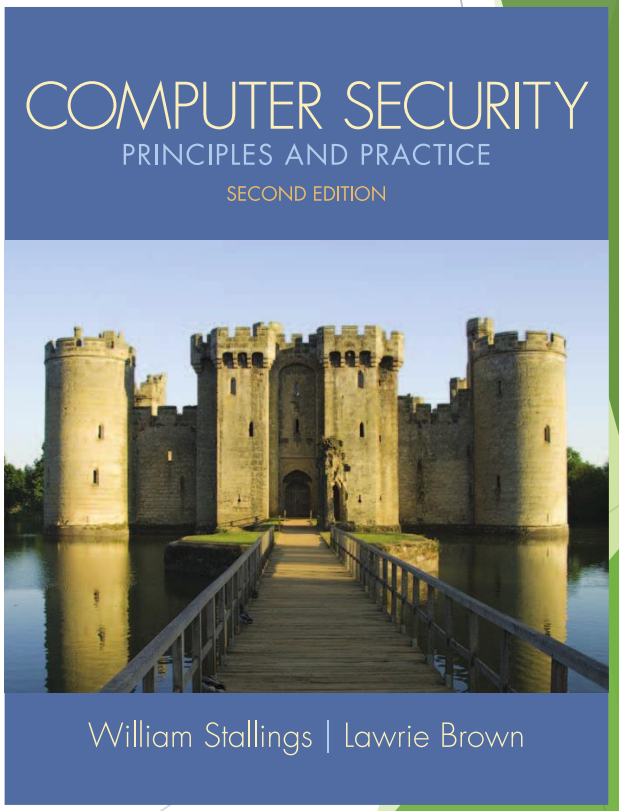
Chapter 13
Trusted Computing and Multilevel Security

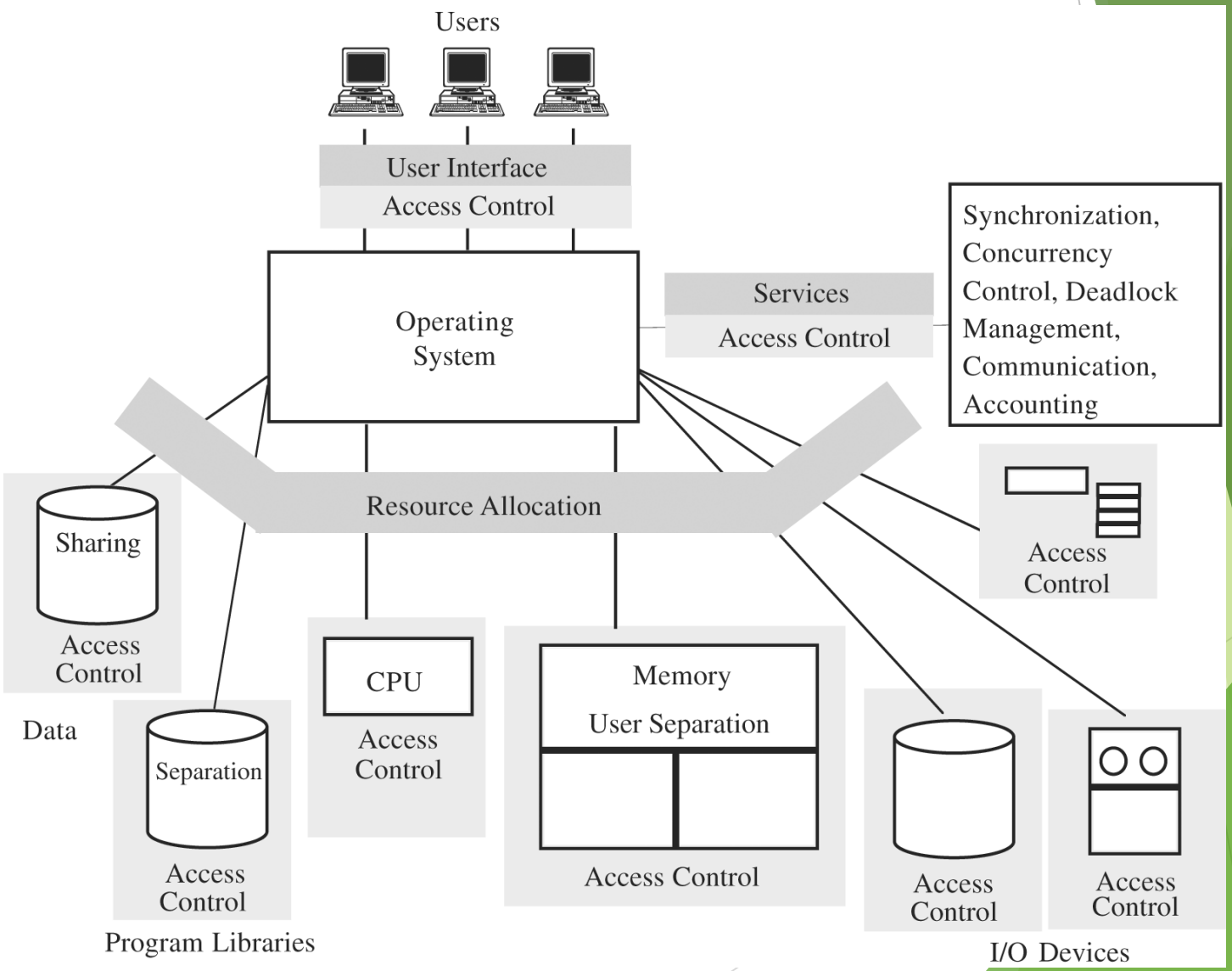


modified from slides of **Lawrie Brown** and **Hesham El-Rewini**

Computer Security Models

- two fundamental computer security facts:
  - all complex software systems have eventually revealed flaws or bugs that need to be fixed
  - it is extraordinarily difficult to build computer hardware/software not vulnerable to security attacks
- problems involved both design and implementation
  - led to development of formal security models

Trusted OS Functions



Bell-LaPadula (BLP) Model

- formal model for access control
  - developed in 1970s
- *subjects* and *objects* are assigned a security class
  - a *subject* has a *security clearance*
  - an *object* has a *security classification*
  - form a hierarchy and are referred to as security levels
    - top secret > secret > confidential > restricted >unclassified
  - security classes control the manner by which a subject may access an object

BLP Model Access Modes

- READ
  - the subject is allowed only read access to the object
- APPEND
  - the subject is allowed only write access to the object
- WRITE
  - the subject is allowed both read and write access to the object
- EXECUTE
  - the subject is allowed neither read nor write access to the object but may invoke the object for execution

