Computer Security: Principles and Practice

Chapter 13 – Trusted Computing and Multilevel Security

Trusted Computing and Multilevel Security

- present some interrelated topics:
  - formal models for computer security
  - multilevel security
  - trusted systems

Formal Models for Computer Security

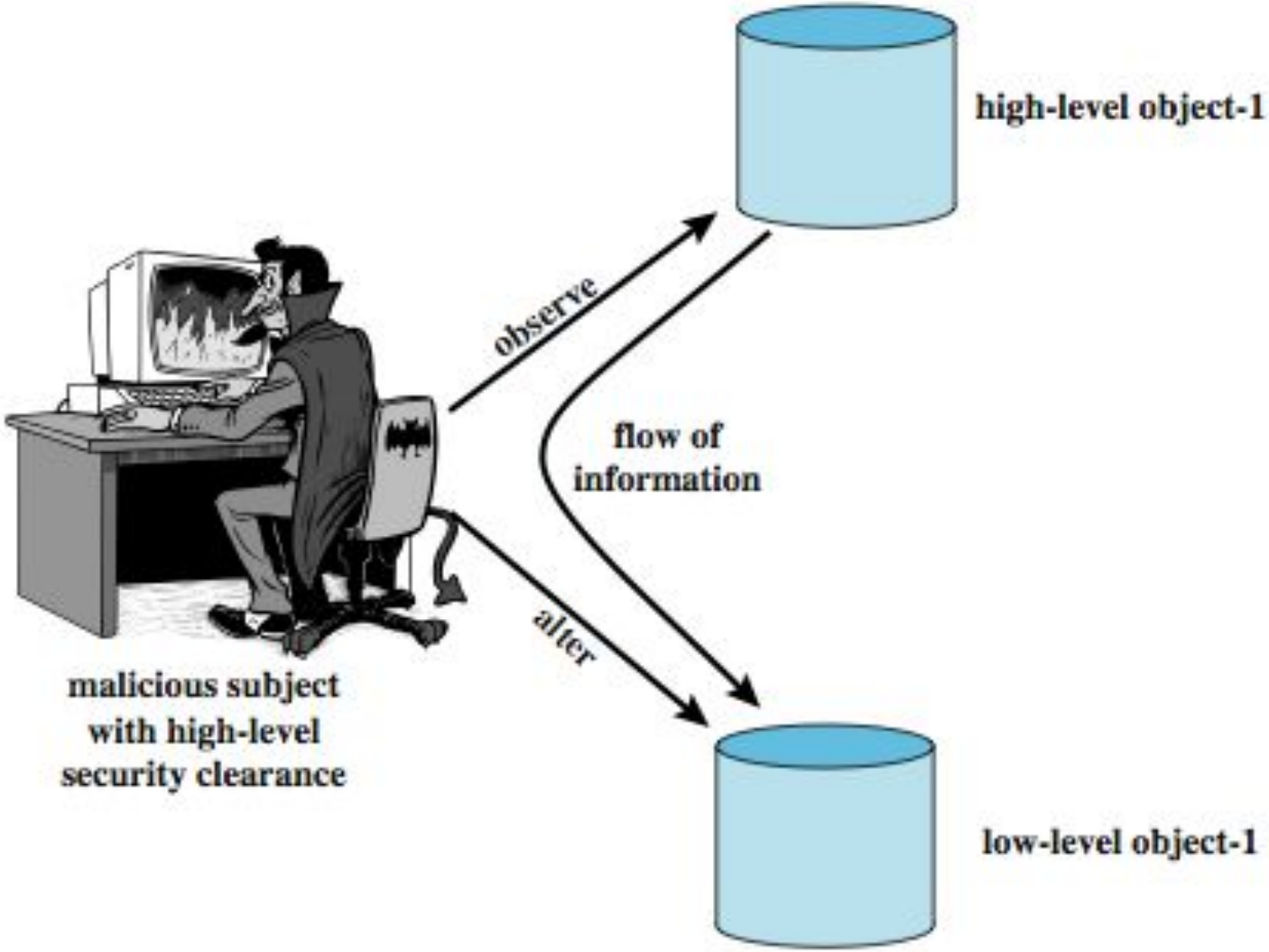
- two fundamental computer security **facts**:
  - all complex software systems have flaw/bugs
  - is extraordinarily difficult to build computer hardware/software not vulnerable to attack
- hence **desire** to prove design and implementation satisfy security requirements
- led to development of **formal security models**
  - initially funded by US Department of Defense

