COMPUTER SECURITY

PRINCIPLES AND PRACTICE

SECOND EDITION

William Stallings | Lawrie Brown

Chapter 13

Trusted Computing and Multilevel Security

Computer Security Models

two fundamental computer security facts:

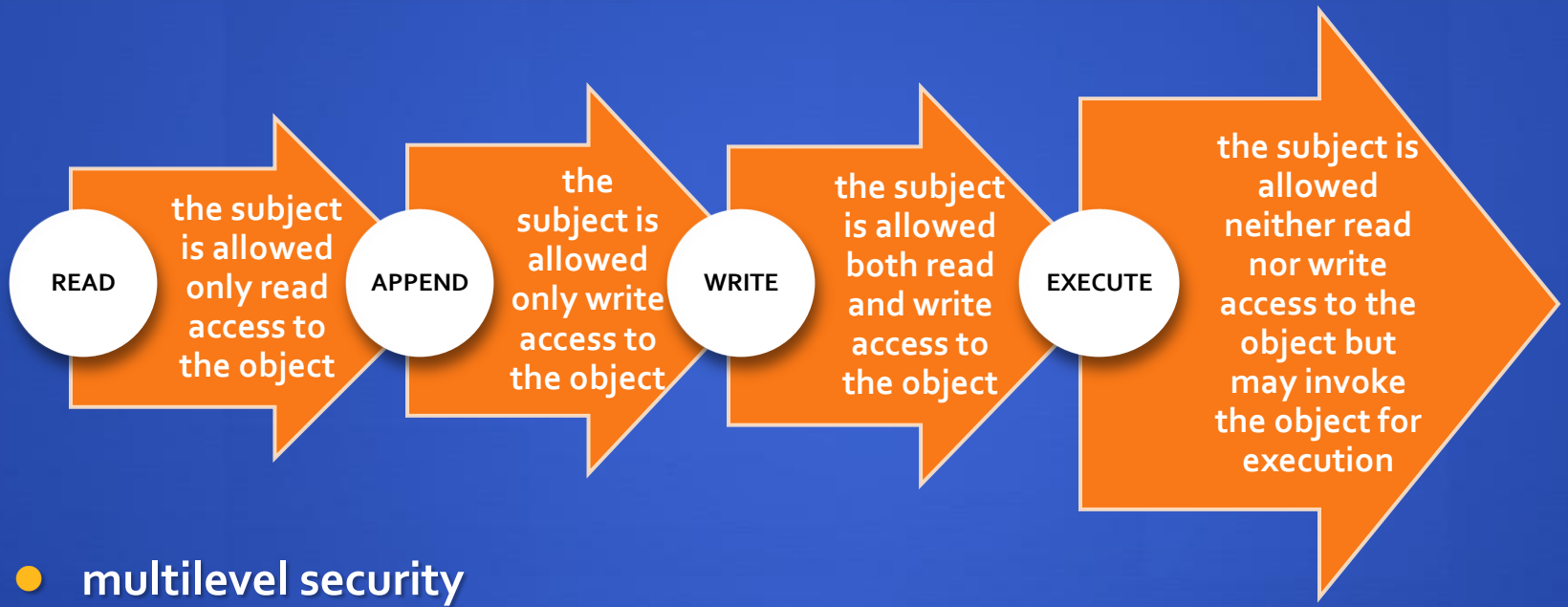
- all complex software systems have eventually revealed flaws or bugs that need to be fixed
- it is extraordinarily difficult to build computer hardware/software not vulnerable to security attacks

- problems involved both design and implementation
- led to development of formal security models
  - initially funded by US Department of Defense
- Bell-LaPadula (BLP) model very influential

Bell-LaPadula (BLP) Model

- developed in 1970s
- formal model for access control
- subjects and objects are assigned a security class
  - top secret > secret > confidential > restricted > unclassified
  - 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

BLP Model Access Modes



- multilevel 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

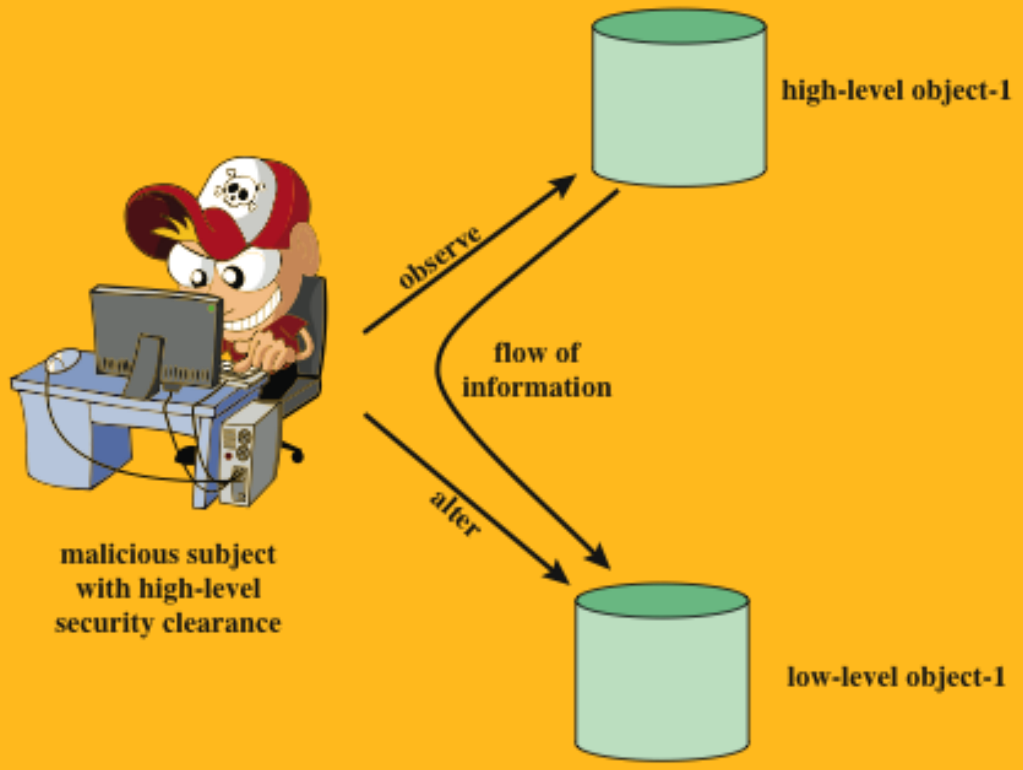


Figure 13.1 Information Flow Showing the Need for the *-property

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 $A_x \in M[S_i\, O_j]$
- BLP gives 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 current level
5. give access permission
6. create an object
7. delete a group of objects

BLP Example

(slide 1 of 3)

