Overview

Applying Formal Methods, Part V

Overview

Refinement: security models (mandatory security policies)

- Incorporate information categories in the classification
- Security levels and integrity levels are partially ordered but not necessarily linearly ordered
- Bell-La Padula models (protect confidentiality)
- Biba models (protect integrity)

The End

Biba Model

Commercial Integrity Policies

Commercial Policies

Primary concern: *integrity*

- Protecting system and its resources from damage
  - Contamination
  - Corruption
  - Misuse

Integrity Levels: Examples

- Octane ratings on gas
- Frequent-flyer status: early access to seats and airport lounges

Maintaining quality rather than confidentiality

Integrity Levels

Syntax

$$\textbf{IntLevel} ::= \textbf{IntLabel} \mid \text{il}(\textbf{PName})$$

$$\textbf{Form} ::= \textbf{IntLevel} \leq_i \textbf{IntLevel} \mid \textbf{IntLevel} =_i \textbf{IntLevel}$$

Semantics

- $\mathcal{M} = \langle W, I, J, K, L, \preceq \rangle$
- W, I, J as before
- K is a non-empty set of *integrity levels*.
- $L : (\textbf{IntLabel} \cup \textbf{PName}) \to K$ is a mapping of integrity labels and simple principal names to an integrity level.

  $$L(\text{il}(A)) = L(A),$$

  for every simple principal name A.
- $\preceq \subseteq K \times K$ is a partial order on K

Semantics :-

$\mathcal{M} = (W, I, J)$

W ← ~~old~~ original components

$(K, L, \leq)$ ← new

evaluation function

$E_{\mathcal{M}}[\![l_1 \leq l_2]\!]$

$E_{\mathcal{M}}[\![l_1 = l_2]\!]$;

Kripke Semantics and Inference Rules

Kripke Semantics

$$\mathcal{E}_{\mathcal{M}}[\![\ell_1 \leq_i \ell_2]\!] = \begin{cases} W, & \text{if } L(\ell_1) \preceq L(\ell_2) \\ \emptyset, & \text{otherwise} \end{cases}$$

$$\mathcal{E}_{\mathcal{M}}[\![\ell_1 =_i \ell_2]\!] = \mathcal{E}_{\mathcal{M}}[\![\ell_1 \leq_i \ell_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\ell_2 \leq_i \ell_1]\!].$$

Inference Rules

$$\ell_1 =_i \ell_2 \stackrel{\text{def}}{=} (\ell_1 \leq_i \ell_2) \wedge (\ell_2 \leq_i \ell_1)$$

Relexivity of $\leq_i$ $\quad \dfrac{}{\ell \leq_i \ell}$

Transitivity of $\leq_i$ $\quad \dfrac{\ell_1 \leq_i \ell_2 \qquad \ell_2 \leq_i \ell_3}{\ell_1 \leq_i \ell_3}$

il $\leq_i$ $\quad \dfrac{\mathsf{il}(P) =_i \ell_1 \qquad \mathsf{il}(Q) =_i \ell_2 \qquad \ell_1 \leq_i \ell_2}{\mathsf{il}(P) \leq_i \mathsf{il}(Q)}$

The End

Biba Model

Biba Strict Integrity Policy

Background

Biba definition of computer system (Biba, 1975)

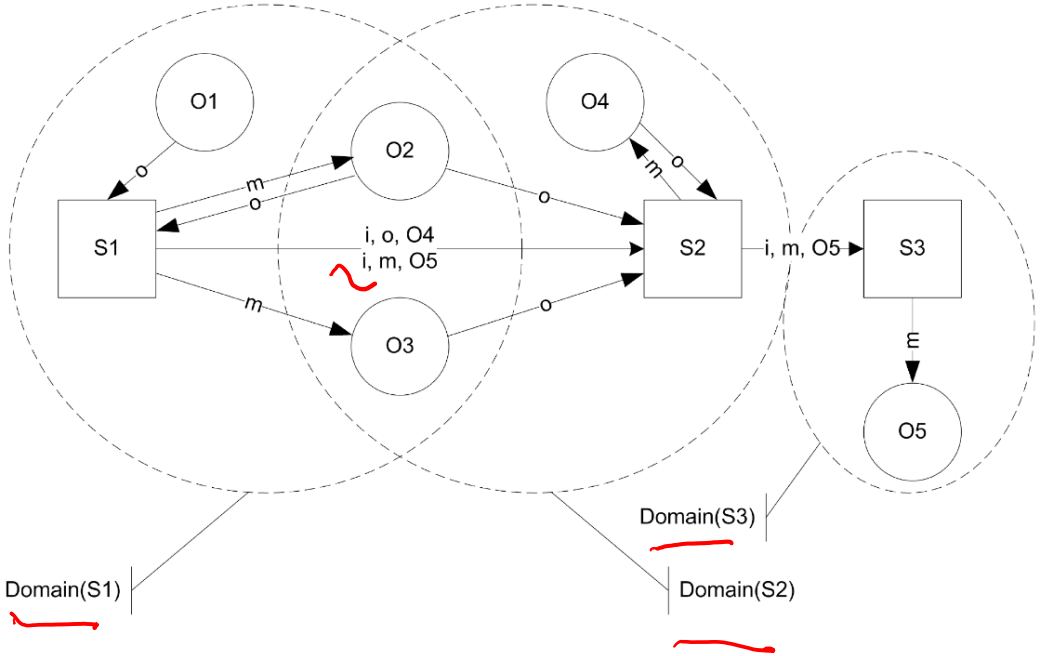
- The concern of computer system integrity is thus the guarantee that a subsystem will perform as it was intended to perform by its creator. We assume that a subsystem has been initially certified (by some system external agency) to perform properly. We then wish to ensure that the subsystem cannot be corrupted to perform in a manner contrary to its certification.

