# PBAC: Provision-Based Access Control Model

Michiharu Kudo, International Journal of Information Security, Springer Verlag, vol. 1, no. 2, pp. 116-130, 2002

Amir Reza Masoumzadeh

Database Security Course
Sharif University of Technology
May 17th, 2006



# Introduction

- Almost all access control models have assumed "grant the access or deny it"
- Bind authorization rules with required operations such as logging and encrypting
- Provisional action: tell the user that his request will be authorized provided he (and/or the system) performs certain security actions, e.g.
  - You are allowed to read sensitive information, but you must sign terms and conditions first
  - If an unauthorized access request is submitted then deny the access and a warning message must be sent to an administrator

# Design Principles (Fundamentals)

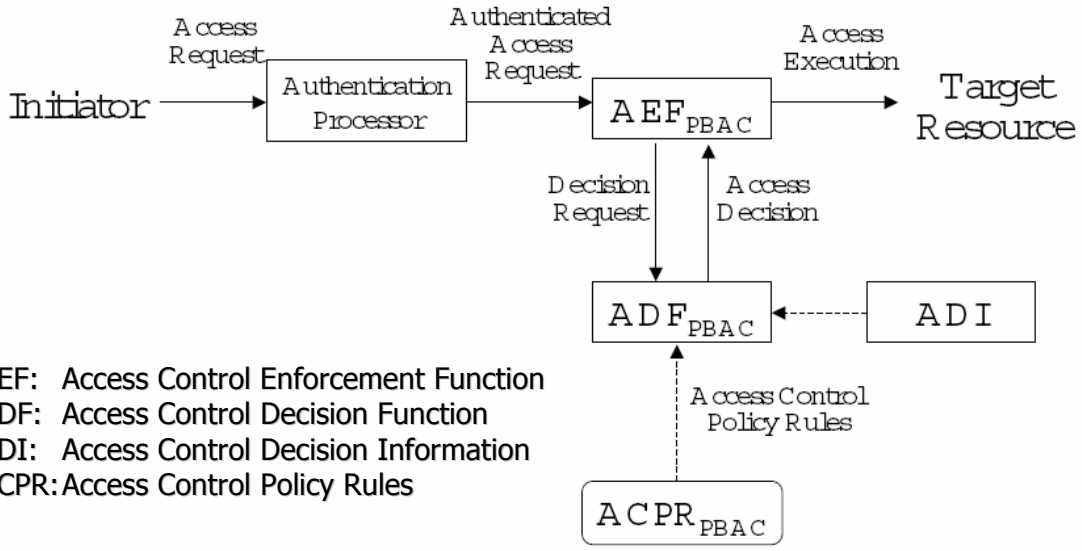
- Property propagation through hierarchies
  - Hierarchical structures greatly facilitate specification and management of the access control policy rules
  - Both permission and provisional action properties work similarly
- Typical propagation policies
  - Most specific property takes precedence
    - Broad and abstract-level policy for larger groups and narrower policies for more specific groups
  - Path traversing
    - Typical in many file systems
- Multiple hierarchies
  - e.g. group membership, object directory, role hierarchy, access modes



# Design Principles (Flexibility)

- Flexible semantics for decision-making process
  - Only positive permission and if not specified the access is denied (closed policy)
  - Only negative permission and if not specified the access is granted (open policy)
  - Both positive and negative permission and prefer positive (or negative) if both hold at the same time
- Flexible provisional authorization policies
  - Provisional actions only associated with object
  - Provisional actions only associated with subject
  - Provisional actions associated with both subject and object

# Authorization Architecture

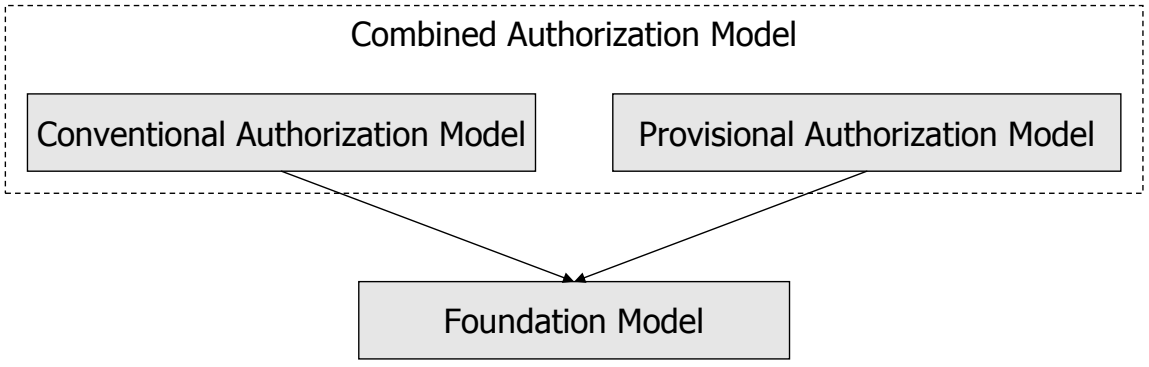



- AEF: Access Control Enforcement Function
- ADF: Access Control Decision Function
- ADI: Access Control Decision Information
- ACPR: Access Control Policy Rules



