A Role Based Administration Model For Attribute

Xin Jin, Ram Krishnan, Ravi Sandhu

SRAS, Sep 19, 2012

Outline

- Motivation
- Background
- Proposed Approach
- Discussion and Future Work

Motivation

- Attribute Based Access Control provide flexibility and scalabilities for distributed access control
  - Role based trust management (RT)[Oakland02]
  - Unified ABAC model (ABAC-alpha) [DBSEC12]
  - Attribute based Encryption (ABE) [CCS06]
  - Usage control model (UCON) [TISSEC]
  - Logical based framework for ABAC [FMSE04], etc.

- What is user attribute(credential, certificate , etc.)?
  - Name, address, country…
  - ID, clearance…

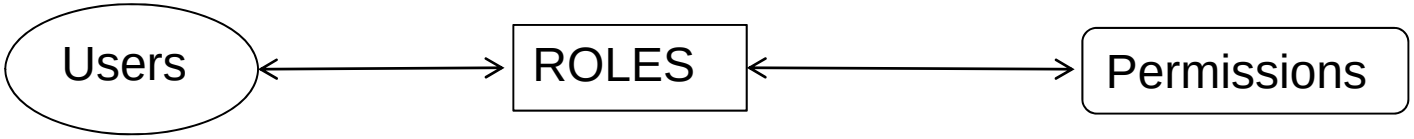
Motivation

- Much effort in how to use and store user attributes.
  - Authorization policy evaluate attribute of subject and object in the request (s, o) (ABAC-ALPHA,UCON, Logical based framework for ABAC)
  - Attribute Encryption (ABE,PKI)
  - Industry Implementation Standard (XACML, SAML, etc.)

- Not enough on user attribute management
  - Who is authorized to assign user attributes?
    - Who is authorized to assign value for Salary(Alice)?
    - Can Bob say that Salary(Alice)=3000?
  - How should permissions be assigned to administrators?
    - Manually one by one? No!

Motivation

- Effective user attribute administration model needed
  - Ease of Administration one of desired features
- Role based Access Control



E.g. ProgramManager

E.g. Assign Alice's salary to 3000, 6000 9000

- Good candidate for user attribute Administration
  - Proved ease of administration
  - Efficient safety analysis
  - Sizable literature, etc.

Background

- User role assignment model (URA97)

Can_assign(administration role, prerequisite role, role range notation)

Example: $\Pr ogrammanager \quad Employee \wedge \overline{projectmanager} \quad \{prj1leader, proj2leader\}$

Can_revoke(administration role, role range notation)

Example: $\Pr ogrammanager \quad \{prj1leader, proj2leader\}$

- User attributes
  - Atomic: User can only have one value for this kind of attribute. E.g. user id, clearance, etc.
  - Set: User can be assigned multiple values. E.g. role, phonenumber, etc.
  - Range(ua), attType(ua)

Proposed Approach

- Generalize Role as one of user attributes
  - Difference between role and user attribute
    - Role represents permissions while attributes do not

      (Alice's *role* is Program Chair VS Alice's *age* is 23)
    - Role has hierarchy while attribute may not

      (Role: CEO is senior role of employee. Attribute: Address)
  - Difference between atomic user attributes and set-valued user attributes
    - Assign new value to atomic user attribute automatically remove old value
    - Add new values to set-valued attribute does not guarantee permissions to delete existing attribute values.

Proposed Approach

- Permissions to each administrative role
  - Add value to set-valued user attributes
  - Delete value from set-valued attributes
  - Assign value to atomic-valued attributes
- Specify the three permission sets for each role
  - Add value to set-valued user attributes
  - Delete value from set-valued attributes
  - Assign value to atomic-valued attributes

Proposed Approach(GURA0)

- Add values to set-valued user attributes:

$$\forall\, sua \in SUA.\, can\_add_{sua} \subseteq AR \times EXPR(sua) \times 2^{Range(sua)}$$

AR: set of administrative roles
SUA: set of set-valued user attributes
EXPR(sua): logical expression composed of user attribute *sua*

Proposed Approach(GURA0)

- Delete values from set-valued user attributes:

$$\forall\, sua \in SUA.\, can\_delete_{sua} \subseteq AR \times EXPR(sua) \times 2^{Range(sua)}$$

AR: set of administrative roles
SUA: set of set-valued user attribute
EXPR(sua): logical expression composed of user attribute *sua*

Proposed Approach(GURA0)

