SmartAuth: User-Centered Authorization for the Internet of Things

Yuan Tian, Nan Zhang, Yueh-Hsun Lin, XiaoFeng Wang, Blase Ur, XianZheng Guo and Patrick Tague

Carnegie Mellon University, Indiana University Bloomington, Samsung, University of Chicago
Smart-home apps improve quality of life
Smart-home apps improve quality of life, but are risky
Users have limited information about what is going on

Alfred Workflow
by SmartThings
Turn your SmartThings-powered lights and switches on/off directly from your Mac by launching the Alfred Workflow and entering "ST On" and "ST Off."

Functionalities explained to the user

Carnegie Mellon University
CyLab
Users have limited information about what is going on

Functionalities explained to the user

Operations that the app indeed perform
Can we notify users about the most important information?

For behaviors related to functionality, we don’t have to. We should notify them about unexpected behaviors.

This app doesn’t need to control the lock!
Challenges

- Security and privacy implications depend on context
  - Same sensor in bedroom vs. outside has very different implications
- Behaviors in code cannot be mapped directly to high-level functionality in description
- Need to support cross-device scenarios
Previous solutions will not work

<table>
<thead>
<tr>
<th>Solution</th>
<th>Context-aware</th>
<th>Automatic</th>
<th>Usable</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manifest Permission</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Prompt Permission</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SmartAuth</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The app measures humidity level via:
- humidity sensor (Tap to set)
- to turns on or turns off
- switch (Tap to set)

The app can lock/unlock your:
- lock (Tap to set)
Redesign the authorization system

Goals:

- Security and Privacy: Share minimum data and capabilities for desired functionality
- IoT specific: Cross-device, context-based, automatic control
- Usability: Assist user to make well-informed decisions, minimize user burdens
- Performance: Lightweight and compatible
SmartAuth overview

Program Analyzer

Content Inspector (NLP e.g., [2])

Consistency Checker

Policy Enforcer

Authorization creator

An example – program analyzer

```java
section("Bathroom humidity sensor") {
    input "bathroom", "capability.relativeHumidityMeasurement",
    title: "Which humidity sensor?"
}

if (shower.value.toInteger() > relHum) {
    coffee.on()
```
An example – NLP and behavior correlation

Description analysis

Entity:
- Coffee machine
- Shower

Condition:
- Taking a shower

Triggers:
- Action: Turn on the coffee machine

Program analysis

Entity:
- Switch
- Humidity sensor
- Lock

Context clue:
- Bathroom for the humidity sensor
- Coffee for the switch

Condition:
- Humidity reading > threshold

Triggers:
- Action: Turn on the switch
- Unlock the door
Interface generation

- Match users’ mental models
  - Less burden for users
  - Alarm users about unexpected behaviors
- Survey users’ perspectives of installing smart-home apps
- Iterative design with pilot studies
Enforcement
Evaluation

- How effective is SmartAuth?
  - How accurate is the policy extraction?
  - How does SmartAuth impact users’ decisions?
- What is the performance overhead?
- How compatible is SmartAuth?
Evaluation: Effectiveness of extracting policies

- Manual analysis to verify all the cases
- 3.9% false positives
  - Limitations of NLP analysis
- No false negatives
Evaluation: User study

- Between-subjects, in-lab study
- 100 participants split into two groups:
  - SmartAuth
  - Current SmartThings interface (manifest-style)
- Five pairs of similar apps
  - Participant chooses one of the two
  - One has unexpected privileges
Example app pairs

- Lights off: Turn lights off when no motion or presence detected for a period of time
- Darken behind me: Turn your lights off after a period of no motion being observed
SmartThing Interface

- **Lights Off**
  - Light switches to turn off
  - Choose light switches
    - Tap to set
  - Turn off when there is no motion and presence
  - Choose motion sensor
    - Tap to set
  - Choose presence sensors
    - Tap to set

- **Darken Behind Me**
  - When there's no movement...
  - Where?
    - Tap to set
  - Monitor the temperature...
  - Which?
    - Tap to set
  - Turn off a light...
  - Which?
    - Tap to set
SmartAuth Interface

- **Lights Off**

  - **Verified Behavior**
    - **ON** The app detects motion status via
      - motion sensor (Bedroom Motion)
      - and detects people presence via
      - presence sensor (Tap to set)
      - to turn off
      - switch (Tap to set)
    - **REMOVE**

- **Darken Behind Me**

  - **Verified Behavior**
    - **ON** The app detects motion status via
      - motion sensor (Tap to set)
      - to turn off
      - switch (Tap to set)
    - **Unexpected Behavior**
      - The app can read the status of your
      - temperature measurement (Tap to set)
    - **REMOVE**
# SmartThing VS SmartAuth

<table>
<thead>
<tr>
<th>Verified Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ON</strong> The app detects motion status via</td>
</tr>
<tr>
<td>- motion sensor (Tap to set)</td>
</tr>
<tr>
<td>- to turn off</td>
</tr>
<tr>
<td>- switch (Tap to set)</td>
</tr>
<tr>
<td>- accordingly</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unexpected Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OFF</strong> The app can read the status of your</td>
</tr>
<tr>
<td>- temperature measurement (Tap to set)</td>
</tr>
</tbody>
</table>

### When there's no movement...

<table>
<thead>
<tr>
<th>Where?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tap to set</td>
</tr>
</tbody>
</table>

### Monitor the temperature...

<table>
<thead>
<tr>
<th>Which?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tap to set</td>
</tr>
</tbody>
</table>

### Turn off a light...

<table>
<thead>
<tr>
<th>Which?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tap to set</td>
</tr>
</tbody>
</table>
Users make better decisions with SmartAuth
Evaluation: Performance

- Pre-processing performance
  One-time cost to platform provider: 10.42 seconds per app on average
- Run-time enforcement:
Evaluation: Compatibility

- Observe behaviors and debug information
- None of the apps crash
- In the extreme case, 3.3% of apps loss functionality when we block all remote access
Takeaways

- **Goal:** Bridge semantic gap between what a user sees (app descriptions) and what an app’s code actually does
  - NLP to understand descriptions and code annotations
  - Program analysis to understand code
  - Match insights from NLP to program analysis

- **Users much more likely to choose safer apps with SmartAuth**

- **Working with Samsung for deployment**
Advertisement :)