CS 5430

Mandatory Access Control

Prof. Clarkson

Spring 2017

Review: Access control

- Subject: entity to which execution can be attributed
- Object: data or resource
- Operation: performed by subject on object
- Right: entitlement to perform operation

Review: DAC

- Discretionary access control (DAC)
  - **Philosophy:** users have the *discretion* to specify policy themselves
  - Commonly, information belongs to the **owner** of object
  - Model: access control **relation**
    - Set of triples (subj,obj,rights)
    - Sometimes described as access control "matrix"
- Implementations:
  - Access control lists (ACLs): each object associated with list of (subject, rights)
  - Privilege lists: each subject associated with list of (object, rights)
  - Capabilities: distributed ways of implementing privilege lists

MAC

- Mandatory access control (MAC)
  - not Message Authentication Code (applied crypto), nor Media Access Control (networking)
  - **philosophy:** central authority *mandates* policy
  - information belongs to the authority, not to the individual users
- Five case studies:
  1. Multi-level security (military)
  2. Brewer-Nash (consulting firm)
  3. Clark-Wilson (business)
  4. Role-based access control (organization)
  5. Clinical information systems (medicine)

1. MULTI-LEVEL SECURITY

Sensitivity

- Concern is **confidentiality** of information
- Documents classified according to sensitivity: risk associated with release of information
- In US:
  - Top Secret
  - Secret
  - Confidential
  - Unclassified

Compartments

- Documents classified according to compartment(s): categories of information (in fact, aka category)
  - cryptography
  - nuclear
  - biological
  - reconnaissance
- **Need to Know Principle:** access should be granted only when necessary to perform assigned duties (instance of Least Privilege)
  - {crypto,nuclear}: must need to know about **both** to access
  - {}: no particular compartments

Labels

- Label: pair of sensitivity level and set of compartments, e.g.,
  - (Top Secret, {crypto, nuclear})
  - (Unclassified, {})
- Users are labeled according to their clearance
- Document is labeled aka classified
  - Perhaps each paragraph labeled
  - Label of document is most restrictive label for any paragraph
- Labels are imposed by organization
- **Notation:** let L(X) be the label of entity X

