Dr. Ammar Haider
Assistant Professor
School of Computing

CS3002 Information Security

Theoretical Models of Access Control

Reference: Stallings SPP chap 27

Why Theoretical Models

- Providing security in a large-scale system is extremely complex
  - Firstly, your product design should incorporate security
  - Secondly, the design implementation should be flawless
    - Too much room for making mistakes in both steps
- A theoretical model can aid in achieving fool proof security.
  - Our design and implementation can then be tested against that model

Multi-Level Security

- An environment where information or assets are classified at different levels of security
- For example in military, classifications could be:
  - Top secret (most sensitive)
  - Secret
  - Confidential
  - Restricted
  - Unclassified (least sensitive)

Bell LaPadula (BLP) Model

- Goal: prevent the unauthorized disclosure of information
  - ONLY deals with confidentiality of information flow
- Bell-LaPadula Model is basis for many, or most, of the other theoretical models
- Proposed in 1970s, in the era of mainframe computers

BLP Model

- A formal model for access control
- Objects (assets) are assigned a security classification
  - Form a hierarchy and are referred to as security levels
  - e.g. top secret > secret > confidential > unclassified
- Subjects have a security clearance
- Security classes control the manner by which a subject may access an object

BLP Access Privileges

read	Subject is allowed view-only access
append	Subject is allowed only write access (can't view)
write	Subject is allowed both view and write access
execute	Subject is allowed neither read nor write access, but may invoke the object for execution

BLP Properties (to enforce)

1) Simple security (ss) property:

- A subject can only read an object of less or equal security level
- No read up

2) * (star) property:

- A subject can only write into an object of greater or equal security level
- No write down

BLP ss-property Example

Security level	***Subject***	***Object***
Top Secret	Tamim	Personnel Files
Secret	Sohail	E-Mail Files
Confidential	Kaleem	Activity Logs
Unclassified	Jamal	Telephone Lists

- Tamim can read all files
- Kaleem cannot read Personnel or E-Mail Files
- Jamal can only read Telephone Lists

BLP – Reading Information

- "Reads up" disallowed, "reads down" allowed
- Simple Security Property:
  Subject s can read object o
  - iff $Level_s$ dominates $Level_o$
  - and s has permission to read o
    - Note: It combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no reads up" rule

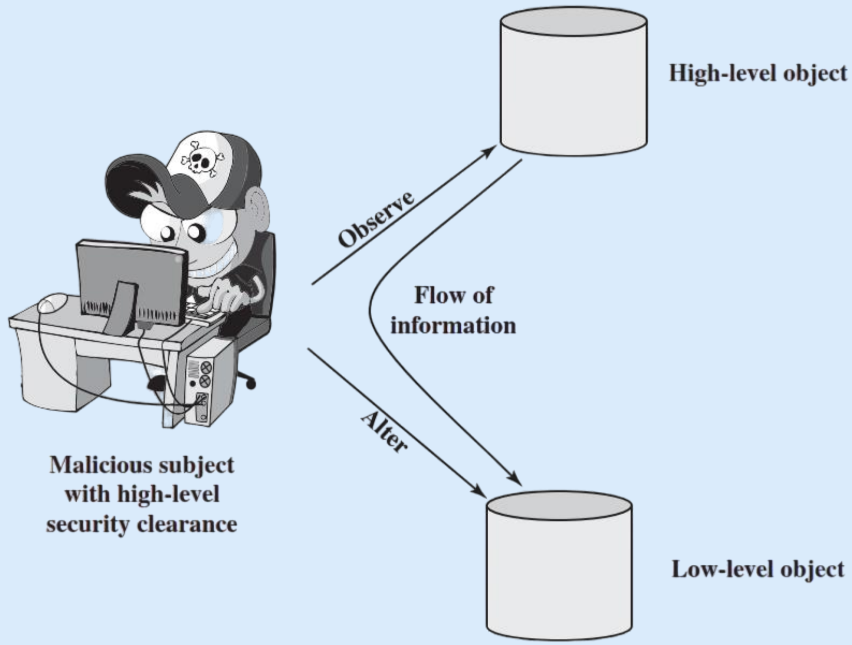
BLP – Writing Information

- Information flows up, not down
  - "Writes up" allowed, "writes down" disallowed
- *-property: Subject s can write object o
  - iff $Level_o$ dominates $Level_s$
  - and s has permission to write o
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no writes down" rule

BLP – Writing Information

Why enforce "no write-down" rule?