    Bell-LaPadula (BLP) model very influential

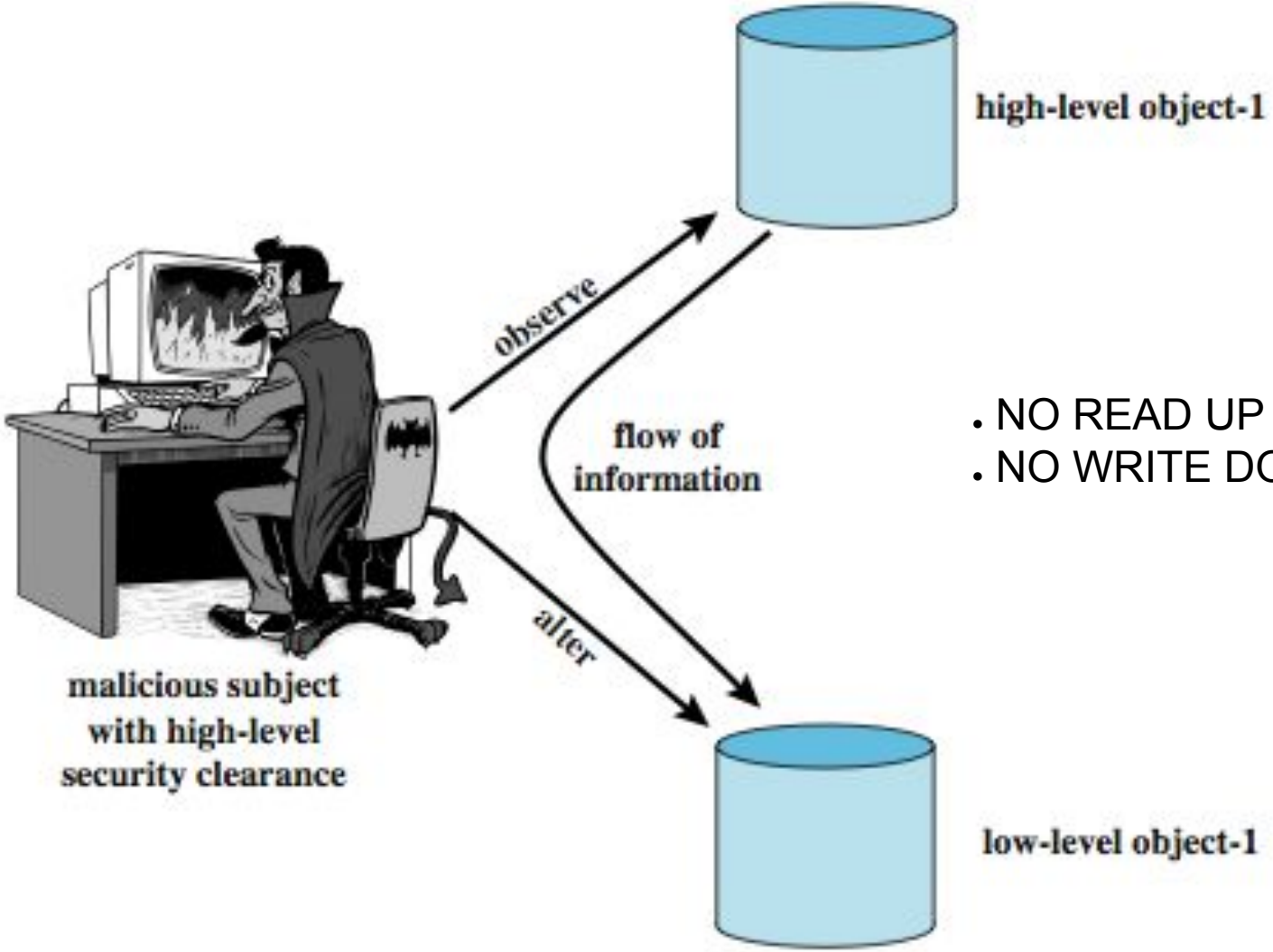
Bell-LaPadula (BLP) Model

- developed in 1970s
- as a formal access control model
- subjects and objects have a security class
  - *top secret* > *secret* > *confidential* > *unclassified*
    e.g., in military environment
  - **subject** has a security clearance level
  - **object** has a security classification level
  - classes control how subject may access an object
- applicable if have info and user categories
- there are 4 different modes: read, append, write, execute
- mostly focus on confidentiality, rather than integrity

Multi-Level Security



Multi-Level Security



- NO READ UP
- NO WRITE DOWN

BLP Formal Description

- based on current state of system (b, M, f, H):

  (current access set *b,* access matrix *M,* level function *f,* hierarchy *H)*

- three BLP properties:

  ss-property: (S_i, O_j, read) has $f_c(S_i) \geq f_o(O_j)$.

  *-property: (S_i, O_j, append) has $f_c(S_i) \leq f_o(O_j)$ and
  (S_i, O_j, write) has $f_c(S_i) = f_o(O_j)$

