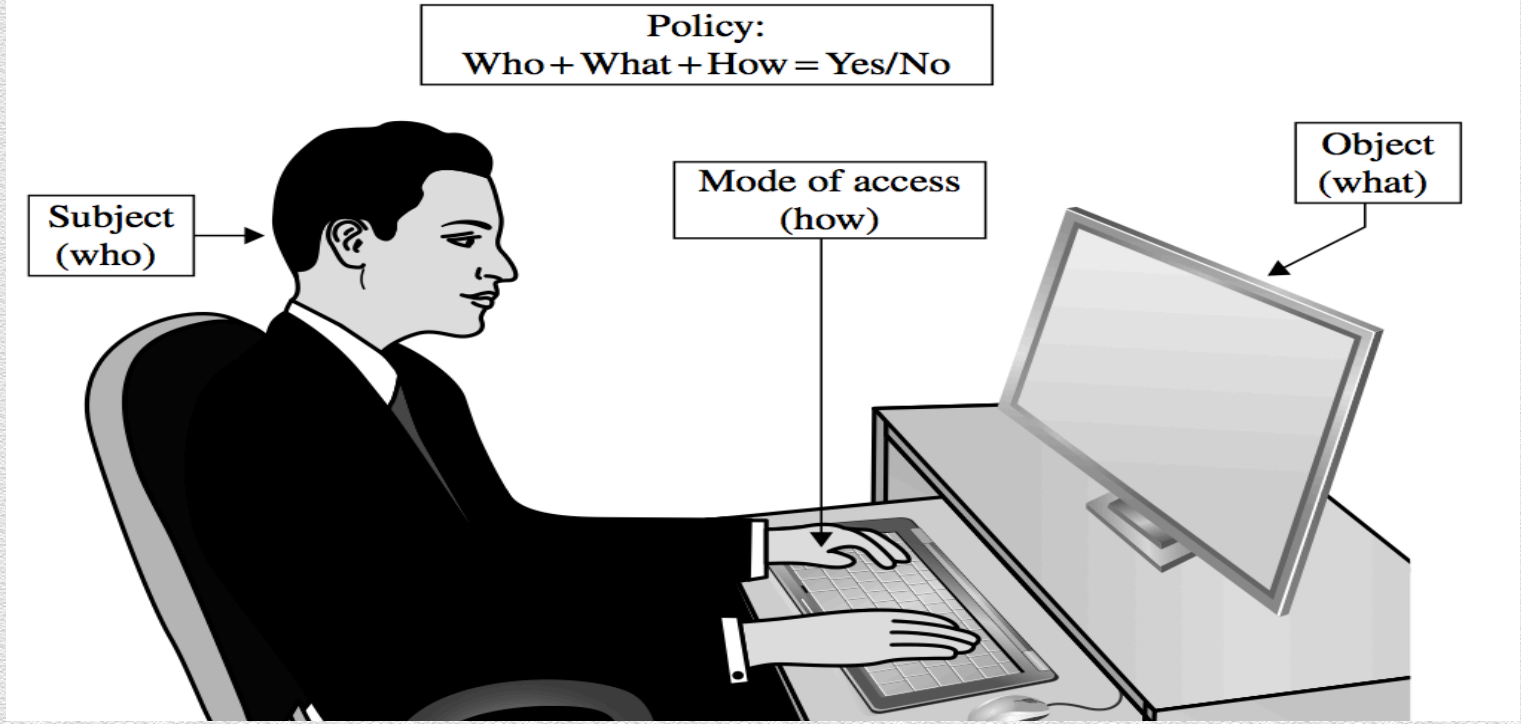
CS306: Course outcomes

- **Terms**
  - describe common security terms and concepts
- **Cryptography**
  - state basics/fundamentals about secret and public key cryptography concepts
- **Attack & Defense**
  - acquire basic understanding for attack techniques and defense mechanisms
- **Impact**
  - acquire an understanding for the broader impact of security and its integral connection to other fields in computer science (such as software engineering, databases, operating systems) as well as other disciplines including STEM, economics, and law
- Ethics
  - acquire an understanding for ethical issues in cyber-security

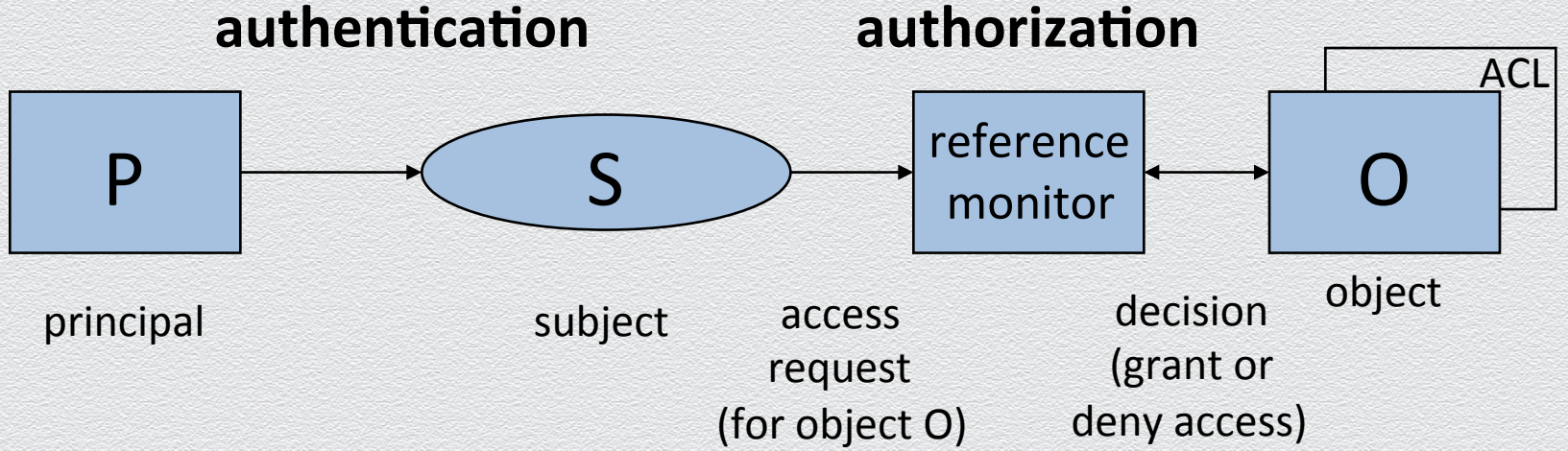
Questions?