- Assign value to atomic-valued user attributes:

$$\forall sua \in AUA.can\_assign_{aua} \subseteq AR \times EXPR(aua) \times 2^{Range(aua)}$$

AR: set of administrative roles
AUA: set of atomic-valued user attributes
EXPR(aua): logical expression composed of user attribute *aua*

Language

- Common Expression Language

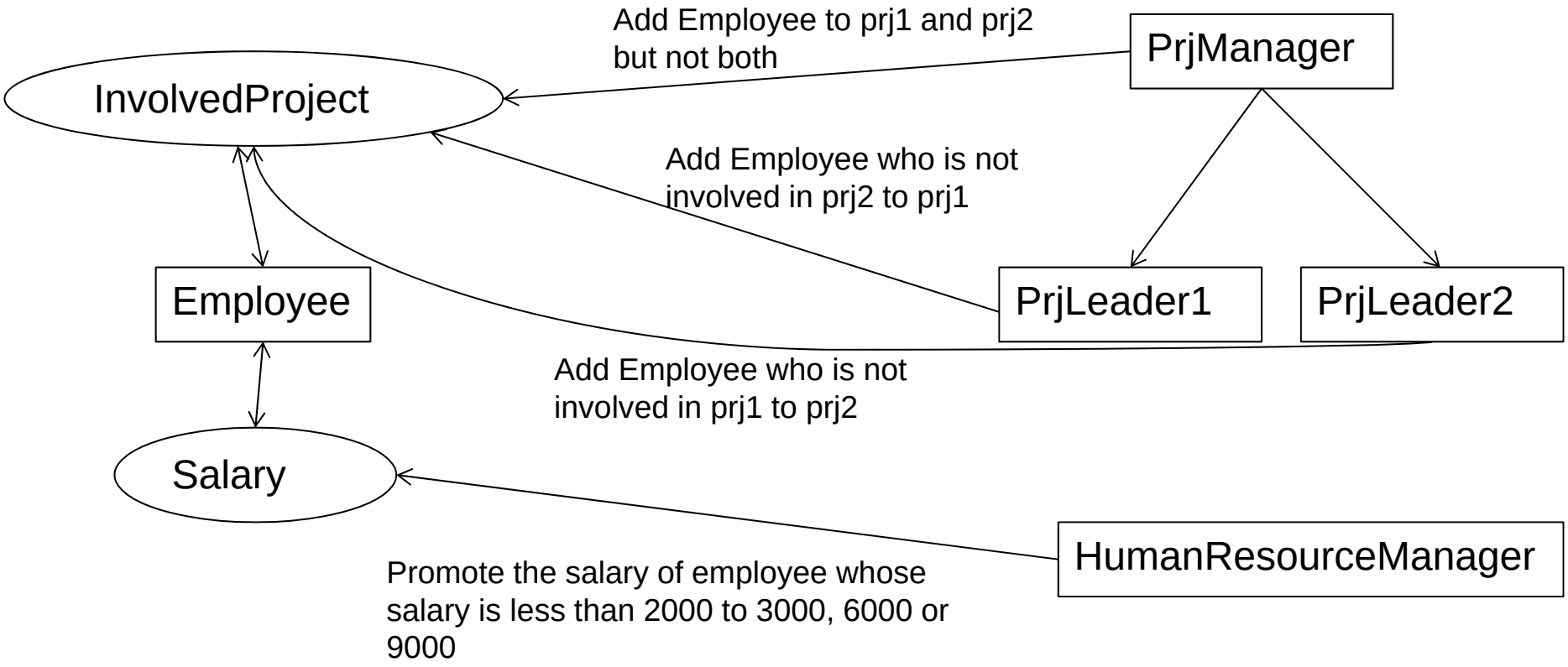
$$\begin{aligned}\varphi ::= {} & \varphi \wedge \varphi \mid \varphi \vee \varphi \mid (\ \varphi\ ) \mid \neg\ \varphi \mid \exists\ \mathrm{x} \in \mathrm{set}.\varphi \mid \\ & \forall\ \mathrm{x} \in \mathrm{set}.\varphi \mid \mathrm{set\ setcompare\ set} \mid \mathrm{atomic} \in \mathrm{set} \mid \\ & \mathrm{atomic} \notin \mathrm{set} \mid \mathrm{atomic\ atomiccompare\ atomic} \\ \mathrm{atomiccompare} ::= {} & < \mid = \mid \leq \mid \neq \\ \mathrm{setcompare} ::= {} & \subset \mid \subseteq \mid \not\subseteq\end{aligned}$$