# PBAC

- Define a fundamental authorization mechanism and then formalize a provision-based access control model




- **Foundation Model:** $\mathcal{FM}$ (FDS, $\rho$, PQ, PC)

# Foundation Model Primitive Components

- **Foundation Data Set:** FDS

1. DS *is a set* $\{T_1, \ldots, T_m, S_1, \ldots, S_n\}$ *such that:* $1 \leq i \leq m$ $\langle T_i, \leq^{TR} \rangle$, $1 \leq j \leq n$ $S_j$ *is a non-tree set.* $T_i$ *and* $S_j$ *each contains finite nodes.*
2. $\mathsf{R}_{m,n}$*is a* $(m+n)$*-ary relation:* $R(T_1, \ldots, T_m, S_1, \ldots, S_n)$.

  - Each $T_i$ represent a hierarchy
    - e.g. group membership hierarchy
  - Each $S_j$ represents authorization properties
    - e.g. access modes and permission flags

- **Property Query:** $\mathsf{PQ} : \bigcup_{i=1}^{n} \mathsf{Q_i}$

$$\mathsf{Q_i} : S_1 \times \cdots \times S_{i-1}, S_{i+1} \times \cdots \times S_n$$

- **Property Collection:** $\mathsf{PC} : \bigcup_{i=1}^{n} \mathcal{P}(S_i)$



# Foundation Model Primitive Functions

- **Property Relation Function:** $\lambda$
  - A mapping $\lambda_R : T_1 \times \cdots \times T_m \rightarrow S_1 \times \cdots \times S_n$
  - $\forall t \in T_1 \times \cdots \times T_m, \forall s \in S_1 \times \cdots \times S_n, s \in \lambda_R(t) \Leftrightarrow R(t,s) \in \mathsf{R}_{m,n}$

- **Property Extraction Function:** $\sigma$
  - A mapping $\sigma_i : S_1 \times \cdots \times S_m \rightarrow S_i$

- **Property Retrieval Function:** $\rho$
  - A mapping $\rho: \mathsf{R}_{m,n} \times \mathsf{TS} \times \mathsf{PQ} \rightarrow \mathsf{PC}$

$$\rho: \sigma_i(\lambda_R(v))$$

$$V = \{v | v \text{ is a set of maximal element of } U\}$$

$$U = \{u | u \in \mathsf{TS} : \langle T_1 \times \cdots \times T_m, \leq^{LO} \rangle, \lambda_R(u) \text{ satisfies } q_i \in \mathsf{Q_i} \subset \mathsf{PQ}\}$$

$$\forall s, s \in \lambda_R(u), \forall j, j \neq i, \sigma_j(s) = \sigma_j(q_i)$$

# PBAC Authorization Model
# PBAC Data Set

- PBAC **Data Set:** PDS

1. $DS^{PBAC}$ *is a set* {ObjSet, GrpSet, ActSet, FlgSet, PvnSet, InsSet, UsrSet, ClsMap, GrpMap}.
2. $R_{2,3}^{PBAC}$ *is a 5-ary relation* $R$(ObjSet, GrpSet, ActSet, FlgSet, PvnSet),

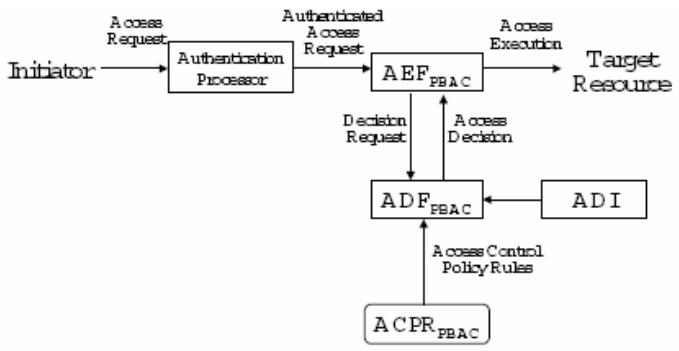
- FlgSet: {+, -}
- ClsMap: InsSet$\rightarrow \mathcal{P}$(ObjSet)
- GrpMap: UsrSet$\rightarrow \mathcal{P}$(GrpSet)