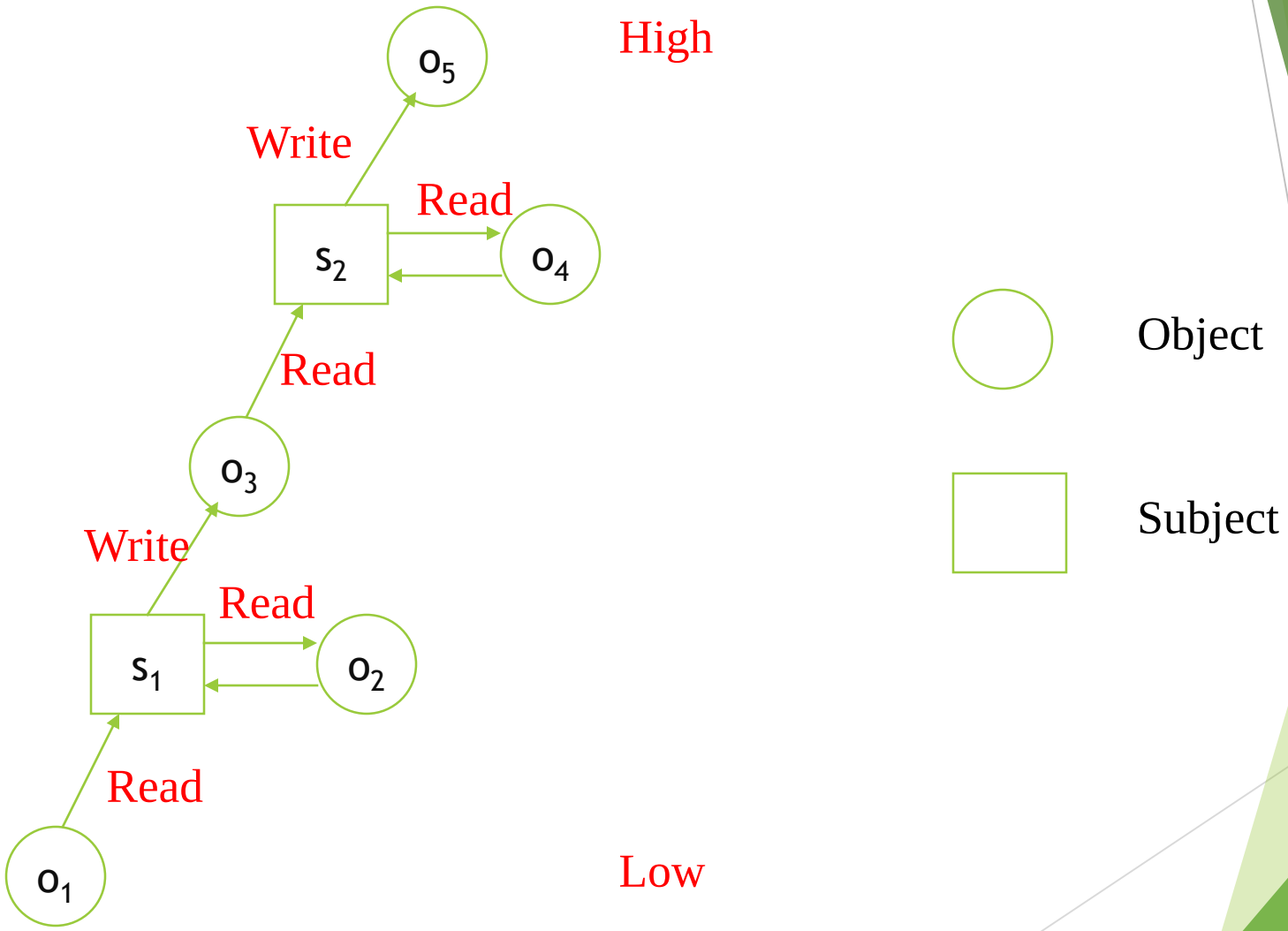
BLP Formal Description

- based on current state of system (b, M, f, H):
  - current access set **b**: triples of (s, o, a)
    - subject **s** has current access to object **o** in access mode **a**
  - access matrix **M**: matrix of M_{ij}
    - access modes of subject S_i to access object O_j
  - level function **f**: security level of subjects and objects
    - $f_o (O_j)$ is the classification level of object O_j
    - $f_s (S_i)$ is the security clearance of subject S_i
    - $f_c (S_i)$ is the current security level of subject S_i
  - hierarchy **H**: a directed rooted tree of objects

BLP Formal Description

- three BLP properties:
  - ss-property: $\forall$ (S_i, O_j, read) has f_c (S_i) $\geq$ f_o (O_j)
  - *-property: $\forall$ (S_i, O_j, append) has f_c (S_i) $\leq$ f_o (O_j) and
    $\forall$ (S_i, O_j, write) has f_c (S_i) = f_o (O_j)
  - ds-property: current (S_i, O_j, A_x) implies $A_x \in M[S_i\ O_j]$
- BLP gives formal theorems
  - theoretically possible to prove system is secure
  - in practice usually not possible

