

# Computer and Network Security Trusted Operating Systems

R. E. Newman

Computer & Information Sciences & Engineering  
University Of Florida  
Gainesville, Florida 32611-6120  
nemo@cise.ufl.edu

# **Trusted Operating Systems (Pfleeger Ch. 7)**

## **Security Models and Policies Software Engineering!**

# **1 Definitions**

## **1.1 Security vs. assurance vs. trust**

- Security is binary, absolute, intrinsic
- Assurance is performance relative to expectations
- Trust is receiver-centric, based on history, relative, graded

## **1.2 Certification vs. accreditation**

- Enforcement of security policy
- Sufficiency of controls
- Evaluation - in assumed environment vs. as deployed

## **2 Policies**

### **2.1 Military - Multilevel Compartmented System (MLS)**

#### **2.1.1 Sensitivity**

- TS/S/C/R/U = rank or level - hierarchical

#### **2.1.2 Compartmented**

- need-to-know categories (set-based) - non-hierarchical

#### **2.1.3 Labels = ⟨level, category set⟩**

- Objects - Classification
- Subjects - Clearance

## 2.1.4 Dominance

$$L1 = \langle R1, C1 \rangle \geq \langle R2, C2 \rangle = L2$$

(L1 dominates L2) iff

$$R1 \geq R2 \text{ and } C1 \subseteq C2$$

Subject  $S$  with label  $L1$  is only allowed to read an object  $O$  with label  $L2$  if  $L1 \geq L2$

## 2.2 Commercial

### 2.2.1 Clark-Wilson

- Based on well-formed transactions
- Users
- Constrained data items
- Transformation procedures
- Access triples  $\langle userID, TP_i, \{CDI_{i1}, CDI_{i2}, \dots\} \rangle$

### 2.2.2 Separation of Duty

- Application specific
- Same person not allowed to perform too many operations in a process
- Requires history of who has done what on a per-operation basis
- e.g., multi-signature authorization

### 2.2.3 Chinese Wall (Conflict of Interest)

- Objects associated with entities
- Entities associated with zero or more Conflict Groups
- Users who have accessed objects associated with some entity may not access objects associated with an entity in the same conflict group

- Requires history of access

## 3 Models

### 3.1 MLS

#### 3.1.1 Lattice model

Lattice  $\langle S, \leq \rangle$  is a partially ordered set (poset)  $S$  with partial order  $\leq$  (transitive and antisymmetric) such that for any  $s_1, s_2 \in S$ , there exists

- a least upper bound (LUB)  $u$ ,  
 $s_1 \leq u$  and  $s_2 \leq u$ , and for all  $u'$  where  $s_1, s_2 \leq u', u \leq u'$ ;
- and a greatest lower bound (GLB)  $l$ ,  
 $l \leq s_1, l \leq s_2$  and for all  $l'$  where  $l' \leq s_1, s_2, l' \leq l$ ;



