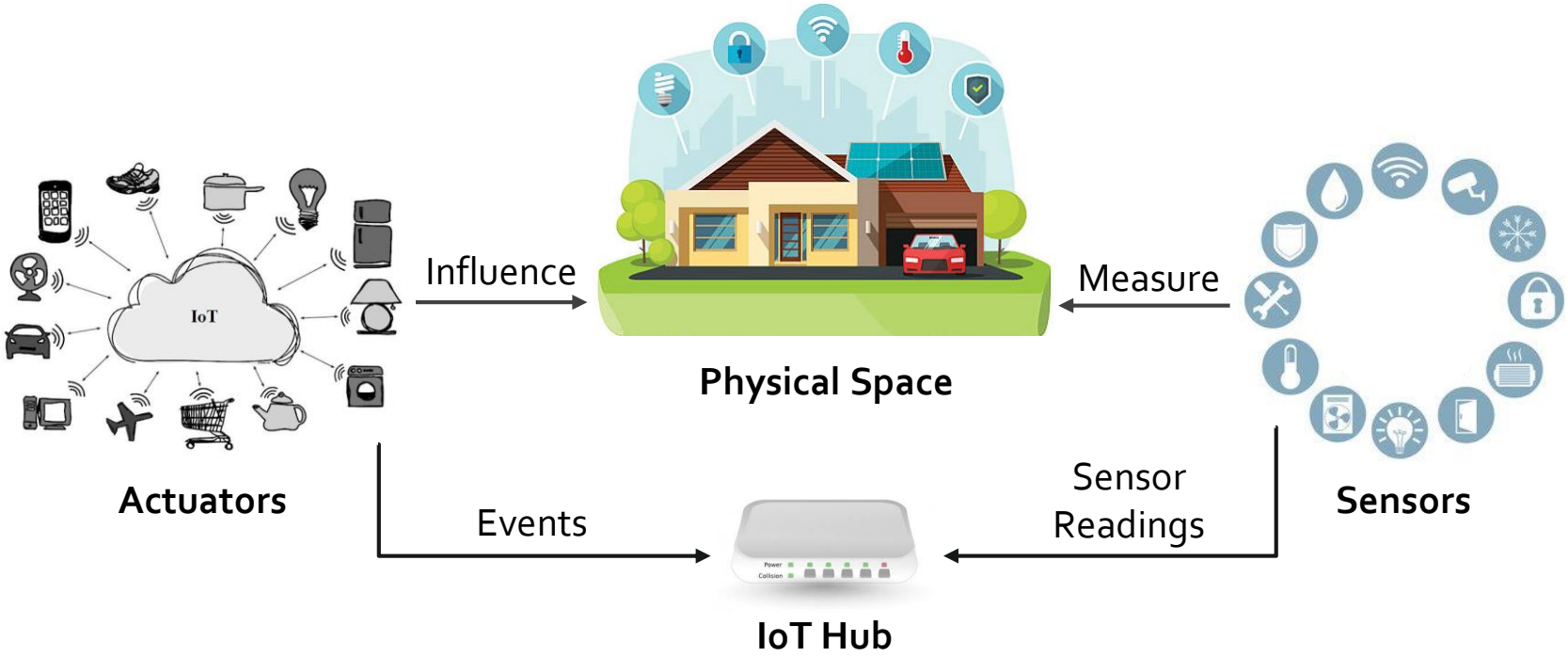
Evasion Attacks and Defenses on Smart Home Physical Event Verification