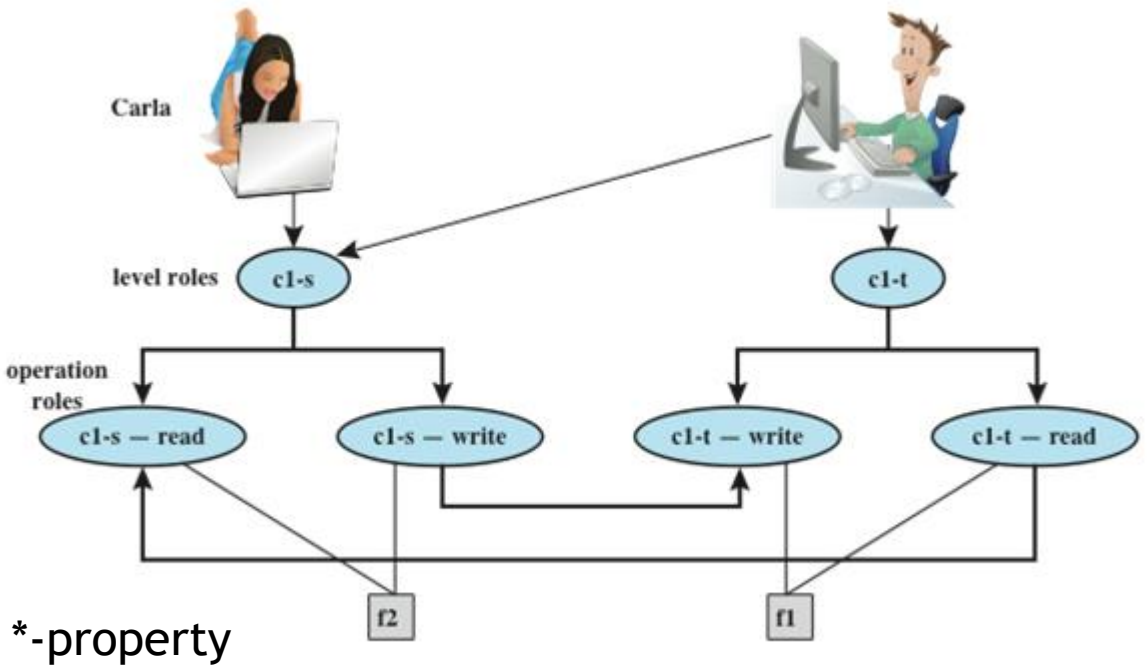
Illustration



BLP Rules

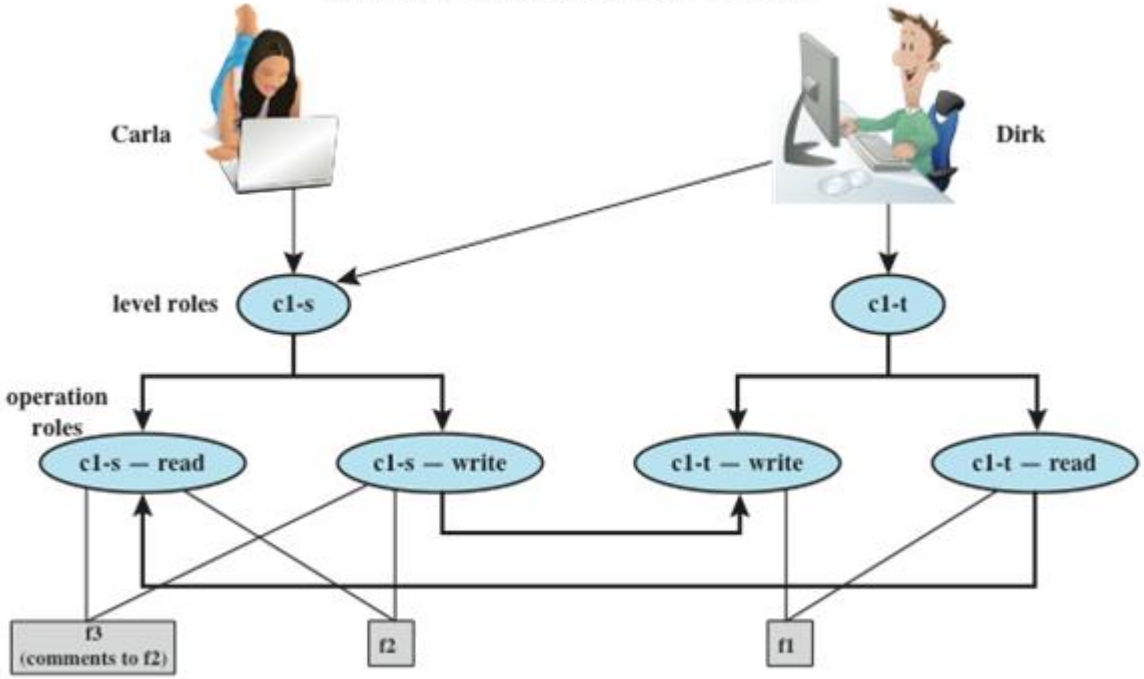
1. get access
2. release access
3. change object level
4. change current level
5. give access permission
6. rescind access permission
7. create an object
8. delete a group of objects