Background (cont.)

Biba definition of computer system (Biba, 1975)

- The integrity problem is the formulation of access control policies and mechanisms that provide a subsystem with the isolation necessary for protection from subversion. Based on an initial assumption of proper behavior (according to some system external standard), we are primarily concerned with protection from intentionally malicious attack: unprivileged, intentionally malicious modification.

Subjects, Objects, and Access



objects

subjects

labels.

Label

Three kinds of access:

1. *Observation:* viewing of information by a subject—includes execution
2. *Modification:* changing an object's state
3. *Invocation:* a request for service by one subject of another

Strict Integrity

Direct operations

- For observations (read, execute)

$$(\ \mathrm{il}(S) \leq_i \ \mathrm{il}(O)) \supset (S \text{ controls } \langle o, O \rangle)$$
$$\neg(\ \mathrm{il}(S) \leq_i \ \mathrm{il}(O)) \supset (S \text{ says } \langle o, O \rangle) \supset \langle trap \rangle$$

- For modifications (write)

$$(\ \mathrm{il}(O) \leq_i \ \mathrm{il}(S)) \supset (S \text{ controls } \langle m, O \rangle)$$
$$\neg(\ \mathrm{il}(O) \leq_i \ \mathrm{il}(S)) \supset (S \text{ says } \langle m, O \rangle) \supset \langle trap \rangle$$

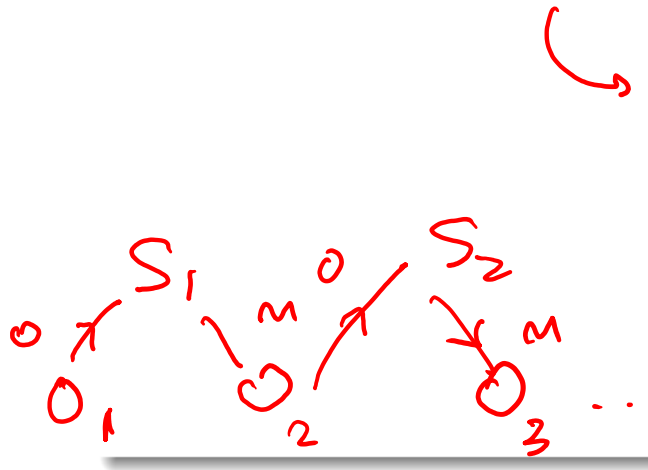
Indirect operations

- Developed in Chapter 13

Transfer Paths

Definition

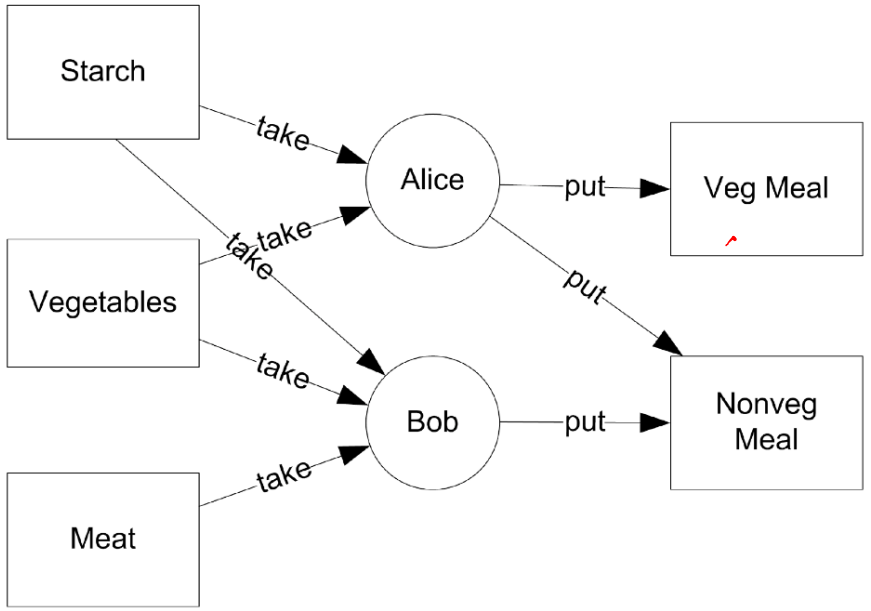
A **transfer path** is a *sequence of objects* $O_1, O_2, \cdots, O_{n+1}$ *and subjects* $S_1, S_2, \cdots, S_n$ such that:



- S_1 can observe/take O_1
- S_1 can modify/put O_2
- S_2 can observe/take O_2
- S_2 can modify/put O_3
- ...
- S_n can observe/take O_n
- S_n can modify/put O_{n+1}

Strict integrity prevents corruption of subjects and objects

Example



Integrity Levels

- NV: non-vegetarian
- V: vegetarian
- ordering: NV $\leq_i$ V

Assignments

Subject or Object	Integrity Level
Alice	V
Bob	NV
Starch	V
Vegetables	V
Meat	NV
Veg Meal	V
Non-veg Meal	NV

Access Matrix

Subject	Starch	Vegetables	Meat	Veg Meal	Non-veg meal
Alice	take	take	-	put	put
Bob	take	take	take	-	put

Strict Integrity Policy for Veg/Non-Veg Meals

from Access matrix

Alice

$$\mathsf{il}(\mathit{Alice}) \leq_i \mathsf{il}(\mathit{Starch}) \supset \mathit{Alice}\ \text{controls}\ \langle \mathit{take}, \mathit{Starch} \rangle$$
$$\mathsf{il}(\mathit{Alice}) \leq_i \mathsf{il}(\mathit{Vegetables}) \supset \mathit{Alice}\ \text{controls}\ \langle \mathit{take}, \mathit{Vegetables} \rangle$$
$$\mathsf{il}(\mathit{veg\ meal}) \leq_i \mathsf{il}(\mathit{Alice}) \supset \mathit{Alice}\ \text{controls}\ \langle \mathit{put}, \mathit{veg\ meal} \rangle$$
$$\mathsf{il}(\mathit{non\text{-}veg\ meal}) \leq_i \mathsf{il}(\mathit{Alice}) \supset \mathit{Alice}\ \text{controls}\ \langle \mathit{put}, \mathit{non\text{-}veg\ meal} \rangle$$

Bob

$$\mathsf{il}(\mathit{Bob}) \leq_i \mathsf{il}(\mathit{Starch}) \supset \mathit{Bob}\ \text{controls}\ \langle \mathit{take}, \mathit{Starch} \rangle$$
$$\mathsf{il}(\mathit{Bob}) \leq_i \mathsf{il}(\mathit{Vegetables}) \supset \mathit{Bob}\ \text{controls}\ \langle \mathit{take}, \mathit{Vegetables} \rangle$$
$$\mathsf{il}(\mathit{Bob}) \leq_i \mathsf{il}(\mathit{Meat}) \supset \mathit{Bob}\ \text{controls}\ \langle \mathit{take}, \mathit{Meat} \rangle$$
$$\mathsf{il}(\mathit{non\text{-}veg\ meal}) \leq_i \mathsf{il}(\mathit{Bob}) \supset \mathit{Bob}\ \text{controls}\ \langle \mathit{put}, \mathit{non\text{-}veg\ meal} \rangle$$

Integrity of transfer path from ingredients to meals preserved

Section 3.5.4 (Detailed discussion of the example).

The End

Confidentiality and Integrity Levels

Confidentiality Level and Integrity Level Relations

Military Security Policies

Primary concern: *confidentiality*

- Information is protected on a *need to know* basis
- Flow of information is governed by classification levels, typically
  - UNCLASSIFIED, CONFIDENTIAL, SECRET, and TOP SECRET
  - $\text{UC} \leq_s \text{C} \leq_s \text{S} \leq_s \text{TS}$

Bell La Padula Model

- Subjects cannot read information at higher classification levels: *"no read up"*
- Subjects cannot write (leak) information to lower classification levels: *"no write down"*

Adding Security Levels to Kripke Semantics

Syntax

$$\textbf{SecLevel} ::= \textbf{SecLabel} \mid \mathsf{sl}(\textbf{PName})$$

$$\textbf{Form} ::= \textbf{SecLevel} \leq_s \textbf{SecLevel} \mid \textbf{SecLevel} =_s \textbf{SecLevel}$$

Semantics

- $\mathcal{M} = \langle W, I, J, K, L, \preceq \rangle$
- W, I, J as before
- K is a non-empty set of *security levels*.
- $L : (\textbf{SecLabel} \cup \textbf{PName}) \to K$ is a mapping of security labels and simple principal names to an security level.

