IS 2150 / TEL 2810
Introduction to Security

James Joshi
Assistant Professor, SIS

Lecture 6
October 4, 2007

Integrity Models

Role based
Access Control

Objective

- Define/Understand various Integrity models
  - Biba’s
  - Clark-Wilson
- Define/Understand
  - Chinese Wall Model
  - Role-based Access Control model
- Overview the secure interoperation issue

Biba's Integrity Policy Model

- Based on Bell-LaPadula
  - Subject, Objects have
    - Integrity Levels with dominance relation
  - Higher levels
    - more reliable/trustworthy
    - More accurate

Biba’s model

- Strict Integrity Policy (dual of Bell-LaPadula)
  - s can **read** $o \leftrightarrow i(s) \leq i(o)$ (no read-down)
    - Why?
  - s can **write** $o \leftrightarrow i(o) \leq i(s)$ (no write-up)
    - Why?
  - s_1 can **execute** $s_2 \leftrightarrow i(s_2) \leq i(s_1)$
    - Why?

Low-water-mark

- Low-Water-Mark Policy
  - *s* can **write** $o \leftrightarrow i(o) \le i(s)$
    - Why?
  - *s* **reads** $o \rightarrow i'(s) = min(i(s), i(o))$
    - Why?
  - s_1 can **execute** $s_2 \leftrightarrow i(s_2) \le i(s_1)$

Clark-Wilson Integrity Model

- Transactions as the basic operation
- 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: $D + YB - W$
- *Well-formed transaction*
  - A series of operations that move system from one consistent state to another
  - State before transaction consistent $\Rightarrow$ state after transaction consistent
- Issue: who examines, certifies transactions done correctly?
  - Separation of duty is crucial

Clark/Wilson Model Entities

- **C**onstrained **D**ata **I**tems (CDI) : data subject to Integrity Control
  - Eg. Account balances
- **U**nconstrained **D**ata **I**tems (UDI): data not subject to IC
  - Eg. Gifts given to the account holders
- **I**ntegrity **V**erification **P**rocedures (IVP)
  - Test CDIs' conformance to integrity constraints at the time IVPs are run (checking that accounts balance)
- **T**ransformation **P**rocedures (TP);
  - Examples?

Clark/Wilson: Certification/Enforcement Rules

- C1: When any IVP is run, it must ensure all CDIs are in valid state
- C2: A TP must transform a set of CDIs from a valid state to another valid state
  - TR must not be used on CDIs it is not certified for
- E1: System must maintain certified relations
  - TP/CDI sets enforced

Clark-Wilson: Certification/Enforcement Rules

- E2: System must control users
  - *user*/TP/CDI mappings enforced
- C3: Relations between (*user*, TP, {CDI}) must support separation of duty
- E3: Users must be authenticated to execute TP
  - Note, unauthenticated users may manipulate UDIs

Clark-Wilson: Certification/Enforcement Rules

- C4: All TPs must log undo information to append-only CDI (to reconstruct an operation)
- C5: A TP taking a UDI as input must either reject it or transform it to a CDI
- E4: Only certifier of a TP may change the list of entities associated with that TP
  - Enforces separation of duty: (HOW?)

Summary

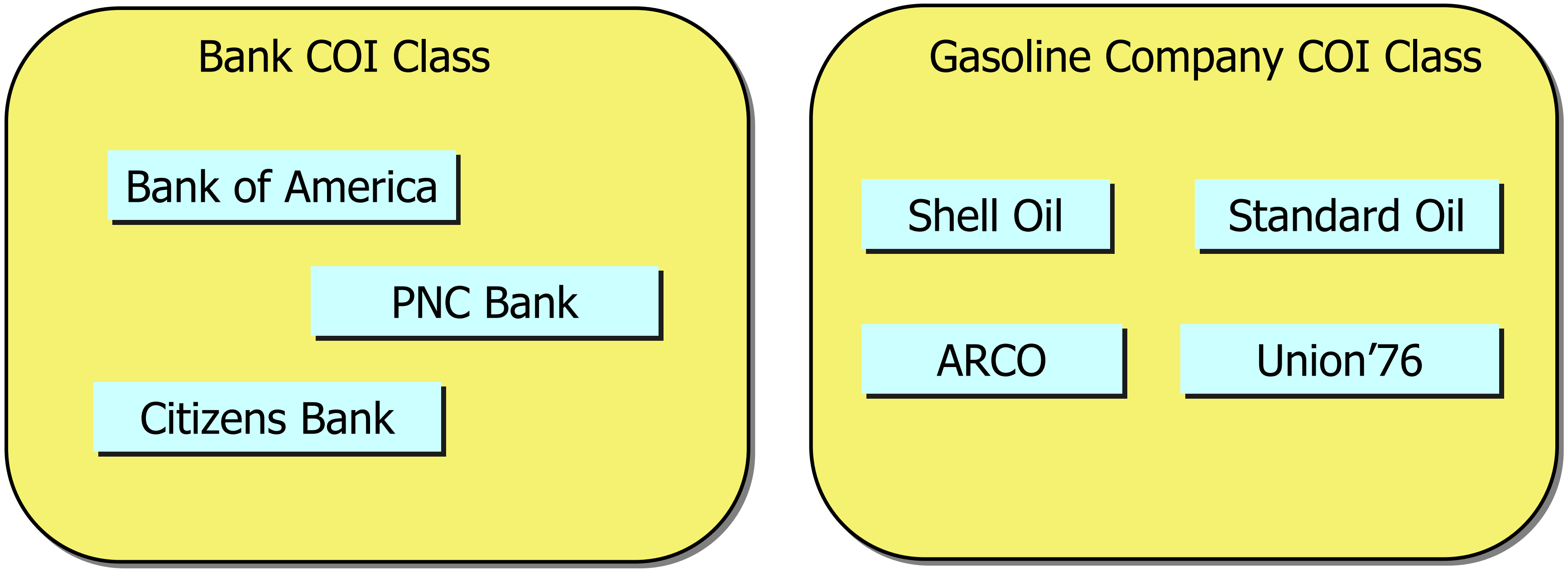
- Integrity policies deal with trust
- Biba based on multilevel integrity
- Clark-Wilson introduce new ideas
  - Commercial firms do not classify data using multilevel scheme
  - they enforce separation of duty
  - Notion of certification is different from enforcement;
    - enforcement rules can be enforced,
    - certification rules need outside intervention, and
    - process of certification is complex and error prone

