Strategies for Incorporating Delegation into Attribute-Based Access Control (ABAC)

Daniel Servos
dservos5@uwo.ca

Sylvia L. Osborn
sylvia@csd.uwo.ca

Western
UNIVERSITY · CANADA
Department of Computer Science

The 9th International Symposium on Foundations & Practice of Security, October 2016

Talk Outline

1. Outline
2. Background
3. Strategies for Incorporating Delegation
    - Attribute Delegation
    - Group Membership Delegation
    - Permission Delegation
4. Qualitative Evaluation
5. Conclusions and Future Work

ABAC Background

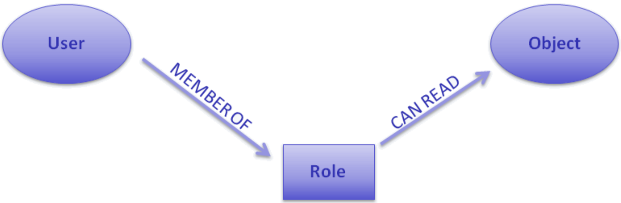
ABAC Background

Role-Based Access Control (RBAC)

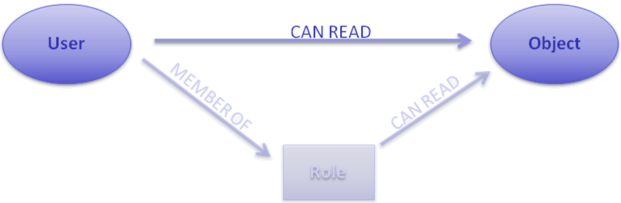
User

Object

Role-Based Access Control (RBAC)



Role-Based Access Control (RBAC)



ABAC Background

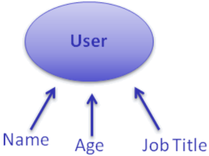
Attribute-Based Access Control (ABAC)

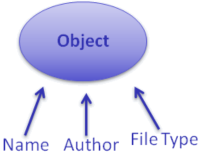
User

Object

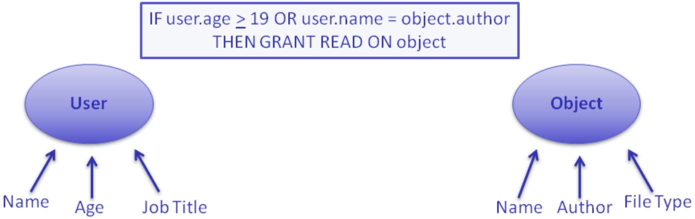
ABAC Background

Attribute-Based Access Control (ABAC)



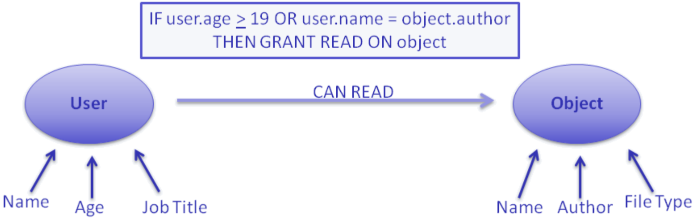


Attribute-Based Access Control (ABAC)



IF user.age $\geq$ 19 OR user.name = object.author
THEN GRANT READ ON object

Attribute-Based Access Control (ABAC)



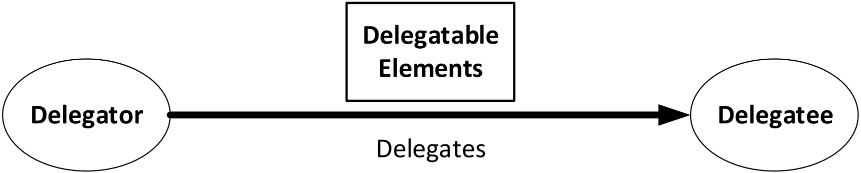
IF user.age ≥ 19 OR user.name = object.author
THEN GRANT READ ON object

Delegation

Delegation

Key components of delegation:

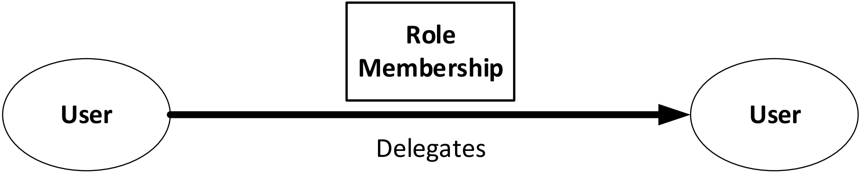
- Delegators
- Delegatable Access Control Elements
- Delegatees



Delegation Components

In RBAC:

- Delegators:
  - Users
- Delegatable Access Control Elements:
  - Role Membership
  - Permissions (via temporary role)
- Delegatees:
  - Users



Delegation Components

In ABAC:

- Delegators:
- Delegatable Access Control Elements:
- Delegatees:

Delegation Components

In ABAC:

- Delegators:
  - Users
  - Groups
- Delegatable Access Control Elements:
- Delegatees:
  - Users
  - Groups

Delegation Components

In ABAC:

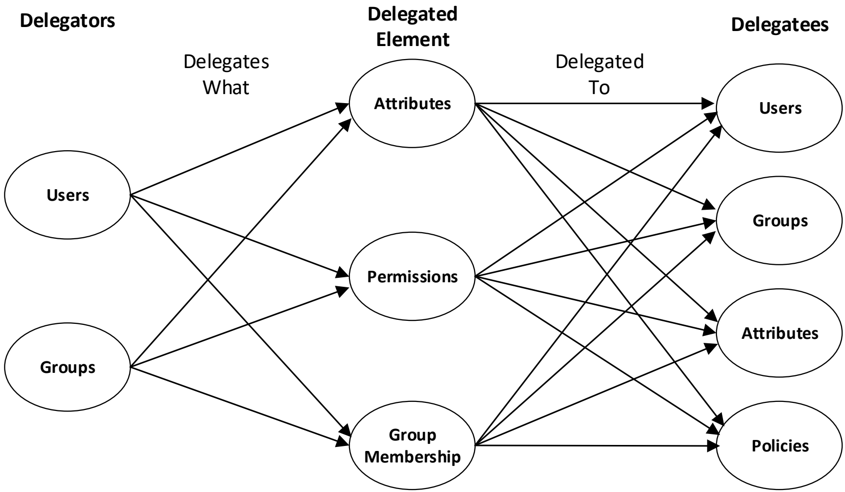
- Delegators:
  - Users
  - Groups
- Delegatable Access Control Elements:
- Delegatees:
  - Users
  - Groups
  - Attributes
  - Policies