BLP Example



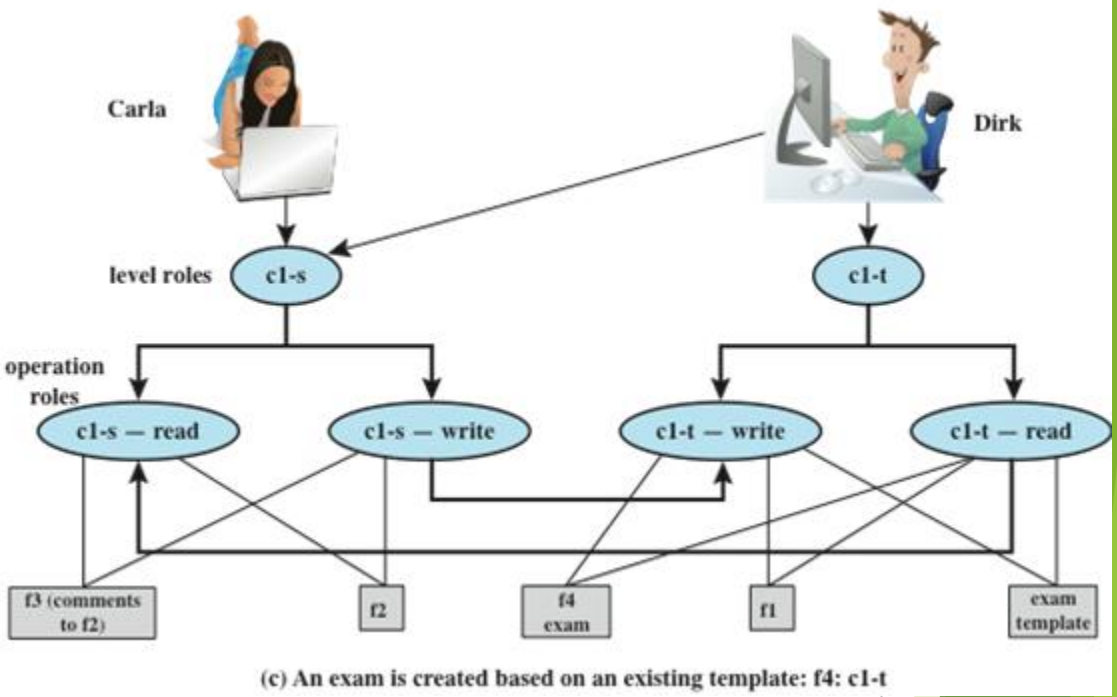
*-property

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



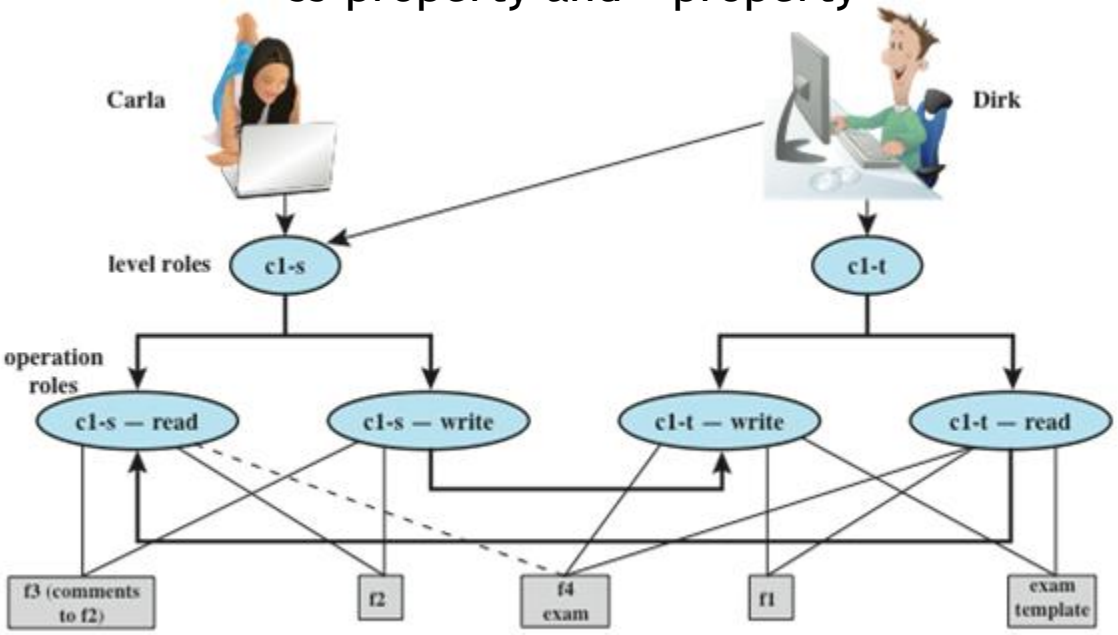
(b) A third file is added: f3: c1-s

BLP Example



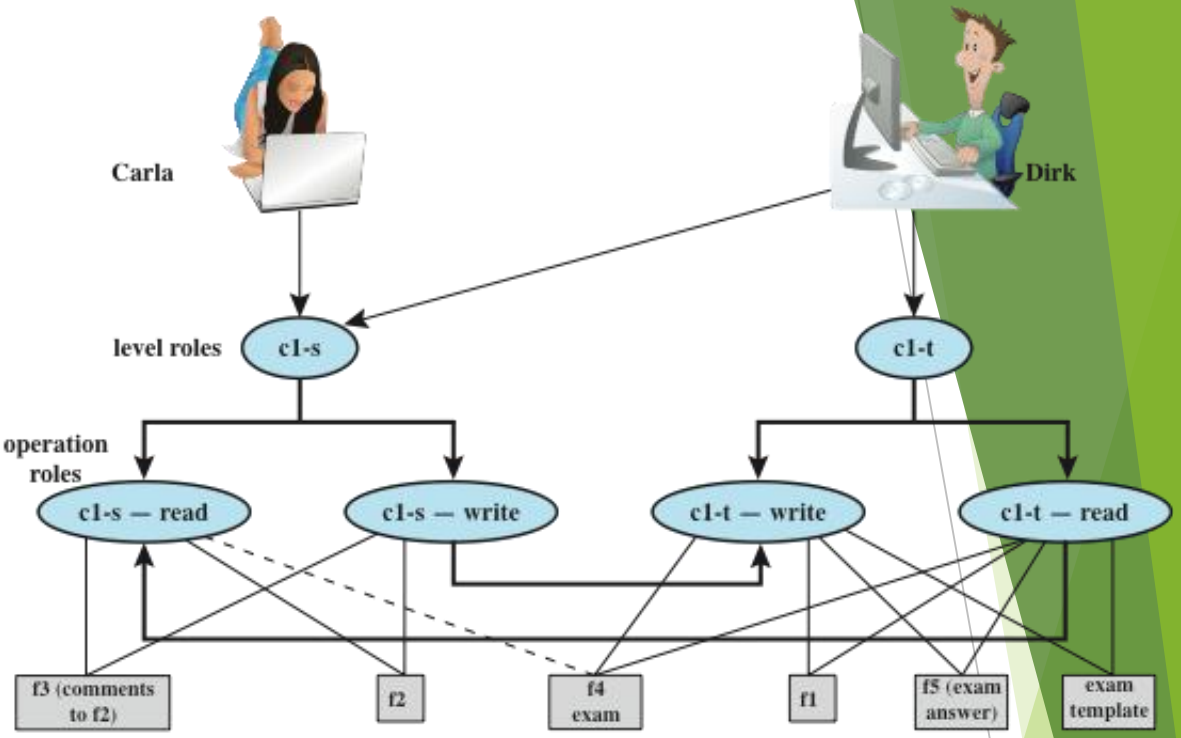
(c) An exam is created based on an existing template: f4: c1-t

ss-property and *-property



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

BLP Example



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

"**downgrade**" in a controlled and monitored manner

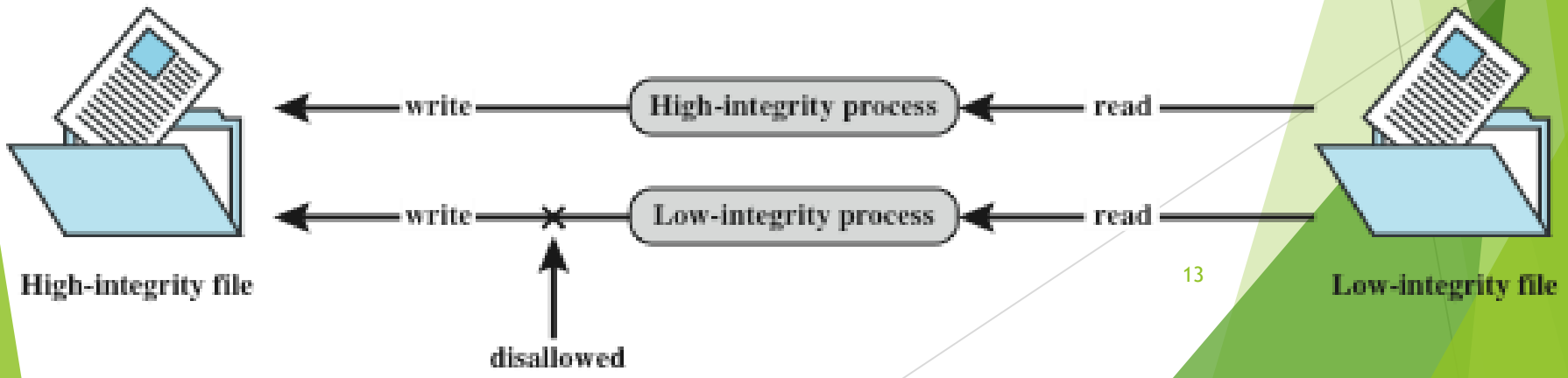
"**classification creep**": information from a range of sources and levels

Confidentiality or Integrity

"**covert channels**": (untrusted) low classified executable data
allowed to be executed by a high clearance (trusted) subject

Biba Integrity Model

- strict integrity policy:
  - **Modify**: To write or update information in an object
  - **Observe**: To read information in an object
  - **Execute**: To execute an object
  - **Invoke**: Communication from one subject to another
  - simple integrity: $I(S) \geq I(O)$
  - integrity confinement: $I(S) \leq I(O)$
  - invocation property: $I(S1) \geq I(S2)$



