THE UNIVERSITY OF BRITISH COLUMBIA

Security Policies

EECE 412

Session 11

Copyright © 2004 Konstantin Beznosov

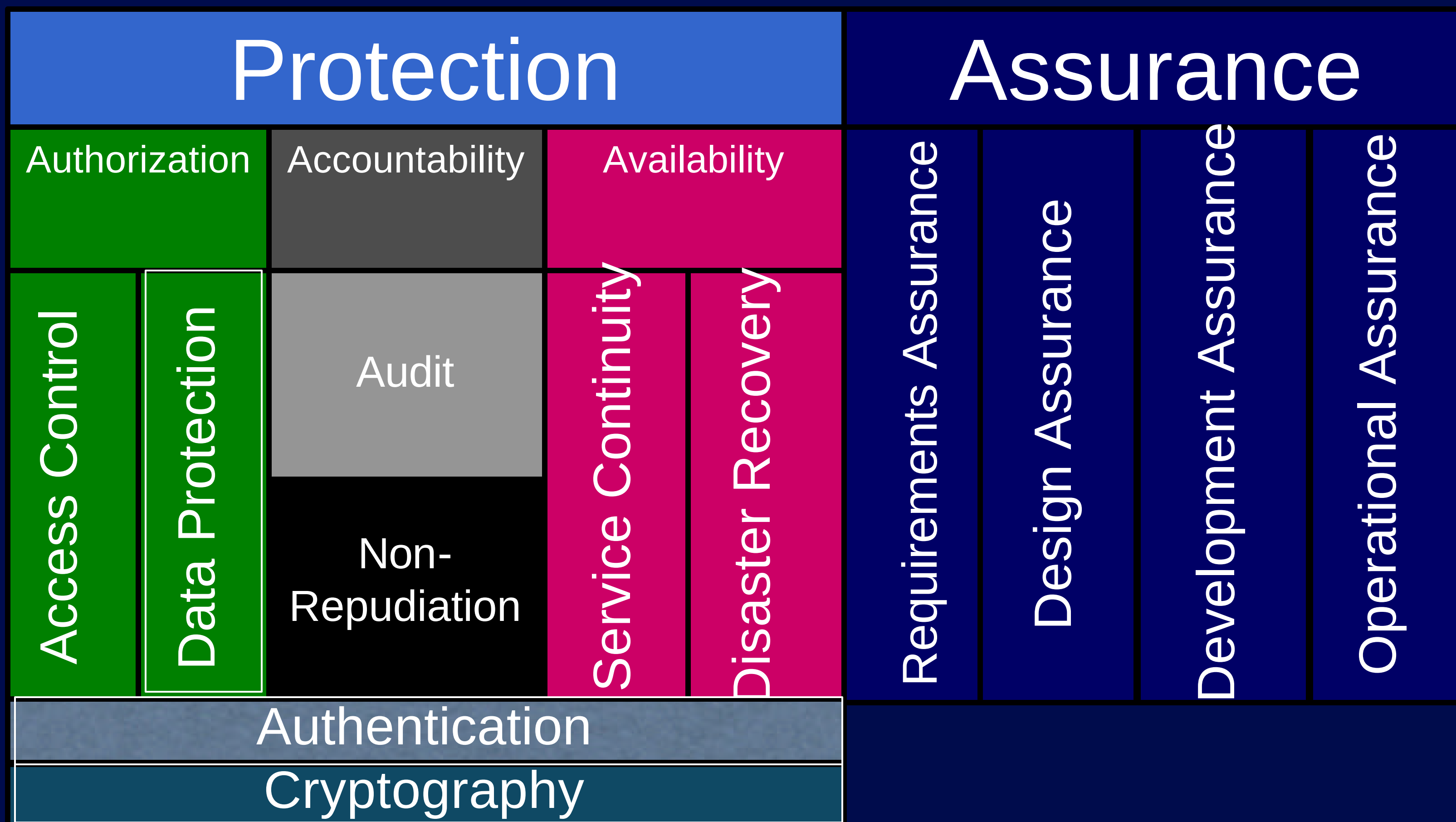
Last Topic Recap

- Content
  - Authentication system definition
  - Password-based authentication
  - Challenge-response authentication
    - S/Key one-time password system
  - Biometric authentication
  - Multi-factor authentication
  - Ways to break and improve authentication systems
- Key points
  - Authentication is not just about cryptography
    - You have to consider system components
  - Passwords are here to stay
    - They provide a basis for most forms of authentication
  - Two or three -factor authentication is the best yet more expensive