# Authorization Architecture Components

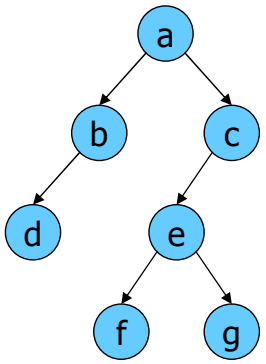
- Authorization Architecture Components: {DR, AD, ADI, $ACPR_{PBAC}$}
  - **Decision Request (DR):** (InsSet, UsrSet, ActSet)
  - **Access Decision (AD):** (FlgSet, $\mathcal{P}$(PvnSet))
  - **Access Control Decision Information:** (ClsMap, GrpMap)
  - **Access Control Policy Rules:**

    $R_{2,3}^{PBAC}$: $R$(ObjSet, GrpSet, ActSet, FlgSet, PvnSet



# Basic Authorization Policies

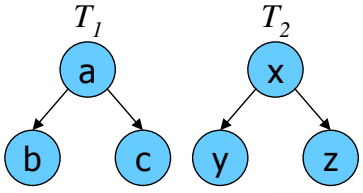
- Propagation policy
  - Most specific property takes precedence ($\pi^{MS}$)
  - Path traversing ($\pi^{PT}$)
  - Example: appropriate rule for "g":
    - $\pi^{MS}$: "g", else "e", else "c", else "a".
    - $\pi^{PT}$: "g", "e", "c", and "a"






# Basic Authorization Policies

- Priority policies for hierarchy
  - Object hierarchy takes precedence ($\pi^{OHP}$)
  - Subject hierarchy takes precedence ($\pi^{SHP}$)
  - Result of the lexicographic ordering of two hierarchies $T_1$, $T_2$ varies according to the order of input to the lexicographic order




$\langle T_1 \times T_2, \leq^{LO} \rangle$

- (a,x)
- (a,y), (a,z)
- (b,x), (c,x)
- (b,y), (b,z), (c,y), (c,z)

$\langle T_2 \times T_1, \leq^{LO} \rangle$

(a,x)
(b,x), (c,x)
(a,y), (a,z)
(b,y), (b,z), (c,y), (c,z)

# Authorization Model

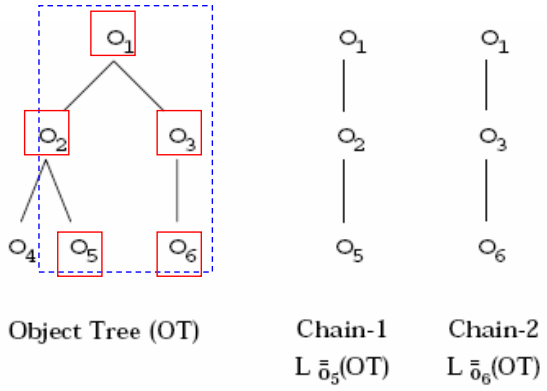
- **Authorization Model:** $\mathcal{AM}$ (PDS, $\delta^{\mathcal{AM}}$, DR, FlgSet)
- **Access Decision Function:** $\delta^{\mathcal{AM}}$: PDS × DR → FlgSet
- 1. Membership handling step
  - ClsMap, GrpMap
- 2. Propagation handling step
  - $L^{=}_{c_j}$(ObjSet) : the chain started from $c_j$ and goes to the root element of ObjSet

| Notation | Policy Name | Definition |
|---|---|---|
| $\pi^{OMS}$ | Object-round MS Policy | $QO_1 : \bigcup_{j=1}^{max_{i_q}} L^{=}_{c_j}$(ObjSet) |
| $\pi^{OPT}$ | Object-round PT Policy | $QO_i : e_i \in \bigcup_{j=1}^{max_{i_q}} L^{=}_{c_j}$(ObjSet) |
| $\pi^{SMS}$ | Subject-round MS Policy | $QS_1 : \bigcup_{k=1}^{max_{u_q}} L^{=}_{g_k}$(GrpSet) |
| $\pi^{SPT}$ | Subject-round PT Policy | $QS_i : e_i \in \bigcup_{k=1}^{max_{u_q}} L^{=}_{g_k}$(GrpSet) |



# Authorization Model

- 2. Propagation handling step – Example of chains:




