Overview

- Access control
  - Discretionary access control (DAC) $\rightarrow$ HRU
  - Mandatory access control (MAC) $\rightarrow$ BLP, Biba, Chinese Wall
  - Role based access control (RBAC) $\rightarrow$ RBAC_0, RBAC_1, RBAC_2, RBAC_3
  - Attribute-based (ABAC)
- Usage control (UCON)
- Privacy-aware access control
- XACML

HRU

Operation (op)	Conditions
create subject s'	$s' \notin O$
create object o'	$o' \notin O$
enter r into $A[s, o]$	$s \in S$ and $o \in O$
delete r from $A[s, o]$	$s \in S$ and $o \in O$
destroy subject s'	$s' \in S$
destroy object o'	$o' \in O$ and $o \notin S$

Command	
$CREATE(s, o)$	create object o and enter own into $A[s, o]$
$CONFER_{read}(s_1, s_2, o)$	if own in $A[s_1, o]$ then enter $read$ into $A[s_2, o]$
$REVOKE_{read}(s_1, s_2, o)$	if own in $A[s_1, o]$ then delete $read$ from $A[s]$
$TRANSFER_{read}(s_1, s_2, o)$	if $*read$ in $A[s_1, o]$ then enter $read$ into $A[s_2, o]$
$TRANSFER\_ONLY_{read}(s_1, s_2, o)$	if $+read$ in $A[s_1, o]$ then delete $+read$ from $A[s_1, o]$ and enter $+read$ into $A[s_2, o]$.

- **copy flag (*):** subject can transfer privilege to others
- **transfer-only flag (+):** subject can transfer privilege to others, but he loses the privilege
- **safety property:** the program will never produce a wrong result (partial correctness)
- **liveness property:** the program will produce a result
- A state **leaks** a right r if there is a command c that enters r into a position of the access matrix that previously did not contain r.
- A state is **secure** with respect to a right r if no sequence of commands can transform the access matrix into a state that leaks r.

Multi-level security

- **Security level (L or λ):** elements of a hierarchical set (e.g. TopSecret, Secret, Confidential, Unclassified)
  - Each subject S has a maximal security class $\lambda_m(S)$ and a current security class $\lambda_C(S)$ such that $\lambda_c(S) \leq \lambda_m(S)$
- **Categories (C):** elements of non hierarchical set (e.g. administrative, financial)
  - Defines the area o f competence of users and objects
- **Dominance ($\succeq$):** $(L_1, C_1) \succeq (L_2, C_2) \implies L_1 \geq L_2 \wedge C_1 \supseteq C_2$
- Security classes together with dominance relation introduce a lattice (SC, $\succeq$):
  - **Reflexivity**: $\forall_{x \in SC}(x \succeq x)$
  - **Transitivity**: $\forall_{x,y,z \in SC}(x \succeq y \wedge y \succeq z \implies x \succeq z)$
  - **Antisymmetry:** $\forall_{x,y \in SC}(x \succeq y \wedge y \succeq x \implies x = y)$
  - **Least upper bound (lub):** for any x and y, the node z that is immediately above
  - **Greatest lower bound (glb):** for any x and y, the node z that is immediately below
- **Strong tranquility property:** Subjects and objects do not change labels during the lifetime of the system
- **Weak tranquility property:** Subjects and objects do not change labels in a way that violates the "spirit" of the security policy

Bell-laPadula (BLP) model

- Simple security property (**no read up**): S can read $O \iff \lambda_c(S) \geq \lambda(O)$
- *-property (**no write down**): S can write $O \iff \lambda_c(S) \leq \lambda(O)$
- Assumes *strong tranquility property*

Biba model

- Simple security property (**no read down**): s can read $o \iff \lambda(o) \succeq \lambda(s)$
- *-property (**no write up**): s can write $o \iff \lambda(s) \succeq \lambda(o)$
- Assumes *strong tranquility property*
- **low-water-mark for subjects** relaxes the read by allowing subjects to read down
- **low-water-mark for objects** relaxes the write by allowing subjects to write up

Biba + BLP

- Security class of each object and subject consists of two labels
  - Secrecy labels λ_s
  - Integrity labels λ_I
- Combined access rules are
  - Subject s can **read** object $o \iff \lambda_s(s) \succeq \lambda_s(o) \wedge \lambda_I(s) \preceq \lambda_I(o)$
  - Subject s can **write** object $o \iff \lambda_s(s) \preceq \lambda_s(o) \wedge \lambda_I(s) \succeq \lambda_I(o)$

Chinese wall

- Prevent information flows that cause conflicts of interest

- Company information is organized hierarchically in 3 levels:
  - **basic objects** (e.g files)
  - **company datasets** (CDs): group objects referring to the same company
  - **conflict of interest classes** (COI): groups all company datasets whose companies are in competition
- **Simple property**: subject s can **read** object o only if
  - o is in the same company dataset as all the objects that s has already accessed within the same conflict of interest class (**history-based**)
  - o belongs to a different conflict of interest class.
- ***-property**: subject s can **write** object o only if
  - access is permitted by the simple property
  - no object can be read by s which is (I) in a different company dataset than the one for which write access is requested and (II) contains unsanitized information
- **Sanitization:** disguising corporate information, preventing the discovery of the identity of a company