- Example: set-valued attributes in GURA0
  EXPR(SUA) is specified using an instance of the above language where

$$\begin{aligned}\mathrm{set} ::= {} & \mathrm{sua(u)} \mid \mathrm{constantSet} \\ \mathrm{atomic} ::= {} & \mathrm{constantAtomic}\end{aligned}$$

Example

Users And User Attributes **Administrative Roles**



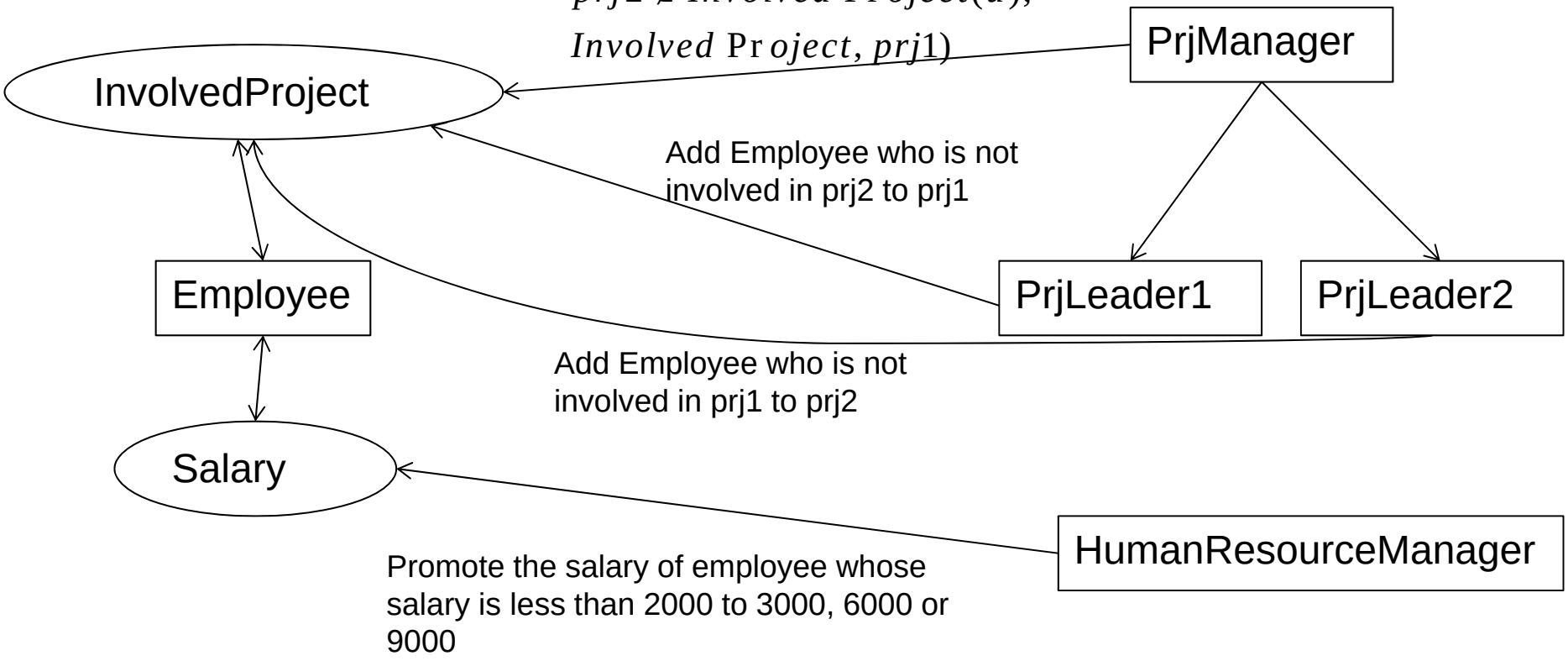
Example

Users And User Attribute

$$can\_add(\Pr jManager, prj1 \notin Involved\ \Pr oject(u), involved\ \Pr oject, prj2)$$

$$can\_add(\Pr jManager, prj2 \notin Involved\ \Pr oject(u), Involved\ \Pr oject, prj1)$$