Delegation Components

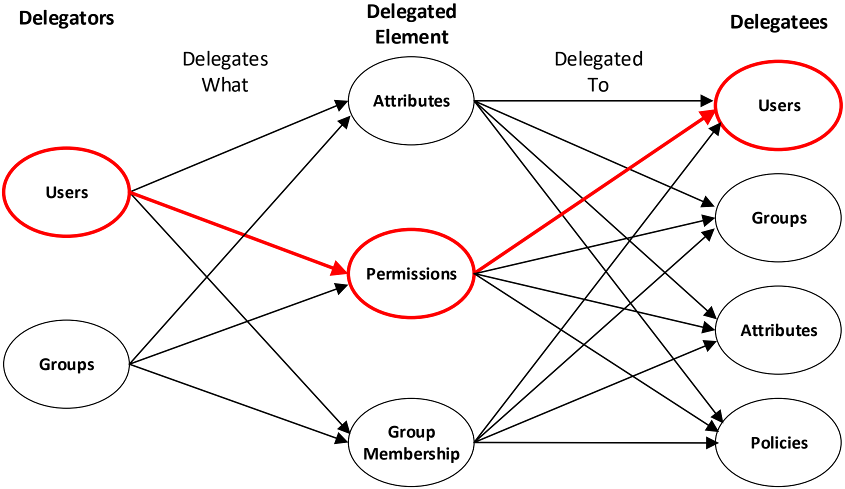
In ABAC:

- Delegators:
  - Users
  - Groups
- Delegatable Access Control Elements:
  - Attributes
  - Permissions
  - Group Membership
- Delegatees:
  - Users
  - Groups
  - Attributes
  - Policies

Strategy Graph

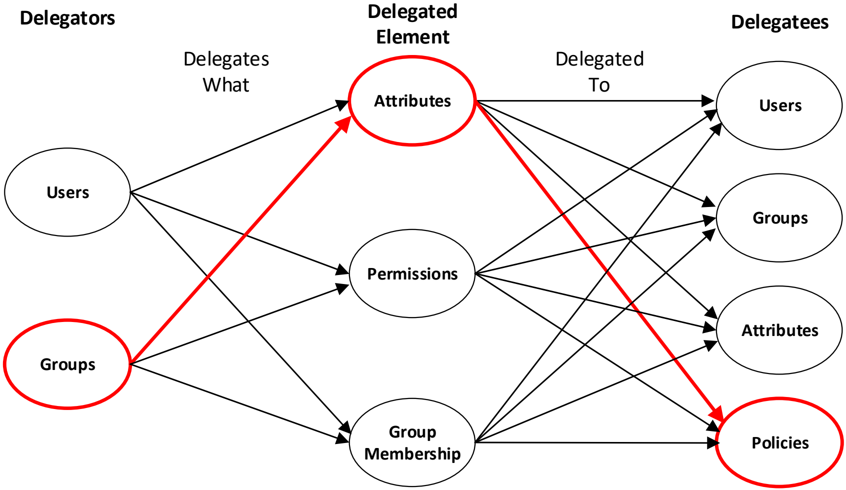


Strategy Graph



User-to-User Permission Delegation

Strategy Graph



Group-to-Policy Attribute Delegation

Delegation Strategies

Delegation Strategy Families

Strategy Name	Delegator	Delegated Element	Delegatee
Attribute Delegation			
User-to-User Attribute Delegation	User	Attribute Set	User
User-to-Group Attribute Delegation	User	Attribute Set	Group
Group-to-Group Attribute Delegation	Group	Attribute Set	Group
Group-to-User Attribute Delegation	Group	Attribute Set	User
User-to-Attribute Attribute Delegation	User	Attribute Set	Attribute
Group-to-Attribute Attribute Delegation	Group	Attribute Set	Attribute
User-to-Policy Attribute Delegation	User	Attribute Set	Policy
Group-to-Policy Attribute Delegation	Group	Attribute Set	Policy
Group Membership Delegation			
User-to-User Membership Delegation	User	Group Membership	User
Group-to-User Membership Delegation	Group	Group Membership	User
Group-to-Group Membership Delegation	Group	Group Membership	Group
User-to-Group Membership Delegation	User	Group Membership	Group
User-to-Attribute Membership Delegation	User	Group Membership	Attribute
Group-to-Attribute Membership Delegation	Group	Group Membership	Attribute
User-to-Policy Membership Delegation	User	Group Membership	Policy
Group-to-Policy Membership Delegation	Group	Group Membership	Policy
Permission Delegation			
User-to-User Permission Delegation	User	Permission Set	User
User-to-Group Permission Delegation	User	Permission Set	Group

Delegation Strategy Families

Strategy Name	Delegator	Delegated Element	Delegatee
Attribute Delegation			
User-to-User Attribute Delegation	User	Attribute Set	User
User-to-Group Attribute Delegation	User	Attribute Set	Group
Group-to-Group Attribute Delegation	Group	Attribute Set	Group
Group-to-User Attribute Delegation	Group	Attribute Set	User
User-to-Attribute Attribute Delegation	User	Attribute Set	Attribute
Group-to-Attribute Attribute Delegation	Group	Attribute Set	Attribute
User-to-Policy Attribute Delegation	User	Attribute Set	Policy
Group-to-Policy Attribute Delegation	Group	Attribute Set	Policy
Group Membership Delegation			
User-to-User Membership Delegation	User	Group Membership	User
Group-to-User Membership Delegation	Group	Group Membership	User
Group-to-Group Membership Delegation	Group	Group Membership	Group
User-to-Group Membership Delegation	User	Group Membership	Group
User-to-Attribute Membership Delegation	User	Group Membership	Attribute
Group-to-Attribute Membership Delegation	Group	Group Membership	Attribute
User-to-Policy Membership Delegation	User	Group Membership	Policy
Group-to-Policy Membership Delegation	Group	Group Membership	Policy
Permission Delegation			
User-to-User Permission Delegation	User	Permission Set	User
User-to-Group Permission Delegation	User	Permission Set	Group
Group-to-User Permission Delegation	Group	Permission Set	User
Group-to-Group Permission Delegation	Group	Permission Set	Group
User-to-Attribute Permission Delegation	User	Permission Set	Attribute

Strategy Name	Delegator	Delegated Element	Delegatee
Attribute Delegation			
User-to-User Attribute Delegation	User	Attribute Set	User
User-to-Group Attribute Delegation	User	Attribute Set	Group
Group-to-Group Attribute Delegation	Group	Attribute Set	Group
Group-to-User Attribute Delegation	Group	Attribute Set	User
User-to-Attribute Attribute Delegation	User	Attribute Set	Attribute
Group-to-Attribute Attribute Delegation	Group	Attribute Set	Attribute
User-to-Policy Attribute Delegation	User	Attribute Set	Policy
Group-to-Policy Attribute Delegation	Group	Attribute Set	Policy
Group Membership Delegation			
User-to-User Membership Delegation	User	Group Membership	User
Group-to-User Membership Delegation	Group	Group Membership	User
Group-to-Group Membership Delegation	Group	Group Membership	Group
User-to-Group Membership Delegation	User	Group Membership	Group
User-to-Attribute Membership Delegation	User	Group Membership	Attribute
Group-to-Attribute Membership Delegation	Group	Group Membership	Attribute
User-to-Policy Membership Delegation	User	Group Membership	Policy
Group-to-Policy Membership Delegation	Group	Group Membership	Policy
Permission Delegation			
User-to-User Permission Delegation	User	Permission Set	User
User-to-Group Permission Delegation	User	Permission Set	Group
Group-to-User Permission Delegation	Group	Permission Set	User
Group-to-Group Permission Delegation	Group	Permission Set	Group
User-to-Attribute Permission Delegation	User	Permission Set	Attribute
Group-to-Attribute Permission Delegation	User	Permission Set	Attribute
User-to-Policy Permission Delegation	User	Permission Set	Policy
Group-to-Policy Permission Delegation	Group	Permission Set	Policy

