Information Security

CS3002

Lecture 23

30th November 2023

Dr. Rana Asif Rehman

Email: r.asif@lhr.nu.edu.pk

Security Models

Confidentiality Policy

- Goal: prevent the unauthorized disclosure of information
  - Deals with information flow
  - Integrity incidental

- Multi-level security models are best-known examples
  - Bell-LaPadula Model basis for many, or most, of these

Bell-LaPadula (BLP) Model

- Security levels arranged in linear ordering
  - Top Secret: highest
  - Secret
  - Confidential
  - Unclassified: lowest
- Levels consist of security clearance L(s)
- Objects have security classification L(o)

Bell-LaPadula (BLP) Model

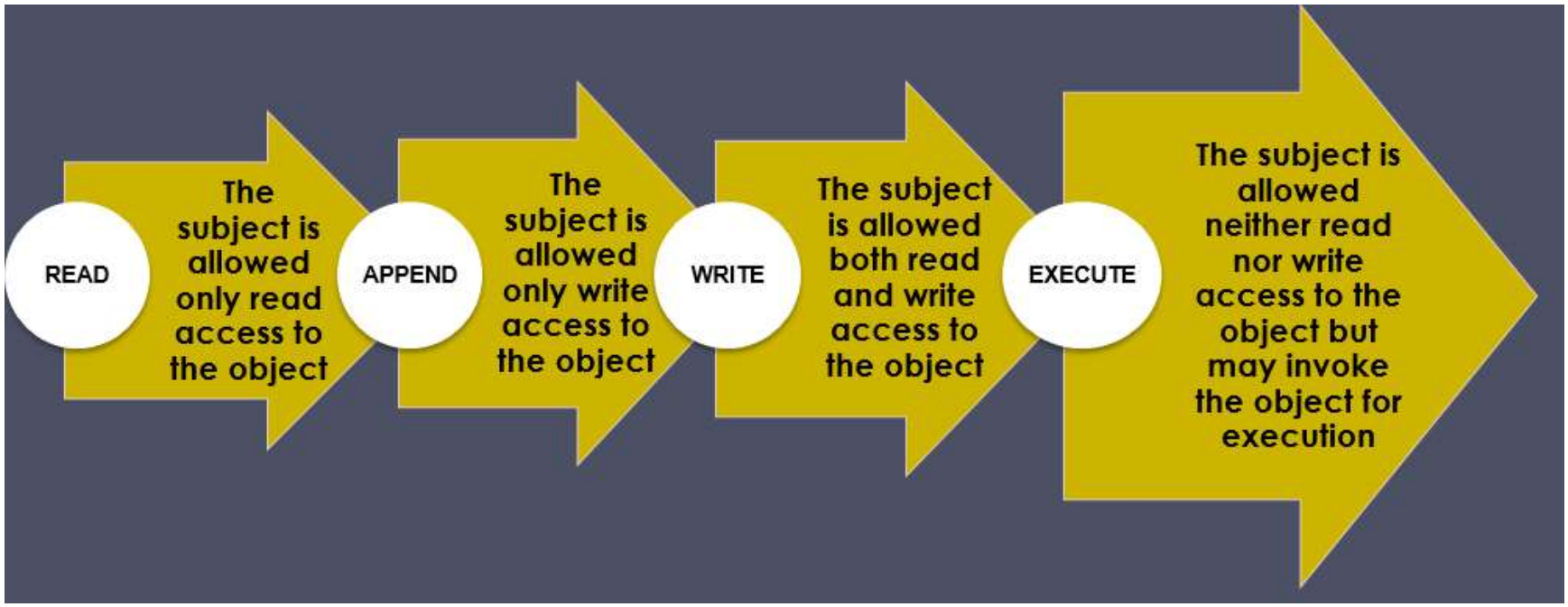
- Formal model for access control
- Subjects and objects are assigned a security class
- Form a hierarchy and are referred to as security levels
- A subject has a security **clearance**
- An object has a security **classification**
- Security classes control the manner by which a subject may access an object