Because otherwise, there may be breach of confidentiality



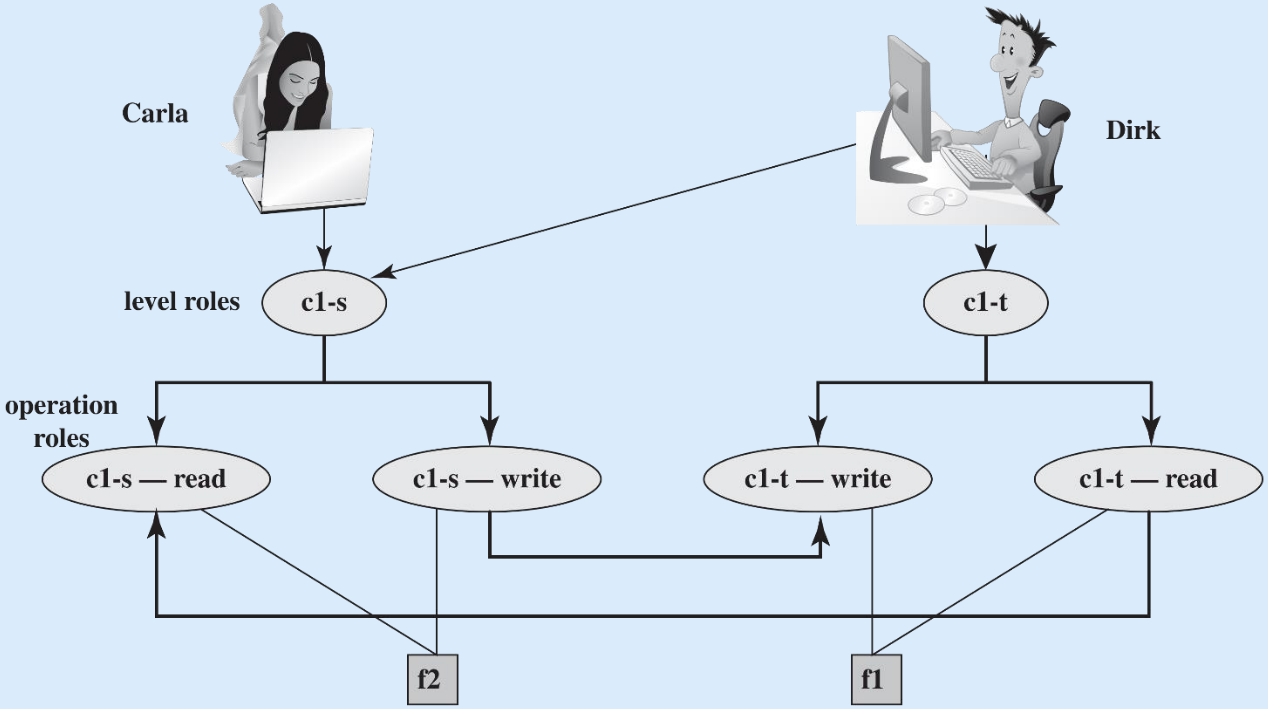
Subject at a high level should not be able to convey info to a subject at a lower level

BLP Example (1/6)

- A role-based access control system
- Two users: Carla (student) and Dirk (teacher) in a course c1
  - Carla (Class: c1-s)
  - Dirk (Class: c1-t)
    - can also login as a student (Class: c1-s)
- A student role has a lower security clearance
- A teacher role has a higher security clearance

