ReBAC in ABAC

by

Tahmina Ahmed
Department of Computer Science
University of Texas at San Antonio
4/29/2016

World-Leading Research with Real-World Impact!
Outline

• Introduction and BackGround
• A Simple ReBAC Model
• Relationships in ABAC
  ➢ Attribute Composition
  ➢ Composite Attribute
• A Composite ABAC Model
• Comparison
  ➢ Expressive Power
  ➢ Complexity
Using Relations For Controlling Access

IoT Application is a Graph

A sample social graph

A sample Provenance Graph (Park et al. 2012)
Existing Access Control Models those use some kind of Relations for authorization policy Expression

- **Social**
  - Uses social relationship to access OSN resources

- **Beyond Social**
  - Uses social relationship/relationship between system entities to access resources in any system

**ReBAC**

- **PBAC**
  - Uses Object’s Data Provenance Relation to access that object

Provenance Based Access Control
What Does ReBAC Mean?

- What does relationship based access control mean?
- What are the core characteristics of a ReBAC Model?
A Simple ReBAC Model (SReBAC[p])

Commands
- addRelation
- deleteRelation
- access

An Example Command Instantiation of SReBAC[3]

- addRelation\(^3\)(u_s:U, u_t:U)
  If \( u_t \notin P^1(u_s) \) then
  \( R \cup = \{ u_s, u_t \} \)
- deleteRelation\(^3\)(u_s:U, u_t:U)
  If \( u_t \in P^1(u_s) \land u_t \notin P^2(u_s) \) then
  \( R \setminus = \{ u_s, u_t \} \)
- access\(^3\)(u_s:U, u_t:U)
  If \( u_t \in P^3(u_s) \) then
  allow
An Example of a Simple ReBAC Command Execution

a. Initial state

b. After execution of action `addRelation(u_3, u_4)`

c. After execution of action `deleteRelation(u_1, u_2)`

d. After execution of action `access(u_2, u_5)`
**Expression of Relationship in ABAC**

**Attribute Composition**
- Needs one attribute: friend
- Policy Expression uses attribute composition

\[
\text{friend}(\text{Alice}) = \{\text{Bob}\} \\
\text{friend}(\text{friend}(\text{Alice})) = \{\text{Carol}\}
\]

**Composite Attribute**
- Needs two attributes
  1. friend
  2. friendoffriend
- Policy Expression uses direct attributes

\[
\text{friend}(\text{Alice}) = \{\text{Bob}\} \\
\text{friendoffriend}(\text{Alice}) = \{\text{Carol}\}
\]
Is this enough to keep the end user as an attribute value for Composite Attribute?

\[ \text{friend}(\text{Alice}) = \{\text{Bob, John}\} \]
\[ \text{friendoffriend}(\text{Alice}) = \{\text{Carol}\} \]
Is this enough to keep the end user as an attribute value for Composite Attribute?

![Diagram of friendship relationships]

After execution of `deleteRelation("Alice", "Bob")`

friend(Alice) = {John}
friendoffriend(Alice) = ?

So we need to keep the relationship path information as a value of a composite attribute.

friendoffriend(Alice) = {Bob.Carol, John.Carol} ---- Before Deletion
friendoffriend(Alice) = {John.Carol} ---- After Deletion
A Composite ABAC Model: $\text{ABAC}_c [n,m]$
SReBAC \([p]\) : Can Express Authorization Policy upto level \(p\)

\(\text{ABAC}_C [n,m]\): Can do \(n\) level attribute composition in authorization policy and has \(m - 1\) composite attributes.

So \(\text{ABAC}_C [n,m]\) can express Authorization Policy upto level \(n \times m\)

**Expressive Power Comparison:**

So if \(p = n \times m\)

SReBAC \([p]\) has same expressive power as \(\text{ABAC}_C [n,m]\)
Comparison: Expressive Power Vs. Complexity

Initial State:

- a. Initial state no relationship between \( u_s \) and \( u_t \), \( m < q \)

SReBAC\([m]\):

- Affected Components:
- b. Affected Components in SReBAC for a new relationship between \( u_s \) and \( u_t \)

ABAC\(_c\)[1, m]：“

- Affected Components:
- d. Update needs for ABAC\(_c\)[1, m]

ABAC\(_c\)[p, q]：“

- Affected Components:
- e. Update needs for ABAC\(_c\)[p, q]
## Comparison: Expressive Power Vs. Complexity

### Complexity Comparison

<table>
<thead>
<tr>
<th>Performance Parameter</th>
<th>SReBAC[n]</th>
<th>ABAC_C[n,1]</th>
<th>ABAC_C[L,m]</th>
<th>ABAC_C [p,q]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Complexity for Maintaining Relationship</td>
<td>$O(</td>
<td>U</td>
<td>+</td>
<td>U</td>
</tr>
<tr>
<td>Time Complexity for Computing Authorization Rule</td>
<td>$O(</td>
<td>U</td>
<td>^n)$</td>
<td>$O(</td>
</tr>
<tr>
<td>Number of update operations need for actions which change relationship</td>
<td>1</td>
<td>2</td>
<td>$</td>
<td>U</td>
</tr>
<tr>
<td>Cost for update (Worst Case)</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
<td>$O(m)$</td>
<td>$O(m)$</td>
</tr>
</tbody>
</table>
Questions/Comments