Hybrid Policies

Chinese Wall Model

- Supports confidentiality and integrity
  - Information flow between items in a Conflict of Interest set
  - Applicable to environment of stock exchange or investment house
- Models conflict of interest
  - *Objects*: items of information related to a company
  - *Company dataset* (CD): contains objects related to a single company
    - Written *CD*(*O*)
  - *Conflict of interest class* (COI): contains datasets of companies in competition
    - Written *COI*(*O*)
    - Assume: each object belongs to exactly one *COI* class

Example


Bank COI Class
Bank of America
PNC Bank
Citizens Bank
Gasoline Company COI Class
Shell Oil
Standard Oil
ARCO
Union'76

CW-Simple Security Property (Read rule)

- CW-Simple Security Property
  - *s* can read *o* *iff* any of the following holds
    - $\exists\, o' \in PR(s)$ such that $CD(o') = CD(o)$
    - $\forall\, o', o' \in PR(s) \Rightarrow COI(o') \neq COI(o)$, or
    - *o* has been "sanitized"

    ($o' \in PR(s)$ indicates o' has been previously read by s)
- Public information may belong to a CD
    - no conflicts of interest arise
    - Sensitive data sanitized

Writing

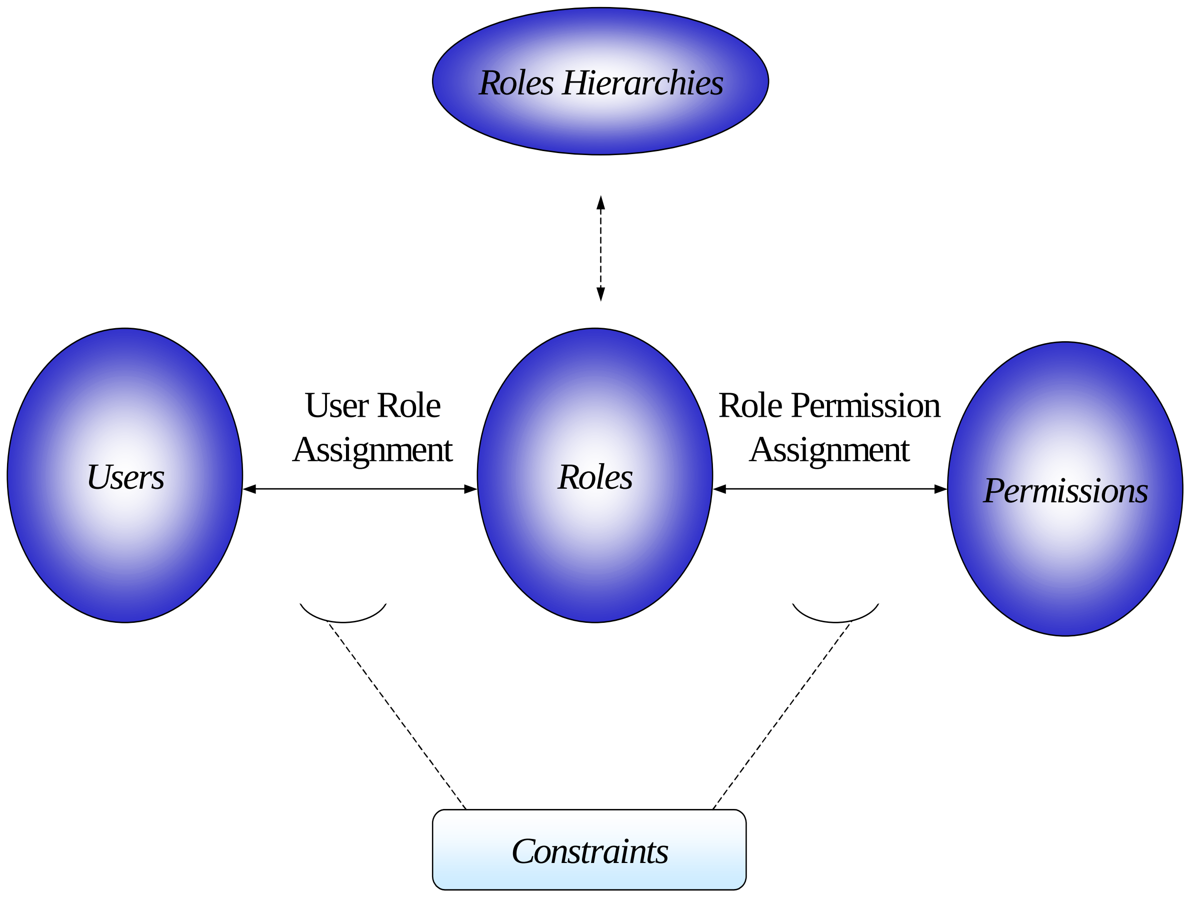
- Alice, Bob work in same trading house
- Alice can read BankOfAmercia’s CD,
- Bob can read CitizensBanks’s CD,
- Both can read ARCO’s CD
- If Alice could write to ARCO’s CD, Bob can read it
  - Hence, indirectly, she can read information from BankOfAmercia’s CD

CW-*-Property (Write rule)

- CW-*- Property
  - *s* can *write* *o* iff the following holds
    - The CW-simple security condition permits S to read O.
    - For all unsanitized objects o', s can read o' $\Rightarrow$ $CD(o') = CD(o)$
  - Alice can read both CDs hence condition 1 is met
  - She can read unsanitized objects of BankOfAmercia, hence condition 2 is false
    - Hence Alice can't write to objects in ARCO's CD.