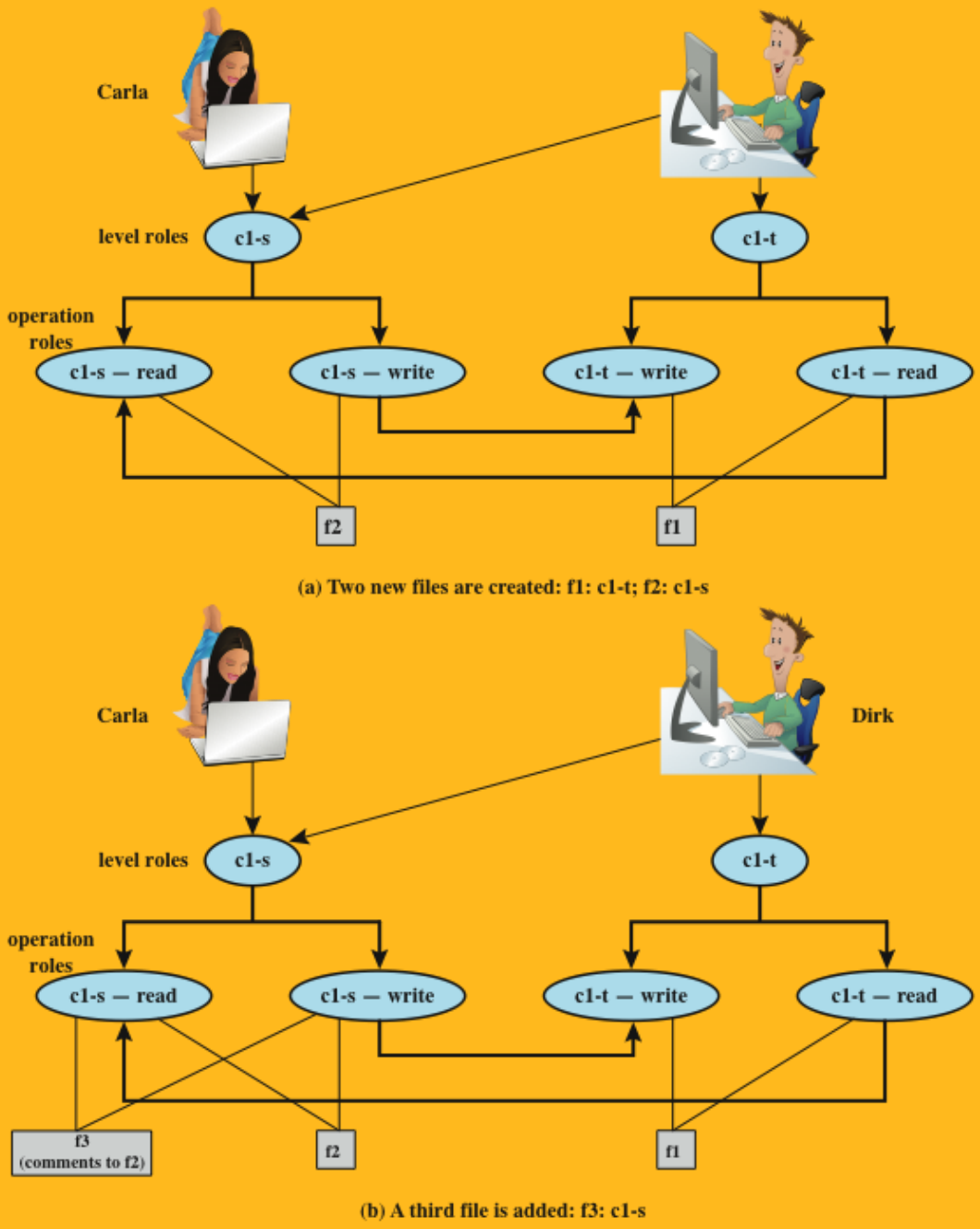
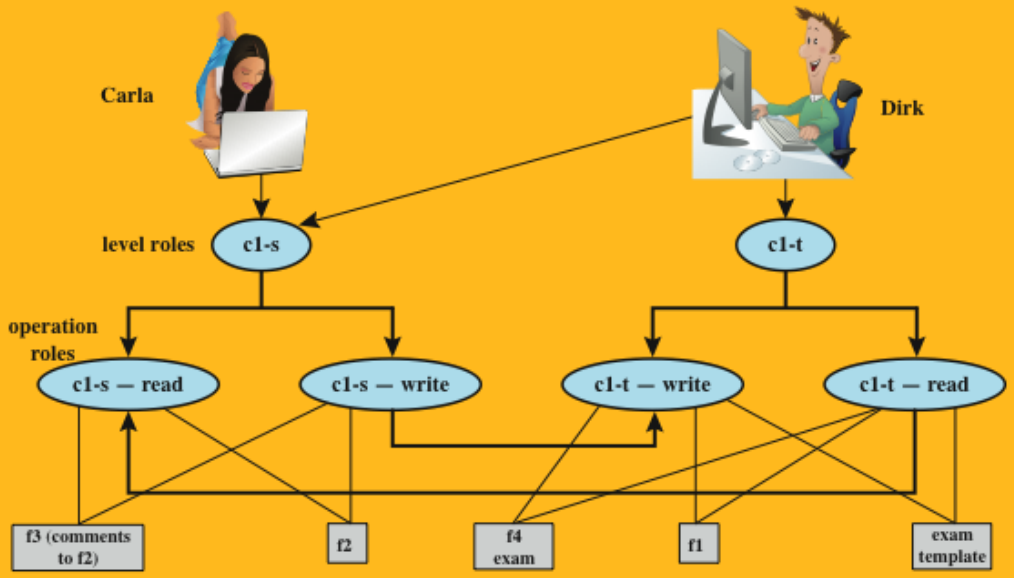


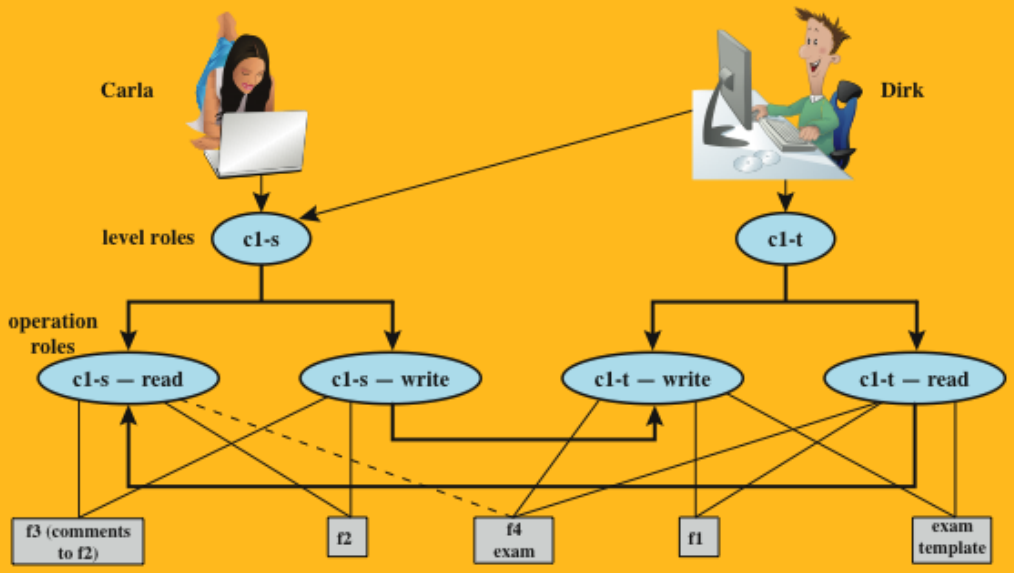
Figure 13.2 Example of Use of BLP Concepts (page 1 of 3)

BLP Example

(slide 2 of 3)