A BLP Example

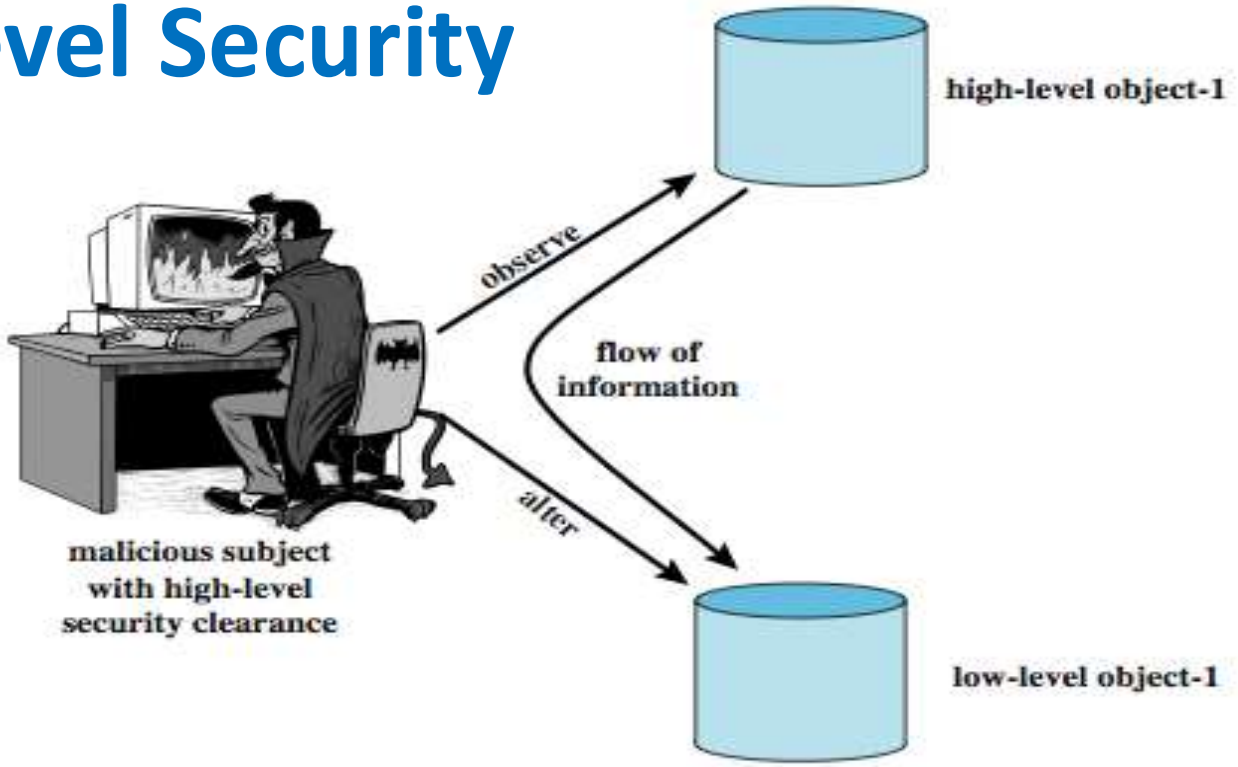
Security level	***Subject***	***Object***
Top Secret	Tamim	Personnel Files
Secret	Sohail	E-Mail Files
Confidential	Kaleem	Activity Logs
Unclassified	Jamal	Telephone Lists

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

Access Privileges



Multilevel Security



- Multiple levels of security and data
- Subject at a high level may not convey info to a subject at a non-comparable level:
  - No read up (ss-property): a subj can only read an obj of less or equal sec level
  - No write down (*-property): a subj can only write into an obj of greater or equal sec level

BLP Formal Description

- Based on current state of system (*b*, *M*, *f*, *H*):
  - Current access set *b (subj, objs, access-mode);* it is the ***current*** access (not permanent)
  - Access matrix *M* (Si is permitted to acces Oj)
  - Level function *f: assigns sec level to each subj and obj;* a subject may operate at that or lower level
  - Hierarchy *H: a directed tree whose nodes are objs:*
    - *Sec level of an obj must dominate (must be greater than) its parents*

BLP Properties

- Three BLP properties: *(c = current)*
  1. ss-property: (S_i, O_j, read) has $f_c(S_i) \geq f_o(O_j)$
  2. *-property: $(S_i, O_j, \text{append})$ has $f_c(S_i) \leq f_o(O_j)$ and
     (S_i, O_j, write) has $f_c(S_i) = f_o(O_j)$
  3. ds-property: (S_i, O_j, A_x) implies $A_x \in M[S_i, O_j]$