Role Based Access Control (RBAC)

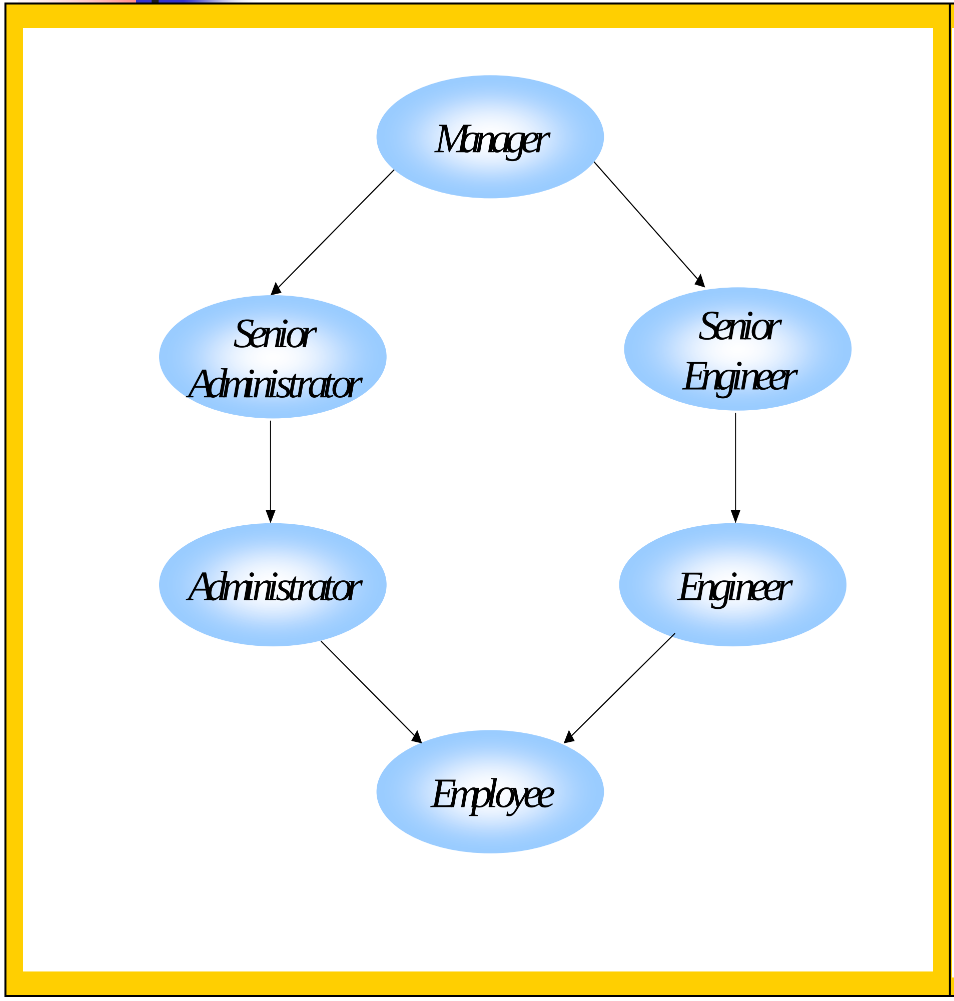
- Access control in organizations is based on "roles that individual users take on as part of the organization"
- A role is "is a collection of permissions"

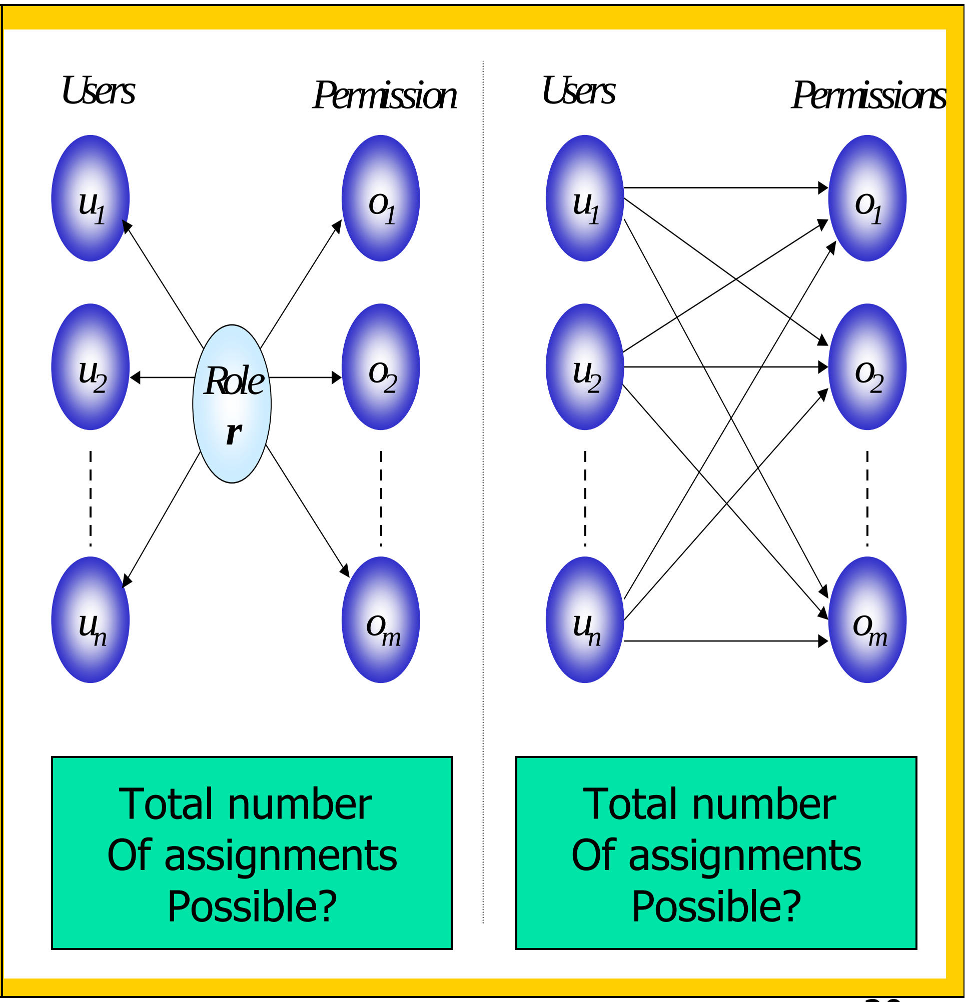


RBAC

- Access depends on function, not identity
  - Example: Allison is bookkeeper for Math Dept. She has access to financial records. If she leaves and Betty is hired as the new bookkeeper, Betty now has access to those records. The role of “bookkeeper” dictates access, not the identity of the individual.

