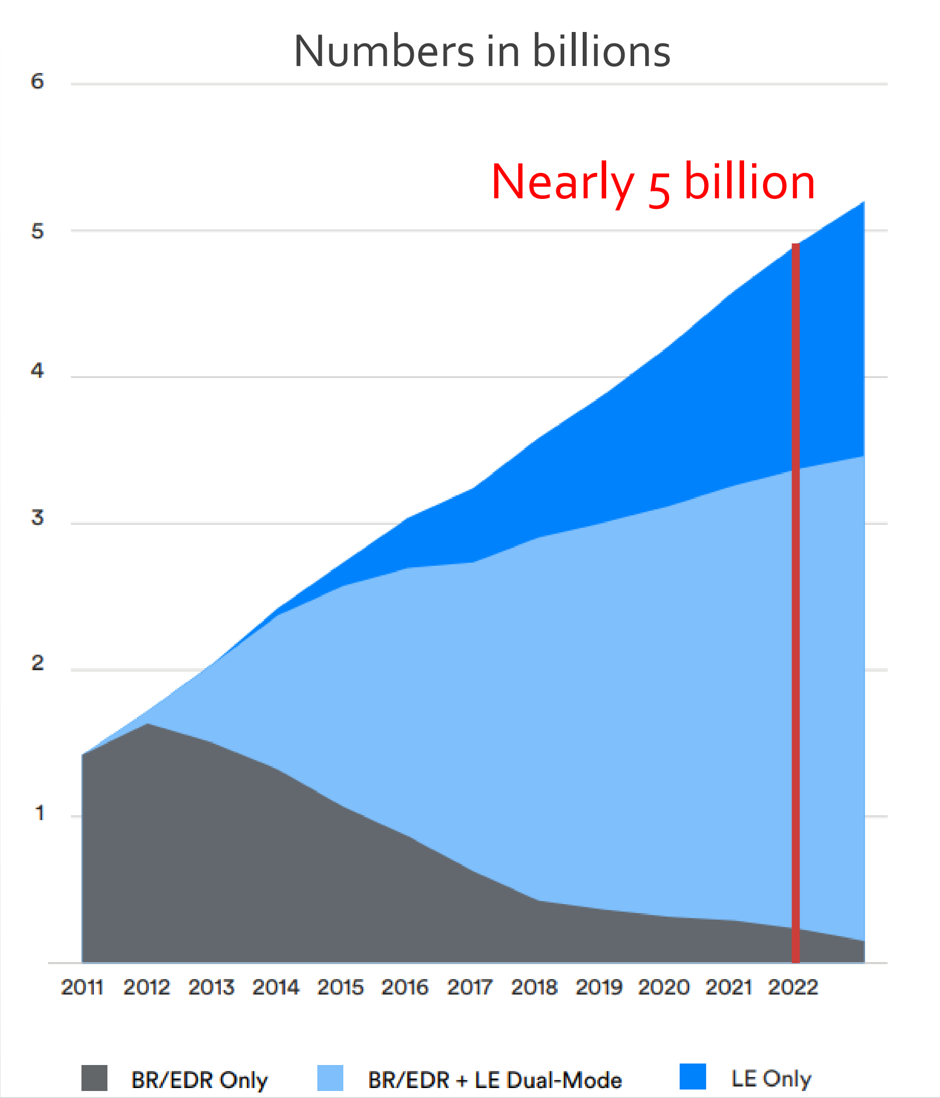
BlueShield: Detecting Spoofing Attacks in Bluetooth Low Energy Networks

Jianliang Wu[1], Yuhong Nan[1], Vireshwar Kumar[1], Mathias Payer[2], Dongyan Xu[1]

