Trusted Computing (Cnt.): Access Control Models

CS463/ECE424

University of Illinois

Outline

Bell-LaPadula (BLP)

Biba

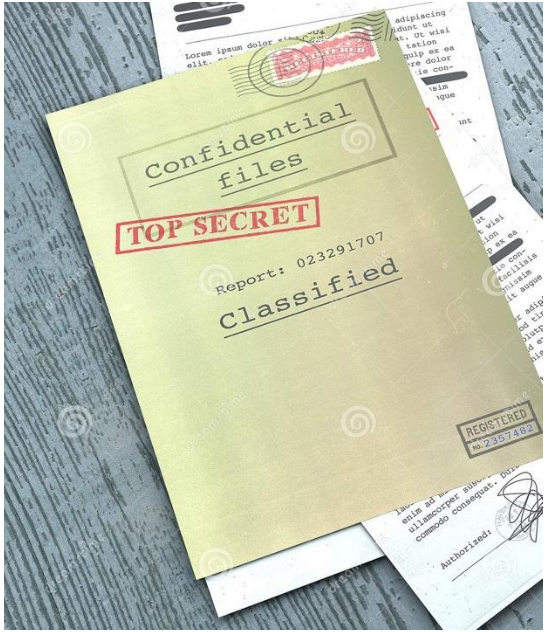
Clark-Wilson

Chinese Wall

Multilevel Security (MLS)

An MLS system

- Has system resources (data, files) at more than one security level (i.e., public and proprietary)
- Permits concurrent access by users who differ in “security clearance and need-to-know”
- Prevents each user from accessing resources for which the user lacks authorization



IETF RFC 2828

Bell-LaPadula (BLP) Model

- Formal model for access control
  - Developed in 1970s
- *Subjects* and *objects* are assigned a security class
  - A subject (user) has a *security clearance*
  - An object (file) has a *security classification*
  - Form a hierarchy and are referred to as security levels
    - top secret > secret > confidential > restricted >unclassified
  - Security classes control how a subject may access an object

BLP Model Access Modes

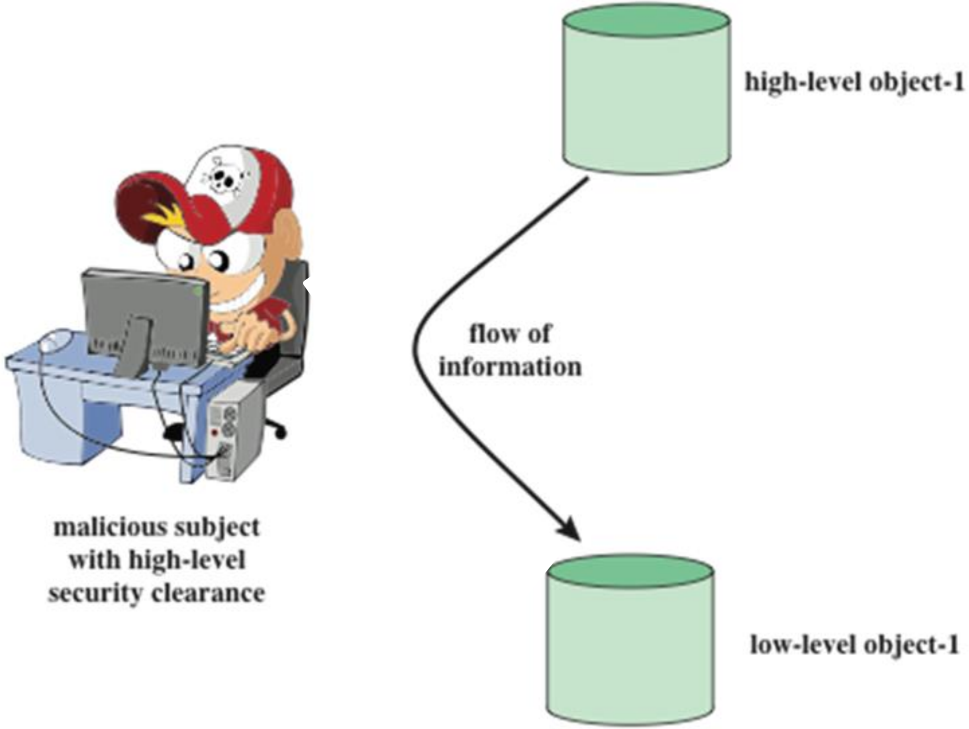
- READ
  - The subject is allowed only read access to the object
- APPEND
  - The subject is allowed only write access to the object
- WRITE
  - The subject is allowed both read and write access to the object
- EXECUTE
  - The subject is allowed neither read nor write access to the object but may invoke the object for execution