8.1 Access control

Access control (AC)



General structure of access control mechanism



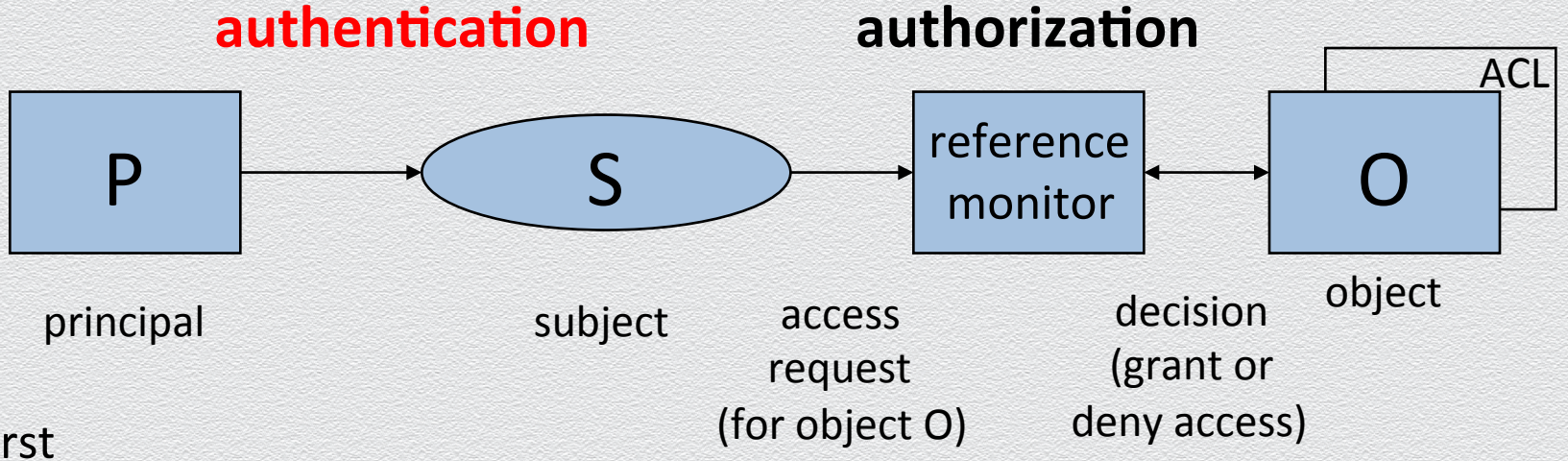
Basic terminology

- Subject/Principal
  - active entity – user or process
- Object
  - passive entity – file or resource
- Access operations
  - vary from basic memory access (read, write) to method calls in object-oriented systems
  - comparable systems may use different access operations or attach different meanings to operations which appear to be the same

Access operation

- Access right
  - right to perform an (access) operation
- Permission
  - typically a synonym for access right
- Privilege
  - typically a set of access rights given directly to roles like administrator, operator, ...

Authentication



First

- reference monitor verifies the identity of the principal making the request
  - a user identity is one example for a principal
  - cf. authentication Vs. identification

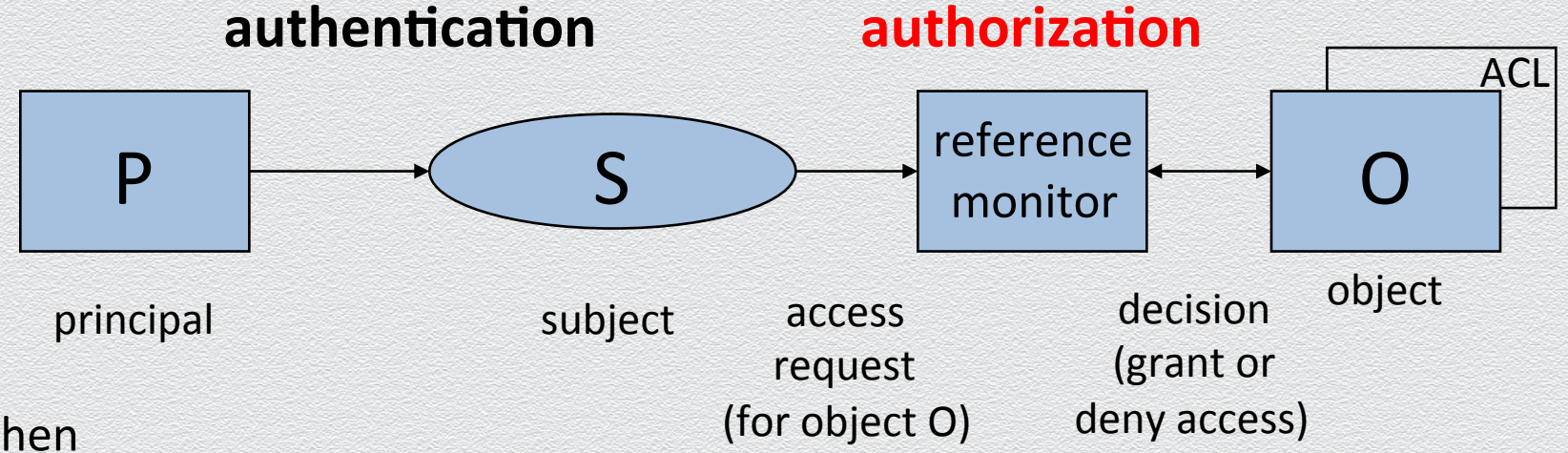
Authentication

- user enters username and password
- if the values entered are correct, the user is “authenticated”
- we could say: “The machine now runs on behalf of the user”
  - this might be intuitive, but it is imprecise
- log on creates a process that “runs with access rights” assigned to the user
  - the process runs under the user identity of the user who has logged on

Users & user identities

- requests to reference monitor do not come directly from a user or a user identity, but from a process
- in the language of access control, the process “speaks for” the user (identity)
- the active entity making a request within the system is called the subject
- must distinguish between three concepts
  - user: person
  - principal: identity (e.g., user name) used in the system, possibly associated with a user
  - process (subject): process running under a given user identity

