

# Database Security: Status and Prospects



**Sushil Jajodia**  
**George Mason University**  
**[csis.gmu.edu](http://csis.gmu.edu)**



# Statistical Databases

- **Objective:** To provide aggregate statistics, while protecting the confidentiality of the individual entity
- **Example:** Permit queries such as “Print the average income of all individuals in Northern Virginia” without disclosing income of a particular individual



# Limitations

- **Impossible to prevent compromises, especially in the face of collusions**





# Discretionary Access Controls

- **Users can protect what they own**
- **Owner may grant access to others**
- **Owner may define the type of access (read/write/execute) given to others**



# Inherent Weakness of DAC

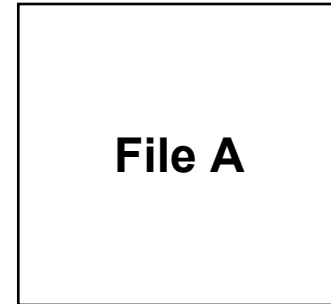
- **Unrestricted DAC allows information from an object which can be read by a subject to be written to any other object**
  - **Bob is denied access to file Y, so he asks cohort Alice to copy Y to X that he can access**
- **Suppose our users are trusted not to do this deliberately. It is still possible for Trojan Horses to copy information from one object to another.**



