Confidentiality Policies

CS461/ECE422 Computer Security I

Fall 2009

Guest Lecture by:

Omid Fatemieh

Reading

- Chapter 5 in CS:
  - 5.1 and 5.2 up to the beginning of 5.2.3
  - 5.3
- Chapter 30 in CS (Lattices)
- Bell-LaPadula and McLean papers linked on class web site if you are interested in the proofs

Outline

- Overview
  - Mandatory versus discretionary controls
  - What is a confidentiality model
- Bell-LaPadula Model
  - General idea
  - Description of rules
- Tranquility

MAC vs DAC

- Discretionary Access Control (DAC)
  - Normal users can change access control state directly assuming they have appropriate permissions
  - Access control implemented in standard OS's, e.g., Unix, Linux, Windows
  - Access control is at the discretion of the user
- Mandatory Access Control (MAC)
  - Access decisions cannot be changed by normal rules
  - Generally enforced by system wide set of rules
  - Normal user cannot change access control schema
- "Strong" system security requires MAC
  - Normal users cannot be trusted

Confidentiality Policy

- Goal: prevent the unauthorized disclosure of information
  - Deals with information flow
  - Unauthorized alteration of information is secondary
- 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
  - Top Secret: highest
  - Secret
  - Confidential
  - Unclassified: lowest
- Levels consist of *security clearance* $L(s)$
  - Objects have *security classification* $L(o)$

Bell, LaPadula 73

Example

security level	*subject*	*object*
Top Secret	Tamara	Personnel Files
Secret	Samuel	E-Mail Files
Confidential	Claire	Activity Logs
Unclassified	Bob	Telephone Lists

- Tamara can read all files
- Claire cannot read Personnel or E-Mail Files
- Bob can only read Telephone Lists

Reading Information

- Information flows *up*, not *down*
  - “Reads up” disallowed, “reads down” allowed
- Simple Security Condition (Step 1)
  - Subject s can read object o iff, $L(o) \leq L(s)$ 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 (Step 1)
  - Subject s can write object o iff $L(s) \leq L(o)$ 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

Basic Security Theorem, Step 1

- If a system is initially in a secure state, and every transition of the system satisfies the simple security condition (step 1), and the *-property (step 1), then every state of the system is secure
  - Proof: induct on the number of transitions