No Read Up and No Write Down

- **No read up**
  - Subject can only read an object of less or equal security level
  - Referred to as the simple security property (**ss-property**)
- **No write down**
  - A subject can only write into an object of greater or equal security level
  - Referred to as the ***-property**

Threat Intuition:

protect the confidentiality of information at



Discretionary Control

- An individual (or role) may grant to another individual (or role) access to a document
  - Based on the owner's discretion, but
  - **These are constrained by the MAC (mandatory access control) rules**
- Site policy overrides any discretionary access controls
- This is called the **ds-property**

A user cannot overwrite the BLP model to give away information to unauthorized persons

BLP Formal Description

- Current state of system: (b, M, f, H)
  - current access set **b**: triples of (s, o, a)
    - subject **s** has current access to object **o** in access mode **a**
    - access mode: read, append, write, execute
  - access matrix **M**: matrix of M_{ij}
    - access modes of subject S_i to access object O_j
  - level function **f**: security level of subjects and objects
    - f_o (O_j) is the classification level of object O_j
    - f_s (S_i) is the security clearance (i.e., maximum security level) of subject S_i
    - f_c (S_i) is the current security level of subject S_i
  - hierarchy **H**: a directed rooted tree of objects

BLP Formal Description

- The three BLP properties:
  - **ss-property**: every (S_i, O_j, read) has $f_c(S_i) \geq f_o(O_j)$
  - ***-property**: every $(S_i, O_j, \text{append})$ has $f_c(S_i) \leq f_o(O_j)$ and

    every (S_i, O_j, write) has $f_c(S_i) = f_o(O_j)$ **[WHY??]**
  - **ds-property**: every (S_i, O_j, A_x) has $A_x \in M_{ij}$

- These are used to define the concepts of secure state and secure system.

BLP Secure System

- The state (b, M, f, H) is **secure** if every element of b satisfies the three properties.
- A **system** defines a set of transitions that allow changes to the four components of the system, (b, M, f, H).
- A system is **secure** if system transitions on secure states result only in secure states.