- BLP give formal theorems
  - Theoretically possible to prove system is secure

ss-property: simple security

*-property: pronounced star

ds-property: discretionary security

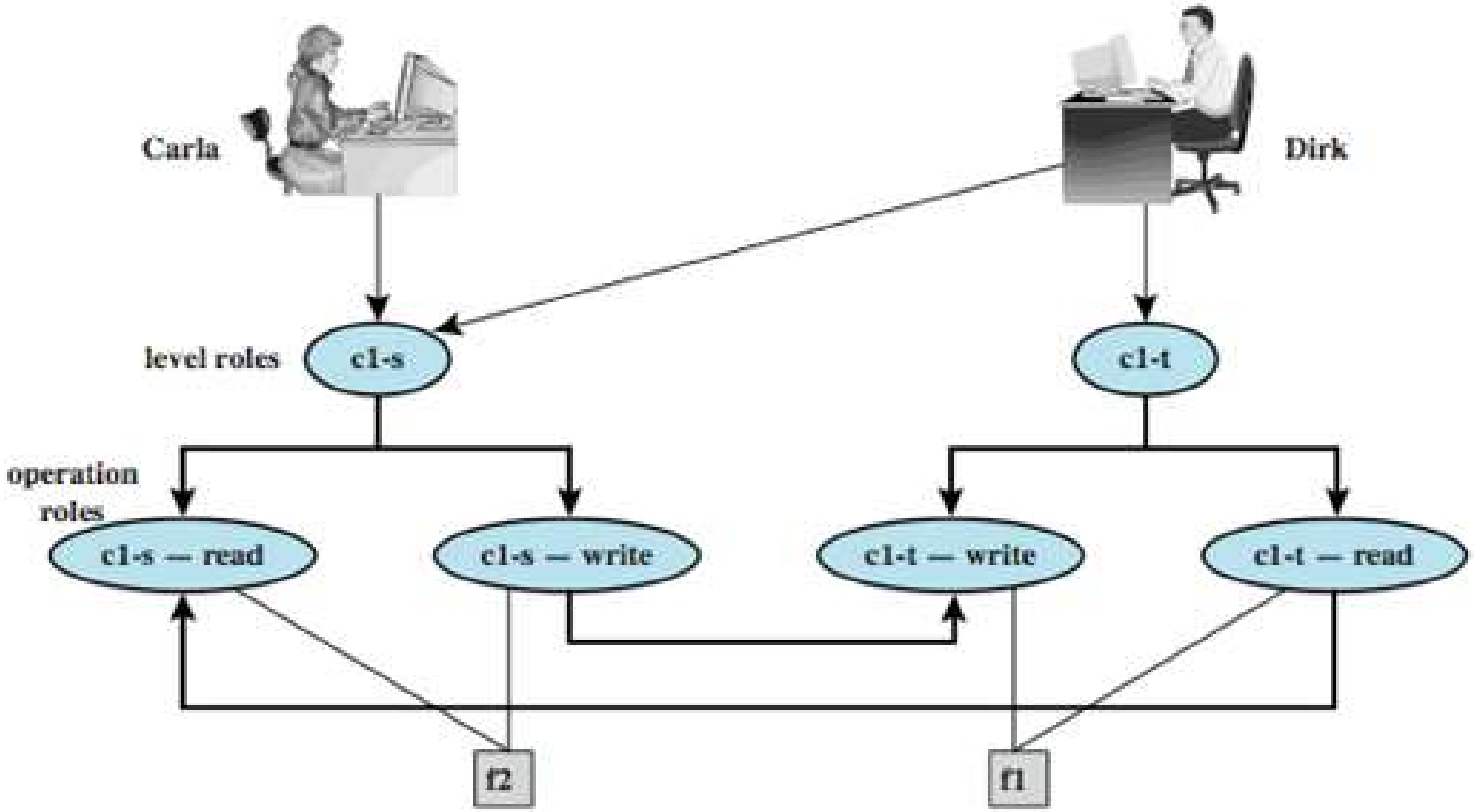
BLP Operations

1. **get access**: add (subj, obj, access-mode) to b
   - used by a subj to initiate an access to an object
2. **release access**: remove (subj, obj, access-mode)
3. **change object level**
4. **change current l**evel (subject)
5. **give access permission**: Add an access mode to M
   - used by a subj to grant access to on an obj
6. **rescind access permission**: reverse of 5
7. **create an object**
8. **delete a group of objects**

BLP Example

- A role-based access control system
- Two users: Carla (student) and Dirk (teacher)
  - Carla (Class: s)
  - Dirk (Class: T); can also login as a students thus (Class: s)
- A student role has a lower security clearance
- A teacher role has a higher security clearance

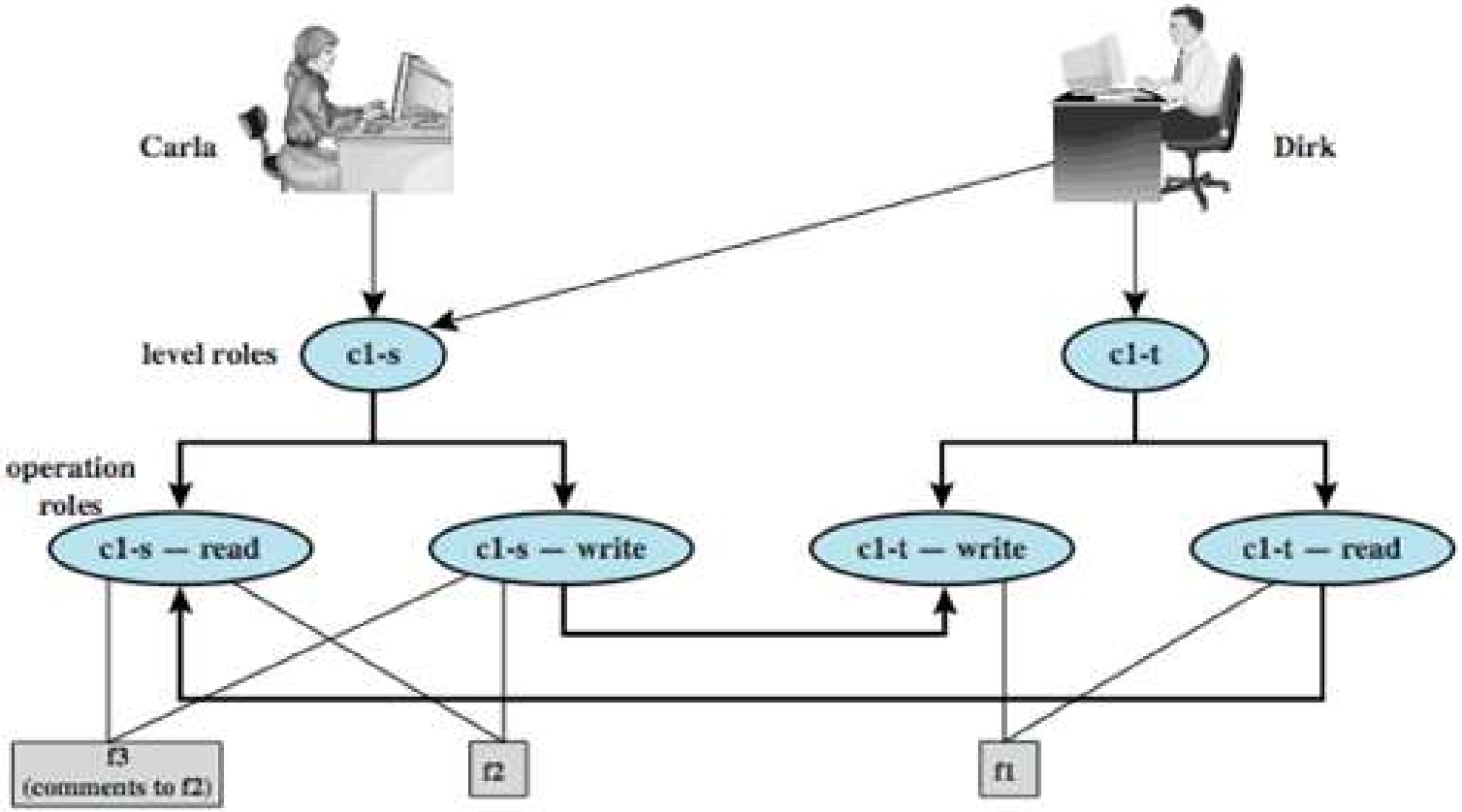
BLP Example



(a) Two new files are created: f1: c1-t; f2: c1-s

- Dirk creates f1; Carla creates f2
- Carla can read/write to f2 but cant read f1
- Dirk can read/write f1 and f2 (if perm)
- Dirk can read/write f2 only as a student