Authorization



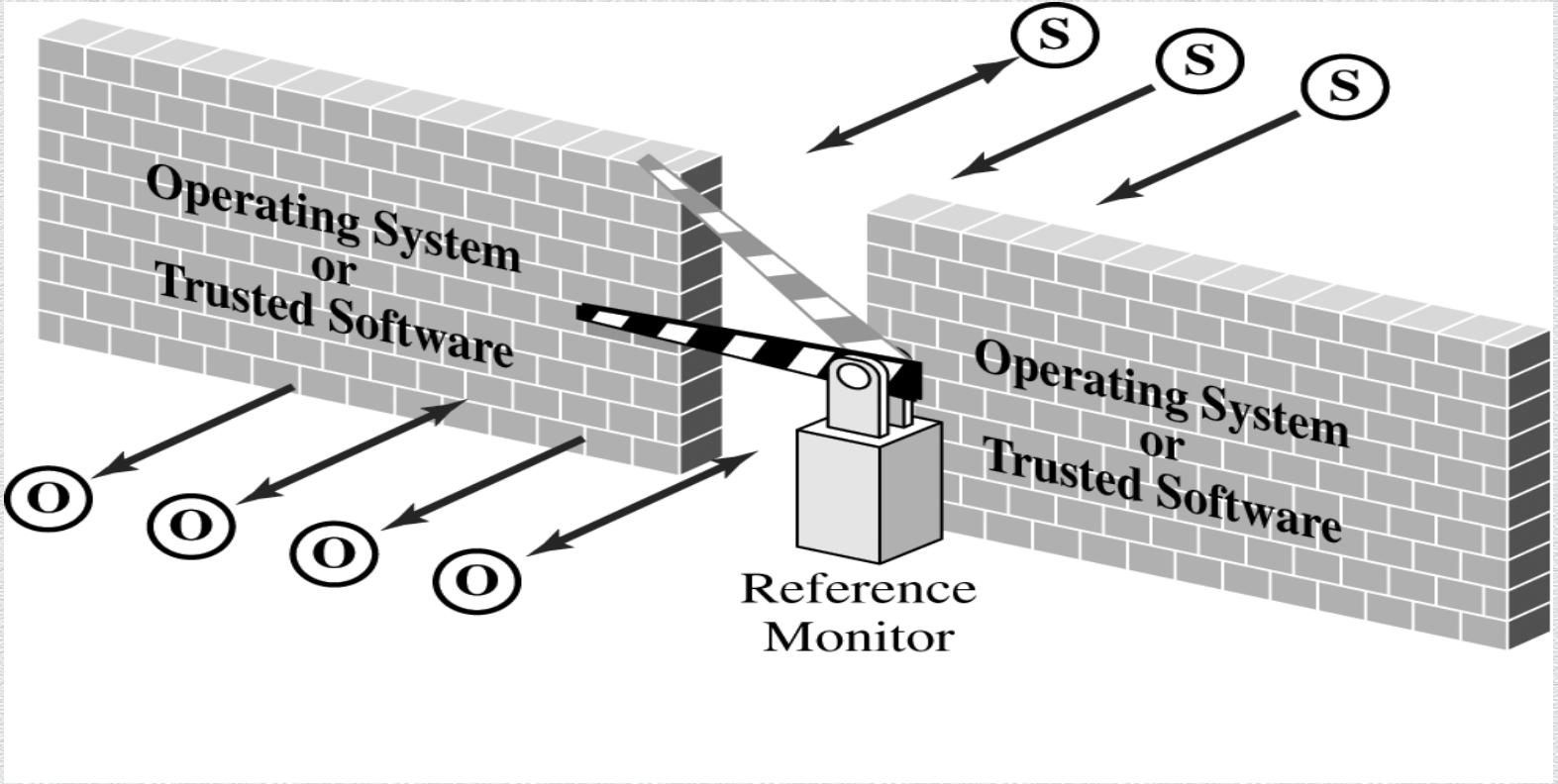
Then

- reference monitor decides whether access is granted or denied
- has to find and evaluate the security policy relevant for the given request
- "easy" in centralized systems; in distributed systems,
  - how to find all relevant policies? how to make decisions if policies may be missing?

Principals & subjects

- a principal is an entity that can be granted access to objects or can make statements affecting access control decisions
  - example: user ID
- subjects operate on behalf of (human users we call) principals
- access is based on the principal's name bound to the subject in some unforgeable manner at authentication time
  - example: process (running under a user ID)