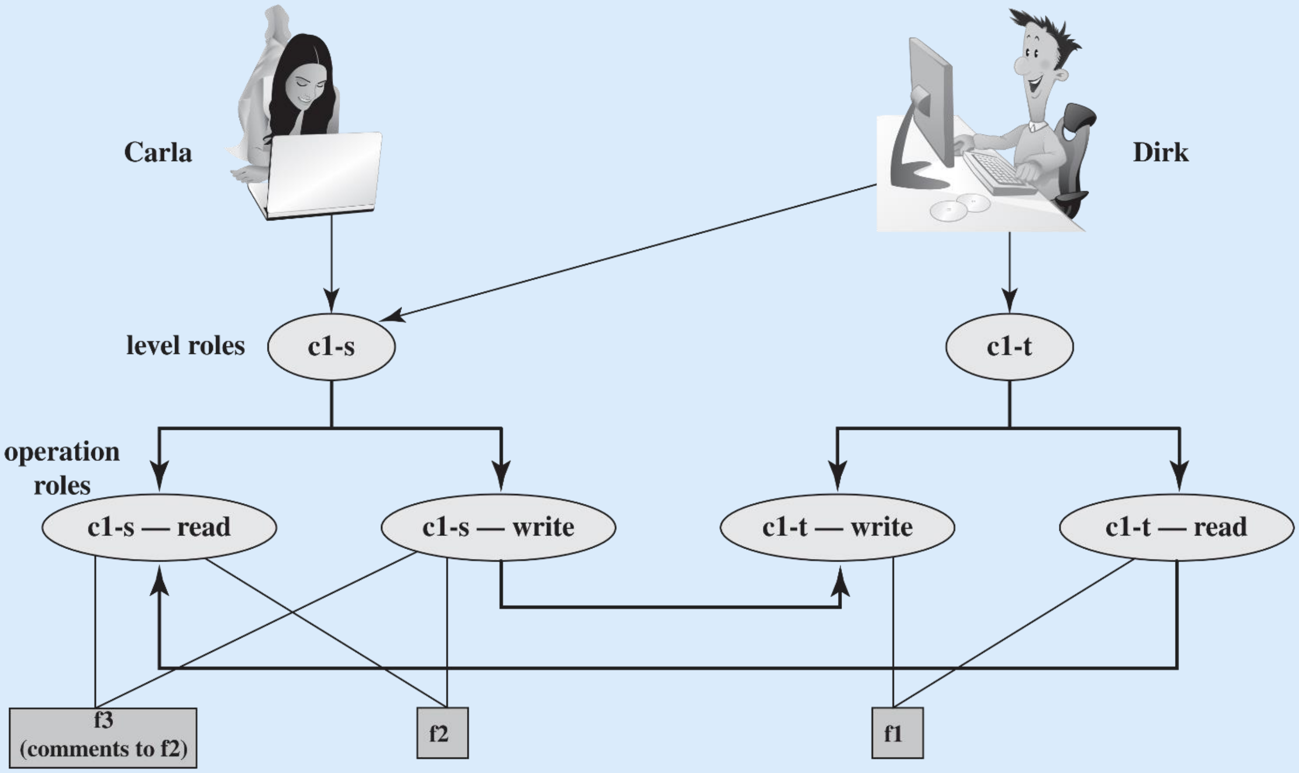
BLP Example (2/6)

- Dirk creates file f1 (e.g. personal notes)
- Carla creates file f2 (e.g. an assignment attempt)
- Carla can read/write to f2 but can not read f1
- Dirk can read/write f1 but f2 is read-only as teacher (if Carla permits)
- Dirk can write to f2 only as a student



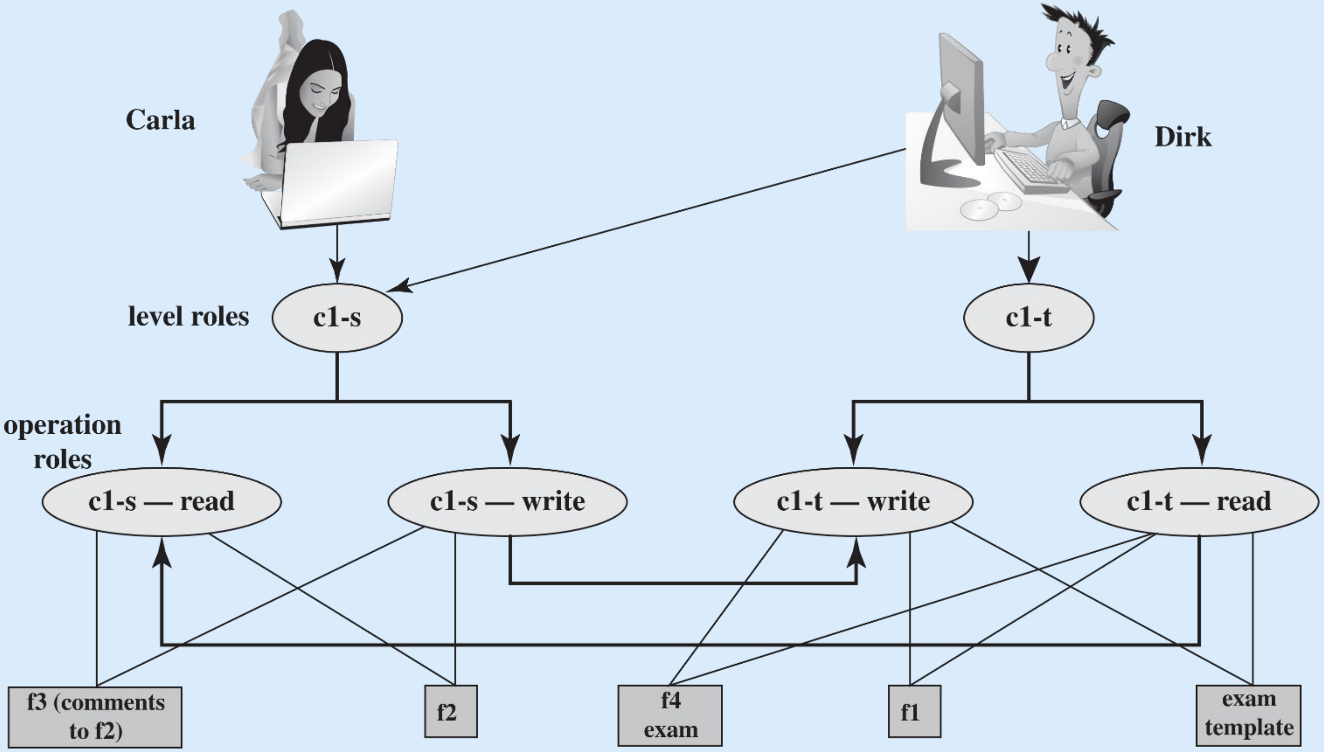
BLP Example (3/6)