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

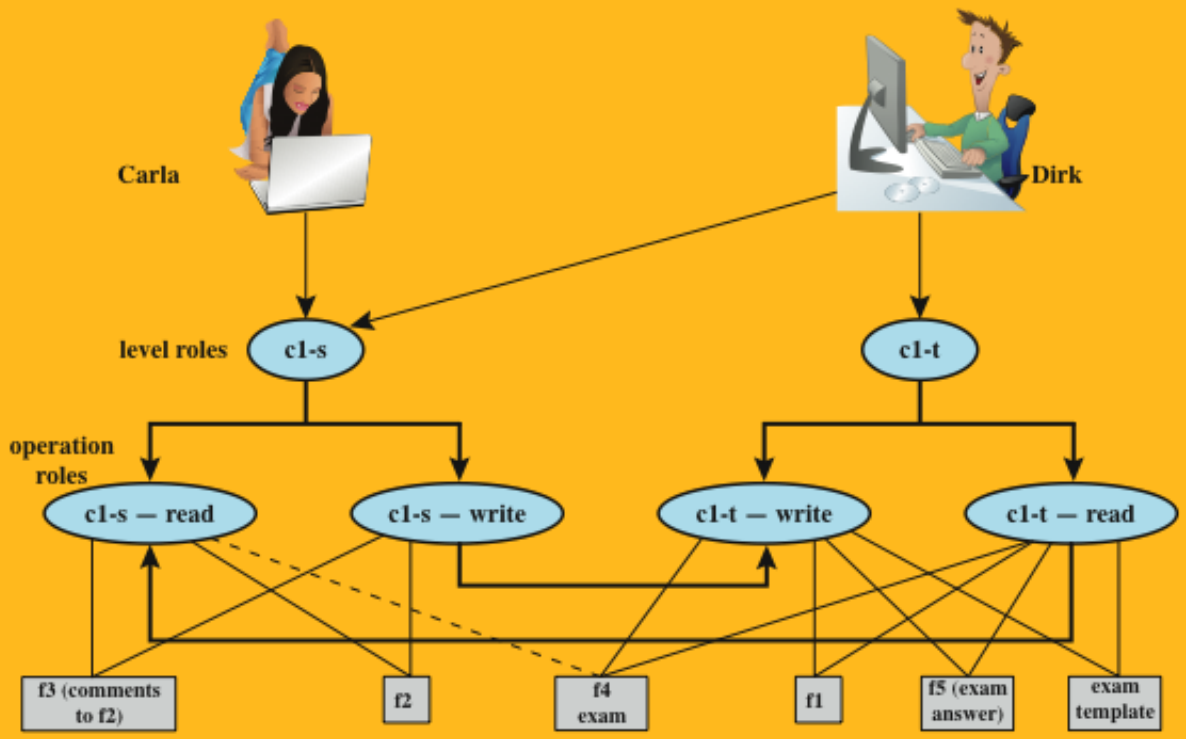


(d) Carla, as student, is permitted acess to the exam: f4: c1-s

Figure 13.2 Example of Use of BLP Concepts (page 2 of 3)

BLP Example

(slide 3 of 3)



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

Figure 13.2 Example of Use of BLP Concepts (page 3 of 3)

Implementation Example

Multics

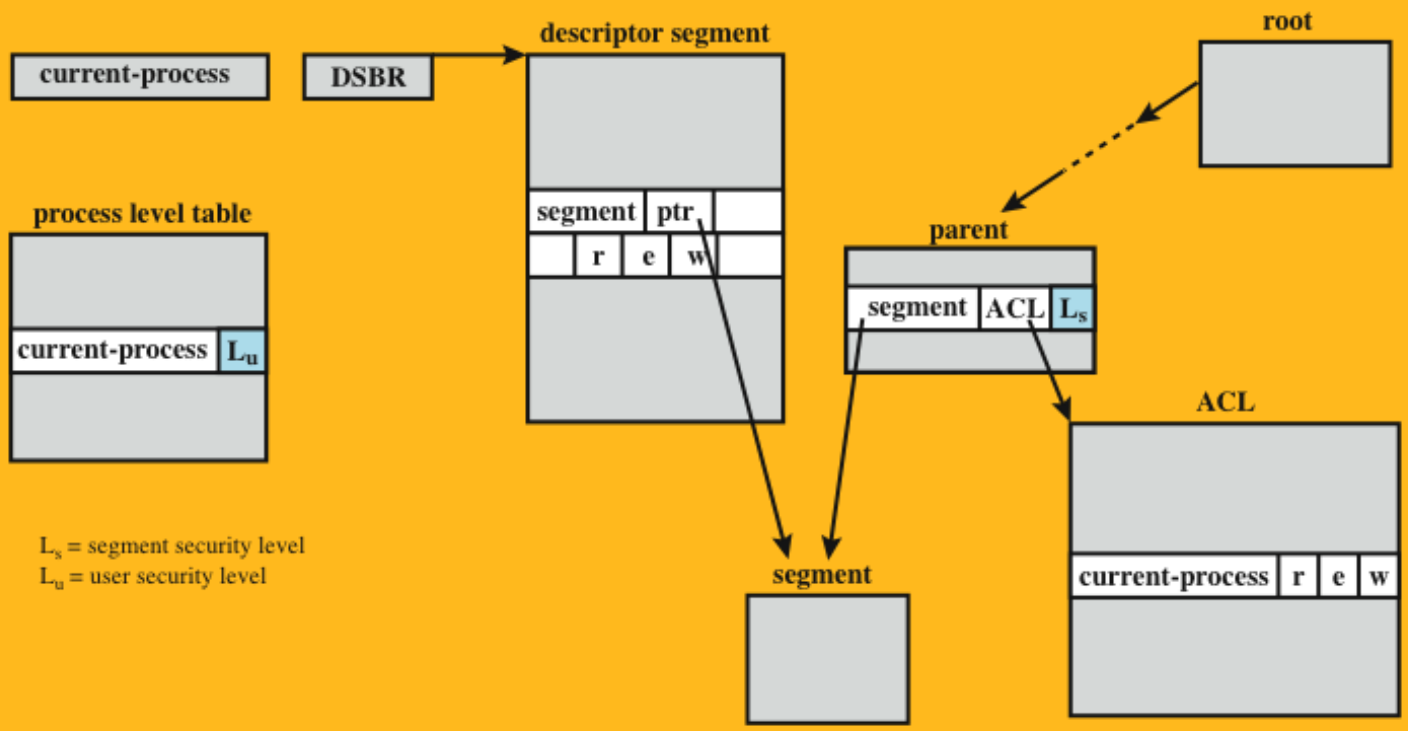


Figure 13.3 Multics Data Structures for MLS

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: $I(S_1) \geq I(S_2)$

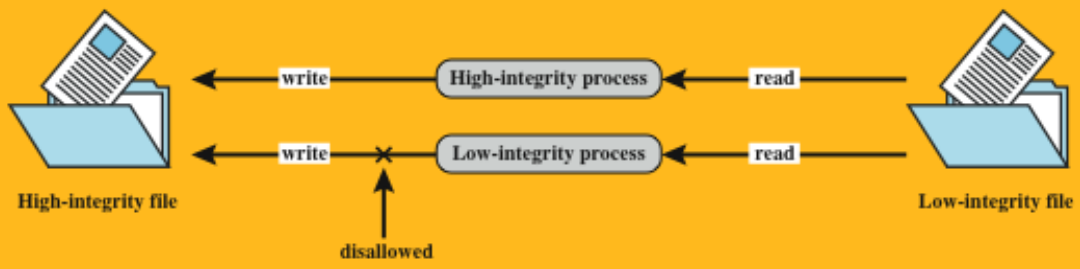


Figure 13.4 Contamination With Simple Integrity Controls [GASS88]