# Trojan Horse Example [1 of 3]

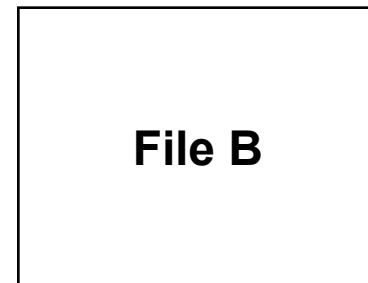
**User Alice**

**r: Alice; w:Alice**



**User Bob**

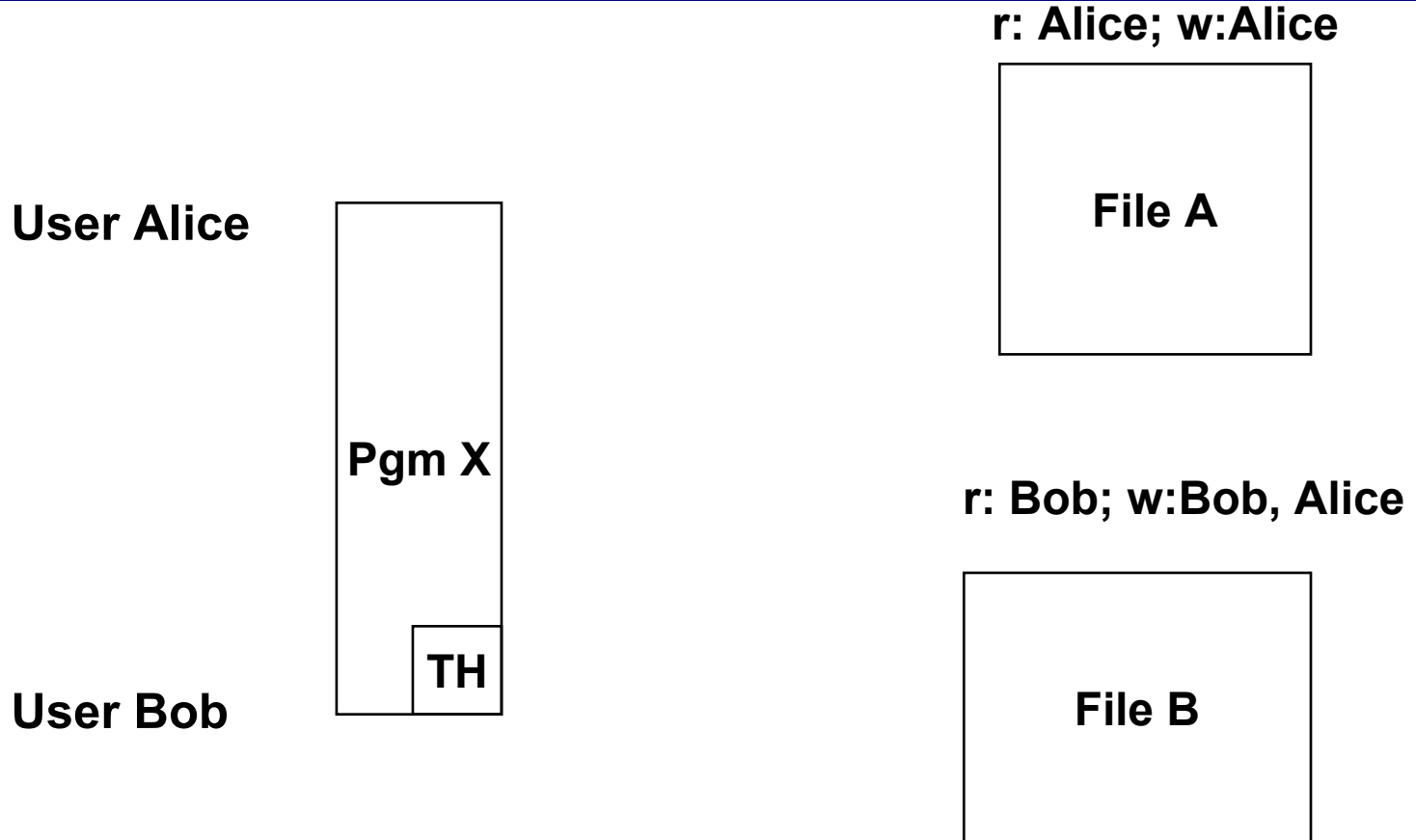
**r: Bob; w:Bob**



**User Bob cannot read the file A!**

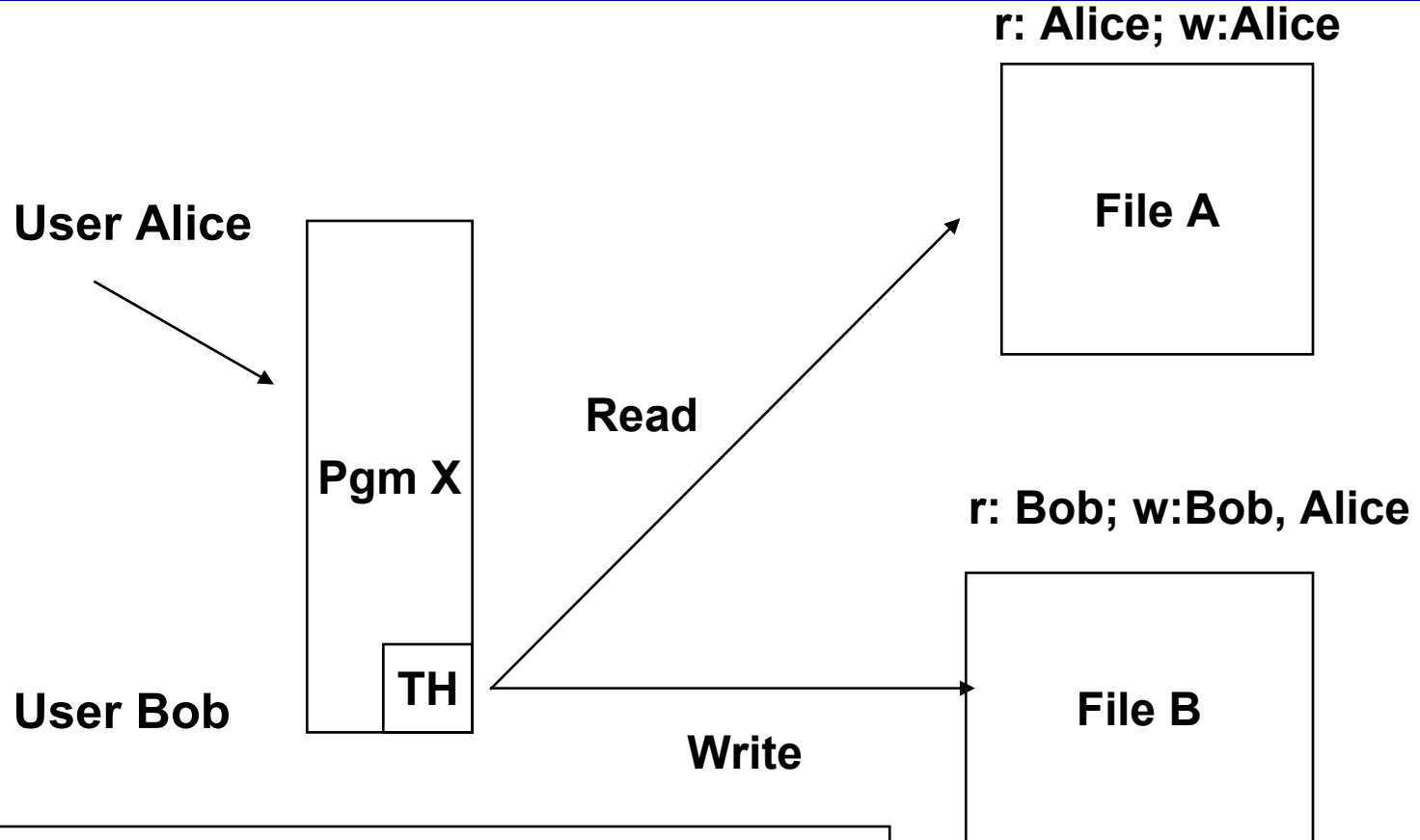


# Trojan Horse Example [2 of 3]





# Trojan Horse Example [3 of 3]



User Bob can read contents of file A copied to file B