RBAC

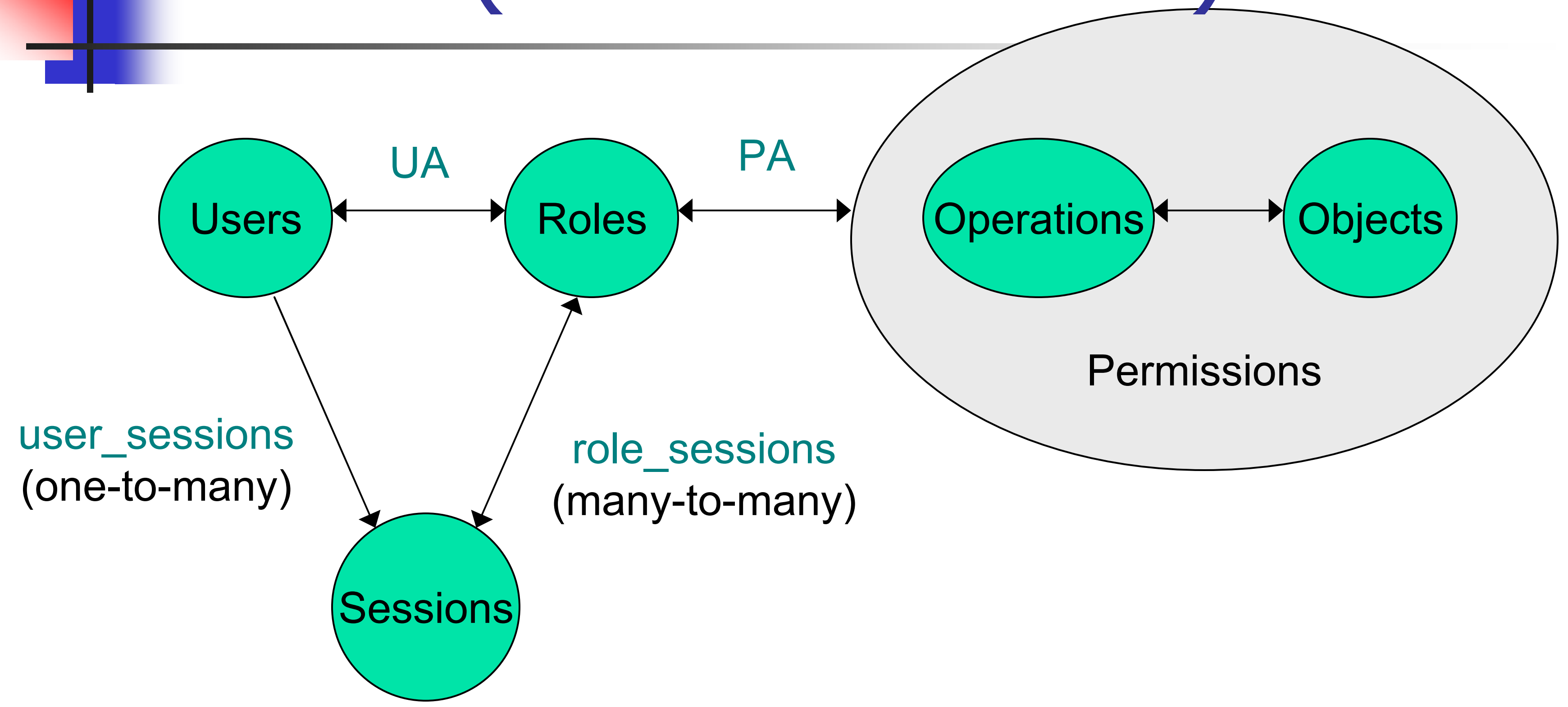




Total number Of assignments Possible?

Total number Of assignments Possible?

RBAC (NIST Standard)



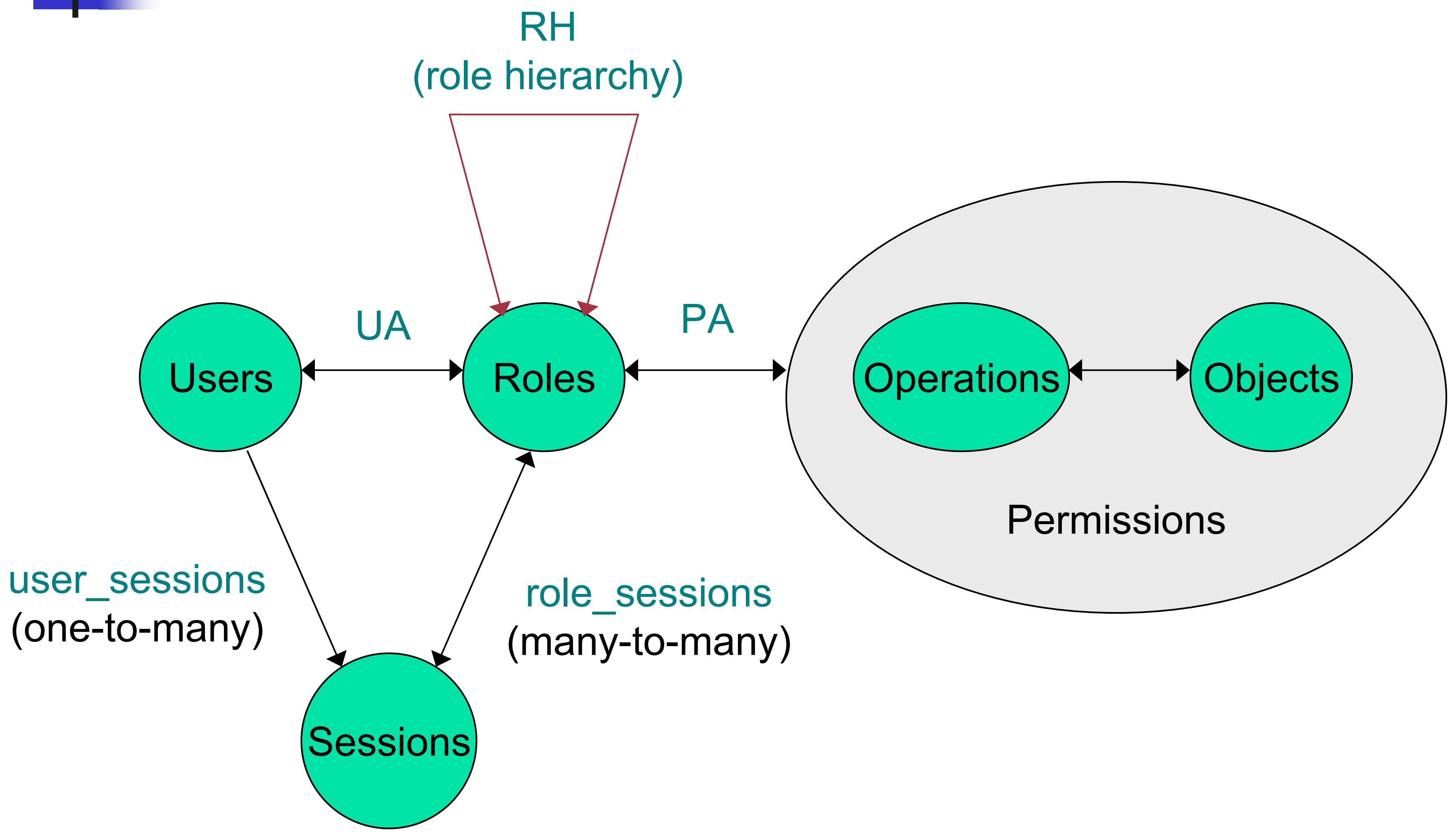
An important difference from classical models is that Subject in other models corresponds to a Session in RBAC

