Computer Security DD2395

http://www.csc.kth.se/utbildning/kth/kurser/DD2395/dasak10/

Spring 2010

Sonja Buchegger
buc@kth.se

Lecture 11, Feb. 22, 2010

Multi-Level Security

Announcements

- Presentation topics, tasks
- More dry-run, presentation slots to come
- Guiding questions for presentation and abstract

Trusted Computing and Multilevel Security

- present some interrelated topics:
  - formal models for computer security
  - multilevel security
  - trusted systems
  - mandatory access control
  - security evaluation

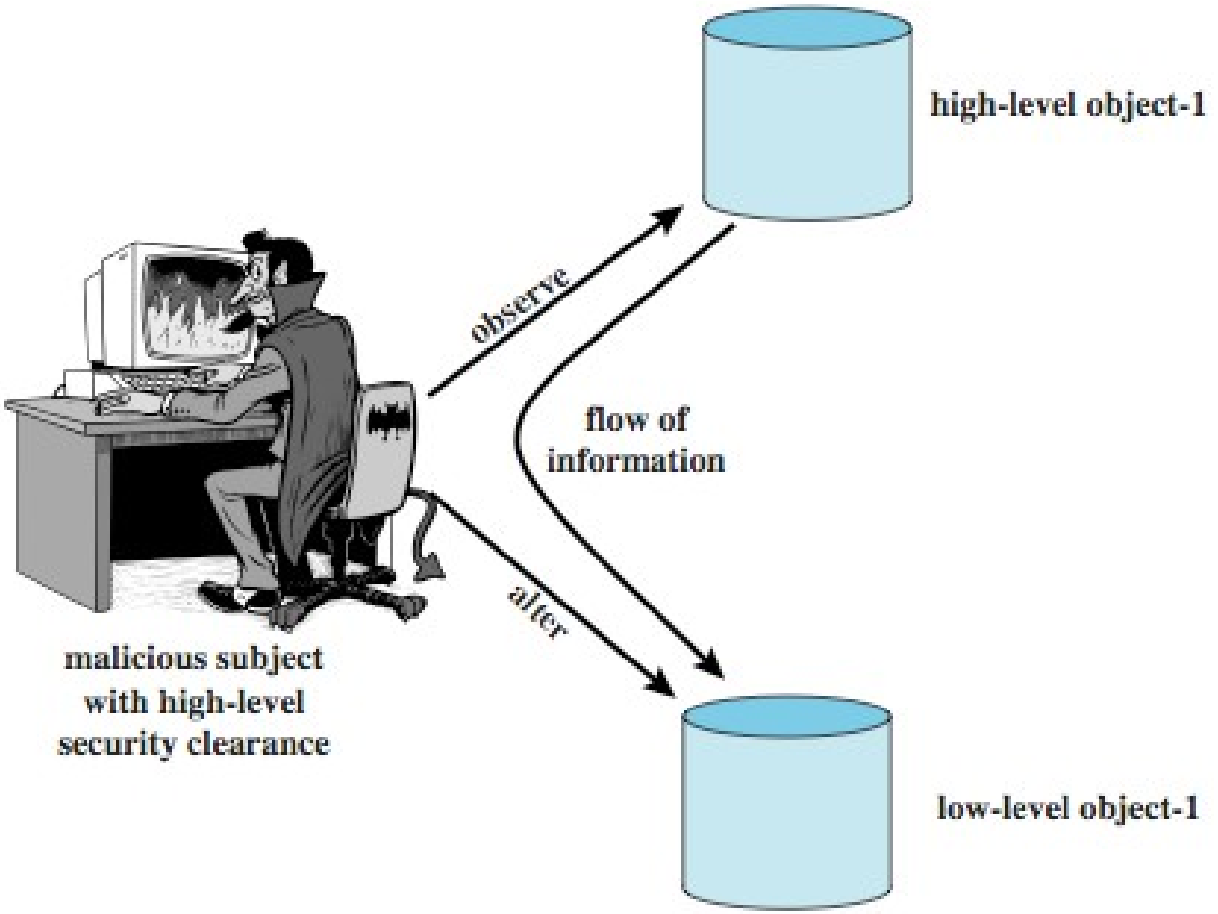
Formal Models for Computer Security

- 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 DoD
- 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
  - subject has a **security clearance** level
  - object has a **security classification** level
  - class control how subject may access an object
- applicable if have info and user categories

Multi-Level Security



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 $A_x \in M[S_i$
- BLP give formal theorems
  - theoretically possible to prove system is secure
  - in practice usually not possible

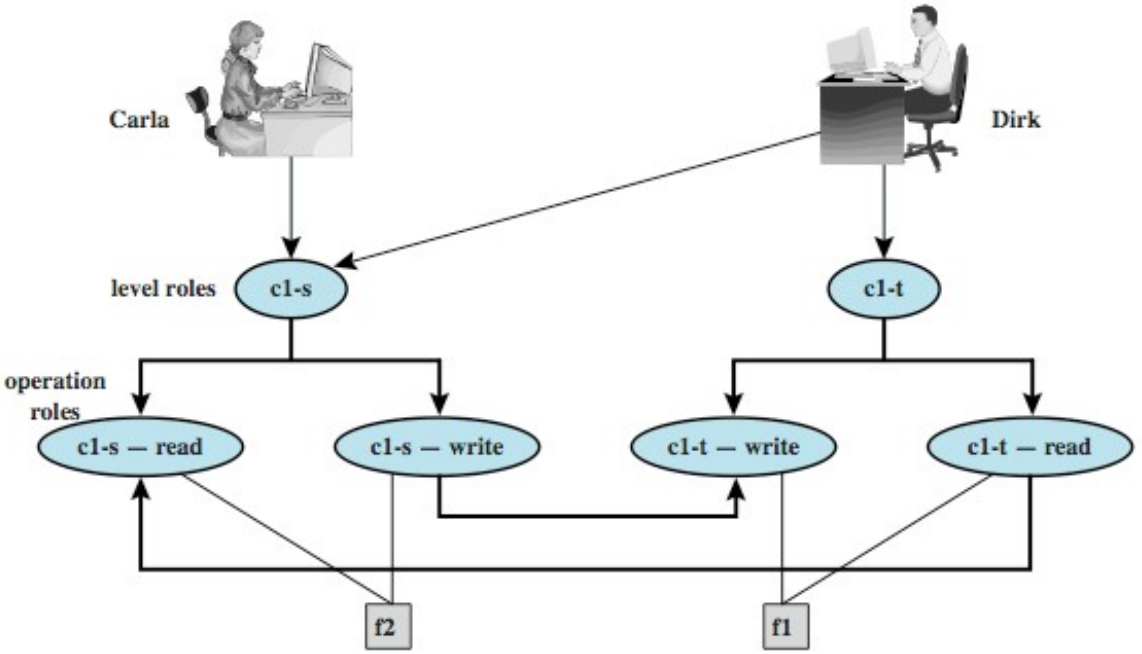
Question

- No read up, no write down. Why no write down?