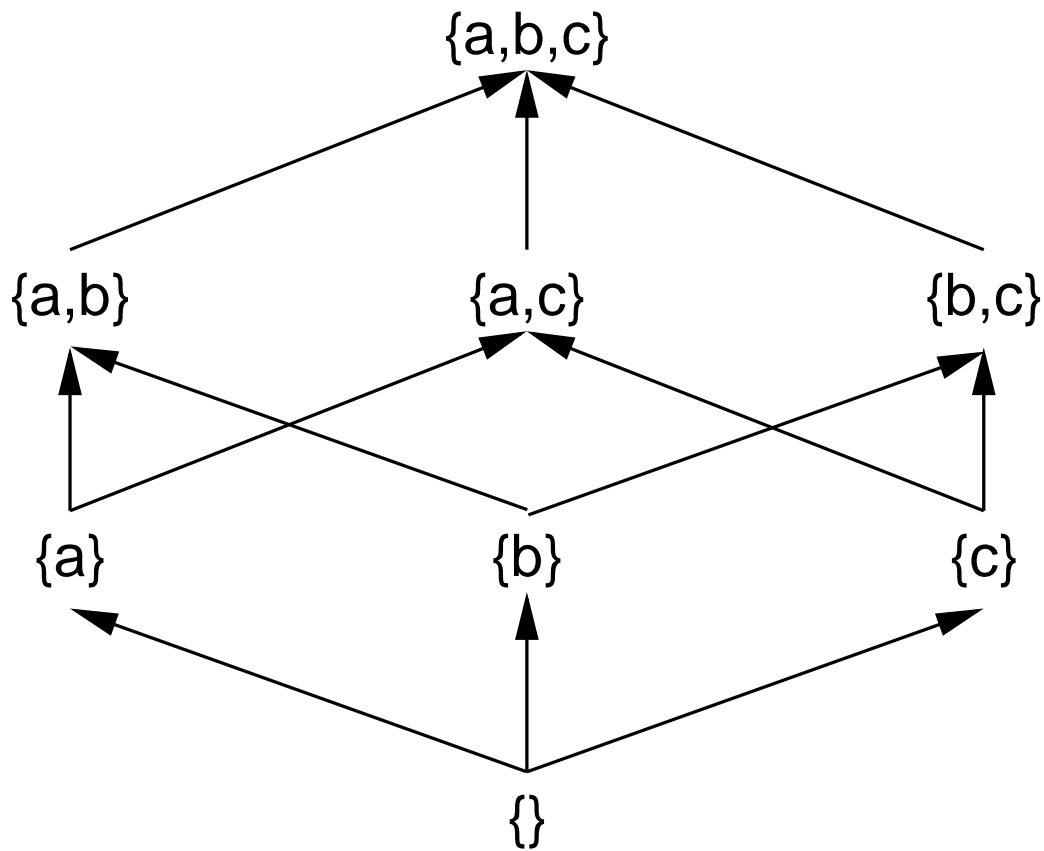


Figure 1: A non-linear lattice

### 3.1.2 BLP

Used for confidentiality of information

- MLS lattice-based labels,  $l(s)$ ,  $l(o)$
- Simple Security Property (no read up): A subject  $s$  may read object  $o$  iff  $l(s) \geq l(o)$
- \*-Property (no write down): A subject  $s$  who may read object  $o$  may also write object  $p$  iff  $l(p) \geq l(o)$

