Access Control Synthesis for Physical Spaces

Petar Tsankov, Mohammad Torabi Dashti, David Basin
Institute of Information Security, ETH Zurich
Corporate buildings
Setting

- Lobby
- Meeting room
- Office

Lobby

Meeting room

Office

Locks
Setting

Local Policy
"Only employees can enter"

Eric

Global requirement
"Employees can access the office from the main entrance"
Setting

Local Policy
“Only employees can enter”

Global requirements
“Employees can access the office from the main entrance”

Wrong deny!

Eric
Local Policy
“Only employees can enter”

Challenge
Come up with local policies that enforce all global requirements

Eric

Global Requirement
“Employees can access the office from the main entrance”
Current Practice

- No policies yet
- Physical space
- Requirements
- Local policies
- Global requirements
- Manual policy writing

Problems

- Cannot satisfy requirements one-by-one
Example

R1: Visitors can access the meeting room
Example

Current Policy:
No policies yet

Local policies:
Physical space

Current Practice:
Manual Policy Writing

Global requirements:
Problems

Victor

R1: Visitors can access the meeting room
Example

R1: Visitors can access the meeting room

R2: Visitors cannot access the meeting room if they have not passed through the lobby
No policies yet

Local policies

Physical space

Current Practice

Manual Policy Writing

Requirements

Global requirements

Problems

Cannot satisfy requirements

Example

Lobby

Meeting room

Victor

R1: Visitors can access the meeting room

R2: Visitors cannot access the meeting room if they have not passed through the lobby

Wrong permit!
**Current Practice**

<table>
<thead>
<tr>
<th>Physical space</th>
<th>Manual Policy Writing</th>
<th>Requirements</th>
<th>Global requirements</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lobby</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Meeting room</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Office</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example

**R1:** Visitors can access the meeting room

**R2:** Visitors cannot access the meeting room if they have not passed through the lobby
Example

R1: Visitors can access the meeting room
R2: Visitors cannot access the meeting room if they have not passed through the lobby
Current Practice

Problems

Cannot satisfy requirements one-by-one

Rewrite policies upon changes to the physical space or requirements
Current Practice

Problems

- Cannot satisfy requirements one-by-one
- Rewrite policies upon changes to the physical space or requirements
- No security guarantees
**Current Practice**

Physical space  
Requirements  
Manual policy writing

**Policy Synthesis**

Physical space  
Requirements  
Automated policy synthesis
Goal

Automatically compute correct local policies for a given physical space and its global requirements

Contributions

- Formalization of physical access control
- Expressive declarative language for specifying global requirements
- Efficient synthesis algorithm based on SMT solving
- Demonstration of the approach on realistic case studies
Formalizing Physical Spaces
Formalizing Physical Spaces

Enclosed space = Node
Lock = Edge

Formalize

entry

lobby

office

corridor

meeting room
Formalizing Physical Spaces

Enclosed space = Node
Lock = Edge
Formalizing Physical Spaces

Enclosed space = Node
Lock = Edge

Label physical spaces with attributes (e.g., to mark security zones)
Local Policies

Attribute-based policies with:
- Subject attributes (e.g. roles)
- Contextual attributes (e.g. time)

Local policy semantics
- An access request maps attributes to values
- A lock grants an access request if the access request satisfies the lock’s local policy

\[(role = \text{visitor}) \land (8 \leq \text{time} \leq 20)\]
Semantics of Physical Access Control

An access request is authorized along a path if all locks along the path grant it:

\[(\text{role} = \text{visitor}) \land (8 \leq \text{time} \leq 20)\]

Example

\[\text{AccReq}_1 = \{ \text{role} \mapsto \text{visitor}, \text{time} \mapsto 6 \}\]
Semantics of Physical Access Control

An access request is authorized along a path if all locks along the path grant it.

