THE UNIVERSITY OF BRITISH COLUMBIA

Security Policies

EECE 412

Session 12

Last Session Recap

- Access Matrix
  - Implementation approaches
    - C-lists
    - ACLs
- Security policies
  - Types of Access Control
    - DAC
    - MAC
    - ORCON
  - CIA
  - Confidentiality policy
  - Integrity policy

Outline

Security policies

- Confidentiality policies
  - Bell LaPadula confidentiality model
- Integrity Policies
  - Biba integrity model
  - Clark-Wilson integrity model

THE UNIVERSITY OF BRITISH COLUMBIA

Confidentiality Policies

What's Confidentiality?

- *X* set of entities, *I* information
- *I* has *confidentiality* property with respect to *X* if
  - no $x \in X$ can obtain information from *I*
  - *I* can be disclosed to others

- Examples?

What's Confidentiality Policy

- Goal: prevent the unauthorized disclosure of information
  - Deals with information flow
  - Integrity incidental
- Multi-level security models are best-known examples
  - Bell-LaPadula Model basis for many, or most, of these

Bell-LaPadula Model, Step 1

- Security levels arranged in linear ordering
- Example:
  - Top Secret: highest
  - Secret
  - Confidential
  - Unclassified: lowest
- Subjects have *security clearance* $L(s)$
- Objects have *security classification* $L(o)$

Example

security level	*subject*	*object*
Top Secret	Alice	Personnel Files
Secret	Bob	E-Mail Files
Confidential	Chiang	Activity Logs
Unclassified	Fred	Telephone Lists

- Alice can read all files
- Chiang cannot read Personnel or E-Mail Files
- Fred can only read Telephone Lists

Reading Information

- Information flows *up*, not *down*
  - "Reads up" disallowed, "reads down" allowed
- Simple Security Property
  - Subject *s* can read object *o* iff, $L(o) = L(s)$ and *s* has permission to read *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no reads up" rule

Writing Information

- Information flows up, not down
  - "writes up" allowed, "writes down" disallowed
- *-Property
  - Subject *s* can write object *o* iff $L(s) = L(o)$ and *s* has permission to write *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no writes down" rule

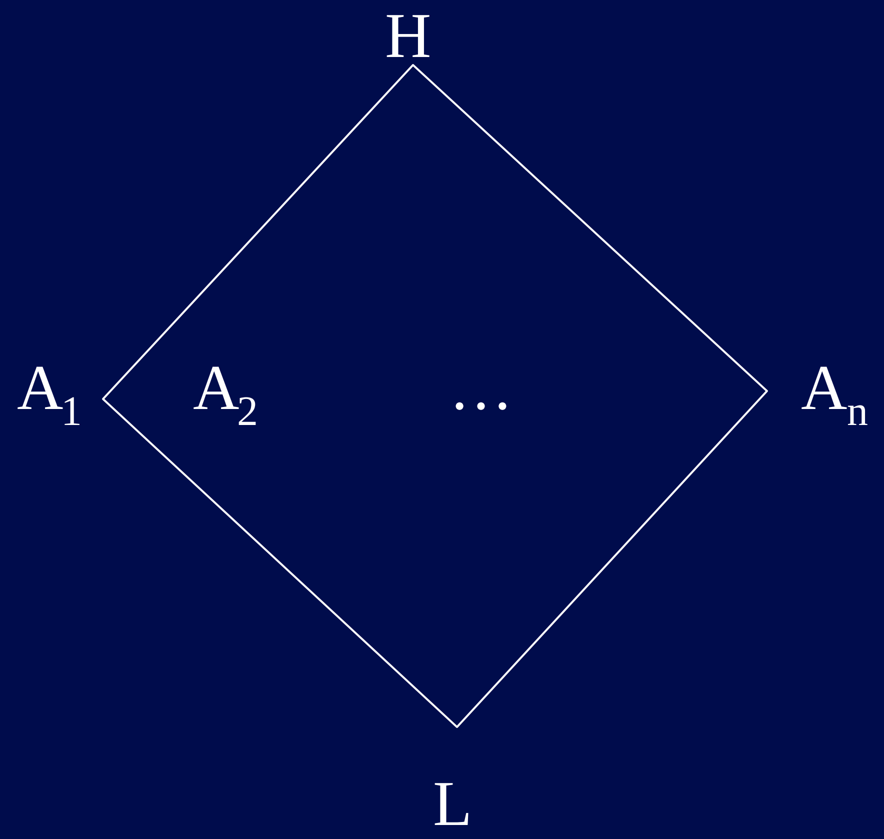
Bell-LaPadula Model, Step 2

- Expand notion of security level to include categories
- Security level is (*clearance*, *category set*)
- Examples
  - (Top Secret, { NUC, EUR, ASI })
  - (Confidential, { EUR, ASI })
  - (Secret, { NUC, ASI })