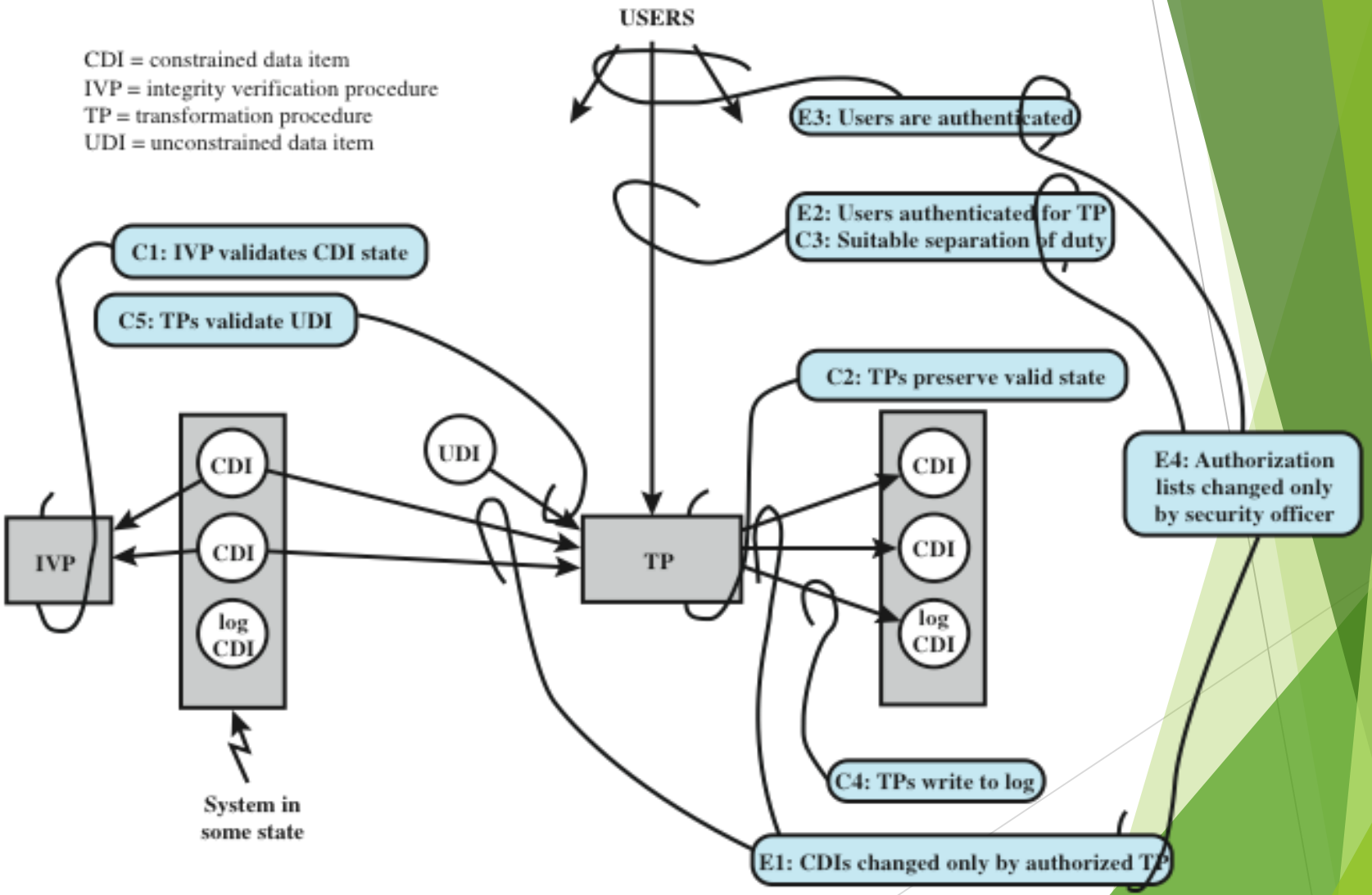
Clark-Wilson Integrity Model

- closely models commercial operations
  - Well-formed transactions
    - A user should not manipulate data arbitrarily
  - Separation of duty among users
    - A person who create or certify a well-formed transaction may not execute it

Clark-Wilson Integrity Model

- Principal components of the model
  - Constrained data items (CDIs)
    - Subject to strict integrity controls
  - Unconstrained data items (UDIs)
    - Unchecked data items
  - Integrity verification procedures (IVPs):
    - Intended to assure that all CDIs conform to some application-specific model of integrity and consistency
  - Transformation procedures (TPs):
    - System transactions that change the set of CDIs from one consistent state to another

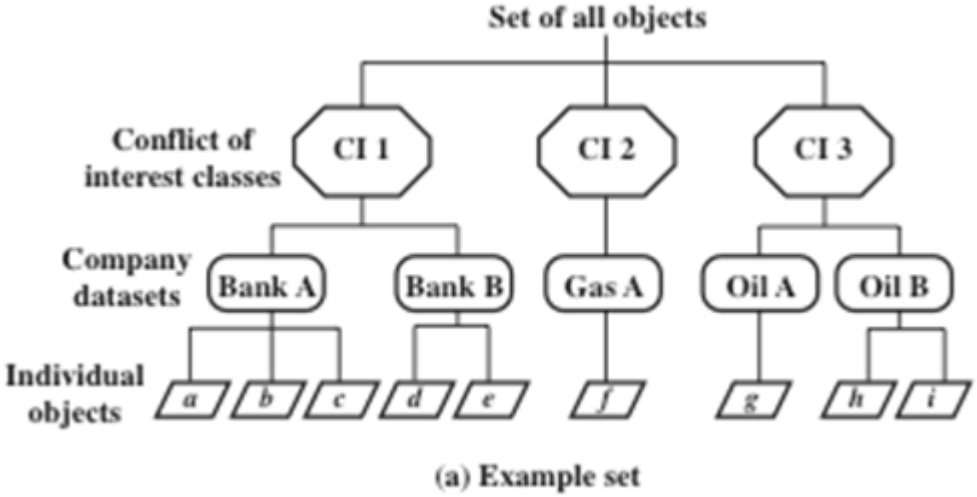
Clark-Wilson Integrity Model



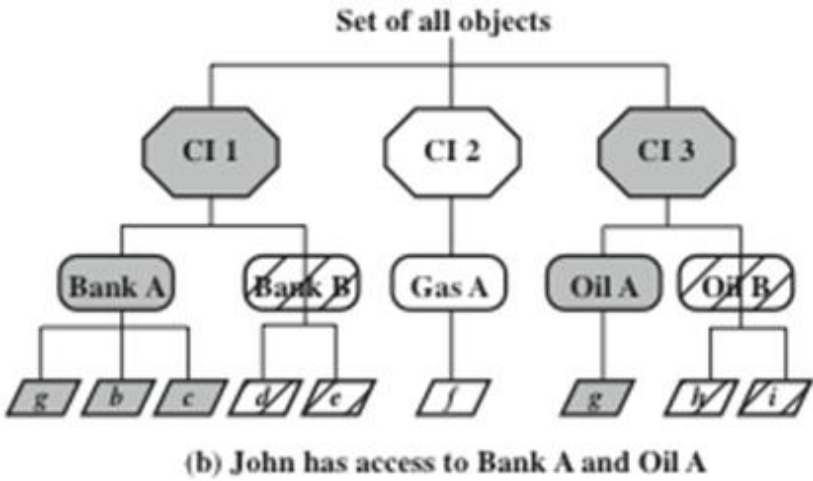
Chinese Wall Model

- integrity and confidentiality
- use both discretionary and mandatory access
  - **Subjects**: Active entities that may wish to access protected objects
  - **Information**: Information organized into a hierarchy
    - **Objects**: Individual items of information, each concerning a single corporation
    - **Dataset** (DS): All objects that concern the same corporation
    - **Conflict of interest** (CI) **class**: All datasets whose corporations are in competition
  - **Access rules**: Rules for read and write access