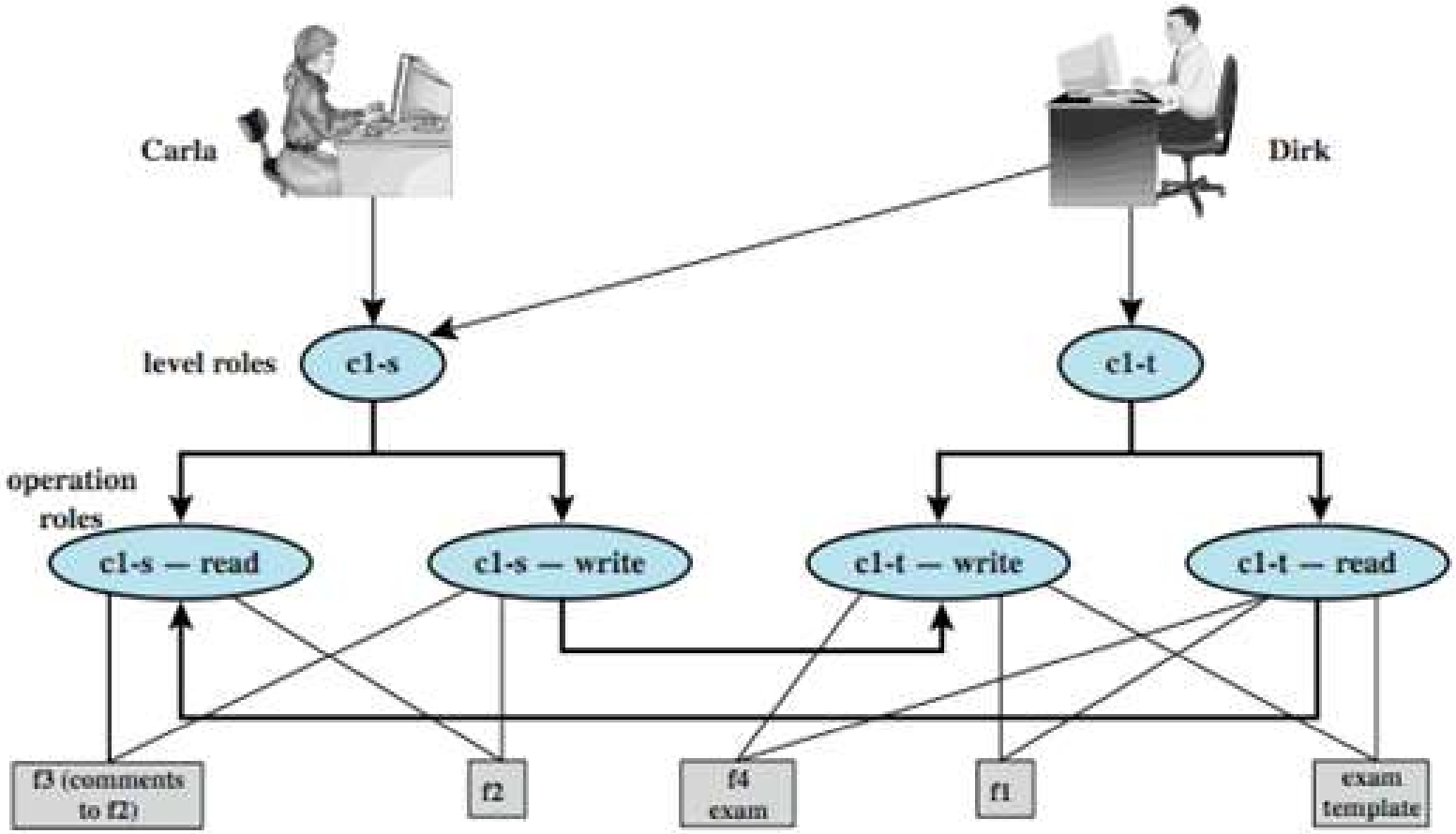
BLP Example cont.



(b) A third file is added: f3: c1-s

- Dirk reads f2; want to create f3 (comments)
- Dirk signs in as a stu (so Carla can read)
- As a teacher, Dirk cannot create a file at stu classification