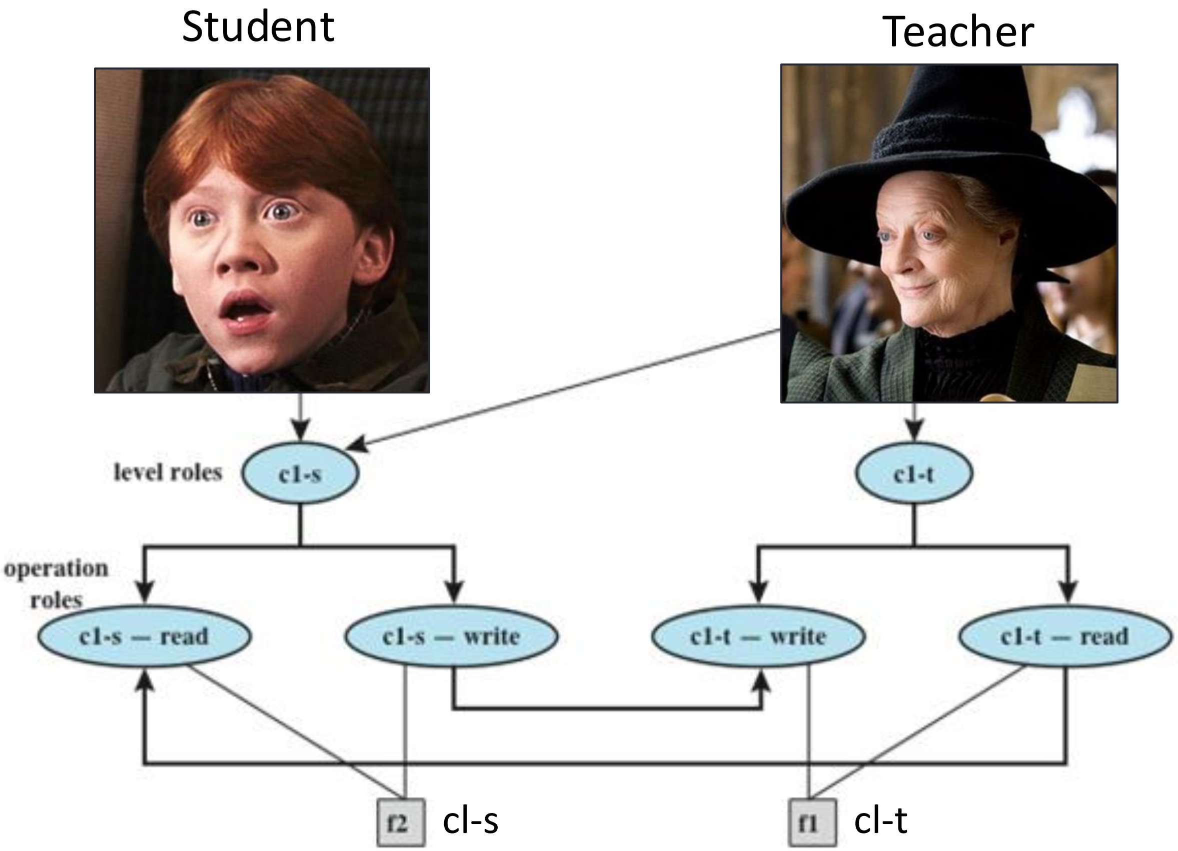
BLP Transition Rules

1. **Get access:** Add a triple (subject, object, access-mode) to the current access set b.
2. **Release access:** Remove a triple (subject , object , access-mode) from the current access set b.
3. **Change object level:** Change the value of $f_o(O_j)$ for some object O_j.
4. **Change current level:** Change the value of $f_c(S_i)$ for some subject S_i.
5. **Give access permission:** Add an access mode to some entry of the access permission matrix M.