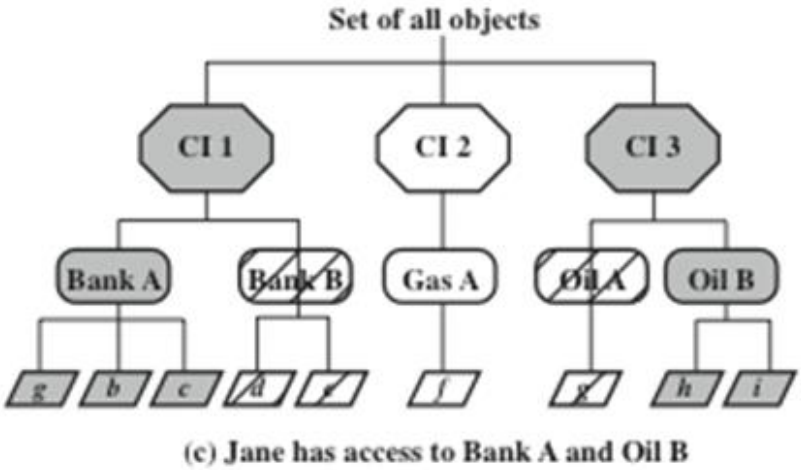
Chinese Wall Model



(a) Example set



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



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

Simple security rule: A S can read O only 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 information

***-property rule:** A S can write an O only if
- S can read O according to the simple security rule, **AND**
- All objects that S can read are in the same DS as O.

sanitized data

Multilevel Security (MLS)

- RFC 2828 defines multilevel security as follows:

"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,

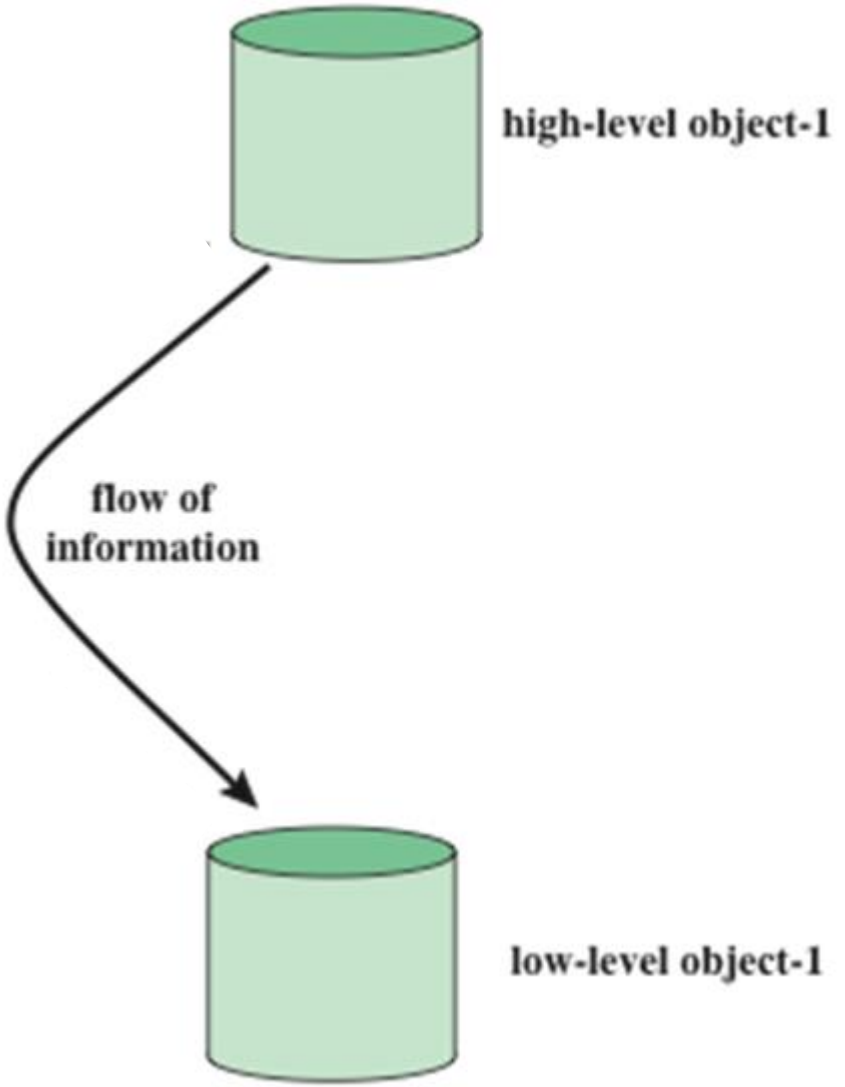
but is able to prevent each user from accessing resources for which the user lacks authorization."

Multi-Level Security

- no read up
  - subject can only read an object of less or equal security level
  - referred to as the *simple security property*
    - ss-property
- no write down
  - a subject can only write into an object of greater or equal security level
  - referred to as the *-property

Multi-Level Security

malicious subject
with high-level
security clearance



Multi-Level Security

- **ds-property :** An individual (or role) may grant to another individual (or role) access to a document
  - based on the owner's discretion,
    - constrained by the MAC rules
- site policy overrides any discretionary access controls

Database Security Classification (MLS Application)

Table

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

Column

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)

Database Classification

Row

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)

Element

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

Database Security: Read Access

- DBMS enforces simple security rule (no read up)
- easy if granularity is entire database or at 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 indicates restricted/empty result
- no extra concerns if have element granularity

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
  - polyinstantiation and insert multiple rows with same key, creates conflicting entries
- same alternatives occur on update
- avoid problem if use database/table granularity

Example of Polyinstantiation

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
James	8	35K	9664	U
Ziggy	8	67K	3054	R

Terminology Related to Trust

- **Trust**: The extent to which someone who relies on a system can have confidence that the system meets its specifications
  - i.e., that the system does what it claims to do and does not perform unwanted functions
- **Trusted system**: A system believed to enforce a given set of attributes to a stated 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.

