İTÜ
Computer Security

Trusted Computing and Multilevel Security

Dr. Şerif Bahtiyar

bahtiyars@itu.edu.tr

Fall 2015

Before Starting

US banks attacked, manipulated and left (heart) bleeding

In April 2014 the cybersecurity world was rocked by the discovery of Heartbleed, the name given to a vulnerability found in one of the systems we use to securely communicate over the internet.

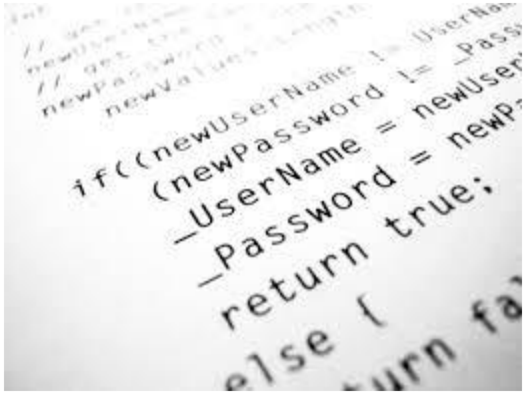
http://www.bbc.com/news/technology-34783770

Outline

- Bell-LaPadula Model (BPL)
- Other Formal Models
- Trusted Systems
- Multilevel Security
- Trusted Computing and Trusted Platform Module

The Bell-LaPadula Model

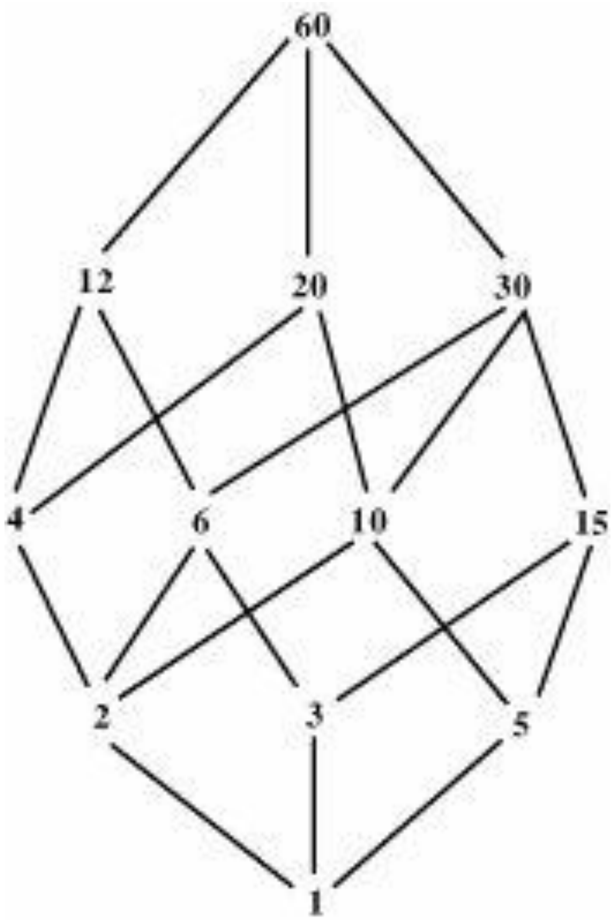
- Two fundamental security facts:
  - All complex software systems have bugs or flaws
  - Difficult to build a computer hardware/software system not vulnerable to security attacks.



- For example, Windows NT OS
  - Introduced in early 1990
  - Promised to have high degree of security
  - Did not deliver on this promise
  - Has wide range of security vulnerabilities

The Bell-LaPadula Model

- Problems to provide strong security involve both design and implementation.
- Hence, there is desire to prove design and implementation that satisfy security requirements.
- Thus, develop formal models of computer security to verify security design and implementation



The Bell-LaPadula Model

- The most influential security model
- Developed in the 1970 as a formal model for access control
- Each subject and each object is assigned to a security class
- Security classes form a strict hierarchy (security levels)
  - top secret -> secret -> confidential -> restricted -> unclassified
- A subject has a security clearance level
- An object has a security classification level
- Classes control how subject may access an object

The Bell-LaPadula Model

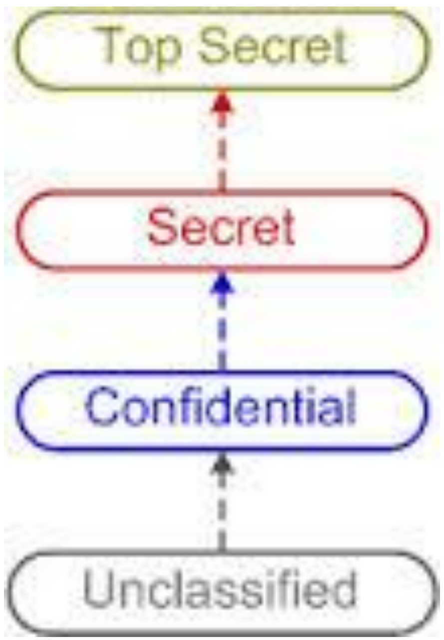
Access modes

- **Read**: subject is allowed only read access to object
- **Append**: subject is allowed only write access to object
- **Write**: subject is allowed both read and write access to object
- **Execute**: subject is allowed neither read nor write access to object but may invoke the object for execution.

The Bell-LaPadula Model

- Multilevel security: it is multiple categories or levels of data.

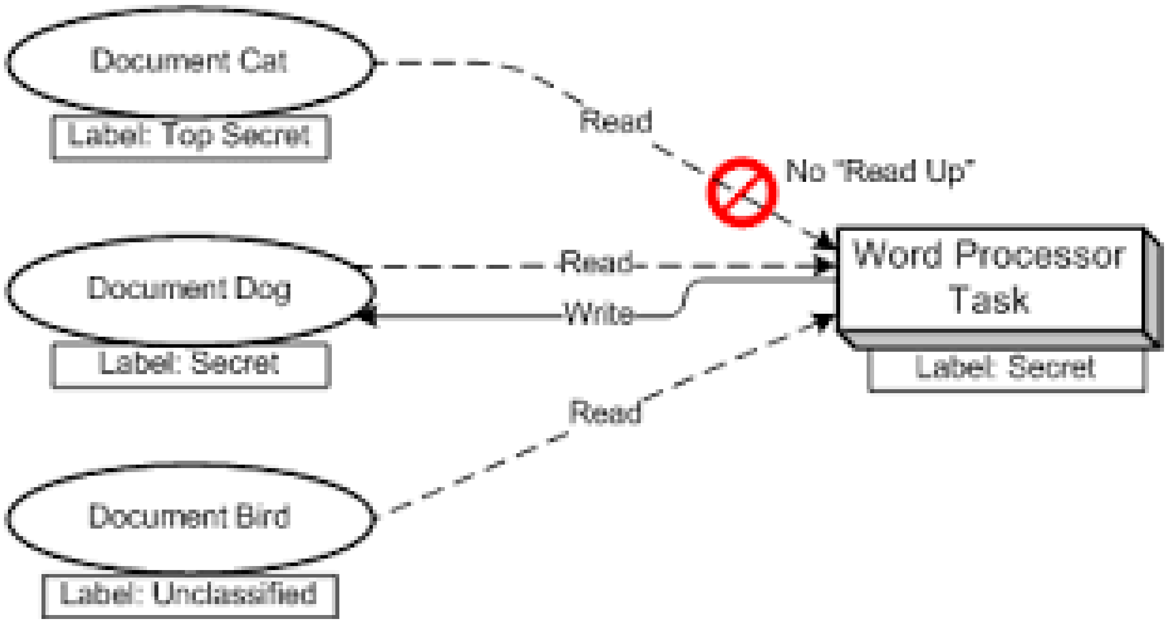
- In confidentiality-centered multilevel security, a subject at a high level may not convey information to a subject at a lower level unless that flow accurately reflects the will of an authorized user as revealed by an authorized declassification.



