Secure Cloud-Assisted Smart Cars: Dynamic Groups and ABAC

Maanak Gupta, James Benson, Farhan Patwa and Ravi Sandhu

L13-2
CS6393
Spring 2020
Scope of Contribution

➢ Contribution
   ❖ Propose formalized ABAC model for cloud assisted applications.
   ❖ Dynamic groups and user preferences.
   ❖ Implementation of the model in AWS.

➢ Scope
   ❖ Single Central Cloud
   ❖ No direct access and physical tampering
   ❖ Communication Channel is encrypted.
   ❖ Data in Cloud is secure
   ❖ In-vehicle security not considered
Attribute Based Access Control

- **ABAC**: Decision based on the attributes of entities
- Attributes are name value pairs: `age (Alice) → 29`
- Core entities in ABAC include:
  - Users
  - Objects
  - Environment or Context
  - Operations

Attributes

- **Authorization Policies**: determine rights just in time
  - retrieve attributes of relevant entities in request
- Enhance flexibility and fine grained access control
Location Groups

- Categorizing wide locations into smaller groups.
- Vehicles dynamically become member based on current GPS, vehicle-type or individual user preferences.
- Ensure relevance of alerts and notifications
Attributes and Alerts

Vehicles move and are assigned to different groups and inherit their attributes/alerts.

- Speed Limit: 50 mph
  - Deer Threat: ON
  - Ice on Road: NO

- Speed Limit: 30 mph
  - Flood Warning: ON
  - Road Work: ON

- Speed Limit: 20 mph
  - School Zone: ON
  - Amber Alert: ABC123
Administrative Questions:
• How the attributes or alerts of groups are updated?
• How are moving entities assigned to groups?
• How groups hierarchy is created?

Operational Questions:
• How attributes and groups are used to provide security?
• How user privacy preferences are considered?

Reported MQTT message

```json
{ "state": { "reported": { "Latitude": "29.4769353", "Longitude": "-98.5018237" } } }
```
CV-ABAC$_G$ Model

![Diagram of CV-ABAC$_G$ Model]

- **Activity Decision**
- **System Level**
- **Activities (A)**
- **Attribute / Policy Association**
- **Many to Many Dynamic Group Association**
- **One to Many Association**

© Maanak Gupta  
World Leading Research with Real World Impact!

8
user, sensor, car, mechanic, restaurant

{ location, size, IP, direction, speed, VIN, cuisine-type}
Model Components

{ read, write, control, notify, administrative actions }
Model Components

Cars, traffic lights, smart-devices

Location groups, service-specific, vehicle-type

Sensor, ECU, on-board apps
Operational and Administrative Activities
{notification, alerts, group hierarchy updates}
Basic Sets and Functions

- S, CO, O, G, OP are finite sets of sources, clustered objects, objects, groups and operations respectively [blue circles in Figure 4].
- A is a finite set of activities which can be performed in system.

- \( \text{ATT} \) is a finite set of attributes associated with S, CO, O, G and system-wide.
- For each attribute \( \text{att} \) in \( \text{ATT} \), \( \text{Range}(\text{att}) \) is a finite set of atomic values.
- \( \text{attType}: \text{ATT} = \{\text{set, atomic}\} \), defines attributes to be set or atomic valued.
- Each attribute \( \text{att} \) in \( \text{ATT} \) maps entities in S, CO, O, G to attribute values. Formally,

\[
\text{att} : S \cup CO \cup O \cup G \cup \{\text{system-wide}\} \to \begin{cases} \text{Range}(\text{att}) \cup \{\bot\} & \text{if attType(\text{att}) = atomic} \\ 2^{\text{Range}(\text{att})} & \text{if attType(\text{att}) = set} \end{cases}
\]