Clark-Wilson Integrity Model

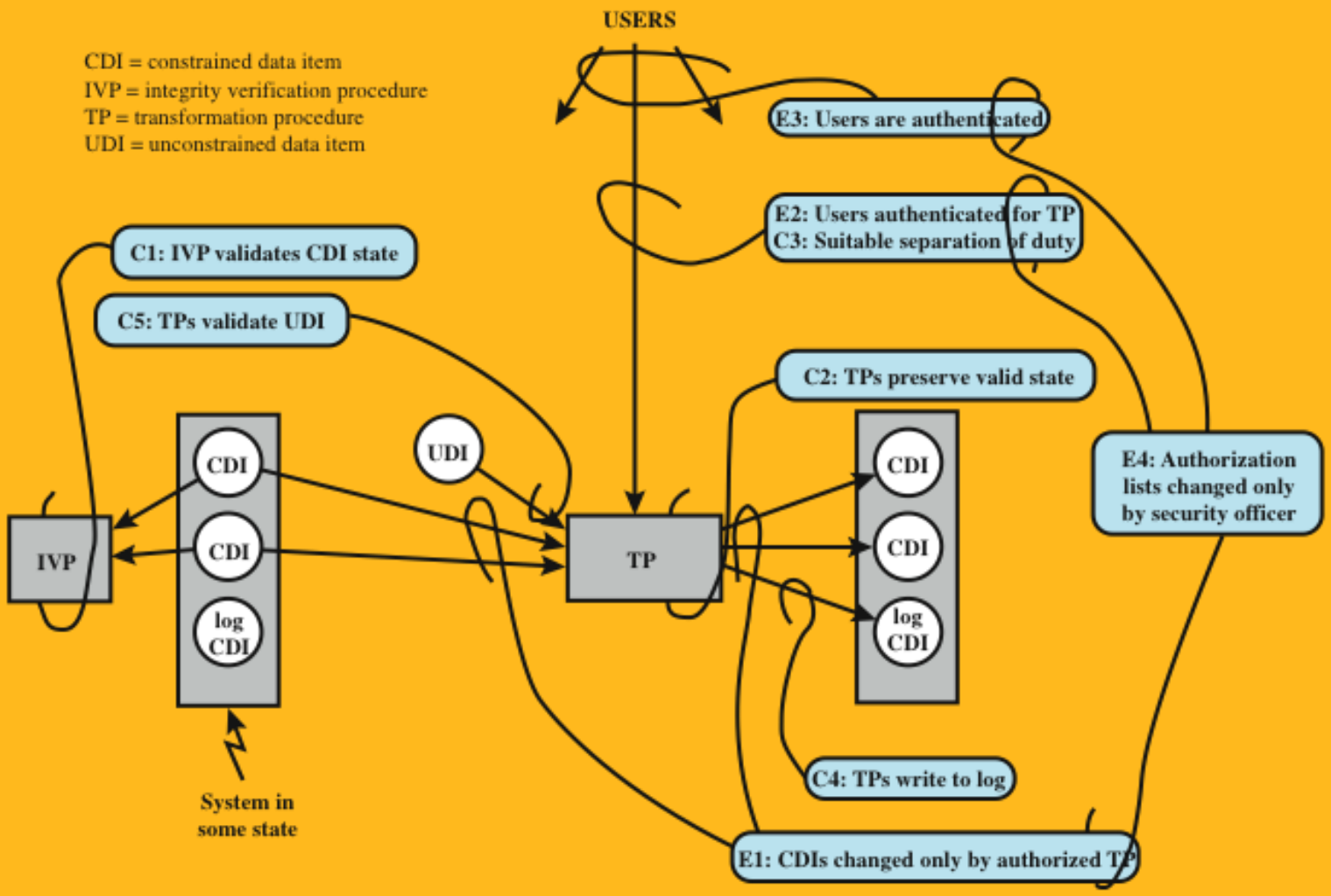


Figure 13.5 Summary of Clark-Wilson System Integrity Rules [CLAR87]

Chinese Wall Model

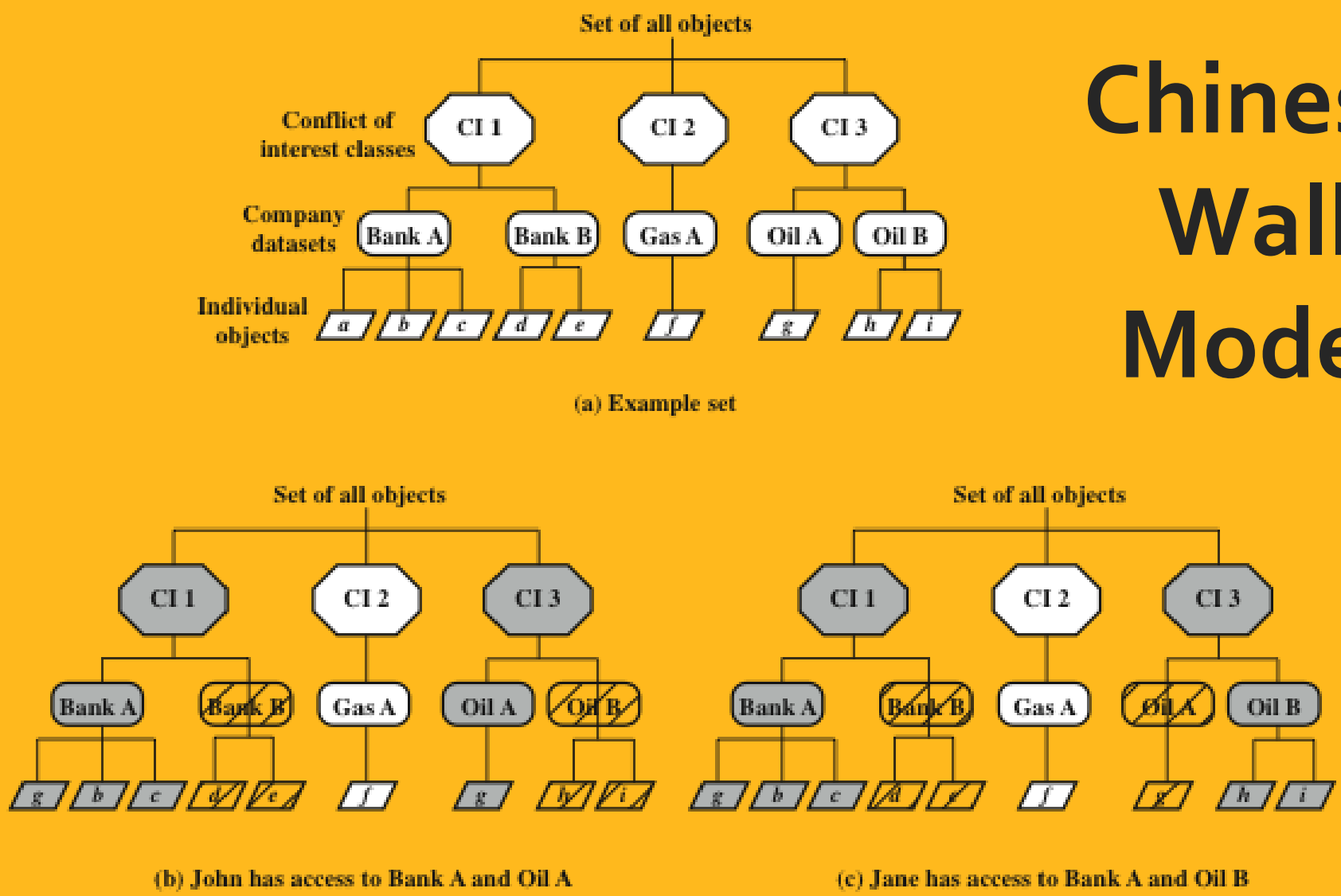


Figure 13.6 Potential Flow of Information Between Two CIs

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.

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.