User-to-User Attribute Delegation	User	Attribute Set	User
User-to-Group Attribute Delegation	User	Attribute Set	Group
Group-to-Group Attribute Delegation	Group	Attribute Set	Group
Group-to-User Attribute Delegation	Group	Attribute Set	User
User-to-Attribute Attribute Delegation	User	Attribute Set	Attribute
Group-to-Attribute Attribute Delegation	Group	Attribute Set	Attribute
User-to-Policy Attribute Delegation	User	Attribute Set	Policy
Group-to-Policy Attribute Delegation	Group	Attribute Set	Policy
Group Membership Delegation			
User-to-User Membership Delegation	User	Group Membership	User
Group-to-User Membership Delegation	Group	Group Membership	User
Group-to-Group Membership Delegation	Group	Group Membership	Group
User-to-Group Membership Delegation	User	Group Membership	Group
User-to-Attribute Membership Delegation	User	Group Membership	Attribute
Group-to-Attribute Membership Delegation	Group	Group Membership	Attribute
User-to-Policy Membership Delegation	User	Group Membership	Policy
Group-to-Policy Membership Delegation	Group	Group Membership	Policy
Permission Delegation			
User-to-User Permission Delegation	User	Permission Set	User
User-to-Group Permission Delegation	User	Permission Set	Group
Group-to-User Permission Delegation	Group	Permission Set	User
Group-to-Group Permission Delegation	Group	Permission Set	Group
User-to-Attribute Permission Delegation	User	Permission Set	Attribute
Group-to-Attribute Permission Delegation	User	Permission Set	Attribute
User-to-Policy Permission Delegation	User	Permission Set	Policy
Group-to-Policy Permission Delegation	Group	Permission Set	Policy

Attribute Delegation

- Delegatees are delegated a subset of the delegator's attribute set (chosen by the delegator).
- Delegated attributes are merged with the delegatee's **directly** assigned attributes.
- Merged (**effective**) attribute set is treated as the delegatee's set for the purposes of policy evaluation.

Attribute Delegation: Examples

```
direct(Alice) =
 {(year, {4}),
  (role, {“undergrad”}),
  (department, {“CompSci”})}
```

Alice

Dave

```
direct(Dave) =
  {(role, {“ProspectiveStudent”})}
```

Attribute Delegation: Examples

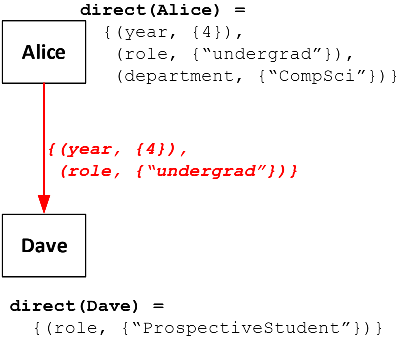
Example 1

Alice wants to delegate attributes to Dave such that he satisfies the policy:

$$\textit{role} = \textit{“undergrad”}\ \textit{AND}\ \textit{year} \geq 2$$

```
direct(Dave) =
  {(role, {"ProspectiveStudent"})}
```

Attribute Delegation: Examples



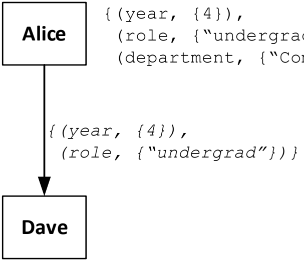
```
direct(Alice) =
  {(year, {4}),
   (role, {“undergrad”}),
   (department, {“CompSci”})}
```

{(year, {4}),
(role, {“undergrad”})}

```
direct(Dave) =
  {(role, {“ProspectiveStudent”})}
```

Attribute Delegation: Examples

```
direct(Alice) =
  {(year, {4}),
   (role, {"undergrad"}),
   (department, {"CompSci"})}
```



Alice

{(year, {4}), (role, {"undergrad"})}

Dave

```
direct(Dave) =
  {(role, {"ProspectiveStudent"})}
```

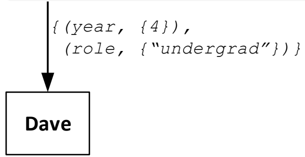
```
effective(Dave) =
  {(role, {"ProspectiveStudent",
           "undergrad"}),
   (year, {4})}
```

Attribute Delegation: Examples

Example 1

Alice wants to delegate attributes to Dave such that he satisfies the policy:

$$\textit{role} = \text{“\textit{undergrad}”}\ \textit{AND}\ \textit{year} \geq 2$$



```
direct(Dave) =
  {(role, {"ProspectiveStudent"})}
```

```
effective(Dave) =
  {(role, {"ProspectiveStudent",
           "undergrad")},
   (year, {4})}
```

Attribute Delegation: Examples

Alice

```
direct(Alice) =
 {(year, {4}),
  (role, {“undergrad”}),
  (department, {“CompSci”})}
```

Bob

```
direct(Bob) =
 {(role, {“faculty”}),
  (department, {“SoftEng”})}
```

Charlie

```
direct(Charlie) =
  {(role, {“grad”}),
   (department, {“SoftEng”})}
```

Attribute Delegation: Examples

Example 2

Alice wants to delegate attributes to Charlie such that he satisfies the policy:

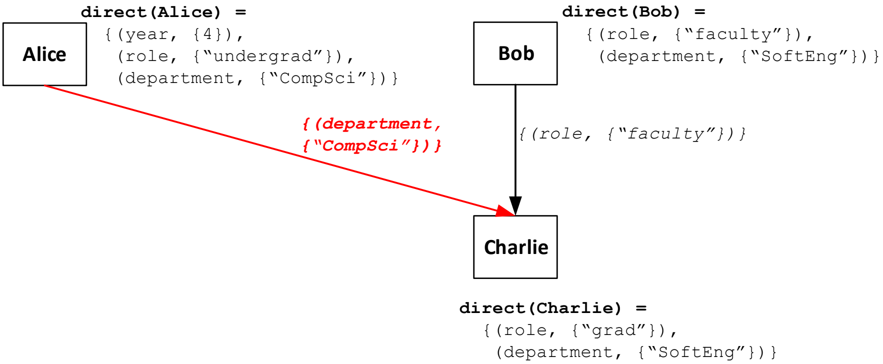
$$role\ IN\ \{\text{"}undergrad\text{"},\ \text{"}grad\text{"}\}\ AND\ department = \text{"}CompSci\text{"}$$

At the same time, Bob wants to delegate attributes to Charlie such that he satisfies the policy:

$$role = \text{"}faculty\text{"}\ AND\ department = \text{"}SoftEng\text{"}$$

```
direct(Charlie) =
  {(role, {"grad"}),
   (department, {"SoftEng"})}
```