Reference monitor



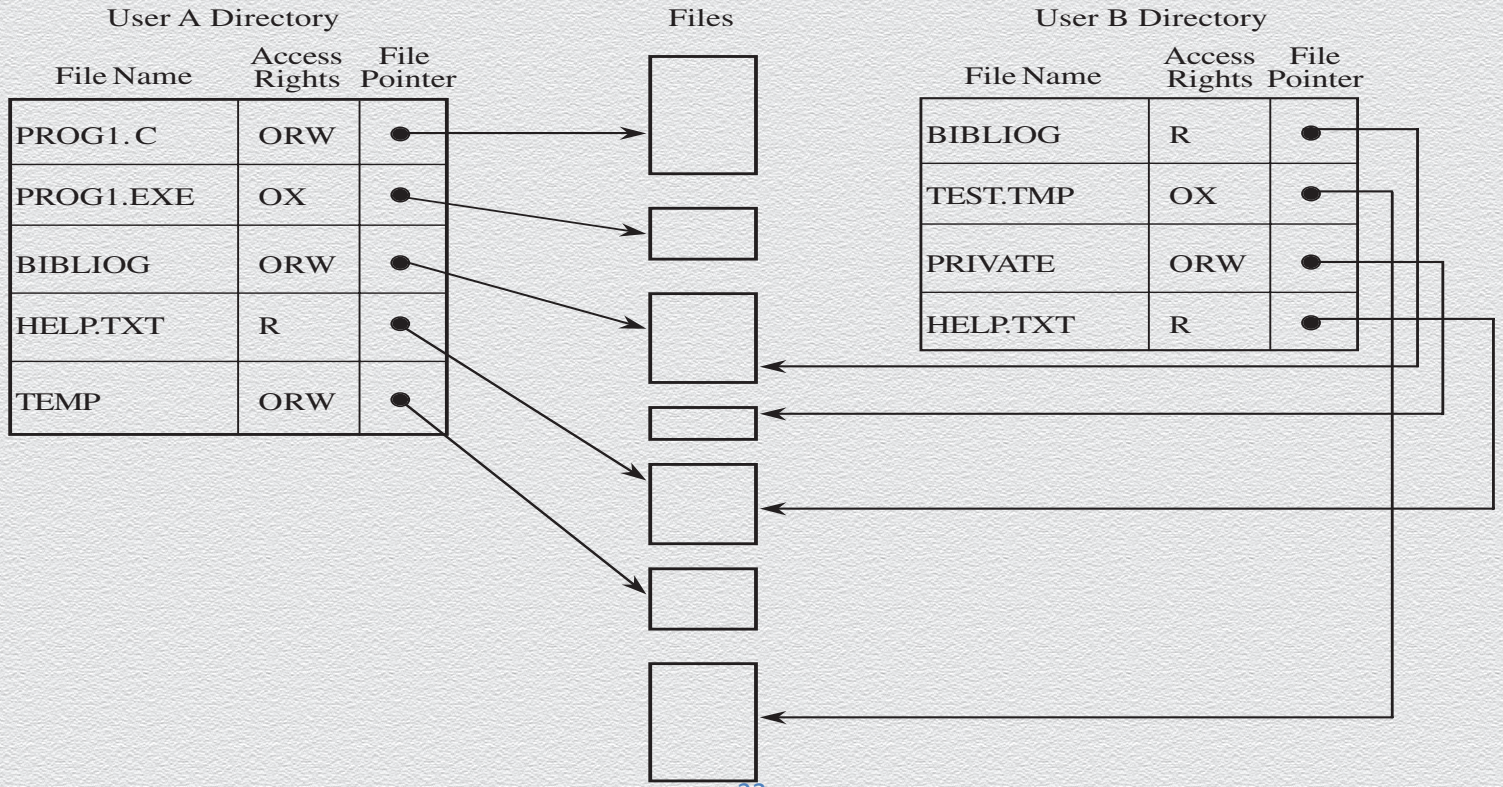
AC policies

- Goals
  - Check every access
  - Enforce least privilege
  - Verify acceptable usage
- Track users’ access
- Enforce at appropriate granularity
- Use audit logging to track accesses

Implementing AC policies

- Reference monitor
- Access control directory
- Access control matrix
- Access control list
- Privilege list
- Capability
- Procedure-oriented access control
- Role-based access control

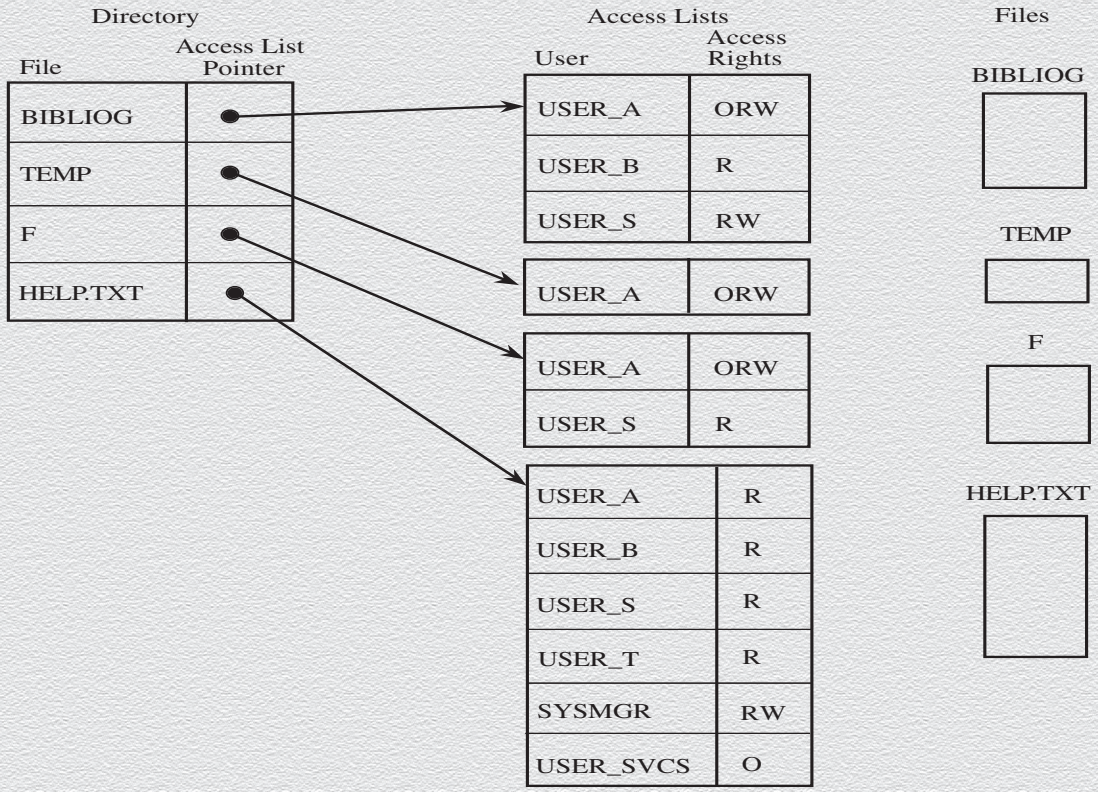
Access control directory



Access control matrix

	BIBLIOG	TEMP	F	HELP.TXT	C_COMP	LINKER	SYS_CLOCK	PRINTER
USER A	ORW	ORW	ORW	R	X	X	R	W
USER B	R	-	-	R	X	X	R	W
USER S	RW	-	R	R	X	X	R	W
USER T	-	-	-	R	X	X	R	W
SYS_MGR	-	-	-	RW	OX	OX	ORW	O
USER_SVCS	-	-	-	O	X	X	R	W

Access control list



Basic access control and information flow models

- Discretionary access control (DAC)
  - owner determines access rights
  - typically identity-based access control: access rights are assigned to users based on their identity
  - e.g., ACM
- Mandatory access control (MAC)
  - system enforce system-wide rules for access control
  - e.g., law allows a court to access driving records without the owners' permission

DAC

- In DAC the user (e.g., owner of resources/files) is responsible for deciding how information is accessed
- Local access decisions of users might conflict with each other
- Basic terms
  - Access control matrix
  - Security policy (specifying who has the access rights to what)
  - Security mechanism (enforce security policies)