Outline

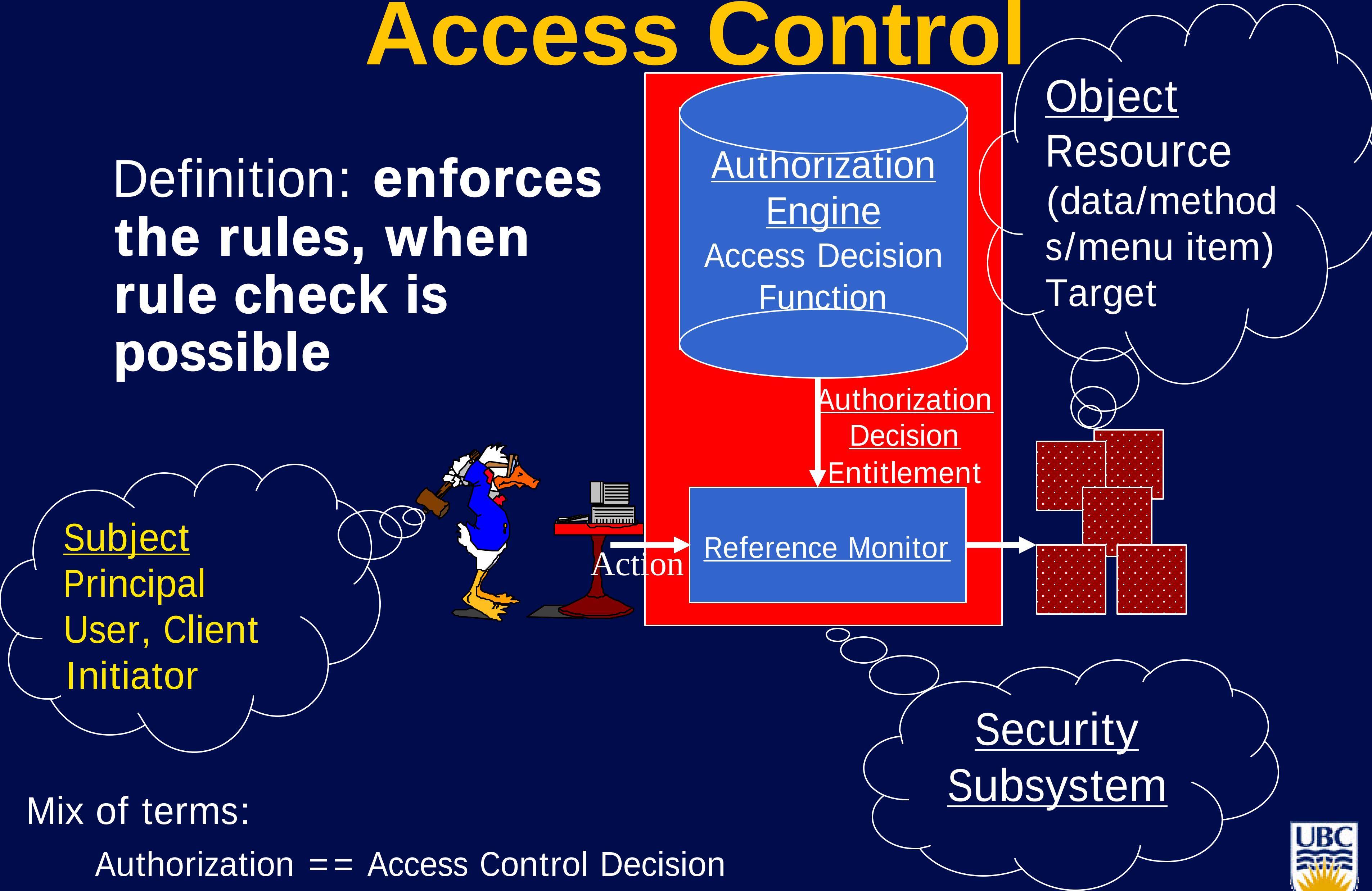
- Access control mechanisms
- Access Matrix
- Security policies
  - Confidentiality policies
    - Bell LaPadula confidentiality model