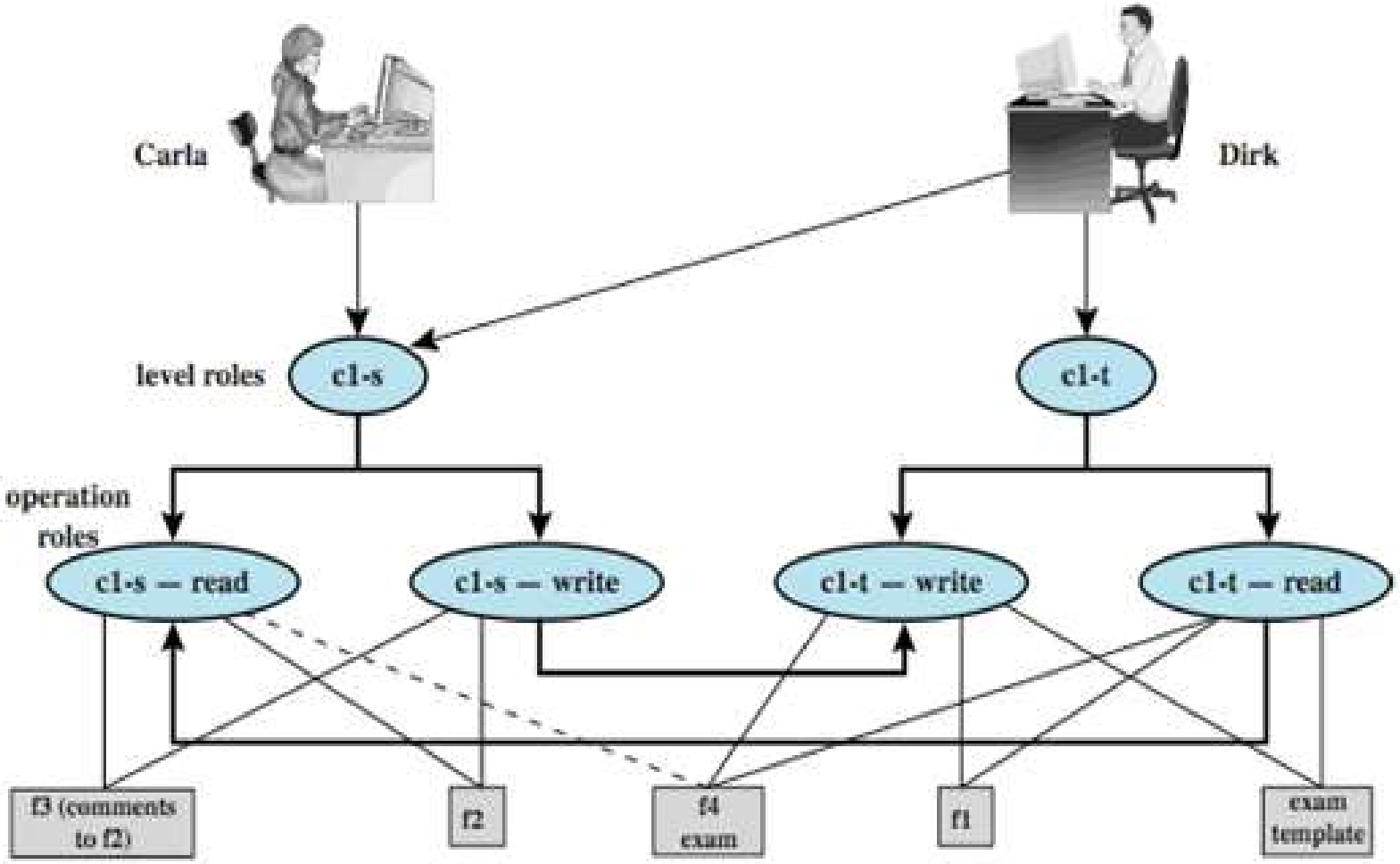
BLP Example cont.



(c) An exam is created based on an existing template: f4: c1-t

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

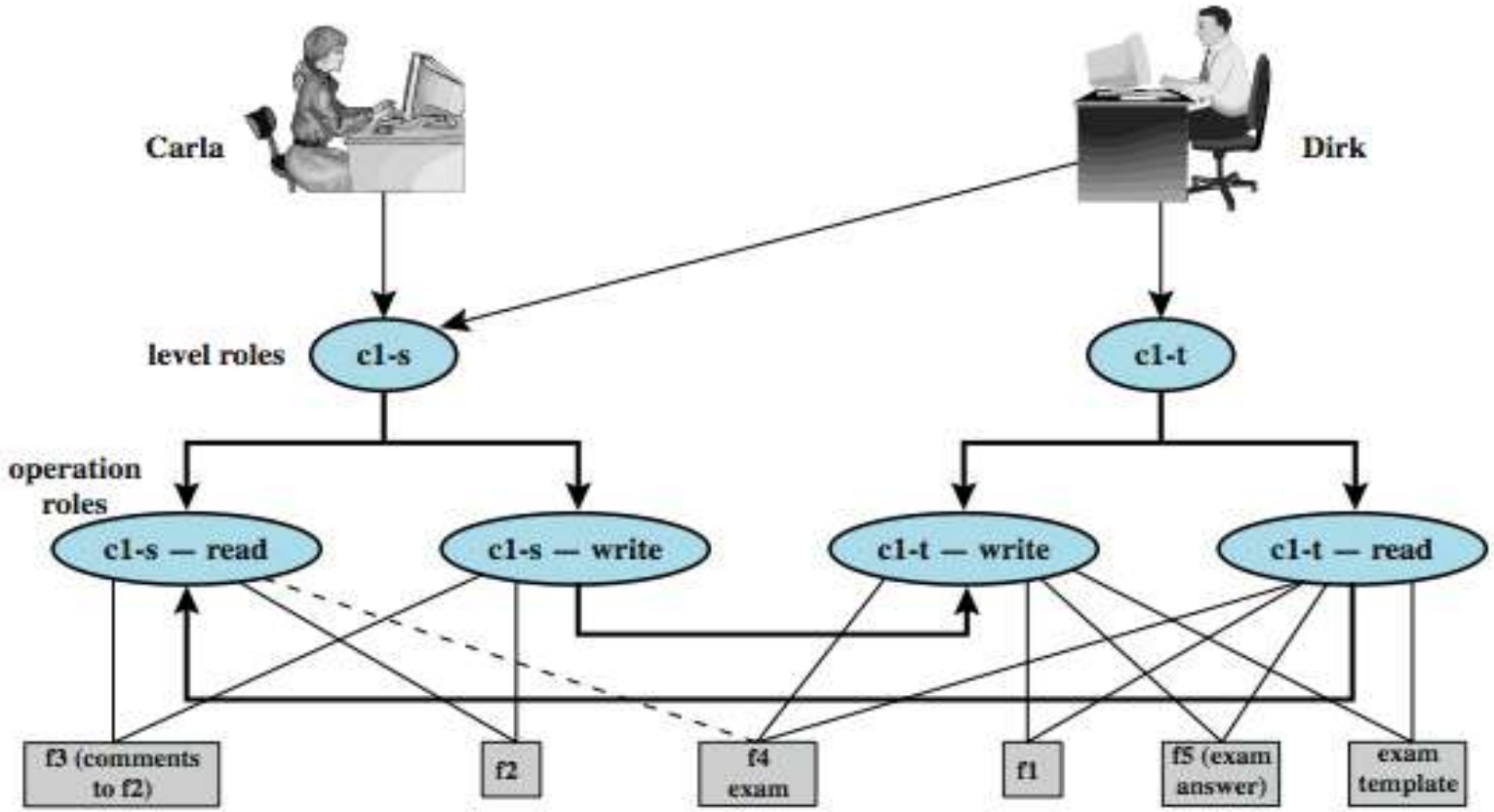
BLP Example cont.