# Lattice-based Model

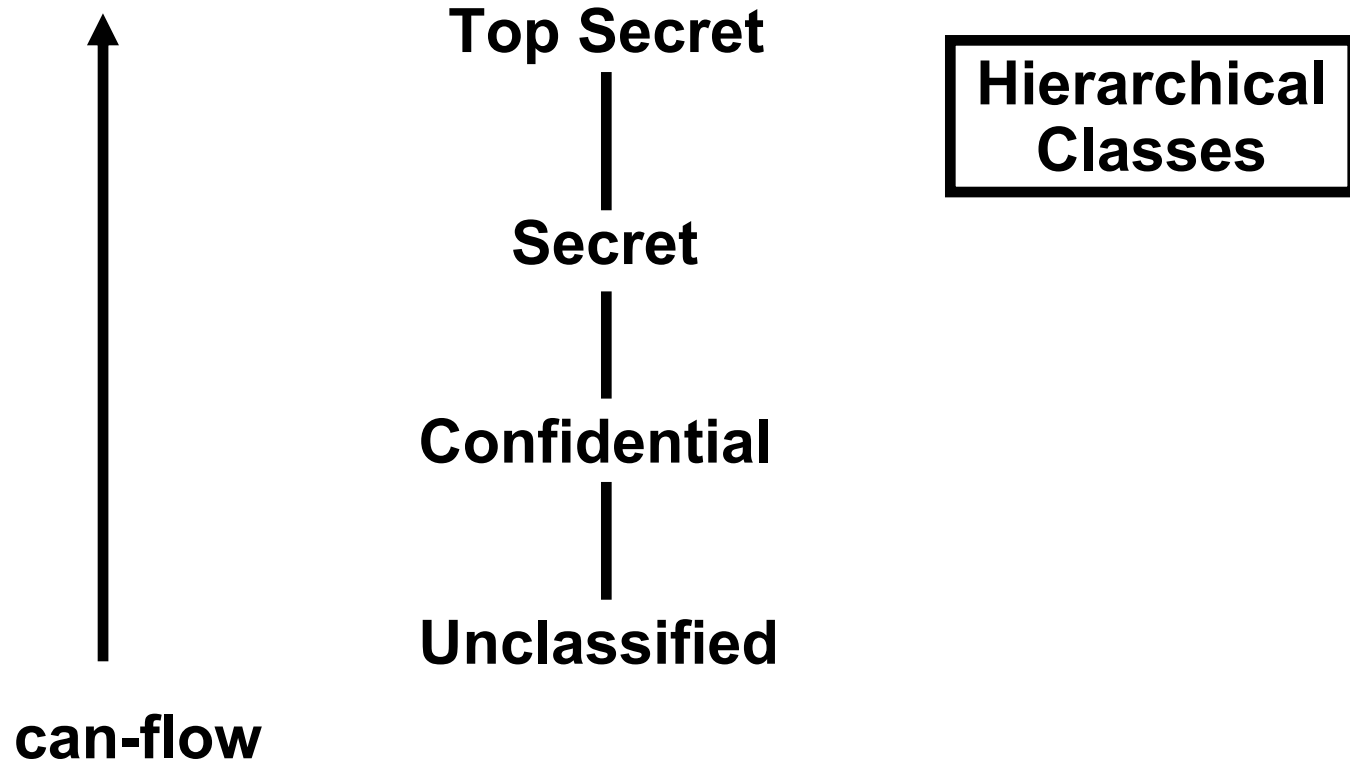
$$\langle SC, \rightarrow, \oplus \rangle$$

- SC**                      **finite set of security classes**
- $\rightarrow \subseteq SC \times SC$**                       **can-flow relation**  
**(partial order)**
- $\oplus: SC \times SC \rightarrow SC$**                       **class-combining operator**  
**(lub operator on SC)**