Continued…

- **Trusted computer system**: A system that employs sufficient hardware and software assurance measures to allow its use for simultaneous processing of a range of sensitive or classified information.
- **Trusted computing base** (TCB): A portion of a system that enforces a particular policy
  - The TCB must be resistant to tampering and circumvention.
  - The TCB should be small enough to be analyzed systematically.
- **Assurance**: A process that ensures a system is developed and operated as intended by the system's security policy.
- **Evaluation**: Assessing whether the product has the security properties claimed for it.
- **Functionality**: The security features provided by a product.

Reference Monitors

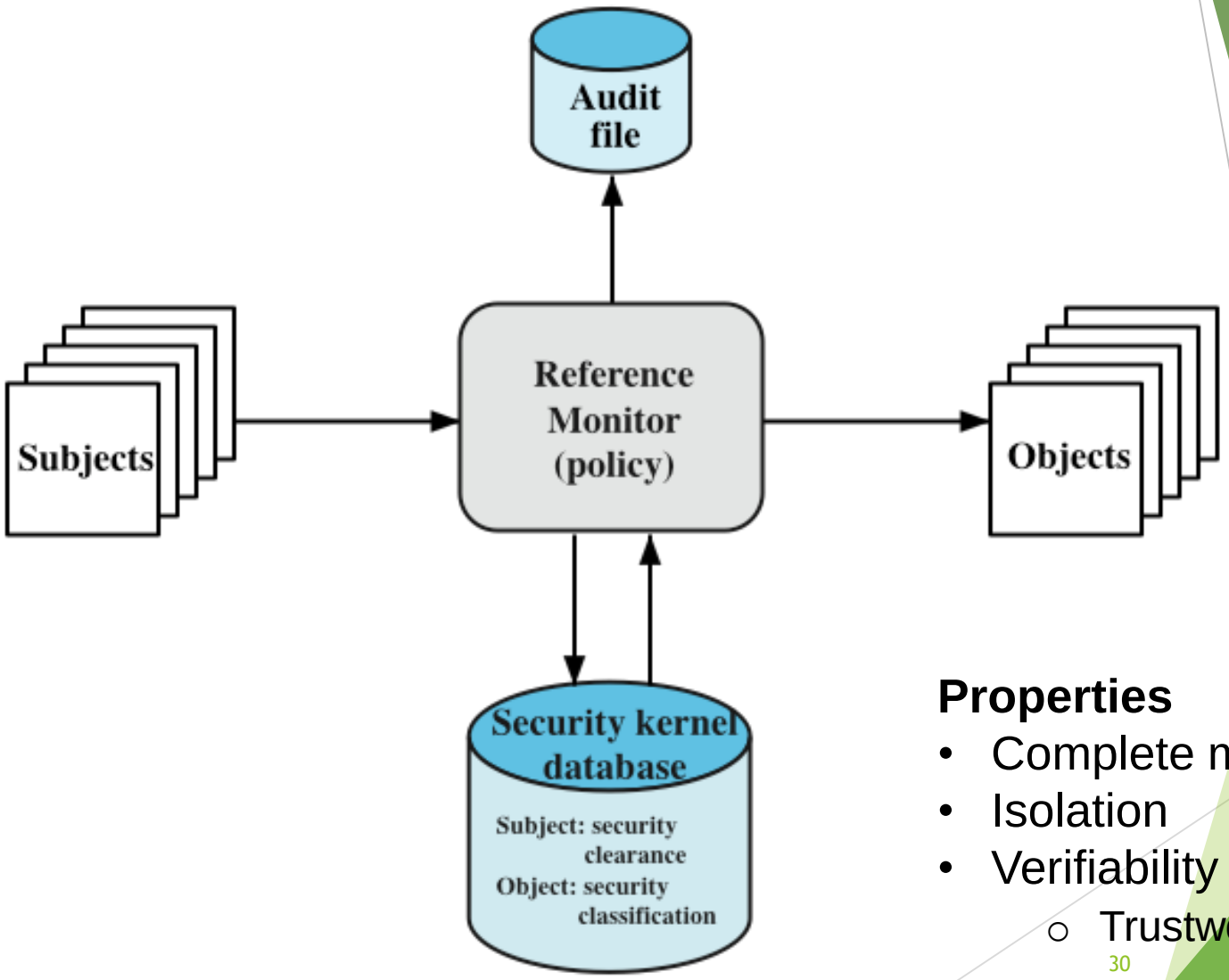


Figure 13.7 Reference Monitor Concept

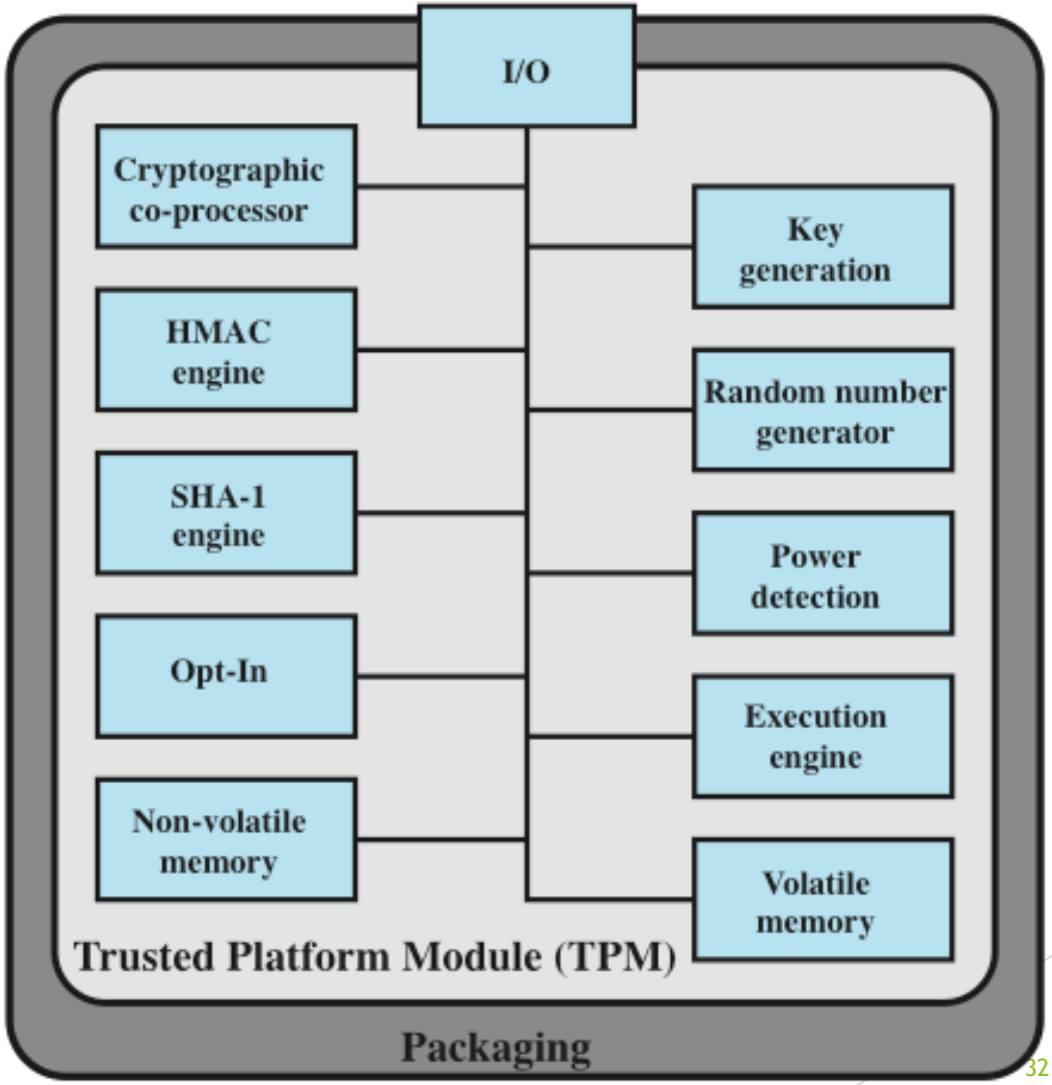
Properties

- Complete mediation
- Isolation
- Verifiability
  - Trustworthy system

Trusted Platform Module (TPM)

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

TPM Functions



Authenticated Boot Service

- responsible for booting entire OS in stages and ensuring each is valid and approved for use
  - at each stage digital signature associated with code is verified
  - TPM keeps a tamper-evident log of the loading process
- log records versions of all code running
  - can then expand trust boundary to include additional hardware and application and utility software
  - confirms component is on the approved list, is digitally signed, and that serial number hasn't been revoked
- result is a configuration that is well-defined with approved components