The Bell-LaPadula Model

A multilevel secure system for confidentiality must enforce

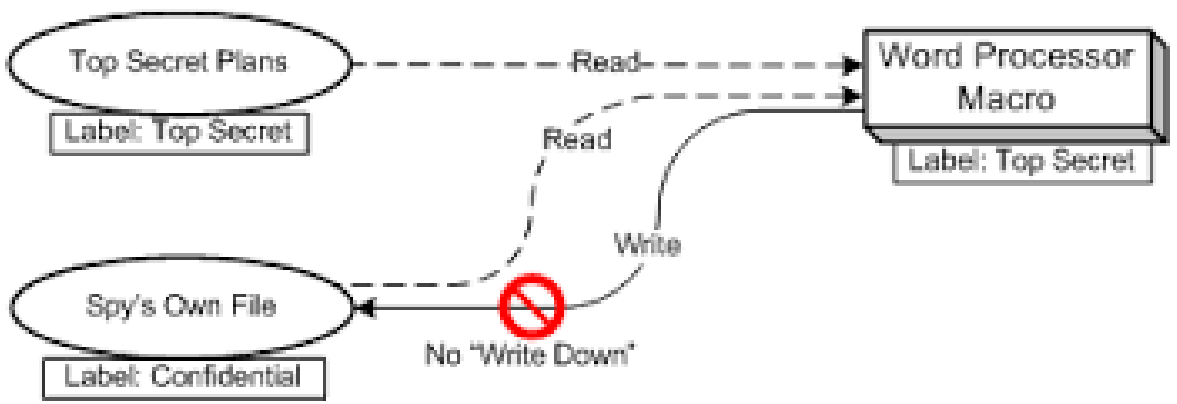
- **No read up**: A subject can only read an object of less or equal security level known as **simple security property (ss-property)**.



The Bell-LaPadula Model

A multilevel secure system for confidentiality must enforce

- **No write down**: A subject can only write into an object of greater or equal security level known as ***-property**.



- **ds-property**: An individual (or role) may grant to another individual (or role) access to a document based on the owner's discretion.

The Bell-LaPadula Model

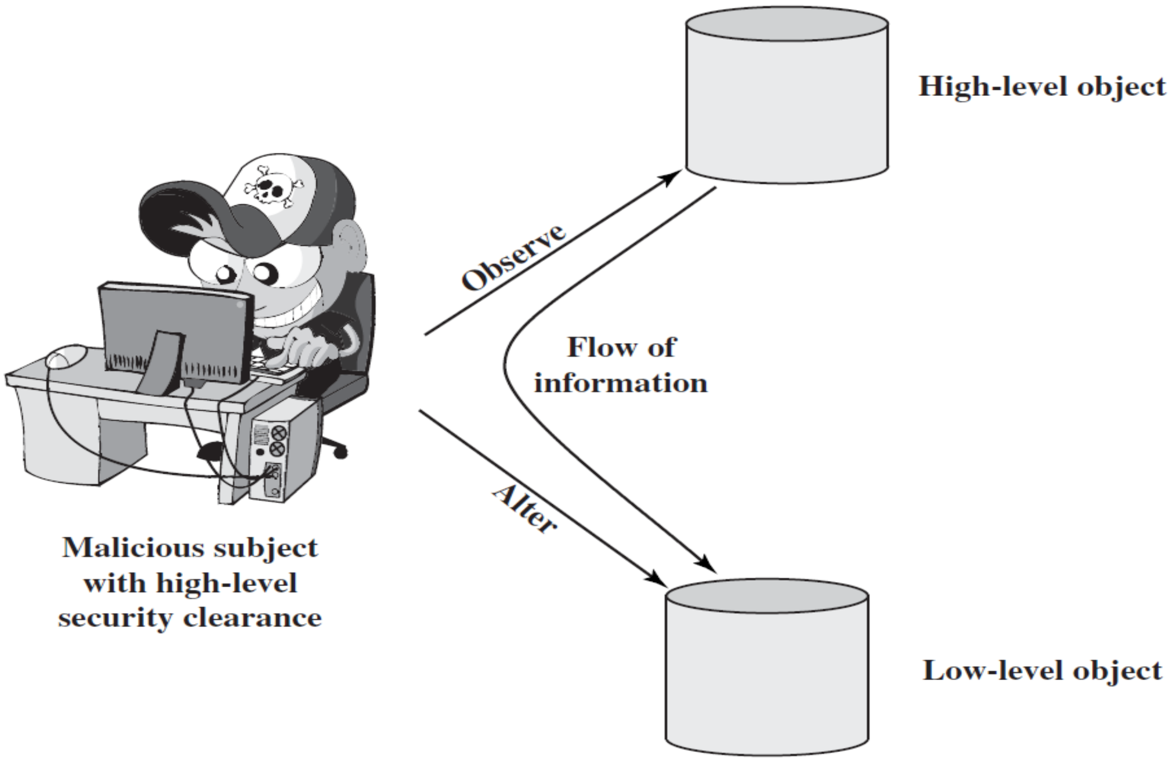


Figure 13.1 Information Flow Showing the Need for the *-Property

- ss-property and *-property provide confidentiality form of mandatory access control.
- All of the three properties provide discretionary access control .

The Bell-LaPadula Model

Formal description of the model

Based on the current state of the system (b, M, f, H)

- b: current access set that is (subject, object, access mode) = (s, o, a)
- M: Access matrix. The matrix element M_{ij} records the access modes in which subject S_i is permitted to access object O_j.
- f: Level function. Assigns a security level to each subject and object. $f_o(O_j)$ classification level of object O_j. $f_s(S_i)$ security clearance of subject S_i. $f_c(S_i)$ current security level of subject S_i.
- H: Hierarchy. A directed rooted tree whose nodes correspond to objects in the system.

The Bell-LaPadula Model

Formal description of the model

Three BPL 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) is current access (is in b) where access mode A_x is recorded in (S_i, O_j) element of M. (S_i, O_j, A_x) implies $A_x \in M[S_i, O_j]$

These properties are used to define confidentiality of secure system.

The Bell-LaPadula Model

- Formal definition of confidentiality
  - Current state (b, M, f, H) is secure if and only if every element of b satisfies the 3 properties.
  - The security state of the system is changed by any operation that causes a change any of the four components of the system, (b, M, f, H).
  - A secure system remains secure so long as any state change does not violate the 3 properties.
- BPL gives formal theorems
  - Theoretically possible to prove system is secure
  - In practice usually not possible