[1] Purdue University [2] EPFL

Motivation

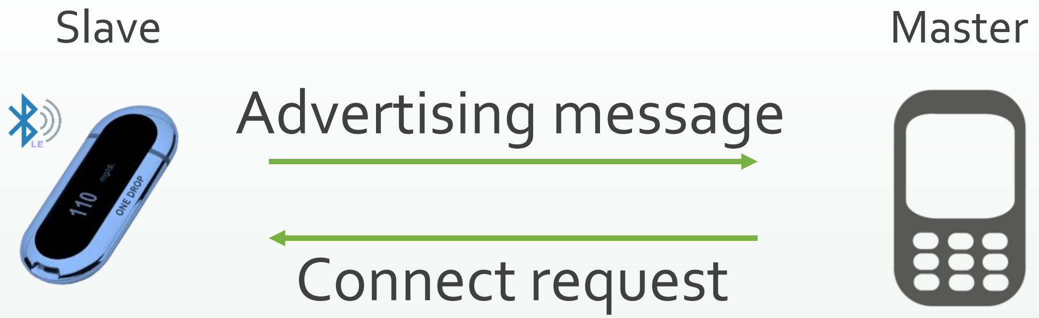
- Bluetooth Low Energy (BLE) devices are ubiquitous
  - Smart homes
    - E.g., Smart lock
  - Smart buildings
    - E.g., temperature sensors



[1]Bluetooth Market Update 2018 (https://www.bluetooth.com/markets/market-report)

Background

- BLE discovery procedure
  - Advertising and scanning
  - **No authentication** in advertising message
- BLE spoofing attack
  - Feed malicious data to the user device (master device)



Background

- BLE security mechanism
  - Pairing
    - Encryption and authentication
  - Pairing is **not mandatory**, many devices (~**80%**) **do not support/enable** pairing (no encryption)[1]
  - Spoofing enabling vulnerabilities in different layers
    - Vulnerable encryption (app layer)
    - Design & implementation flaws (Bluetooth stack)
    - 0-day vulnerabilities (other parts of both devices and smartphones)

[1]: https://www.owasp.org/images/archive/6/6f/20170811005623%21OWASP2017_HackingBLEApplications_TalMelamed.pdf

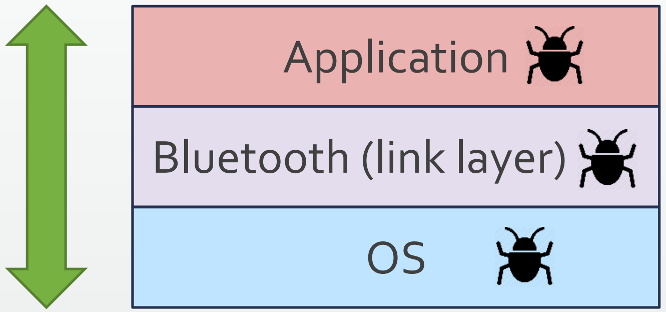
BlueShield: design objectives

- Vulnerability (and device) agnostic
  - No ad-hoc, case-by-case software-level fix
  - Support billions of legacy devices
    - No device firmware modification or reverse engineering needed
- Practical and effective
  - Fully transparent to deployment environment
    - No intervention to the BLE device and the user under normal usage scenario
  - Off-the-shelf, easy to deploy
    - Low-cost, commodity hardware
  - High accuracy and low latency

Spoofing detection : challenges and solutions

Challenge

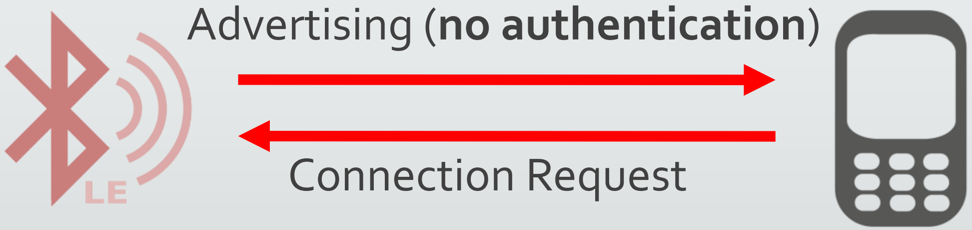
1. How to protect all different layers?