Total number of subjects possible?

Core RBAC (relations)

- Permissions = $2^{\text{Operations x Objects}}$
- UA $\subseteq$ Users x Roles
- PA $\subseteq$ Permissions x Roles
- *assigned_users*: Roles $\rightarrow 2^{\text{Users}}$
- *assigned_permissions*: Roles $\rightarrow 2^{\text{Permissions}}$
- *Op*(p): set of operations associated with permission p
- *Ob*(p): set of objects associated with permission p
- *user_sessions*: Users $\rightarrow 2^{\text{Sessions}}$
- *session_user*: Sessions $\rightarrow$ Users
- *session_roles*: Sessions $\rightarrow 2^{\text{Roles}}$
  - *session_roles*(*s*) = $\{r \mid (\text{session\_user}(s), r) \in \text{UA}\}$
- *avail_session_perms*: Sessions $\rightarrow 2^{\text{Permissions}}$

RBAC with Role Hierarchy



RBAC with General Role Hierarchy

- RH $\subseteq$ Roles x Roles is a partial order
  - called the inheritance relation
  - written as $\geq$.

  $(r_1 \geq r_2) \rightarrow \mathit{authorized\_users}(r_1) \subseteq \mathit{authorized\_users}(r_2)$ & $\mathit{authorized\_permisssions}(r_2) \subseteq \mathit{authorized\_permisssions}(r_1)$

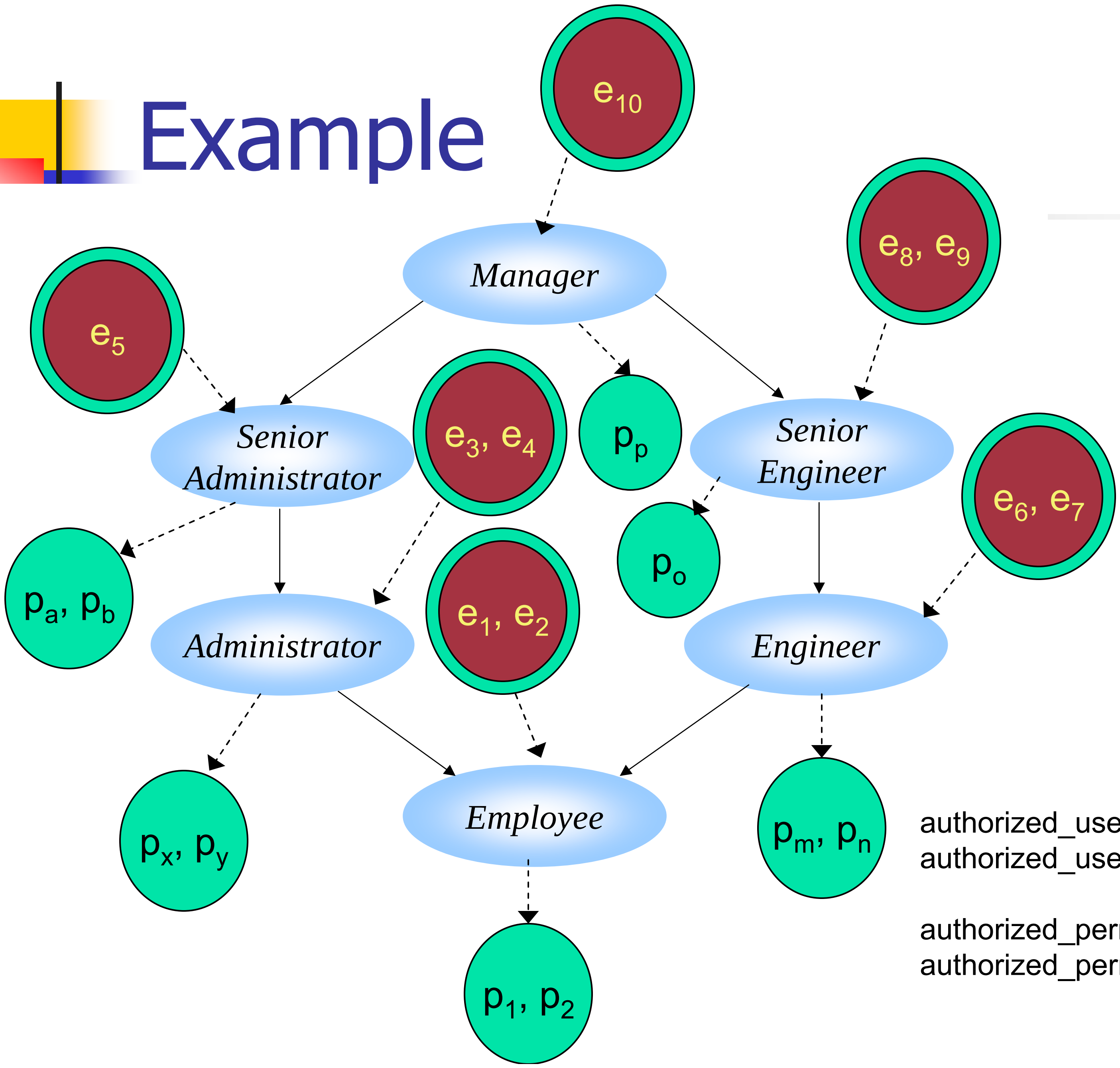
- *authorized_users*: Roles$\rightarrow 2^{\text{Users}}$