- Dirk reads f2; want to create f3 (comments on Carla's submission)
- Dirk signs in as a student to create f3 (so that Carla can read)
  - As a teacher, Dirk cannot create a file at student classification



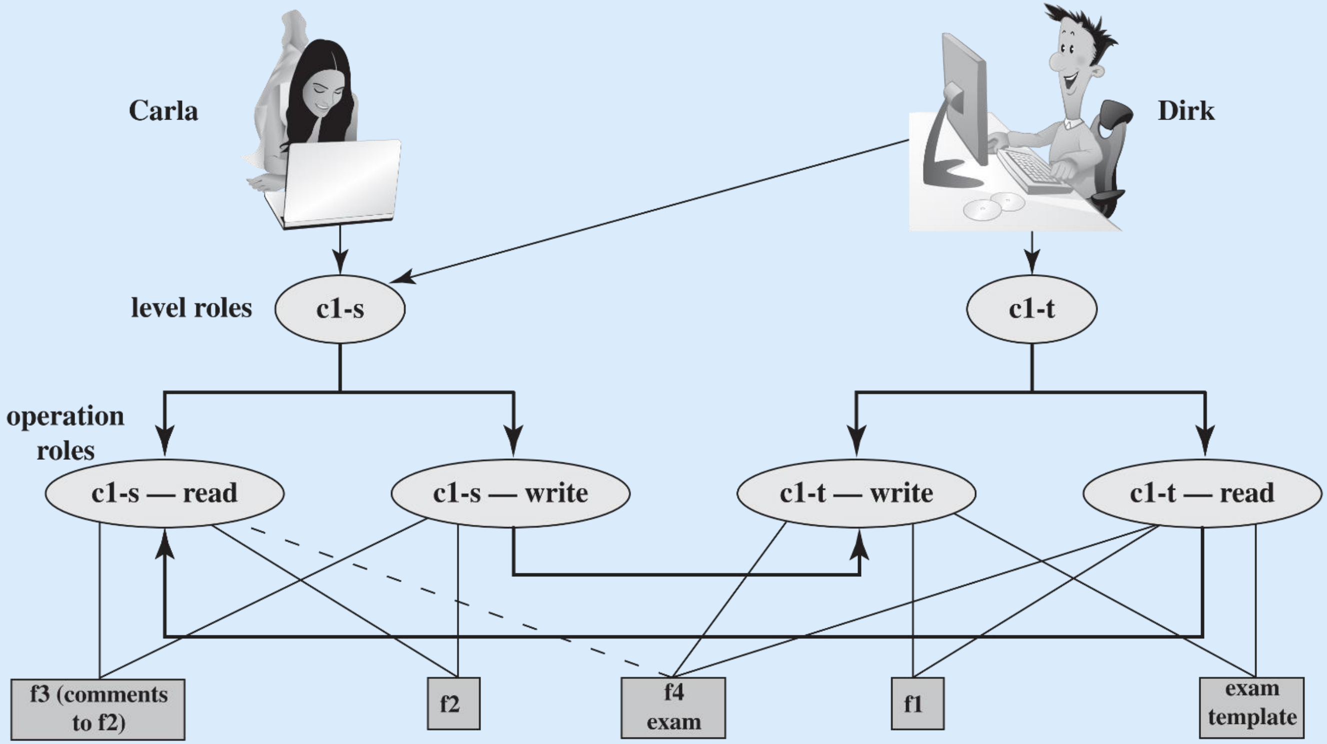
BLP Example (4/6)