Restrictiveness of labels

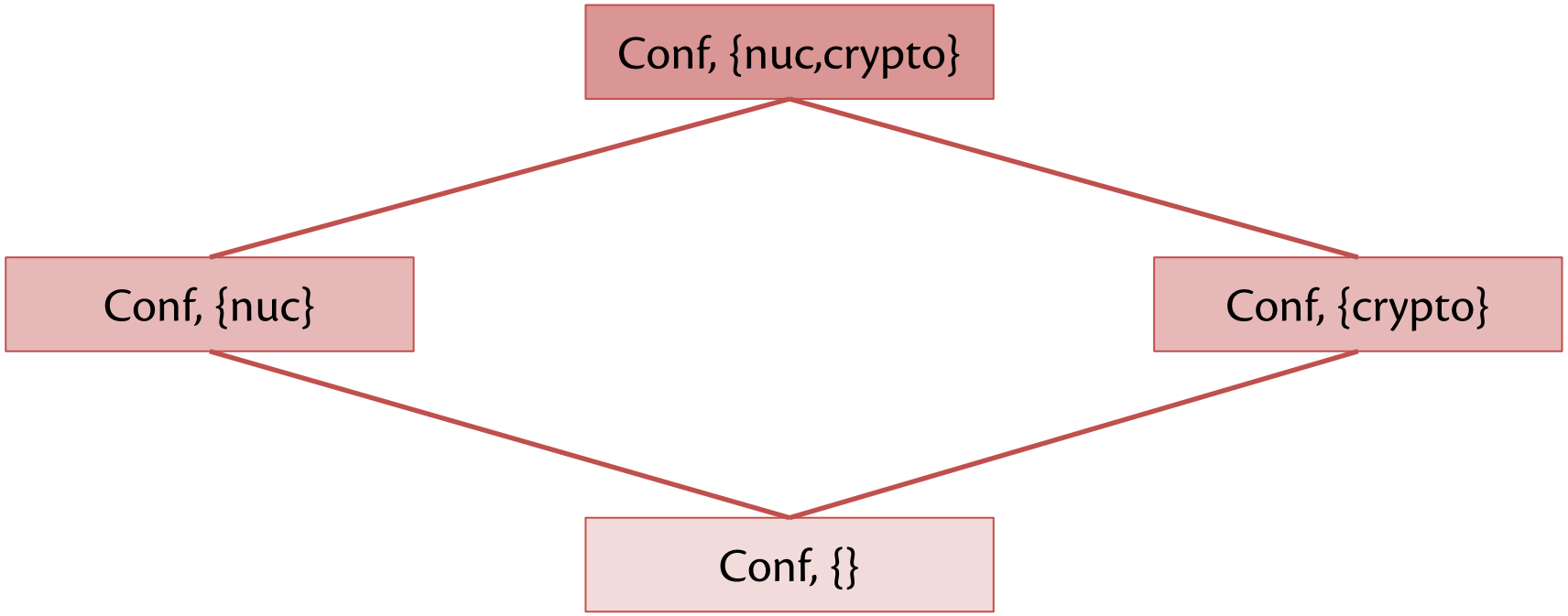
Notation: L1 $\sqsubseteq$ L2

- means L1 is no more restrictive than L2
  - less precisely: L1 is less restrictive than L2
  - another reading: information may flow from L1 to L2
  - also: L1 is dominated by L2
- e.g.
  - (Unclassified,{}) $\sqsubseteq$ (Top Secret, {})
  - (Top Secret, {crypto}) $\sqsubseteq$ (Top Secret, {crypto,nuclear})