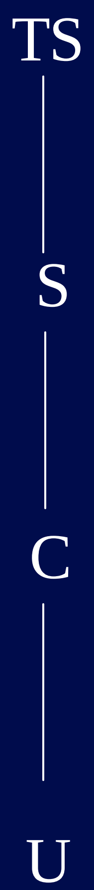
Levels and Lattices

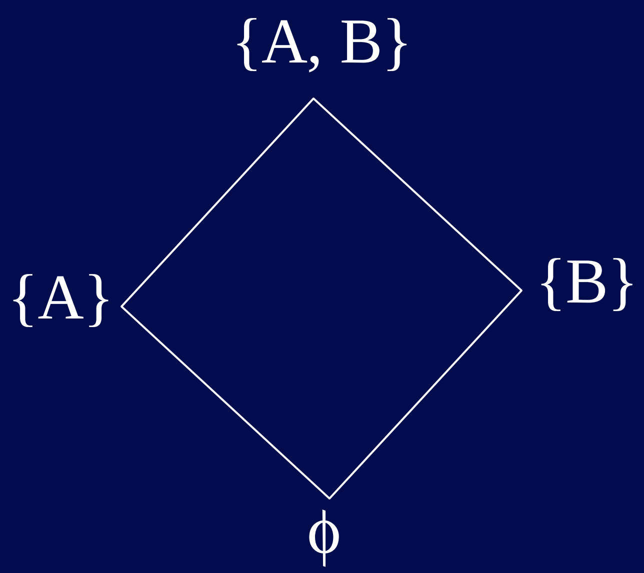
- (A, C) *dominates* (A¢, C¢) iff $A' = A$ and C¢ $\subseteq C$
- Examples
  - (Top Secret, {NUC, ASI}) *dom* (Secret, {NUC})
  - (Secret, {NUC, EUR}) *dom* (Confidential,{NUC, EUR})
  - (Top Secret, {NUC}) $\neg$*dom* (Confidential, {EUR})
- Let C be set of classifications, K set of categories. Set of security levels $L = C \times K$, *dom* form lattice