Where We Are


Protection
Assurance
Authorization
Accountability
Availability
Access Control
Data Protection
Audit
Non-Repudiation
Service Continuity
Disaster Recovery
Requirements Assurance
Design Assurance
Development Assurance
Operational Assurance
Authentication
Cryptography

Authorization Mechanisms: Access Control

Definition: **enforces the rules, when rule check is possible**



Mix of terms:

- Authorization == Access Control Decision
- Authorization Engine == Policy Engine

Access Matrix

Copyright © 2004 Konstantin Beznosov

Object System

Subjects

Objects

Access Matrix

A

	Subject 1	Subject 2	Subject 3	File 1	File 2	Process 1
Subject 1	*owner control	*owner control	*call	*owner *read *write		
Subject 2			call	*read	write	wakeup
Subject 3			owner control	read	*owner	

- Subjects are objects
- Objects are not subjects

Access Matrix Structure

objects (entities)

subjects

	o_1	…	o_m	s_1	…	s_n
s_1						
s_2						
…						
s_n						

- Subjects $S = \{ s_1,\ldots,s_n \}$
- Objects $O = \{ o_1,\ldots,o_m \}$
- Rights $R = \{ r_1,\ldots,r_k \}$
- Entries $A[s_i, o_j] \subseteq R$
- $A[s_i, o_j] = \{ r_x, \ldots, r_y \}$ means subject s_i has rights $r_x, \ldots, r_y$ over object o_j

Example

- Processes *p*, *q*
- Files *f*, *g*
- Rights *r*, *w*, *x*, *a*, *o*

	f	*g*	*p*	*q*
p	*rwo*	*r*	*rwxo*	*w*
q	*a*	*ro*	*r*	*rwxo*

Matrix Implementation Techniques

1. $T = \{<d,x,A_{d,x}>\}$ – impractical
   a) Only relevant parts of A need to be handy
   b) Could be very inefficient for some As (e.g. public files)
   c) List of objects to which d has access
2. Capability = $<x,A_{d,x}>$
   - C-lists
   - Attach C-list to domains
   - Addresses (a), (c) and potentially (b)
3. attach the protection information to the object: $A_x(d)$
   - Access key – capability used for identification, (credential)
   - {<access key, {access attributes}>} – access control list (ACL)

Group Work

ACLs are good for revoking individual's access to a particular file.

- How hard is it to revoke a user's access to a particular set of files, but not to all files, with ACLs?
- Compare and contrast this with the problem of revocation using capabilities.

Access Matrix Summary

- Object System
  - Subjects, objects, access matrix
  - Objects are shared
  - All subjects are objects
    - not all objects are subjects
  - Matrix modification rules
- Matrix implementation
  - Capability lists
  - Access control lists

Security Policies

Copyright © 2004 Konstantin Beznosov

What's Security Policy?

- Policy partitions system states into:
  - Authorized (secure)
    - These are states the system can enter
  - Unauthorized (nonsecure)
    - If the system enters any of these states, it's a security violation
- Secure system
  - Starts in authorized state
  - Never enters unauthorized state
- Authorized state in respect to what?

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

- Example:
  - *X* set of students
  - *I* final exam answer key
  - *I* is confidential with respect to *X* if students cannot obtain final exam answer key

What’s Integrity?

- *X* set of entities, *I* information
- *I* has *integrity* property with respect to *X* if all $x \in X$ trust information in *I*
- *Examples?*

Types of Access Control

- Discretionary Access Control (DAC, IBAC)
  - individual user sets access control mechanism to allow or deny access to an object
- Mandatory Access Control (MAC)
  - system mechanism controls access to object, and individual cannot alter that access
- Originator Controlled Access Control (ORCON)
  - originator (creator) of information controls who can access information

Question

- Policy disallows cheating
  - Includes copying homework, with or without permission
- A class has students do homework on computer
- Alice forgets to read-protect her homework file
- Bob copies it
- Who cheated?
  - Alice, Bob, or both?