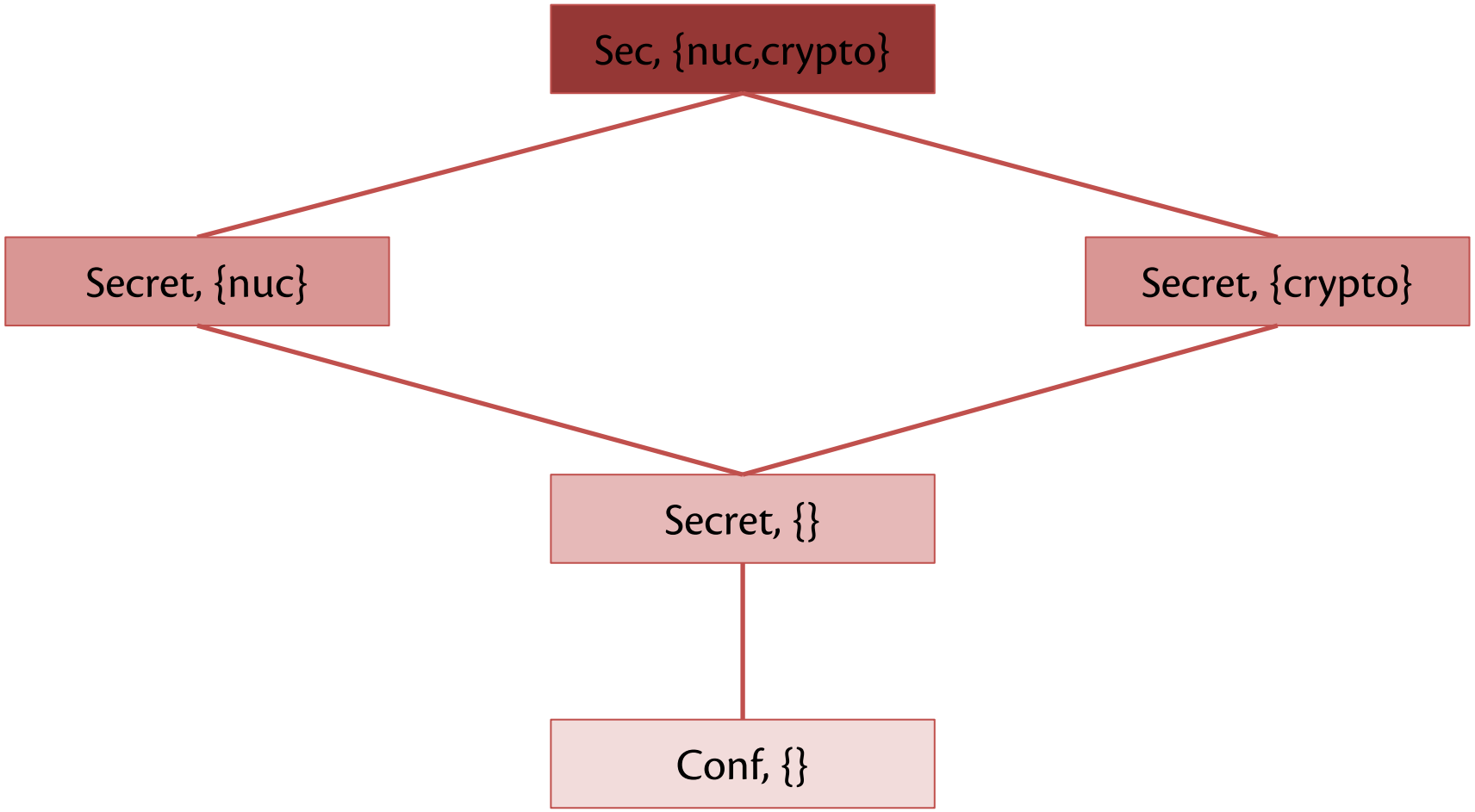
Restrictiveness of labels

- **Definition:**
  - Let L1 = (S1, C1) and L2 = (S2, C2)
  - L1 $\sqsubseteq$ L2 iff S1 $\leq$ S2 and C1 $\subseteq$ C2
  - Where $\leq$ is order on sensitivity:
    Unclassified $\leq$ Confidential $\leq$ Secret $\leq$ Top Secret
- Partial order:
  - Some labels are incomparable
  - e.g. (Secret, {crypto}) vs. (Top Secret, {nuclear})