Example

\[ AccReq_1 = \{ \text{role} \mapsto \text{visitor}, \text{time} \mapsto 6 \} \]
Specifying Global Requirements
Requirement Examples

Visitors can access the meeting room
Requirement Examples

- Visitors can access the meeting room
- Non-employees cannot access the office
Requirement Examples

- Visitors can access the meeting room
- Non-employees cannot access the office
- Visitors cannot access the meeting room if they have not passed through the lobby
The SpCTL Language

**Key features**
- Subject & contextual attributes
  - e.g. *role, time*
- Resource attributes
  - e.g. *securityZone*
- Quantification over paths

**Common patterns**
- Permission
- Prohibition
- Waypointing

**Example:** 
\((\text{role} = \text{visitor}) \land (8 \leq \text{time} \leq 20) \Rightarrow EF(id = mr)\)
The SpCTL Language

Key features
- Subject & contextual attributes: e.g. role, time
- Resource attributes: e.g. security
- Quantification over subject & contextual attributes

Common patterns
- Permission
- Prohibition
- Waypointing

Example: \((\text{role} = \text{visitor}) \land (8 \leq \text{time} \leq 20) \Rightarrow EF (\text{id} = mr)\)

Constraint over subject & contextual attributes

CTL formula over resource attributes
Policy Synthesis Problem
Policy Synthesis Problem

Input
- Physical space

Requirements
- (role = visitor) \[ \Rightarrow EF(id = mr) \]
- ...

Policy Synthesis

Output
- (role = employee)

How hard is this problem?

Unsat
Complexity of Policy Synthesis

**Theorem 1.** The policy synthesis problem is decidable.

**Proof.** We give a synthesis algorithm that uses CTL controller synthesis as a subroutine.

**Theorem 2.** The policy synthesis problem is NP-hard.

**Proof.** Through reduction from propositional satisfiability to policy synthesis.
Complexity of Policy Synthesis

**Theorem 1.** The policy synthesis problem is decidable.

**Proof.** We give a synthesis algorithm that uses CTL controller synthesis as a subroutine.

**Theorem 2.** The policy synthesis problem is NP-hard.

**Proof.** Through reduction from propositional satisfiability to policy synthesis.
Policy Synthesis using SMT Solving

Physical space

Requirements

SMT Solving

(role = visitor)
⇒ EF(id = mr)
...

Unsat
Policy Synthesis using SMT Solving

Physical space

Requirements

 Encode the requirements' satisfaction using SMT constraints

A model identifies correct local policies

\[ \text{(role} = \text{visitor}) \Rightarrow EF(id = mr) \]

Unsat
Encode the requirements' satisfaction using SMT constraints.
Encode the requirements' satisfaction using SMT constraints.

Policy Synthesis Algorithm

Example Template

\( (role = \text{visitor}) \land (8 \leq \text{time} \leq 20) \)

(example instantiation)
Encode the requirements’ satisfaction using SMT constraints

Policy Synthesis Algorithm

Reduce search space

Synthesize concise local policies

Physical space

Requirements

SMT Solving

(\text{role} = \text{visitor}) \implies \text{EF}(\text{id} = \text{mr})

\ldots

Unsat
Implementation and Evaluation
Implementation

Our system is publicly available

https://github.com/ptsankov/SpCTL
<table>
<thead>
<tr>
<th>Building</th>
<th>Physical spaces</th>
<th>Locks</th>
<th>Requirements</th>
<th>Synthesis time</th>
</tr>
</thead>
<tbody>
<tr>
<td>KABA Headquarters</td>
<td>20</td>
<td>41</td>
<td>10</td>
<td>25s</td>
</tr>
<tr>
<td>ETH’s CS Department</td>
<td>66</td>
<td>127</td>
<td>14</td>
<td>10s</td>
</tr>
<tr>
<td>Airport Terminal</td>
<td>13</td>
<td>32</td>
<td>15</td>
<td>2s</td>
</tr>
</tbody>
</table>
Global requirements vs local enforcement

“Only employees can enter”

Employees can access the office from the main entrance

Summary

Policy synthesis framework

Approach scales to realistic problems