6. **Rescind access permission:** Delete an access mode from some entry of M.
7. **Create an object:** Attach an object to the current tree structure H as a leaf.
8. **Delete a group of objects:** Detach from H an object and all other objects beneath it in the hierarchy. This renders the group of objects inactive.

s: student

t: teacher

*-property



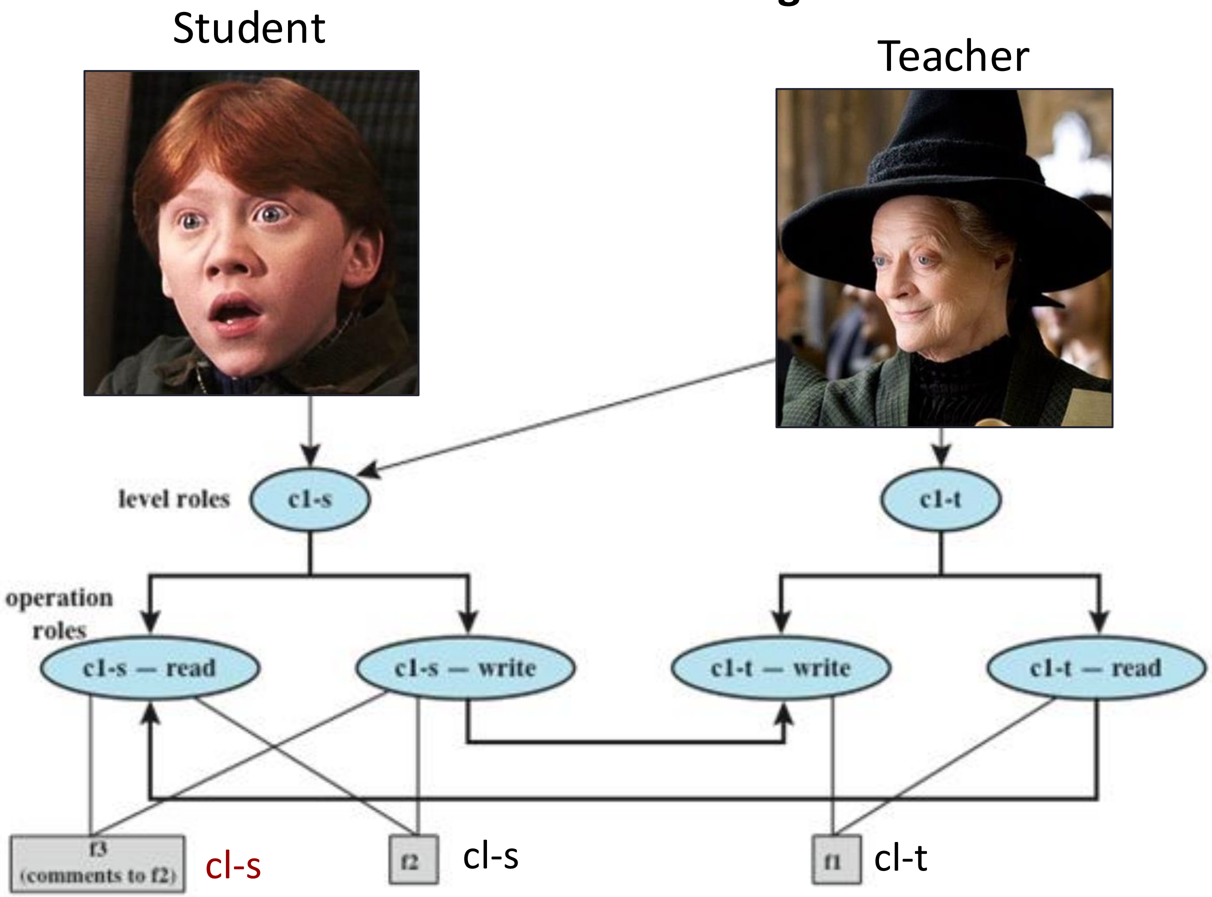
Two files are created: f1: cl-t; f2: cl-s