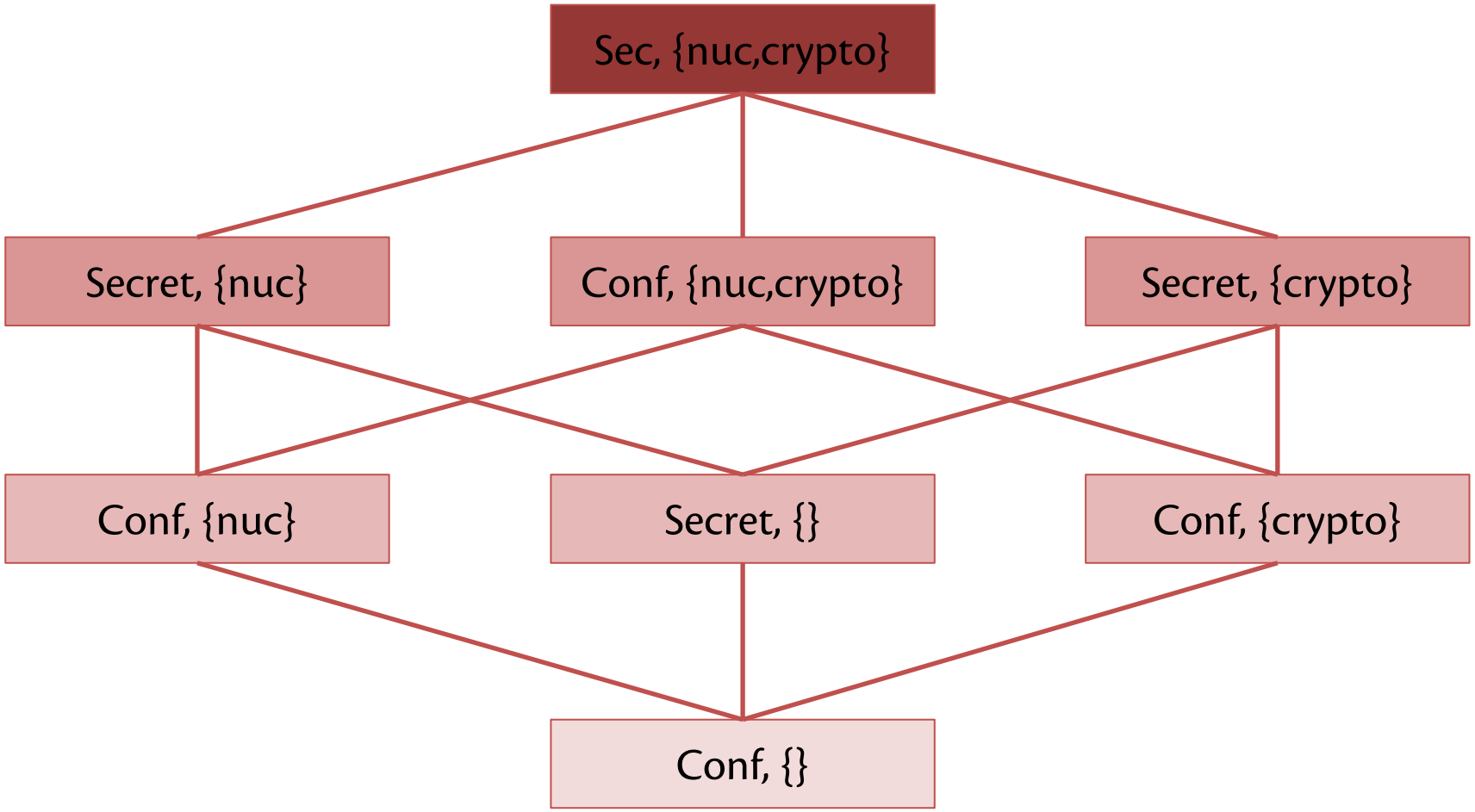
Label partial order



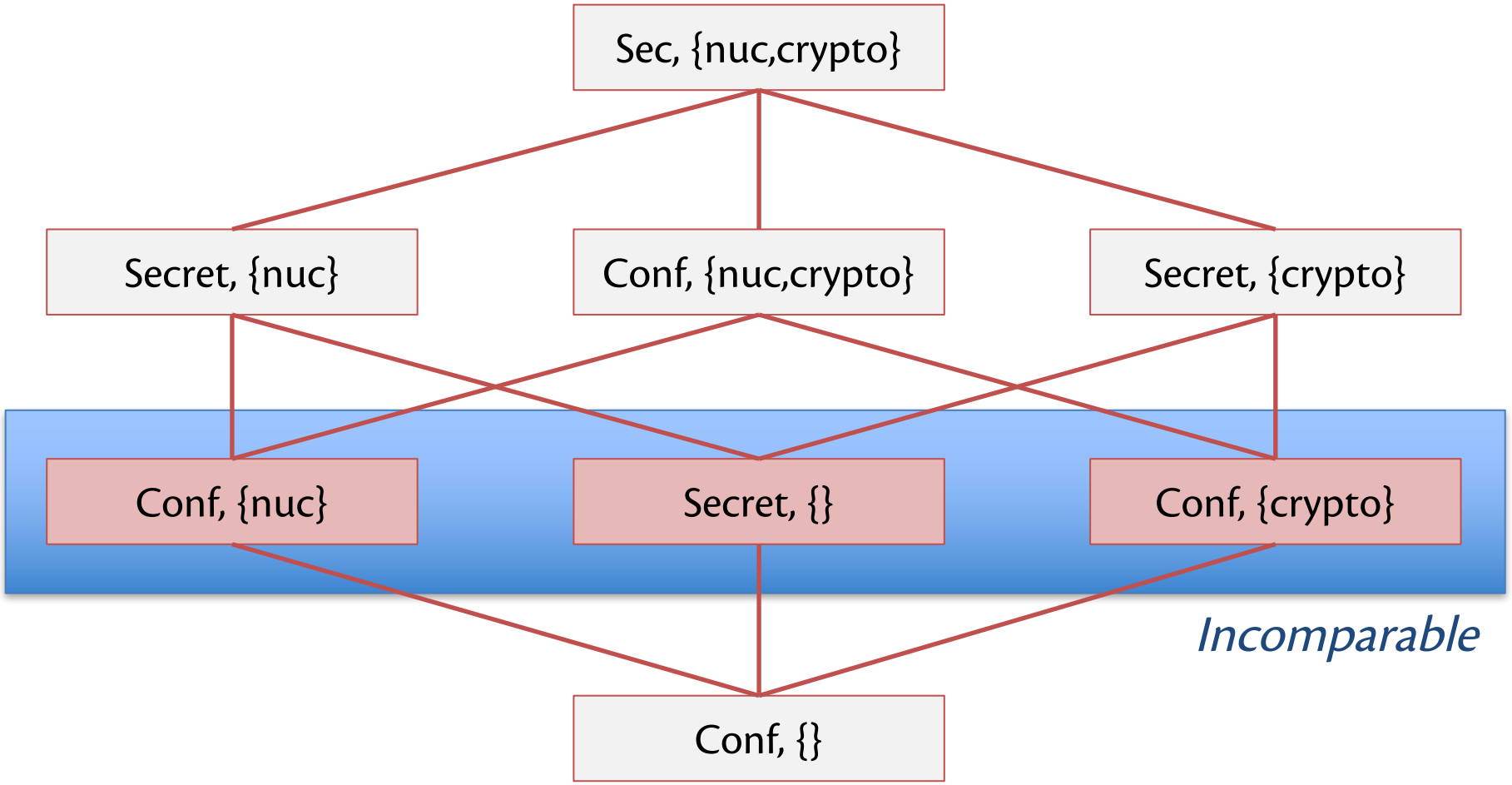
Label partial order



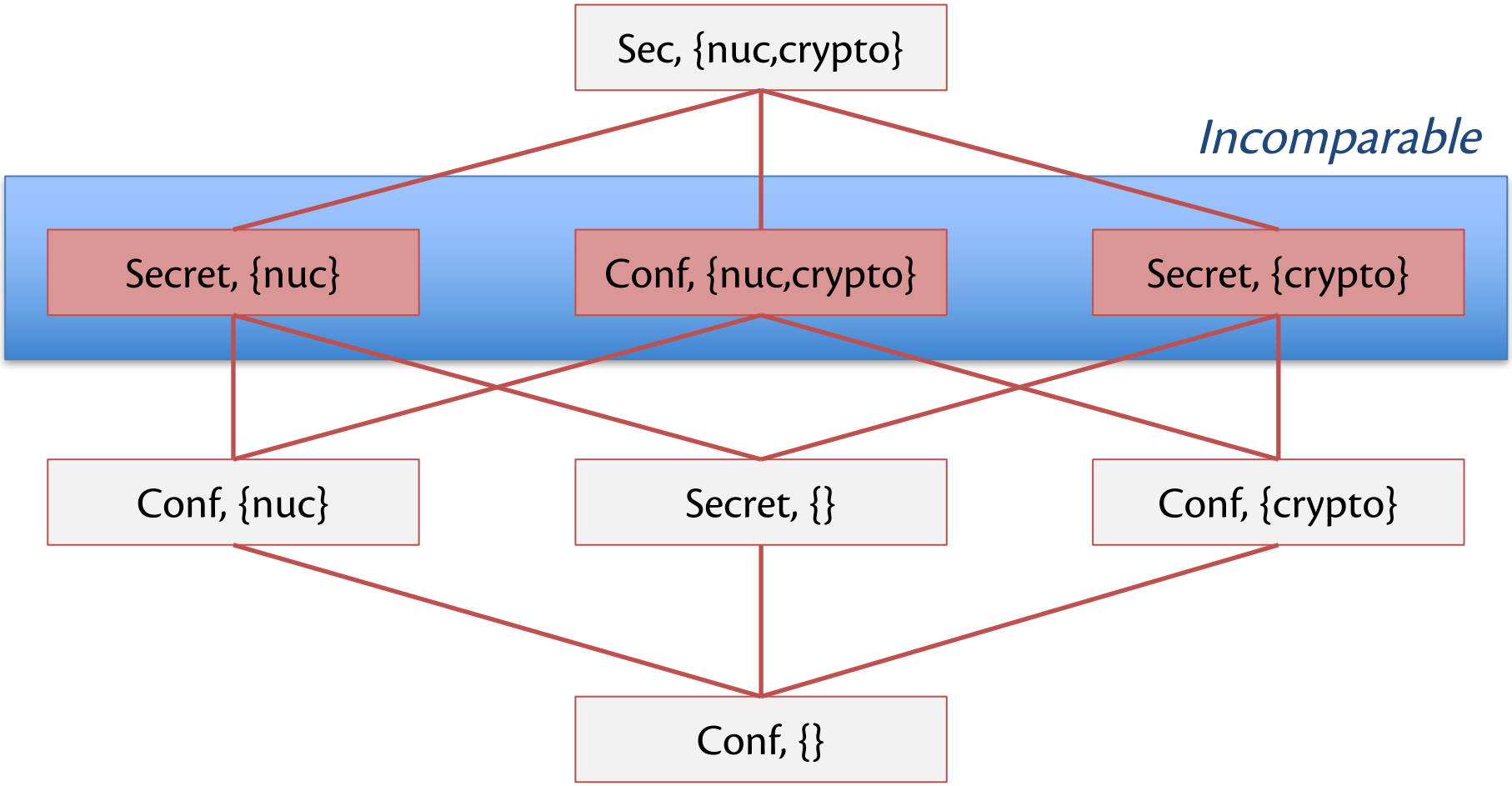
Label partial order



Label partial order



Label partial order



Access control with MLS

- When may a subject read an object?
  - **Threat:** subject attempts to read information for which it is not cleared
  - e.g., subject with clearance Unclassified attempts to read Top Secret information
- When may a subject write an object?
  - **Threat:** subject attempts to *launder* information by writing into a lower-security object
  - e.g., subject with clearance Top Secret reads Top Secret information then writes it into an Unclassified file

Access control with MLS

Threat of concern is **subject** not **user**:

- Users trustworthy by virtue of vetting process for security clearance
- Out of scope (e.g.): user who views Top Secret information and calls the *Washington Post*
- But still want to enforce Least Privilege
- And malicious programs are a threat...