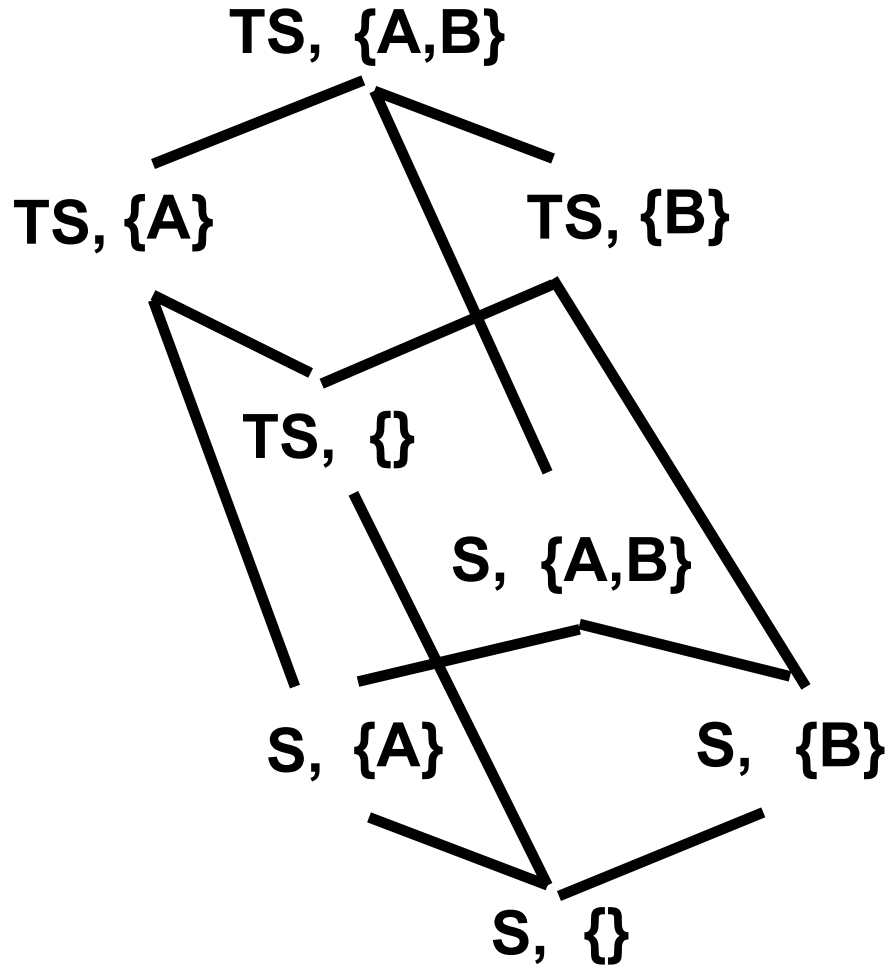
# Lattice Structures



- reflexive and transitive edges are implied but not shown



# Hierarchical Classes with Compartments





# Bell-LaPadula (BLP) Model

- **The Simple Security Property:**

**A subject  $S$  is allowed a read access to an object  $O$  only if  $L(S)$  is greater than or equal to  $L(O)$ .**

- **The \*-property:**

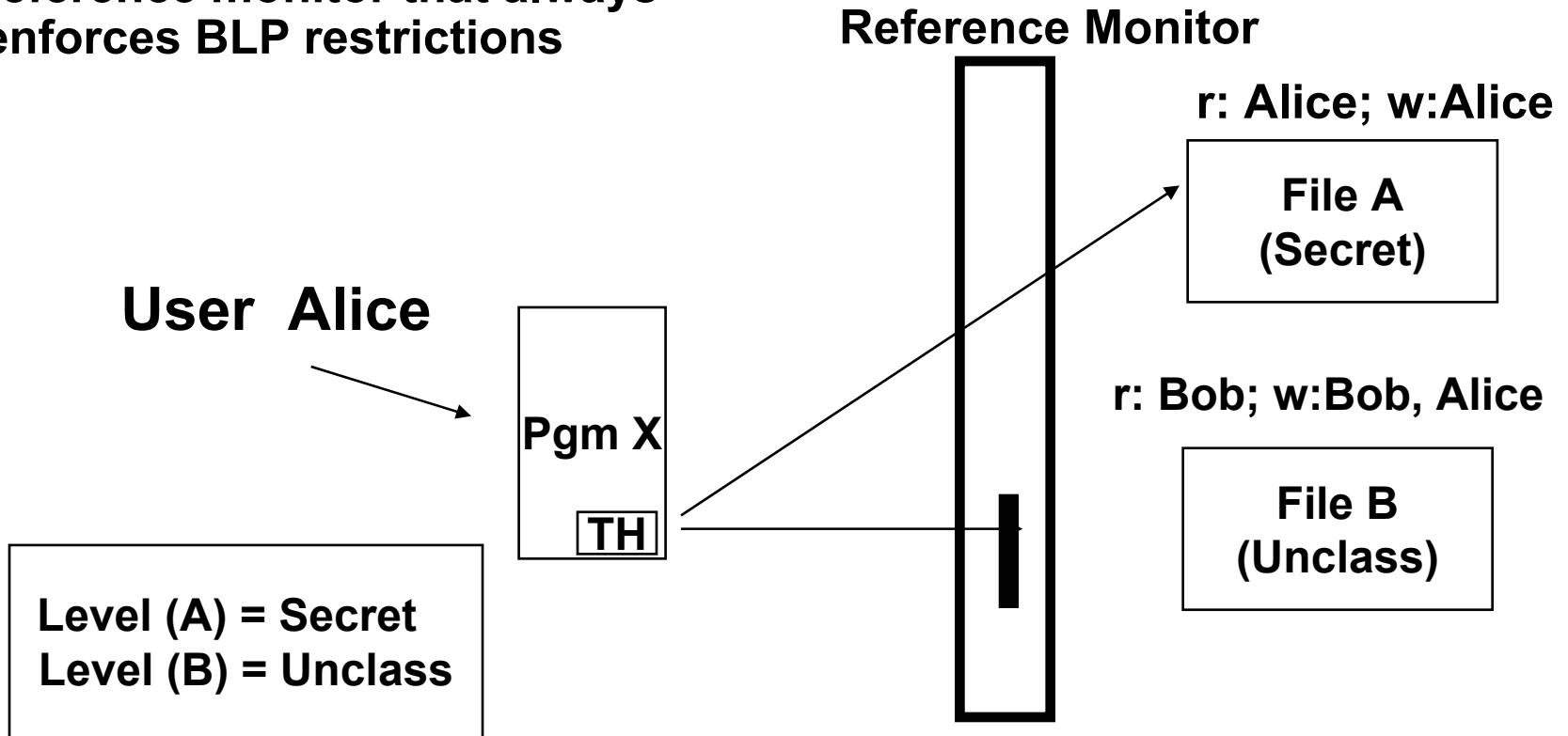
**A subject  $S$  is allowed a write access to an object  $O$  only if  $L(S)$  is less than or equal to  $L(O)$ .**

**These restrictions together ensure that there is no direct flow of information from high to low subjects!**



# Why the \*-Property

Suppose that there is a reference monitor that always enforces BLP restrictions





# Limitations

- **Information tends to become overclassified**
- **Does not protect against against violations that produce illegal information flow through indirect means**
  - **Inference Channels - A user at a low security class uses the low data to infer information about high security class**
  - **Covert channels - Require two active agents, one at a low level and the other at a high level and an encoding scheme**