Administrative Roles

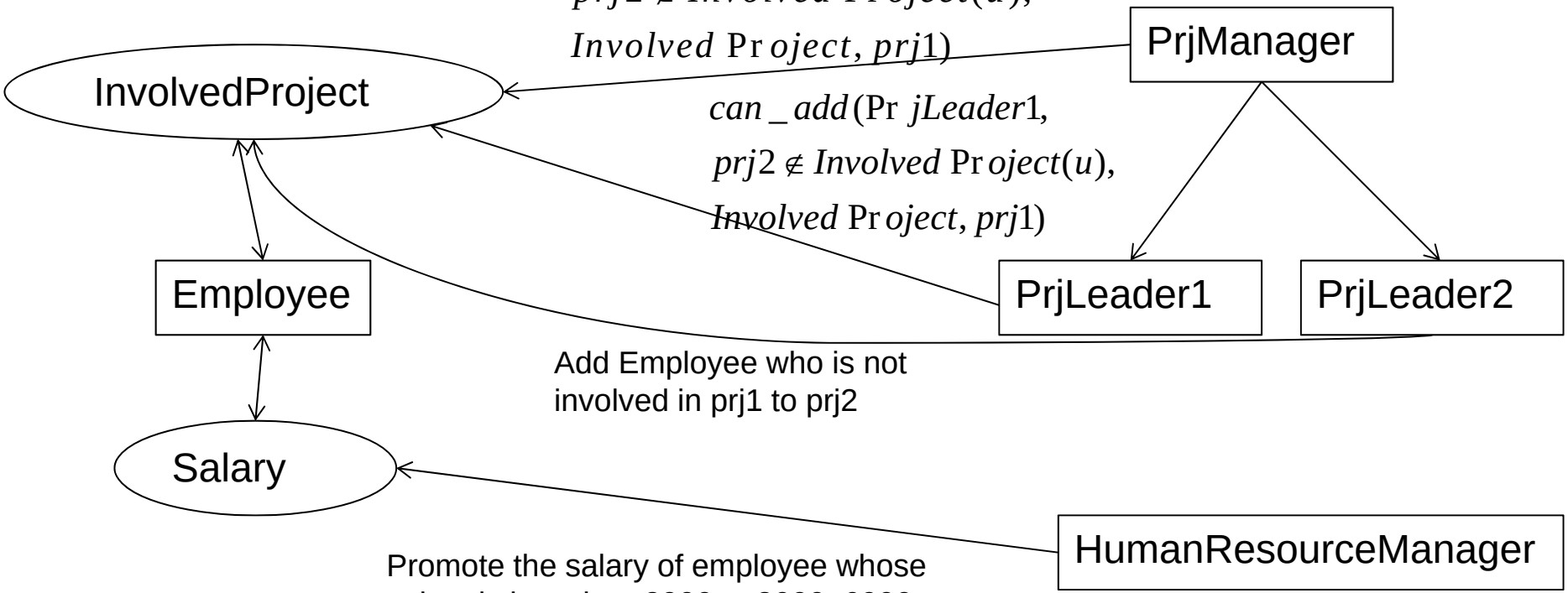


Example

Users And User Attribute

Administrative Roles

$can\_add(\text{Pr}\ jManager,$
$prj1 \notin Involved\ \text{Pr}oject(u),$
$involved\ \text{Pr}oject, prj2)$
$can\_add(\text{Pr}\ jManager,$
$prj2 \notin Involved\ \text{Pr}oject(u),$
$Involved\ \text{Pr}oject, prj1)$