- Dirk as a teacher creates an exam (f4)
- Must log in as a teacher to read exam template



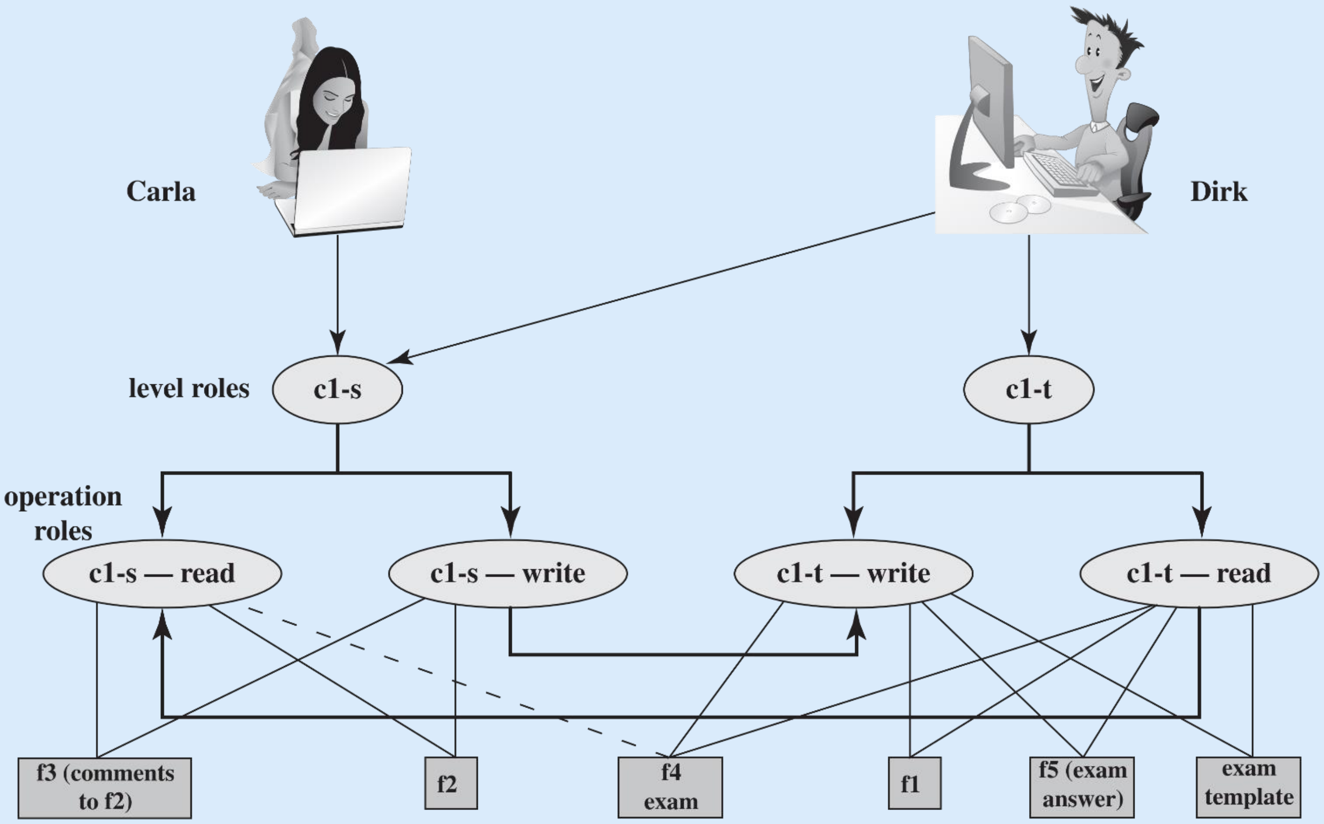
BLP Example (5/6)

- Dirk wants to give Carla access to read f4
- Dirk can't do that; an admin must do
  - An admin downgrades f4 class to c1-s



BLP Example (6/6)

- Carla writes answers to f5 at teacher level (c1-t)
  - An example of write up (allowed in BLP)
- Dirk can read f5