# Covert Channels

- **A covert channel is a communication channel based on the use of system resources not normally intended for communication between the subjects (processes) in the system**



# Covert Channels

High Principal



High Trojan Horse  
Infected Subject

- Information is leaked unknown to the high principal

Low Principal



Low Trojan Horse  
Infected Subject



COVERT  
CHANNEL



# MLS Relational Data Model

<b>Starship</b>	<b>Objective</b>		<b>Destination</b>		<b>TL</b>
<b>Enterprise U</b>	<b>Exploration U</b>		<b>Talos U</b>		<b>U</b>
<b>Enterprise U</b>	<b>Spying S</b>		<b>Rigel S</b>		<b>S</b>
<b>Enterprise U</b>	<b>Spying S</b>		<b>Mars TS</b>		<b>TS</b>

- **Core properties**
  - **Entity Integrity**
  - **Null Integrity**
  - **Inter-Instance Integrity**
  - **Polyinstantiation Integrity**



# Polyinstantiation

<b>S-instance:</b>	<b>Starship</b>		<b>Objective</b>		<b>Destination</b>	
	<b>Enterprise</b>	<b>U</b>	<b>Exploration</b>	<b>U</b>	<b>Talos</b>	<b>U</b>
	<b>Voyager</b>	<b>S</b>	<b>Spying</b>	<b>S</b>	<b>Mars</b>	<b>S</b>
<b>U-instance:</b>	<b>Starship</b>		<b>Objective</b>		<b>Destination</b>	
	<b>Enterprise</b>	<b>U</b>	<b>Exploration</b>	<b>U</b>	<b>Talos</b>	<b>U</b>