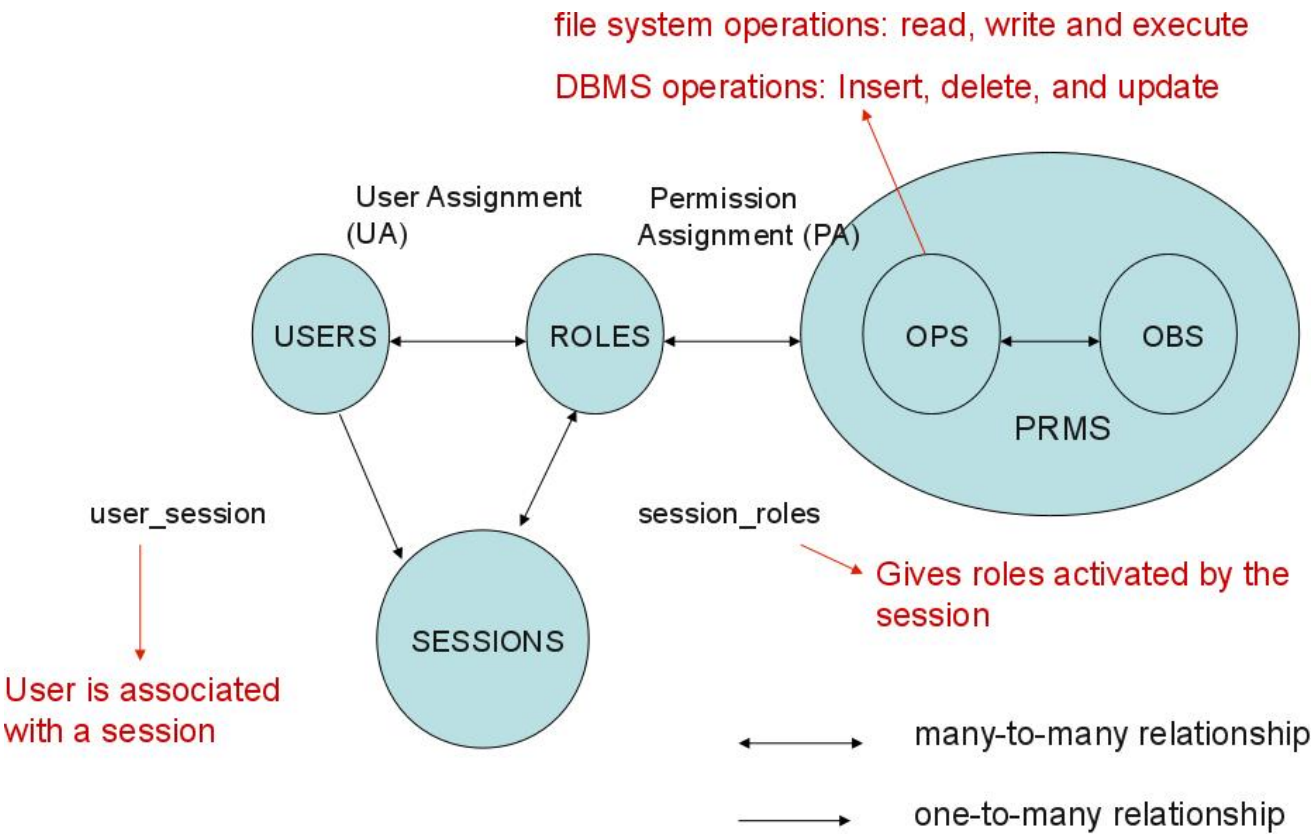
RBAC

Overview of Rule Based Access Control

- $RBAC_0$
  - Roles based on job functions
  - User assignment (UA) to roles
  - Permissions assigned (PA) to rules
- $RBAC_1 = RBAC_0 +$ Role hierarchy (RH)
- $RBAC_2 = RBAC_0 +$ Constraints
- $RBAC_3 = RBAC_1 + RBAC_2$

RBAC$_0$: Core model - notation

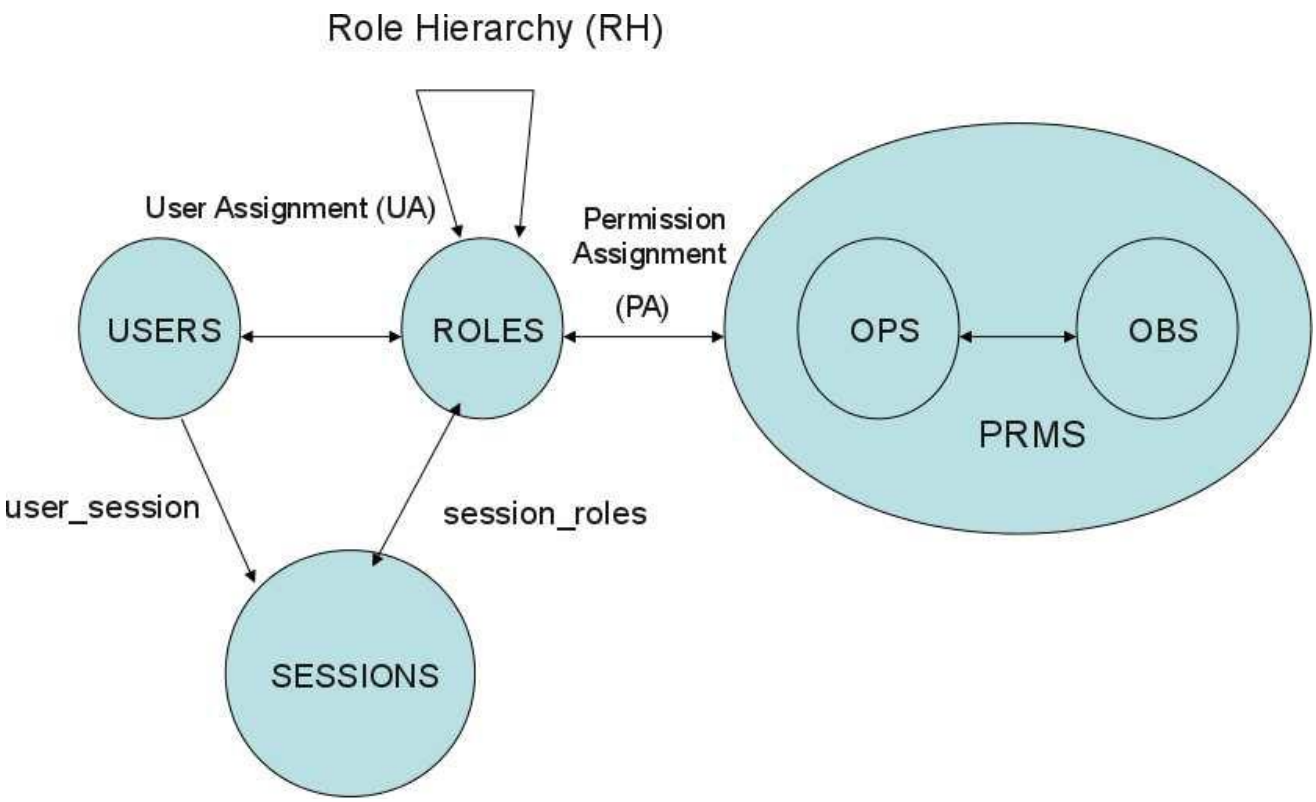
- $U \rightarrow$ Users
- $R \rightarrow$ Roles
- $OPS \rightarrow$ Operations
- $OBS \rightarrow$ Objects
- $P \subseteq OPS \times OBS \rightarrow$ Permissions
- $SE \rightarrow$ Session
- $UA \subseteq U \times R \rightarrow$ User assignment
- $PA \subseteq P \times R \rightarrow$ Permission assignment



RBAC$_1$: Role Hierarchy

- $RH \rightarrow$ Role Hierarchy

- Some roles **subsume** others: a GP can perform all actions that a physician can perform (plus other actions). Granting access to role R implies that access is granted for all specialized roles of R.
- Sometimes the reversed role hierarchy is used: **dominance** relation instead of **specialization**