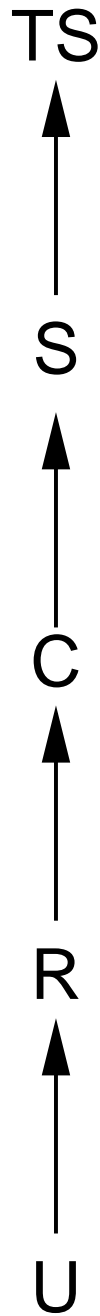


Figure 2: A linear lattice

### 3.1.3 Biba

Used for integrity of information

- MLS lattice-based labels  $I(s)$ ,  $I(o)$

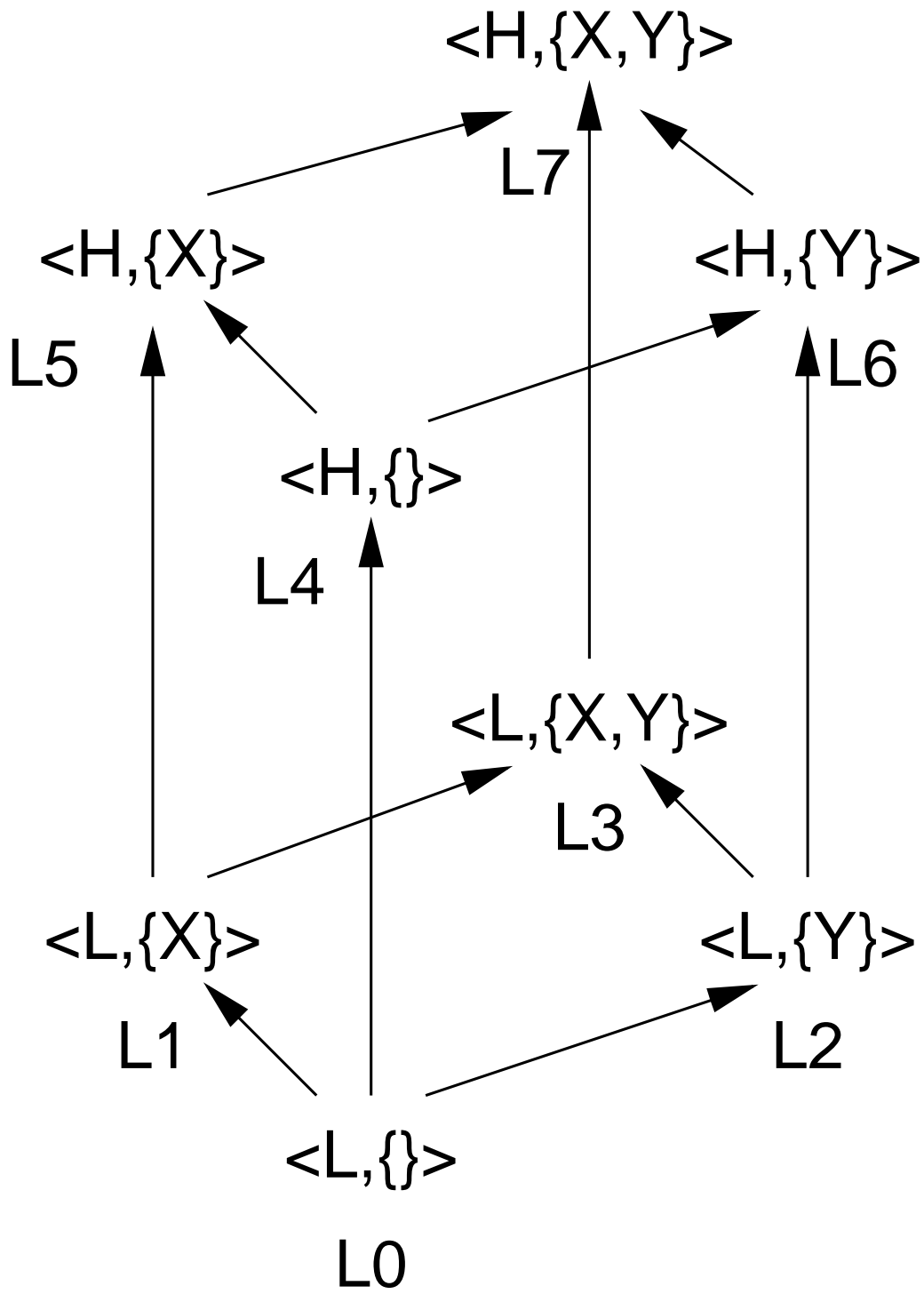


Figure 3: A simple example MLS lattice with labels

- Simple Integrity Property (no write up): A subject  $s$  may write object  $o$  iff  $I(s) \geq I(o)$
- Integrity \*-Property (no read down): A subject  $s$  who may read object  $o$  may also write object  $p$  iff  $I(o) \geq I(p)$

## 3.2 Theoretical Limitations

### 3.2.1 Graham-Denning

- Classic ACM model
- Eight primitive actions
  1. Create/destroy object
  2. Create/destroy subject
  3. Read access rights - determine  $A(s, o)$
  4. Grant access right - owner  $s$  of  $o$  may modify  $A(s', o)$
  5. Delete access right - if  $s$  is owner of  $o$  or  $s$  controls  $s'$  then  $s$  may remove access right from  $A(s', o)$
  6. Transfer access right - if  $A(s, o)$  contains  $r^*$  then  $s$  may copy  $r^*$  (or  $r$  if limited) to  $A(s', o)$

### 3.2.2 Harrison-Ruzzo-Ullman

- Similar to GD model
- Command = Condition then Operation Sequence
- Operation sequence based on primitives (similar to GD)
- Question: Can  $s$  obtain access right  $r$  to  $o$ ?
  - If single operation per command, then decidable
  - Otherwise, not decidable