Limitations of BLP Model

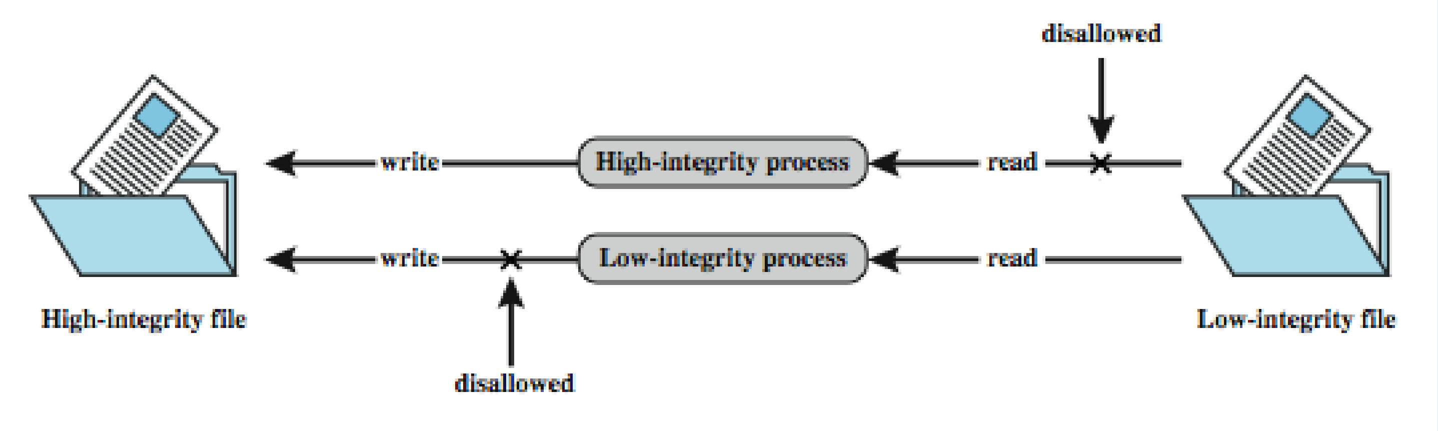
- Incompatibility of confidentiality and integrity
- Classification of data changes over time
  - BLP has no provision to manage the downgrade of objects
- In the presence of shared resources, *-property may not be enforced
- A bit complex to implement

Biba Integrity Model

- Another model from 1970s.
- Deals with information integrity only
  - the case where data must be visible at multiple security levels but should be modified in a controlled ways.
  - Prevent unauthorized modifications
- Like BLP, it also works with subjects and objects
- Each subject and object is assigned an integrity level
  - Denoted by I(S) and I(O)
  - Levels should be hierarchical
    - e.g. measure data accuracy as extreme > high > medium > low
- There are four access modes
  - modify, observe, execute, invoke
  - invoke means to communicate information b/w subjects

Biba Policies to enforce

- Strict integrity policy: Prevent low integrity data from **contaminating** the high integrity data
  - Simple integrity: modify only if $I(S) \geq I(O)$
    - i.e. can write down, but **no write up**
  - Integrity confinement: read only if $I(S) \leq I(O)$
    - i.e. can read up, but **no read down**
  - Invocation property: invoke/comm only if $I(S_1) \geq I(S_2)$



Clark-Wilson Integrity Model

- More practical integrity model from 1987
- Aimed at commercial world (instead of military)
- Goals
  - Prevent unauthorized users from making modifications
  - Prevent authorized users from making improper modifications
    - Biba model addressed the first goal only
- Two concepts
  - Well-formed transactions: a user can manipulate data in constrained ways only
  - Separation of duty: one can create a transaction but not execute it

Clark-Wilson Integrity Model

- Main components
  - Users
  - CDI: constrained data items (whose integrity should be preserved)
  - UDI: unconstrained items
  - IVPs: integrity verification procedures that assure all CDIs conform to integrity/consistency rules
  - TPs: transaction procedures that change CDIs
- Example: In a banking system, CDIs could be account balance, loan application, cheques etc. TPs are deposit, withdraw etc. One UDI could be user’s input at ATM pin keypad

- In CW model, a security officer needs to define **access triples** (subject, TP, objects). This ensures that even authorized users can not take malicious actions.

Certification & Enforcement of Rules

These rules ensure the integrity of data items.

‘Certification’ preformed by a security officer.