If $\{O_5,O_6\}$ returned by ClsMap (step1) and $\pi^{PT}$

If $\{O_5,O_6\}$ returned by ClsMap (step1) and $\pi^{MS}$

# Authorization Model

- 3. Hierarchy priority handling step

| Notation | Policy Name | Definition |
|---|---|---|
| $\pi^{OHP}$ | Object takes precedence policy | $TS_{i,j}$:$\langle QO_i \times QS_j, \leq^{LO} \rangle$ |
| $\pi^{SHP}$ | Subject takes precedence policy | $TS_{i,j}$:$\langle QS_j \times QO_i, \leq^{LO} \rangle$ |

- 4. Rule Selection step

$$R^{PBAC}_{2,3,<i,j>} \in R(o, s, \_, \_, \_) \Leftrightarrow \forall o \in QO_i, \forall s \in QS_j$$



# Authorization Model

- 5. Property Retrieval function call step
  - Calls $\rho$ with arguments $R^{PBAC}_{2,3,<i,j>}$, $TS_{i,j}$, and $q_2$
- 6. Conflict resolution function call step
  - Calls $\gamma^{\mathcal{AM}}: \mathcal{P}(\text{FlgSet}) \rightarrow \text{FlgSet}$
    - Denials take precedence
    - Grants take precedence
    - Conflicts make an exception
    - Default denial policy
    - Default grant policy

# Provisional Authorization Model

- **Provisional Authorization Model:** $\mathcal{PAM}$
  $(\ \mathsf{PDS},\ \delta^{\mathcal{PAM}},\ \mathsf{DR}, \mathcal{P}(\mathsf{PvnSet}),\ \mathsf{FlgSet})$
- **Provisional Access Decision Function:**
  $\delta^{\mathcal{PAM}}\colon \mathsf{PDS} \times \mathsf{DR} \times \mathsf{FlgSet} \rightarrow \mathcal{P}(\mathsf{PvnSet})$

- 1-4 are identical with those described in the
- 5. Property Retrieval function call step
  - Calls $\rho$ with arguments $\mathsf{R}^{\mathsf{PBAC}}_{2,3,<\mathsf{i,j}>}$, $\mathsf{TS}_{\mathsf{i,j}}$ and $q_3$

- 6. Conflict resolution function call step
  - Calls $\gamma^{\mathcal{PAM}}\colon \mathcal{P}(\mathsf{PvnSet}) \rightarrow \mathcal{P}(\mathsf{PvnSet})$
    - rearrangement provisional actions

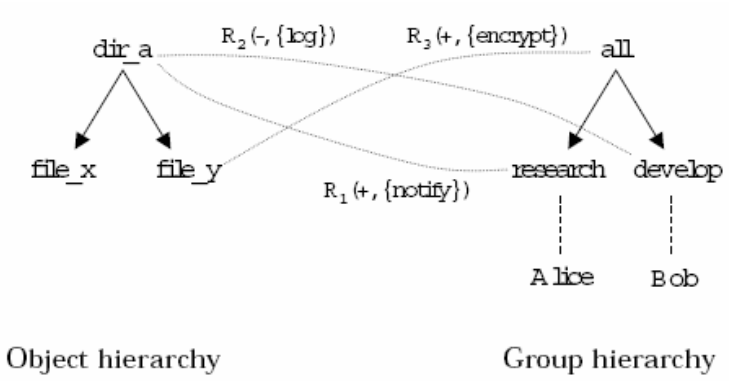


# Combined Authorization Model

- **Combined Provisional Authorization Model:** $\mathcal{CM}$
  - A sequence of $\mathcal{AM}$ , $\mathcal{PAM}$ : $(\mathcal{AM}, \mathcal{PAM}).\ \mathcal{CM}$
  - Works as if it had $\delta^{\mathcal{CM}}\colon \mathsf{PDS} \times \mathsf{DR} \rightarrow \mathsf{FlgSet} \times \mathcal{P}(\mathsf{PvnSet})$

- Semantics:
  - A decision request is allowed if the resolved permission is positive, provided all provisional actions are executed successfully.
  - A decision request is not allowed if the resolved permission is denial, however all provisional actions must still be executed.