- POL is a finite set of authorization policies associated with individual S, CO, O, G.
- \( \text{directG} : CO \to G \), mapping each clustered object to a system group, equivalently \( \text{CGA} \subseteq CO \times G \).
- \( \text{parentCO} : O \to CO \), mapping each object to a clustered object, equivalently \( OCA \subseteq O \times CO \).
- \( GH \subseteq G \times G \), a partial order relation \( \geq_g \) on G. Equivalently, \( \text{parentG} : G \to 2^G \), mapping group to a set of parent groups in hierarchy.
Effective Attributes of Groups, Clustered Objects and Objects (Derived Functions)

- For each attribute $att$ in ATT such that $attType(att)$ = set:
  
  - $effG_{att} : G \rightarrow \mathcal{2}^{\text{Range}(att)}$, defined as $effG_{att}(g_i) = \text{att}(g_i) \cup \bigcup_{g \in \{g_j | g_j \geq g_i \}} effG_{att}(g)$.
  
  - $effCO_{att} : CO \rightarrow \mathcal{2}^{\text{Range}(att)}$, defined as $effCO_{att}(co) = \text{att}(co) \cup effG_{att}(\text{directG(co)})$.
  
  - $effO_{att} : O \rightarrow \mathcal{2}^{\text{Range}(att)}$, defined as $effO_{att}(o) = \text{att}(o) \cup effCO_{att}(\text{parentCO(o)})$.

- For each attribute $att$ in ATT such that $attType(att)$ = atomic:
  
  - $effG_{att} : G \rightarrow \text{Range}(att) \cup \{\bot\}$, defined as $effG_{att}(g_i) = \begin{cases} \text{att}(g_i) & \text{if } \forall g' \in \text{parentG}(g_i). \ effG_{att}(g') = \bot \\ effG_{att}(g') & \text{if } \exists \text{parentG}(g_i). \ effG_{att}(\text{parentG}(g_i)) \neq \bot \text{ then select parent } g' \text{ with } effG_{att}(g') \neq \bot \text{ updated most recently.} \end{cases}$
  
  - $effCO_{att} : CO \rightarrow \text{Range}(att) \cup \{\bot\}$, defined as $effCO_{att}(co) = \begin{cases} \text{att}(co) & \text{if } effG_{att}(\text{directG(co)}) = \bot \\ effG_{att}(\text{directG(co)}) & \text{otherwise} \end{cases}$
  
  - $effO_{att} : O \rightarrow \text{Range}(att) \cup \{\bot\}$, defined as $effO_{att}(o) = \begin{cases} \text{att}(o) & \text{if } effCO_{att}(\text{parentCO(o)}) = \bot \\ effCO_{att}(\text{parentCO(o)}) & \text{otherwise} \end{cases}$

Attributes more Dynamic
Attributes Inheritance
Authorization Functions (Policies)

- Authorization Function: For each \( op \in OP \), \( \text{Auth}_{op}(s : S, ob : CO \cup O \cup G) \) is a propositional logic formula returning true or false, which is defined using the following policy language:

- \( \alpha := \alpha \land \alpha \mid \alpha \lor \alpha \mid (\alpha) \mid \neg \alpha \mid \exists x \in \text{set.} \alpha \mid \forall x \in \text{set.} \alpha \mid \text{set} \Delta \text{set} \mid \text{atomic} \in \text{set} \mid \text{atomic} \notin \text{set} \)

- \( \Delta := \subset \mid \subseteq \mid \not\subseteq \mid \cap \mid \cup \)

- \( \text{set} := \text{eff}_{\text{att}}(i) \mid \text{att}(i) \)

- \( \text{atomic} := \text{eff}_{\text{att}}(i) \mid \text{att}(i) \mid \text{value} \)

- \( \text{for att} \in \text{ATT}, i \in S \cup CO \cup O \cup G \cup \{\text{system-wide}\}, \text{attrType}(\text{att}) = \text{set} \)

- \( \text{for att} \in \text{ATT}, i \in S \cup CO \cup O \cup G \cup \{\text{system-wide}\}, \text{attrType}(\text{att}) = \text{atomic} \)

- Administrators in the police department can send alert to location-groups in city limits.