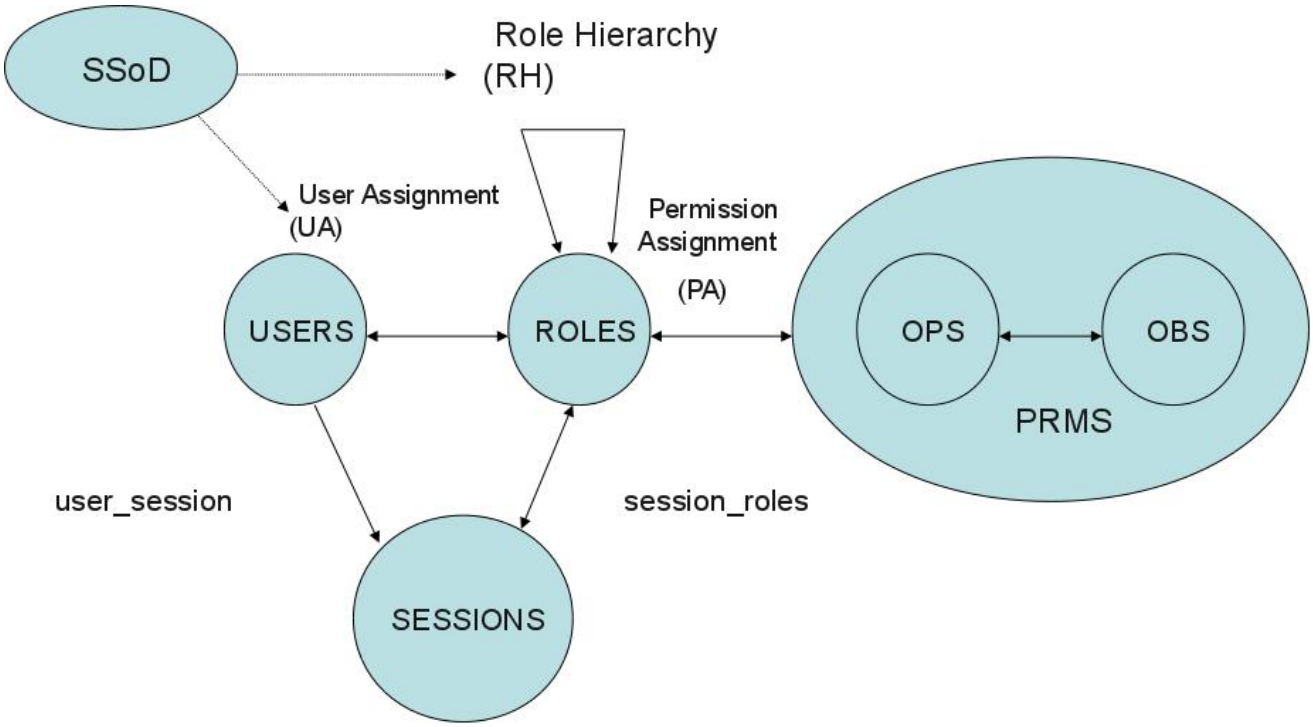


$RBAC_2$: Constraints

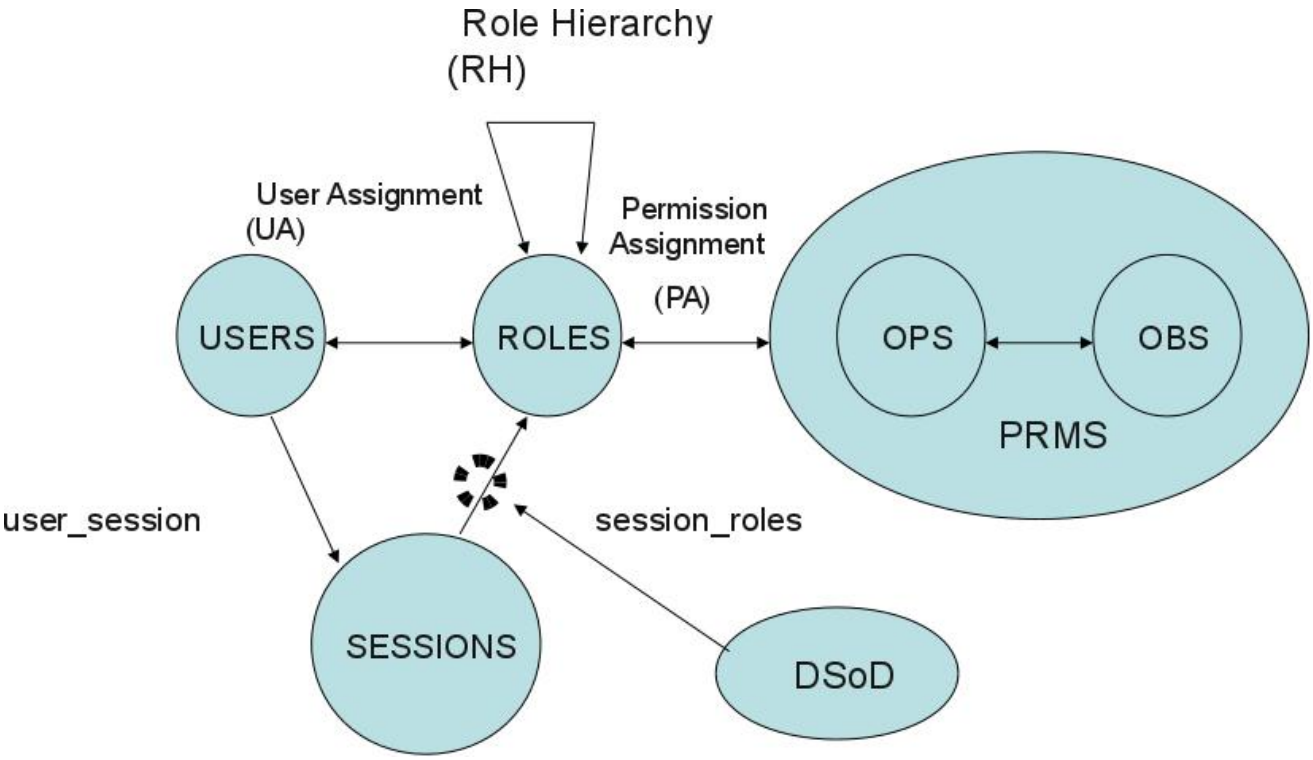
- **Static Separation of Duty** (SSoD): restrict the permissions that can be assigned to a user
  - $ssod(ps, n) \rightarrow$ at least n users are needed to cover all permissions in permission set ps (with $|ps| \geq n$)
- **Dynamic Separation of Duty** (DSoD): restrict the permissions that can be exercised by a user
  - Object based Separation of Duty: no user is allowed to perform all the actions in a business task on a collection of object(s)
  - History-based Separation of Duty: no user may act upon an object that she has previously acted upon
- **Static Mutually Exclusive Roles:** Static separation of duty. Restrictions on the roles that can be assigned to a user
  - $smer(rs, n) \rightarrow$ any user cannot be assigned to n or more roles in role set rs
- **Dynamic Mutually Exclusive Roles:** Dynamic separation of duty. Restriction on the roles that a user can activate in a session
  - $dmer(rs, n) \rightarrow$ users cannot simultaneously activate n or more roles from role set rs in one session
- Other constraints
  - Cardinality constraints on User Assignment
    - E.g. at most k users can belong to the role
  - Prerequisite roles

- A user can only be assigned/activated to a particular role only if it is already assigned/activated to some other specified role.