Muslum Ozgur Ozmen, Ruoyu Song, Habiba Farrukh, and Z. Berkay Celik

March 2, 2023

Smart Homes

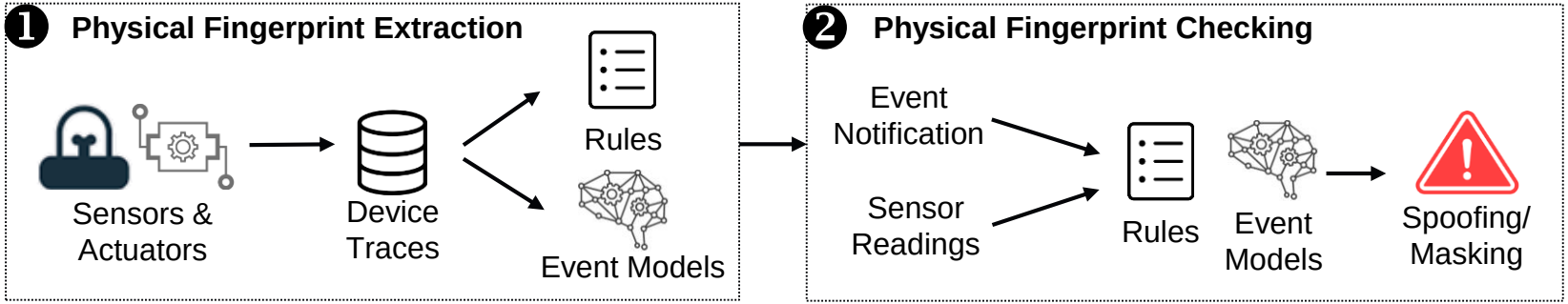

IoT
Influence
Measure
Physical Space
Actuators
Events
Sensor Readings
Sensors
IoT Hub

Event Spoofing and Masking

- **Event Spoofing**: An adversary reports to the IoT hub a fake event notification that did not physically occur

- **Event Masking**: An adversary suppresses the notification of an event that physically occurred

Event Verification Systems



- Example:

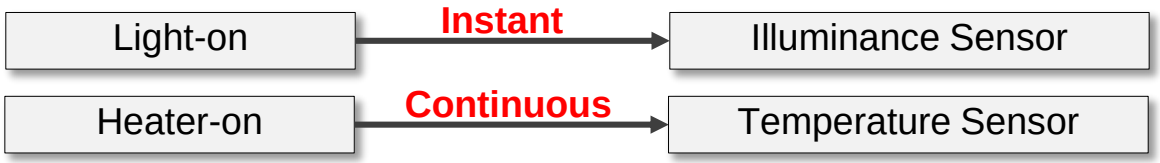
Offline: $Light\text{-}bulb\text{-}on \leftrightarrow S1.illum = high,\ S2.illum = high$

Online:

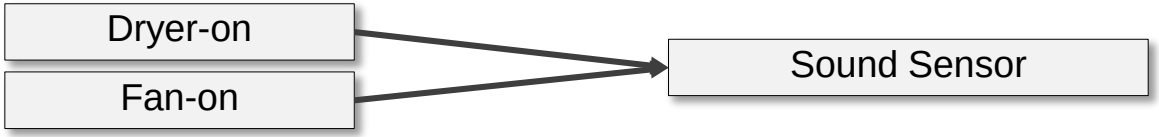
S1.illum = low
S2.illum = low
Spoofing Attack Detected!

Complex Physical Relations

- Continuous and instant physical influences



- Aggregated influences



- Distance between devices
  - The actuator's influence on sensor readings monotonically decreases when the distance between devices increases

Unfortunately, existing EVS ignore the complex physical relations between actuators and sensors, making them vulnerable to evasion