Attribute Delegation: Examples



direct(Alice) =
{(year, {4}),
(role, {"undergrad"}),
(department, {"CompSci"})}

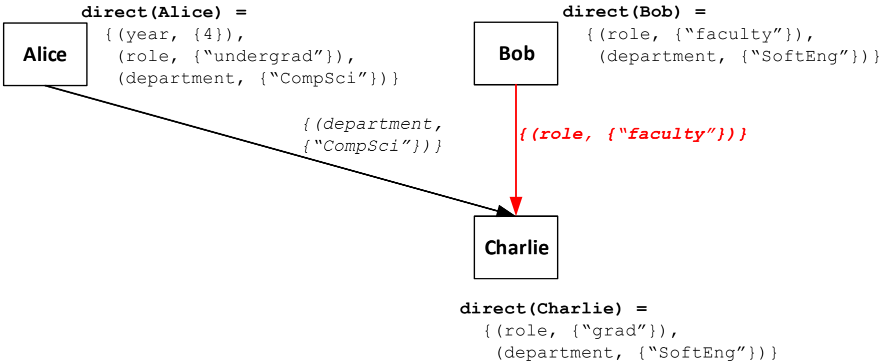
direct(Bob) =
{(role, {"faculty"}),
(department, {"SoftEng"})}

{(department,
{"CompSci"})}

{(role, {"faculty"})}

direct(Charlie) =
{(role, {"grad"}),
(department, {"SoftEng"})}

Attribute Delegation: Examples

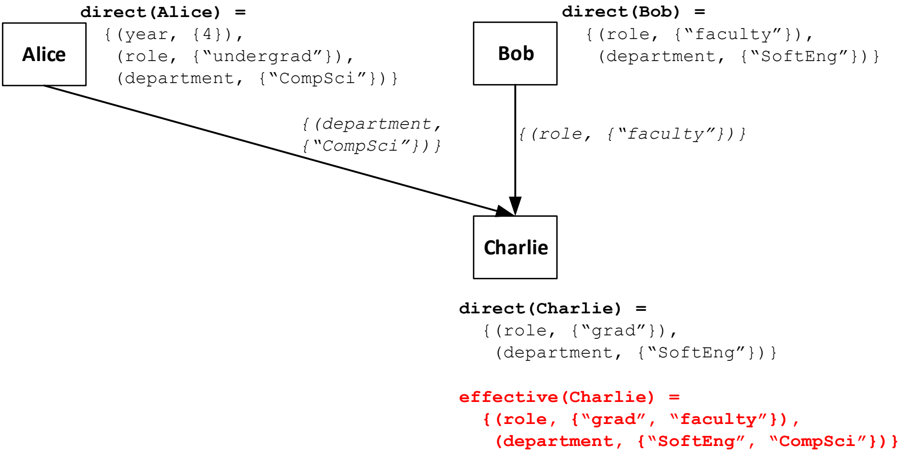


direct(Alice) =
{(year, {4}),
(role, {"undergrad"}),
(department, {"CompSci"})}

direct(Bob) =
{(role, {"faculty"}),
(department, {"SoftEng"})}

direct(Charlie) =
{(role, {"grad"}),
(department, {"SoftEng"})}

Attribute Delegation: Examples



```
direct(Alice) =
 {(year, {4}),
  (role, {"undergrad"}),
  (department, {"CompSci"})}
```

```
direct(Bob) =
 {(role, {"faculty"}),
  (department, {"SoftEng"})}
```

```
direct(Charlie) =
 {(role, {"grad"}),
  (department, {"SoftEng"})}
```

```
effective(Charlie) =
 {(role, {"grad", "faculty"}),
  (department, {"SoftEng", "CompSci"})}
```

Attribute Delegation: Examples

Example 2

Alice wants to delegate attributes to Charlie such that he satisfies the policy:

$$role\ IN\ \{\text{"}undergrad\text{"},\ \text{"}grad\text{"}\}\ AND\ department = \text{"}CompSci\text{"}$$

At the same time, Bob wants to delegate attributes to Charlie such that he satisfies the policy:

$$role = \text{"}faculty\text{"}\ AND\ department = \text{"}SoftEng\text{"}$$

```
direct(Charlie) =
  {(role, {"grad"}),
   (department, {"SoftEng"})}

effective(Charlie) =
  {(role, {"grad", "faculty"}),
   (department, {"SoftEng", "CompSci"})}
```

Attribute Delegation: Problems/Benefits

Advantages of Attribute Delegation:

- Simple, easy to implement
- Works in distributed/SSO systems
- No extra computations/considerations at PEP or PDP

Attribute Delegation: Problems/Benefits

Issues with Attribute Delegation:

Attribute Delegation: Problems/Benefits

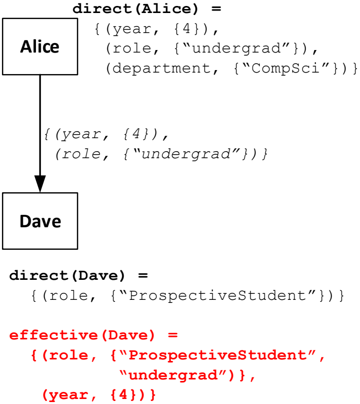
Issues with Attribute Delegation:

- Conflicting policy evaluations

Attribute Delegation: Problems/Benefits

Issues with Attribute Delegation:

- Conflicting policy evaluations



```
direct(Alice) =
  {(year, {4}),
   (role, {"undergrad"}),
   (department, {"CompSci"})}
```

Alice

{(year, {4}),
(role, {"undergrad"})}

Dave

```
direct(Dave) =
  {(role, {"ProspectiveStudent"})}
```

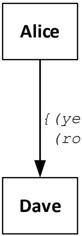
```
effective(Dave) =
  {(role, {"ProspectiveStudent",
           "undergrad")},
   (year, {4})}
```

Attribute Delegation: Problems/Benefits

Issues with Attribute Delegation:

- Conflicting policy evaluations

```
direct(Alice) =
 {(year, {4}),
  (role, {“undergrad”}),
  (department, {“CompSci”})}
```



```
direct(Dave) =
  {(role, {“ProspectiveStudent”})}

effective(Dave) =
  {(role, {“ProspectiveStudent”,
           “undergrad”)},
   (year, {4})}
```

role $\neq$ *“ProspectiveStudent”*

Attribute Delegation: Problems/Benefits

Issues with Attribute Delegation:

- Conflicting policy evaluations

Attribute Delegation: Problems/Benefits

Issues with Attribute Delegation:

- Conflicting policy evaluations
- User collusion

Attribute Delegation: Problems/Benefits

Issues with Attribute Delegation:

- Conflicting policy evaluations
- User collusion

Oscar

```
direct(Oscar) =
  {(year, {4}),
   (department, {“CompSci”})}
```

Mallory

```
direct(Mallory) =
  {(year, {1}),
   (department, {“SoftEng”})}
```

Example 3

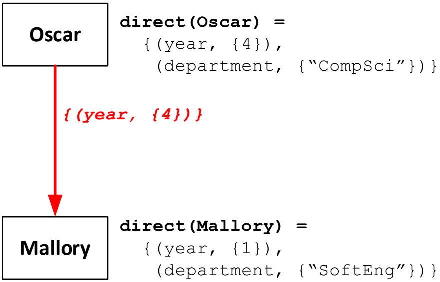
Oscar and Mallory want to collude to pass the policy:

$$year > 2 \ AND \ department = \text{“}SoftEng\text{”}$$

Attribute Delegation: Problems/Benefits

Issues with Attribute Delegation:

- Conflicting policy evaluations
- User collusion



```
direct(Oscar) =
  {(year, {4}),
   (department, {"CompSci"})}
```

```
direct(Mallory) =
  {(year, {1}),
   (department, {"SoftEng"})}
```

Example 3

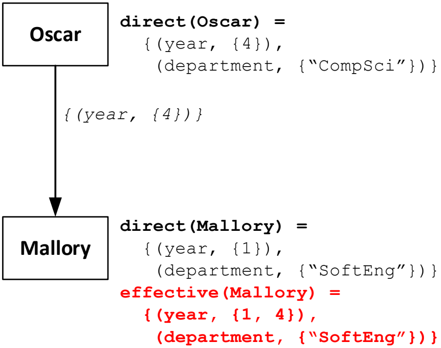
Oscar and Mallory want to collude to pass the policy:

$$\textit{year} > 2 \textit{ AND department} = \textit{“SoftEng”}$$

Attribute Delegation: Problems/Benefits

Issues with Attribute Delegation:

- Conflicting policy evaluations
- User collusion



```
direct(Oscar) =
  {(year, {4}),
   (department, {"CompSci"})}
```

```
direct(Mallory) =
  {(year, {1}),
   (department, {"SoftEng"})}
effective(Mallory) =
  {(year, {1, 4}),
   (department, {"SoftEng"})}
```

Example 3

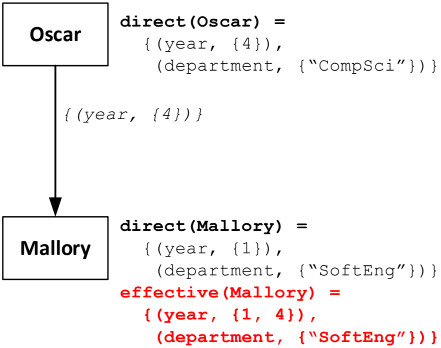
Oscar and Mallory want to collude to pass the policy:

$$year > 2\ AND\ department = \text{“SoftEng”}$$

Attribute Delegation: Problems/Benefits

Issues with Attribute Delegation:

- Conflicting policy evaluations
- User collusion



Oscar

```
direct(Oscar) =
  {(year, {4}),
   (department, {"CompSci"})}
```

{(year, {4})}

Mallory

```
direct(Mallory) =
  {(year, {1}),
   (department, {"SoftEng"})}
effective(Mallory) =
  {(year, {1, 4}),
   (department, {"SoftEng"})}
```

Example 3

Oscar and Mallory want to collude to pass the policy:

$$year > 2\ AND\ department = \text{“SoftEng”}$$

Attribute Delegation: Problems/Benefits

Issues with Attribute Delegation:

- Conflicting policy evaluations
- User collusion
- Effective attribute not descriptive of the delegatee

Attribute Delegation: Problems/Benefits

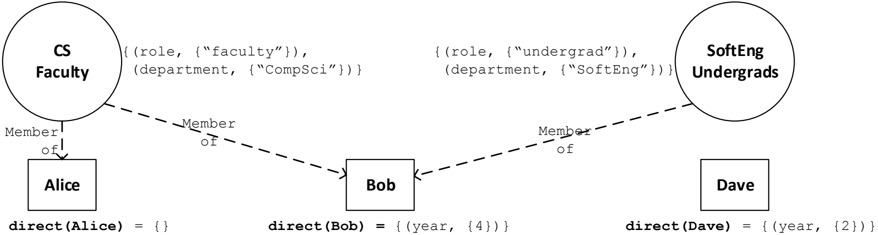
Issues with Attribute Delegation:

- Conflicting policy evaluations
- User collusion
- Effective attribute not descriptive of the delegatee
- User comprehension

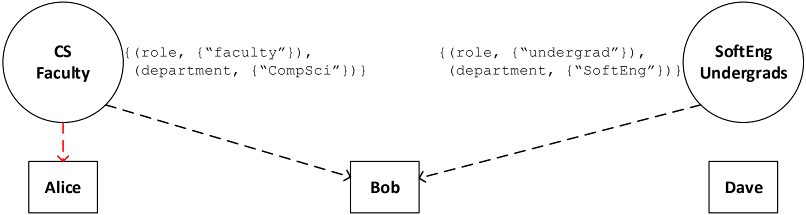
Group Membership Delegation

- Requires an ABAC model like HGABAC or GURAG which supports user groups (in which members inherit attributes).
- Group membership is delegated, rather than individual or subsets of attributes.
- Delegatee's effective attribute set is the combination of their directly assigned and inherited attributes (include those inherited from delegated memberships).

Group Membership Delegation: Example


CS Faculty
{(role, {"faculty"}), (department, {"CompSci"})}
{(role, {"undergrad"}), (department, {"SoftEng"})}
SoftEng Undergrads
Member of
Member of
Member of
Alice
Bob
Dave
direct(Alice) = {}
direct(Bob) = {(year, {4})}
direct(Dave) = {(year, {2})}

Group Membership Delegation: Example



direct(Alice) = {}

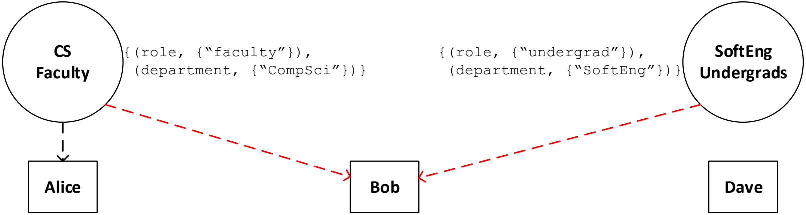
direct(Bob) = {(year, {4})}

direct(Dave) = {(year, {2})}

inherited(Alice) =
{(role, {“faculty”}),
(department, {“CompSci”})}

effective(Alice) =
{(role, {“faculty”}),
(department, {“CompSci”})}

Group Membership Delegation: Example



CS Faculty

{(role, {“faculty”}),
(department, {“CompSci”})}

{(role, {“undergrad”}),
(department, {“SoftEng”})}

SoftEng Undergrads

Alice

Bob

Dave

direct(Alice) = {}

inherited(Alice) =
{(role, {“faculty”}),
(department, {“CompSci”})}

effective(Alice) =
{(role, {“faculty”}),
(department, {“CompSci”})}

direct(Bob) = {(year, {4})}

**inherited(Bob) =
{(role, {“faculty”, “undergrad”}),
(department, {“CompSci”, “SoftEng”})}**

**effective(Bob) =
{(yaer, {4}),
(role, {“faculty”, “undergrad”}),
(department, {“CompSci”, “SoftEng”})}**

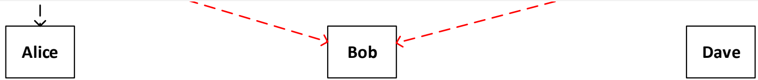
direct(Dave) = {(year, {2})}

Group Membership Delegation: Example

Example 4

Bob wishes to delegate his membership in the SoftEng Undergrads group to Dave such that he can satisfy the policy:

$$year \geq 2 \; AND \; department = \text{“SoftEng”}$$

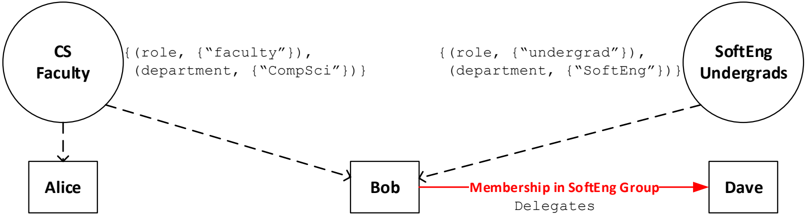


```
direct(Alice) = {}

inherited(Alice) =
{(role, {“faculty”}),
 (department, {“CompSci”})}

effective(Alice) =
{(role, {“faculty”}),
 (department, {“CompSci”})}
```

```
direct(Bob) = {(year, {4})}

inherited(Bob) =
{(role, {“faculty”, “undergrad”}),
 (department, {“CompSci”, “SoftEng”})}

effective(Bob) =
{(yaer, {4}),
 (role, {“faculty”, “undergrad”}),
 (department, {“CompSci”, “SoftEng”})}
```

```
direct(Dave) = {(year, {2})}
```

Group Membership Delegation: Example


CS Faculty

{(role, {“faculty”}),
(department, {“CompSci”})}

{(role, {“undergrad”}),
(department, {“SoftEng”})}

SoftEng Undergrads

Alice

Bob

Membership in SoftEng Group

Delegates

Dave

direct(Alice) = {}

inherited(Alice) =
{(role, {“faculty”}),
(department, {“CompSci”})}

effective(Alice) =
{(role, {“faculty”}),
(department, {“CompSci”})}

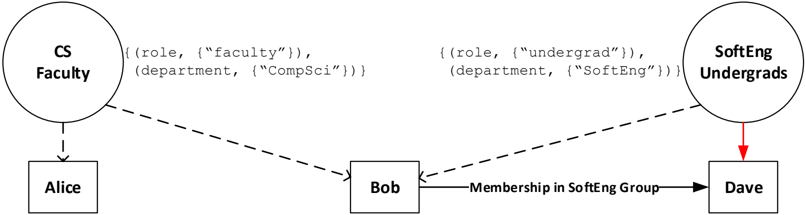
direct(Bob) = {(year, {4})}

inherited(Bob) =
{(role, {“faculty”, “undergrad”}),
(department, {“CompSci”, “SoftEng”})}

effective(Bob) =
{(yaer, {4}),
(role, {“faculty”, “undergrad”}),
(department, {“CompSci”, “SoftEng”})}

direct(Dave) = {(year, {2})}

Group Membership Delegation: Example



direct(Alice) = {}

inherited(Alice) =
{(role, {“faculty”}),
(department, {“CompSci”})}

effective(Alice) =
{(role, {“faculty”}),
(department, {“CompSci”})}

direct(Bob) = {(year, {4})}

inherited(Bob) =
{(role, {“faculty”, “undergrad”}),
(department, {“CompSci”, “SoftEng”})}

effective(Bob) =
{(yaer, {4}),
(role, {“faculty”, “undergrad”}),
(department, {“CompSci”, “SoftEng”})}

direct(Dave) = {(year, {2})}

inherited(Dave) =
{(role, {“undergrad”}),
(department, {“SoftEng”})}

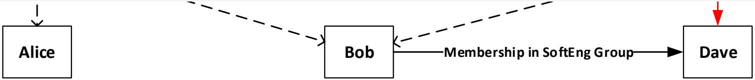
effective(Dave) =
{(year, {2}),
(role, {“undergrad”}),
(department, {“SoftEng”})}

Group Membership Delegation: Example

Example 4

Bob wishes to delegate his membership in the SoftEng Undergrads group to Dave such that he can satisfy the policy:

$$year \geq 2\ AND\ department = \text{“SoftEng”}$$



```
direct(Alice) = {}

inherited(Alice) =
{(role, {"faculty"}),
 (department, {"CompSci"})}

effective(Alice) =
{(role, {"faculty"}),
 (department, {"CompSci"})}
```

```
direct(Bob) = {(year, {4})}

inherited(Bob) =
{(role, {"faculty", "undergrad"}),
 (department, {"CompSci", "SoftEng"})}

effective(Bob) =
{(yaer, {4}),
 (role, {"faculty", "undergrad"}),
 (department, {"CompSci", "SoftEng"})}
```

```
direct(Dave) = {(year, {2})}

inherited(Dave) =
{(role, {"undergrad"}),
 (department, {"SoftEng"})}

effective(Dave) =
{(year, {2}),
 (role, {"undergrad"}),
 (department, {"SoftEng"})}
```

Group Membership Delegation: Problems/Benefits

Advantages of Group Membership Delegation:

- Easier to constrain
- User collusion is harder
- Attributes remain descriptive of delegatee
- Improved user comprehension

Issues with Group Membership Delegation:

- Requires user group support
- Issues shared with Attribute Delegation:
  - Conflicting policy evaluations
  - User collusion
- Undelegatable Attributes

Group Membership Delegation: Problems/Benefits

Advantages of Group Membership Delegation:

- Easier to constrain
- User collusion is harder
- Attributes remain descriptive of delegatee
- Improved user comprehension

Issues with Group Membership Delegation:

- Requires user group support
- Issues shared with Attribute Delegation:
    - Conflicting policy evaluations
    - User collusion
- Undelegatable Attributes

Group Membership Delegation: Problems/Benefits

Advantages of Group Membership Delegation:

- Easier to constrain
- User collusion is harder
- Attributes remain descriptive of delegatee
- Improved user comprehension

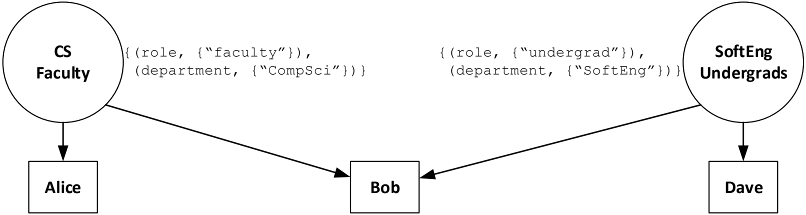
Issues with Group Membership Delegation:

- Requires user group support
- Issues shared with Attribute Delegation:
  - Conflicting policy evaluations
  - User collusion
- Undelegatable Attributes

Permission Delegation

- Permissions obtained from satisfying a policy are delegated directly.
- Delegated permissions are valid so long as the policy is satisfied by the original delegator.
- When a group is acting as the delegator, delegatable permissions are the set of permissions a user would be granted if they had the same attribute set as assigned to the group.