Solution

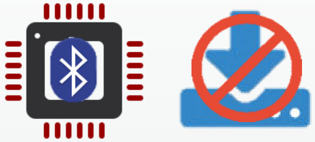
Focusing on the discovery step – advertising messages

- Easy fix: sign advertising messages
- But it needs **firmware/protocol modification** (not practical)



Spoofing detection : challenges and solutions

Challenge



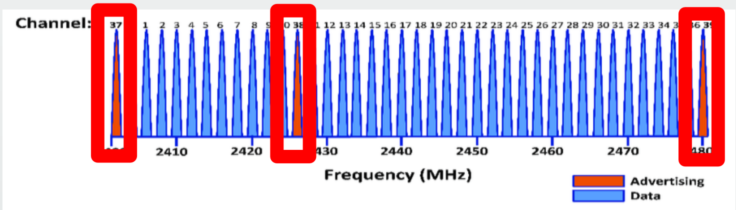
2. How to protect devices that do not support firmware modification?

Solution

Monitoring advertising messages externally

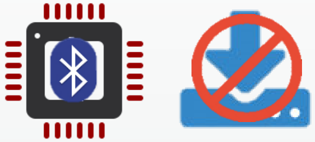
- We can detect spoofing attack if we can identify malicious advertising messages

BLE uses 3 advertising channels



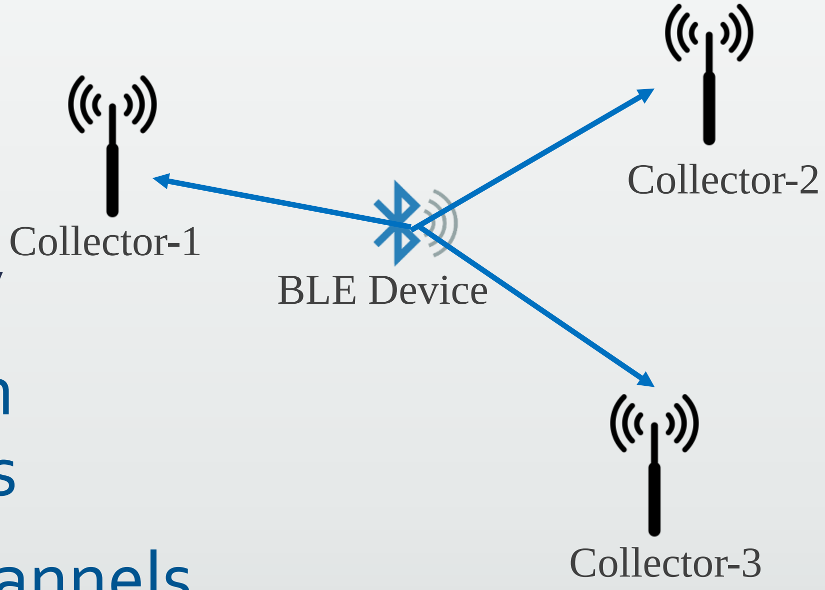
Spoofing detection : challenges and solutions

Challenge



2. How to protect devices that do not support firmware modification?

Solution

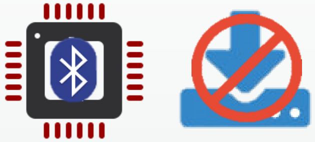


Monitoring advertising messages externally

- We can detect spoofing attack if we can identify malicious advertising messages
- Use 3 collectors cover all advertising channels

Spoofing detection : challenges and solutions

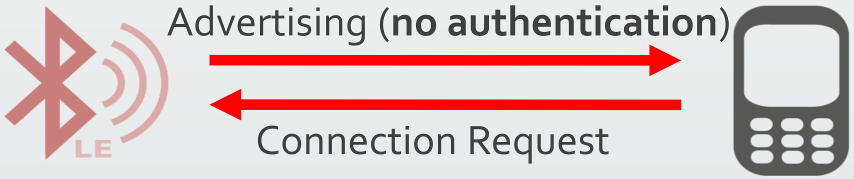
Challenge



2. How to protect devices that do not support firmware modification?

Solution

Monitoring advertising messages externally

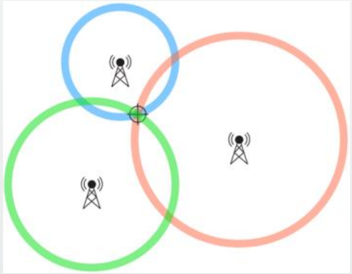


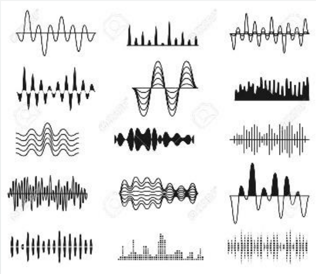
- We can detect spoofing attack if we can identify malicious advertising messages.
- Use 3 collectors cover all advertising channels
- But the attacker can **forge the advertising content**