  $$L(\mathsf{sl}(A)) = L(A),$$

  for every simple principal name A.
- $\preceq \subseteq K \times K$ is a partial order on K

Kripke Semantics and Inference Rules

Kripke Semantics

$$\mathcal{E}_{\mathcal{M}}[\![\ell_1 \leq_s \ell_2]\!] = \begin{cases} W, & \text{if } L(\ell_1) \preceq L(\ell_2) \\ \emptyset, & \text{otherwise} \end{cases}$$

$$\mathcal{E}_{\mathcal{M}}[\![\ell_1 =_s \ell_2]\!] = \mathcal{E}_{\mathcal{M}}[\![\ell_1 \leq_s \ell_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\ell_2 \leq_s \ell_1]\!].$$

Inference Rules

$$\ell_1 =_s \ell_2 \stackrel{\text{def}}{=} (\ell_1 \leq_s \ell_2) \wedge (\ell_2 \leq_s \ell_1)$$

Relexivity of $\leq_s$ $\quad \dfrac{}{\ell \leq_s \ell}$

Transitivity of $\leq_s$ $\quad \dfrac{\ell_1 \leq_s \ell_2 \qquad \ell_2 \leq_s \ell_3}{\ell_1 \leq_s \ell_3}$

$sl \leq_s$ $\quad \dfrac{\mathsf{sl}(P) =_s \ell_1 \quad \mathsf{sl}(Q) =_s \ell_2 \quad \ell_1 \leq_s \ell_2}{\mathsf{sl}(P) \leq_s \mathsf{sl}(Q)}$

partial order is a relation that is reflexive anti-symmetric & transitive

Commercial Policies

Primary concern: *integrity*

- Protecting system and its resources from damage
  - Contamination
  - Corruption
  - Misuse

Integrity Levels: Examples

- Octane ratings on gas
- Frequent-flyer status: early access to seats and airport lounges

Maintaining quality rather than confidentiality

Kripke Semantics and Inference Rules

Kripke Semantics

$$\mathcal{E}_{\mathcal{M}}[\![\ell_1 \leq_i \ell_2]\!] = \begin{cases} W, & \text{if } L(\ell_1) \preceq L(\ell_2) \\ \emptyset, & \text{otherwise} \end{cases}$$

$$\mathcal{E}_{\mathcal{M}}[\![\ell_1 =_i \ell_2]\!] = \mathcal{E}_{\mathcal{M}}[\![\ell_1 \leq_i \ell_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\ell_2 \leq_i \ell_1]\!].$$

Inference Rules

$$\ell_1 =_i \ell_2 \stackrel{\text{def}}{=} (\ell_1 \leq_i \ell_2) \wedge (\ell_2 \leq_i \ell_1)$$

Relexivity of $\leq_i$ $\quad \dfrac{}{\ell \leq_i \ell}$

Transitivity of $\leq_i$ $\quad \dfrac{\ell_1 \leq_i \ell_2 \qquad \ell_2 \leq_i \ell_3}{\ell_1 \leq_i \ell_3}$

$il \leq_i$ $\quad \dfrac{\mathsf{il}(P) =_i \ell_1 \quad \mathsf{il}(Q) =_i \ell_2 \quad \ell_1 \leq_i \ell_2}{\mathsf{il}(P) \leq_i \mathsf{il}(Q)}$

Partial-Order Relation

- A relation ≤ on a set *A* (i.e., ≤ $\subseteq$ *A* x *A*) is a *partial order* provided ≤ is *reflexive*, ***anti****-symmetric*, and *transitive*
- In addition, if the partial order ≤ satisfies the totality condition:

$$\forall\, x, y \in A, \qquad either\ x \leq y\ or\ y \leq x$$

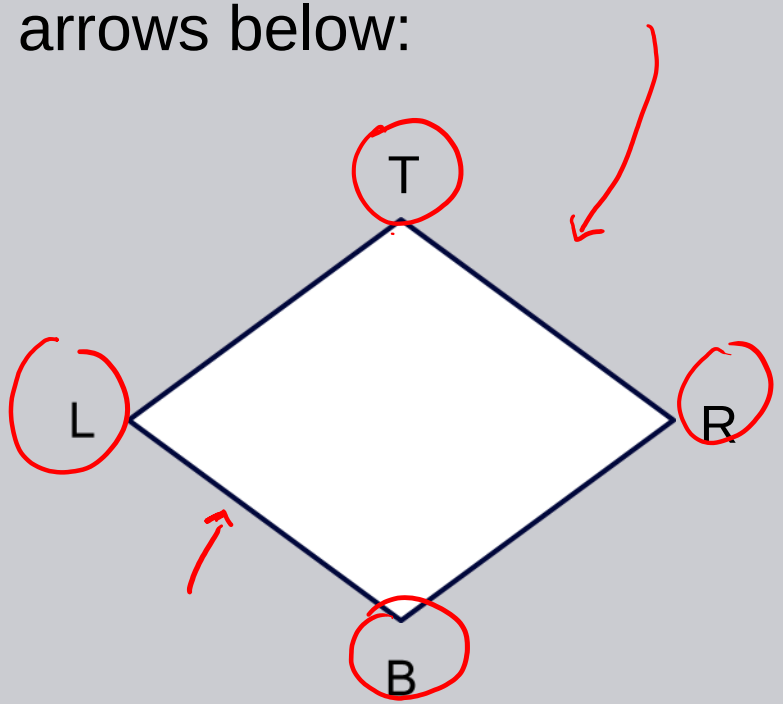
- We will call the relation ≤ a *total order*