  $\mathit{authorized\_users}(r) = \{u \mid r' \geq r \,\&\, (r', u) \in UA\}$

- *authorized_permissions*: Roles$\rightarrow 2^{\text{Permissions}}$

  $\mathit{authorized\_permissions}(\mathrm{r}) = \{p \mid r \geq r' \,\&\, (p, r') \in PA\}$

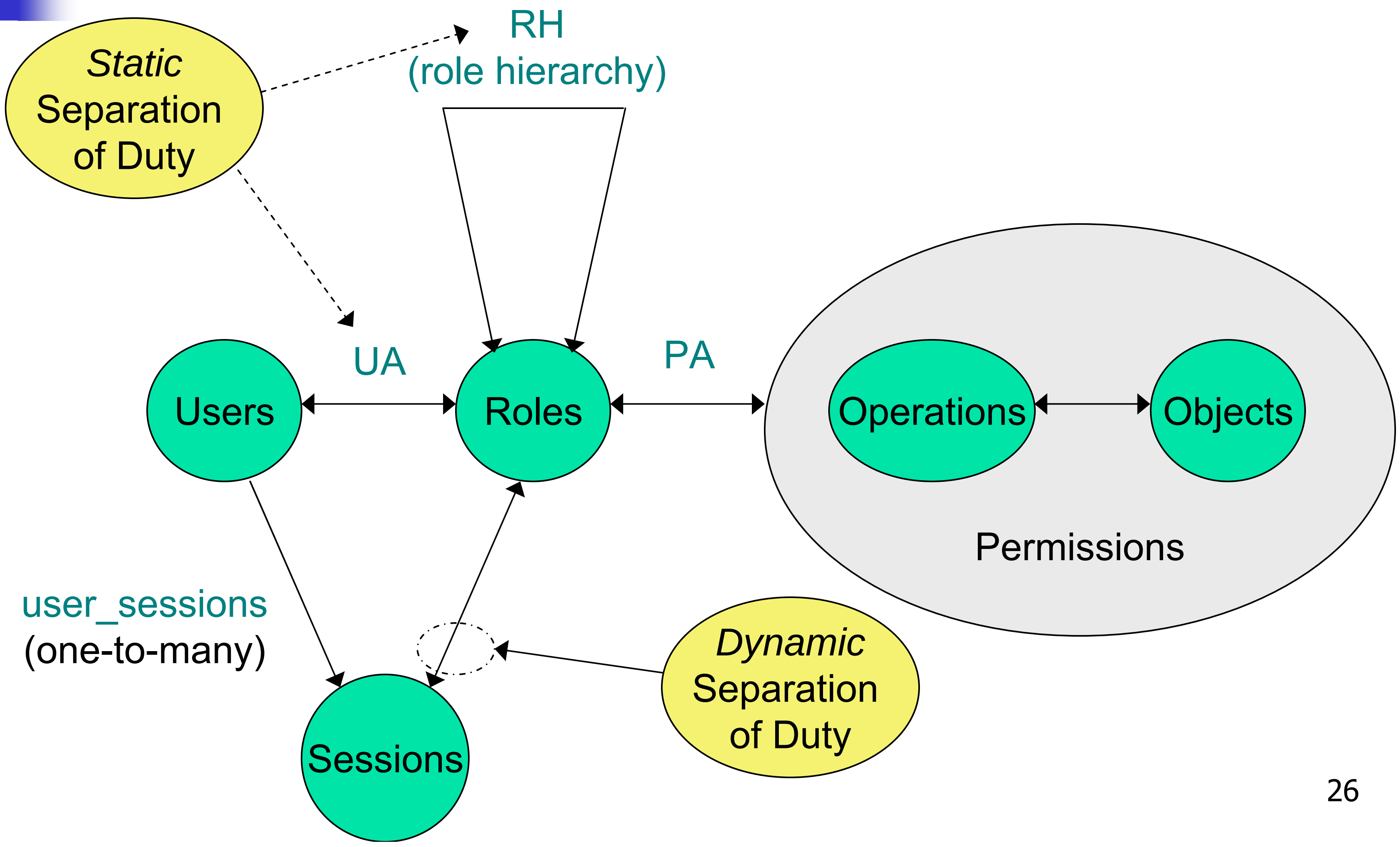
Example



authorized_users(Employee)?
authorized_users(Administrator)?

authorized_permissions(Employee)?
authorized_permissions(Administrator)?

Constrained RBAC


Static Separation of Duty
RH (role hierarchy)
UA
PA
Users
Roles
Operations
Objects
Permissions
user_sessions (one-to-many)
Sessions
Dynamic Separation of Duty

Static Separation of Duty

- $SSD \subseteq 2^{Roles} \times N$
- In absence of hierarchy
  - Collection of pairs (RS, n) where RS is a role set, $n \geq 2$

    for all $(RS, n) \in SSD$, *for all* $t \subseteq RS$:

    $|t| \geq n \rightarrow \bigcap_{r \in t} assigned\_users(\mathrm{r}) = \varnothing$
- In presence of hierarchy
  - Collection of pairs (RS, n) where RS is a role set, n ≥ 2;

    for all $(RS, n) \in SSD$, *for all* $t \subseteq RS$:

    $|t| \geq n \rightarrow \bigcap_{r \in t} authorized\_uers(r) = \varnothing$

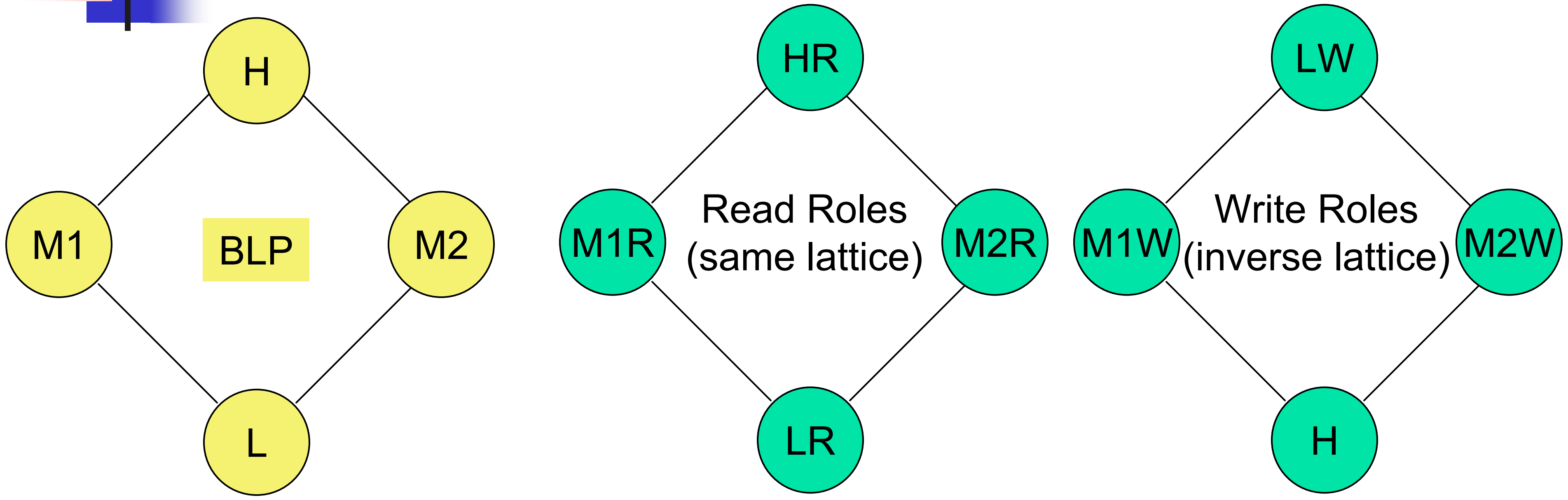
Dynamic Separation of Duty

- $DSD \subseteq 2^{\text{Roles}} \times N$
  - Collection of pairs (RS, n) where RS is a role set, $n \geq 2$;
  - A user cannot activate n or more roles from RS
  - What if both SSD and DSD contains (RS, n)?
    - Consider (RS, n) = ($\{r_1, r_2, r_3\}$, 2)?
    - If SSD – can r_1, r_2 *and* r_3 be assigned to u?
    - If DSD – can r_1, r_2 *and* r_3 be assigned to u?

Advantages of RBAC

- Allows Efficient Security Management
  - Administrative roles, Role hierarchy
- Principle of least privilege allows minimizing damage
- Separation of Duty constraints to prevent fraud
- Allows grouping of objects
- Policy-neutral - Provides generality
- Encompasses DAC and MAC policies

MAC using RBAC



Transformation rules

- $R = \{L_1R, L_2R, \ldots, L_nR, L_1W, L_2W, \ldots, L_nW\}$
- Two separate hierarchies for $\{L_1R, L_2R, \ldots, L_nR\}$ and $\{L_1W, L_2W, \ldots, L_nW\}$
- Each user is assigned to exactly two roles: xR and LW
- Each session has exactly two roles yR and yW
- Permission (o, r) is assigned to xR iff (o, w) is assigned to xW)

RBAC’s Benefits

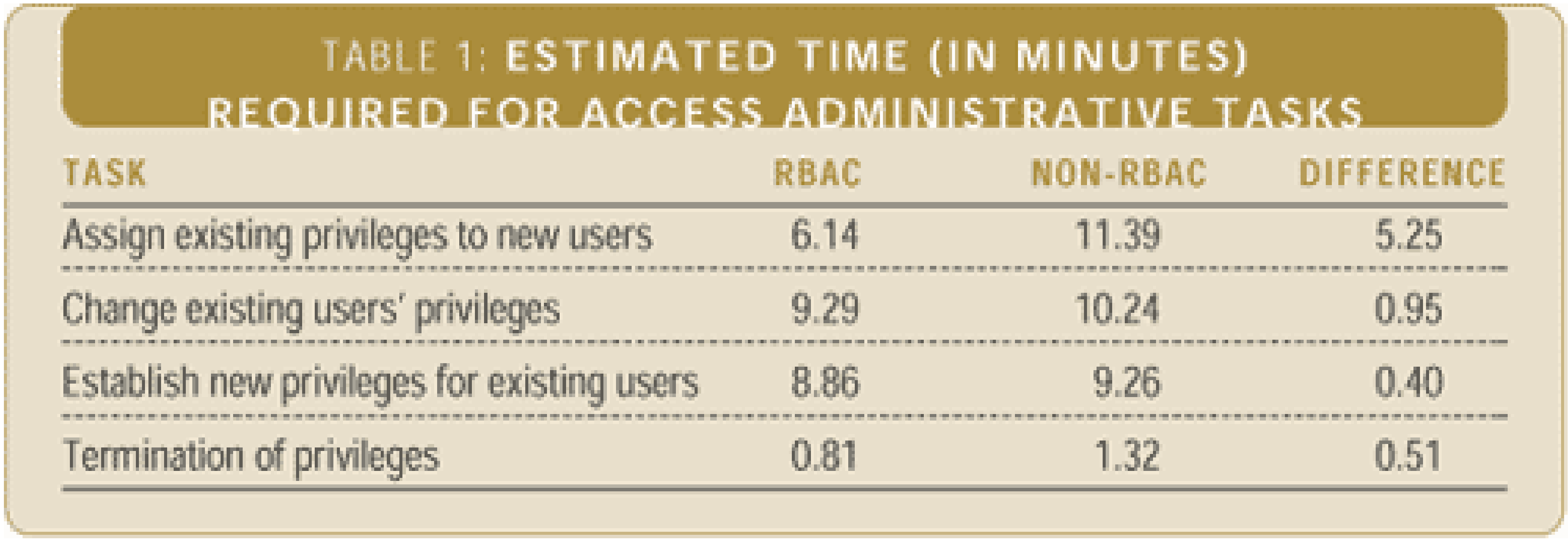


TABLE 1: ESTIMATED TIME (IN MINUTES) REQUIRED FOR ACCESS ADMINISTRATIVE TASKS

TASK	RBAC	NON-RBAC	DIFFERENCE
Assign existing privileges to new users	6.14	11.39	5.25
Change existing users' privileges	9.29	10.24	0.95
Establish new privileges for existing users	8.86	9.26	0.40
Termination of privileges	0.81	1.32	0.51

Cost Benefits

- Saves about 7.01 minutes per employee, per year in administrative functions
  - Average IT admin salary - $59.27 per hour
  - The annual cost saving is:
    - $6,924/1000;
    - $692,471/100,000

How do we get this?