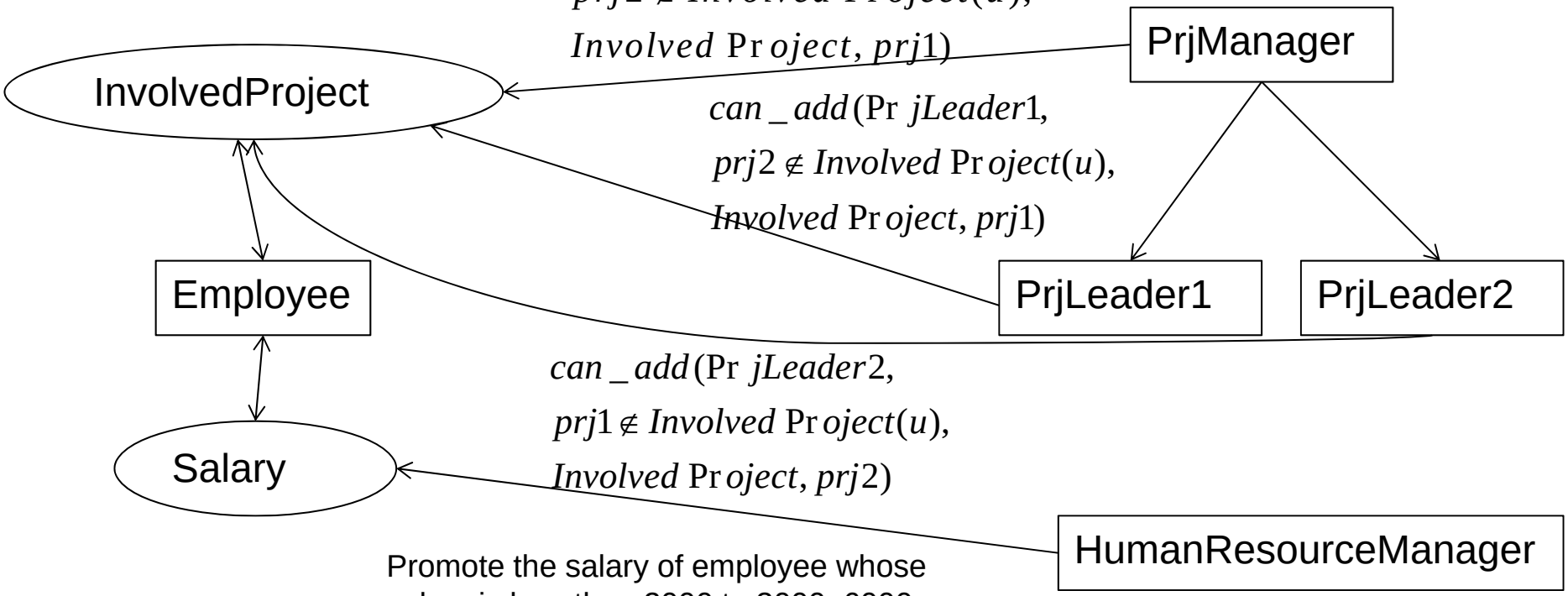


Example

Users And User Attribute

Administrative Roles

$can\_add(\text{Pr}\ jManager,$
$prj1 \notin Involved\ \text{Pr}oject(u),$
$involved\ \text{Pr}oject, prj2)$
$can\_add(\text{Pr}\ jManager,$
$prj2 \notin Involved\ \text{Pr}oject(u),$
$Involved\ \text{Pr}oject, prj1)$