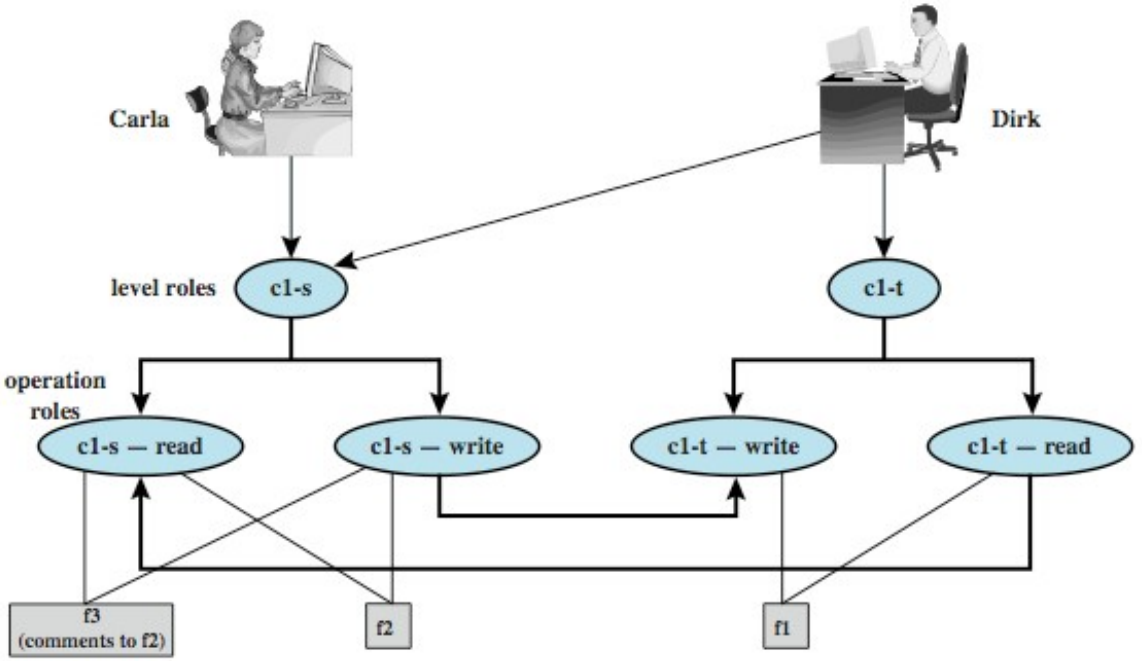
BLP Rules

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

BLP Example

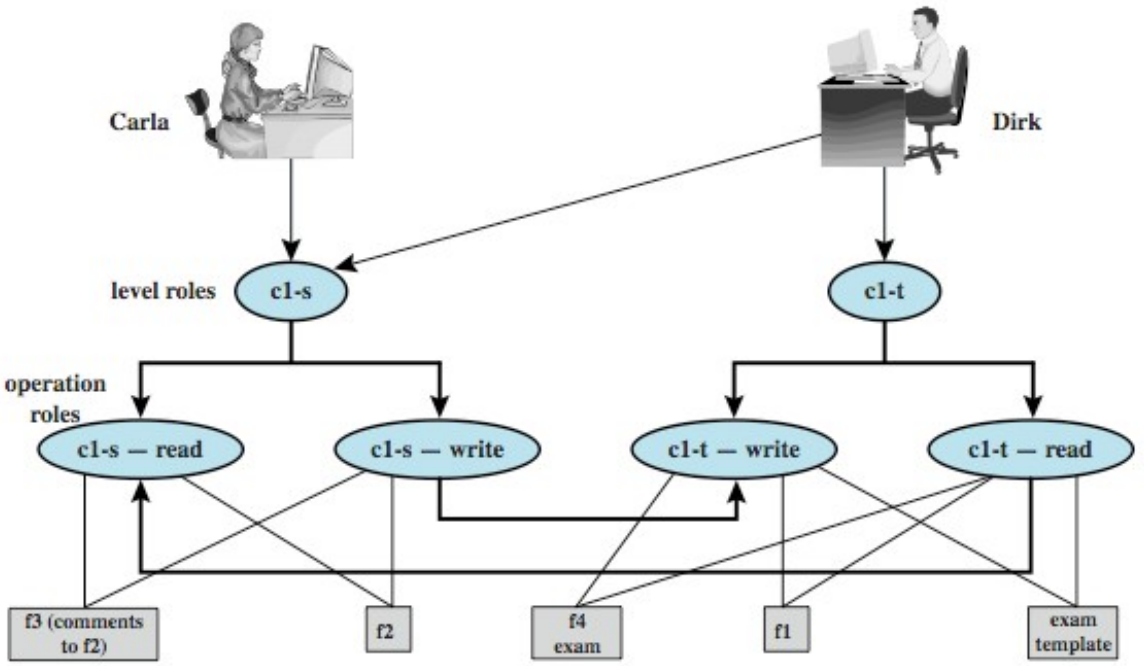


(a) Two new files are created: f1: c1-t; f2: c1-s

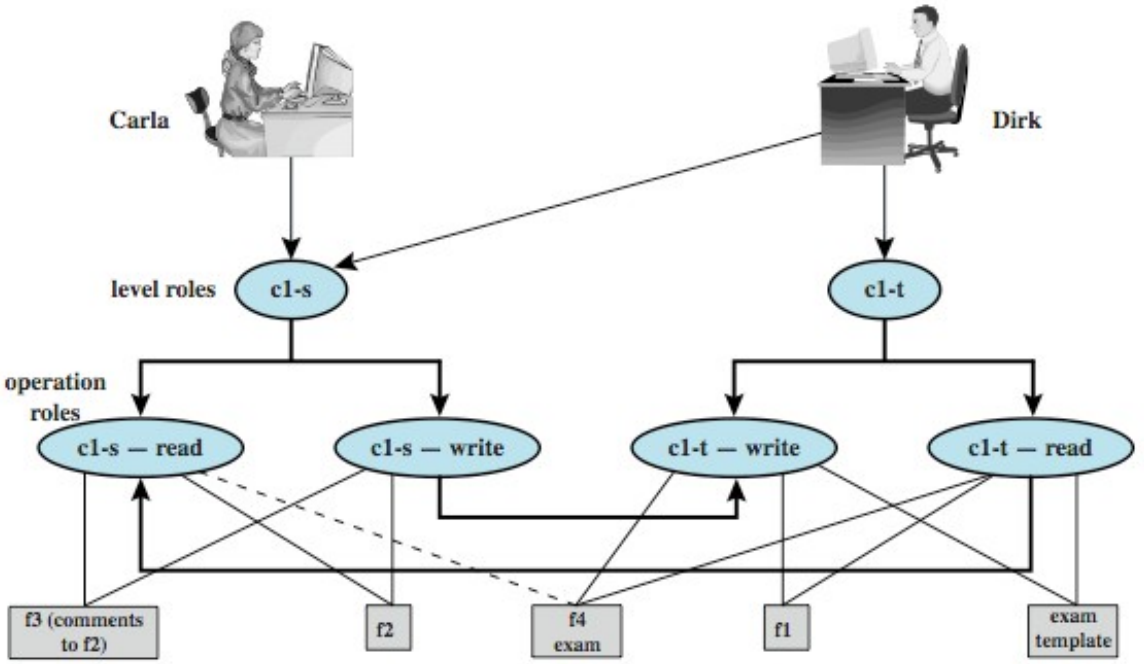


(b) A third file is added: f3: c1-s

BLP Example cont.