**U-user: Insert (Voyager, Exploration, Mars).**

**Refuse Update — Covert channel**

**Accept Update:**

- 1) Overwrite high data — May lead to serious integrity problems**
- 2) Do not overwrite high data — Show the high user both tuples**

**The 2nd option leads to entity polyinstantiation.**



# Example 1

- **Public\_Info – Everyone is allowed access**
- **US\_Only – US citizens only**
- **Internal\_Reports – Everyone unless explicitly denied**
- **Accounts – Only those explicitly authorized**
- **TVA\_Project – Project members only; temporarily contains information that should not be shared**



# Example 2: Electronic Library System

- **An article A that is published under project P can be made available to all members of P**
- **Any user at the rank of manager or above can access A**
- **If the user is a sponsor who funded the project or the writing of A, then A can be released only after a proprietary notice is added**
- **No one else should have access to the article**



# Example 3: Sealed-bid Auctions

- **Three kinds of participants**
  - Auctioneer, Supplier, Bidders
- **The supplier fills in the item to be auctioned, closing time, and the minimum price**
- **All users are either registered or must register before they can submit a bid**
- **Bidders may submit a bid by specifying the item and a bidding price if the current time is before the close of the auction**
- **Auctioneer can fill in “no good” in the status field if the maximum price of all the bids  $<$  minimum price and “completed” if the maximum price  $\geq$  minimum price**



# Options

- **Unix**
  - **xrwxrwxrw**
- **Database management systems**
  - **GRANT <privilege> ON <relation>**  
**TO <users>**  
**[WITH GRANT OPTION]**
  - **REVOKE <privileges>**  
**[ON <relations>]**  
**FROM <users>**



# Limitations

- **Based on simple paradigm (e.g., simple authorization tuples) and with basic functionalities**
- **Access control rules under the control of a single party**
- **Separated from user's authentication**