\[
\text{Auth}_{\text{alert}}(u:U, g:G) :: \text{dept}(u) \ \text{Police} \land \text{parent-city}(g) = \text{Austin} \land \ \text{Austin} \in \text{jursidiction}(u).
\]

- Only mechanic in the technician department from Toyota-X dealership must be able to read sensor in Camry LE. Further, this operation must be done between time 9 am to 6 pm.

\[
\text{Auth}_{\text{read}}(u:U, co:CO) :: \text{role}(u) \ \text{Technician} \land \text{employer}(u) = \text{Toyota-X} \land \ \text{make}(co) = \text{Toyota} \land \text{model}(co) = \text{Camry LE} \land \ \text{operation_time}(u) \in \{9am,10,11\ldots6pm\}
\]
Activity Authorization Decision

Authorization Decision

A source \( s \in S \) is allowed to perform an activity \( a \in A \), stated as \( \text{Authorization}(a : A, s : S) \), if the required policies needed to allow the activity are included and evaluated to make final decision. These multi-layer policies must be evaluated for individual operations \( (\text{op}_i \in \text{OP}) \) to be performed by source \( s \in S \) on relevant objects \( (x_i \in \text{CO} \cup \text{UG}) \).

Formally, \( \text{Authorization}(a : A, s : S) \Rightarrow \text{Auth}_{\text{op}_1}(s : S, x_1), \text{Auth}_{\text{op}_2}(s : S, x_2), \ldots, \text{Auth}_{\text{op}_n}(s : S, x_n) \)

Evaluate all relevant policies to make a decision

A restaurant in group A must be allowed to send notifications to all vehicles in location group A and group B.

I only want notifications from Cheesecake factory.

System defined

DECISION

User Preference
Implementation in Amazon Web Services (AWS)
Vehicles and Groups

4 Location Groups
(static demarcation)

Vehicles movement
(coordinates generated using Google API)

{'Received new coordinates from:', 'Vehicle-1'}
Sun May 27 02:56:30 2018
Location A
  Car-A : [u'Vehicle-1', u'Vehicle-2']
  Bus-A : []
Location B
  Car-B : []
  Bus-B : [u'Vehicle-6']
Location C
  Car-C : [u'Vehicle-3', u'Vehicle-4']
  Bus-C : []
Location D
  Car-D : []
  Bus-D : [u'Vehicle-5']

Snapshot (table keeps changing)
Implemented Policies

➢ Administrative Policy
  ❖ Road side motion sensor with [id = 1] and current GPS in group [Location-A] can only [modify] attribute [Deer Threat] to value [ON, OFF] for group [Location-A].

➢ Operational Policy
  
  Restaurant Notification Use Case
  System Defined Policy
  ❖ A restaurant located within group [Location-A] can only [send notifications] to members of groups [Location-A, Location-B].

  User Preferences
  ❖ Send notifications only between [7 pm to 9 pm] only on [Wednesdays].
### Performance Metrics

<table>
<thead>
<tr>
<th>Number of Requests</th>
<th>Policy Enforcer Execution Time (in ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0501</td>
</tr>
<tr>
<td>20</td>
<td>0.1011</td>
</tr>
<tr>
<td>30</td>
<td>0.1264</td>
</tr>
<tr>
<td>40</td>
<td>0.1630</td>
</tr>
<tr>
<td>50</td>
<td>0.1999</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>nth Request</th>
<th>With ABAC Policy</th>
<th>Without Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>41st</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>42nd</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>43rd</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>44th</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>45th</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>46th</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

**Policy Enforcement Time**

**Relevance of Alerts and Notifications**
Comparing Policy vs No Policy Execution Time
Conclusion and Future Work

- Proposed an **Attribute Based Access Control** solution for cloud assisted Smart Cars.
  - Introduced Dynamic Groups
  - Supports User Privacy Preferences and Location Centric
  - Proof of Concept implementation in AWS

- Future Research
  - Extensive and detailed evaluation
  - V2V and V2I secure trusted communication using Edge
  - Location preserving approaches