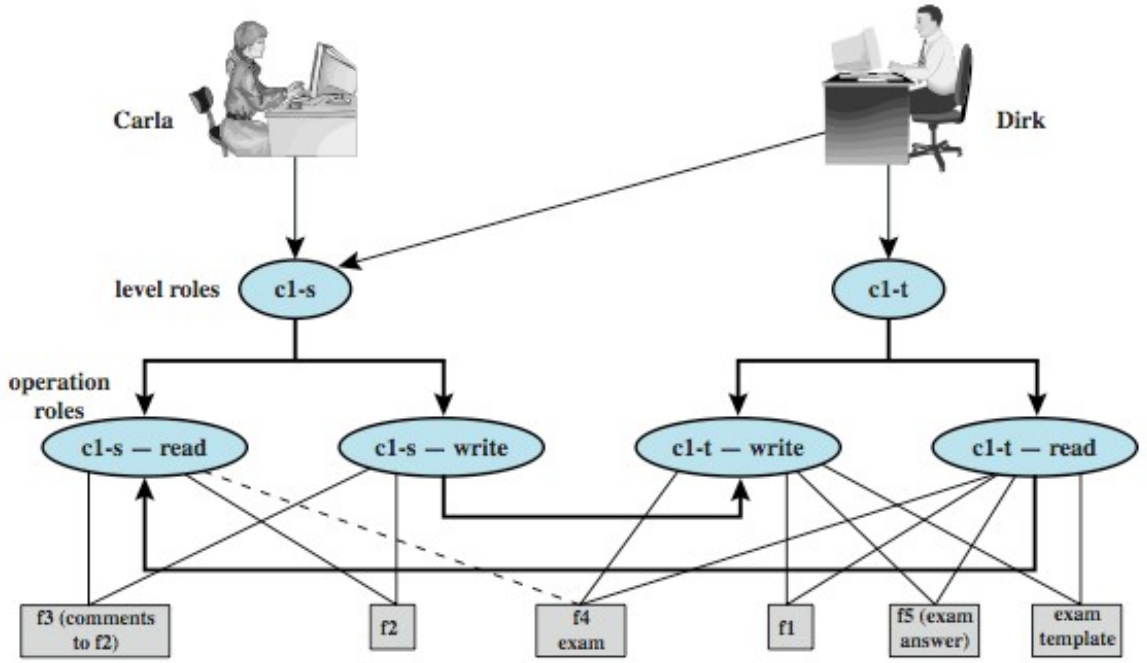


(c) An exam is created based on an existing template: f4: c1-t



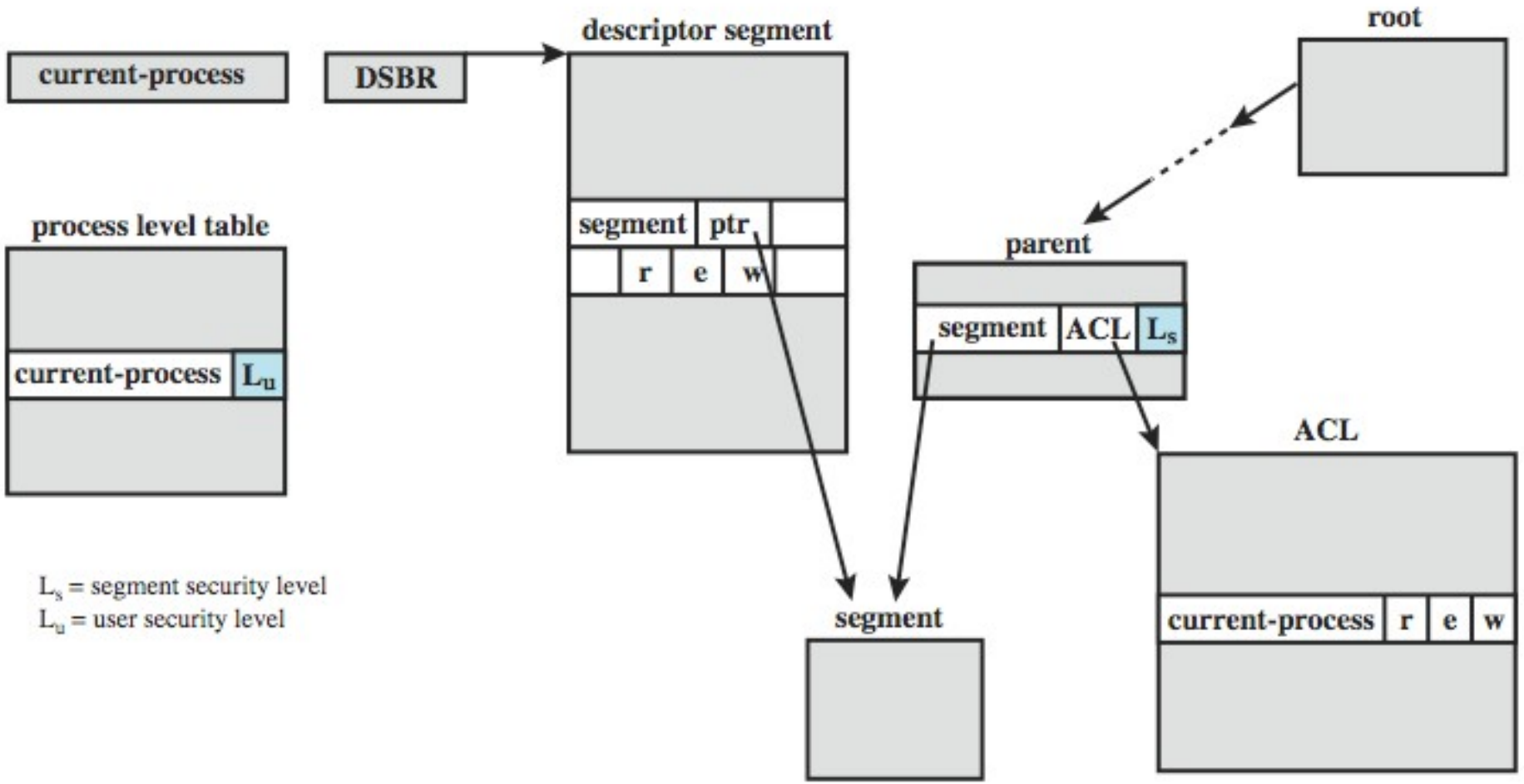
(d) Carla, as student, is permitted acess to the exam: f4: c1-s

BLP Example cont.



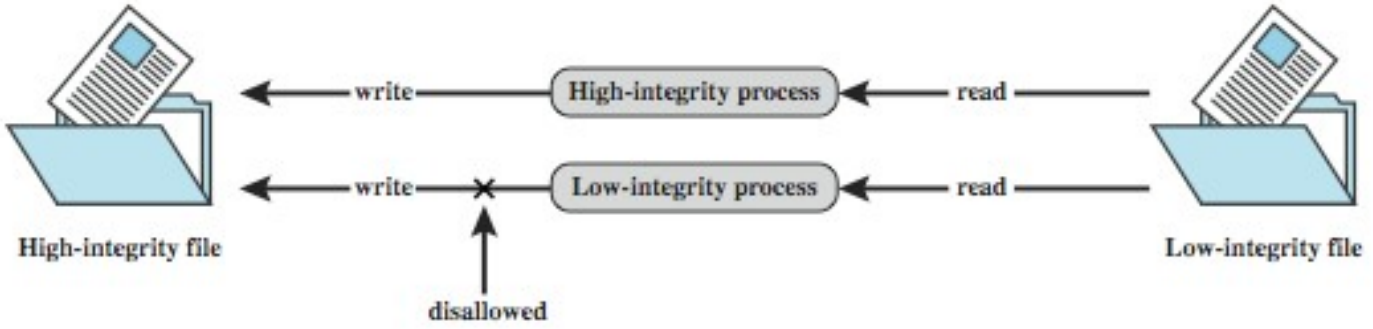
(e) The answers given by Carla are only accessible for the teacher: f5: c1-t