Spoofing detection : challenges and solutions

Challenge

3. How to distinguish malicious advertising from the benign ones?





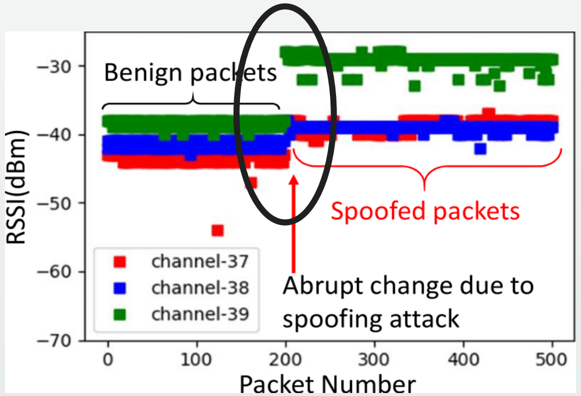
Physical features

- Received Signal Strength Indicator (RSSI), Location
- Carrier Frequency Offset (CFO), Device (radio) specific

Spoofing detection : challenges and solutions

Challenge

3. How to distinguish malicious advertising from the benign ones?



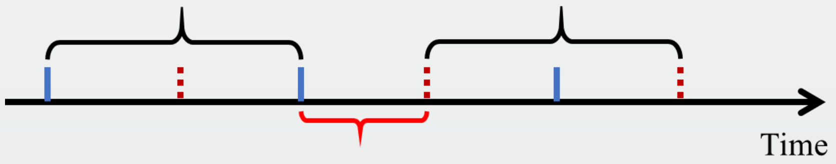
Physical features

- Received Signal Strength Indicator (RSSI), Location
- Carrier Frequency Offset (CFO), Device (radio) specific
- Monitoring these physical features (to detect abnormal changes)
- But it will introduce **high false positives**

Spoofing detection : challenges and solutions

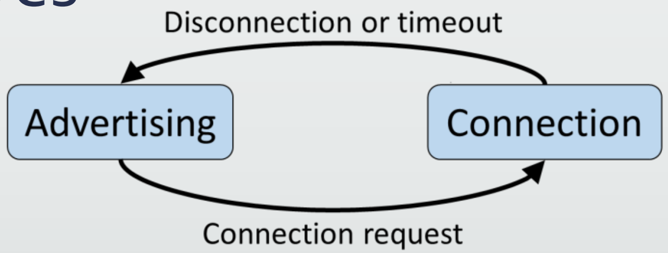
Challenge

3. How to distinguish malicious advertising from the benign ones?



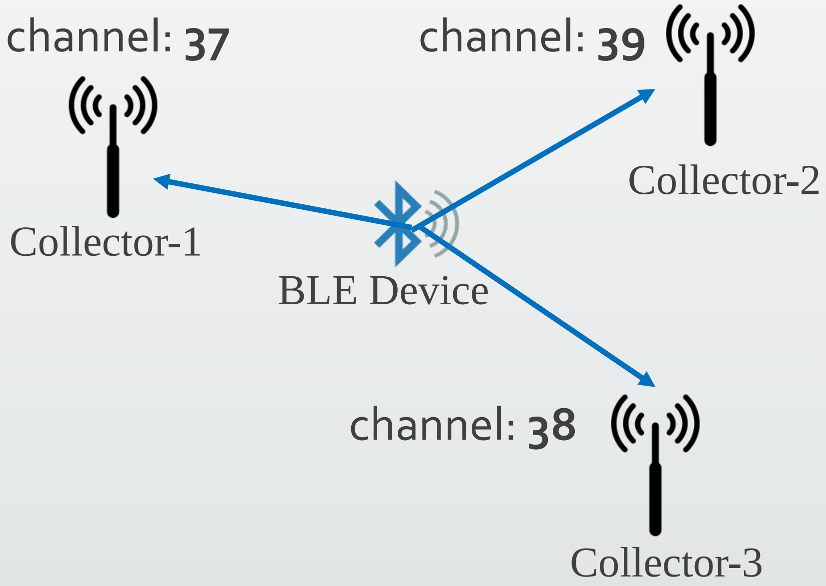
Protocol features to reduce false positives