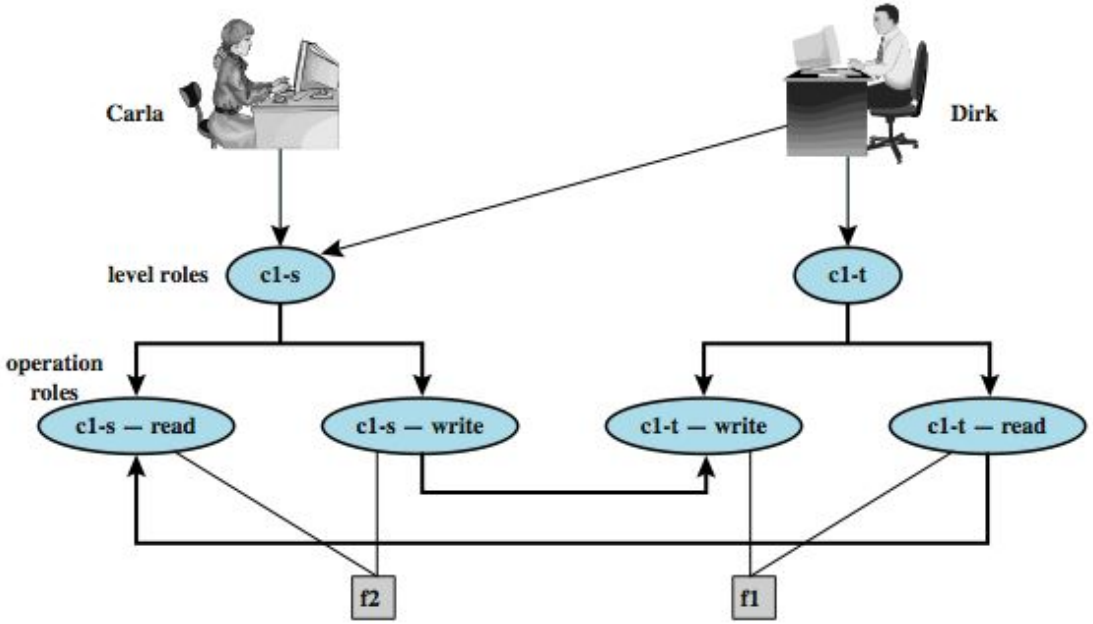
  ds-property: (S_i, O_j, A_x) implies $A_x \in M[S_iO_j]$

- BLP give formal theorems
  - theoretically possible to prove system is secure
  - in practice usually not possible

BLP Rules

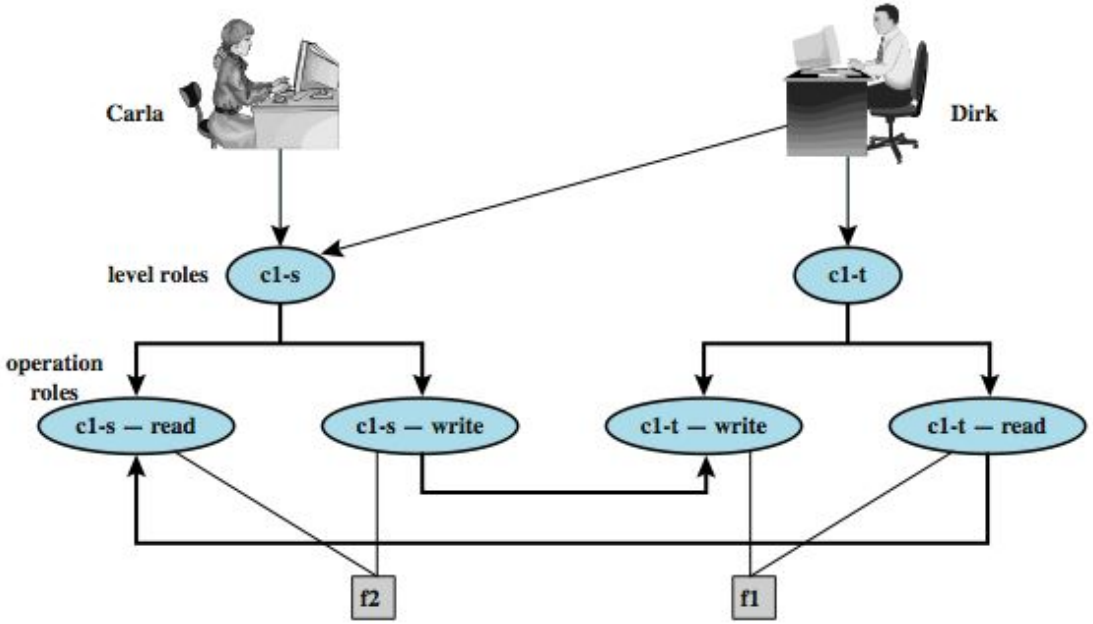
1. get access
2. release access
3. change object level
4. change subject level
5. give access permission
6. rescind access permission
7. create an object
8. delete a group of objects