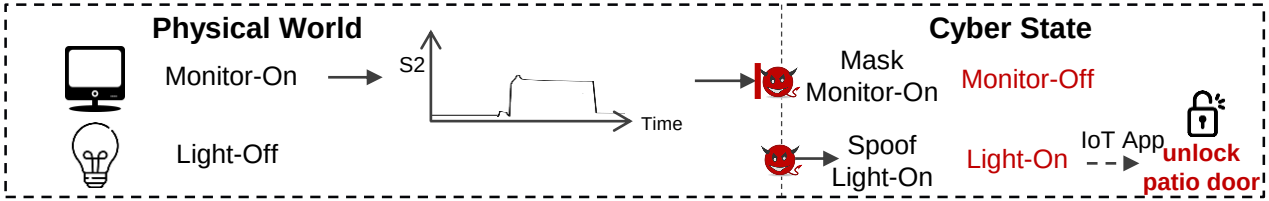
Evasion Attacks

Fingerprint Encapsulation

End States	
Physical	Cyber
TV-On	TV-On
Light-Off	Light-On

Fingerprint Concealment

End States	
Physical	Cyber
AC-On	AC-Off
Oven-On	Oven-Off

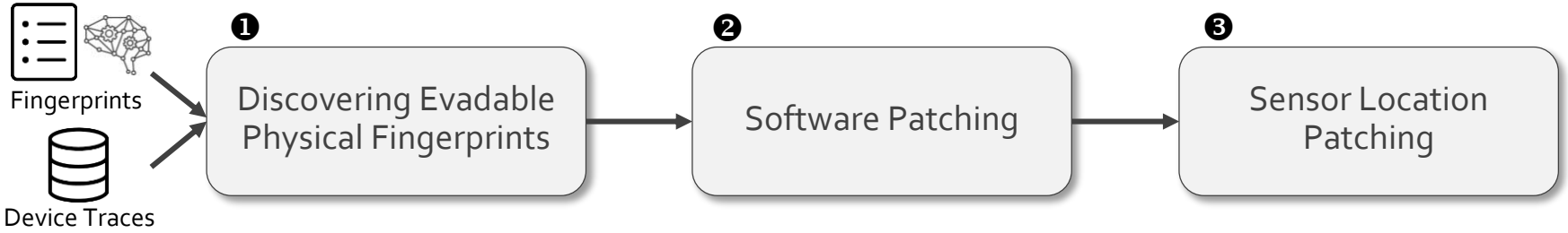


Fingerprint Indistinguishability

End States	
Physical	**Cyber**
Monitor-On	Monitor-Off
Light-Off	Light-On

Our System

- We propose a system to make EVS robust against evasion attacks



Discovering Evadable Physical Fingerprints

- We check whether
  - an event's fingerprint is satisfied when other events occur

$$E_i \leftrightarrow \{S_{i,1} = High, S_{i,2} = High, \ldots, S_{i,n} = High\}$$

$$E_j \leftrightarrow \{S_{j,1} = High, S_{j,2} = High, \ldots, S_{j,m} = High\}$$

$$S_{j,1}, S_{j,2}, \ldots, S_{j,m} \subseteq S_{i,1}, S_{i,2}, \ldots, S_{i,n}$$

  - an event's fingerprint is concealed when other events occur

$$E_i \leftrightarrow S = Inc, E_j \leftrightarrow S = Dec$$

Software Patching

- We derive new fingerprints to define aggregated influences from events

Example:

$$E_1 \leftrightarrow \{S_1 = High, S_2 = Med\}$$

$$E_2 \leftrightarrow \{S_1 = High, S_2 = Med, S_3 = High\}$$

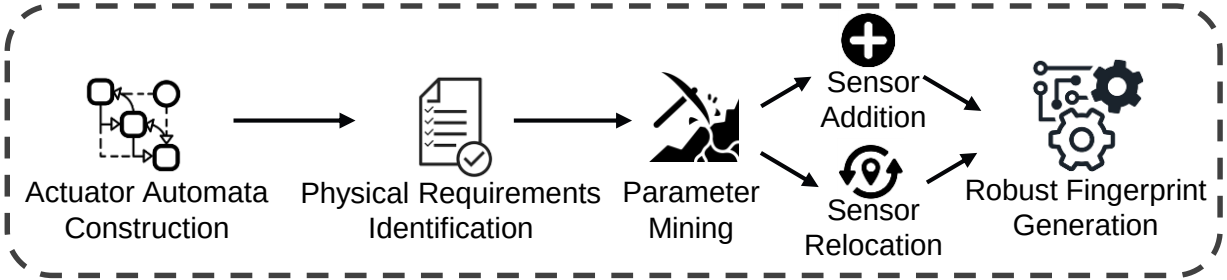
$$E_1 \subseteq E_2 \rightarrow \text{Spoof } E_1 \text{ when } E_2 \text{ occurs}$$

$$E_{agg} \leftrightarrow \{S_1 = Agg\_High, S_2 = High, S_3 = High\}$$

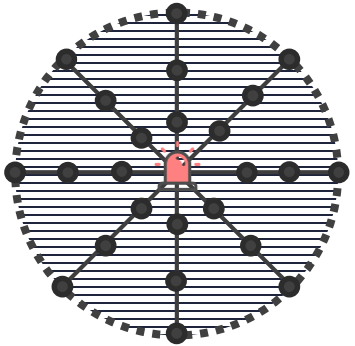
Limitation: The EVS cannot distinguish the event's aggregated influence as well