Table 13.1

Terminology Related to Trust

Reference Monitors

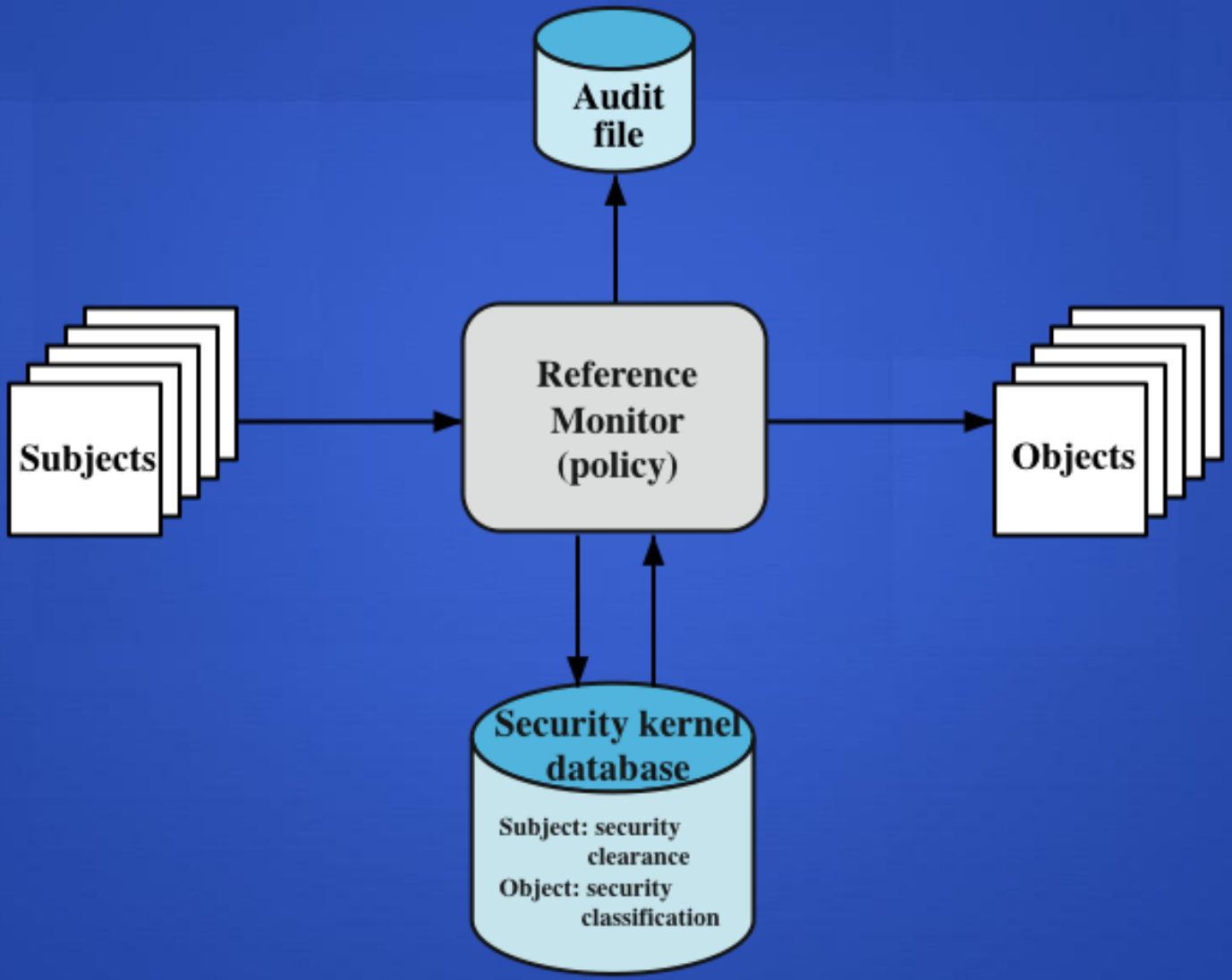


Figure 13.7 Reference Monitor Concept

Trojan Horse Defense

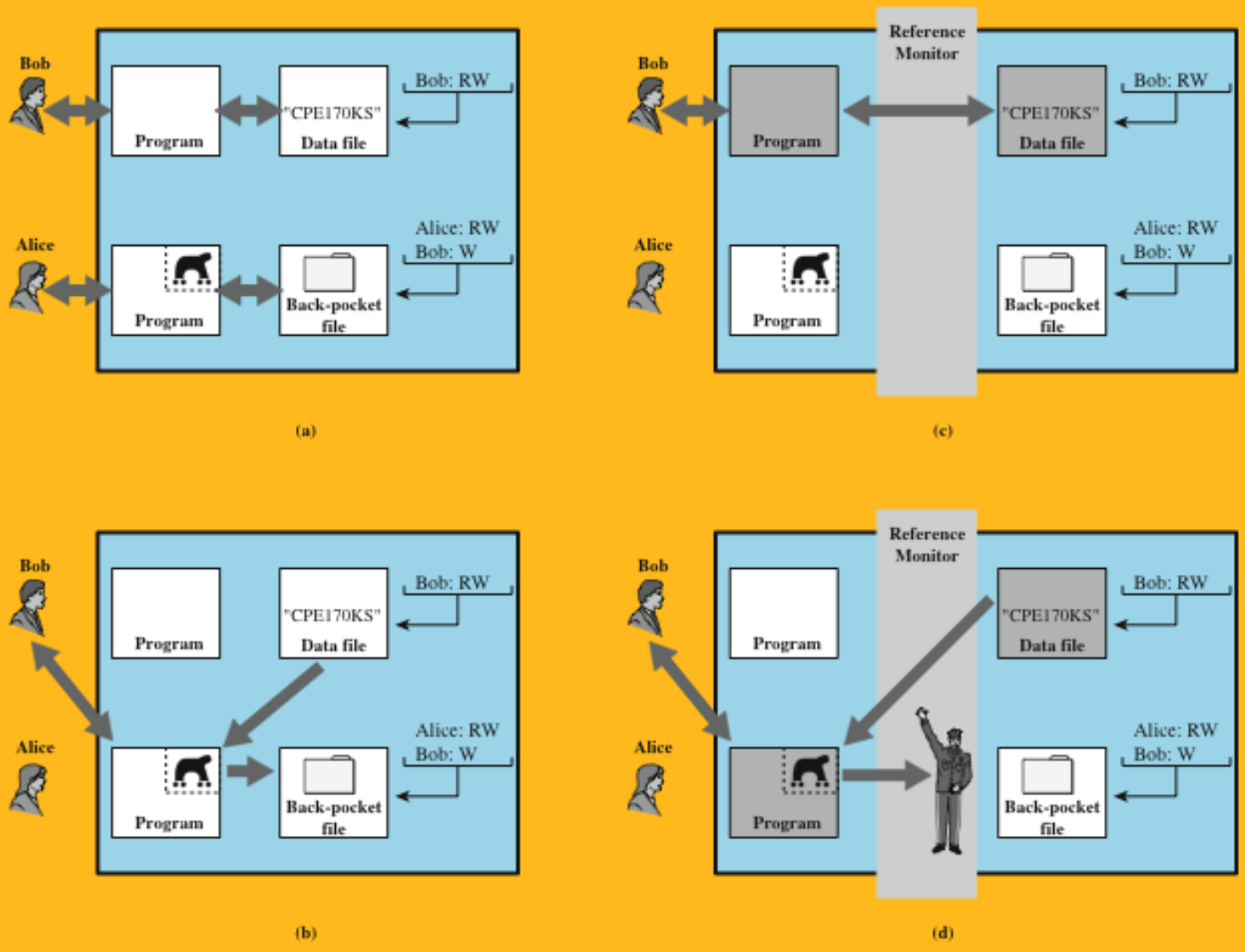


Figure 13.8 Trojan Horse and Secure Operating System

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."

Table 13.2

RBAC Elements

U, a set of users R and AR, disjoint sets of (regular) roles and administrative roles P and AP, disjoint sets of (regular) permissions and administrative permissions S, a set of sessions
$PA \subseteq P \times R$, a many-to-many permission to role assignment relation $APA \subseteq AP \times AR$, a many-to-many permission to administrative role assignment relation
$UA \subseteq U \times R$, a many-to-many user to role assignment relation $AUA \subseteq U \times AR$, a many-to-many user to administrative role assignment relation
$RH \subseteq R \times R$, a partially ordered role hierarchy $ARH \subseteq AR \times AR$, partially ordered administrative role hierarchy (both hierarchies are written as $\geq$ in infix notation)
$user : S \rightarrow U$, a function mapping each session s_i to the single user $user(s_i)$ (constant for the session's lifetime) $roles : S \rightarrow 2^{R \cup AR}$ maps each session s_i to a set of roles and administrative roles $roles(s_i) \subseteq \{ r \mid (\exists r' \geq r)[(user(s_i), r') \in UA \cup AUA]\}$ (which can change with time) session s_i has the permissions $\cup_{r \in roles(si)}\{p \mid (\exists r'' \leq r) \in PA \cup APA]\}$
There is a collection of constraints stipulating which values of the various components enumerated above are allowed or forbidden.