# An Example




1. PDS: $\{DS^{PBAC}, R^{PBAC}_{2,3}\}$
   ObjTr : {dir_a, file_x, file_y | dir_a $\prec$ file_x, dir_a $\prec$ file_y}, $\langle$ ObjSet, $\leq^{TR}$
   SbjTr : {all, research, develop | all $\prec$ research, all $\prec$ develop}, $\langle$ GrpSet, $\leq$ $\rangle$
   ActSet : {read, write}, FlgSet : {+, −}, PvnSet : {log, notify, encrypt}
   InsSet : {dir_a, file_x, file_y} (InsSet is equivalent to ObjSet)
   UsrSet : {Alice, Bob}

   ClsMap(file_y) returns {file_y}, GrpMap(Alice) returns {research}
   $R^{PBAC}_{2,3}$: $\{R_1, R_2, R_3\}$
   $R_1$ : $R$(dir_a, research, read, +, {notify})
   $R_2$ : $R$(dir_a, develop, read, −, {log})
   $R_3$ : $R$(file_y, all, read, +, {encrypt})
2. DR: (file_y, Alice, read)

$\pi^{OPT}$, $\pi^{SMS}$, and $\pi^{OHP}$



# An Example

- Membership handling
- Propagation handling:
  - $QO_1$ = {file_y}, $QO_2$ = {dir_a}, $QS_1$= {all, research}
- Hierarchy priority handling:
  - $TS_{1,1}$ = $\langle QO_1 \times QS_1, \leq^{LO} \rangle$ $TS_{2,1}$ = $\langle QO_2 \times QS_1, \leq^{LO} \rangle$
- Rule Selection:
  - $R^{PBAC}_{2,3,<1,1>}$ = {R3} $R^{PBAC}_{2,3,<2,1>}$ = {R1}
- Property (permission) retrieval function call:
  - $\rho(R^{PBAC}_{2,3,<1,1>}, TS_{1,1}, read)$= "+"
  - $\rho(R^{PBAC}_{2,3,<2,1>}, TS_{2,1}, read)$= "+"
- Property (provision) retrieval function call:
  - $\delta^{PAM}$ = {encrypt, notify}
- Access Decision = ( "+", {encrypt, notify})

# Role Hierarchy

- A role hierarchy is a set $\{\text{RolSet}, \text{RolTr}, \text{RolMap}\}$
  - RolSet is a set of roles
  - RolTr: $\langle \text{RolSet}, \leq^{TR} \rangle$ indicates the role hierarchy
  - RolMap is a function: $\text{UsrSet} \rightarrow \mathcal{P}(\text{RolSet})$
    - returns the current roles which have been activated by the initiating subject
- Replacement of group hierarchy
- Addition
  - $\text{DS}^{\text{PBAC}}$: (ObjSet, GrpSet, RolSet, ActSet, FlgSet, PvnSet, InsSet, UsrSet, ClsMap, GrpMap, RolMap)
  - $\text{R}_{m,n}$: $\text{R}^{\text{RBAC}}_{3,3}$: (ObjSet, GrpSet, RolSet, ActSet, FlgSet, PvnSet)
  - More variation in priority policy:

    $\pi^{OGTP}$: TS=$\langle QO \times QS \times QR, \leq^{LO} \rangle$



# Application Examples

- Web Application
  - Data confidentiality

    **R1:** $R(\textit{credit-card}, \textit{InternetUser}, \text{write}, +, \{\text{encrypt}(\textit{KEY01})\})$
    **R2:** $R(\textit{credit-card}, \textit{Manager}, \text{read}, +, \{\text{decrypt}(\textit{KEY01})\})$
  - Non-repudiation

    **R3:** $R(\textit{purchase\_order}, \textit{Company\_A}, \text{create}, +, \{\text{verify\_user})\})$
    **R4:** $R(\textit{order\_receipt}, \textit{Company\_A}, \text{read}, +, \{\text{sign\_server}\})$
  - Charging access fees

    **R5:** $R(\textit{service\_A}, \textit{Cred\_G}, \text{execute}, +, \{\text{count}(\textit{auth\_G})\})$
    **R6:** $R(\textit{service\_A}, \textit{Cred\_G}, \text{execute}, +, \{\text{charge}(\textit{10nc}, \textit{auth\_G})\})$
  - Firewall systems

    **R7:** $R(\textit{server\_A}, \textit{aaa.bbb.ccc.ddd}, \text{connect}, +, \{\text{notify}\})$
    **R8:** $R(\textit{server\_A}, \textit{xxx.yyy.vvv.zzz}, \text{connect}, +, \{\text{notify}, \text{investigate}\})$

# Application Examples

- Traditional access control policies
  - Chinese wall

**R9:** *R(bank_A,Staff,*read,+,_*) if user has not accessed bank_B's data*
**R10:** *R(bank_B,Staff,*read,+,_*) if user has not accessed bank_A's data*

**R9':** *R(bank_A,Staff,*read,+,{confirm, verify_user, log}*) if user has not accessed bank_B's data*
**R10':** *R(bank_B,Staff,*read,+,{confirm, verify_user, log}*) if user has not accessed bank_A's data*



?

Thanks