‘Enforcement’ done by the system

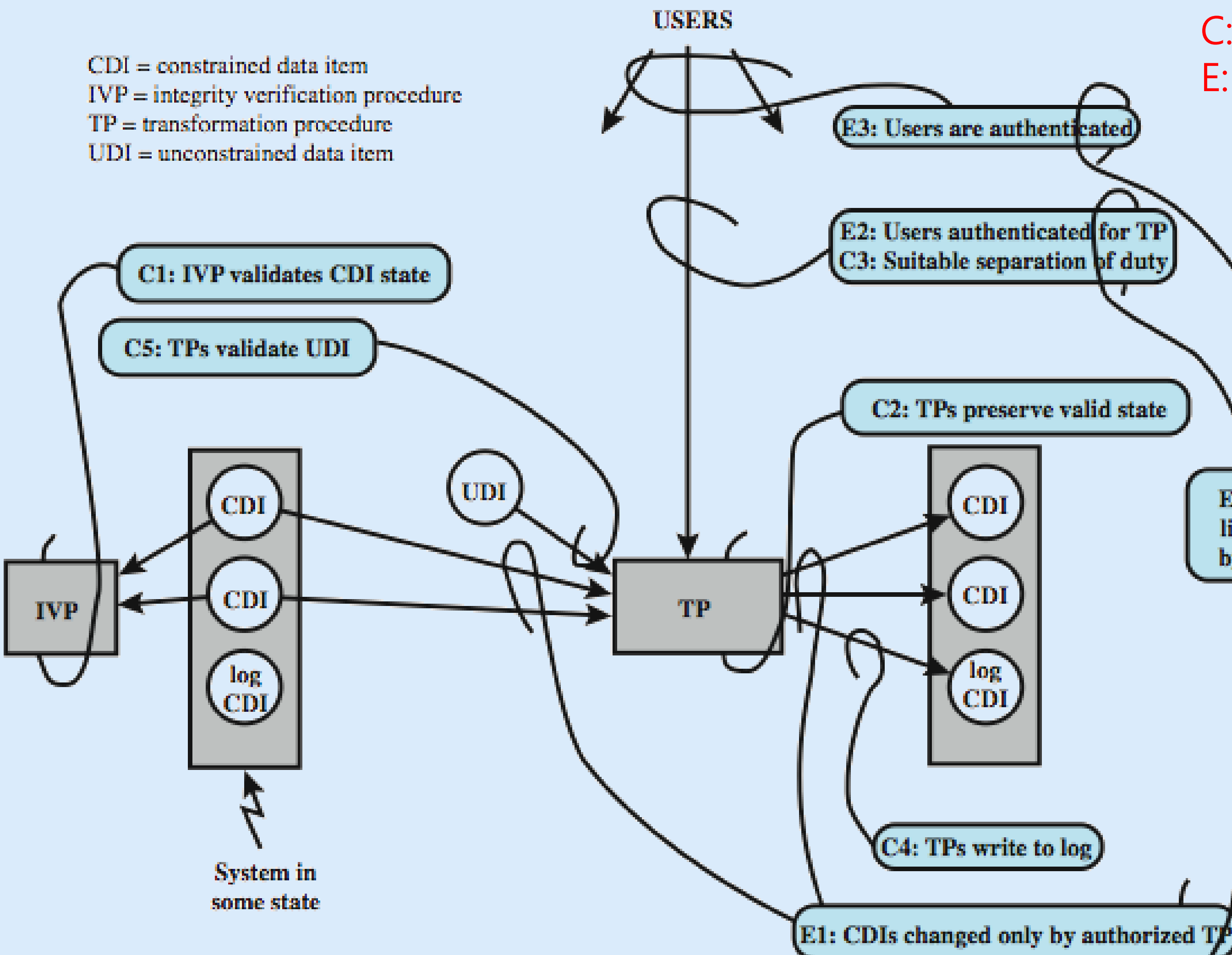
- C1: IVPs must ensure that all CDIs are in valid states
- C2: All TPs must be certified (must take a CDI from a valid initial state to a valid final state)
  - A certified TP looks like: TPi (CDIa, CDIb, CDIc, …)
- E1: The system must maintain a list of relations specified in C2
- E2: The system must maintain a list of allowed (User, TPi, (CDIa, CDIb, …)) combinations, i.e. access triples

Certification & Enforcement of Rules

- C3: The list of relations in E2 must be certified to meet separation of duties
- E3 The system must authenticate each user when executing a TP
- C4: All TPs must be certified to write their details to a log file
- C5: Any TP that takes UDI as in input value must be certified to perform only valid transactions
- E4: Only the agent permitted to certify entities is allowed to do so

Clark-Wilson Integrity Model

C: Certification rules
E: Enforcement rules



The Chinese Wall Model

- Also called Brewer and Nash model.
- Not a general purpose security model, it is meant for a very specific commercial concern
- Addresses the conflict of interest (CI or CoI) problem.
- Think law firms, accounting firms, and other consultancy services, providing services to several of clients. Some of those clients might be competitor to each other.