How does teacher give feedback via comments?

s: student

t: teacher

*-property

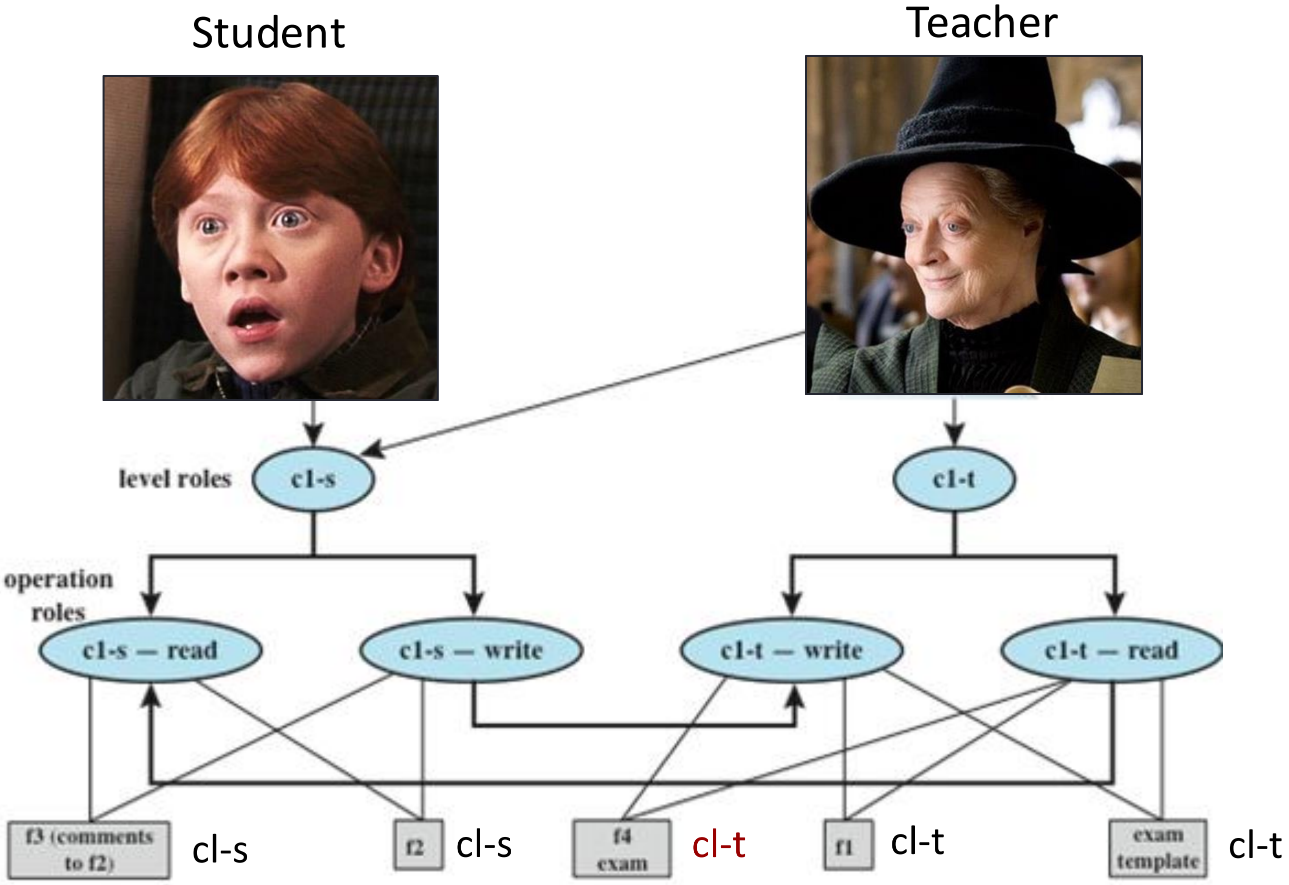


A third file is added f3: cl-s

s: student
t: teacher

ss-property
and *-property

How does teacher create an exam?