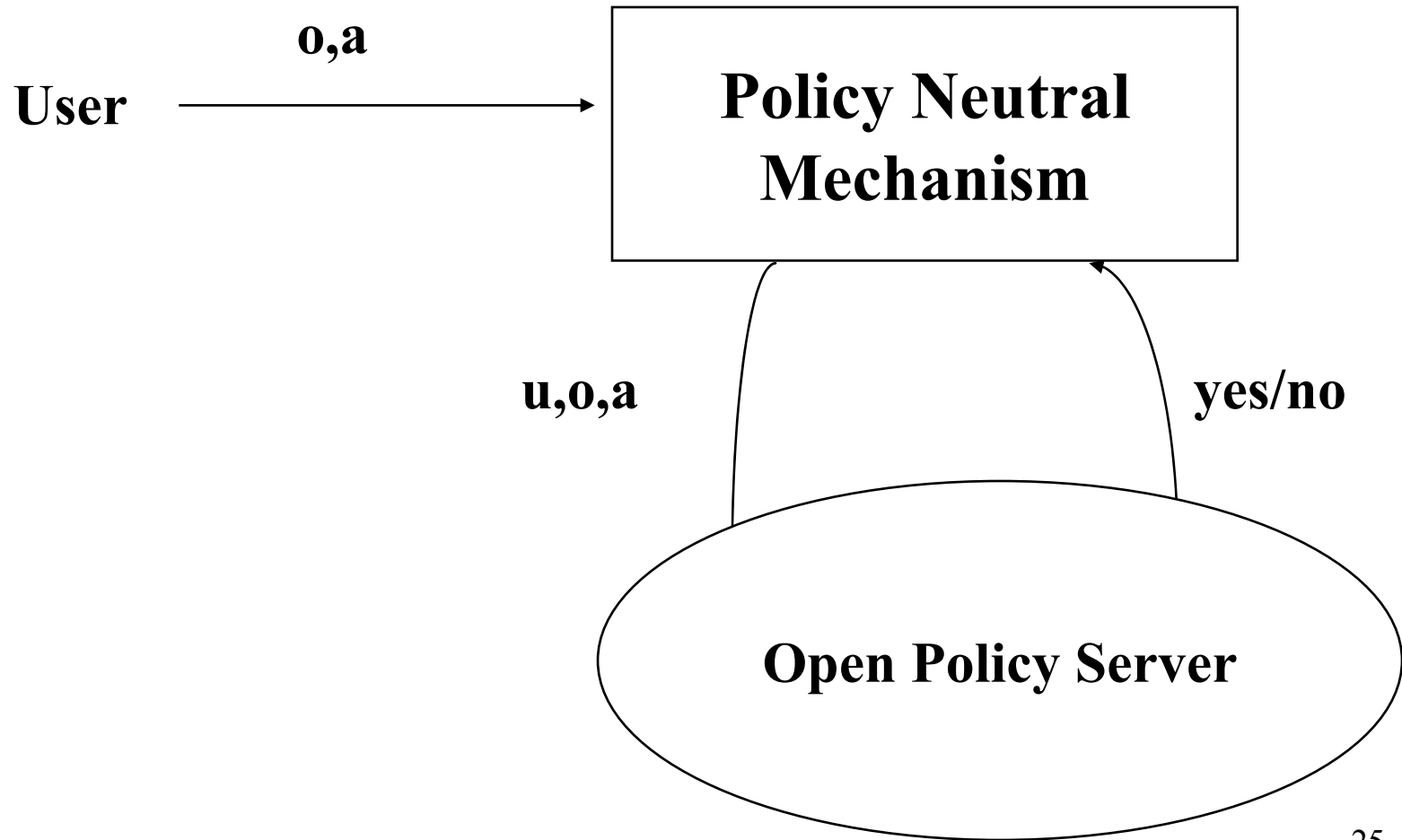


# Suggested Approach

- **Separation between policy server and enforcement mechanism**
  - Policy neutral mechanism
  - For each policy, a security server is specified
  - Change of policy  $\Rightarrow$  Change of server



# Policy Neutral Mechanism





# Current Access Control Models

**Recent access control models try to include at least**

- **Positive as well as negative authorizations**
- **Authorization propagation based on abstraction hierarchies (e.g., user, group, role, object)**
- **Conflict resolution and decision strategies**
- **Additional implication relationships**

**GOAL: Support flexible, multiple policies**



# Other Extensions

- **Different independent access control policies may need to be maintained and applied in combination (Policy composition)**
- **Authentication may not always be possible or wanted (Access control based on attributes)**
- **Need to augment expressiveness of access control - Purpose-based restrictions and Support for dynamic conditions (Interactive access control)**



# Scientific Data Sharing

- **Scientific data sharing has great promise and growing interest**
  - **Online genetic databases have played a large role in the mapping of human genome**
  - **Online sky surveys enable researchers worldwide to analyze datasets collected by others, and even to generate new multi-spectral datasets**
- **NIH would like to see data sharing of data that are unique and cannot be readily replicated (such as human neuroimager)**

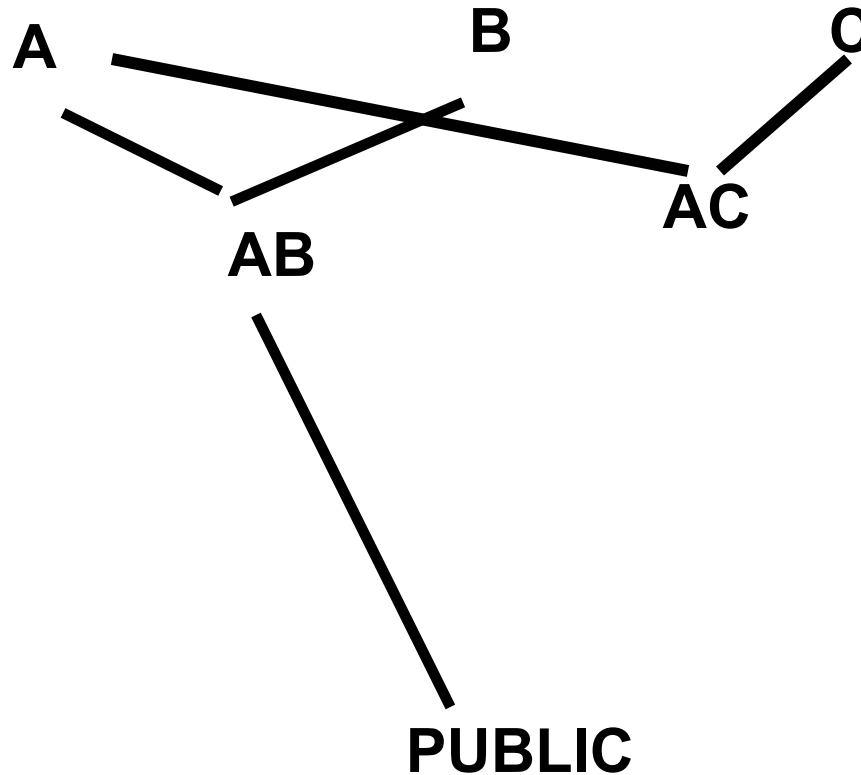


# Researcher Concerns

- **Would like to benefit from first use of their data**
- **Protect the identity and privacy of human research participants**
- **Express their sharing intent: Intended scope of visibility for a dataset**