Permission Delegation: Example



CS Faculty

{(role, {“faculty”}),
(department, {“CompSci”})}

{(role, {“undergrad”}),
(department, {“SoftEng”})}

SoftEng Undergrads

Alice

Bob

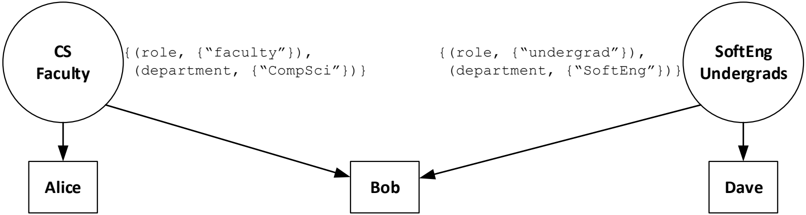
Dave

```
direct(Alice) = {}

inherited(Alice) =
{(role, {“faculty”}),
 (department, {“CompSci”})}

effective(Alice) =
{(role, {“faculty”}),
 (department, {“CompSci”})}
```

```
direct(Bob) = {(year, {4})}

inherited(Bob) =
{(role, {“faculty”, “undergrad”}),
 (department, {“CompSci”, “SoftEng”})}

effective(Bob) =
{(yaer, {4}),
 (role, {“faculty”, “undergrad”}),
 (department, {“CompSci”, “SoftEng”})}
```

```
direct(Dave) = {(year, {2})}

inherited(Dave) =
{(role, {“undergrad”}),
 (department, {“SoftEng”})}

effective(Dave) =
{(year, {2}),
 (role, {“undergrad”}),
 (department, {“SoftEng”})}
```

Permission Delegation: Example



CS Faculty

{(role, {"faculty"}), (department, {"CompSci"})}

{(role, {"undergrad"}), (department, {"SoftEng"})}

SoftEng Undergrads

Alice

Bob

Dave

direct(Alice) = {}

inherited(Alice) = {(role, {"faculty"}), (department, {"CompSci"})}

direct(Bob) = {(year, {4})}

inherited(Bob) = {(role, {"faculty", "undergrad"}), (department, {"CompSci", "SoftEng"})}

direct(Dave) = {(year, {2})}

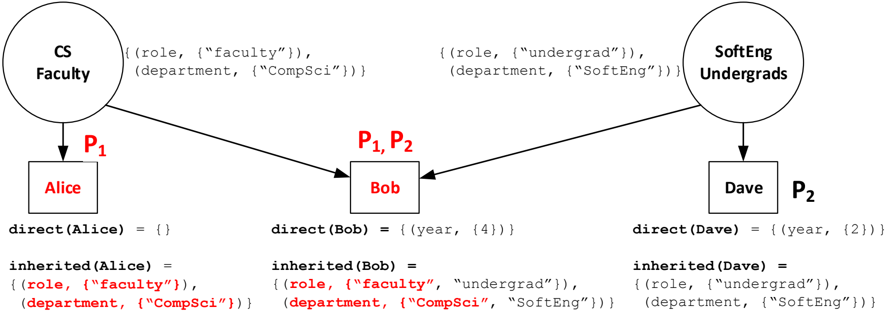
inherited(Dave) = {(role, {"undergrad"}), (department, {"SoftEng"})}

Example 5

$$\textit{role} = \textit{``faculty'' AND department} = \textit{``CompSci''} \Rightarrow p_1$$

$$\textit{year} \geq \textit{2 AND TIME} > \textit{9:00AM AND TIME} < \textit{5:00PM} \Rightarrow p_2$$

Permission Delegation: Example



CS Faculty

{(role, {“faculty”}), (department, {“CompSci”})}

{(role, {“undergrad”}), (department, {“SoftEng”})}

SoftEng Undergrads

P_1

Alice

P_1, P_2

Bob

Dave

P_2

direct(Alice) = {}

inherited(Alice) =
{(**role, {“faculty”}**),
(**department, {“CompSci”}**)}

direct(Bob) = {(year, {4})}

inherited(Bob) =
{(**role, {“faculty”**, “undergrad”}),
(**department, {“CompSci”**, “SoftEng”})}

direct(Dave) = {(year, {2})}

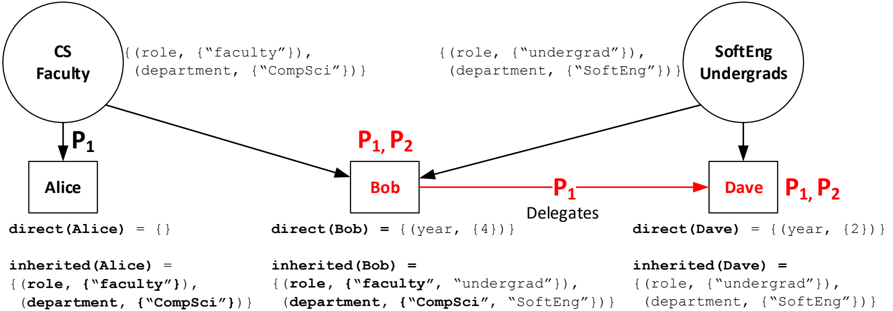
inherited(Dave) =
{(role, {“undergrad”}),
(department, {“SoftEng”})}

Example 5

$$role = \text{“}faculty\text{”}\ AND\ department = \text{“}CompSci\text{”} \Rightarrow p_1$$

$$year \geq 2\ AND\ TIME > 9{:}00AM\ AND\ TIME < 5{:}00PM \Rightarrow p_2$$

Permission Delegation: Example



{(role, {"faculty"}), (department, {"CompSci"})}

{(role, {"undergrad"}), (department, {"SoftEng"})}

direct(Alice) = {}

inherited(Alice) = {(**role**, **{"faculty"}**), (**department**, **{"CompSci"}**)}

direct(Bob) = {(year, {4})}

inherited(Bob) = {(**role**, **{"faculty"**, "undergrad"}), (**department**, **{"CompSci"**, "SoftEng"})}

direct(Dave) = {(year, {2})}

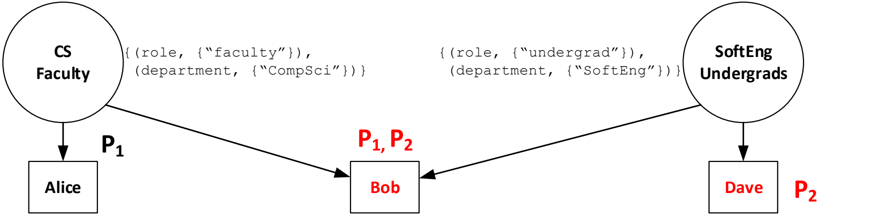
inherited(Dave) = {(role, {"undergrad"}), (department, {"SoftEng"})}