BLP Example



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

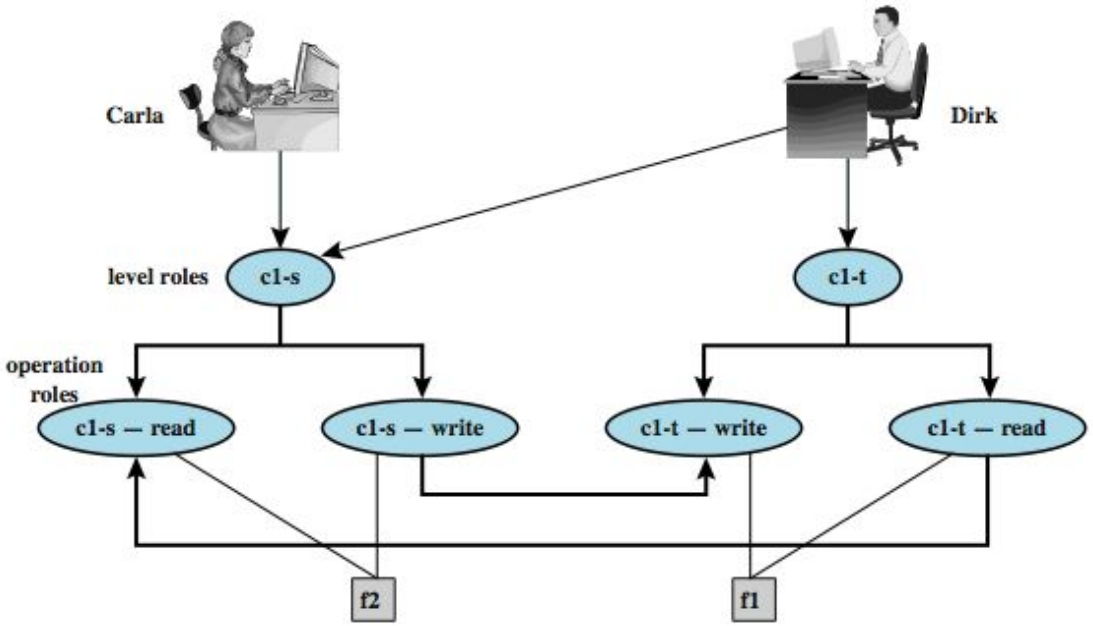
BLP Example



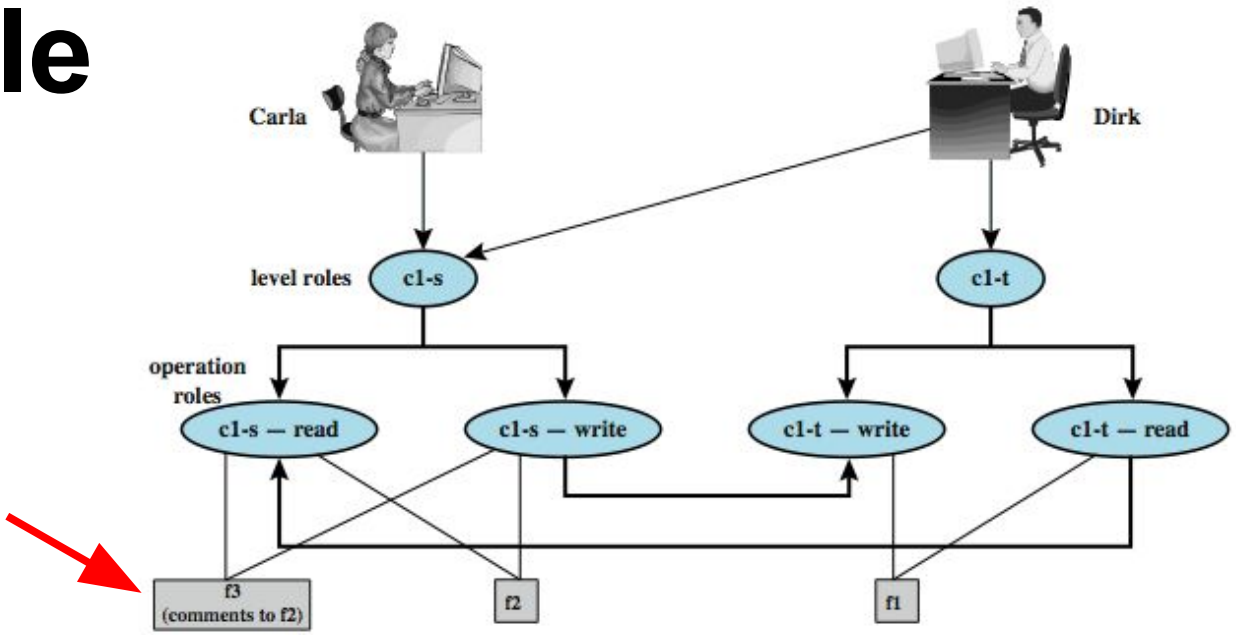
(a) Two new files are created: f1: c1-t; f2: c1-s

Dirk reads f2 and wants to create a file with some comments for Carla.