DAC and MAC

- When is DAC insufficient?
  - when owner cannot be trusted for the discretion of the data and external protection of the data is necessary
  - e.g., DAC has the danger of right propagation
    - A can read X and write Y
    - B can read Y, but no access to X
    - A reads X, write the content of X to Y, B got access to X
- MAC
  - non-discretionary
  - labels are assigned to subjects and objects
  - owner has no special privileges
  - e.g., Bell-Lapadula, lattices models, SELinux by NSA

Traditional models for MAC

- Bell-LaPadula (BLP)
  - About confidentiality
- Biba
  - About integrity with static/dynamic levels

Bell-LaPadula security model

- The Bell-LaPadula (BLP) model is about information confidentiality
- It was developed to formalize the US Department of Defense multilevel security policy

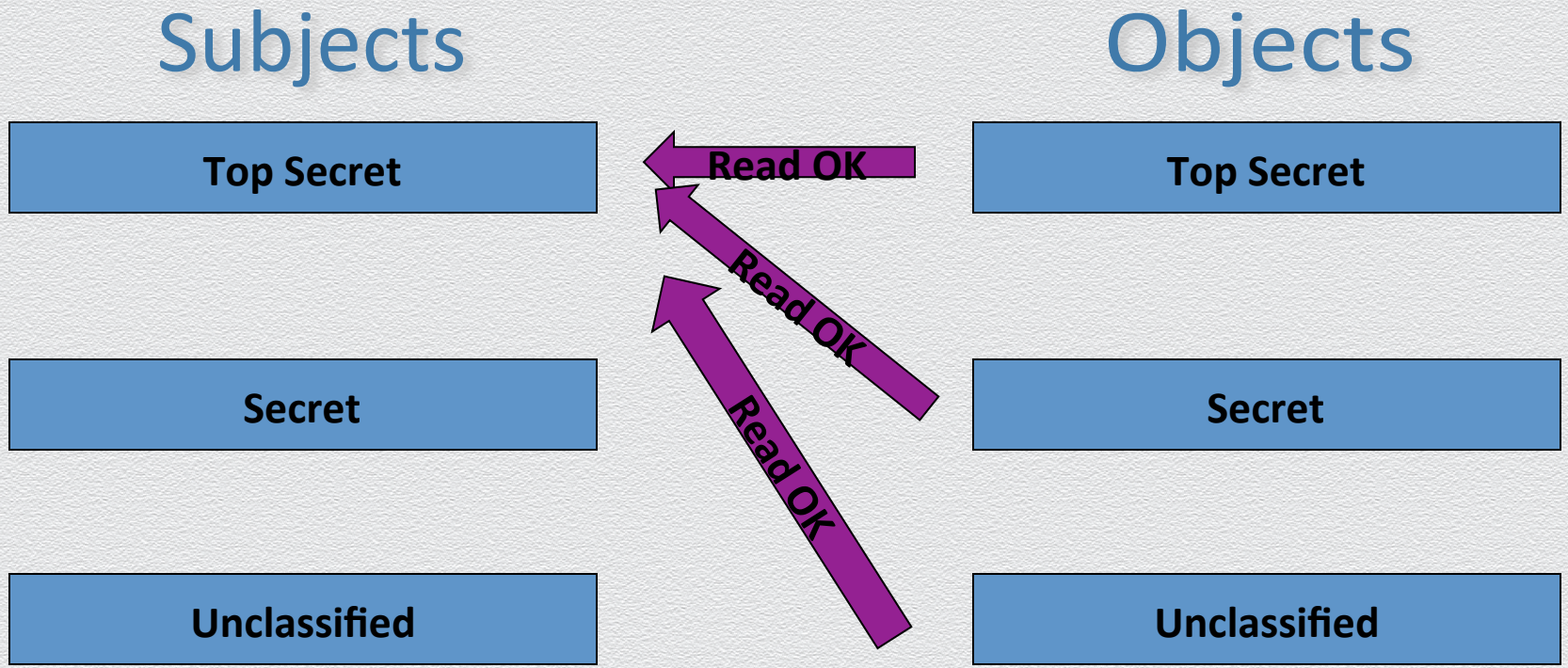
Bell – LaPadula - details

- Each user subject and information object has a fixed security class – labels
- Use the notation **≤** to indicate **dominance**
- Simple Security (ss) property:

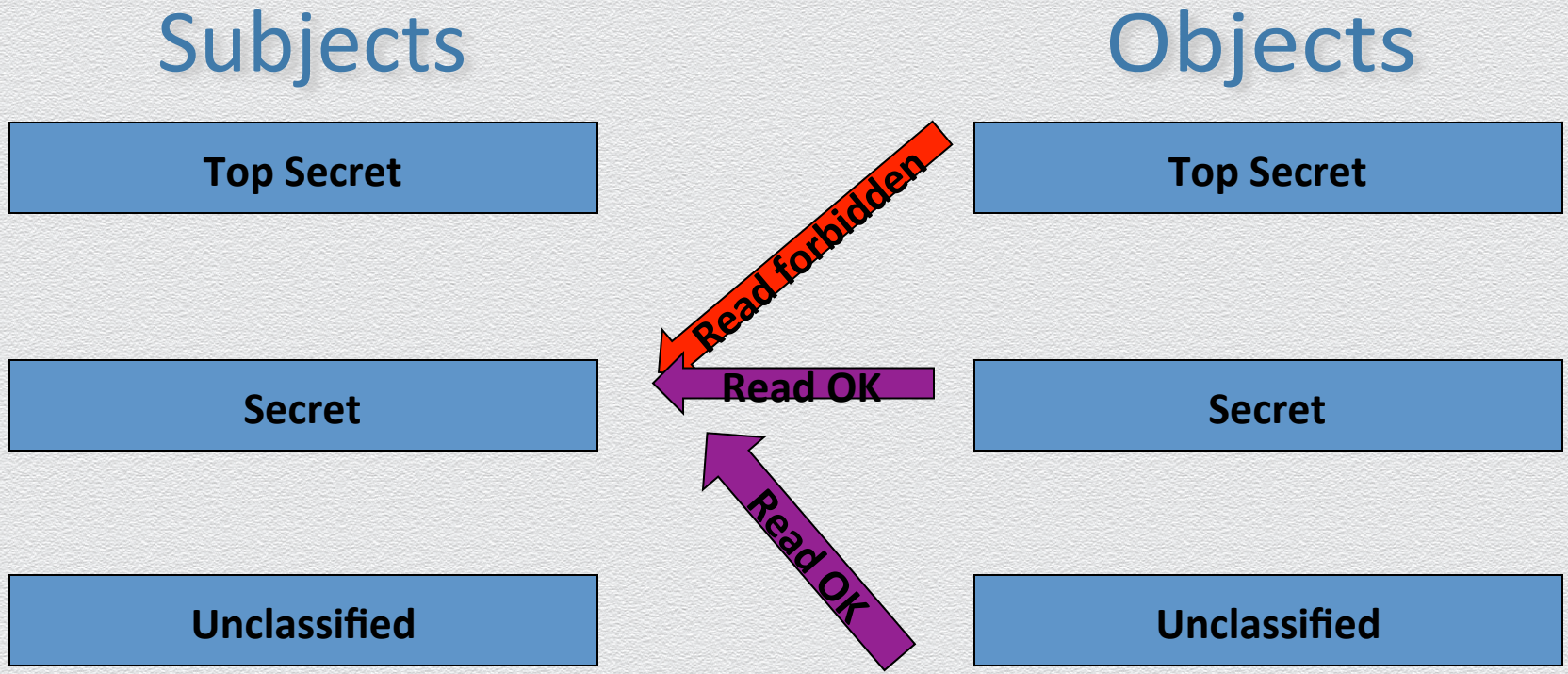
  the no read-up property

  - s subject s has read access to an object o iff the class of the subject C(s) is greater than or equal to the class of the object C(o)
  - i.e. subjects s can read objects o iff C(o) ≤ C(s)