Trojan Horse

Access control with MLS

- When may a subject read an object?
  - **S may read O iff L(O) ⊑ L(S)**
  - object's classification must be below (or equal to) subject's clearance
  - "no read up"
- When may a subject write an object?
  - **S may write O iff L(S) ⊑ L(O)**
  - object's classification must be above (or equal to) subject's clearance
  - "no write down"
- Beautiful **symmetry** between these

Reading with MLS

- Scenario:
  - Colonel with clearance (Secret, {nuclear, Europe})
  - DocA with classification (Confidential, {nuclear})
  - DocB with classification (Secret, {Europe, US})
  - DocC with classification (Top Secret, {nuclear, Europe})
- Which documents may Colonel **read**?
  - Recall: S may read O iff L(O) $\sqsubseteq$ L(S)
  - DocA: (Confidential, {nuclear}) $\sqsubseteq$ (Secret, {nuclear, Europe})
  - DocB: (Secret, {Europe, US}) $\not\sqsubseteq$ (Secret, {nuclear, Europe})
  - DocC: (Top Secret, {nuclear, Europe}) $\not\sqsubseteq$ (Secret, {nuclear, Europe})

Writing with MLS

- Scenario:
  - Colonel with clearance (Secret, {nuclear, Europe})
  - DocA with classification (Confidential, {nuclear})
  - DocB with classification (Secret, {Europe, US})
  - DocC with classification (Top Secret, {nuclear, Europe})
- Which documents may Colonel **write**?
  - Recall: S may write O iff L(S) $\sqsubseteq$ L(O)
  - DocA: (Secret, {nuclear, Europe}) $\not\sqsubseteq$ (Confidential, {nuclear})
  - DocB: (Secret, {nuclear, Europe}) $\not\sqsubseteq$ (Secret, {Europe, US})
  - DocC: (Secret, {nuclear, Europe}) $\sqsubseteq$ (Top Secret, {nuclear, Europe})

Reading and writing with MLS

- Scenario:
  - Colonel with clearance (Secret, {nuclear, Europe})
  - DocA with classification (Confidential, {nuclear})
  - DocB with classification (Secret, {Europe, US})
  - DocC with classification (Top Secret, {nuclear, Europe})
- Summary:
  - DocA: Colonel may read but not write
  - DocB: Colonel may neither read nor write
  - DocC: Colonel may write but not read

Perplexities of writing with MLS

1. Blind write: subject may not read higher-security object yet may write it
   - Useful for logging
   - Some implementations prohibit writing up as well as writing down
2. **User** who wants to write lower-security object may not
   - **Attenuation of privilege:** login at a lower security level than clearance
   - Motivated by Trojan Horse
   - Nice (annoying?) application of Least Privilege
3. Declassification violates "no write down"
   - Encryption or billing procedure produces (e.g.) Unclassified output from Secret information
   - Traditional solution is trusted subjects who are not constrained by access control rules

Prevention of laundering

- Earlier concern: "subject with clearance Top Secret reads Top Secret information then writes it into an Unclassified file"
- More generally:
  - S reads O1 then writes O2
  - where L(O2) $\sqsubset$ L(O1)
  - and regardless of L(S)
- **Prohibited by MLS rules:**
  - S read O1, so L(O1) $\sqsubseteq$ L(S)
  - S wrote O2, so L(S) $\sqsubseteq$ L(O2)
  - So L(O1) $\sqsubseteq$ L(S) $\sqsubseteq$ L(O2)
  - Hence L(O1) $\sqsubseteq$ L(O2)
  - But combined with L(O2) $\sqsubset$ L(O1), we have L(O1) $\sqsubset$ L(O1)
  - Contradiction!
- So access control rules would defeat laundering, Trojan Horse, etc.