$RBAC_2$ with Static Mutually Exclusive Roles



$RBAC_2$ with Dynamic Mutually Exclusive Roles



Example question

Define a RBAC3 system to regulate permissions within a bank branch. The system should implement the following requirements:

1. A bank employee can be a clerk, a manager or the head of the bank branch
2. A bank branch can have only one head.
3. The head of the bank branch is a manager.
4. Bank employees can make loan offers to customers.
5. Loan offers should be reviewed by a different clerk or a manager before they can be approved.
6. If the amount of the loan offer is lower than $10K, the offer should be approved by a manager.
7. If the amount of the loan offers is equal or greater than $10K, the offer must be approved by two managers.
8. A bank employee cannot approve loan offers he made or reviewed.

Solution

Roles (R) = {employee, clerk, manager, head, offerer, reviewer, approver_1, approver_2} Objects (OBS) = {loan$_{\geq 10K}$, loan$_{<10K}$ } Operations (OPS) = {offer, review, approve} Session (SE): A session is created when a loan request is received.

Permission assignment

Role	Permission
Offerer	(offerer, loan$_{\geq 10K}$), (offerer, loan$_{<10K}$)
Reviewer	(review, loan$_{\geq 10K}$), (review, loan$_{<10K}$)
Approver_1	(approve, loan$_{\geq 10K}$), (approve, loan$_{<10K}$)
Approver_2	(approve, loan$_{\geq 10K}$)

Role hierarchy

- Employee
  - Clerk
  - Manager
    - Head

Cardinality constraints

- $|\{u|(u, head) \in UA\}| = 1$

Prerequisite roles

- $(u, offerer) \in Session_i \rightarrow (u, employee) \in Session_i$
- $(u, reviewer) \in Session_i \rightarrow (u, clerk) \in Session_i$
- $(u, reviewer) \in Session_i \rightarrow (u, manager) \in Session_i$
- $(u, approver\_1) \in Session_i \rightarrow (u, manager) \in Session_i$
- $(u, approver\_2) \in Session_i \rightarrow (u, manager) \in Session_i$

Mutually exclusive roles

- $dmer(\{offerer, reviewer\}, 2)$
- $dmer(\{offerer, reviewer, approver\_1, approver\_2\}, 2)$

RT0 syntax

- A, B, D: principals
- r, r1, r2: role names
- A.r: a role (a principal + a role name)

Four types of credentials:

Type	Explanation
Type 1: A.r ← D	Role A.r contains principal D as a member
Type 2: A.r ← B.r1	A.r contains role B.r1 as a subset
Type 3: A.r ← A.r1.r2	A.r ⊇ B.r2 for each B in A.r1
Type 4: A.r ← A1.r1 ∩ A2.r2	A.r contains the intersection

Example	semantics	definition
Epub.discount ← Alice	Alice ∈ [[Epub.discount]]	Alice belongs to the role Epu.discount
Epub.discount ← StateU.student	[[StateU.student]] ⊆ [[Epub.discount]]	if StateU states that X is a student then I state that X gets a discount
Epub.discount ← AccredBureau.university.student	For every X ∈ [[AccredBureau.university]], [[X.student]] ⊆ [[Epub.discount]]	If AccredBureau states that X is an accredited university and X states that Y is a student then I state that Y gets a discount
ITbizz.maysign ← ITbizz.manager ∩ ITbizz.senior	[[ITbizz.manager]] ∩ [[ITbizz.senior]] ⊆ [[ITbizz.maysign]]	Anyone showing a manager certificate and a senior certificate, both signed by ITbizz may sign

Example question

Find the semantics.

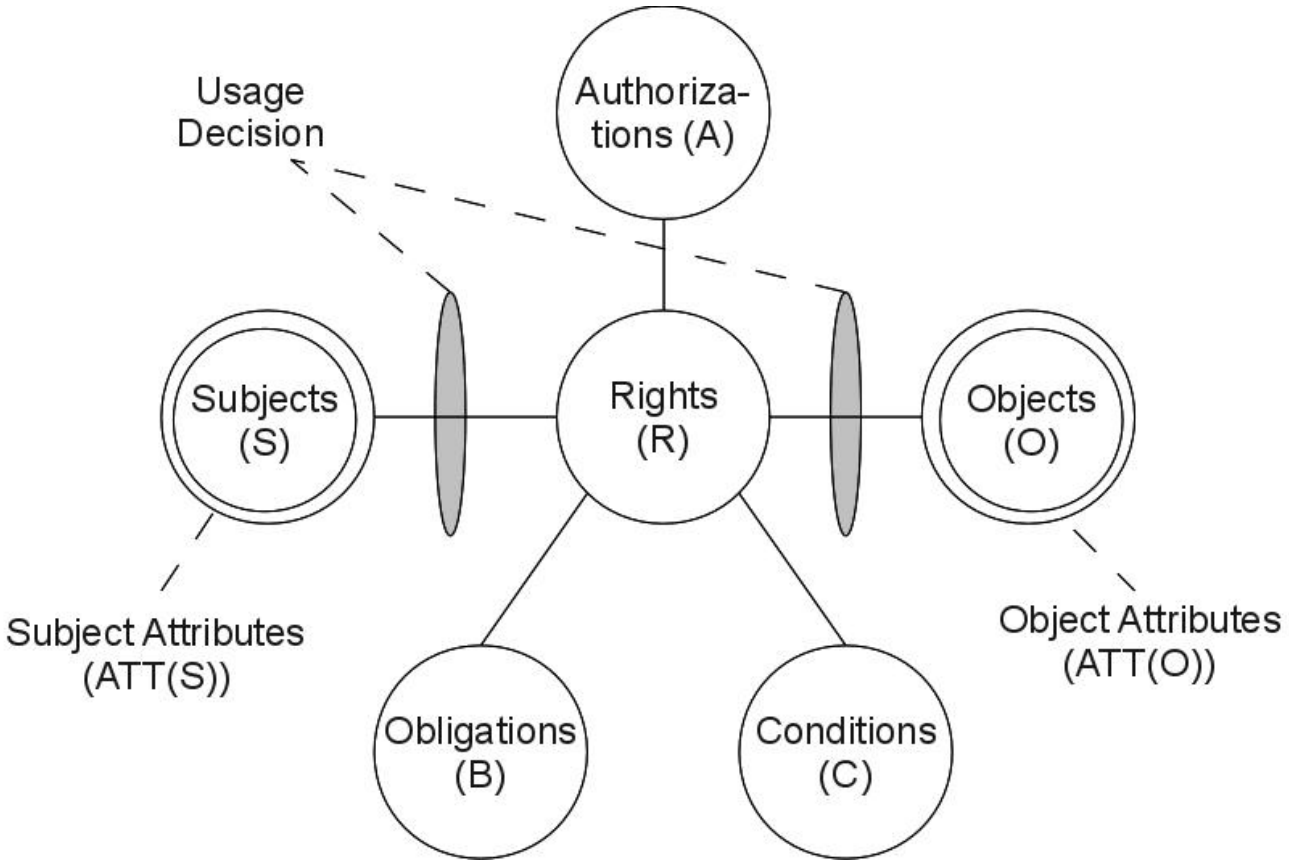
- Alice.s ← Alice.u.v
- Alice.u ← Bob
- Bob.v ← Charlie
- Bob.v ← Charlie.s
- Charlie.s ← David
- Charlie.s ← Edward