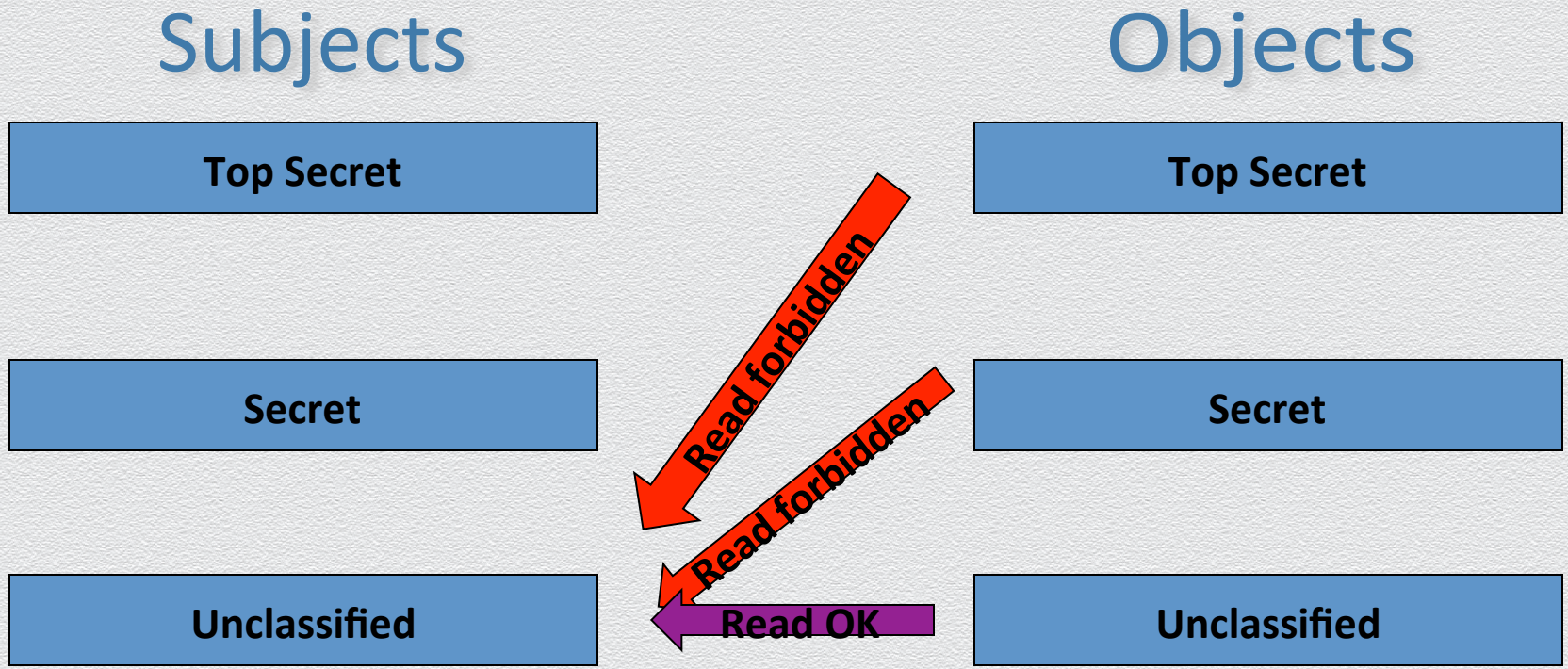
Access control: Bell-LaPadula



Access control: Bell-LaPadula



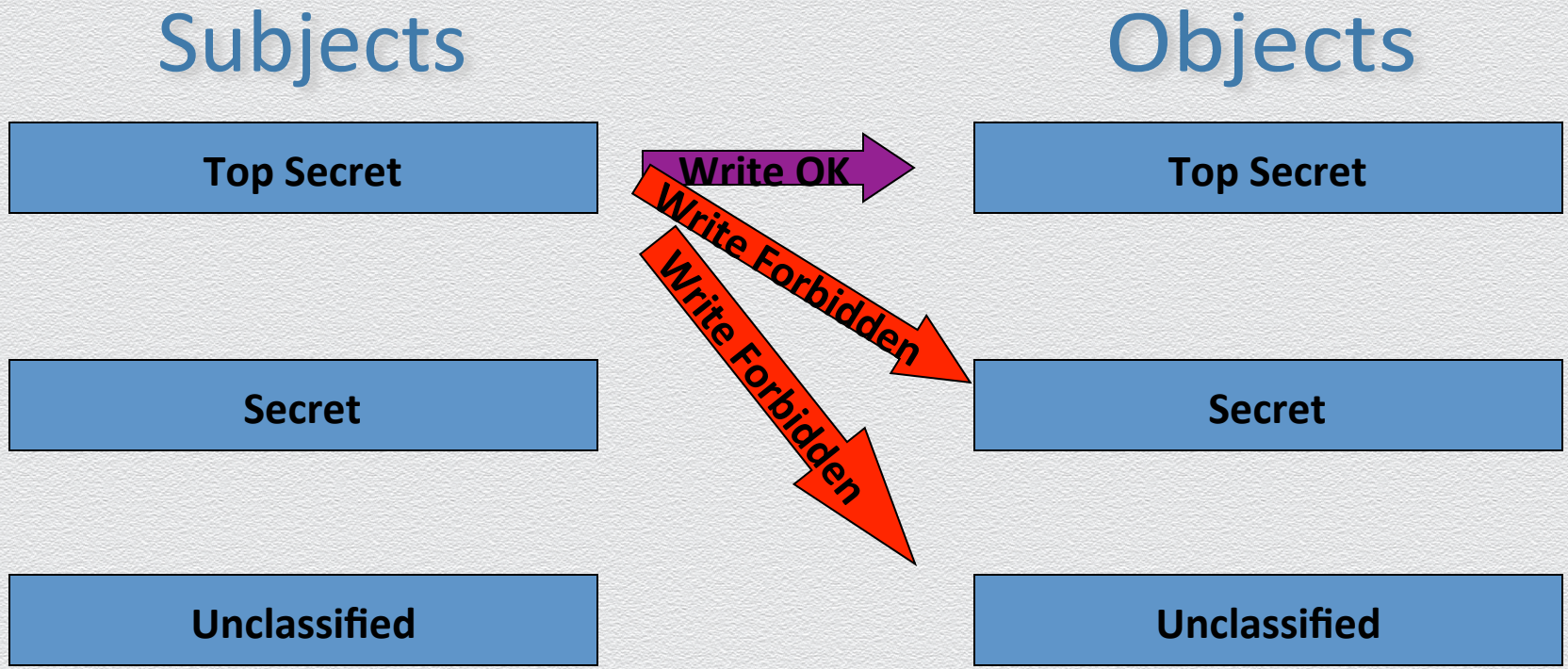
Access control: Bell-LaPadula



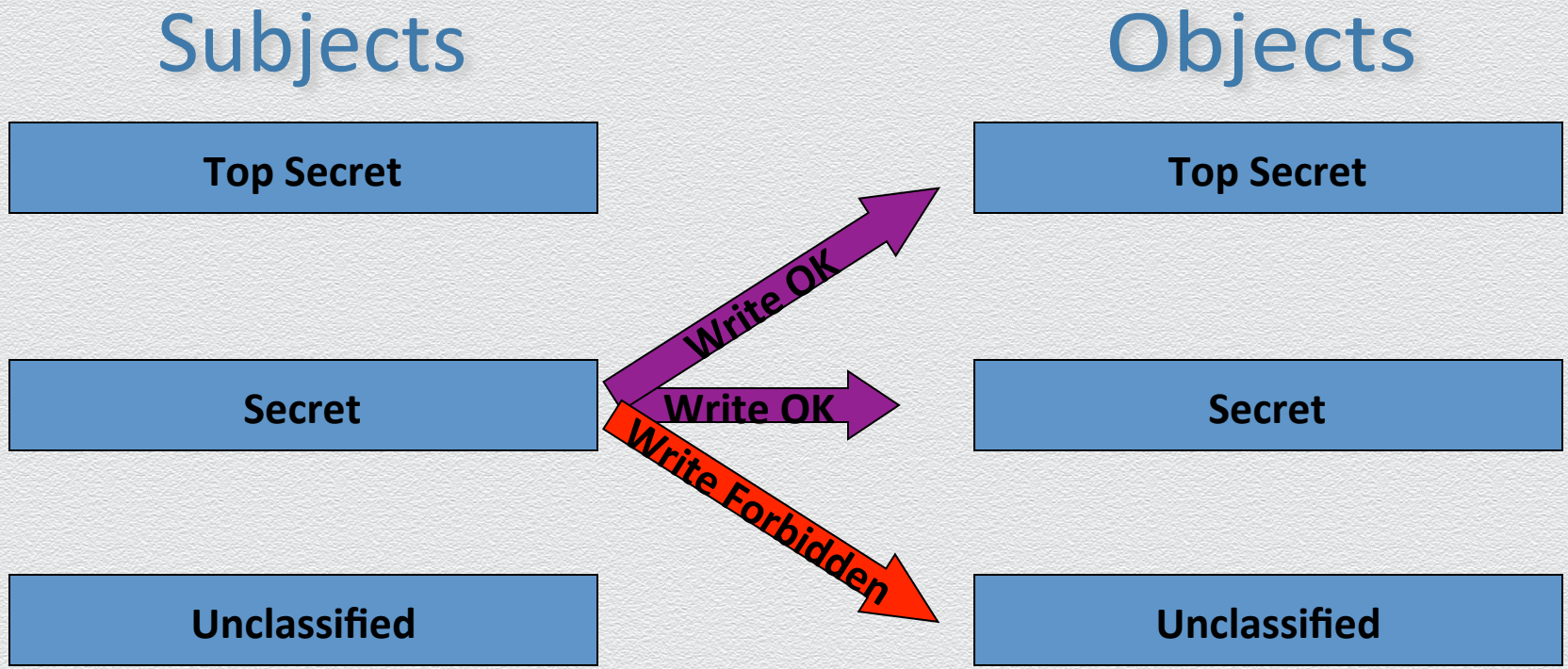
Bell - LaPadula (2)

- * property (star):
  the no write-down property
  - A subject s can write to object *p* if C(s) ≤ C (p)