(d) Carla, as student, is permitted acess to the exam: f4: c1-s

- 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 cont.



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

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

Reading Information - New

- "Reads up" disallowed, "reads down" allowed

- Simple Security Condition
  - Subject *s* can read object *o* iff *L*(*s*) *dom L*(*o*) and *s* has permission to read *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no reads up" rule

Writing Information - New

- Information flows up, not down
  - "Writes up" allowed, "writes down" disallowed

- *-Property (Step 2)
  - Subject *s* can write object *o* iff *L*(*o*) *dom L*(*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

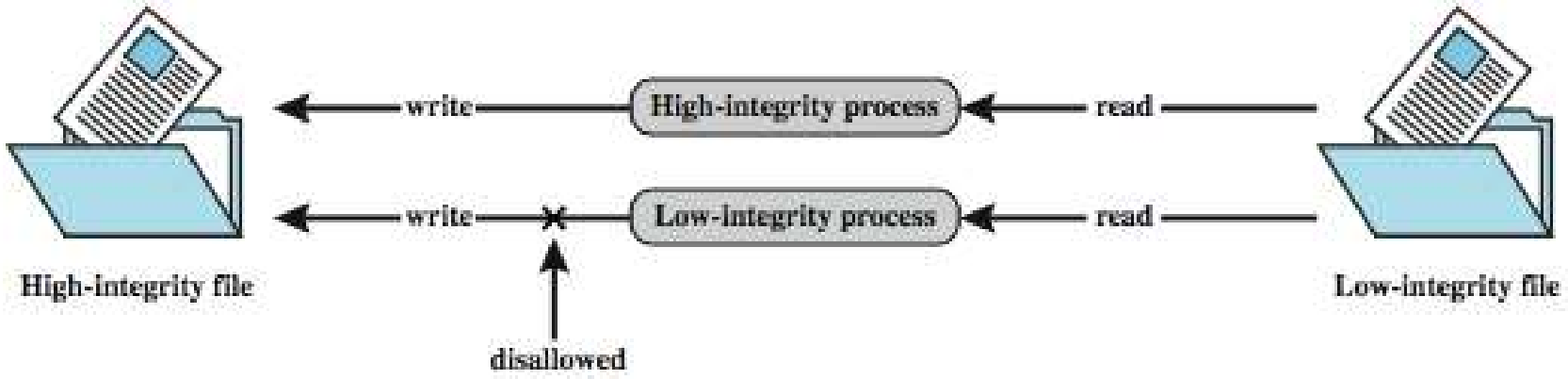
Limitation of BLP model

- Incompatibility of confidentiality and integrity
- Classification of data changes over time
- If data needs to migrate to higher security classification, a trusted user has to be downgraded!
- In the presence of shared resources, *-property may not be enforced
- A bit complex to implement