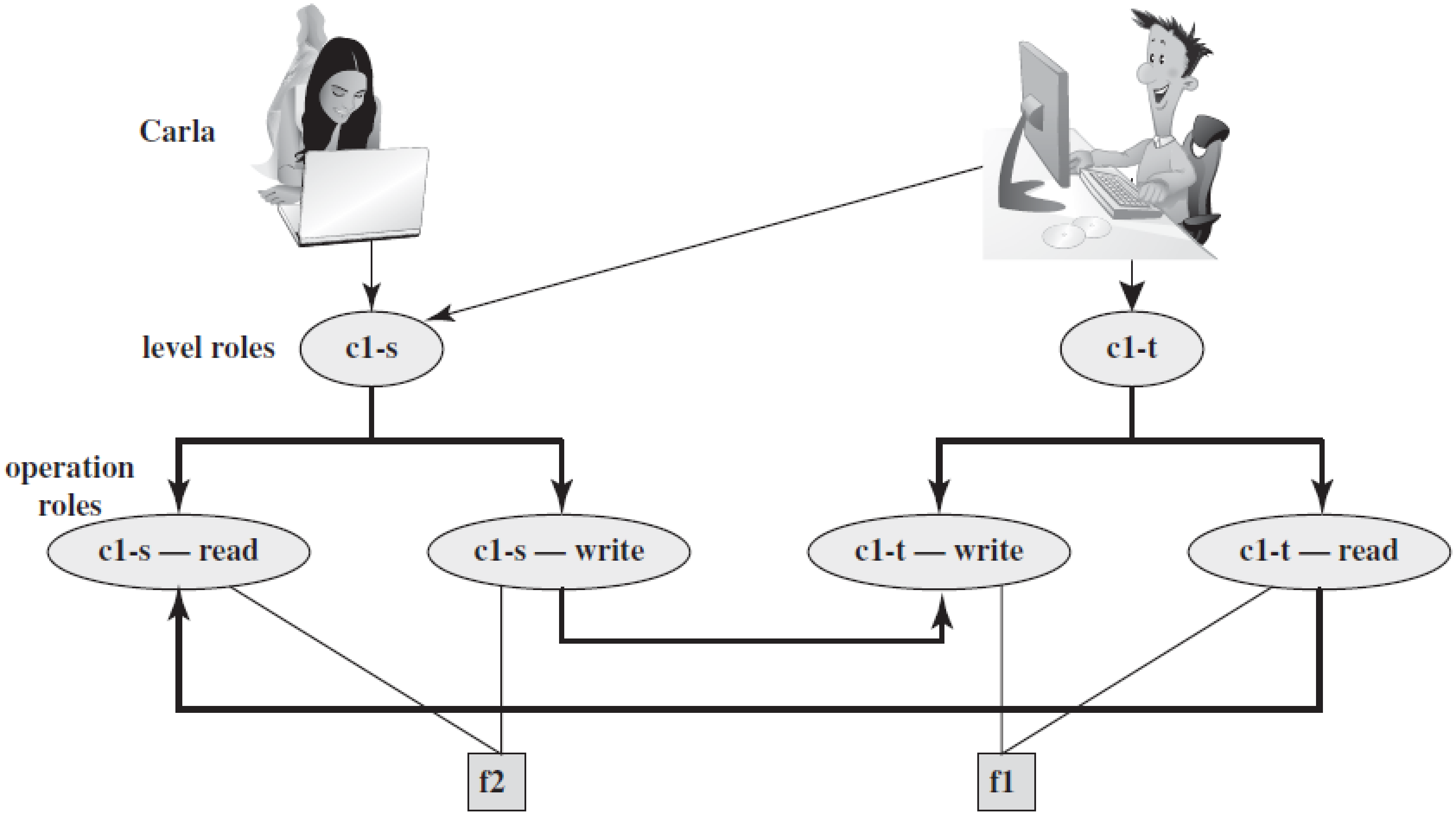
The Bell-LaPadula Model

BPL rules based on abstract operations

- Get access: add a triple (S,O,A) to current access set b.
- Release access: remove a triple (S,O,A) from the current access set b.
- Change object level: change $f_o(O_j)$
- Change current level: change $f_c(S_i)$
- Give access permission: Add an access mode to some entry of the access permission matrix *M*.
- Rescind (cancle) access permission: Delete an access mode from some entry of M.
- Create an object: Attach an object to the current tree structure H as a leaf.
- Delete a group of objects: Detach from H an object and all other objects beneath it in the hierarchy.