Solution

- [[Charlie.s]] = {David, Edward}
- [[Bob.v]] = {Charlie, David, Edward}
- [[Alice.u]] = {Bob}
- [[Alice.s]] = {Charlie, David, Edward}

UCON

- continuity of decisions
  - **pre-decision:** Decide approval or denial of request
  - **ongoing-decision:** Revoke or continue to allow exercise of usage
- mutability of attributes
  - **pre-update:** update before usage
  - **ongoing-update:** update during usage
  - **post-update:** update after usage



Example question (1)

Discuss the UCON model needed to specify a policy supporting the following scenario, and write the UCON policy in the identified model. A Dutch service provider offers pay-per-use services within the Netherlands. A user should be registered with the service provider in order to access a service. Registration is required when a user requests a service for the first time. Moreover, a user can access a service only if he has sufficient credit. The cost of the service is subtracted from the credit of the user after the usage of the service. There can be only 10 simultaneous usages of the same service.

Hint: Model registration to the service provider using obligations.

Solution (1)

Question 1

S: set of users
O: set of objects
M: a set of money amount
IP: set of IPs
$IP_{NL} \subseteq IP$: set of IPs in NL

$OBS = S$
$OBO = \{service\}$
$OB = \{subscribe\}$

$location : S \to IP$
$subscription : S \to \{true, false\}$
$\#usage : O \to \mathbb{N}$ number of usages
$credit : S \to M$
$cost : O \to M$

$ATT(s) : \{location, credit, subscription\}$
$ATT(o) : \{\#usage, cost\}$

$$getPreOBL(s, o, r) = \begin{cases} (s, service, subscribe) & subscription(s) = false \\ \emptyset & subscription(s) = true \end{cases}$$

$allow(s, o, r) \Rightarrow location(s) \in IP_{NL}$
$allow(s, o, r) \Rightarrow preFulfilled(getPreOBL(s, o, r))$
$allow(s, o, r) \Rightarrow credit(s) \geq cost(o)$
$allow(s, o, r) \Rightarrow \#usage(o) < 10$

Or equivalently:
$allow(s, o, r) \Rightarrow location(s) \in IP_{NL} \wedge preFulfilled(getPreOBL(s, o, r)) \wedge credit(s) \geq cost(o) \wedge \#usage(o) < 10$

$preUpdate(subscription(s)) : subscription(s) = true$
$preUpdate(\#usage(o)) : \#usage(o) = \#usage(o) + 1$
$postUpdate(\#usage(o)) : \#usage(o) = \#usage(o) - 1$
$postUpdate(credit(s)) : credit(s) = credit(s) - cost(o)$

3

Example question (2)

A content provider offers an on-demand media streaming service. To access the service, users should subscribe to the service. The provider allows two types of subscription: Basic and Gold. Depending on the type of subscription, users can simultaneously connect a different number of devices to the provider's library of online content. In particular, the Basic subscription allows a user to connect one device whereas the Gold subscription allows a user to connect up to five devices. The service is offered only within the Netherlands.

Solution (2)

Model: UCON$_{preA_{13}preB_1}$

- Pre Authorization (preA)
  - Access is only offered within the Netherlands
  - Constraint on the number of devices that a user can connect to
- Pre Obligation (preB)
  - Subscription to the service
- Update
  - Record whether a user is subscribed (and the type of subscription)
  - Record the number of devices currently connected

Question 1

S: set of users
O: set of objects
IP: set of IPs
$IP_{NL} \subseteq IP$: set of IPs in NL
$SubType = \{Basic, Gold\}$ type of subscription

$OBS = S$
$OBO = \{service\}$
$OB = \{subscribe\text{-}Basic, subscribe\text{-}Gold\}$

$location : S \to IP$
$subscription : S \nrightarrow SubType$ (partial function)
$ndevice : S \to \mathbb{N}$ number of devices a user has connected to the service

$ATT(s) : \{location, subscription, ndevice\}$
$ATT(o) : \{\}$

$$getPreOBL(s, o, r) = \begin{cases} (s, service, subscribe\text{-}Basic) & subscription(s) = \perp \text{ and } s \text{ wants Basic subscription } (*) \\ (s, service, subscribe\text{-}Gold) & subscription(s) = \perp \text{ and } s \text{ wants Gold subscription } (**) \\ \emptyset & subscription(s) = Basic \vee subscription(s) = Basic \end{cases}$$

(symbol '$\perp$' indicates 'undefined')

$allow(s, o, r) \Rightarrow location(s) \in IP_{NL}$
$allow(s, o, r) \Rightarrow preFulfilled(getPreOBL(s, o, r))$
$allow(s, o, r) \Rightarrow (subscription(s) = Basic \wedge ndevice(s) < 1) \vee (subscription(s) = Basic \wedge ndevice(s) < 5)$

$preUpdate(subscription(s)) : subscription(s) = Basic\ (*)$
$preUpdate(subscription(s)) : subscription(s) = Gold\ (**)$
$preUpdate(ndevice(s)) : ndevice(s) = ndevice(s) + 1$
$postUpdate(ndevice(s)) : ndevice(s) = ndevice(s) - 1$

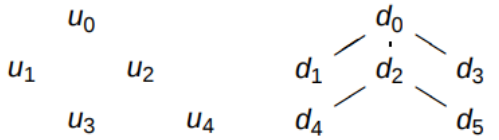
BLP-model in UCON

- L lattice of security labels with dominance relation $\succeq$
- clearance: $S \to L$
- classification: $O \to L$
- $ATT(S) = \{clearance\}$
- $ATT(O) = \{classification\}$
- $allowed(s, o, read) \implies clearance(s) \succeq classification(o)$
- $allowed(s, o, write) \implies clearance(s) \preceq classification(o)$

EPAL

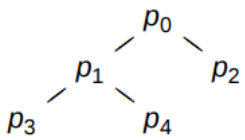
Question 3(a)

Let $Voc = (UH, DH, PH, AH, OM)$ be a vocabulary where the user hierarchy UH, data hierarchy DH, purpose hierarchy PH and action hierarchy AH are defined in the figure, and $OM = (O, \rightarrow)$ is the obligation model with $O = \{o_1, o_2, o_3, o_4, o_5, o_6\}$ and $\rightarrow = \{o_3 \rightarrow o_2, o_2 \rightarrow o_5, o_4 \rightarrow o_5\}$.