Answer

- Bob cheated
  - Policy forbids copying homework assignment
  - Bob did it
  - System entered unauthorized state (Bob having a copy of Anne's assignment)
- If not explicit in computer security policy, certainly implicit
  - Not credible that a unit of the university allows something that the university as a whole forbids, unless the unit explicitly says so

Answer Part #2

- Alice didn't protect her homework
  - Not required by security policy
- She didn't breach security
- If policy said students had to read-protect homework files, then Alice did breach security
  - She didn't do this

Key Points about Policies and Mechanisms

- Policies describe what is allowed
- Mechanisms control how policies are enforced

Confidentiality Policies

Copyright © 2004 Konstantin Beznosov

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, $L(o) \geq L(s)$ 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

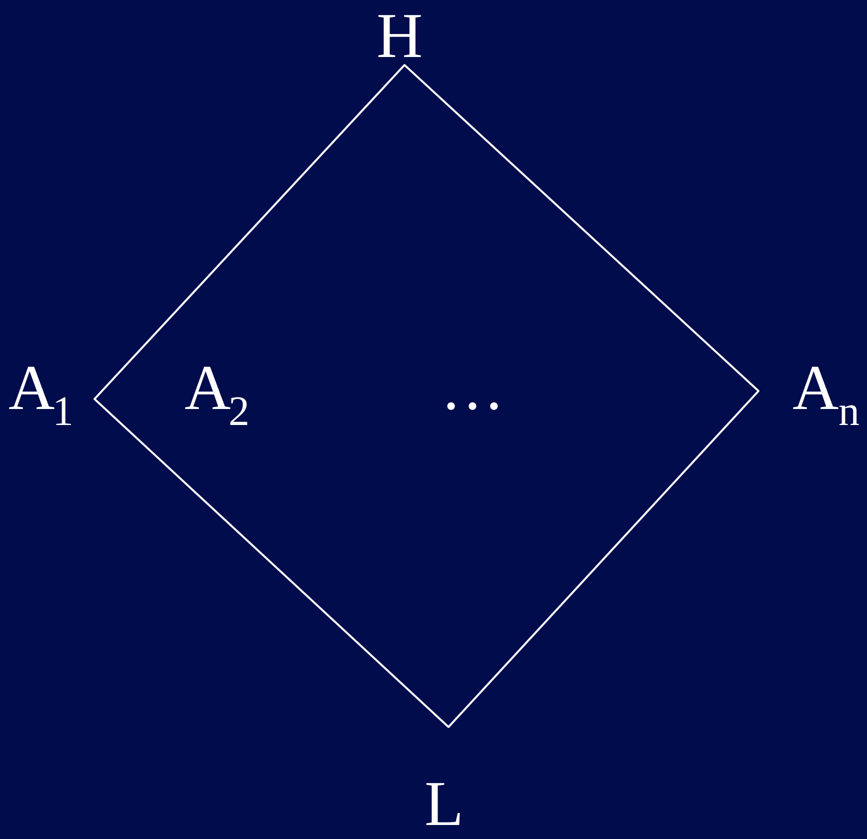
Bell-LaPadula Model, Step 2

- 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 })

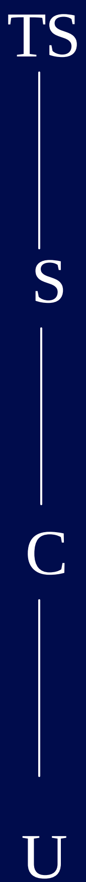
Levels and Lattices

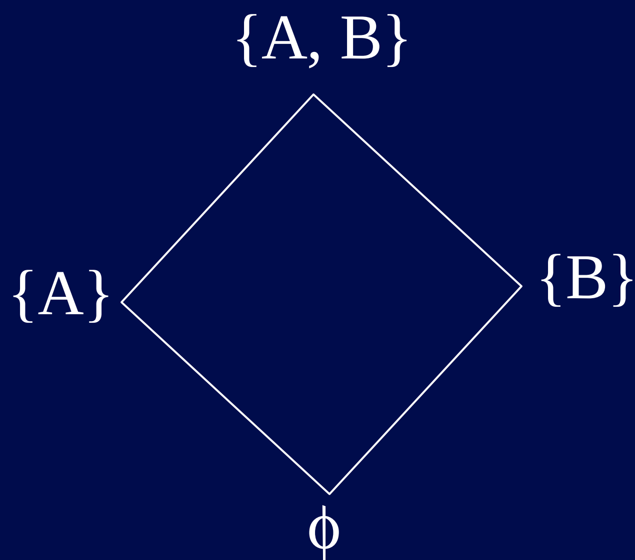
- (*A*, *C*) *dominates* (*A*¢, *C*¢) iff $A' = A$ and *C*¢ $\subseteq$ *C*
- Examples
  - (Top Secret, {NUC, ASI}) *dom* (Secret, {NUC})
  - (Secret, {NUC, EUR}) *dom* (Confidential,{NUC, EUR})
  - (Top Secret, {NUC}) $\neg$*dom* (Confidential, {EUR})
- Let *C* be set of classifications, *K* set of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

