Access Control for Smart Objects

Jan Janak, Hyunwoo Nam, Henning Schulzrinne

Internet Real-Time Laboratory

Columbia University

This work is sponsored by AT&T Research.
Office Automation with Smart Objects

- Multiple controllers
- Need to specify and enforce policy
- A variety of inputs
- Provide reasonable default policies for SO

Overhead Light Policy:

```python
id==\"irt_switch\" or location==\"irt_lab\" or (action==\"OFF\" and app==\"SECE\") or Group==\"Facilities\"
```
Initial System Architecture

- Controllers send requests with a set of attributes.
- PEP verifies attribute values and signs the request.
- Policy documents specify conditions and required attributes.
- Each Smart Object type has a default policy that can be overridden.
What Makes it Complicated?

<table>
<thead>
<tr>
<th>SO Candidates</th>
<th>Policy Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Lights</td>
<td>• Identity</td>
</tr>
<tr>
<td>• Motion detectors</td>
<td>• Date and time</td>
</tr>
<tr>
<td>• Door locks</td>
<td>• Proximity</td>
</tr>
<tr>
<td>• Wall sockets</td>
<td>• Geo-location</td>
</tr>
<tr>
<td>• Towel dispensers</td>
<td>• Effort (press 3x)</td>
</tr>
<tr>
<td>• Fire alarm buttons</td>
<td>• Result of a vote</td>
</tr>
<tr>
<td>• Elevator controls</td>
<td>• Current state of SO</td>
</tr>
<tr>
<td>• Phones</td>
<td>• Organizational role</td>
</tr>
<tr>
<td>• Indoor location</td>
<td>• Randomness</td>
</tr>
</tbody>
</table>
Open Questions

• How do we describe and enforce access restrictions applied to Smart Objects?

• What protocols can we use to implement attribute-based access control?

• Mapping of credentials to CoAP/HTTP requests?

• Where is policy enforced? How do SOs learn the outcome?

• Default policy from SO manufacturers?