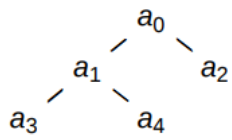


(m) User Hierarchy

(n) Data Hierarchy



(o) Purpose Hierarchy



(p) Action Hierarchy

$$pol_1 = \begin{cases} \langle (u_0, d_2, p_1, a_0)(+, true, \{o_1\}) \rangle \\ \langle (u_2, d_0, p_2, a_1)(-, true, \{o_2\}) \rangle \\ \langle (u_2, d_4, p_0, a_0)(\circ, true, \{o_3\}) \rangle \\ \langle (u_1, d_0, p_1, a_0)(+, true, \{o_4\}) \rangle \\ \text{Default ruling:} - \\ \text{Default obligations:} \{o_5\} \end{cases}$$

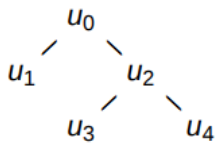
$$pol_2 = \begin{cases} \langle (u_0, d_2, p_1, a_0)(+, true, \{o_1\}) \rangle \\ \langle (u_2, d_0, p_2, a_1)(-, true, \{o_3\}) \rangle \\ \langle (u_2, d_4, p_0, a_0)(\circ, true, \{o_3\}) \rangle \\ \langle (u_1, d_0, p_1, a_0)(+, true, \{o_4\}) \rangle \\ \langle (u_0, d_0, p_0, a_0)(-, true, \{o_4, o_6\}) \rangle \\ \text{Default ruling:} + \\ \text{Default obligations:} \{o_5\} \end{cases}$$

Determine whether pol_2 is a refinement of pol_1: pol_2 is a refinement of pol_1

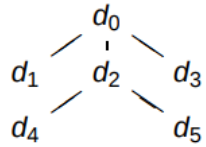
- Default ruling: in pol_1 $-$, in pol_2 $+$.
- The fifth rule of pol_2 ($\langle (u_0, d_0, p_0, a_0)(-, true, \{o_4, o_6\}) \rangle$) is not in pol_1. This rule applies to all queries, thus acting as a default rule. The ruling of this rule is equal to the default ruling of pol_1. Also, $\{o_4, o_6\} \rightarrow \{o_4\} \rightarrow \{o_5\}$.
- Rule 2 of pol_1 and pol_2 have same scope but different obligations. But $\{o_3\} \rightarrow \{o_2\}$.

Question 3(b)

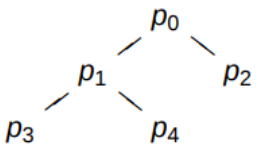
Let $Voc = (UH, DH, PH, AH, OM)$ be a vocabulary where the user hierarchy UH, data hierarchy DH, purpose hierarchy PH and action hierarchy AH are defined in the figure, and $OM = (O, \rightarrow)$ is the obligation model with $O = \{o_1, o_2, o_3, o_4, o_5, o_6\}$ and $\rightarrow = \{o_3 \rightarrow o_2, o_2 \rightarrow o_5, o_4 \rightarrow o_5\}$.



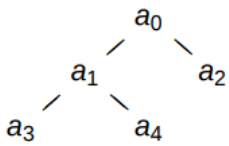
(y) User Hierarchy



(z) Data Hierarchy



() Purpose Hierarchy



() Action Hierarchy

$$pol_1 = \begin{cases} \langle (u_0, d_2, p_1, a_0)(+, true, \{o_1\}) \rangle \\ \langle (u_2, d_0, p_2, a_1)(-, true, \{o_2\}) \rangle \\ \langle (u_2, d_4, p_0, a_0)(\circ, true, \{o_3\}) \rangle \\ \langle (u_1, d_0, p_1, a_0)(+, true, \{o_4\}) \rangle \\ \text{Default ruling:} - \\ \text{Default obligations:} \{o_5\} \end{cases}$$

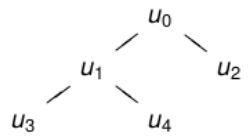
$$pol_3 = \begin{cases} \langle (u_0, d_2, p_1, a_0)(+, true, \{o_1\}) \rangle \\ \langle (u_2, d_0, p_2, a_1)(-, true, \{o_2\}) \rangle \\ \langle (u_2, d_4, p_0, a_0)(\circ, true, \{o_3\}) \rangle \\ \langle (u_1, d_0, p_1, a_0)(+, true, \{o_4\}) \rangle \\ \langle (u_4, d_2, p_2, a_4)(+, true, \{o_6\}) \rangle \\ \text{Default ruling:} - \\ \text{Default obligations:} \{o_5\} \end{cases}$$

Determine whether pol_3 is a refinement of pol_1: pol_3 is a refinement of pol_1

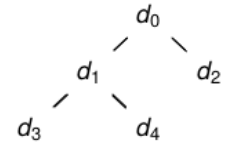
- The fifth rule of pol_3 ($\langle (u_4, d_2, p_2, a_4)(+, true, \{o_6\}) \rangle$) is not in pol_1. If there exists a request to which this rule is applied, then pol_3 is not a refinement of pol_1.
- However, the fifth rule of pol_3 is covered by the second rule. Thus, the fifth rule is never applied.

Question 3(a)

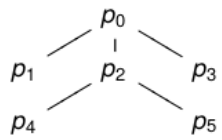
Let $Voc = (UH, DH, PH, AH, OM)$ be a vocabulary where the user hierarchy UH, data hierarchy DH, purpose hierarchy PH and action hierarchy AH are defined below, and $OM = (O, \rightarrow)$ is the obligation model with $O = \{o_1, o_2, o_3, o_4, o_5, o_6\}$ and $\rightarrow = \{o_2 \rightarrow o_3, o_2 \rightarrow o_5, o_3 \rightarrow o_4\}$.



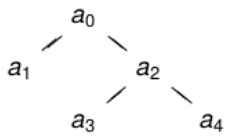
(m) User Hierarchy



(n) Data Hierarchy



(o) Purpose Hierarchy



(p) Action Hierarchy

$$pol_1 = \begin{cases} \langle (u_0, d_1, p_0, a_2)(+, true, \{o_1\}) \rangle \\ \langle (u_1, d_0, p_2, a_2)(-, true, \{o_2\}) \rangle \\ \langle (u_2, d_4, p_4, a_0)(\circ, true, \{o_1\}) \rangle \\ \langle (u_1, d_0, p_1, a_2)(+, true, \{o_5\}) \rangle \\ \text{Default ruling:} - \\ \text{Default obligations:} \{o_4, o_5\} \end{cases}$$

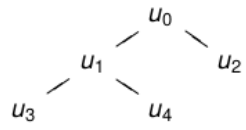
$$pol_2 = \begin{cases} \langle (u_0, d_1, p_0, a_2)(+, true, \{o_1\}) \rangle \\ \langle (u_1, d_0, p_2, a_2)(-, true, \{o_2\}) \rangle \\ \langle (u_2, d_4, p_4, a_0)(\circ, true, \{o_1\}) \rangle \\ \langle (u_3, d_1, p_4, a_3)(+, true, \{o_6\}) \rangle \\ \langle (u_1, d_0, p_1, a_2)(+, true, \{o_5\}) \rangle \\ \text{Default ruling:} - \\ \text{Default obligations:} \{o_2\} \end{cases}$$

Determine whether pol_2 is a refinement of pol_1: pol_2 is a refinement of pol_1

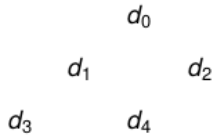
- Default obligations: in pol_1 $\{o_4, o_5\}$, in pol_2 $\{o_2\}$. However, $\{o_2\} \rightarrow \{o_4, o_5\}$ (according to the definition of $\rightarrow$: $o_2 \rightarrow o_5$ and $o_2 \rightarrow o_3 \rightarrow o_4$)
- The fourth rule of pol_2 $((u_3, d_1, p_4, a_3)(+, true, \{o_6\}))$ is not in pol_1. However, this rule will never be reached as it is covered by the first rule!!