Example 5

$$role = \text{"faculty"}\ AND\ department = \text{"CompSci"} \Rightarrow p_1$$

$$year \geq 2\ AND\ TIME > 9{:}00AM\ AND\ TIME < 5{:}00PM \Rightarrow p_2$$

Permission Delegation: Example



CS Faculty

{(role, {"faculty"}), (department, {"CompSci"})}

SoftEng Undergrads

{(role, {"undergrad"}), (department, {"SoftEng"})}

P_1

P_1, P_2

P_2

Alice

Bob

Dave

direct(Alice) = {}

inherited(Alice) =
{(role, {"faculty"}),
(department, {"CompSci"})}

direct(Bob) = {**(year, {4})**}

inherited(Bob) =
{(role, {"faculty", "undergrad"}),
(department, {"CompSci", "SoftEng"})}

direct(Dave) = {**(year, {2})**}

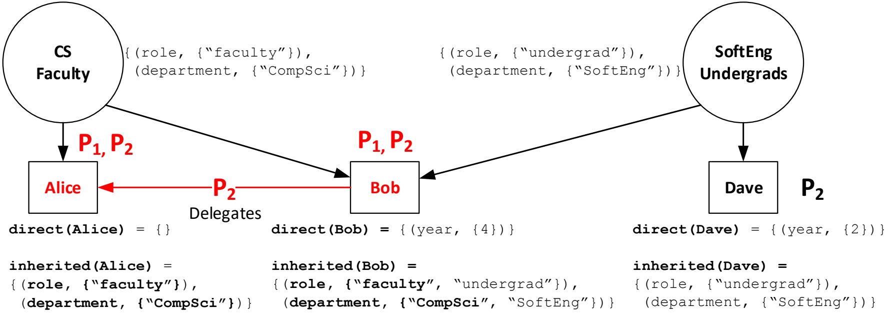
inherited(Dave) =
{(role, {"undergrad"}),
(department, {"SoftEng"})}

Example 5

$$\textit{role} = \textit{``faculty'' AND department} = \textit{``CompSci''} \Rightarrow p_1$$

$$\textit{year} \geq \textit{2 AND TIME} > \textit{9:00AM AND TIME} < \textit{5:00PM} \Rightarrow p_2$$

Permission Delegation: Example



CS Faculty
{(role, {"faculty"}), (department, {"CompSci"})}
{(role, {"undergrad"}), (department, {"SoftEng"})}
SoftEng Undergrads
P_1, P_2
P_1, P_2
Alice
P_2
Delegates
Bob
Dave
P_2

direct(Alice) = {}

inherited(Alice) =
{(role, {"faculty"}),
(department, {"CompSci"})}

direct(Bob) = {(year, {4})}

inherited(Bob) =
{(role, {"faculty", "undergrad"}),
(department, {"CompSci", "SoftEng"})}

direct(Dave) = {(year, {2})}

inherited(Dave) =
{(role, {"undergrad"}),
(department, {"SoftEng"})}

Example 5

$$role = \text{“}faculty\text{”}\ AND\ department = \text{“}CompSci\text{”} \Rightarrow p_1$$

$$year \geq 2\ AND\ TIME > 9{:}00AM\ AND\ TIME < 5{:}00PM \Rightarrow p_2$$

Permission Delegation: Problems/Benefits

Advantages of Permission Delegation:

- No changes to delegatee's attribute set
- No conflicting policy evaluations
- No user collusion
- Improved user comprehension

Issues with Permission Delegation:

- Implementation complexity
- Persistent evaluation of policies required

Permission Delegation: Problems/Benefits

Advantages of Permission Delegation:

- No changes to delegatee's attribute set
- No conflicting policy evaluations
- No user collusion
- Improved user comprehension

Issues with Permission Delegation:

- Implementation complexity
- Persistent evaluation of policies required

Qualitative Evaluation/Comparison

Informal evaluation based on following qualitative attributes:

- Required Features
- User Comprehension
- Attributes Remain Descriptive of Subject
- Potential for Conflicting Policy Evaluations
- Persistent Evaluation of Policies Required
- Implementation Complexity

Qualitative Evaluation/Comparison: Results

Strategy	Requires Features	User Comprehension	Attributes Remain Descriptive	Conflicting Policy Evaluations	Persistent Evaluation Required
Attribute Delegation					
User-to-User	Core ABAC	Low	No	Yes	No
User-to-Group	Core ABAC	Low	No	Yes	No
Group-to-Group	Core ABAC, User Groups	Low	Depends on Group	Yes	No
Group-to-User	Core ABAC, User Groups	Low	Depends on Group	Yes	No
User-to-Attribute	Core ABAC	Low	No	Yes	No
Group-to-Attribute	Core ABAC, User Groups	Low	Depends on Group	Yes	No
User-to-Policy	Core ABAC	Very Low	No	Yes	Yes
Group-to-Policy	Core ABAC, User Groups	Very Low	Depends on Group	Yes	Yes
Group Membership Delegation					
User-to-User	Core ABAC, User Groups	Medium	Depends on Group	Yes	No
Group-to-User	Core ABAC, User Groups	Medium	Depends on Group	Yes	No
Group-to-Group	Core ABAC, User Groups	Medium	Depends on Group	Yes	No
User-to-Group	Core ABAC, User Groups	Medium	Depends on Group	Yes	No
User-to-Attribute	Core ABAC, User Groups	Medium	Depends on Group	Yes	No
Group-to-Attribute	Core ABAC, User Groups	Medium	Depends on Group	Yes	No
User-to-Policy	Core ABAC, User Groups	Low to Medium	Depends on Group	Yes	Yes
Group-to-Policy	Core ABAC, User Groups	Low to Medium	Depends on Group	Yes	Yes
Permission Delegation					
User-to-User	Core ABAC	High	Yes	No	Yes
User-to-Group	Core ABAC	High	Yes	No	Yes
Group-to-User	Core ABAC, User Groups	High	Yes	No	Yes
Group-to-Group	Core ABAC, User Groups	High	Yes	No	Yes
User-to-Attribute	Core ABAC	High	Yes	No	Yes
Group-to-Attribute	Core ABAC, User Groups	High	Yes	No	Yes
User-to-Policy	Core ABAC	Medium to High	Yes	No	Yes
Group-to-Policy	Core ABAC, User Groups	Medium to High	Yes	No	Yes

Conclusions

- The ideal strategy largely depends on the needs and requirements of the implementing system.
- In general:
  - Permission Delegation strategies are ideal for systems requiring high user comprehension, removing conflicting policy evaluations and user collusion.
  - Attribute Delegation strategies are ideal when it is not possible to continually evaluate policies or low implementation complexity is desired.
  - Group Membership Delegation strategies provide higher user comprehension with similar results to Attribute Delegation but require user group support.

Future Work

- Using multiple strategies simultaneously could provide new possibilities for delegation.
- Existing policy conflict resolution techniques could help mitigate the issues faced by Attribute and Group Membership Delegation.
- Formalizing the strategies described in this work will allow for in-depth analysis and aid integration into existing ABAC models.
- Extending an existing model with each strategy would allow for a more quantitative evaluation and provide a reference model for future work.
- Revocation?, Multi-level delegation?, Monotonicity?, Totality?, etc.

Future Work

- Using multiple strategies simultaneously could provide new possibilities for delegation.
- Existing policy conflict resolution techniques could help mitigate the issues faced by Attribute and Group Membership Delegation.
- Formalizing the strategies described in this work will allow for in-depth analysis and aid integration into existing ABAC models.
- Extending an existing model with each strategy would allow for a more quantitative evaluation and provide a reference model for future work.
- Revocation?, Multi-level delegation?, Monotonicity?, Totality?, etc.