Figure 13.9

Role Hierarchy User Assignments

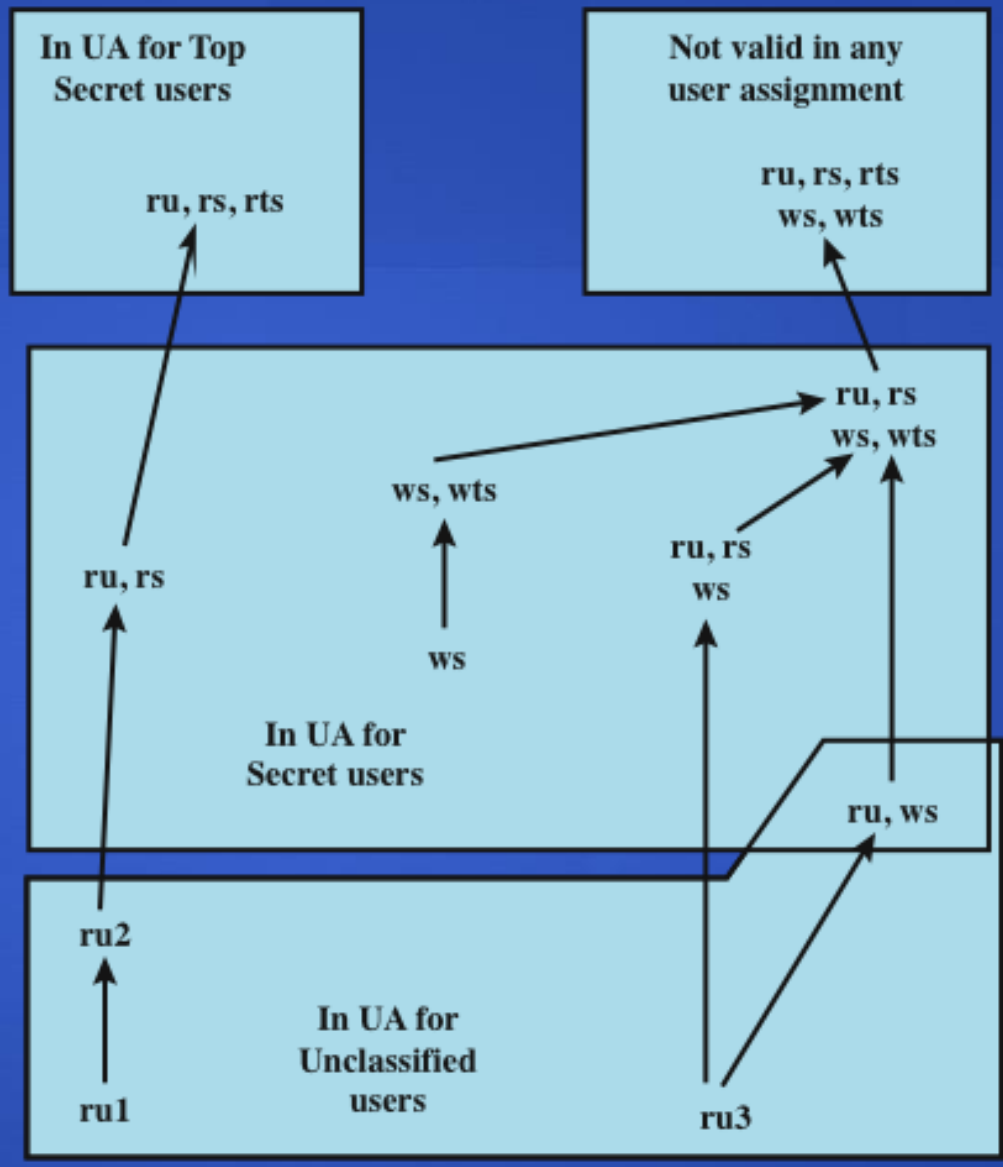


Figure 13.9 A Role Hierarchy and Its User Assignments [OSBO00]

Database Classification

Table

Column

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)

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

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

Authenticated Boot Service

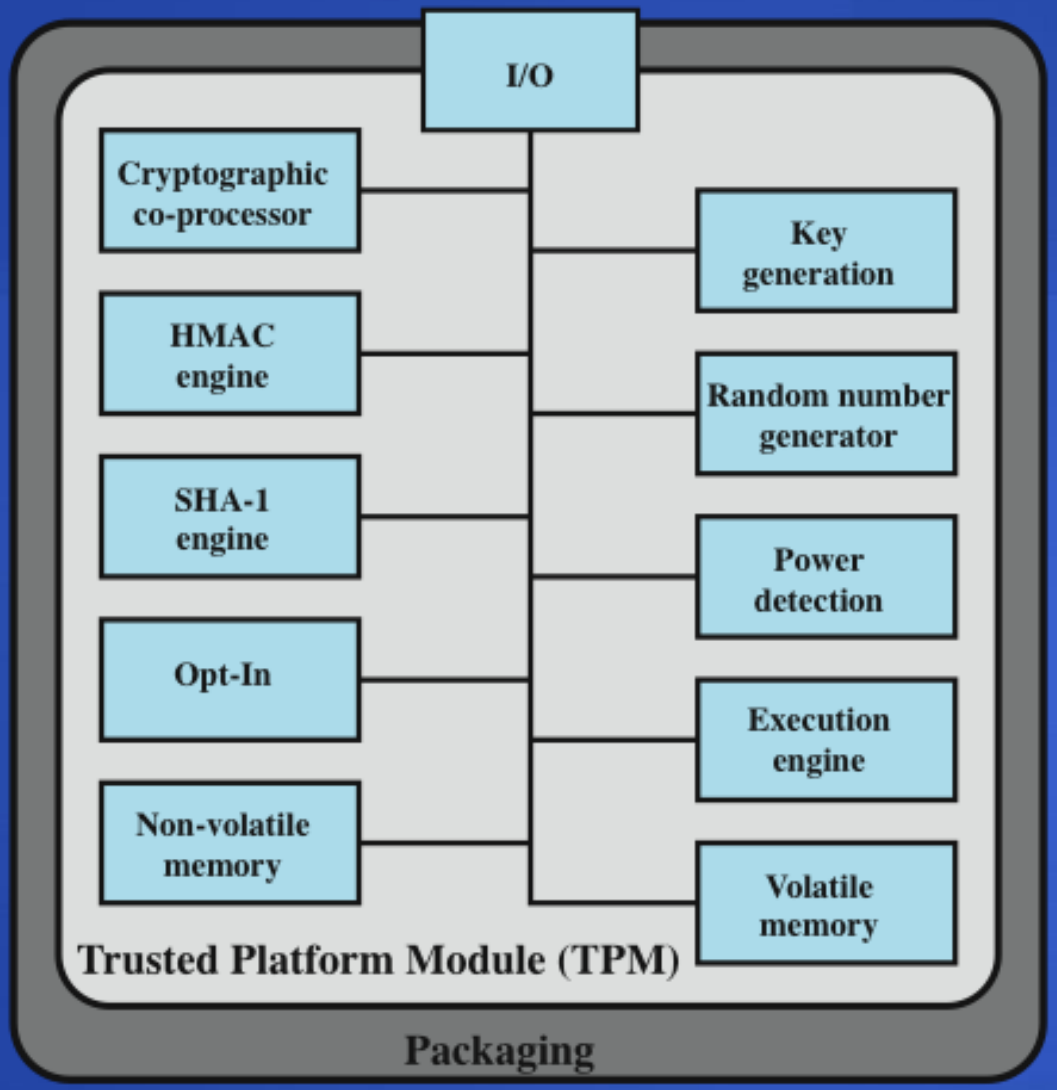
- responsible for booting entire OS in stages and ensuring each is valid and approved for use
  - at each stage digital signature associated with code is verified
  - TPM keeps a tamper-evident log of the loading process
- log records versions of all code running
  - can then expand trust boundary to 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