Question 3(b)

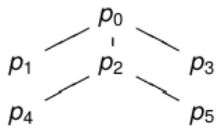
Let $Voc = (UH, DH, PH, AH, OM)$ be a vocabulary where the user hierarchy UH, data hierarchy DH, purpose hierarchy PH and action hierarchy AH are defined below, and $OM = (O, \rightarrow)$ is the obligation model with $O = \{o_1, o_2, o_3, o_4, o_5, o_6\}$ and $\rightarrow = \{o_2 \rightarrow o_3, o_2 \rightarrow o_5, o_3 \rightarrow o_4\}$.



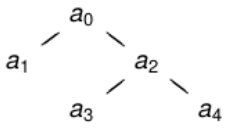
(y) User Hierarchy



(z) Data Hierarchy



() Purpose Hierarchy



() Action Hierarchy

$$pol_1 = \begin{cases} \langle (u_0, d_1, p_0, a_2)(+, true, \{o_1\}) \rangle \\ \langle (u_1, d_0, p_2, a_2)(-, true, \{o_2\}) \rangle \\ \langle (u_2, d_4, p_4, a_0)(\circ, true, \{o_1\}) \rangle \\ \langle (u_1, d_0, p_1, a_2)(+, true, \{o_5\}) \rangle \\ \text{Default ruling:} - \\ \text{Default obligations:} \{o_4, o_5\} \end{cases}$$

$$pol_3 = \begin{cases} \langle (u_0, d_1, p_0, a_2)(+, true, \{o_1\}) \rangle \\ \langle (u_1, d_0, p_2, a_2)(-, true, \{o_3\}) \rangle \\ \langle (u_2, d_4, p_4, a_0)(\circ, true, \{o_1\}) \rangle \\ \langle (u_1, d_0, p_1, a_2)(+, true, \{o_5\}) \rangle \\ \langle (u_0, d_0, p_0, a_0)(-, true, \{o_4, o_5\}) \rangle \\ \text{Default ruling:} + \\ \text{Default obligations:} \{o_4, o_5\} \end{cases}$$

Determine whether pol_3 is a refinement of pol_1:

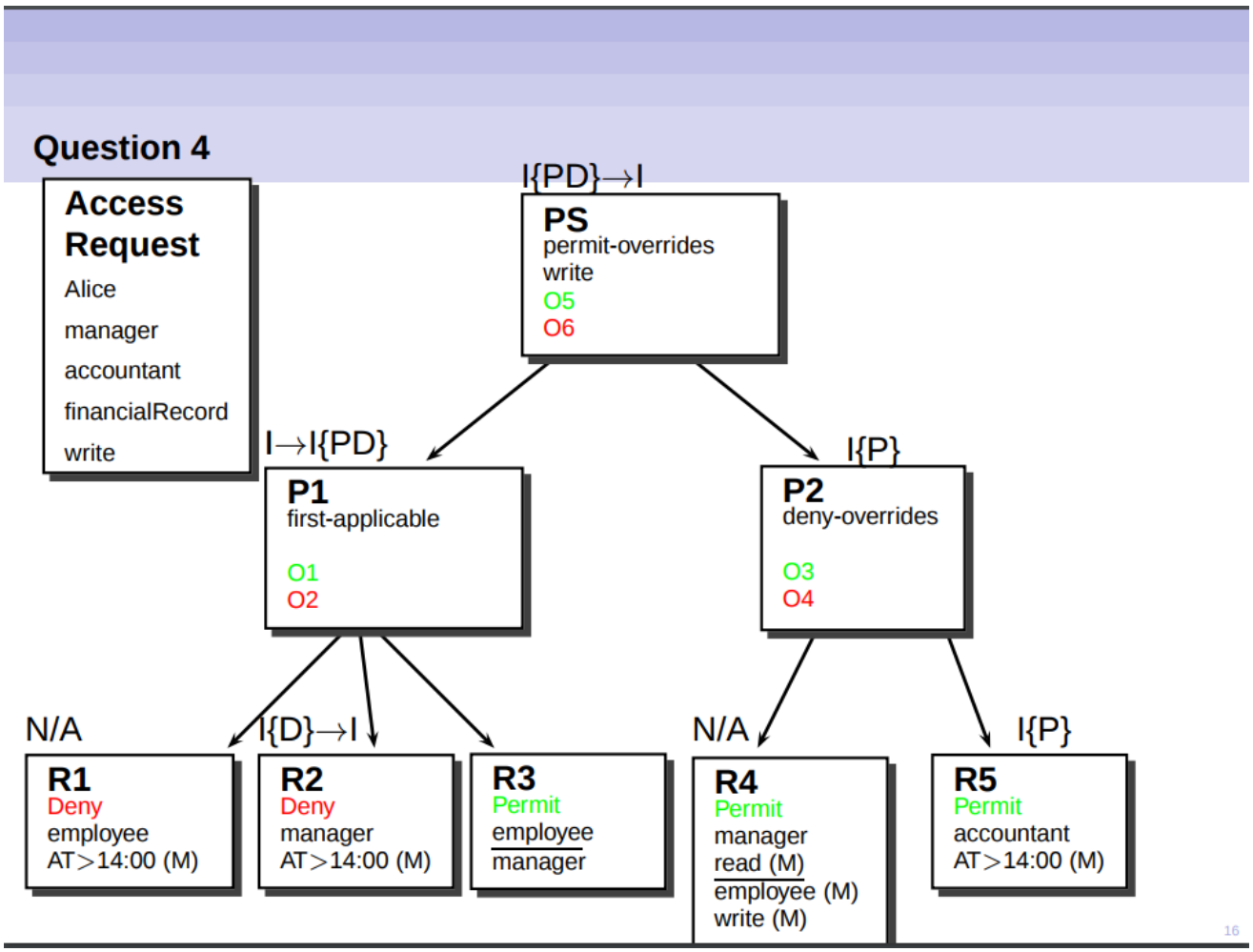
- The fifth rule of pol_3 ($(u_0, d_0, p_0, a_0)(-, true, \{o_4, o_5\})$) is not in pol_1. Also, the default ruling and default obligations of pol_1 and pol_3 are different. However, it is easy to observe that the fifth rule of pol_3 behaves as the default ruling and default obligation of pol_1 and the default ruling and default obligations of pol_1 will be never reached.
- Obligations in the second rule: in pol_1 $\{o_2\}$, in pol_2 $\{o_3\}$. We can observe that $o_3 \not\rightarrow o_2$.