# A Lattice-based Model for Information Sharing





# A Lattice-based Model for Information Sharing

$$\langle SC, \rightarrow, \otimes \rangle$$

**SC**                      **finite set of data access domains**

$\rightarrow \subseteq SC \times SC$               **can-flow relation  
(partial order)**

$\otimes : SC \times SC \rightarrow SC$  **class-combining operator  
(glb operator on SC)**



# Fair Information Flow

- **The Simple Sharing Property:**

A subject **S** is allowed a read access to an object **O** only if  $L(S)$  is greater than or equal to  $L(O)$ .

- **The Confinement Property:**

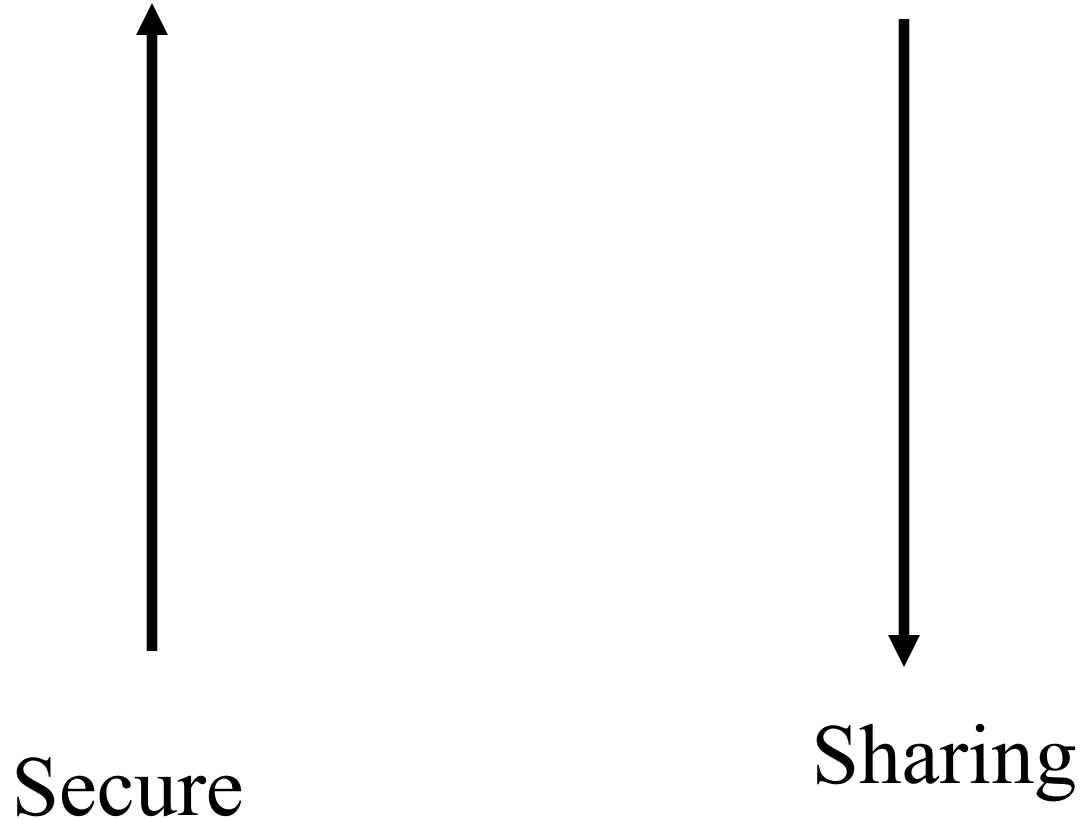
A subject **S** is allowed a write access to an object **O** only if  $L(S)$  is greater than or equal to  $L(O)$ .

- **The Fairness Property:**

If a subject **S** reads object **O**, then  $\text{write-label}(S) \leq \text{glb}(\text{write-label}(S), \text{label}(O))$ .



# Security vs Sharing



Secure

Sharing





# Final Remarks

- **Cryptography is the solution**
- **Firewalls are the solution**
- **Intrusion detection systems are the solution**

**NOT!!!**





# Future Focus

- **Access control**
- **Application security**
- **DBMS security**