BLP

[Bell and LaPadula 1973]

- Formal mathematical model of MLS plus access control matrix
- Proof that information cannot leak to subjects not cleared for it
- "No read up": simple security property
- "No write down": *-property
- *"The influence of [BLP] permeates all policy modeling in computer security"* –Matt Bishop
  - Influenced Orange Book
  - Led to research field "foundations of computer security"

BLP, for integrity

- BLP is about confidentiality
- Adapted to integrity by Biba [1977]: same rules, different lattice
  - Instead of Unclassified and Secret, labels could be Untrusted and Trusted
- Recall L1 $\sqsubseteq$ L2 means "L1 may flow to L2"
  - BLP: low secrecy sources may flow to high secrecy sinks
    - Hence Unclassified $\sqsubseteq$ Secret, but not v.v.
  - Biba: low integrity sources may not flow to high integrity sinks
    - Hence Trusted $\sqsubseteq$ Untrusted, but not v.v.
  - High vs. low is "flipped" (lattices are *duals*)

Biba model

- **S may read O iff L(O) ⊑ L(S)**
  - E.g., Trusted subject cannot read Untrusted object
  - But Untrusted subject may read Trusted object
- **S may write O iff L(S) ⊑ L(O)**
  - E.g., Trusted subject may write Untrusted object
  - But Untrusted subject may not write Trusted object

MLS/BLP in OSs

DG/UX [1985]

- Three regions:
  Virus Protection $\sqsubseteq$ User Region $\sqsubseteq$ Administrative Region
- Writing up is prohibited (no blind writes)
- User Region: users are cleared here
- Virus Protection: executables that implement the system
  - Can't be written by users (no write down)
  - Can be executed (okay to read down)
- Administrative Region: authorization & authentication database (assigns labels), audit logs
  - Can't be read by users (no read up)
  - Can't be changed by users (no blind writes)

MLS/BLP in OSs

- SELinux [open source release by NSA 2000]
- TrustedBSD [2000], influences iOS and OS X

2. BREWER-NASH

Conflict of interest

Setting: consulting firm

- e.g., stock exchange, investment bank, law firm
- Consultant represents two clients
  - Best interest of those clients conflict
  - Consultant could help one at expense of the other
  - Consultant has a conflict of interest (COI)
- Norms (laws, regulations, ethics) prohibit consultant from exploiting COI
- After some time (days, years, never), COI might expire

Conflict of interest

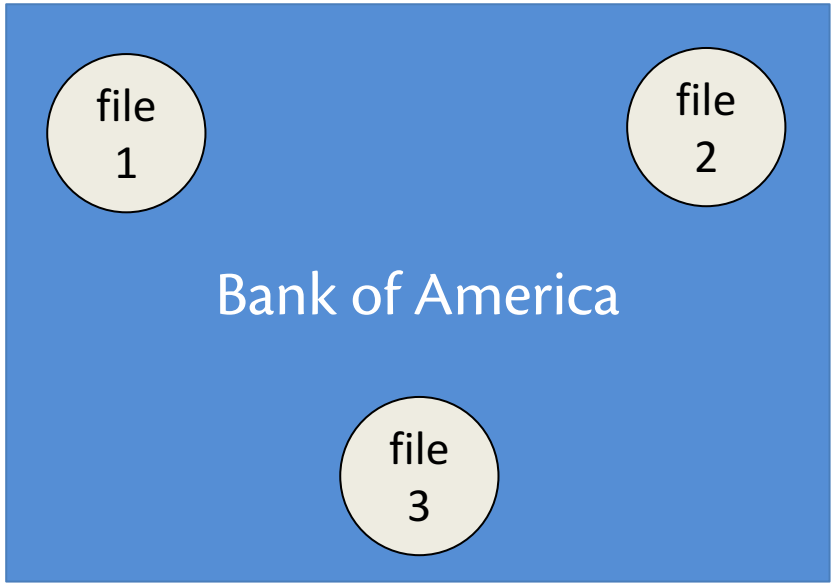
- Typical paper implementation:
  - Consultant maintains public CV
    - Entry in CV for each client
    - Entry has been sanitized and approved by client, e.g., "Sep 2015-Apr 2016: consulted on security requirements for a new branch accounting system for a major US retail bank"
  - Manager checks CV before assigning consultant to new client
  - Client receives CV to double-check from their perspective
- Brewer and Nash [1989] invented a MAC policy for this setting
  - Often known as Chinese Wall (CW)
  - Other names: ethics wall, screen