The Chinese Wall Model

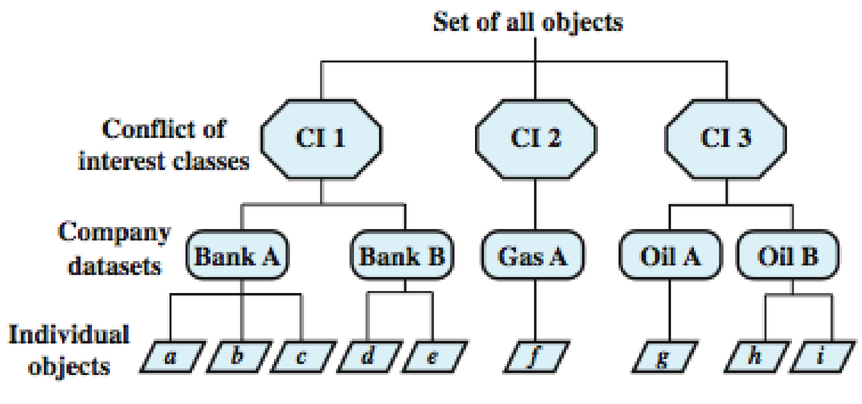
Model elements

- subjects: active entities interested in accessing protected objects
- information
  - objects: individual data items, each about a corporation
  - datasets (DS): all objects concerning one corporation
  - CI class: datasets whose corporation are in competition (conflict of interest or CI)
- access rules: rules for reading/writing data

The Chinese Wall Model

- Not a true multilevel security model
  - the history of a subject's access determines access control
- Subjects are only allowed access to info that is not held to conflict with any other info they already possess
- Once a subject accesses info from one dataset, a wall is set up to protect info in other datasets in the same CI

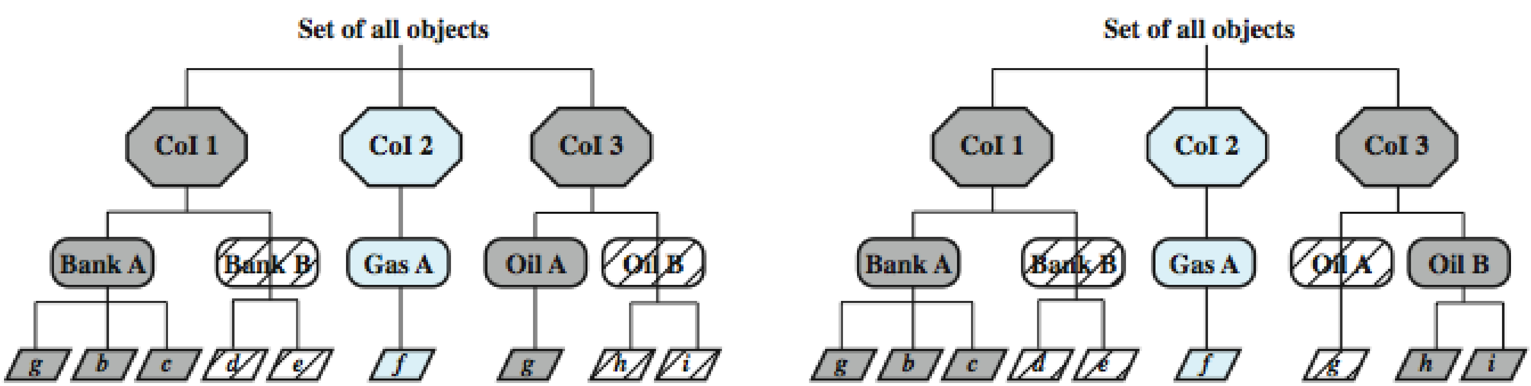
The Chinese Wall Model



(a) Example set

simple security rule (read): S can read O 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 info

***-property rule (write):** S can write O only if S can read O **AND** all objects that S can read are in the same DS as O.



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

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

Question: what can John write to?
what about Jane?

Chinese Wall vs BLP

- CW is based on access history
  BLP is history-less

- BLP can capture CW state at any time, but cannot track changes over time
  – BLP security levels would need to be updated each time an access is allowed