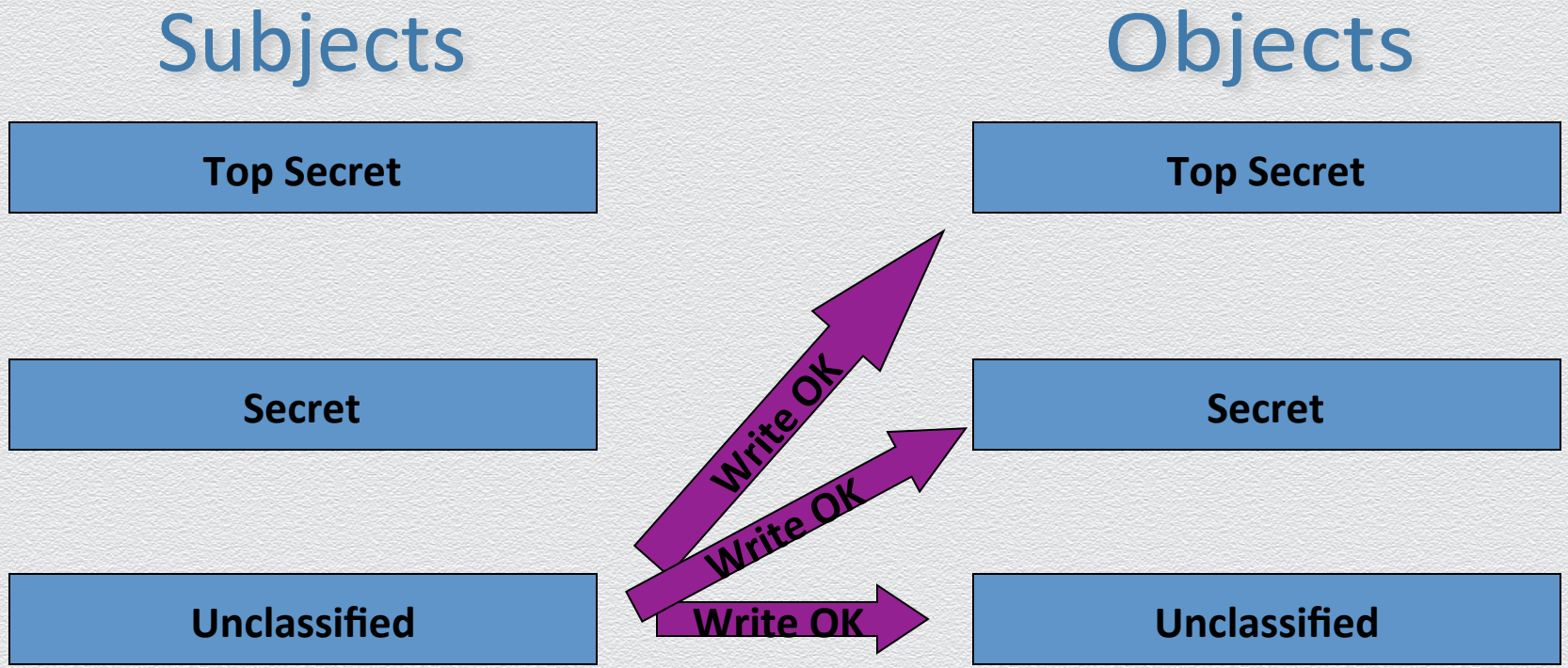
Access control: Bell-LaPadula



Access control: Bell-LaPadula



Access control: Bell-LaPadula



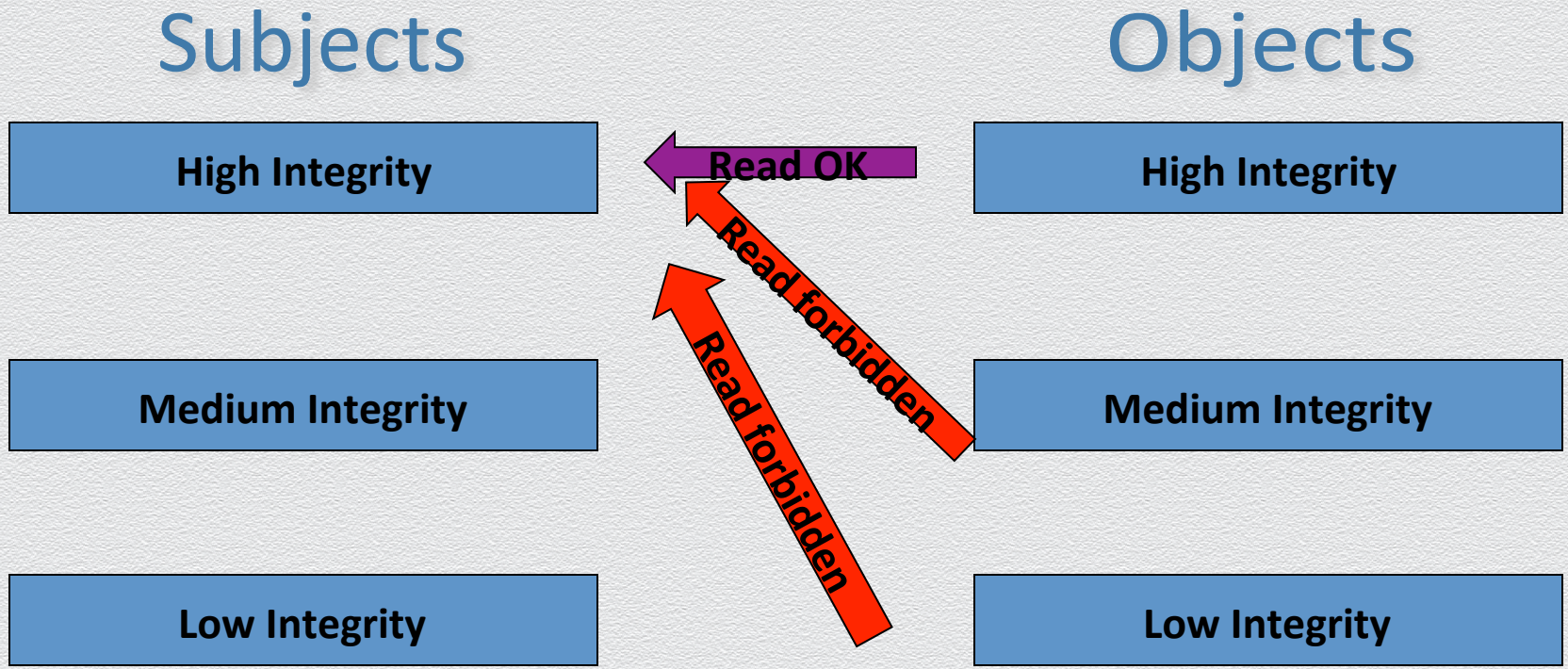
Security models - Biba

- Based on the Cold War experiences, information *integrity* is also important, and the Biba model, complementary to Bell-LaPadula, is based on the flow of information where preserving integrity is critical.
- The “dual” of Bell-LaPadula

Integrity control: Biba

- Designed to preserve integrity, not limit access
- Three fundamental concepts:
  - Simple Integrity Property – no read down
  - Star Integrity Property (*) – no write up
  - No execute up

Integrity control: Biba



Integrity control: Biba