### 3.2.3 Take-Grant

- Graphical version of ACM models
- Grant arcs - allow  $s$  to grant *any* right  $r$  that  $s$  has to  $o$  to  $s'$  if  $s$  has a grant arc to  $s'$
- Take arcs - allows  $s$  to take (i.e., give to  $s$ ) *any* right  $r$  that  $s'$  has to  $o$  if  $s$  has a take arc to  $s'$

### **3.3 RBAC**

- Named roles associates with rights
- Subjects may bind to roles according to role-binding rights
- Roles in hierarchy with inheritance



## 4 Design

### 4.1 Design elements

#### 4.1.1 Least privilege

#### 4.1.2 Economy of mechanism

#### 4.1.3 Open design

#### 4.1.4 Complete mediation

#### 4.1.5 Permission-based

#### 4.1.6 Separation of privilege

#### 4.1.7 Least common mechanism

## 4.2 Features

4.2.1 User authentication

4.2.2 Memory protection

4.2.3 File & device I/O access control

4.2.4 Object allocation & access control

4.2.5 Sharing enforcement

4.2.6 Fairness

4.2.7 IPC/synchronization

4.2.8 OS protection (esp. protection data protection)

## **4.3 Trusted OS Features**

**4.3.1 User I&A**

**4.3.2 DAC**

**4.3.3 MAC**

**4.3.4 Object reuse protection**

**4.3.5 Complete mediation**

**4.3.6 Audit/audit reduction**

**4.3.7 Trusted path**

**4.3.8 IDS**

## **4.4 Kernelized Design**

### **4.4.1 Reference Monitor**

1. tamperproof
2. always invoked
3. small enough to be trusted

## 4.4.2 TCB

- consists of
  1. H/W
  2. files
  3. protected memory
  4. IPC
- monitors
  1. process activation
  2. execution domain switching
  3. memory protection
  4. I/O operation

## **4.5 Separation/Isolation**

## **4.6 Virtualization**

## **4.7 Layered Design**

1. Layered trust
2. Ring systems
3. Gates

## **5 Assurance**

### **5.1 Flaws**

### **5.2 Assurance methods**

1. Software engineering practice - design/validation
2. Testing
3. Penetration testing (tiger teams)
4. Formal verification

### **5.3 Evaluation**

### 5.3.1 TCSEC - Orange Book

1. Effort to provide standardized terminology and levels of features and assurance for computing systems
2. Four basic levels: A, B, C, D
  - (a) D: Minimal Protection
  - (b) C: Discretionary Protection
    - i. C1: Discretionary Security Protection
    - ii. C2: Controlled Access Protection
  - (c) B: Mandatory Protection
    - i. B1: Labeled Security Protection
    - ii. B2: Structured Protection
    - iii. B3: Security Domains
  - (d) A: Verified Protection
    - i. A1: Verified Design
  - (e)
3. Assurance and features placed on linear scale
4. Limited success
5. Starting point for modern efforts



### 5.3.2 Green Book

1. Separates assurance from feature set
2. Ten feature sets, including five modeling C1 through B3=A1 plus five new ones for communications, databases
3. Eight assurance (quality) levels,  $Q_1$  to  $Q_6$  correspond to TCSEC C1 through A1;  $Q_7$  goes beyond A1 assurance
4. Supported evaluation by independent, commercial evaluation facilities

### **5.3.3 British Evaluation**

1. Foundation of Common Criteria
2. Claims language = action phrases and target phrases with parameters
3. Six assurance levels
4. Sets of claims expected to be bundled for popular features
5. Process specifications for licensed independent evaluators

### **5.3.4 European ITSEC Evaluation**

- combined British and German

### **5.3.5 Canadian Criteria**

- merged into CC

### **5.3.6 US Combined Federal Criteria**

- combined ITSEC and Canadian - merged into CC

### **5.3.7 Common Criteria**

1. Defines classes of interest to security
2. Classes parent families of functions/assurance needs
3. Components combined to make packages for families
4. Packages combined into requirement sets/assertions for products
5. Target of Evaluation (TOE)

## **5.4 Non-assurance**

1. Yelling louder
2. Security through Obscurity
3. Internal penetrate and patch
4. Challenges (external p&p)