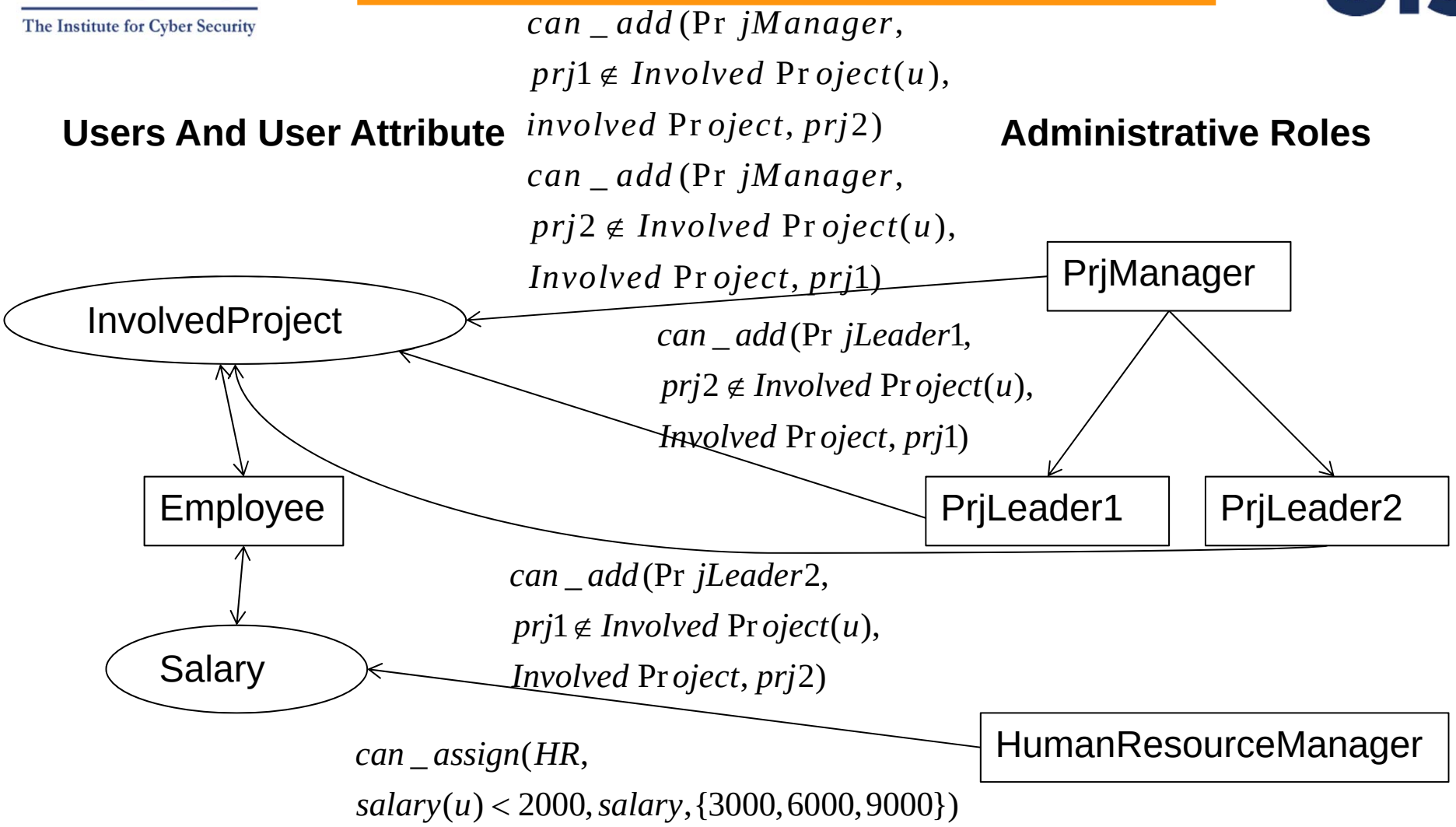


Promote the salary of employee whose salary is less than 2000 to 3000, 6000 or 9000

Example

Users And User Attribute

Administrative Roles



Proposed Approach(GURA1)

- Limited Expressive Power
  - Expression(ua) ONLY composed using the user attribute in context is not enough.

    Example: PrjLeader1 is only authorized to add employee whose experience with Java is more than 3 years and whose skills include C++.

- Least Privilege is not achieved
  - Expression(ua) CAN NOT restrict users to the least set. Need other user attributes to compose the expression.

Proposed Approach(GURA1)

- Further Generalization

$$\forall sua \in SUA.can\_add_{sua} \subseteq AR \times EXPR(sua) \times 2^{Range(sua)}$$

↓

$$\forall sua \in SUA.can\_add_{sua} \subseteq AR \times EXPR(UA) \times 2^{Range(sua)}$$

↓

EXPR(UA): logical expression composed of all user attribute *ua in UA*

Proposed Approach(GURA1)

- Further Generalization for GURA1:

$$\forall sua \in SUA.can\_delete_{sua} \subseteq AR \times EXPR(UA) \times 2^{Range(sua)}$$

$$\forall aua \in AUA.can\_assign_{aua} \subseteq AR \times EXPR(UA) \times 2^{Range(aua)}$$

Example

- Example using GURA1

Table 5: *Examples*

can_add relation for attribute involvedprj:
(prj1leader, prj2 $\notin$ involvedprj(u) $\wedge$ trainingpassed(u) = true $\wedge$ clearance(u) $>$ S $\wedge$ C $\in$ skills(u), {prj1})

(prj2leader, prj1 $\notin$ involvedprj(u) $\wedge$ trainingpassed(u) = true $\wedge$ clearance(u) $>$ S $\wedge$ C $\in$ skills(u), {prj2})

can_add relation for attribute skill:
(secretary, NULL, {C ,C++, Java})

can_delete relation for attribute involvedprj:
(prj1leader, prj1 $\in$ involvedprj(u), {prj1})
(prj2leader, prj2 $\in$ involvedprj(u), {prj2})

can_delete relation for attribute skill:
(secretary, NULL, {C, C++, Java})

can_assign relation for attribute trainingpassed:
(trainingmanager, NULL, {true, false})

can_assign relation for attribute clearance:
(humanmanager, NULL, {TS, S, C, U})

Discussions and Future Work

- Advantages
  - Advantages of RBAC inherited
- Limitations
  - Awkward for distributed administration of user attributes
  - For fine-grained user attribute administration, many roles with slight different set of permissions need to defined.
- Future Work
  - Delegation
  - Attribute based user attribute management

Any Questions?