MULTICS Example

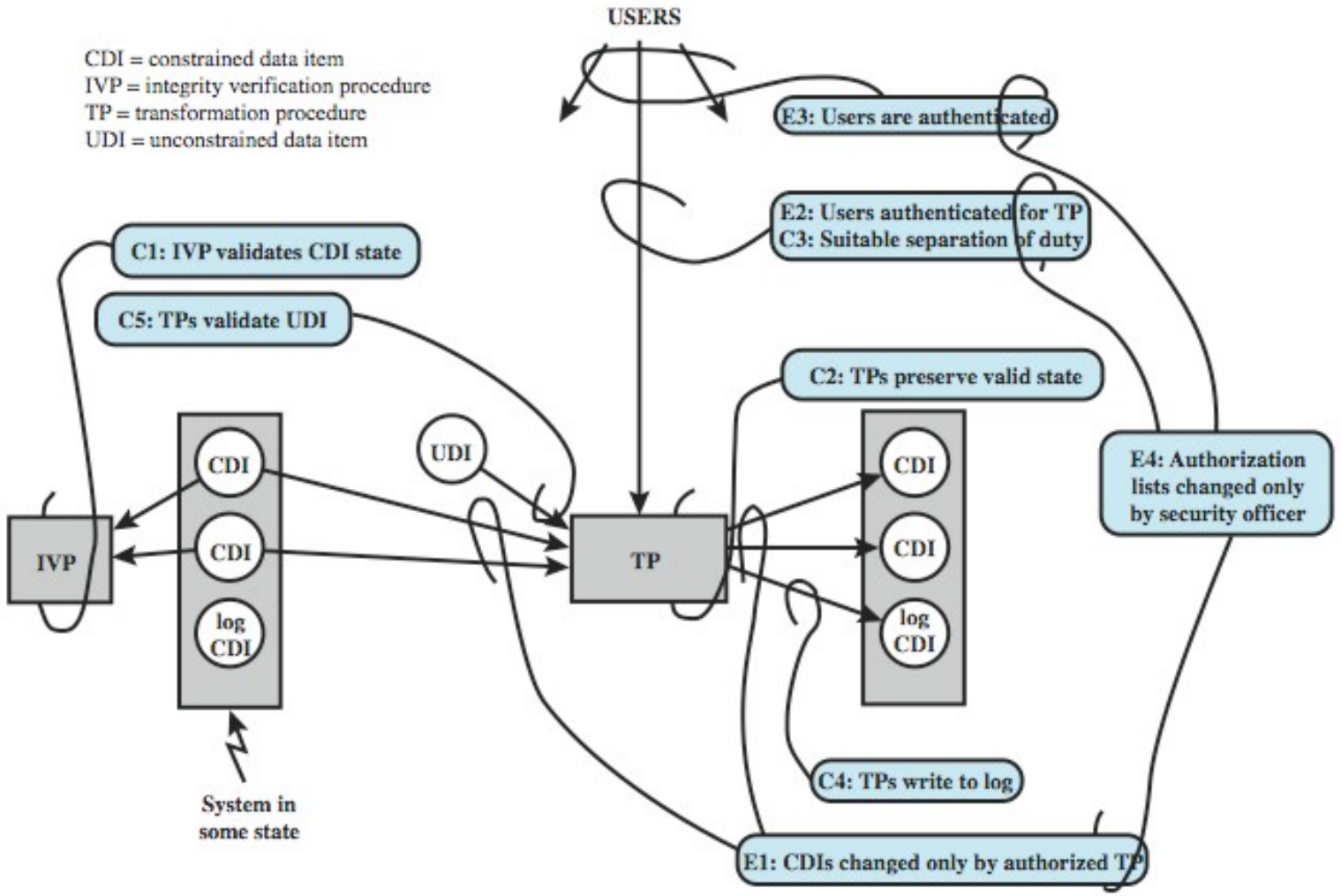


Biba Integrity Model

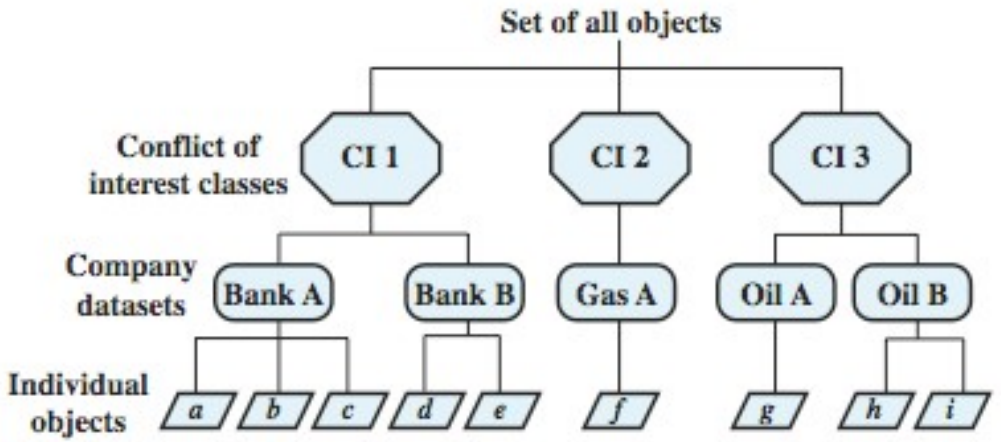
- various models dealing with integrity
- strict integrity policy:
  - simple integrity: $I(S) \geq I(O)$
  - integrity confinement: $I(S) \leq I(O)$
  - invocation property: $I(S_1) \geq I(S_2)$