An exam is created based on an existing template f4: cl-t

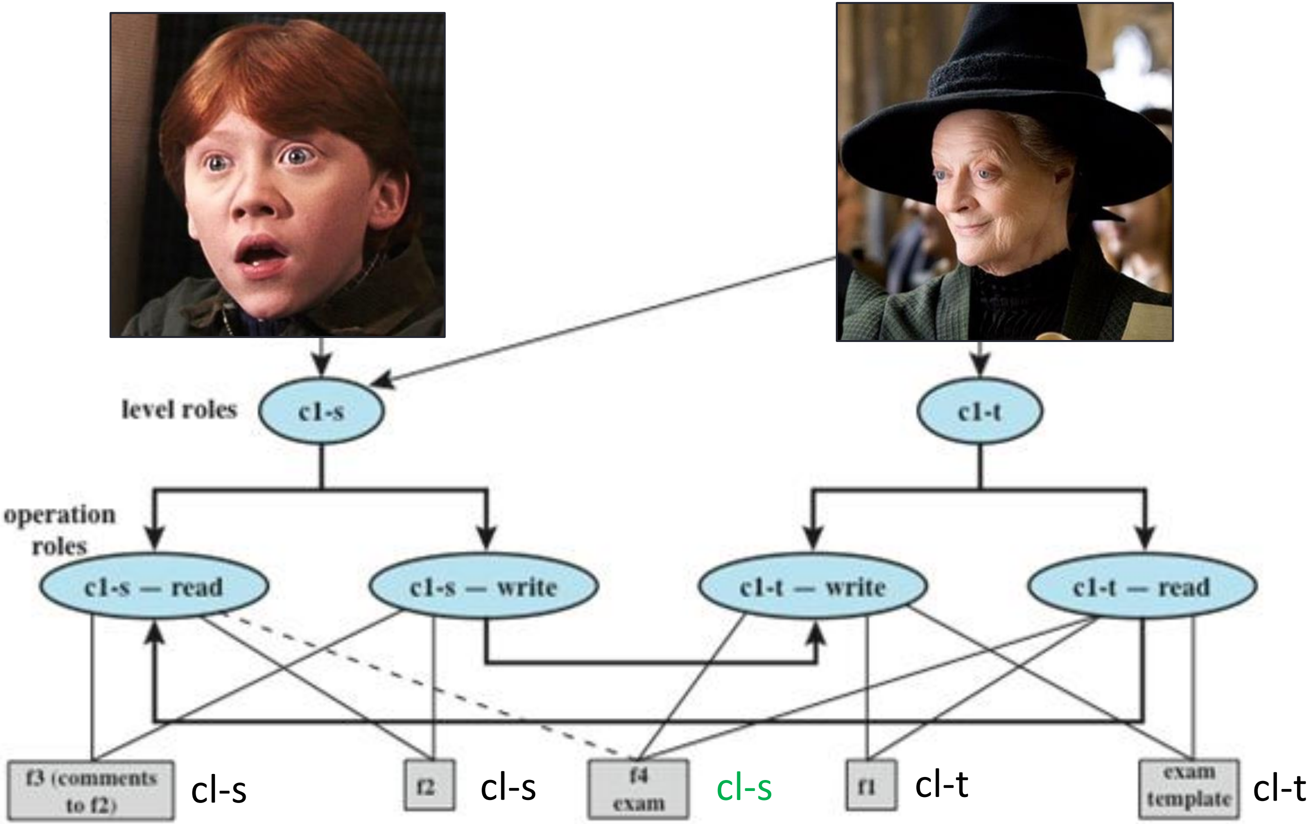
We need **secure transition rules!**

s: student
t: teacher

ss-property
and *-property

Student

Teacher

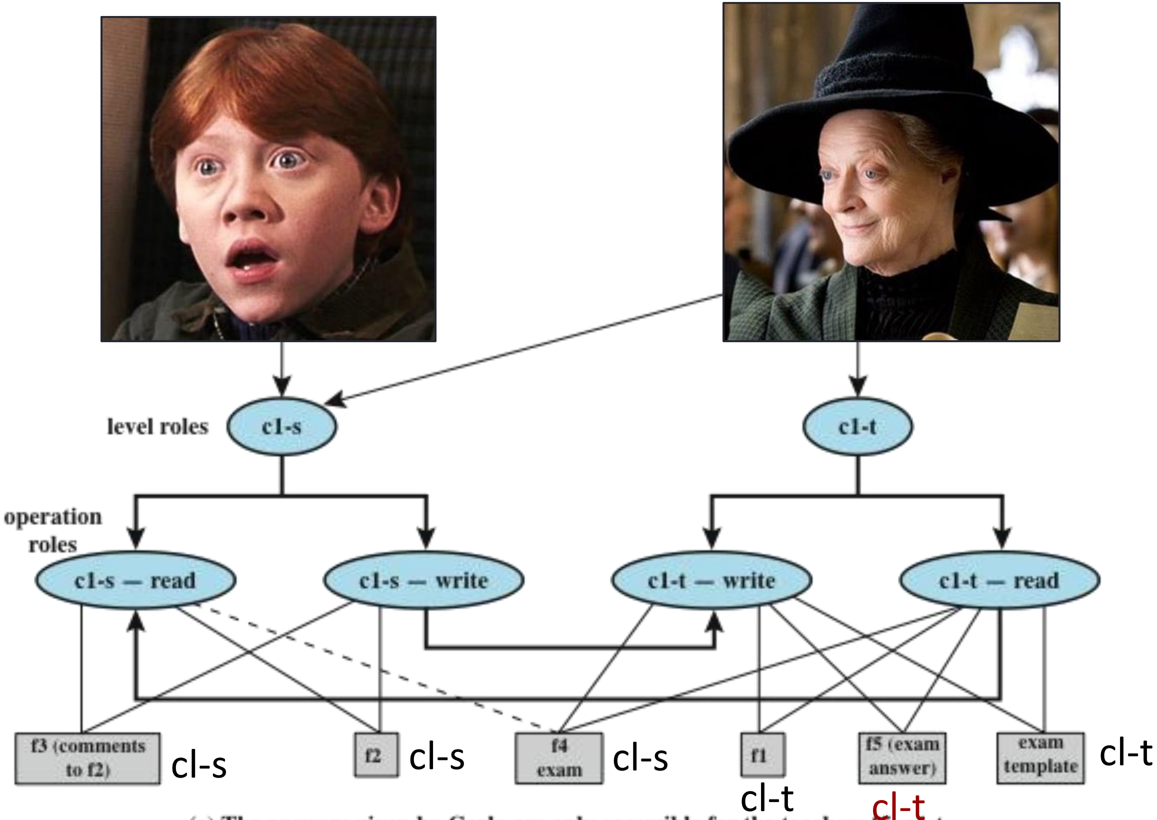


The student is permitted to access to the exam f4: cl-s

For evaluation, we need **secure transition rules also!**

s: student
t: teacher

ss-property
and *-property



(e) The answers given by Carla are only accessible for the teacher: f5: c1-t

The answers submitted by the student is only accessible for the teacher f5: cl-t

Limitations to the BLP Model

- BLP **does not address integrity issues**

- The *-property is **difficult to implement**
  - Inferences from ordinary actions of higher-level subjects (side channels)
  - Deliberate communications by higher-level subjects (covert channels)

- The BLP formalism **does not include de-classification protocols**.