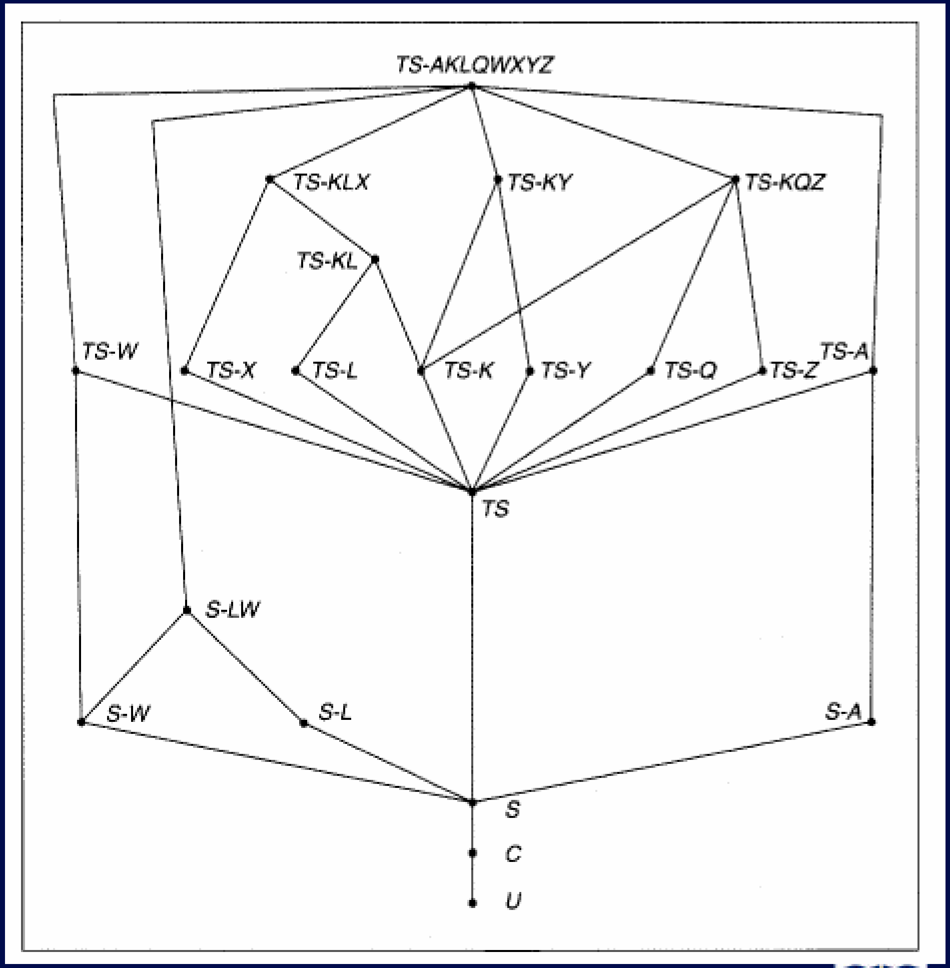
Bounded Isolated Classes



The Military Lattice







Levels and Ordering

- Security levels partially ordered
  - Any pair of security levels may (or may not) be related by *dom* 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

Problem

- Colonel has (Secret, {NUC, EUR}) clearance
- Major has (Secret, {EUR}) clearance
- Major can talk to colonel (“write up” or “read down”)
- Colonel cannot talk to major (“read up” or “write down”)
- Clearly absurd!

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, { NUC, EUR })
  - Colonel sets *curlevel* to (Secret, { EUR })
  - Now *L*(Major) *dom* *curlevel*(Colonel)
    - Colonel can write to Major without violating "no writes down"

Key Points Regarding Confidentiality Policies

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

Next Session Preview

- Integrity policies
  - Biba integrity model
  - Clark-Wilson integrity model
- Hybrid policies
  - Chinese Wall model
  - Role-based access control model