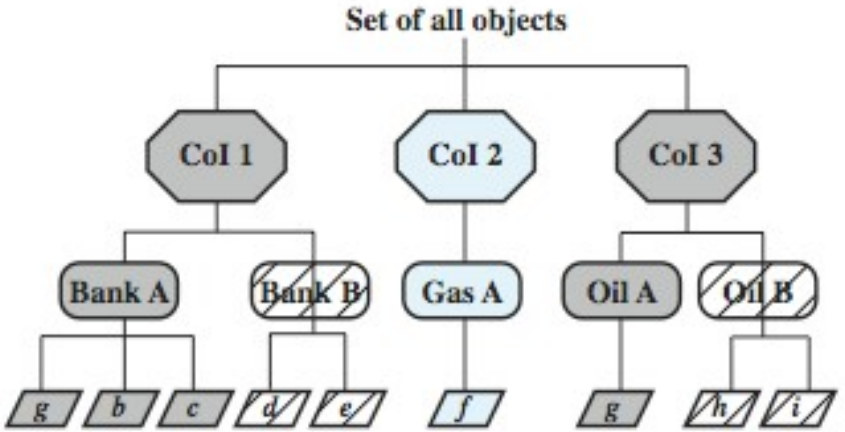
Clark-Wilson Integrity Model



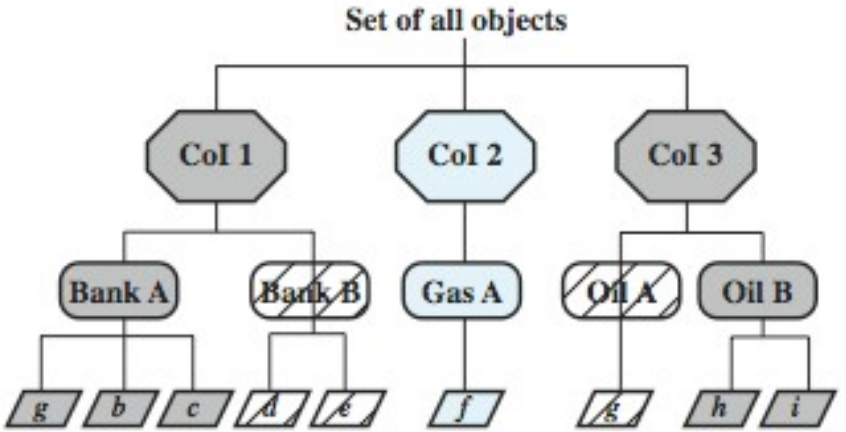
Chinese Wall Model



(a) Example set

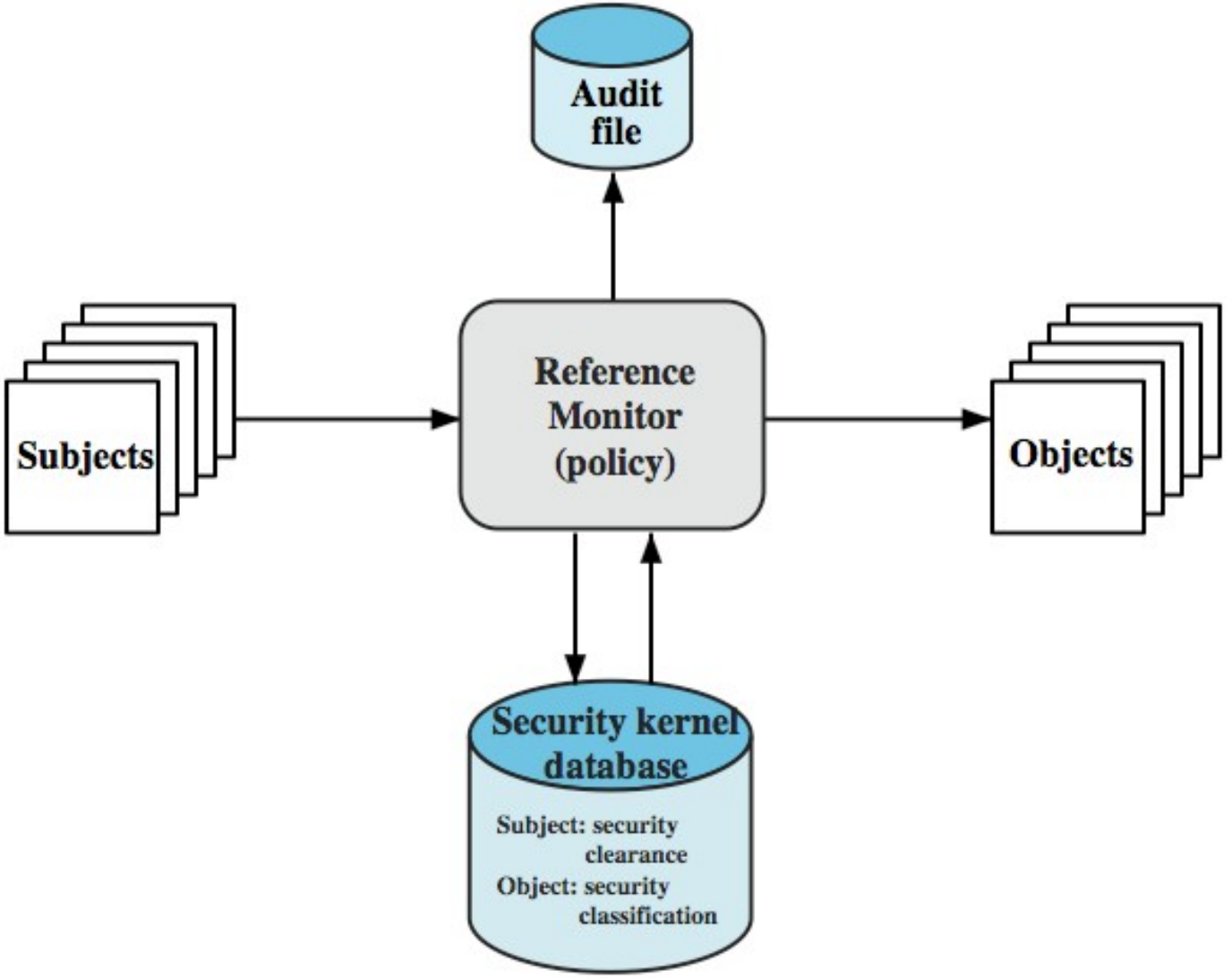


(b) John has access to Bank A and Oil A

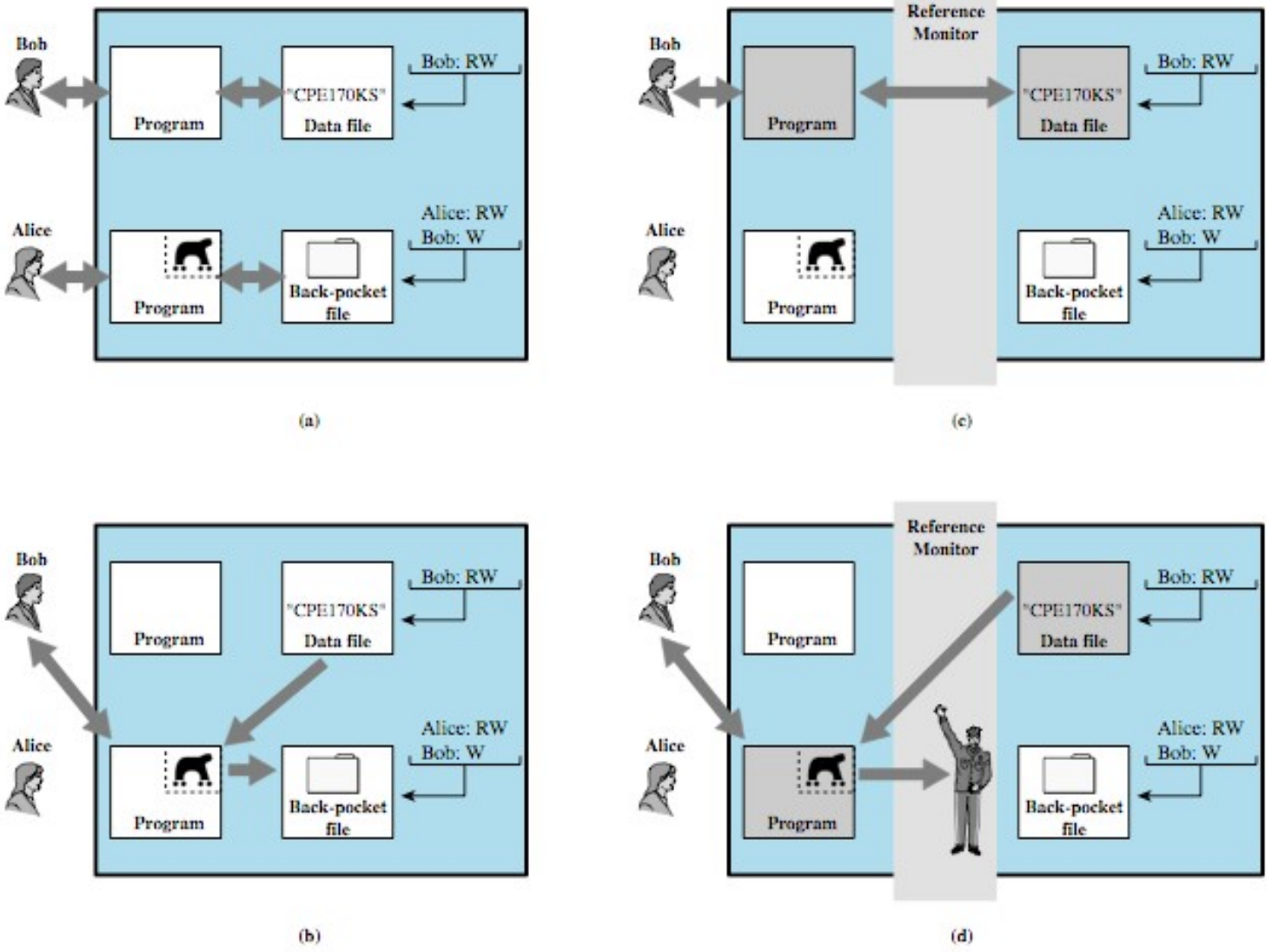


(c) Jane has access to Bank A and Oil B

Reference Monitors



Trojan Horse Defence



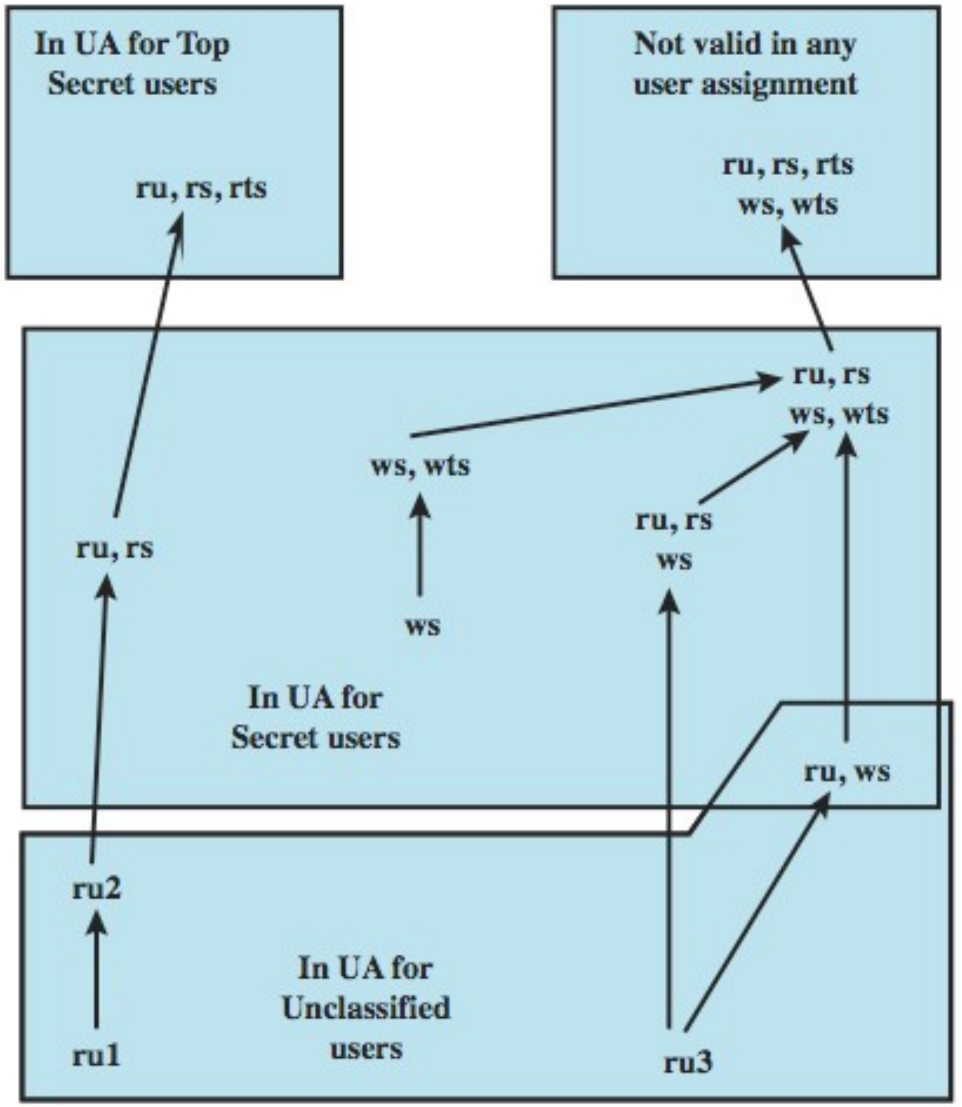
Multilevel Security (MLS)

- 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

- rule based access control (RBAC) 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

RBAC MLS Example



MLS Database Security

Department Table - U		
Did	Name	Mgr
4	accts	Cathy
8	PR	James

Employee - R			
Name	Did	Salary	Eid
Andy	4	43K	2345
Calvin	4	35K	5088
Cathy	4	48K	7712
James	8	55K	9664
Ziggy	8	67K	3054

(a) Classified by table

Department Table		
Did -U	Name -U	Mgr -R
4	accts	Cathy
8	PR	James

Employee			
Name -U	Did -U	Salary -R	Eid -U
Andy	4	43K	2345
Calvin	4	35K	5088
Cathy	4	48K	7712
James	8	55K	9664
Ziggy	8	67K	3054

(b) Classified by column (attribute)

MLS Database Security

Department Table			
Did	Name	Mgr	
4	accts	Cathy	R
8	PR	James	U

Employee				
Name	Did	Salary	Eid	
Andy	4	43K	2345	U
Calvin	4	35K	5088	U
Cathy	4	48K	7712	U
James	8	55K	9664	R
Ziggy	8	67K	3054	R

(c) Classified by row (tuple)

Department Table		
Did	Name	Mgr
4 - U	accts - U	Cathy - R
8 - U	PR - U	James -R

Employee			
Name	Did	Salary	Eid
Andy - U	4 - U	43K - U	2345 - U
Calvin - U	4 - U	35K - U	5088 - U
Cathy - U	4 - U	48K - U	7712 - U
James - U	8 - U	55K - R	9664 - U
Ziggy - U	8 - U	67K - R	3054 - U

(d) Classified by element

MLS Database Security
Read Access

- DBMS enforces simple security rule (no read up)
- easy if granularity entire database / table level
- inference problems if have column granularity
  - if can query on restricted data can infer its existence
    - `SELECT Ename FROM Employee WHERE Salary > 50K`
  - solution is to check access to all query data
- also have problems if have row granularity
  - null response indictes restricted/empty result
- no extra concerns if have element granularity

MLS Database Security Write Access

- enforce *-security rule (no write down)
- have problem if a low clearance user wants to insert a row with a primary key that already exists in a higher level row:
  - can reject, but user knows row exists
  - can replace, compromises data integrity
  - can polyinstantiation and insert multiple rows with same key, creates conflicting entries
- same alternatives occur on update
- avoid problem if use database / table granularity