Great Wall of China

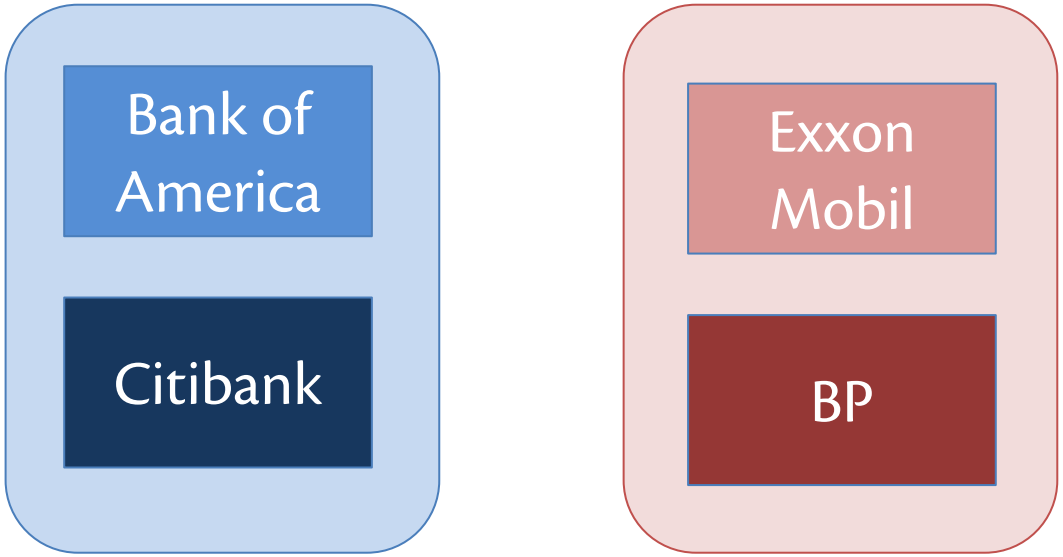
Brewer-Nash model

- Object: contains sensitive information about companies
  - a file about Bank of America's trade secrets
  - but not its addresses, phone numbers, etc.
- Company dataset (CD): all the objects related to a single company
  - all the files about Bank of America
- Conflict of interest class (COI): all the company datasets for which the companies compete
  - all the files about banks

Brewer-Nash model



Brewer-Nash model



CD

COI

Breaches

Prevent two kinds of breaches of the wall:

- One consultant works on more than one CD inside a COI
- Two consultants each work on their own CD inside COI but cooperate to write that information to a shared object

Access control with Brewer-Nash

- When may a subject read an object?
  - **S may read O iff**
    S has never read any O' such that
    COI(O) = COI(O') and CD(O) != CD(O')
  - Subject may not read from two CDs inside same COI
  - Requires tracking history of objects read by subject
- When may a subject write an object?
  - **S may write O iff**
    S has never read any O' such that
    CD(O) != CD(O')
  - Subject may not write to any other CD after reading from one

Reading with Brewer-Nash

- **S may read O iff**
  S has never read any O' such that
  COI(O) = COI(O') and CD(O) != CD(O')
- If S has never read anything, S has free choice of what to read next
- Once S does read object from CD1 in COI1, a wall is erected around S
  - Cannot read other CDs from same COI
  - But can read from different COI
- If S does read from CD2 in COI2, wall changes shape
  - CD1 and CD2 inside wall
  - All other CDs from COI1 and COI2 outside the wall

Writing with Brewer-Nash

- **S may write O iff**
  S has never read any O' such that
  CD(O) != CD(O')
- If S has never read anything, S has free choice of what to write
- If S has read from CD1, S may write only to CD1
- If S has read from CD1 and CD2, S may not write at all
  - e.g. read from Bank of America and Exxon Mobil:
    - Now cannot write anywhere
    - Writing to Bank of America could leak info about Exxon Mobil and vv.
- Seems overly prohibitive...

Users with Brewer-Nash

- A **subject** who has read two CDs may not write
- But that need not be true of a **user**
- Track read objects for:
  - user over its lifetime
  - subject over its lifetime (which is shorter than user)
  - distinguish what user has learned vs. what subject has learned
- As with MLS, user can choose to login at lower security level
  - **Attenuation of privilege:** give up the subject's right to read from CDs that have previously been read by user
  - Subject assigned that security level
  - So user could have multiple subjects with different security levels

Users with Brewer-Nash

Example: Jane has read CD1 from COI1 and nothing from COI2

- Jane could login
  - with right to read CD1
  - or without that right
- Then subject on behalf of Jane reads CD2 from COI2: that is recorded for Jane as well and influences future subjects of hers
- Can Jane's subject write?
  - With right to read CD1: no
  - Without right: yes
- Jane's subject always prohibited from reading CD1' from COI1, regardless of whether right to read CD1 is enabled

So if user wants to work with different CDs, they can! Just disable access to the rest.

Upcoming events

- [Wed] A5 due

Do not trust the horse, Trojans!
Whatever it is, I fear the Greeks, even bringing gifts.
– Virgil, Aeneid*, Book II*