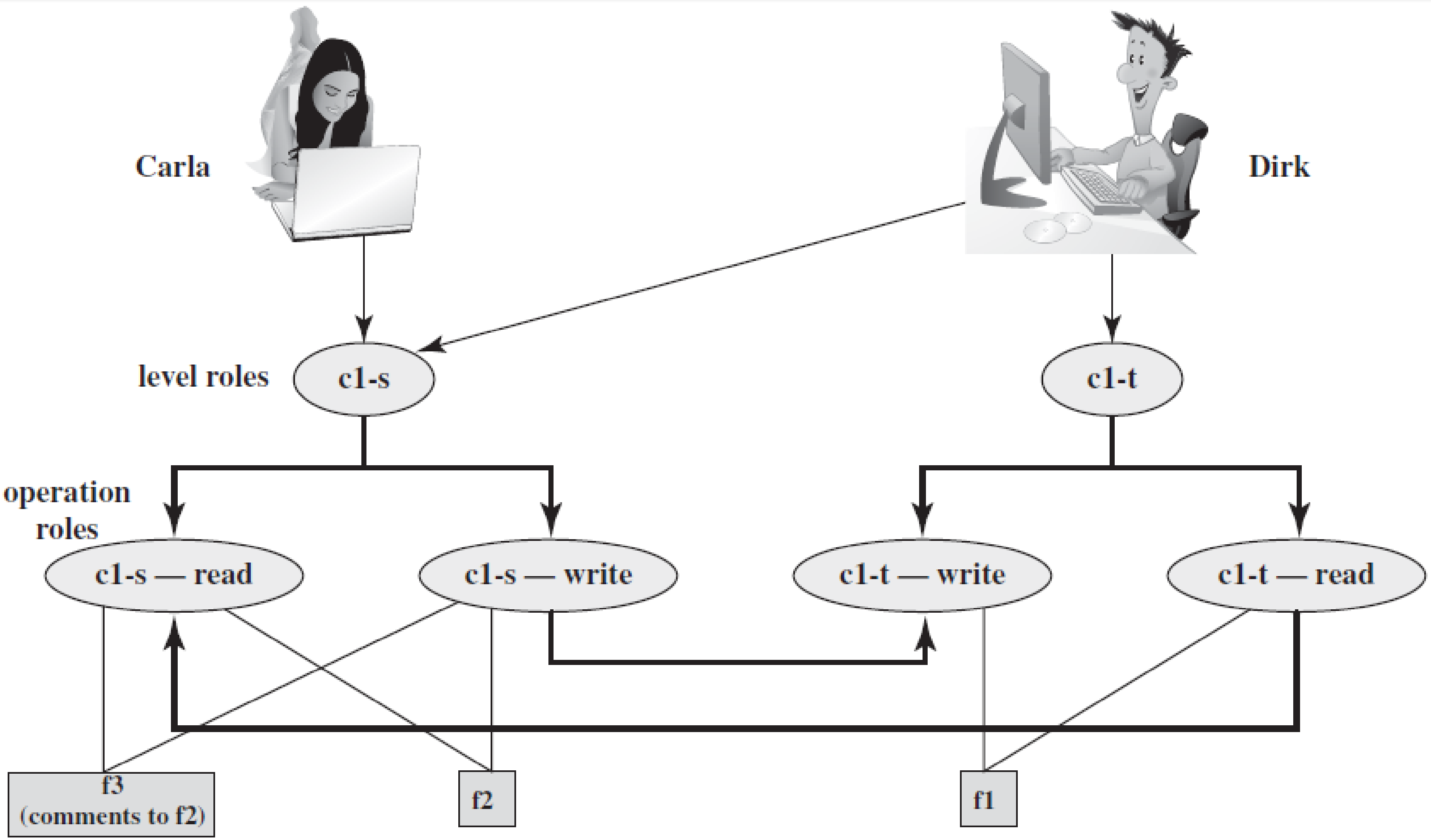
The Bell-LaPadula Model

Example of BPL Use-1



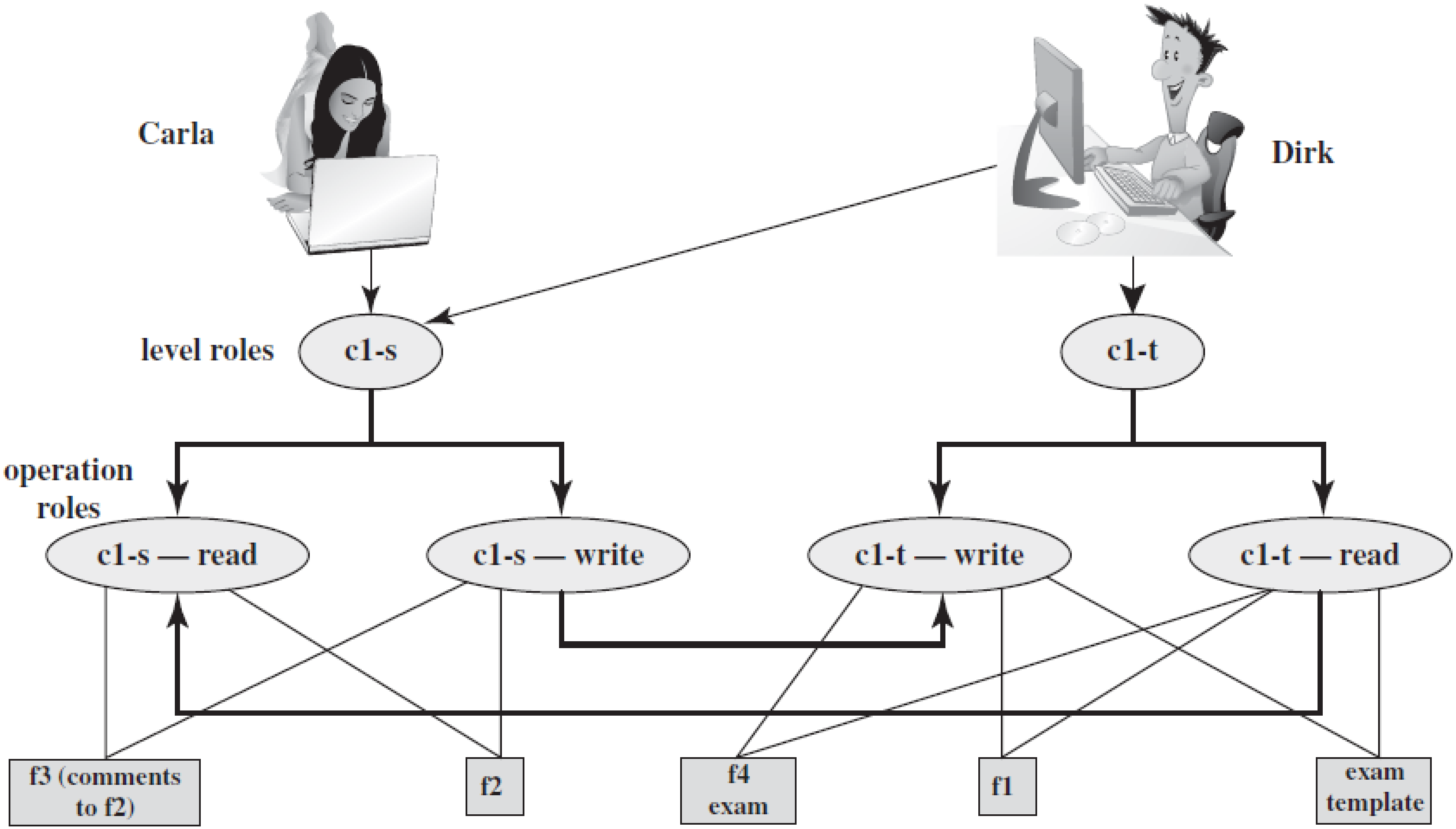
(a) Two new files are created: f1: c1-t; f2: c1-s