Cost Benefits

- Reduced Employee downtime
  - if new transitioning employees receive their system privileges faster, their productivity is increased
  - 26.4 hours for non-RBAC; 14.7 hours for RBAC
  - For average employee wage of $39.29/hour, the annual productivity cost savings yielded by an RBAC system:
    - $75000/1000; $7.4M/100,000

      (Considering employee turnover of 13% and annual growth rate of 3%)

Policy Composition

Problem: *Consistent* Policies

- Policies defined by different organizations
  - Different needs
  - But sometimes subjects/objects overlap
- Can all policies be met?
  - Different categories
    - Build lattice combining them
  - Different security levels
    - Need to be *levels* – thus must be able to order
  - What if different DAC and MAC policies need to be integrated?

Secure Interoperability

- Principles of secure interoperation [Gong, 96]

  Principle of autonomy

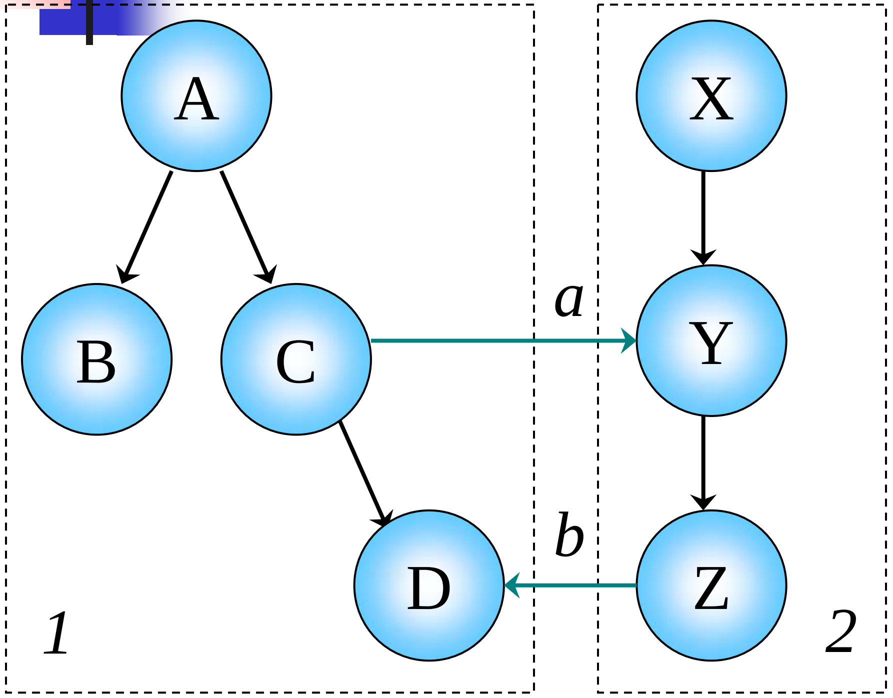
  - If an access is permitted within an individual system, it must also be permitted under secure interoperation

  Principle of security

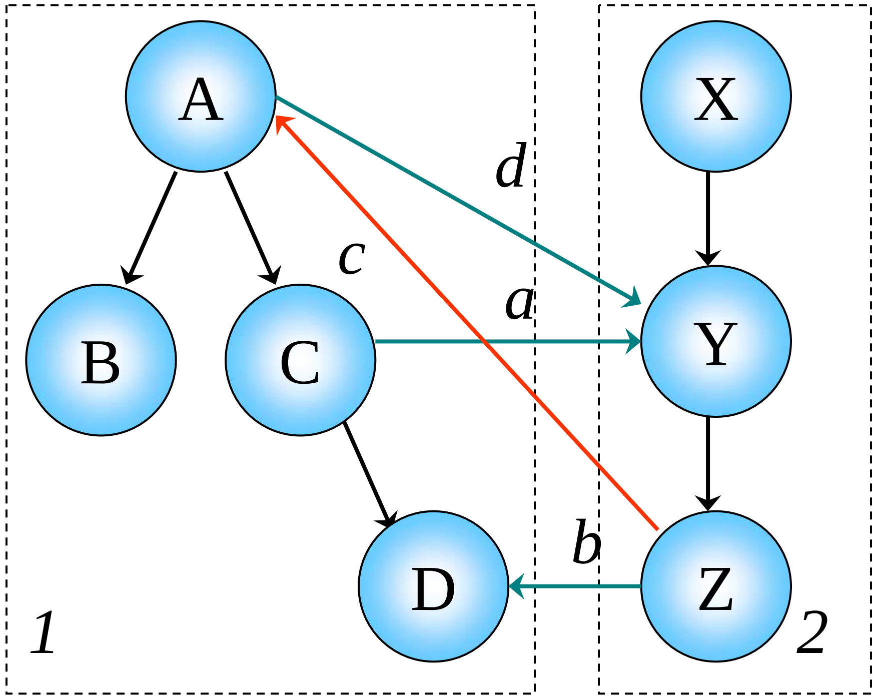
  - If an access is not permitted within an individual system, it must not be permitted under secure interoperation

- Interoperation of secure systems can create new security breaches

Secure Interoperability (Example)



F_{12} = {a, b}



F_{12} = {a, b, c, d}

F_{12} - permitted access between systems 1 and 2

(1) F_{12} = {a, b, d}
Direct access

(2) F_{12} = {c}
Indirect access

Summary

- Integrity polices
  - Level based and non-level based
- Chinese wall is a dynamic policy
  - Conflict classes
- RBAC – several advantages
  - based on duty/responsibility/function
  - Economic benefits as well as diversified