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

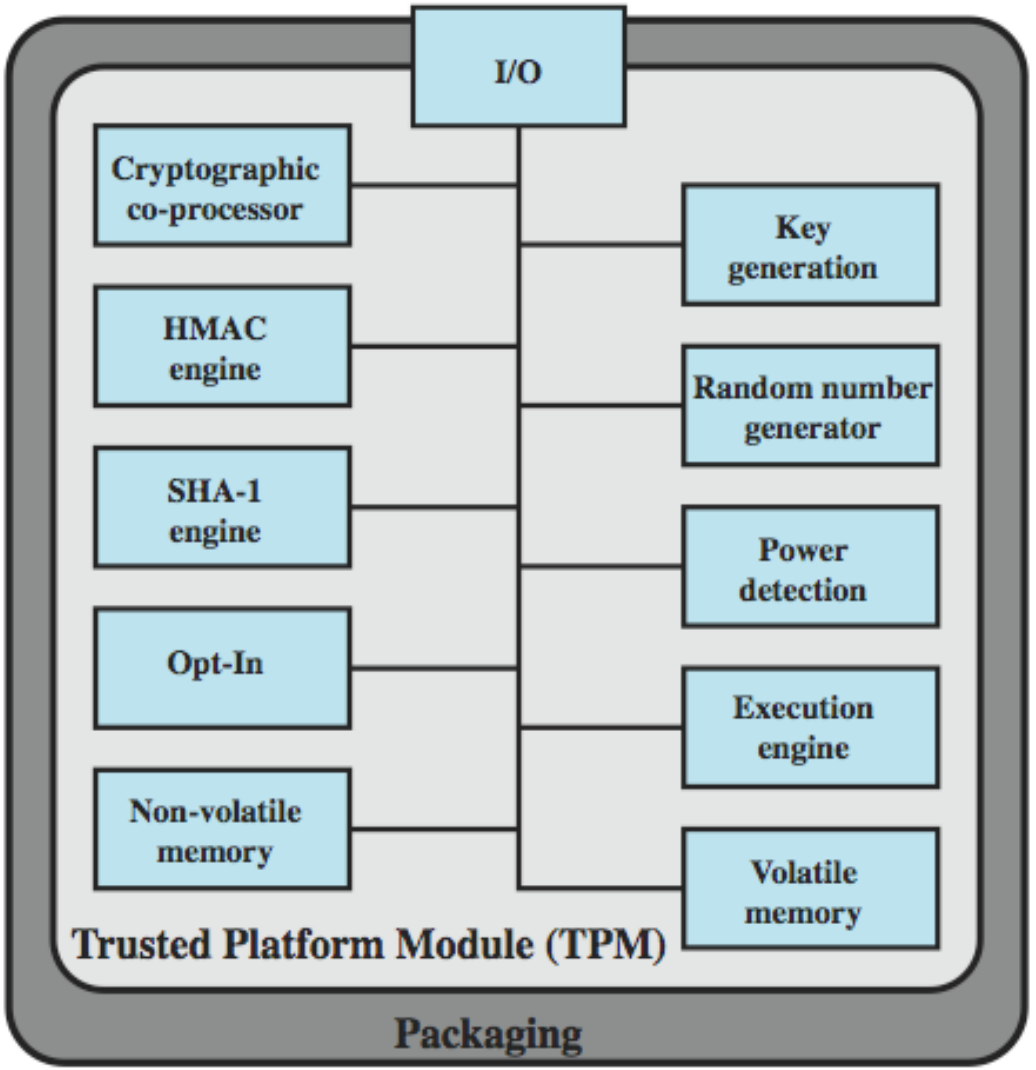
Certification Service

- 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

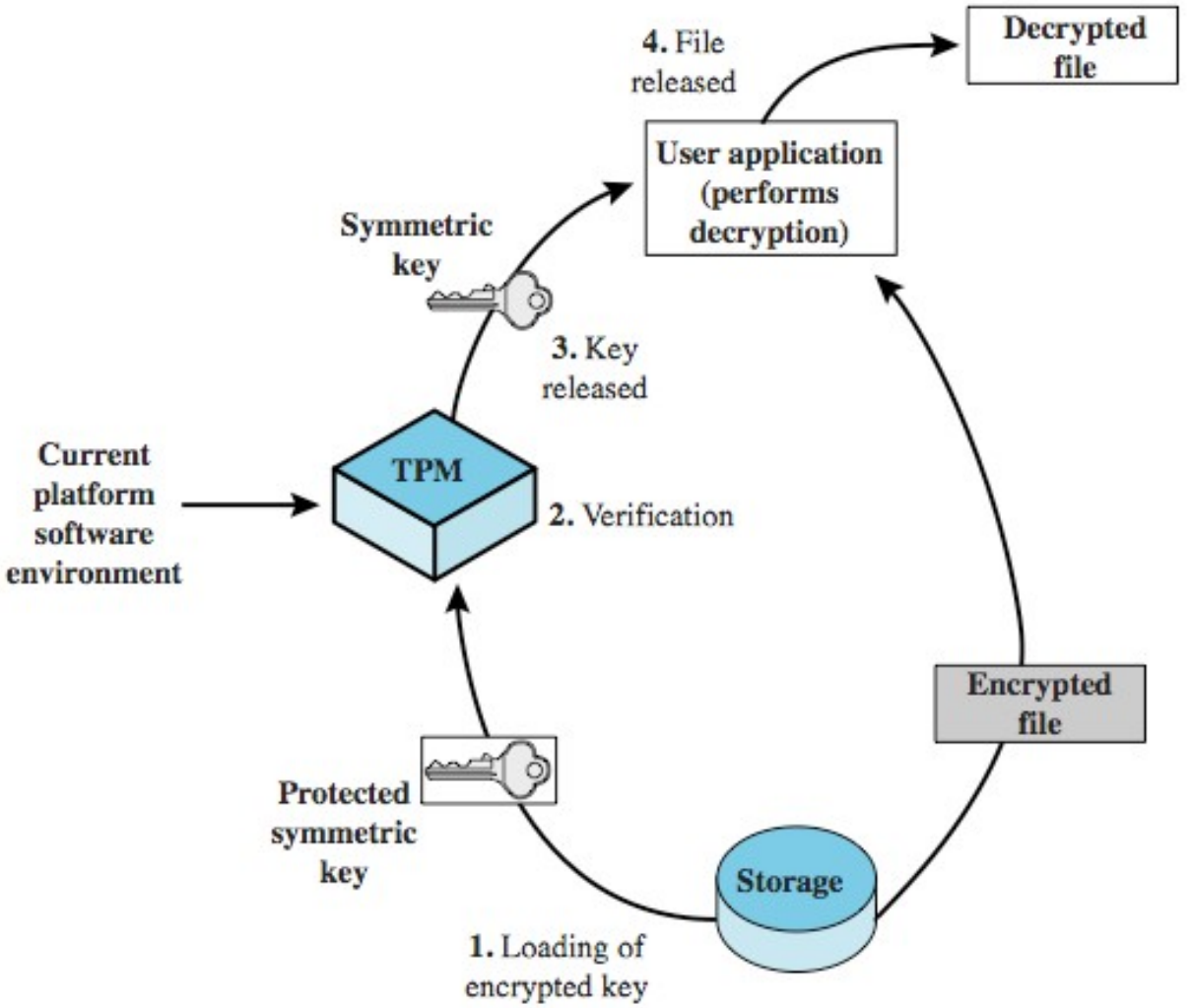
Encryption Service

- 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



Protected Storage Function



Trusted Systems

- security models aimed at enhancing trust
- work started in early 1970's leading to:
  - Trusted Computer System Evaluation Criteria (TCSEC), Orange Book, in early 1980s
  - further work by other countries
  - resulting in Common Criteria in late 1990s
- also Computer Security Center in NSA
  - with Commercial Product Evaluation Program
  - evaluates commercially available products
  - required for Defense use, freely published