The Bell-LaPadula Model
Example of BPL Use-2



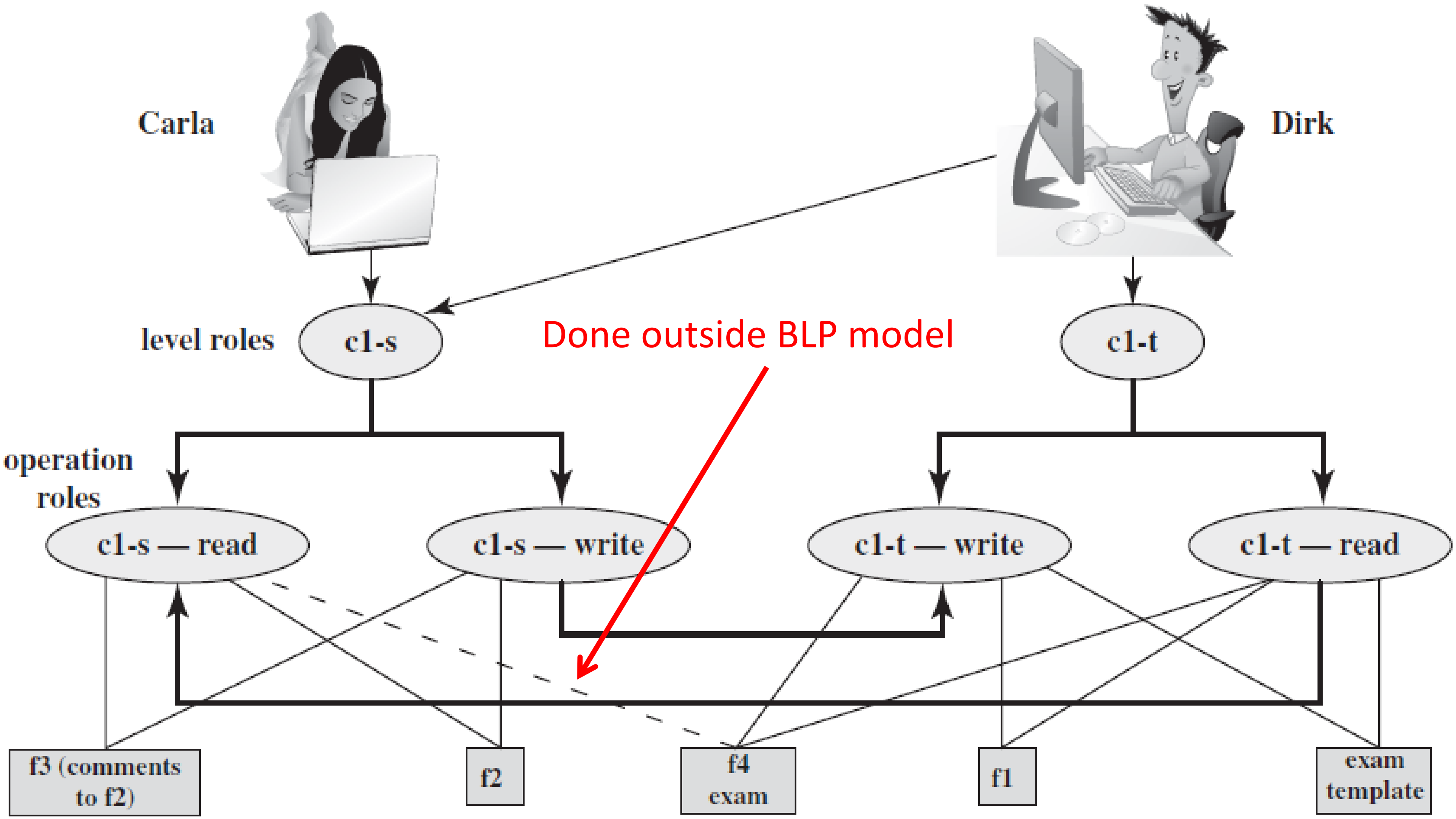
(b) A third file is added: f3: c1-s

The Bell-LaPadula Model
Example of BPL Use-3



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

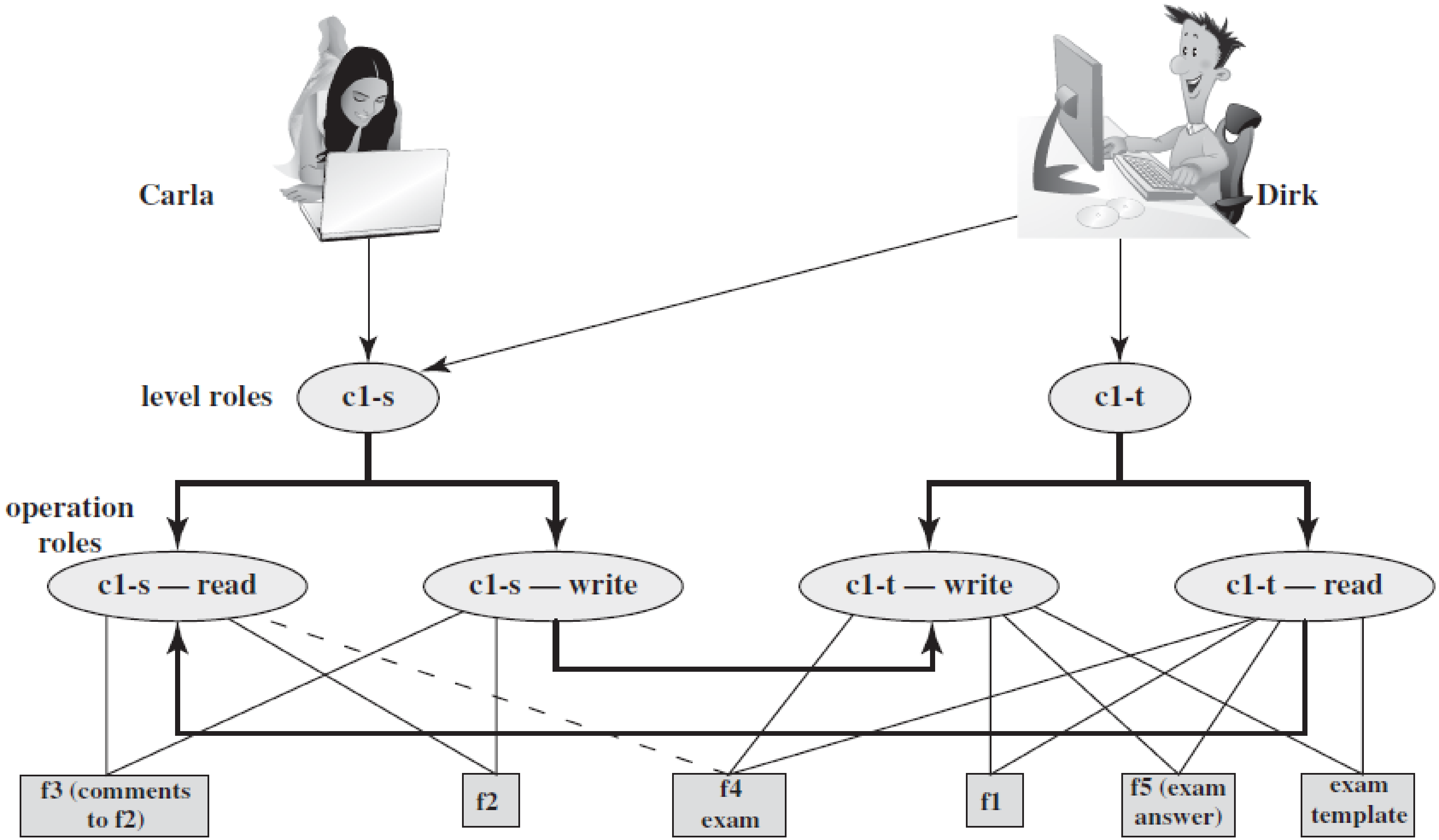
The Bell-LaPadula Model
Example of BPL Use-4



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

The Bell-LaPadula Model

Example of BPL Use-5



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

The Bell-LaPadula Model

BPL limitations

- No provision for downgrading
- Can only edit a file at one security level while reading at same or lower level
- Classification creep occurs if a documents consolidates from many sources and levels
- Usability and implementation problems

Biba Integrity Model

- BPL deals with confidentiality and is concerned with unauthorized disclosure of information, whereas, Biba model deals with integrity and is concerned with the unauthorized modification of data.

- In Biba, data are visible to users at multiple or all security levels but should only be modified by authorized agents.

- Each subject and object is assigned an integrity level, denoted as *I(S)* and *I(O)*.

Biba Integrity Model

Access Modes

- Modify: write or update
- Observe: read
- Execute
- Invoke: communication from one subject to another

Biba Integrity Model

The strict integrity policy rules:

- Simple Integrity: A subject can modify an object only if $I(S) \geq I(O)$
- Integrity confinement: A subject can read an object only if $I(S) \leq I(O)$
- Invocation property: A subject can invoke another subject only if $I(S_1) \geq I(S_2)$

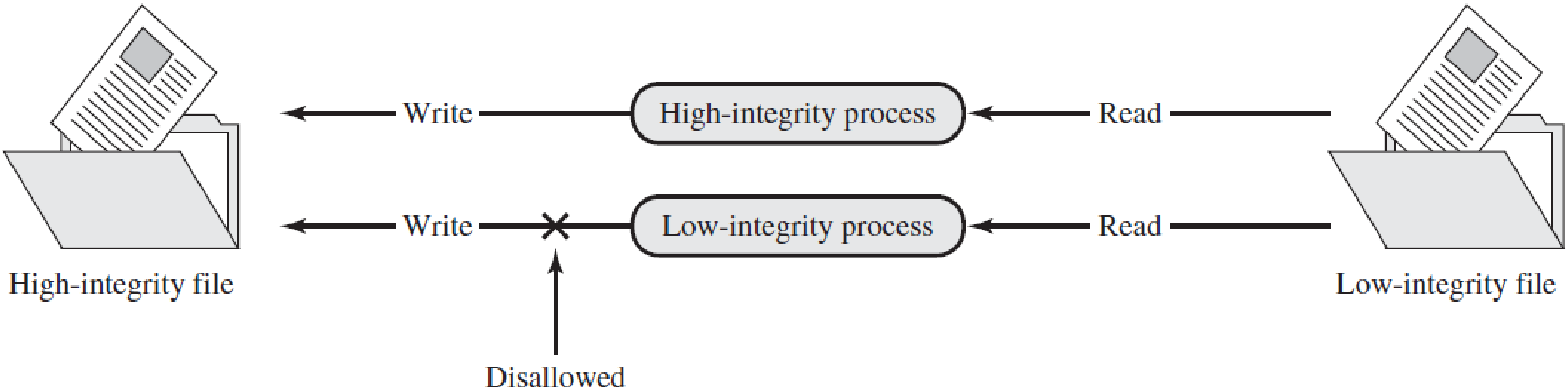


Figure 13.4 Contamination with Simple Integrity Controls

Clark-Wilson Integrity Model

- Clark-Wilson Model (CWM) is aimed at commercial rather than military applications and it closely models real commercial operations.
- The concepts of CWM
  - Well-formed transactions: A user should not manipulate data arbitrarily, but only in constrained ways that preserve the integrity.
  - Separation of duty among users: Any person permitted to create or certify a well-formed transaction may not be permitted to execute it.