Biba Model

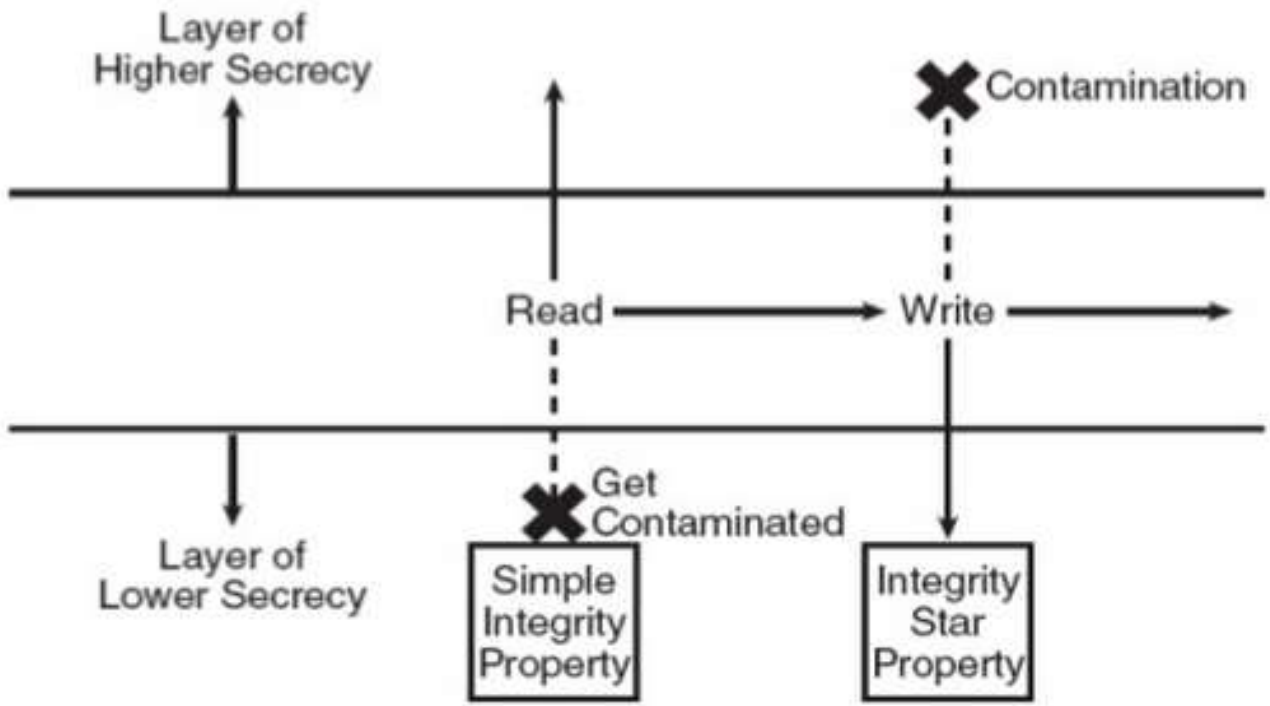
Biba Integrity Model

- Deals with integrity and deal with the case where data must be visible at multiple security levels but should be modified in a controlled ways.
- Strict integrity policy:
  - Simple integrity: *modify only if* $I(S) \geq I(O)$
  - Integrity confinement: *read only if* $I(S) \leq I(O)$
  - Invocation property: *invoke/comm only if* $I(S_1) \geq I(S_2)$



Biba Integrity Model

- Simple integrity: *modify only if* I(*S*) ≥ I(*O*)
- Integrity confinement: *read only if* I(*S*) ≤ I(*O*)
- Invocation property: *invoke/comm only if* I(S_1) ≥ I(S_2)



Clark-Wilson Integrity Model (Self-Study)

- Two concepts
  - Well-formed transactions: a user can manipulate data in constrained ways
  - Separation of duty: one can create a transaction but not execute it

- CDI: constrained data items (loan app; checks)
- UDI: unconstrained items
- IVPs: procedures that assure all CDIs conform to integrity/consistency rules
- TPs: transactions that change CDIs
- Very practical; used in commercial world

Certified and Enforcement Rules

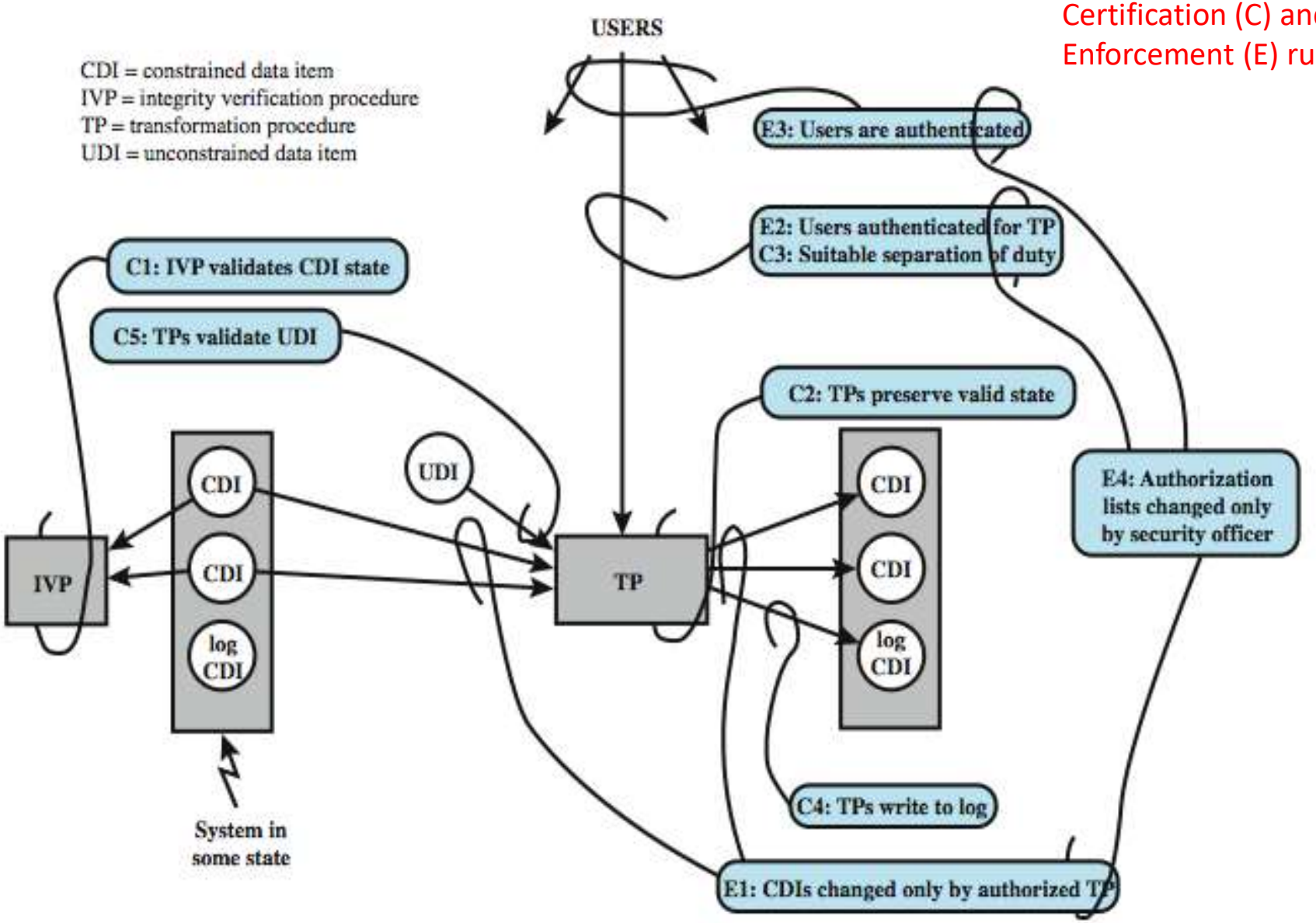
- C1: IVPs must ensure that all CDIs are in valid states
- C2: All TPs must be certified (must take a CDI from a valid state to a valid final state)
  - (Tpi, CDIa, CDIb, CDIc, …)
- E1: The system must maintain a list of relations specified in C2
- E2: The system must maintain a list of (User, Tpi, (CDIa, CDIb, …))