Hasse Diagram

- When *A* is finite, a *partial order* ≤ on *A* is often represented by a *Hasse* diagram, which is a directed graph *H*, where:
  - Each element in *A* is a vertex of *H*
  - An edge from *x* (“higher”) to *y* (“lower”) is drawn whenever $x \neq y$, $x \leq y$ *and there is no* z ($z \neq x$, $z \neq y$) with $x \leq z \leq y$
- *Confidentiality and integrity levels for the extended Kripke structures are represented by Hasse diagrams*

Example

P: a relation that is a partial order but not a total order

Relation P	Hasse diagram
Fill in the blanks below: → Transitive P = { (T, T), (L, L), (R, R), (B, B), (B, L), (B, R), (B, T), (L, T), (R, T), . . . }	Add arrows below: 

The End

Confidentiality and Integrity Levels

Classification and Categories

Classifications and Categories

Categories

- Information can be sorted into categories, e.g., US, Nuclear, Europe, Asia, etc.
- Categories and classification levels, e.g., unclassified, confidential, secret, top secret, etc., are combined to form a *partial ordering* into a clearance level (L, C) where L is a classification level and $C \subseteq Cat$, where Cat is a set of categories.
- Categories are ordered by *subset*

Partial Ordering of Classification Levels

Properties defining Partial Orders

- **Reflexivity**: For all classification levels L, $L \leq L$.
- **Transitivity**: For all classification levels L_1, L_2, and L_3, if $L_1 \leq L_2$ and $L_2 \leq L_3$, then $L_1 \leq L_3$.
- **Anti-symmetry**: For all classification levels L_1 and L_2, if $L_1 \leq L_2$ and $L_2 \leq L_1$, then $L_1 = L_2$ (i.e., no cycles exist in the relation $\leq$).

Example

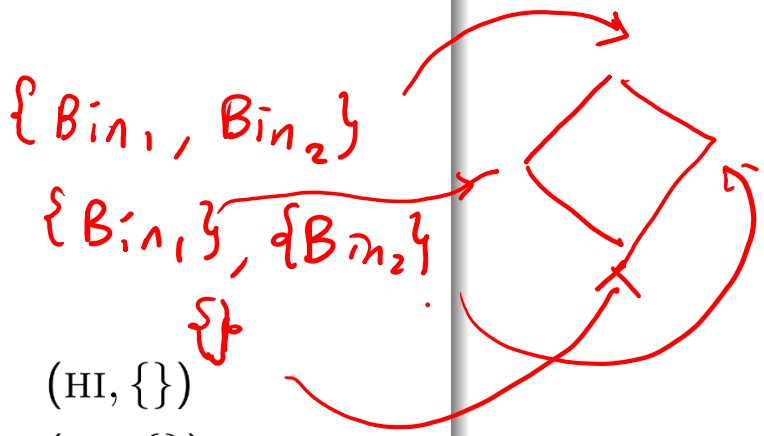
Example 13.1

- Set of classification levels $\mathcal{L} = \{\text{HI}, \text{LO}\}$, with the partial order $\leq$ defined over $\mathcal{L}$ as follows:

$$\text{LO} \leq \text{HI}.$$

- The set of categories is the set $Cat = \{\text{BIN}_1, \text{BIN}_2\}$.
- Eight possible compound levels:

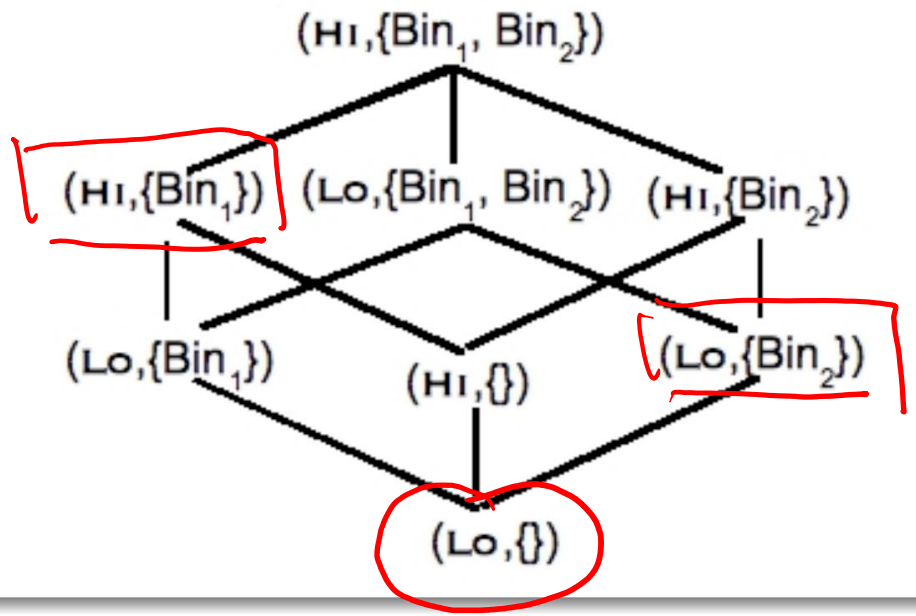
$$\begin{array}{cccc} (\text{HI}, \{\text{BIN}_1, \text{BIN}_2\}) & (\text{HI}, \{\text{BIN}_1\}) & (\text{HI}, \{\text{BIN}_2\}) & (\text{HI}, \{\}) \\ (\text{LO}, \{\text{BIN}_1, \text{BIN}_2\}) & (\text{LO}, \{\text{BIN}_1\}) & (\text{LO}, \{\text{BIN}_2\}) & (\text{LO}, \{\}) \end{array}$$



Draw dom $\subseteq \mathcal{L} \times$ Cat

Example 13.1, continued

Hasse Diagram



Bin1 :: Navy

Bin2 :: Airforce

Hi : U S

Lo : U C

incomparable.

every item

Example 13.2

→ Chapter 13

Example 13.3.

The End

Bell-La Padula Model, Revisited

The Revised Bell-La Padula Model

Bell-La Padula, Revisited

Review

- **Simple security condition:** A principal P can read object O *if and only if* P's confidentiality level is at least as high as O's and P has discretionary read access to O.
- ***-property:** A principal P can write to object O *if and only if* O's confidentiality level is at least as high as P's and P has discretionary write access to O.

Access control and safety

Access control condition

Conditions under which a reference monitor should grant access to O.

- If P's confidentiality level is at least as high as O's and P has discretionary read access to O, then P can read O.
- Describable in access control logic.

Safety condition

Applies to the entire system: must be verified by checking every potential consequence of the rules and mechanisms governing access.

- If P can read O, then P's confidentiality level is at least as high as O's and P has discretionary read access to O.
- Must be verified system wide.

The End

Bell-La Padula Model, Revisited

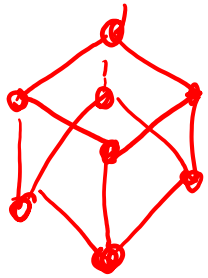
Examples

Example 13.4

Informal Description

Subject or Object	Confidentiality Level
Carol	(HI, {BIN$_1$, BIN$_2$})
Kate	(LO, {BIN$_2$})
O_1	(HI, {BIN$_1$, BIN$_2$})
O_2	(LO, {BIN$_2$})
O_3	(LO, {BIN$_1$})
O_4	(LO, {})

	Object			
Subject	O_1	O_2	O_3	O_4
Carol	read	-	read	-
Kate	write	-	-	read



Formal Description

$$(\ \mathsf{sl}(O_1) \leq_s \mathsf{sl}(Carol)) \supset (Carol \text{ controls } \langle read, O_1\rangle),$$
$$(\ \mathsf{sl}(O_1) \leq_s \mathsf{sl}(Carol)) \supset (Carol \text{ controls } \langle read, O_3\rangle),$$
$$(\ \mathsf{sl}(Kate) \leq_s \mathsf{sl}(O_1)) \supset (Kate \text{ controls } \langle write, O_1\rangle),$$
$$(\ \mathsf{sl}(O_4) \leq_s \mathsf{sl}(Kate)) \supset (Kate \text{ controls } \langle read, O_4\rangle).$$

Example 13.4

Discussions

How to verify

"No read up"

"No write down"

in a general setting

where $\leq_S$ is partially (but not totally ordered).