- Sensors that make Boolean-typed readings

Sensor Location Patching

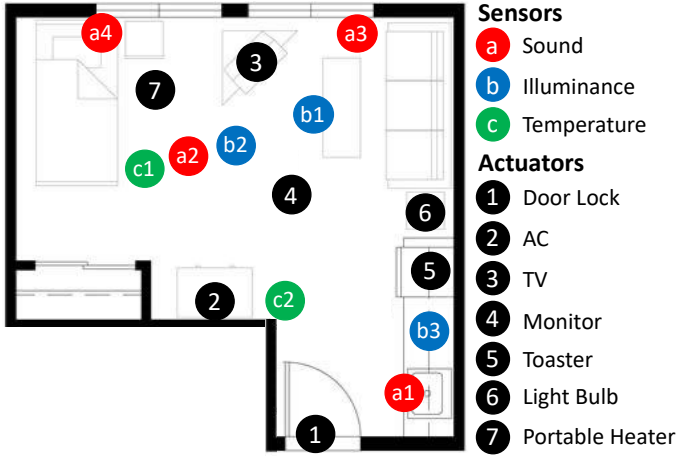


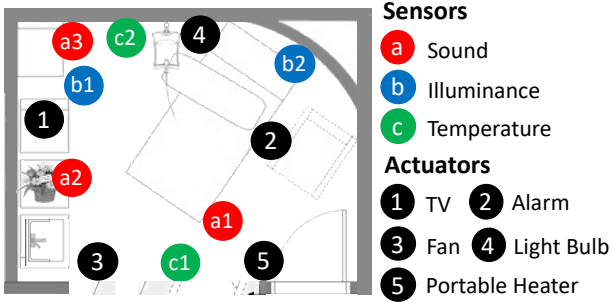
Output:



Evaluation

- Two state-of-the-art EVS
- Two smart home testbeds
  - 12 actuators
  - 16 sensors



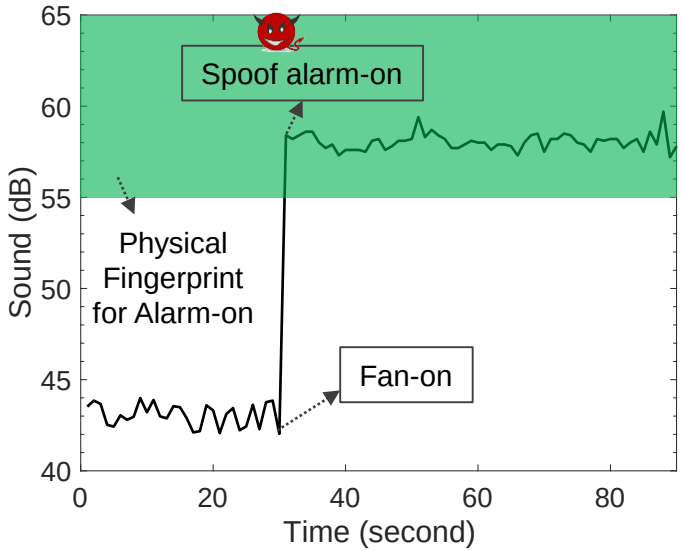


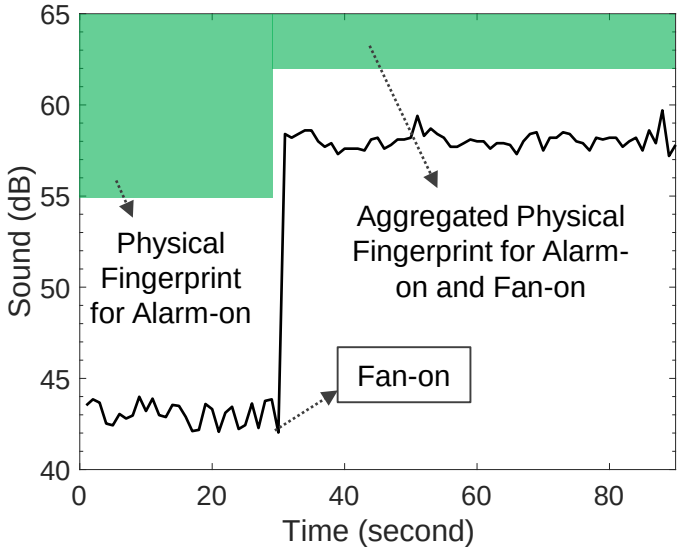
Results

- 71% of the physical fingerprints extracted by existing EVS are vulnerable to **evasion attacks**
- **Software patching** prevents 52% of the evasion attacks
- **Sensor location patching** prevents *all* remaining evasion attacks

Case Study

- Software patching prevents the spoofing attack against alarm-on





Conclusion

- We show that EVS do not consider complex physical interactions between devices, allowing an adversary to evade them
- We propose two complementary defenses:
  - Software patching creates new physical fingerprints that define the aggregated influences from events and integrates them into EVS
  - Sensor location patching is a security-by-design approach that finds a sensor placement to ensure events have unique physical fingerprints
- Our approach builds robust physical event fingerprints for EVS, allowing them to properly mitigate realistic attack vectors

Thank you! Questions?

mozmen@purdue.edu

https://github.com/purseclab/EVS_Evasion