Certified and Enforcement 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
- C5: Any TP that takes UDI as in input value must be certified to perform valid transaction
- E4: Only the agent permitted to certify entitles is allowed to do so

Clark-Wilson Integrity Model

Certification (C) and Enforcement (E) rules



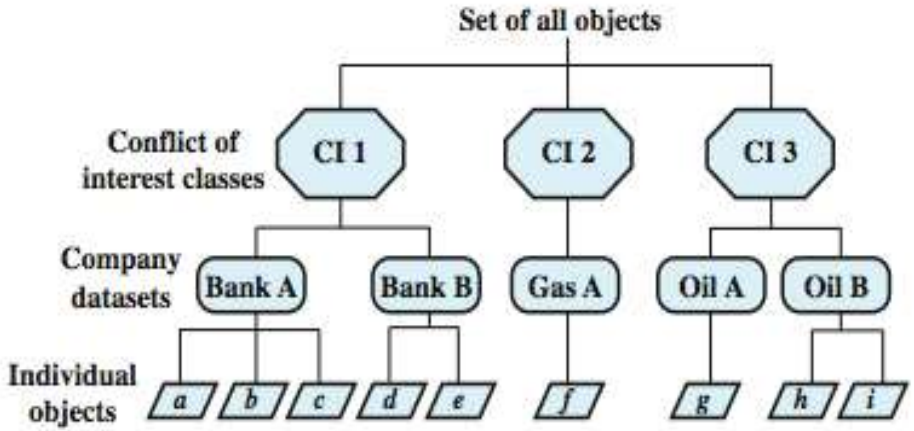
The Chinese Wall Model

- Hybrid model: addresses integrity and confidentiality
- Addresses conflict of interest (CI or CoI)
- Model elements
  - **subjects**: active entities interested in accessing protected objects
  - **information**
    - objects: individual data items, each about a corp
    - datasets (DS): all objects concerning one corp
    - CI class: datasets whose corp are in competition (conflict of interest or CI)
  - **access rules**: rules for reading/writing data

The Chinese Wall Model

- Not a true multilevel secure 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

Chinese Wall Model

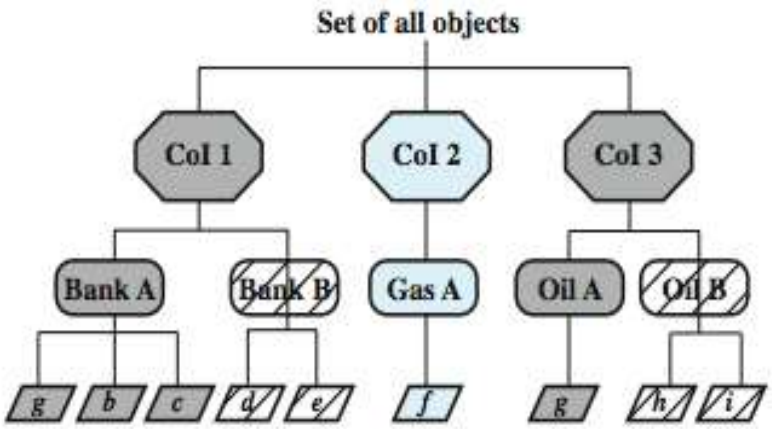


(a) Example set

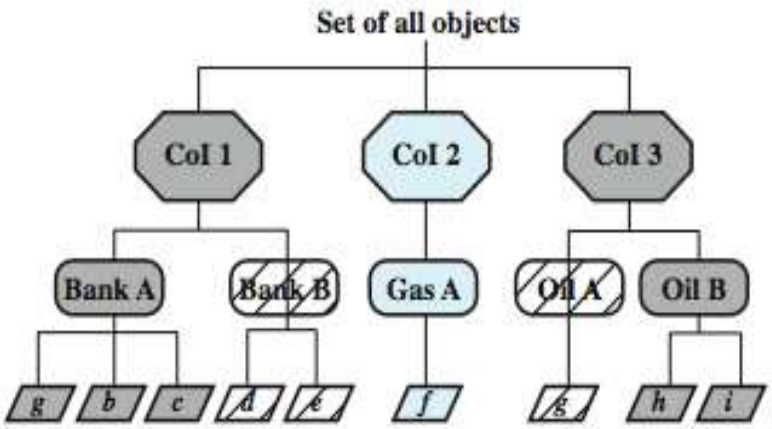
Simple sec rule (read): S can read O if O is in the same DS as an object already accessed by S OR O belongs to a CoI from which S has not yet accessed any info

***-property (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.

Question: what can John or Jane write to?



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



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

Compare CW to Bell-LaPadula

- 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