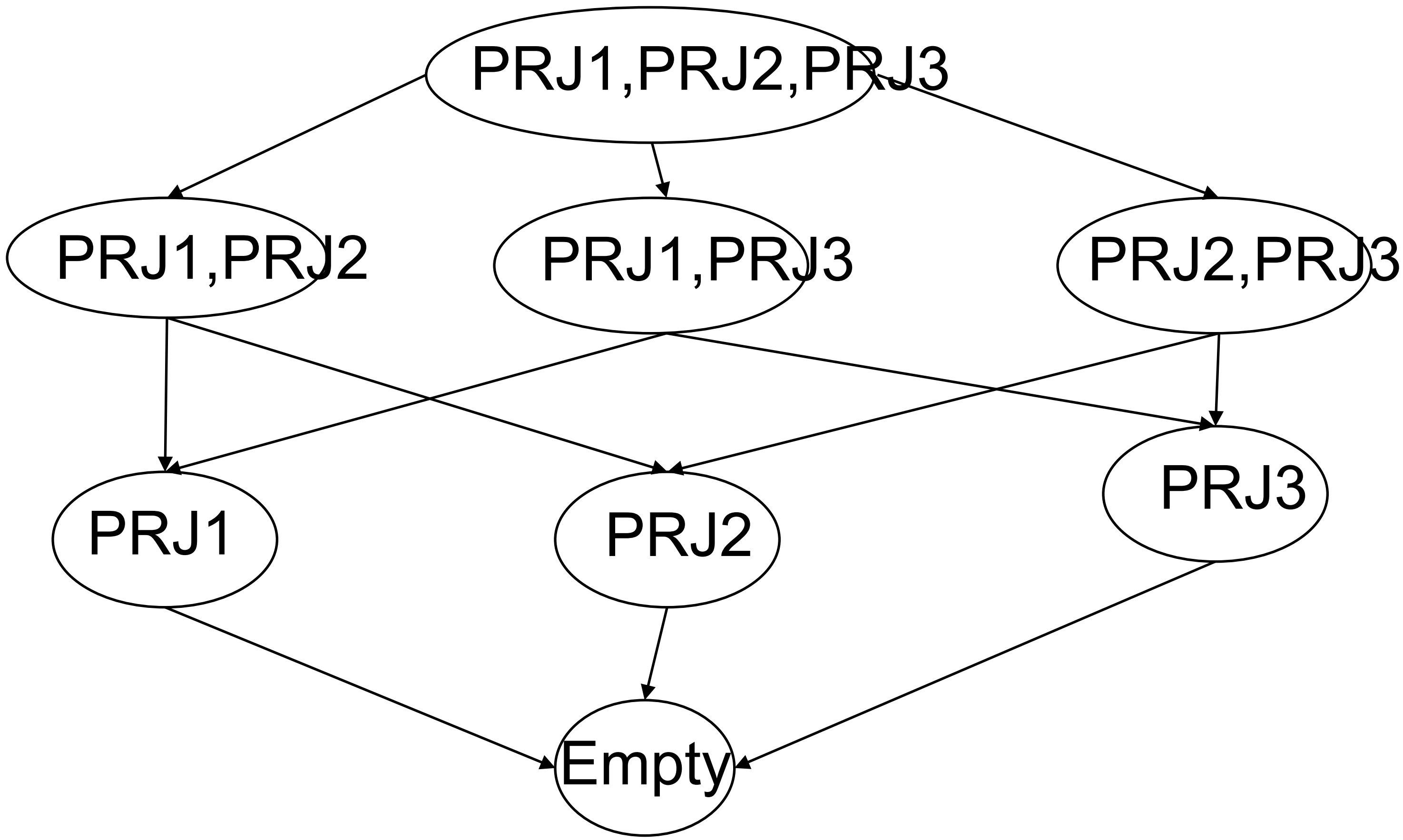
Bell-LaPadula Model, Step 2

- Expand notion of security level to include categories (also called compartments)
- Security level is (*clearance*, *category set*)
- Examples
  - (Top Secret, { PRJA, PRJB, PRJC })
  - (Confidential, { PRJB, PRJC })
  - (Secret, { PRJA, PRJB })