BLP Example

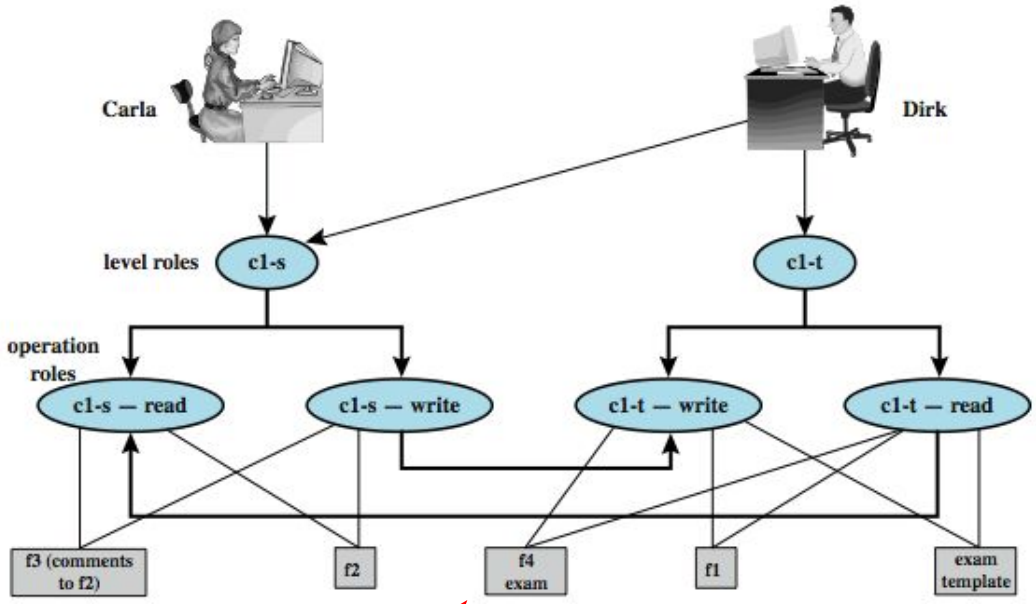


(a) Two new files are created: f1: c1-t; f2: c1-s

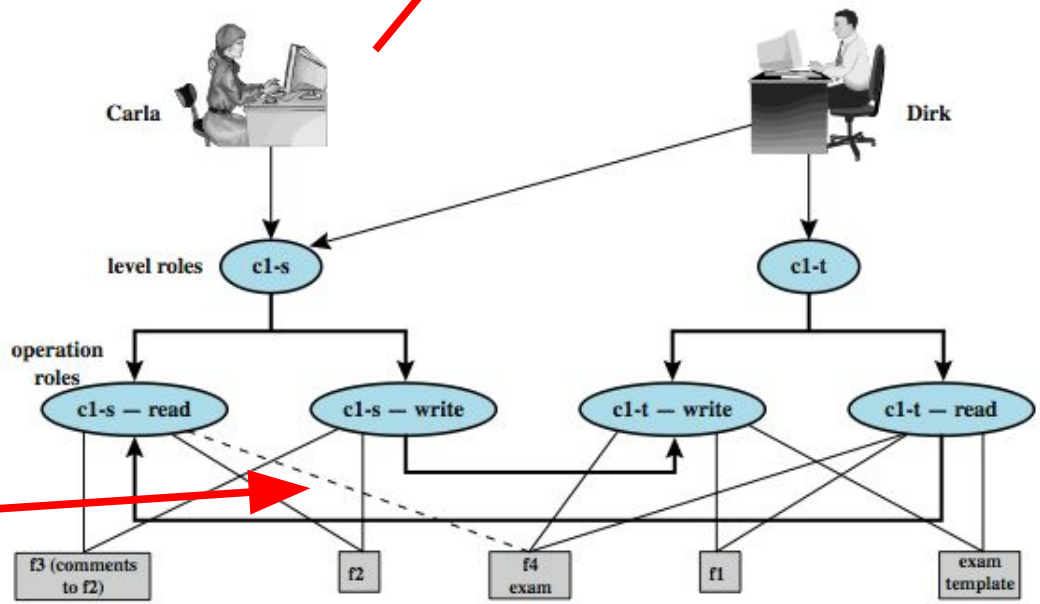


(b) A third file is added: f3: c1-s

BLP Example cont.