Example 13.5

Trusted vs. Trustworthy

- Dexter is a corrupt system administrator who has supervisory privileges in an organization's information system. Because Dexter is trusted (but not trustworthy), he is in a position to bypass the security of the system.
- Prior to being fired, Dexter installs a back door so that he can rewrite personnel records. He alters the reference monitor that guards the personnel records so that it accepts a secret identifier:

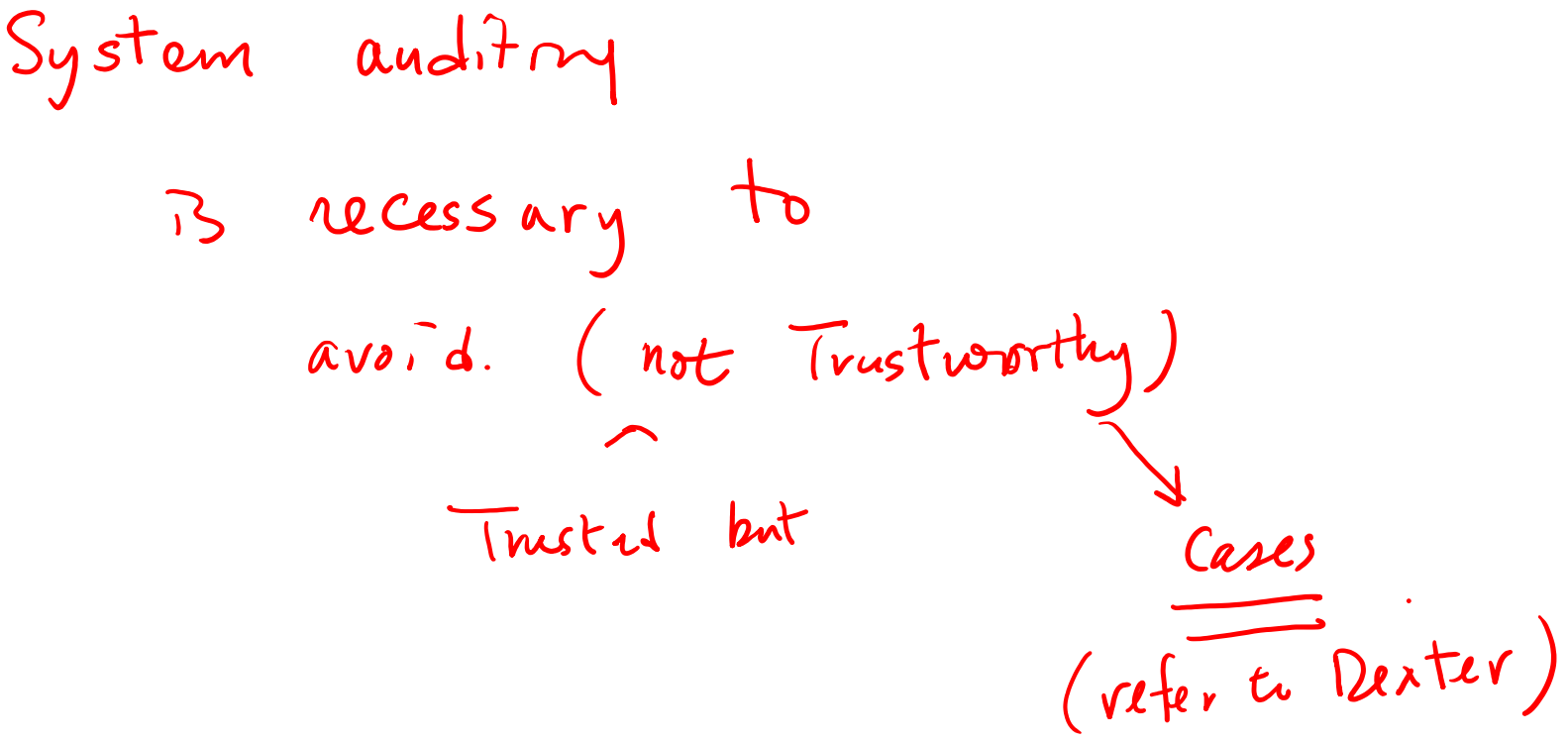
$$\textit{secretID} \text{ controls } \langle \textit{write}, \text{personnel records} \rangle.$$

- By doing so, Dexter has bypassed all the mandatory access-control checks. This back door can be detected only by auditing Dexter's actions and by inspecting the subsystems accessed by Dexter: no reference monitor can possibly detect that Dexter's change violates the access policy.

System auditing.

Example 13.5

Discussions



Confidentiality Levels: Some Practical Considerations

People may reduce their security level

- Illustrates the difference between people and processes
- People can choose to use systems below their confidentiality level

Example 13.6

- Kamal has a confidentiality level of (TS, {NUC, US, EUR, ASIA}), and his staff engineer Sarah has a confidentiality level of (TS, {NUC, US}).
- To communicate with Sarah, Kamal logs onto a TS system that has only NUC,US as allowable categories, and sends his message.
- Having reduced his current confidentiality level to Sarah's, any message he writes will be readable by Sarah.

Example 13.6

Discussions

Practical Considerations

security level may be lowered

to handle. ~~adequat~~ adequate.

flow of information.

The End

Biba Models, Revisited

The Revised Biba Model

Beginning Remarks

- The Biba model introduced earlier is a simplified version that focuses on direct access operations (i.e., observations and modifications).
- We will now address indirect access operations (i.e., invocations of one subject by the other).