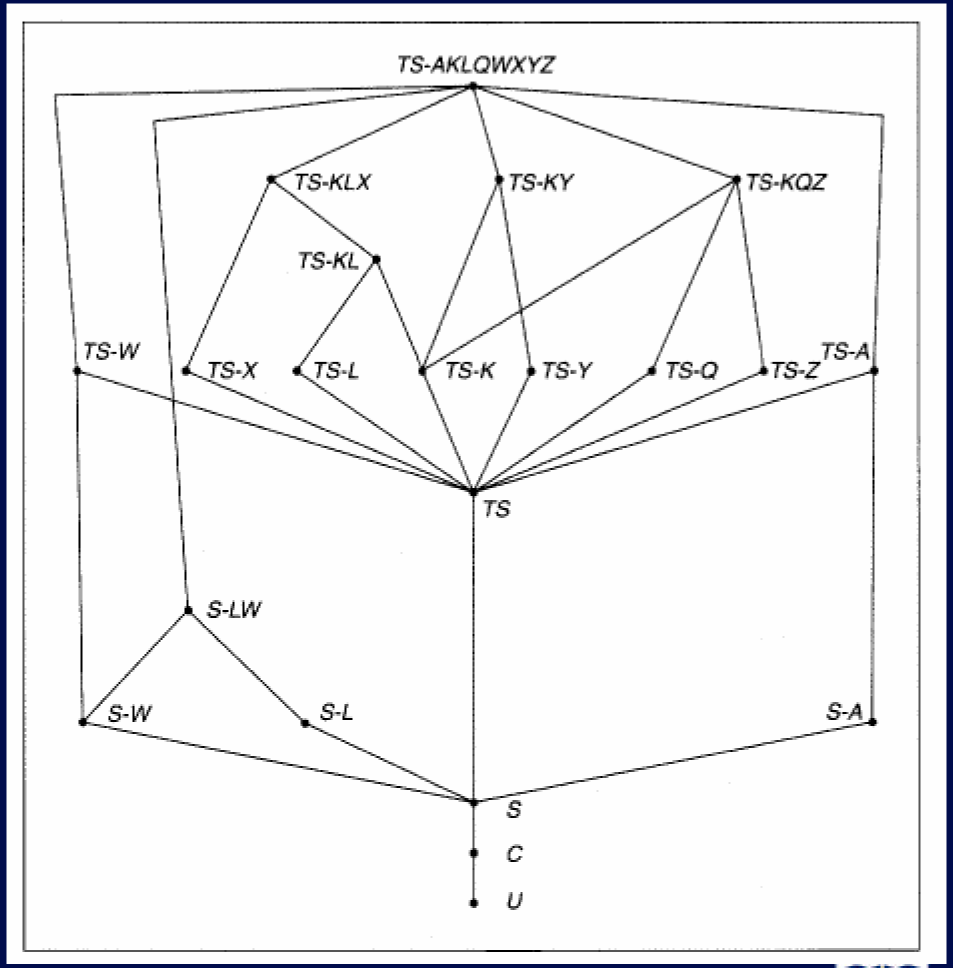
Bounded Isolated Classes



The Military Lattice







Levels and Ordering

- Security levels partially ordered
  - Any pair of security levels may (or may not) be related by "*dominates*" relation
- Note:
  - "dominates" serves the role of "greater than"
  - "greater than" is a total ordering, though

Reading Information

- Information flows *up*, not *down*
  - "reads up" disallowed, "reads down" allowed
- Simple Security Property (Step 2)
  - Subject *s* can read object *o* iff *L*(*s*) *dom L*(*o*) and *s* has permission to read *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no reads up" rule

Writing Information

- Information flows up, not down
  - “Writes up” allowed, “writes down” disallowed
- *-Property (Step 2)
  - Subject *s* can write object *o* iff *L*(*o*) *dom L*(*s*) and *s* has permission to write *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called “no writes down” rule

Groups

Group 1	Group 2	Group 3	Group 4	Group 5
Chiang, Joyce	Tsang, Jeannette	Ong, Tieng Pei	Wei, Qiang	Tung, Jeffrey
Huang, Ben	Li-Heng Lin, Mike	Kler, Jeffrey	Fong, Claudia	Kan, Jason
Darwish, Wesam	Wong, Chun-Yue	Milojkovic, Aleksandar	Lee, Larix	Tsai, Johnson
Kwan, Michael	Markandan, Kartik	Chow, Jacqueline	Handoko, Handika	Hung,Wallace
Chan,Ryan	Woo,Wing Keong	Leung, Wing	Lee, Johnson	Cheuk Lun
Tse, Janet	Lau, Ivan	Yen, Horng	Leung,Michael	Chang, Steven
Yan Ha,Shu	Vo,Tuan Ann	Elizabeth-Tiedje, Megan	Yeung, Derrick	Lai, Kevin
Zhao, Samson	Cheung, Jason	Li, John	Lam, Victor	

Each Group

Develop configuration (i.e., label graph, and clearance and classification assignments) for access control mechanisms based on Bell-LaPadula model for the following application and policy

Application:

- 10 students: s_1 … s_{10}
- 3 instructors: i_1, i_2, i_3
- 5 courses: c_1, … c_5
  - $C_1 = \{i_1, \{s_1, s_2, s_3\}\}$
  - $C_2 = \{i_2, \{s_3, s_4, s_5\}\}$
  - $C_3 = \{i_3, \{s_5, s_6, s_7\}\}$
  - $C_4 = \{i_1, \{s_7, s_8, s_9\}\}$
  - $C_5 = \{\{i_2, i_3\}, \{s_8, s_9, s_{10}\}\}$

Policy:

1. Students can
   1. read course material and assignment instructions for the courses they are registered
   2. submit (i.e., write) their assignments for the registered courses
2. Instructors can
   1. read student submitted assignments for the courses they teach, and
   2. post (i.e., write) course material and assignment instructions for their courses

Key Points Regarding Confidentiality Policies

- Confidentiality policies restrict flow of information
- Bell-LaPadula model supports multilevel security
  - Cornerstone of much work in computer security policies

Integrity Policies

Biba Integrity Model (1977)

- Set of subjects S, objects O, integrity levels I, relation $\leq \subseteq I \times I$ holding when second dominates first or same
- min: $I \times I \rightarrow I$ returns lesser of integrity levels
- i: $S \cup O \rightarrow I$ gives integrity level of entity
- $\underline{r}$: $S \times O$ means $s \in S$ can read $o \in O$
- $\underline{w}$: $S \times O$ means $s \in S$ can write $o \in O$
- $\underline{x}$: $S \times O$ means $s \in S$ can execute $o \in O$

What does a higher integrity level of an object mean?