Clark-Wilson Integrity Model

- The principle components of the model:
  - Constrained data items (CDIs): Subject to strict integrity controls
  - Unconstrained data items (UDIs): Unchecked data items
  - Integrity verification procedures (IVPs): Assure that all CDIs conform to some application specific model of integrity and consistency
  - Transformation procedures (TPs): Transactions that change the set of CDIs from one consistent state to another.
- CWM enforces integrity by means of certification and enforcement rules of TPs.

Chinese Wall Model

- The model was developed for commercial applications in which conflict of interests can arise.

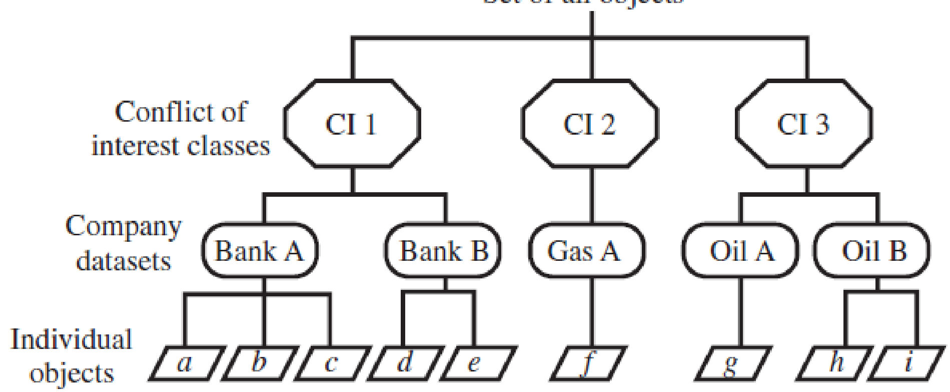
- Does not assign security levels -> not true multilevel security.
- Uses history of a subject's previous accesses to determine access control.

Chinese Wall Model

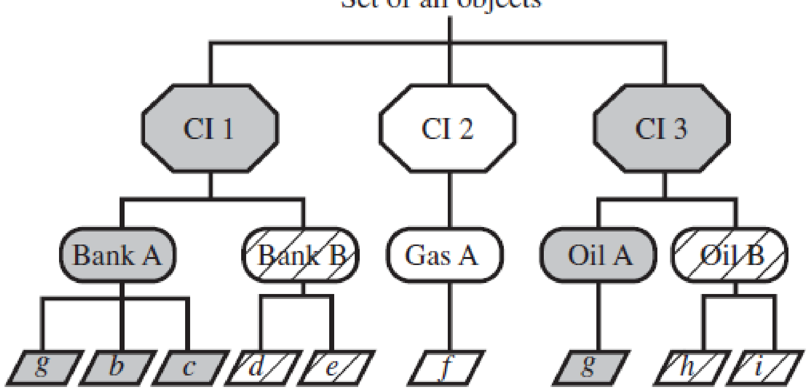
The elements of the model

- Subjects: users and processes
- Information: corporate information with a hierarchy of three levels
  - Objects: items of information, each concerning a single corporation
  - Dataset (DS): all objects
  - Conflict of interest (CI) class: all datasets whose corporations are in competition
- Access rules

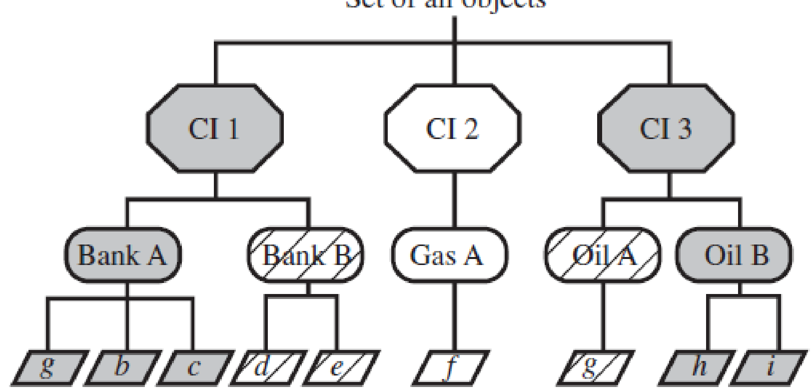
Chinese Wall Model



(a) Example set



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



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

The Concept of Trusted Computing

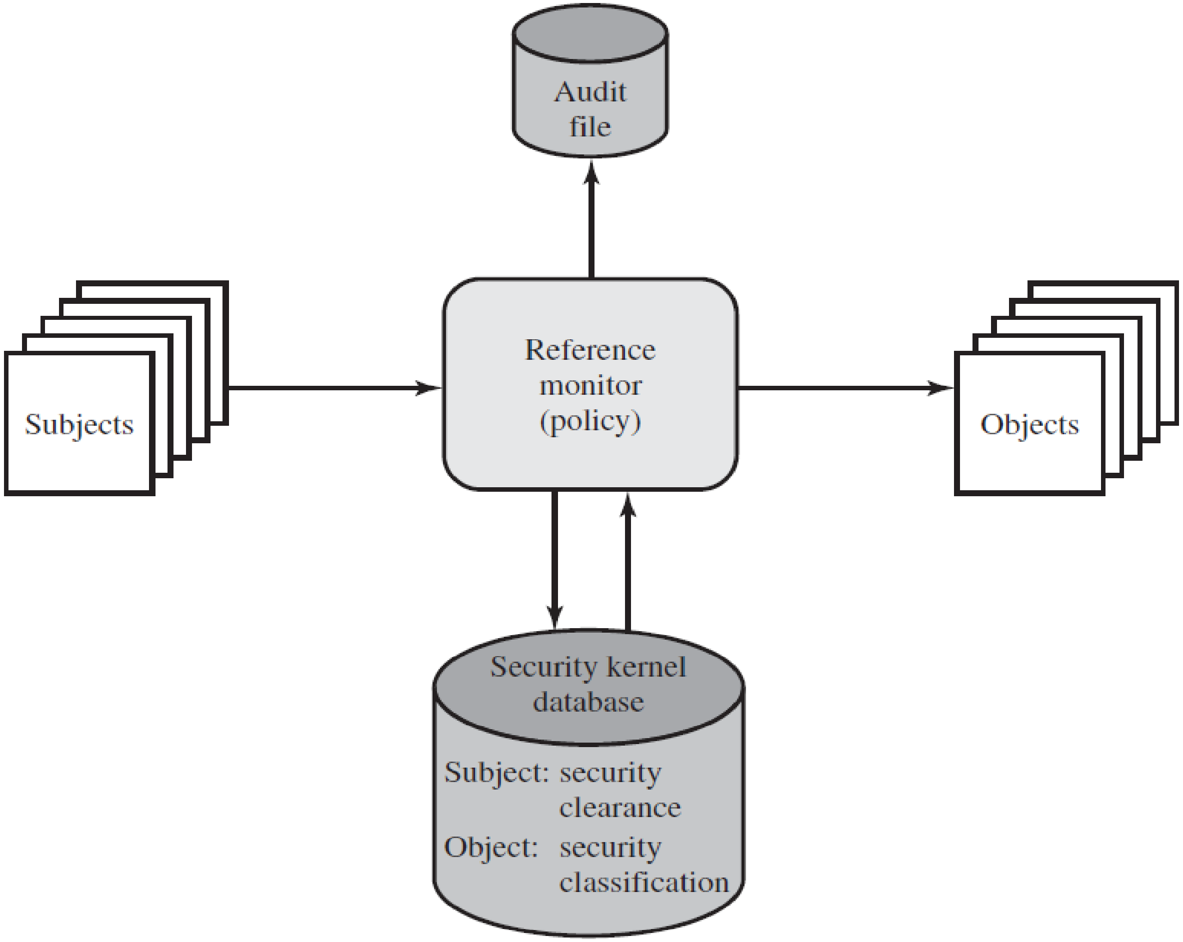
- Early 1970s
- U.S. Department of Defense
- Initially did not gain a serious foothold in the commercial market
- Recently, the interest reemerged