Strict Integrity

Direct operations

- For observations (read, execute)

$$(\ \mathrm{il}(S) \leq_i \ \mathrm{il}(O)) \supset (S \text{ controls } \langle o, O \rangle)$$
$$\neg(\ \mathrm{il}(S) \leq_i \ \mathrm{il}(O)) \supset (S \text{ says } \langle o, O \rangle) \supset \langle trap \rangle$$

- For modifications (write)

$$(\ \mathrm{il}(O) \leq_i \ \mathrm{il}(S)) \supset (S \text{ controls } \langle m, O \rangle)$$
$$\neg(\ \mathrm{il}(O) \leq_i \ \mathrm{il}(S)) \supset (S \text{ says } \langle m, O \rangle) \supset \langle trap \rangle$$

Indirect operations

- Developed in Chapter 13

How about i

Biba's Strict Integrity, Revisited

Guiding Principles

1. **Simple integrity condition:** A subject S can observe O *if and only if* $\text{il}(S) \leq \text{il}(O)$ and S has discretionary observe access to O.
2. **Integrity *-property:** A subject S can modify O *if and only if* $\text{il}(O) \leq \text{il}(S)$ and S has discretionary modify access to O.
3. **Invocation condition:** A subject S_1 and invoke subject S_2 *if and only if* $\text{il}(S_2) \leq \text{il}(S_1)$ and S_1 has discretionary invocation rights to S_2.

Maintain integrity of transfer paths

Strict integrity requires that the integrity level of each object and subject in a transfer path must be at least as high as that of the subject or object that immediately follows it.

Strict-Integrity in the logic

Access-control conditions

1. **Simple-integrity access-control condition:**

$$(\ \mathrm{il}(S) \leq\ \mathrm{il}(O)) \supset (S \text{ controls } \langle observe, O\rangle)$$

2. **Integrity *-property access-control condition:**

$$(\ \mathrm{il}(O) \leq\ \mathrm{il}(S)) \supset (S \text{ controls } \langle modify, O\rangle)$$

3. **Invocation access-control conditions:**

$$\begin{cases} (\ \mathrm{il}(S_2) \leq_i\ \mathrm{il}(S_1)) \supset (S_1 \text{ controls } \langle i, S_2, o, O\rangle), \\ (\ \mathrm{il}(S_2) \leq_i\ \mathrm{il}(S_1)) \supset (S_1 \text{ controls } \langle i, S_2, m, O\rangle), \end{cases}$$

where $\langle i, S, o, O\rangle$ and $\langle i, S, m, O\rangle$ respectively denote the propositions *"invoke subject S to observe O"* and *"invoke subject S to modify O."*

Example

Discussions (Example 13.10)

Table 13.3

Integrity levels

↑

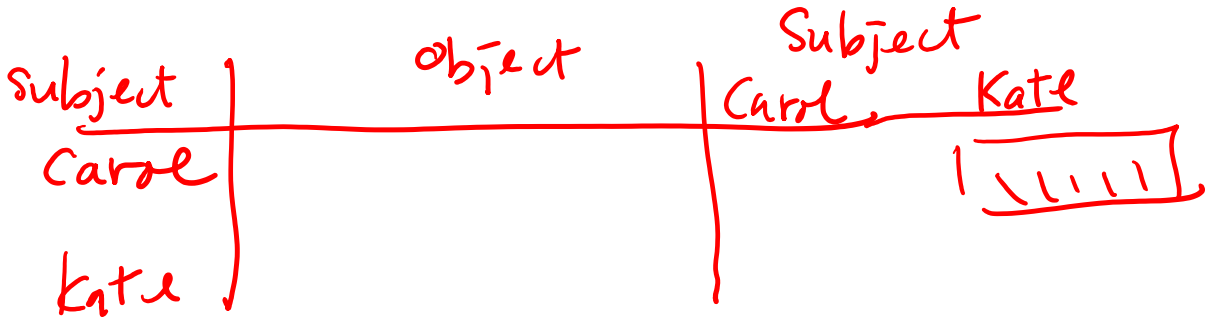
partially ordered

(can have incomparable levels)

Table 13.4

Discretionary

Access Control Matrix



The End

Weekly Summary

Applying Formal Methods V

Refinement of Security Models

- In many situations, security and integrity levels are compound levels that include a classification level with a set of categories. They are partial orders (in general,) often presented using a Hasse diagram.
- The Bell-La Padula and Biba policies implicitly include both implications: an access-control condition and safety property (system-wide).

Refinement of Security Models (cont.)

- Access-control conditions for the two models can be expressed using the language defined for their extended Kripke models.
- The access-control conditions of the Biba model include indirect access (i.e., the invocation of one subject by another) and are also describable using the language for its extended Kripke model. Hence, it can be verified rigorously.

The End