Biba Integrity Model: Actions

- **Modify**: To write or update information in an object
- **Observe**: To read information in an object
- **Execute**: To execute an object
- **Invoke**: Communication from one subject to another

No Write Up and No Read Down

- **No write up**
  - A subject can only write into an object of lower or equal security level

- **No read down**
  - Subject can only read an object of higher or equal security level

Biba Integrity Model: Rules

- **Simple integrity:** A subject can modify an object only if the integrity level of the subject dominates the integrity level of the object: I(S) ≥ I(O).
- **Integrity confinement:** A subject can read an object only if the integrity level of the subject is dominated by the integrity level of the object: I(S) ≤ I(O).
- **Invocation property:** A subject can invoke another subject only if the integrity level of the first subject dominates the integrity level of the second subject:
  I(S1) ≥ I(S2).

Clark-Wilson Integrity Model

- More practical than prior models
  - Developed mainly for banks!
- Model commercial operations
  - **Well-formed transactions**
    - A user should not manipulate data arbitrarily
  - **Separation of duty among users**
    - A person who creates or certifies a well-formed transaction **is not allowed** to execute it

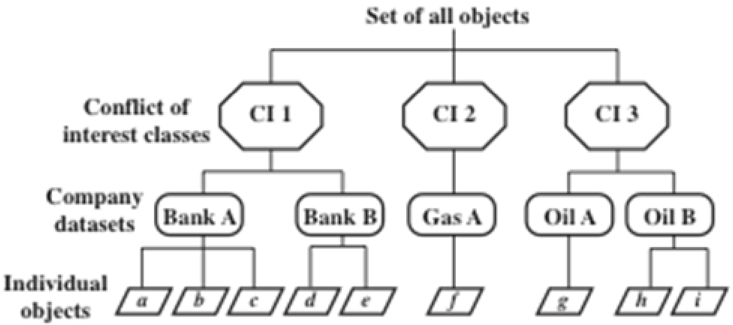
Clark-Wilson Concepts

- Constrained data items (CDIs)
  - Subject to strict integrity controls
- Unconstrained data items (UDIs)
  - Unchecked data items
- Integrity verification procedures (IVPs):
  - Intended to assure that all CDIs conform to some application-specific model of integrity and consistency
- Transformation procedures (TPs):
  - System transactions that change the set of CDIs from one consistent state to another

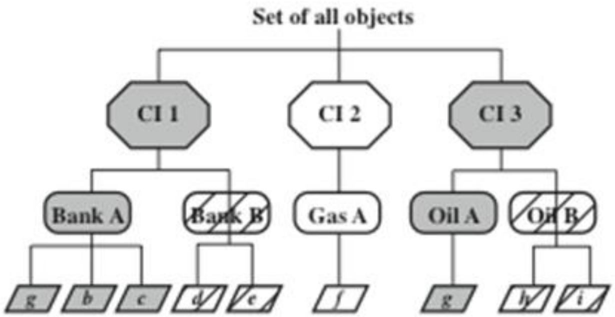
Chinese Wall Model

- Use discretionary and mandatory access to address integrity and confidentiality concerns
  - **Subjects**: Active entities that may wish to access protected objects
  - **Information**: Information organized into a hierarchy
    - **Objects**: Individual items of information, each concerning a single corporation
    - **Dataset** (DS): All objects that concern the same corporation
    - **Conflict of interest** (CI) **class**: All datasets whose corporations are in competition
  - **Access rules**: Rules for read and write access

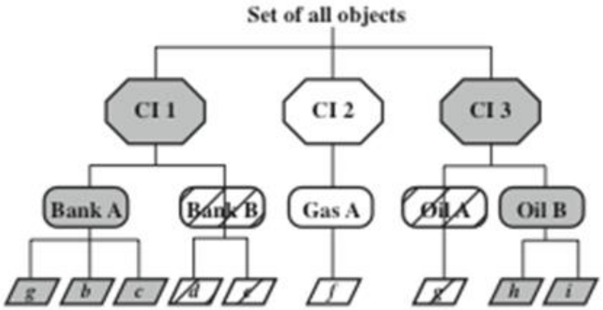
Chinese Wall Model



(a) Example set



(b) John has access to Bank A and Oil A



(c) Jane has access to Bank A and Oil B

sanitized data

Simple security rule: S can read O only if

- O is in the same DS as an object already accessed by S, **OR**
- O belongs to a CI from which S has not yet accessed any information

***-property rule:** S can write O only if

- S can read O according to the simple security rule, **AND**
- All objects that S can read are in the same DS as O.

Reading

Computer Security: Principles and Practice (2nd Edition), Stallings, Pearson HE, Inc. Chapter 13 Trusted Computing and Multilevel Security