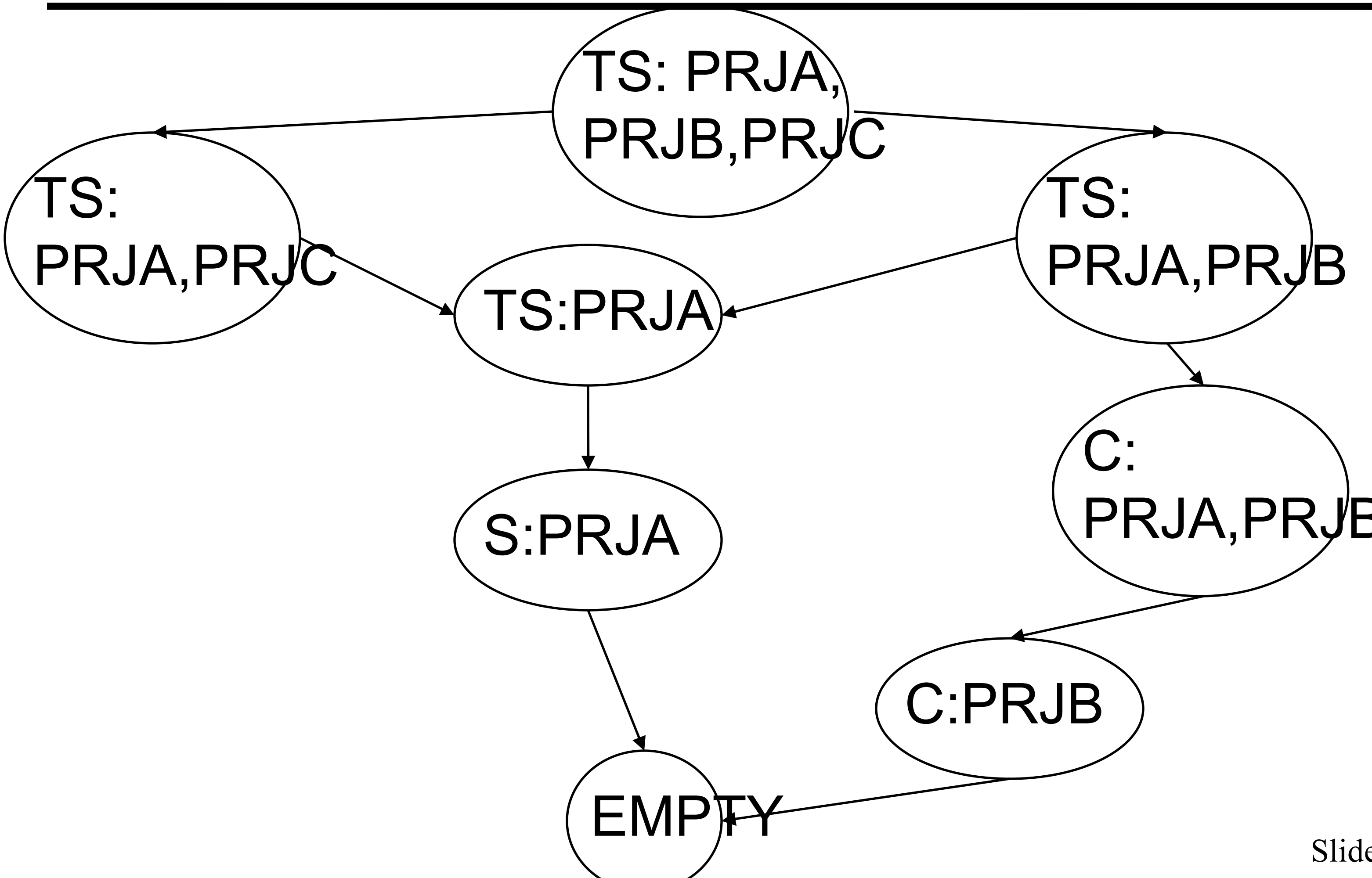
Levels and Lattices

- (A, C) *dom* (A', C') iff $A' \le A$ and $C' \subseteq C$
- Examples
  - (Top Secret, {PRJA, PRJC}) *dom* (Secret, {PRJA})
  - (Secret, {PRJA, PRJB}) *dom* (Confidential,{PRJA, PRJB})
  - (Top Secret, {PRJA}) ¬*dom* (Confidential, {PRJB})
  - (Secret, {PRJA}) ¬*dom* (Confidential,{PRJA, PRJB})
- Let C be set of classifications, K set of categories. Set of security levels $L = C \times K$, *dom* form lattice
  - *Partially ordered set*
  - *Any pair of elements*
    - *Have a greatest lower bound*
    - *Have a least upper bound*

Example Lattice



Subset Lattice



Total Order

- A total order (or "totally ordered set“) is a set plus a relation on the set that satisfies the following properties (e.g. ≤ on the set of integer values N)
- Reflexivity: a ≤ a
- Anti-symmetry: a ≤ b and b ≤ a implies a=b
- Transitivity: a ≤ b and b ≤ c implies a ≤ c
- Comparability: For any a and b in N, either a ≤b or b ≤a

Partial Order

- Partial order has all the properties of Total Order, except for Comparability
- Example: The relation ≤ on the set of complex numbers C. For example
  - Neither $1+4i \leq 2+3i$ nor $2+3i \leq 1+4i$

Lattice Definition

- A lattice is a combination of a set of elements S and a relation R meeting the following criteria:
  - R is reflexive, antisymmetric, and transitive on elements of S
  - For every s, t in S, there exists a greatest lower bound
  - For every s,t in S, there exists a least upper bound

Levels and Ordering

- Security levels plus *dom* form a partial order
  - Any pair of security levels may (or may not) be related by *dom*
- "dominates" serves the role of "greater than" in step 1
  - "greater than" is a total ordering, though

Reading Information

- Information flows *up*, not *down*
  - "Reads up" disallowed, "reads down" allowed
- Simple Security Condition (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

Basic Security Theorem, Step 2

- If a system is initially in a secure state, and every transition of the system satisfies the simple security condition (step 2), and the *-property (step 2), then every state of the system is secure
  - Proof: induct on the number of transitions
  - In actual Basic Security Theorem, discretionary access control treated as third property, and simple security property and *-property phrased to eliminate discretionary part of the definitions — but simpler to express the way done here.

Problem

- Colonel has (Secret, {PROJ1, PROJ2}) clearance
- Major has (Secret, {PROJ2}) clearance
- Can Major write data that Colonel can read?
- Can Major read data that Colonel wrote?

Solution

- Define maximum, current levels for subjects
  - *maxlevel*(*s*) *dom curlevel*(*s*)
- Example
  - Treat Major as an object (Colonel is writing to him/her)
  - Colonel has *maxlevel* (Secret, {PROJ1, PROJ2 })
  - Colonel sets *curlevel* to (Secret, { PROJ2 })
  - Now *L*(Major) *dom curlevel*(Colonel)
    - Colonel can write to Major without violating “no writes down”

Principle of Tranquility

- Subjects and objects may not change their security levels once instantiated
- Raising object's security level
  - Information once available to some subjects is no longer available
  - Usually assume information has already been accessed, so this does nothing
- Lowering object's security level
  - The *declassification problem*
  - Essentially, a "write down" violating *-property
  - Solution: define set of trusted subjects that *sanitize* or remove sensitive information before security level lowered

Types of Tranquility

- Strong Tranquility: The clearances of subjects, and the classifications of objects, do not change during the lifetime of the system
  - Resolves the mentioned problems
  - Inflexible and not practical
- Weak Tranquility: The clearances of subjects, and the classifications of objects change in accordance with a specified policy
  - Moderates the restriction to allow harmless changes of security levels
  - Flexible

Example

- DG/UX System
  - Only a trusted user (security administrator) can lower object's security level
  - In general, process MAC labels cannot change
    - If a user wants a new MAC label, needs to initiate new process
    - Cumbersome, so user can be designated as able to change process MAC label within a specified range
- Other systems allow multiple labeled windows to address users operating a multiple levels

Key Points

- Confidentiality models restrict flow of information
- Bell-LaPadula defines security it in terms of 3 properties
  - simple security condition
  - *-property
  - discretionary security property
- Theorems are assertions about these properties
- Cornerstone of much work in computer security