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

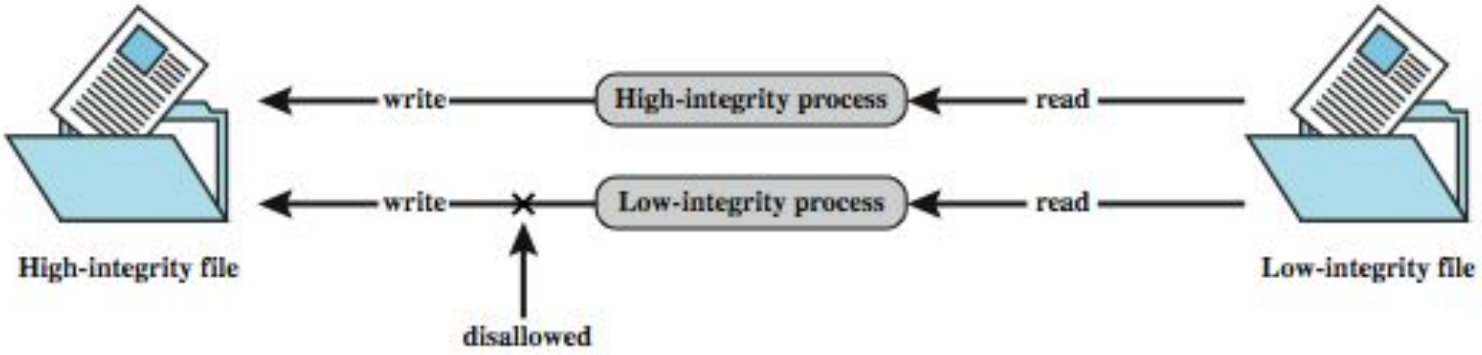


(d) Carla, as student, is permitted acess to the exam: f4: c1-s

Must be allowed outside BLP

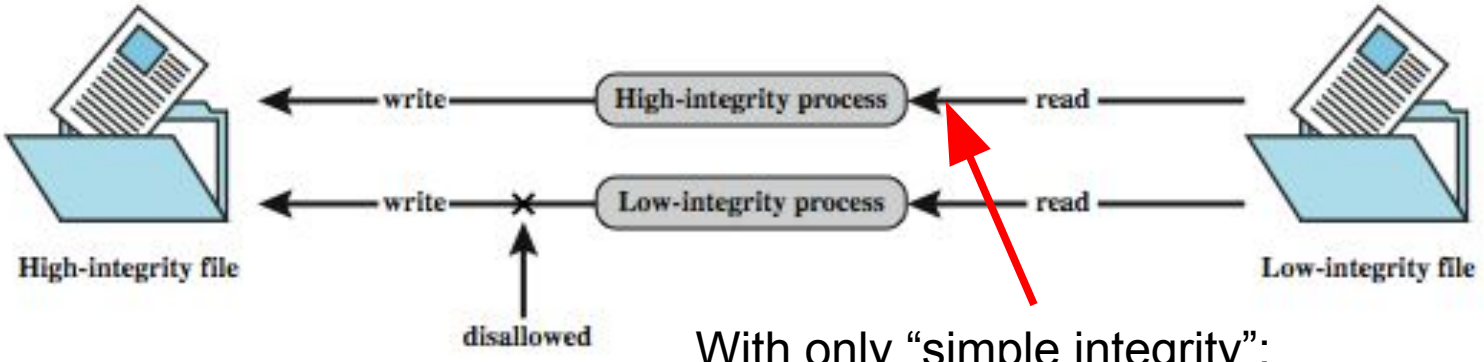
Biba Integrity Model

- various models dealing with **integrity**
- strict integrity policy:
  - simple integrity: S can write if $I(S) \geq I(O)$
  - integrity confinement: S can read if $I(S) \leq I(O)$
  - invocation property: S_1 can inv. S_2 if $I(S_1) \geq I(S_2)$



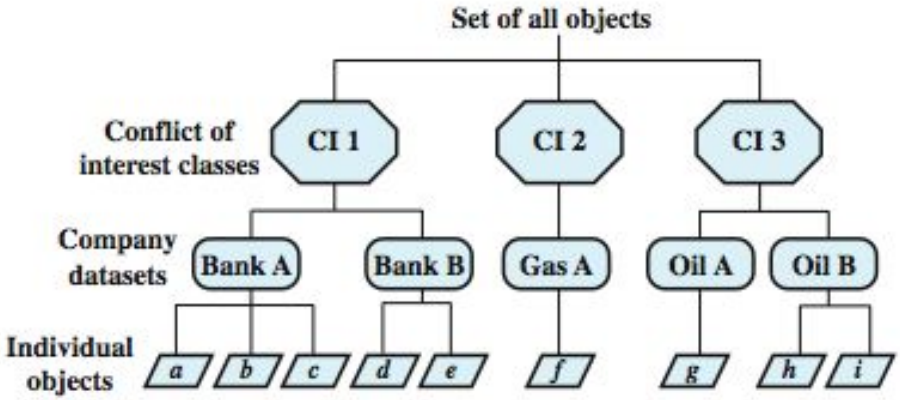
Biba Integrity Model

- various models dealing with **integrity**
- strict integrity policy:
  - simple integrity: S can write if $I(S) \geq I(O)$
  - integrity confinement: S can read if $I(S) \leq I(O)$
  - invocation property: S_1 can inv. S_2 if $I(S_1) \geq I(S_2)$



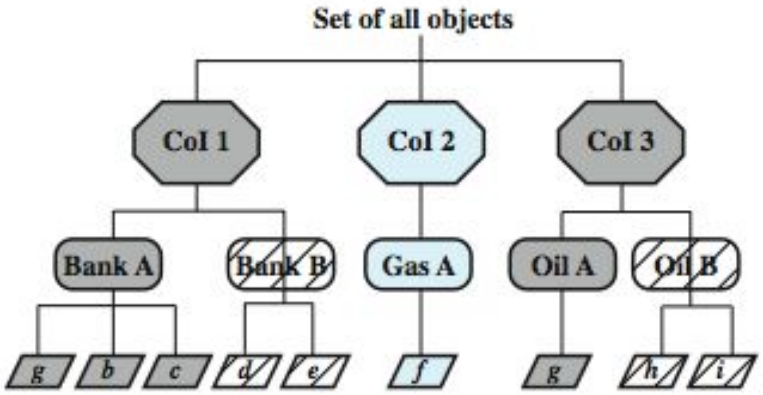
With only “simple integrity”: contamination is possible