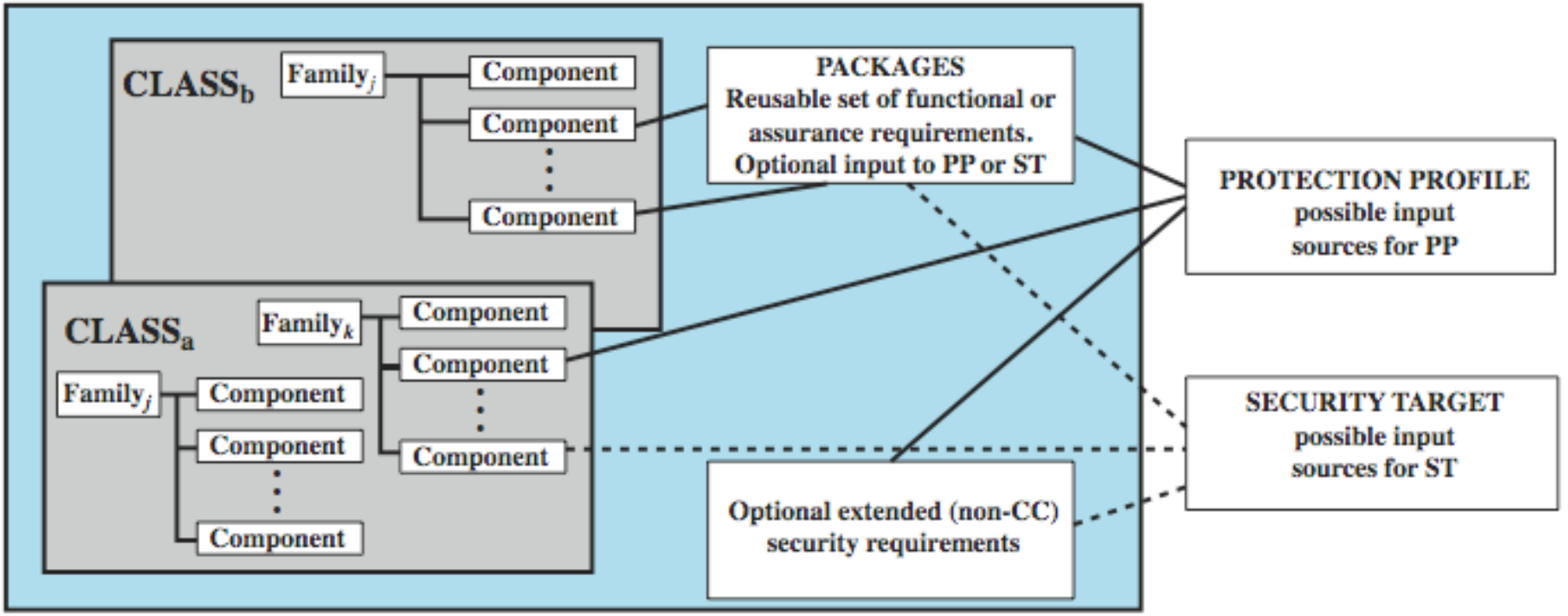
Common Criteria (CC)

- ISO standards for security requirements and defining evaluation criteria to give:
  - greater confidence in IT product security
  - from formal actions during process of:
  - development using secure requirements
  - evaluation confirming meets requirements
  - operation in accordance with requirements
- evaluated products are listed for use

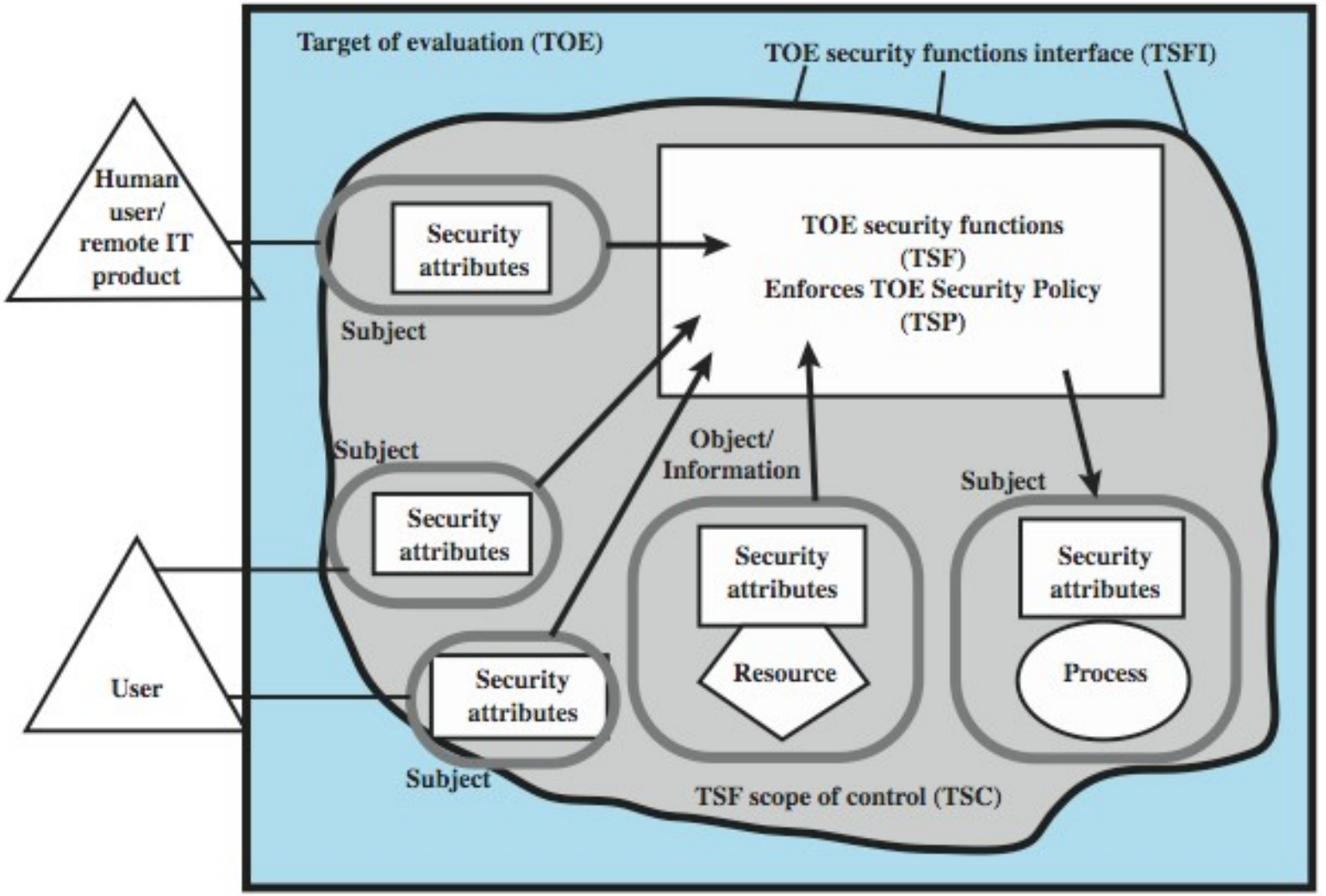
CC Requirements

- have a common set of potential security requirements for use in evaluation
- target of evaluation (TOE) refers product / system subject to evaluation
- functional requirements
  - define desired security behavior
- assurance requirements
  - that security measures effective correct
- have classes of families of components

CC Profiles and Targets



CC Security Paradigm



Smartcard PP

- simple PP example
- describes IT security requirements for smart card use by sensitive applications
- lists threats
- PP requirements:
  - 42 TOE security functional requirements
  - 24 TOE security assurance requirements
  - IT environment security requirements
- with rationale for selection

Assurance

- “degree of confidence that the security controls operate correctly and protect the system as intended”
- applies to:
  - product security requirements, security policy, product design, implementation, operation
- various approaches analyzing, checking, testing various aspects

CC Assurance Levels

- EAL 1 - functionally tested
- EAL 2: structurally tested
- EAL 3: methodically tested and checked
- EAL 4: methodically designed, tested, and reviewed
- EAL 5: semiformally designed and tested
- EAL 6: semiformally verified design and tested
- EAL 7: formally verified design and tested

Evaluation

- ensure security features correct & effective
- performed during / after TOE development
- higher levels need greater rigor and cost
- input: security target, evidence, actual TOE
- result: confirm security target satisfied for TOE
- process relates security target to some of TOE:
  - high-level design, low-level design, functional spec, source code, object code, hardware realization
- higher levels need semiformal / formal models

Evaluation Parties & Phases

- evaluation parties:
  - sponsor - customer or vendor
  - developer - provides evidence for evaluation
  - evaluator - confirms requirements satisfied)
  - certifier - agency monitoring evaluation process
- phases:
  - preparation, conduct of evaluation, conclusion
- government agency regulates, e.g. US CCEVS
- have peering agreements between countries
  - saving time / expense by sharing results

Summary

- Bell-Lapadula security model
- other models
- reference monitors & trojan horse defence
- multilevel secure RBAC and databases
- trusted platform module
- common criteria
- assurance and evaluation