pol_3 is NOT a refinement of pol_1. Consider request (u_0, d_0, p_0, a_0):

- pol_1: $+, \{o_2\}$
- pol_3: $+, \{o_3\}$

17

XACML


Question 4
Access Request
Alice
manager
accountant
financialRecord
write
I{PD}→I
PS
permit-overrides
write
O5
O6
I→I{PD}
P1
first-applicable
O1
O2
I{P}
P2
deny-overrides
O3
O4
N/A
R1
Deny
employee
AT>14:00 (M)
I{D}→I
R2
Deny
manager
AT>14:00 (M)
R3
Permit
employee
manager
N/A
R4
Permit
manager
read (M)
employee (M)
write (M)
I{P}
R5
Permit
accountant
AT>14:00 (M)
16

Reduction

Exercise (Exam 28/1/2015)

Consider an operator α defined over the three-valued decision set $\mathcal{D}_3 = \{1, 0, \bot\}$ defined as follows:

α	1	0	$\bot$
1	$\bot$	$\bot$	1
0	$\bot$	$\bot$	0
$\bot$	1	0	$\bot$

1. Define α over the seven-valued decision set $\mathcal{D}_7$ point-wise (Recall $\mathcal{D}_7 = \wp(\mathcal{D}_3) \setminus \emptyset$)
2. Let $\mathcal{D}_6 = \{P, D, NA, I(P), I(D), I(PD)\}$ be a six-valued decision set and $\rho_{76} : \mathcal{D}_7 \to \mathcal{D}_6$ a decision reduction that maps a decision in $\mathcal{D}_7$ to a decision in $\mathcal{D}_6$ such that

$$\rho_{76}(d) = \begin{cases} P & \text{if } d = 1 \\ D & \text{if } d = \{0\} \\ NA & \text{if } d = \bot \\ I(P) & \text{if } d = \{1, \bot\} \\ I(D) & \text{if } d = \{0, \bot\} \\ I(PD) & \text{if } d = \{1, 0\} \vee d = \{1, 0, \bot\} \end{cases}$$

Determine whether ρ_{76} is safe with respect to the operator defined over $\mathcal{D}_7$.

Let $\mathcal{D}_6 = \{P, D, NA, I(P), I(D), I(PD)\}$ be a six-valued decision set and $\rho_{76} : \mathcal{D}_7 \to \mathcal{D}_6$ a decision reduction that maps a decision in $\mathcal{D}_7$ to a decision in $\mathcal{D}_6$ such that

$$\rho_{76}(d) = \begin{cases} P & \text{if } d = 1 \\ D & \text{if } d = \{0\} \\ NA & \text{if } d = \bot \\ I(P) & \text{if } d = \{1, \bot\} \\ I(D) & \text{if } d = \{0, \bot\} \\ I(PD) & \text{if } d = \{1, 0\} \vee d = \{1, 0, \bot\} \end{cases}$$

Determine whether ρ_{76} is safe with respect to the operator defined over $\mathcal{D}_7$.

α	1	0	$\bot$	$1, \bot$	$0, \bot$	$1, 0$	$1, 0, \bot$
1	$\bot$	$\bot$	1	$1, \bot$	$1, \bot$	$\bot$	$1, \bot$
0	$\bot$	$\bot$	0	$0, \bot$	$0, \bot$	$\bot$	$0, \bot$
$\bot$	1	0	$\bot$	$1, \bot$	$0, \bot$	$1, 0$	$1, 0, \bot$
$1, \bot$	$1, \bot$	$0, \bot$	$1, \bot$	$1, \bot$	$1, 0, \bot$	$1, 0, \bot$	$1, 0, \bot$
$0, \bot$	$1, \bot$	$0, \bot$	$0, \bot$	$1, 0, \bot$	$0, \bot$	$1, 0, \bot$	$1, 0, \bot$
$1, 0$	$\bot$	$\bot$	$1, 0$	$1, 0, \bot$	$1, 0, \bot$	$\bot$	$1, 0, \bot$
$1, 0, \bot$	$1, \bot$	$0, \bot$	$1, 0, \bot$	$1, 0, \bot$	$1, 0, \bot$	$1, 0, \bot$	$1, 0, \bot$

Observe that $\{1, 0, \bot\}$ and $\{1, 0\}$ should have the same behavior because $\rho_{76}(\{1, 0, \bot\}) = \rho_{76}(\{1, 0\})$. However:

$$\rho_{76}(\alpha(\{1\}, \{1, 0, \bot\})) = \rho_{76}(\{1, \bot\}) = I(P)$$
$$\rho_{76}(\alpha(\{1\}, \{1, 0\})) = \rho_{76}(\{\bot\}) = NA$$

Purpose based access control

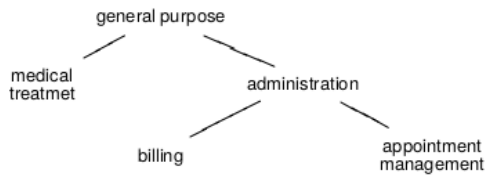
Question 2 (Purpose-based Access Control)

The medical staff of a hospital (i.e., doctors and nurses) can read patients' medical records for providing medical treatment. A doctor can treat a patient if he has worked at least two years in the hospital or he has at least five years of experience. A nurse can only access medical records of those patients in his/her department. Administrative staff can access patient information for billing purposes. In addition, receptionists (who are part of the administrative staff) can access patients' demographic information and doctors' schedule for managing appointments. Administrative staff can only access patients' information within the hospital network.

- Define the purpose hierarchy and role hierarchy along with role and system attributes for the scenario above.
- Define the access purpose authorizations for the scenario.
- Determine whether a receptionist with ten year experience can request access to the medical record of a patient in order to make an appointment (Assume the receptionist is within the hospital network). Justify the answer.

Question 2 (Access Purpose Verification)

Purpose Hierarchy



Role Hierarchy

$\langle medical\ treatment, \langle doctor, Year@Hospital \geq 2 \vee YearExperience \geq 5 \rangle \rangle$
$\langle medical\ treatment, \langle nurse, department = patientDepartment \rangle \rangle$
$\langle billing, \langle administration\ staff, inHospitalNetwork = true \rangle \rangle$
$\langle appointment\ management, \langle receptionist, inHospitalNetwork = true \rangle \rangle$

Determine whether a receptionist with ten year experience can request access to the medical record of a patient in order to make an appointment (Assume the receptionist is within the hospital network).

Access Request: (u,receptionist, medical record, appointment management)

Yes. Access purpose is valid.

Note: The verification of the intended purpose of data is done in the purpose compliance step. Access purpose verification is only used to check whether a user can specify a certain access purpose in the request.