The Concept of Trusted Computing

- Trust: the extent to which someone who relies on a system can have a confidence that the system meets its specifications.
- Trusted system: A system believed to enforce a given set of attributes to a stored degree of assurance.
- Trustworthiness: Assurance that a system deserves to be trusted, such that the trust can be guaranteed in some convincing way, such as through formal analysis or code review.

The Concept of Trusted Computing

Reference Monitors is a controlling element in the hardware and OS of a computer that regulates the access of subjects to objects on the basis of security parameters of the subject and the object.

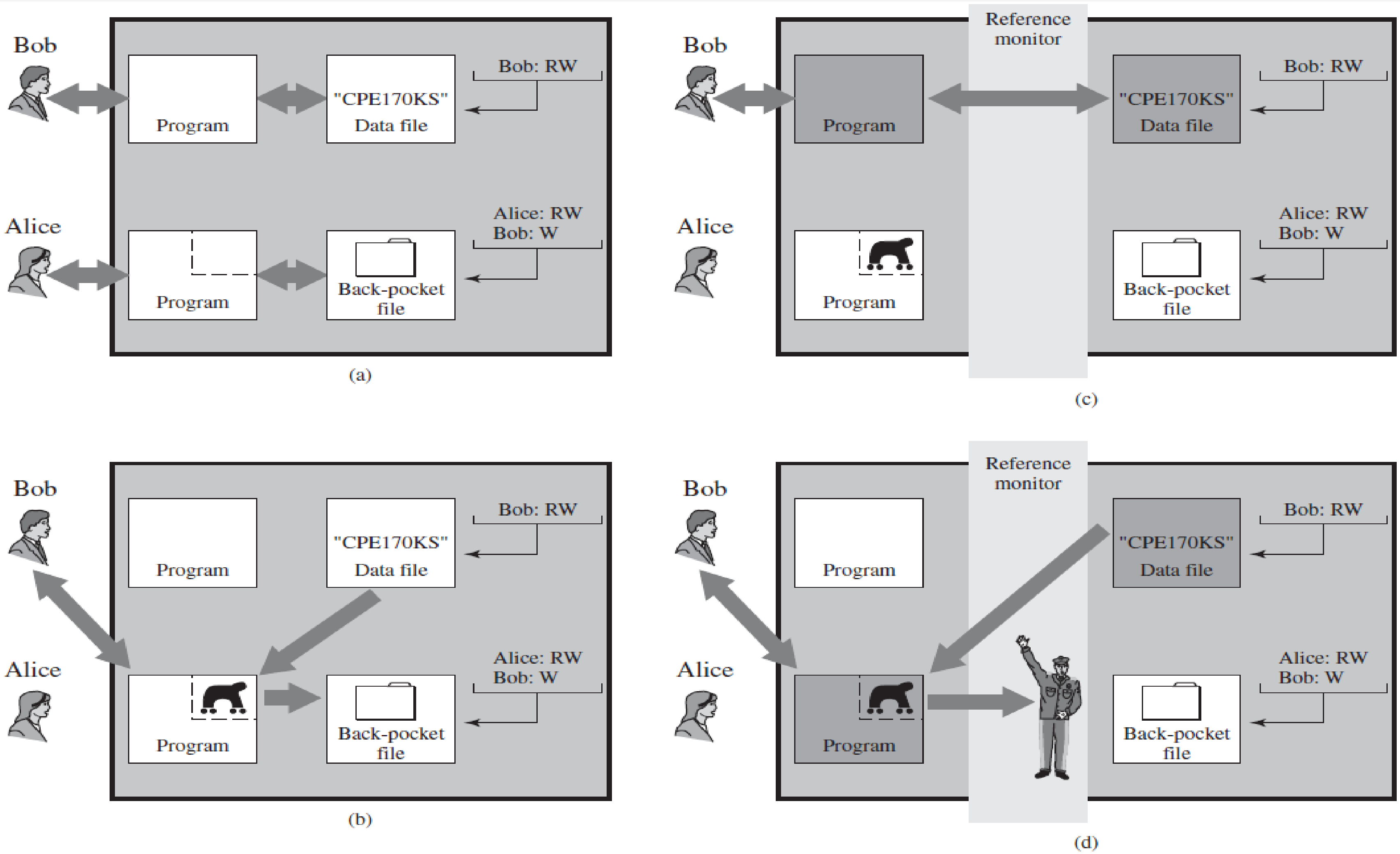


The Concept of Trusted Computing

Properties of Reference Monitor

- Complete mediation: security rules are enforced on every access.
- Isolation: protected from unauthorized modification.
- Verifiability: prove that the reference monitor does complete mediation and isolation correctly.

The Concept of Trusted Computing
Trojan Horse Example



Multilevel Security

- Multilevel Secure (MLS): A class of system that has system resources at more than one security level 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.

- Multilevel Security for RBAC: RBAC can implement BPL MLS rules given:
  - Security constraints on users
  - Constraints on read/write permissions
  - Read and write level role access definitions
  - Constraint on user-role assignment

Multilevel Security

Database Security and MLS

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)

Multilevel Security

Database Security and MLS

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

(b) Classified by element

Multilevel Security
Database Security and MLS

Read Access

- DBMS enforces simple security rule (no read up)
- Easy if classification granularity of all database
- Inference problems if have common granularity
  - Query on restricted data

    SELECT Ename FROM Employee WHERE Salary > 50K
  - Solution is to check access of all data
- Problems with row granularity
  - null response indicates restricted/empty result

Multilevel Security
Database Security and MLS

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
- Avoided by using a classification granularity of database or table

Trusted Computing and Trusted Platform Module

- Trusted Platform Module (TPM)
  - An industry standard developed by Trusted Computing Group
  - A hardware module
  - At the heart of a hardware/software approach of trusted computing
  - Trusted Computing (TC) is used to refer such hardware and software

- TC employs a TPM chip in personal computer motherboard or a smart card or integrated into the main processor, together with HW and SW that in some sense has been approved or certified to work with the TPM.