TPM Functions

Figure 13.12 TPM Component Architecture

Protected Storage Function

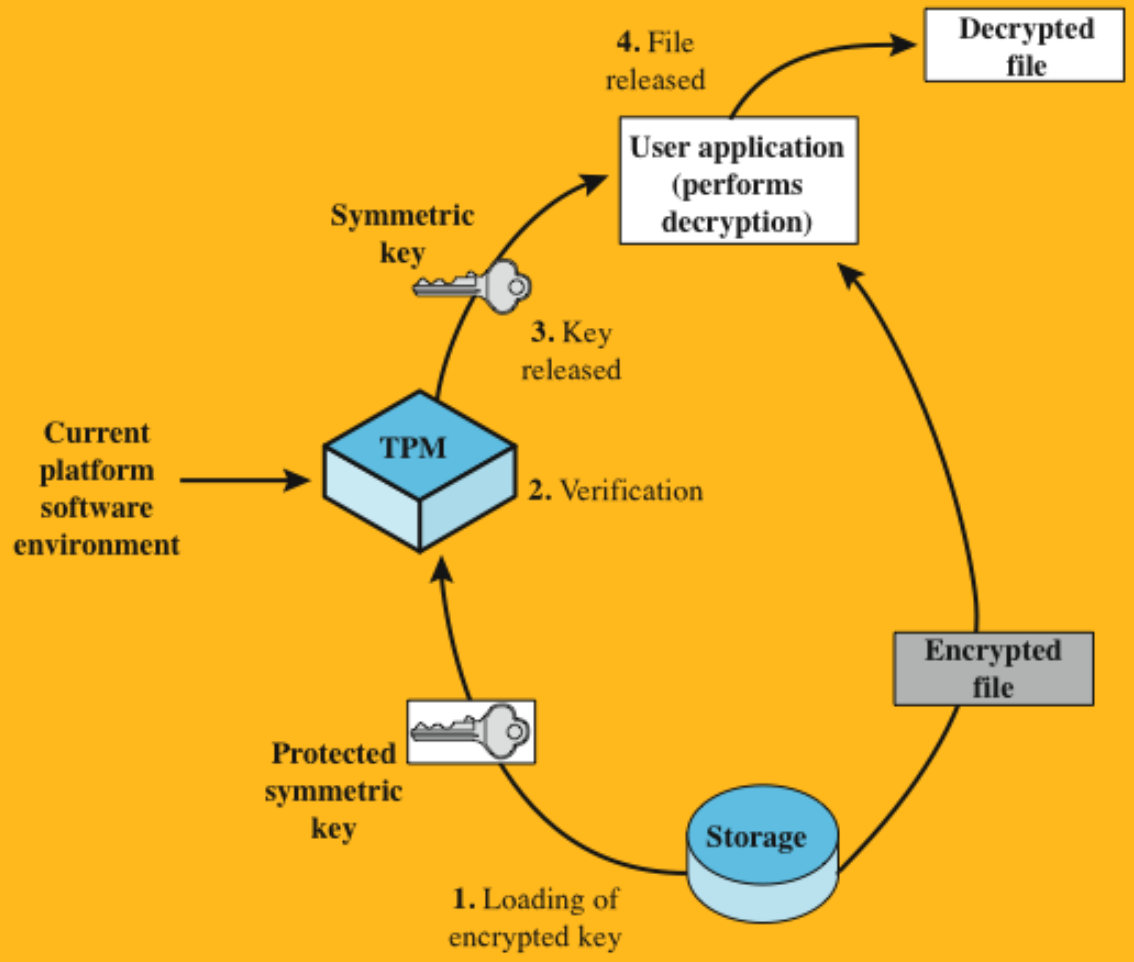


Figure 13.13 Decrypting a File Using a Protected Key

Common Criteria (CC)

- *Common Criteria for Information Technology and Security Evaluation*
  - ISO standards for security requirements and defining evaluation criteria
- aim is to provide greater confidence in IT product security
  - development using secure requirements
  - evaluation confirming meets requirements
  - operation in accordance with requirements
- following successful evaluation a product may be listed as CC certified
  - NIST/NSA publishes lists of evaluated products

CC Requirements

common set of potential security requirements for use in evaluation

target of evaluation (TOE)

- refers to the part of product or system subject to evaluation

functional requirements

- define desired security behavior

assurance requirements

- basis for gaining confidence that the claimed security measures are effective and implemented correctly

class

- collection of requirements that share a common focus or intent

component

- describes a specific set of security requirements
- smallest selectable set

Table 13.3

CC Security Functional Requirements

Class	Description
Audit	Involves recognizing, recording, storing and analyzing information related to security activities. Audit records are produced by these activities, and can be examined to determine their security relevance.
Cryptographic support	Used when the TOE implements cryptographic functions. These may be used, for example, to support communications, identification and authentication, or data separation.
Communications	Provides two families concerned with nonrepudiation by the originator and by the recipient of data.
User data protection	Specifies requirements relating to the protection of user data within the TOE during import, export, and storage, in addition to security attributes related to user data.
Identification and authentication	Ensure the unambiguous identification of authorized users and the correct association of security attributes with users and subjects.
Security management	Specifies the management of security attributes, data and functions.
Privacy	Provides a user with protection against discovery and misuse of his or her identity by other users.
Protection of the TOE security functions	Focused on protection of TSF (TOE security functions) data, rather than of user data. The class relates to the integrity and management of the TSF mechanisms and data.
Resource utilization	Supports the availability of required resources, such as processing capability and storage capacity. Includes requirements for fault tolerance, priority of service, and resource allocation.
TOE access	Specifies functional requirements, in addition to those specified for identification and authentication, for controlling the establishment of a user's session. The requirements for TOE access govern such things as limiting the number and scope of user sessions, displaying the access history, and modifying of access parameters.
Trusted path/channels	Concerned with trusted communications paths between the users and the TSF and between TSFs.

Table 13.4

CC Security Assurance Requirements

Class	Description
Configuration management	Requires that the integrity of the TOE is adequately preserved. Specifically, configuration management provides confidence that the TOE and documentation used for evaluation are the ones prepared for distribution.
Delivery and operation	Concerned with the measures, procedures, and standards for secure delivery, installation, and operational use of the TOE, to ensure that the security protection offered by the TOE is not compromised during these events.
Development	Concerned with the refinement of the TSF from the specification defined in the ST to the implementation, and a mapping from the security requirements to the lowest level representation.
Guidance documents	Concerned with the secure operational use of the TOE, by the users and administrators.
Life cycle support	Concerned with the life cycle of the TOE include life cycle definition, tools and techniques, security of the development environment, and remediation of flaws found by TOE consumers.
Tests	Concerned with demonstrating that the TOE meets its functional requirements. The families address coverage and depth of developer testing, and requirements for independent testing.
Vulnerability assessment	Defines requirements directed at the identification of exploitable vulnerabilities, which could be introduced by construction, operation, misuse or incorrect configuration of the TOE. The families identified here are concerned with identifying vulnerabilities through covert channel analysis, analyzing the configuration of the TOE, examining the strength of mechanisms of the security functions, and identifying flaws introduced during development of the TOE. The second family covers the security categorization of TOE components. The third and fourth cover the analysis of changes for security impact, and the provision of evidence that procedures are being followed. This class provides building blocks for the establishment of assurance maintenance schemes.
Assurance maintenance	Provides requirements that are intended to be applied after a TOE has been certified against the CC. These requirements are aimed at assuring that the TOE will continue to meet its security target as changes are made to the TOE or its environment.