Intuition for Integrity Levels

- The higher the level, the more confidence
  - That a program will execute correctly
  - That data is accurate and/or reliable
- Note relationship between integrity and trustworthiness
- Important point: *integrity levels are **not** security levels*

Low-Water-Mark Policy

- Idea: when s reads o, $i(s) = min(i(s), i\ (o))$; s can only write objects at lower levels
- Rules
  1. $s \in S$ can write to $o \in O$ if and only if $i(o) = i(s)$.
  2. If $s \in S$ reads $o \in O$, then *i*¢(*s*) = $min(i(s), i(o))$, where *i*¢(*s*) is the subject's integrity level after the read.
  3. $s_1 \in S$ can execute $s_2 \in S$ if and only if $i(s_2) = i(s_1)$.

Problems

- Subjects' integrity levels decrease as system runs
  - Soon no subject will be able to access objects at high integrity levels
- Alternative: change object levels rather than subject levels
  - Soon all objects will be at the lowest integrity level

Ring Policy

- Idea: subject integrity levels static
- Rules
  1. $s \in S$ can write to $o \in O$ if and only if $i(o) = i(s)$.
  2. Any subject can read any object.
  3. $s_1 \in S$ can execute $s_2 \in S$ if and only if $i(s_2) = i(s_1)$.
- Eliminates indirect modification problem

Strict Integrity Policy (a.k.a., “Biba’s Model”)

- Similar to Bell-LaPadula model
  1. $s \in S$ can read $o \in O$ iff $i(s) = i(o)$
  2. $s \in S$ can write to $o \in O$ iff $i(o) = i(s)$
  3. $s_1 \in S$ can execute $s_2 \in S$ iff $i(s_2) = i(s_1)$
- Add compartments and discretionary controls to get full dual of Bell-LaPadula model

LOCUS and Biba

- Goal: prevent untrusted software from altering data or other software
- Approach: make levels of trust explicit
  - *credibility rating* based on estimate of software's trustworthiness (0 untrusted, *n* highly trusted)
  - *trusted file systems* contain software with a single credibility level
  - Process has *risk level* or highest credibility level at which process can execute
  - Must use *run-untrusted* command to run software at lower credibility level

Clark-Wilson Integrity Model

Model

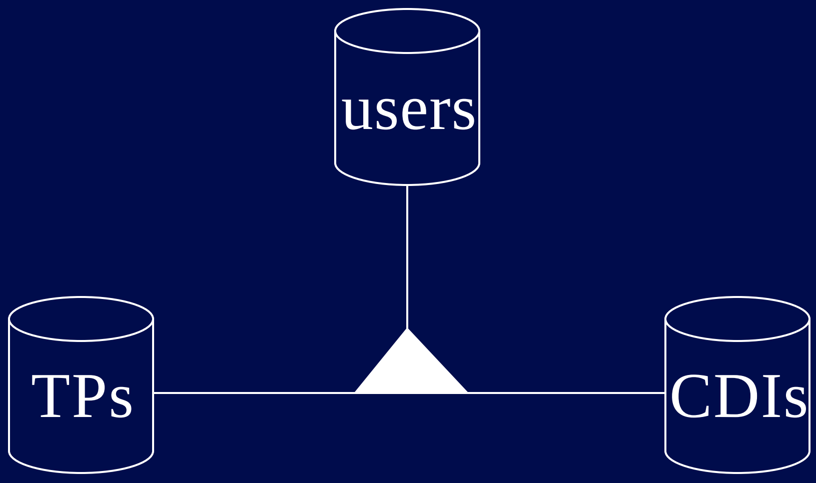
- Integrity defined by a set of constraints
  - Data in a *consistent* or valid state when it satisfies these
- Example: Bank
  - *D* today's deposits, *W* withdrawals, *YB* yesterday's balance, *TB* today's balance
  - Integrity constraint: $YB + D - W = TB$
- *Well-formed transaction* move system from one consistent state to another
- Issue: who examines, certifies transactions done correctly?
  - The principle of separation of duty

Entities in the Model

- CDIs: constrained data items
  - Data subject to integrity controls
- UDIs: unconstrained data items
  - Data not subject to integrity controls
- IVPs: integrity verification procedures
  - Procedures that test the CDIs conform to the integrity constraints
- TPs: transaction procedures
  - Procedures that take the system from one valid state to another

The Idea

Constrain who can do what by defining authorized triples: (user, TP, {CDI})



Key Points

- Integrity policies deal with trust
  - As trust is hard to quantify, these policies are hard to evaluate completely
  - Look for assumptions and trusted users to find possible weak points in their implementation
- Biba, Lipner based on multilevel integrity
- Clark-Wilson focuses on separation of duty and transactions

Next Session Preview

Hybrid policies

- Chinese Wall model
- Role-based access control model