Trusted Computing and Trusted Platform Module

- Trusted Computing Approach: TPM generates keys that it shares with vulnerable components that pass data around the system. The keys can be used to encrypt the data flow through the machine.
- TPM works with TC-enabled software to assure that data it receives are trustworthy.
- Trusted Computing Services
  - Authenticated boot
  - Certification
  - Encryption

Trusted Computing and Trusted Platform Module

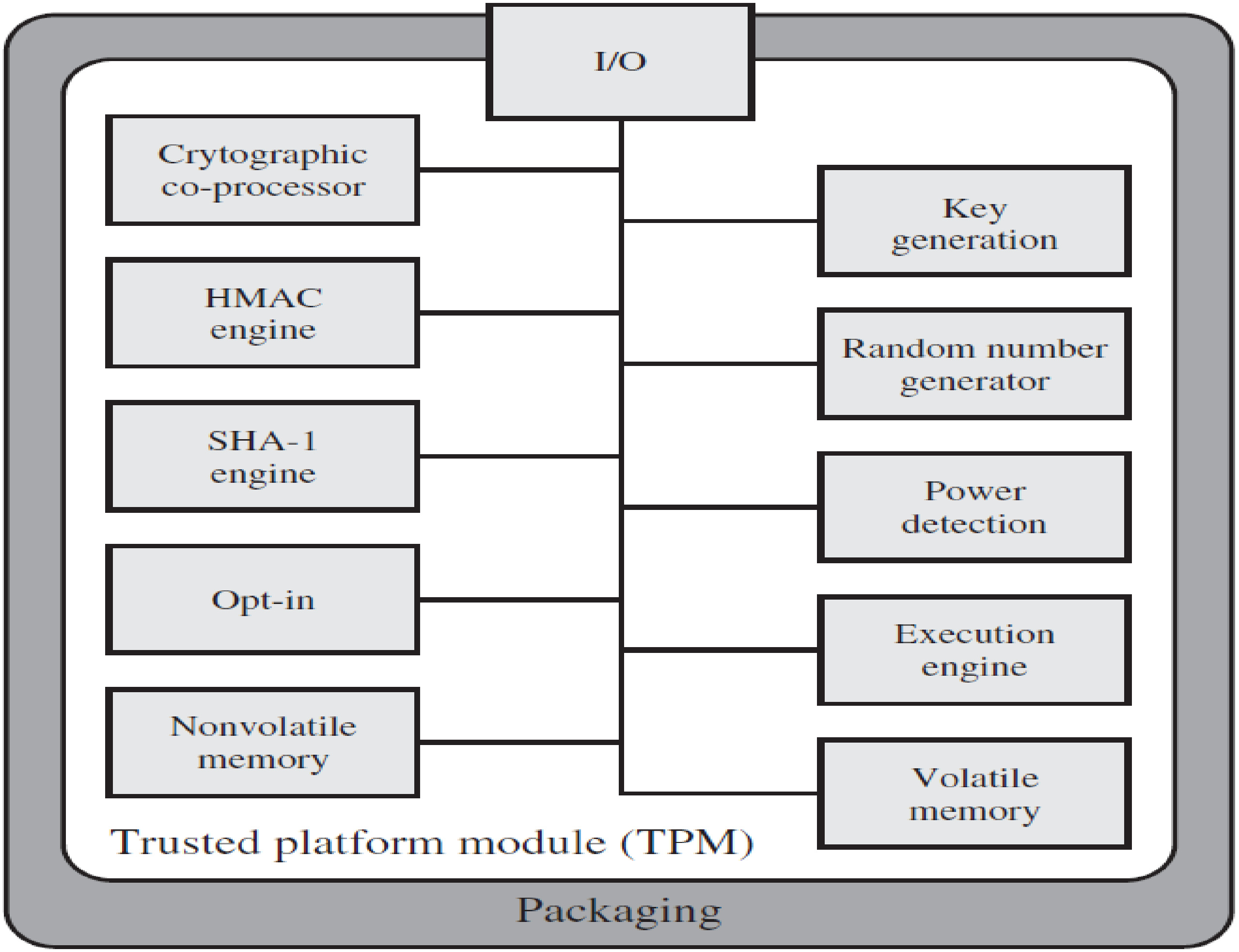


Figure 13.12 TPM Component Architecture

Trusted Computing and Trusted Platform Module

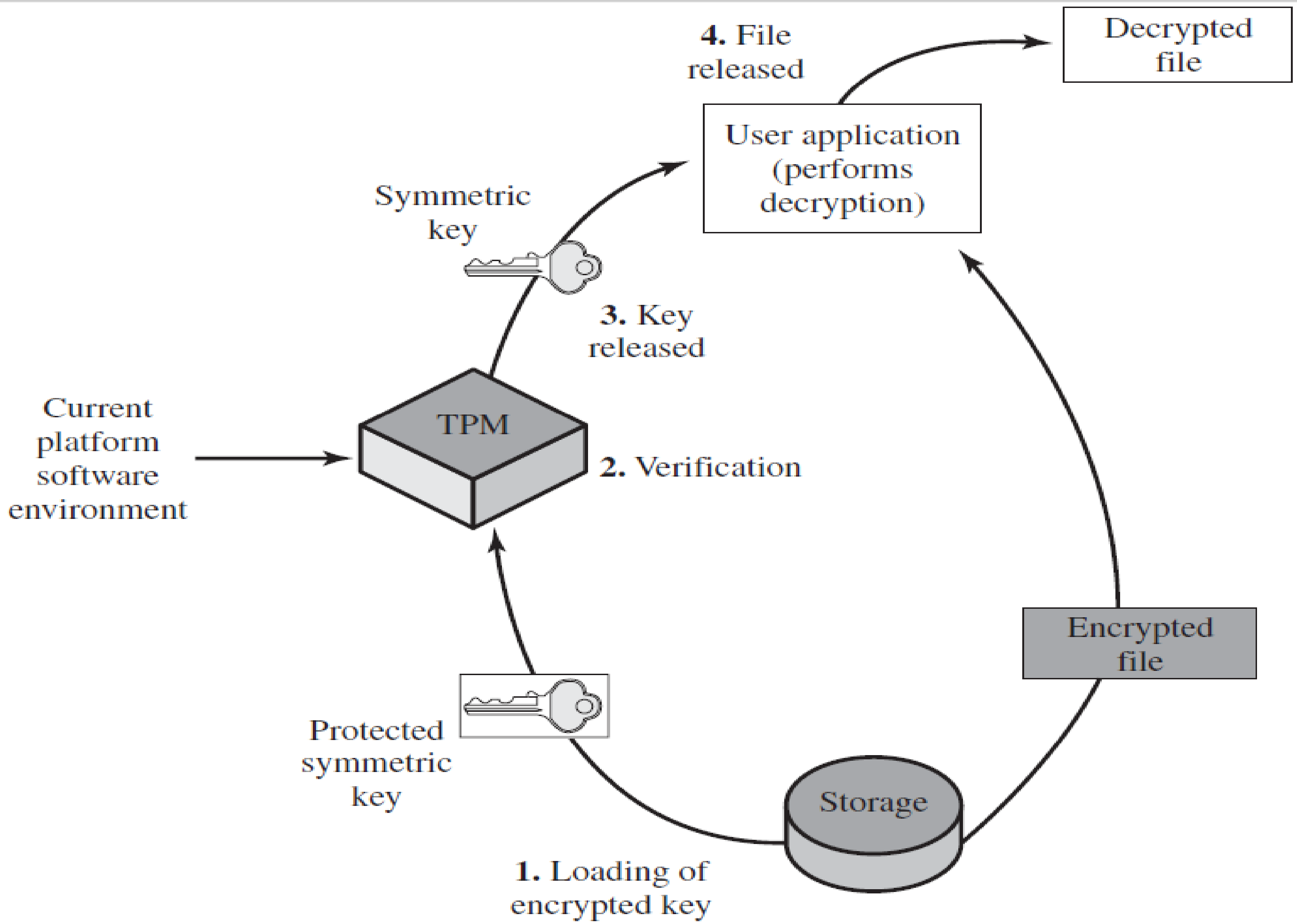


Figure 13.13 Decrypting a File Using a Protected Key

Summary

- BPL security model
- Biba, Clark-Wilson, and Chinese Wall models
- The concept of trusted computing
- Multilevel security
- Trusted computing