Organization and Construction of CC Requirements

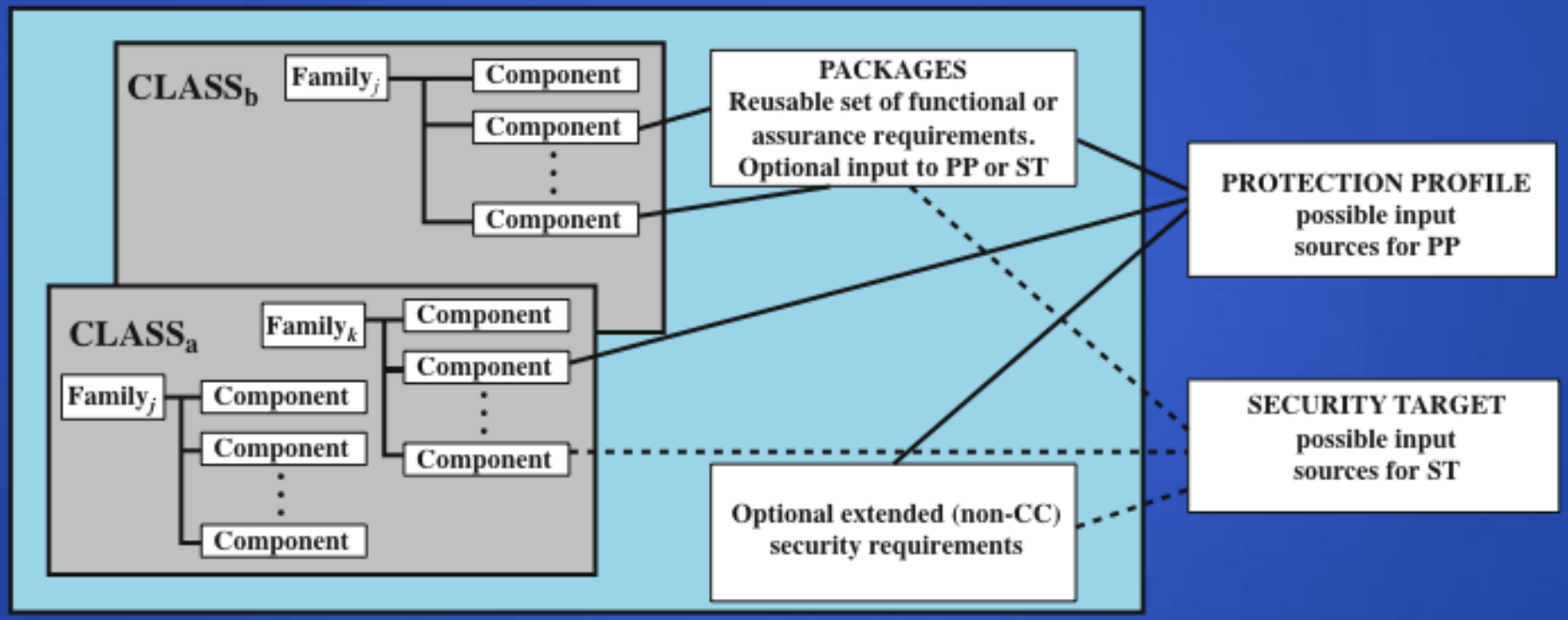


Figure 13.14 Organization and Construction of Common Criteria Requirements

CC Security Paradigm

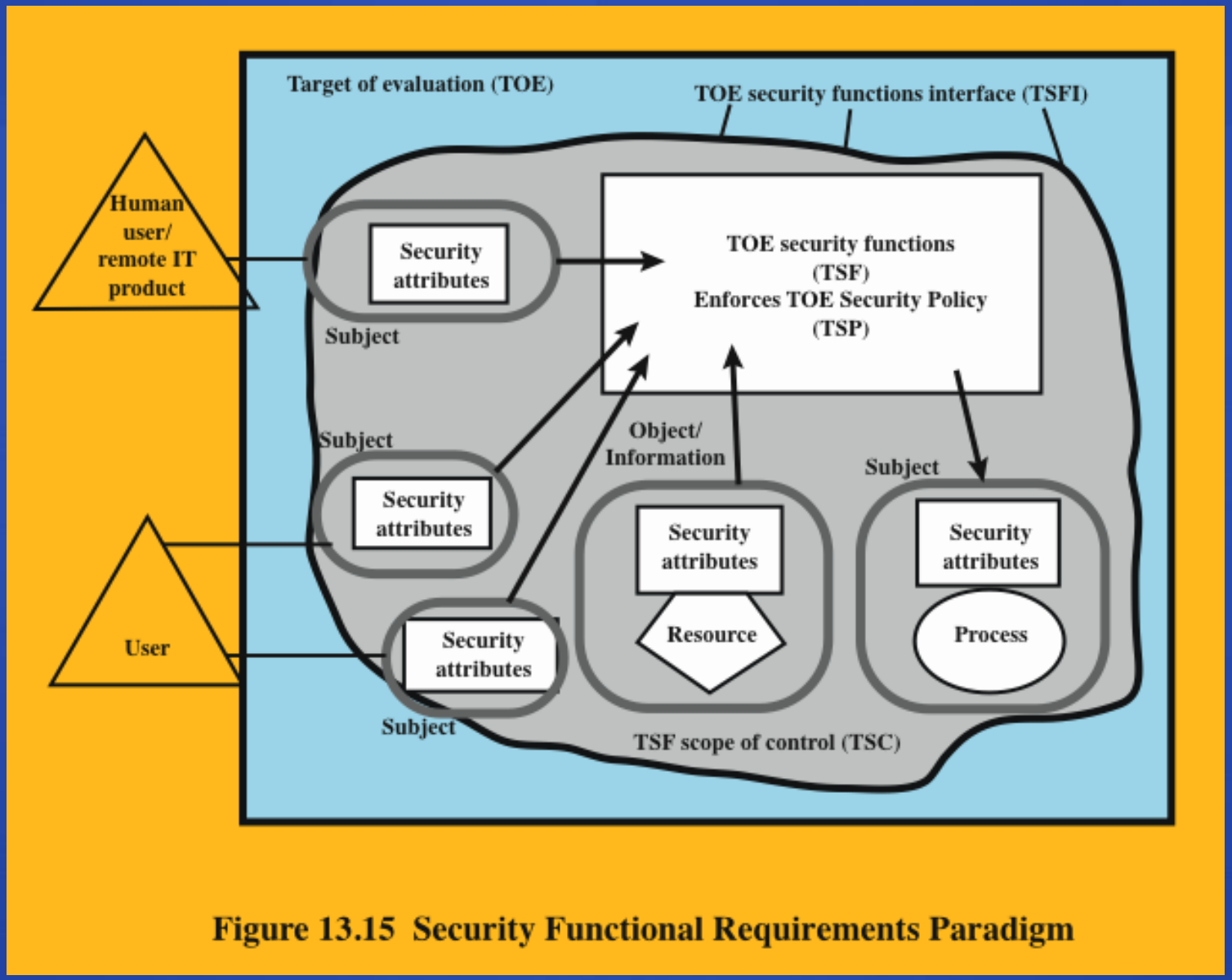


Figure 13.15 Security Functional Requirements Paradigm

Protection Profile (PP)

- smart card provides simple PP example
- describes IT security requirements for smart card use by sensitive applications

threats that must be addressed:

- physical probing
- invalid input
- linkage of multiple operations

security objectives

- reflect the stated intent to counter identified threats and comply with identified organizational security policies

security requirements

- provided to thwart specific threats and to support specific policies under specific assumptions

Security Assurance

"…degree of confidence that the security controls operate correctly and protect the system as intended. Assurance is not, however, an absolute guarantee that the measures work as intended."

Assurance and Evaluation

target audiences:

consumers

- select security features and functions
- determine the required levels of security assurance

developers

- respond to security requirements
- interpret statements of assurance requirements
- determine assurance approaches and level of effort

evaluators

- use the assurance requirements as criteria when evaluating security features and controls
- may be in the same organization as consumers or a third-party evaluation team

- assurance
  - deals with security features of IT products
  - applies to:
    - requirements
    - security policy
    - product design
    - product implementation
    - system operation

Scope of Assurance

system architecture

- addresses both the system development phase and the system operations phase

system integrity

- addresses the correct operation of the system hardware and firmware

system testing

- ensures security features have been tested thoroughly

covert channel analysis

- attempts to identify any potential means for bypassing security policy

trusted facility management

- deals with system administration

configuration management

- requirements are included for configuration control, audit, management, and accounting

design specification and verification

- addresses the correctness of the system design and implementation with respect to the system security policy

trusted recovery

- provides for correct operation of security features after a system recovers from failures, crashes, or security incidents

trusted distribution

- ensures that protected hardware, firmware, and software do not go through unauthorized modification during transit from the vendor to the customer

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: semi-formally designed and tested
- EAL 6: semi-formally verified design and tested
- EAL 7: formally verified design and tested

Evaluation

- ensures security features work correctly and effectively and show no exploitable vulnerabilities
- performed in parallel with or after the development of the TOE
- higher levels entail: greater rigor, more time, more cost
- principle input: security target, evidence, actual TOE
- result: confirm security target is satisfied for TOE
- process relates security target to high-level design, low-level design, functional specification, source code implementation, and object code and hardware realization of the TOE
- degree of rigor and depth of analysis are determined by assurance level desired

Evaluation Parties and Phases

- evaluation parties:
  - sponsor - customer or vendor
  - developer - provides evidence for evaluation
  - evaluator - confirms requirements are satisfied
  - certifier - agency monitoring evaluation process
- monitored and regulated by a government agency in each country
- Common Criteria Evaluation and Validation Scheme (CCEVS)
  - operated by NIST and the NSA

Phases

Summary

- computer security models
  - Bell-Lapadula
  - Biba Integrity Model
  - Clark-Wilson Integrity Model
  - Chinese Wall Model
- trusted systems
  - reference monitors
  - Trojan Horse Defense
- application of multilevel security
  - role-based access control
  - database security
- common criteria for information technology security evaluation
  - requirements
  - profiles and targets
- trusted computing and the trusted platform module
  - authenticated boot service
  - certification service
  - encryption service
  - TPM functions
  - protected storage
- assurance and evaluation
  - target audience
  - scope of assurance
  - common criteria evaluation assurance levels
  - evaluation process