Chinese Wall Model

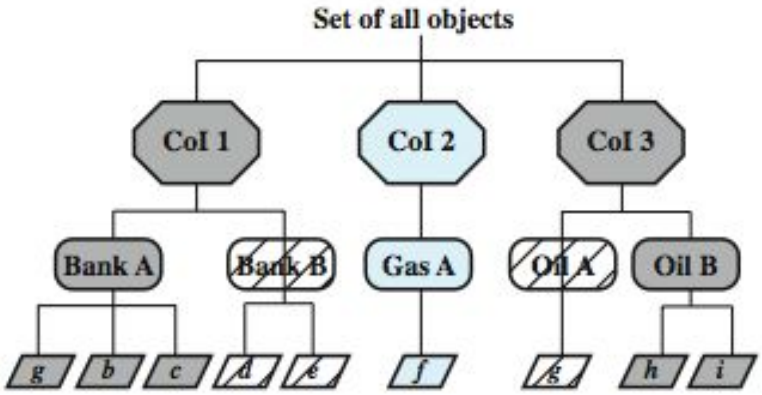


(a) Example set

- SS rule: wall
- * rule: to avoid CI

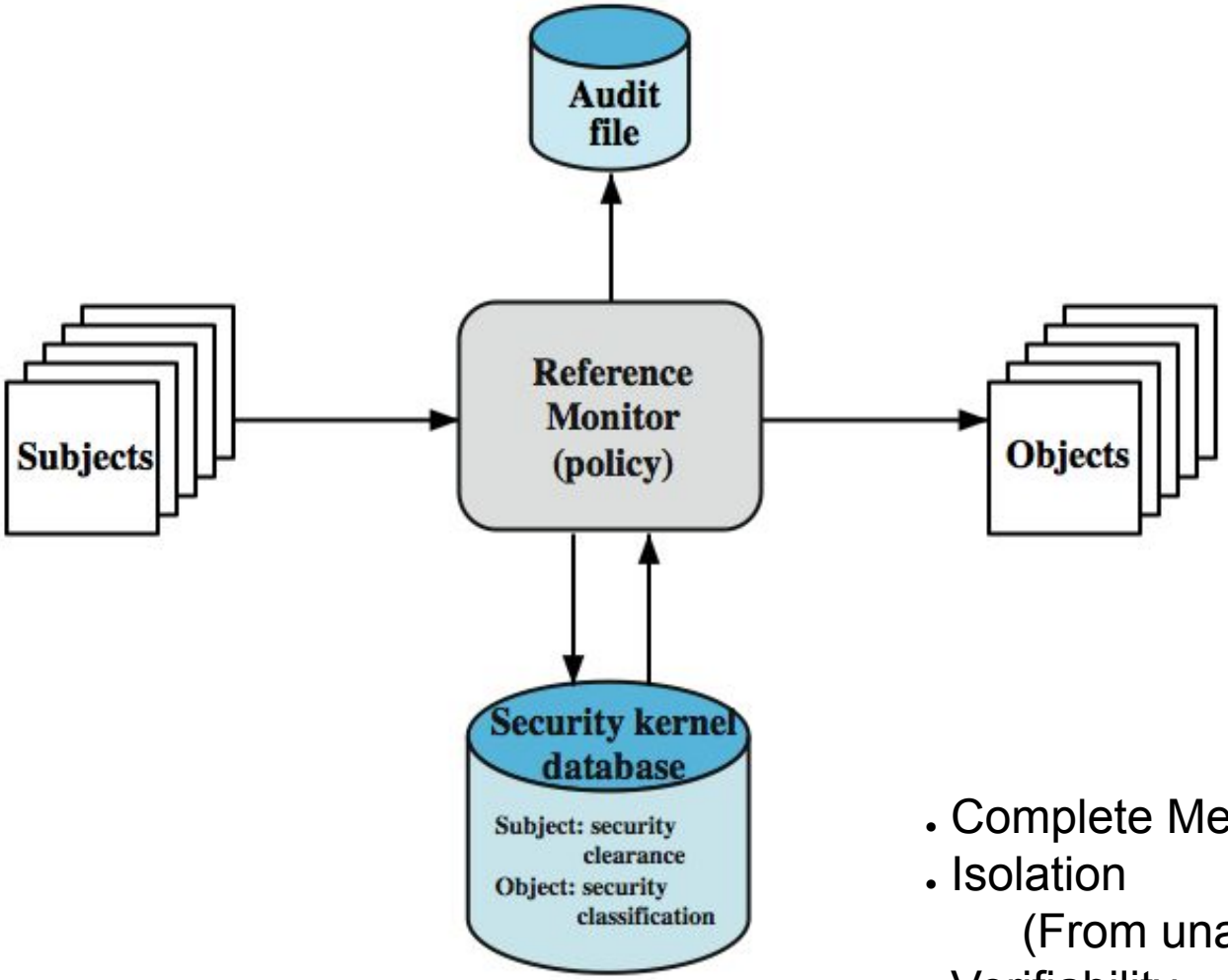


(b) John has access to Bank A and Oil A



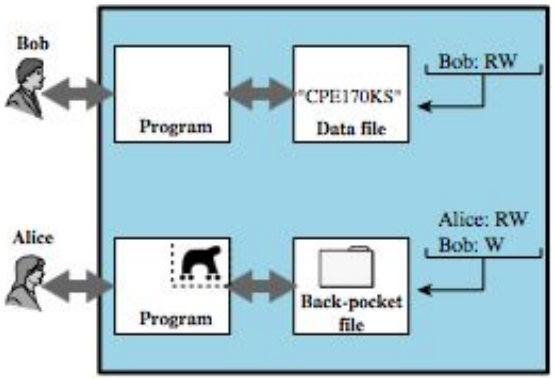
(c) Jane has access to Bank A and Oil B

Reference Monitors

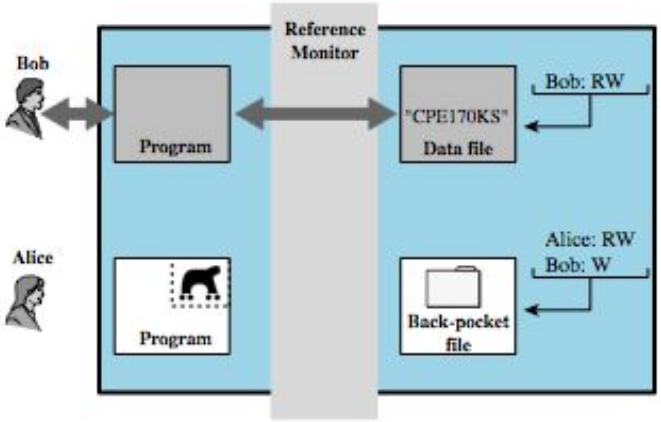


- Complete Mediation
- Isolation
  (From unauth. modification)
- Verifiability

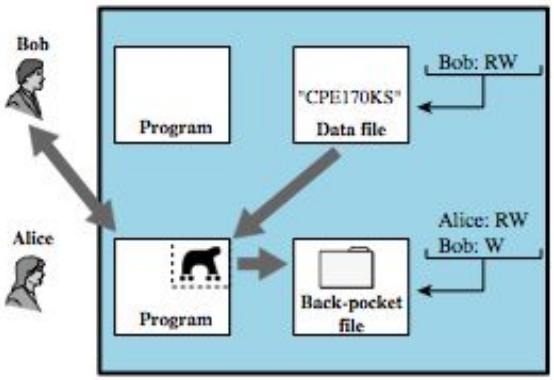
Trojan Horse Defence



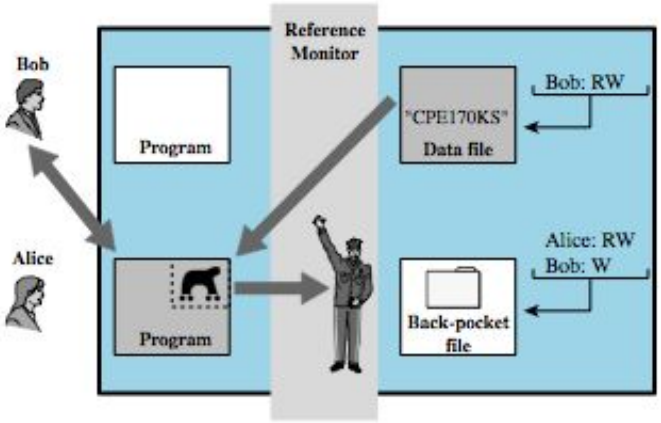
(a)



(c)



(b)



(d)

Multilevel Security (MLS)

➢a class of system that:

- has system resources (particularly stored information) at more than one security level
  (i.e., has different types of sensitive resources)
- and that permits concurrent access by users who differ in security clearance and need-to-know,
- but is able to prevent each user from accessing resources for which the user lacks authorization.

MLS Security for Role-Based Access Control

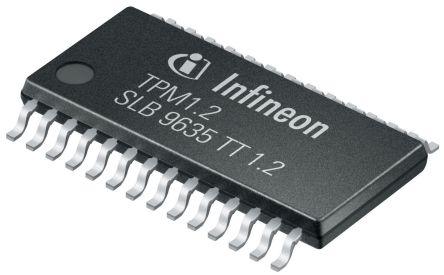
➢ **RBAC** (role based access control)

can implement BLP MLS rules given:

- security constraints on users
- constraints on read/write permissions
- read and write level role access definitions
- constraint on user-role assignments

Trusted Platform Module (TPM)

- concept from Trusted Computing Group
- hardware module at heart of hardware / software approach to trusted computing
- uses a TPM chip on
  - motherboard, smart card, processor
  - working with approved hardware / software
  - generating and using crypto keys
- has 3 basic services: authenticated boot, certification, and encryption



Authenticated Boot Service

- responsible for booting entire O/S in stages
- ensuring each is valid and approved for use
  - verifying digital signature associated with code
  - keeping a tamper-evident log
- log records versions of all code running
- can then expand trust boundary
  - TPM verifies any additional software requested
    - confirms signed and not revoked
- hence know resulting configuration is well-defined with approved components

Certification Service

- once have authenticated boot
- TPM can certify configuration to others
  - with a digital certificate of configuration info
  - giving another user confidence in it
- include challenge value in certificate to also ensure it is timely
- provides hierarchical certification approach
  - trust TPM then O/S then applications

Encryption Service

- encrypts data so it can be decrypted
  - by a certain machine in given configuration
- depends on
  - master secret key unique to machine
  - used to generate secret encryption key for every possible configuration only usable in it
- can also extend this scheme upward
  - create application key for desired application version running on desired system version

TPM Functions