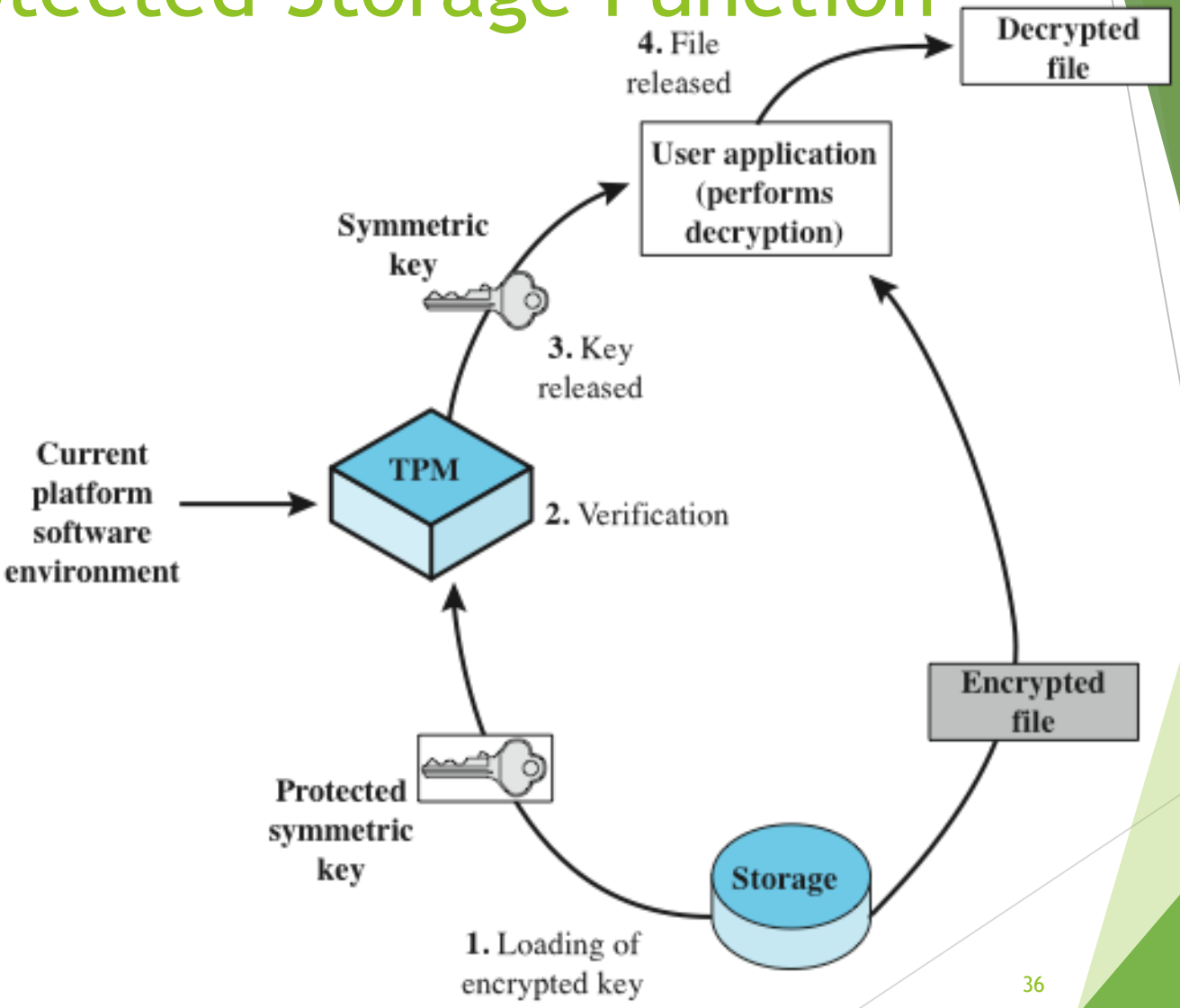
Certification Service

- once a configuration is achieved and logged the TPM can certify configuration to others
  - can produce a digital certificate
  - confidence that configuration is unaltered because:
    - TPM is considered trustworthy
    - only the TPM possesses this TPM's private key
- include challenge value in certificate to also ensure it is timely
- provides a hierarchical certification approach
  - hardware/OS configuration
  - OS certifies application programs
  - user has confidence is application configuration

Encryption Service

- encrypts data so that it can only be decrypted by a machine with a certain configuration
- TPM maintains a master secret key unique to machine
  - used to generate secret encryption key for every possible configuration of that machine
- can extend scheme upward
  - provide encryption key to application so that decryption can only be done by desired version of application running on desired version of the desired OS
  - encrypted data can be stored locally or transmitted to a peer application on a remote machine

Protected Storage Function



Summary

- computer security models
  - Bell-Lapadula
  - Biba Integrity Model
  - Clark-Wilson Integrity Model
  - Chinese Wall Model
- trusted systems
  - reference monitors
- application of multilevel security
  - database security
- trusted computing and the trusted platform module
  - authenticated boot service
  - certification service
  - encryption service
  - TPM functions
  - protected storage