- Advertising interval
- State transition
  - Use different detect mechanism in different state



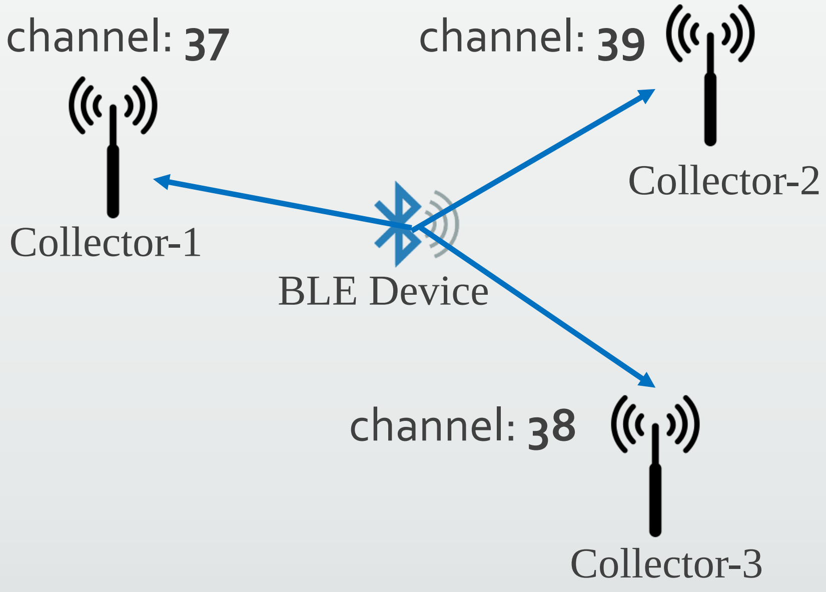
Detecting strong adversary (SDR)

- Software Defined Radio (SDR) can mimic some physical feature (CFO, RSSI)
- Moving target defense
  - CFO and RSSI varies on the same collector from different channels



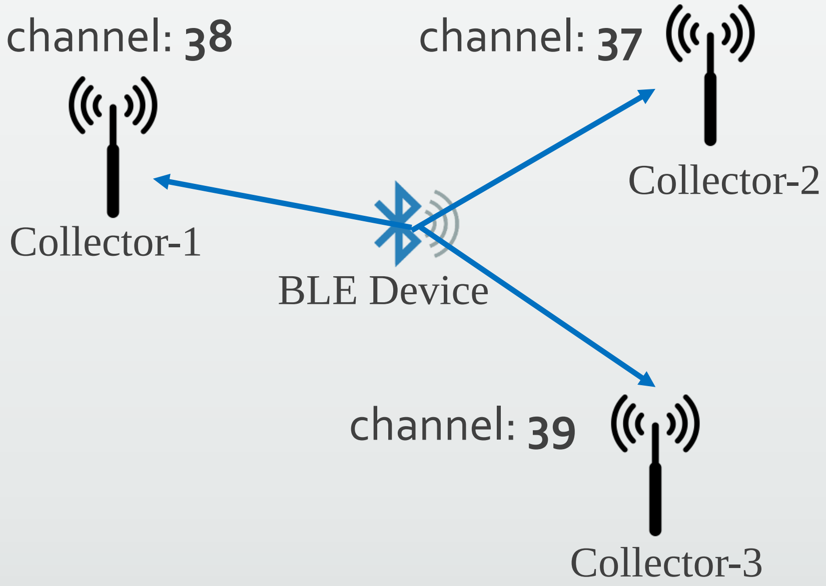
Detecting strong adversary (SDR)

- Software Defined Radio (SDR) can mimic some physical feature (CFO, RSSI)
- Moving target defense
  - CFO and RSSI varies on the same collector from different channels
  - Re-assign the channels after a random period

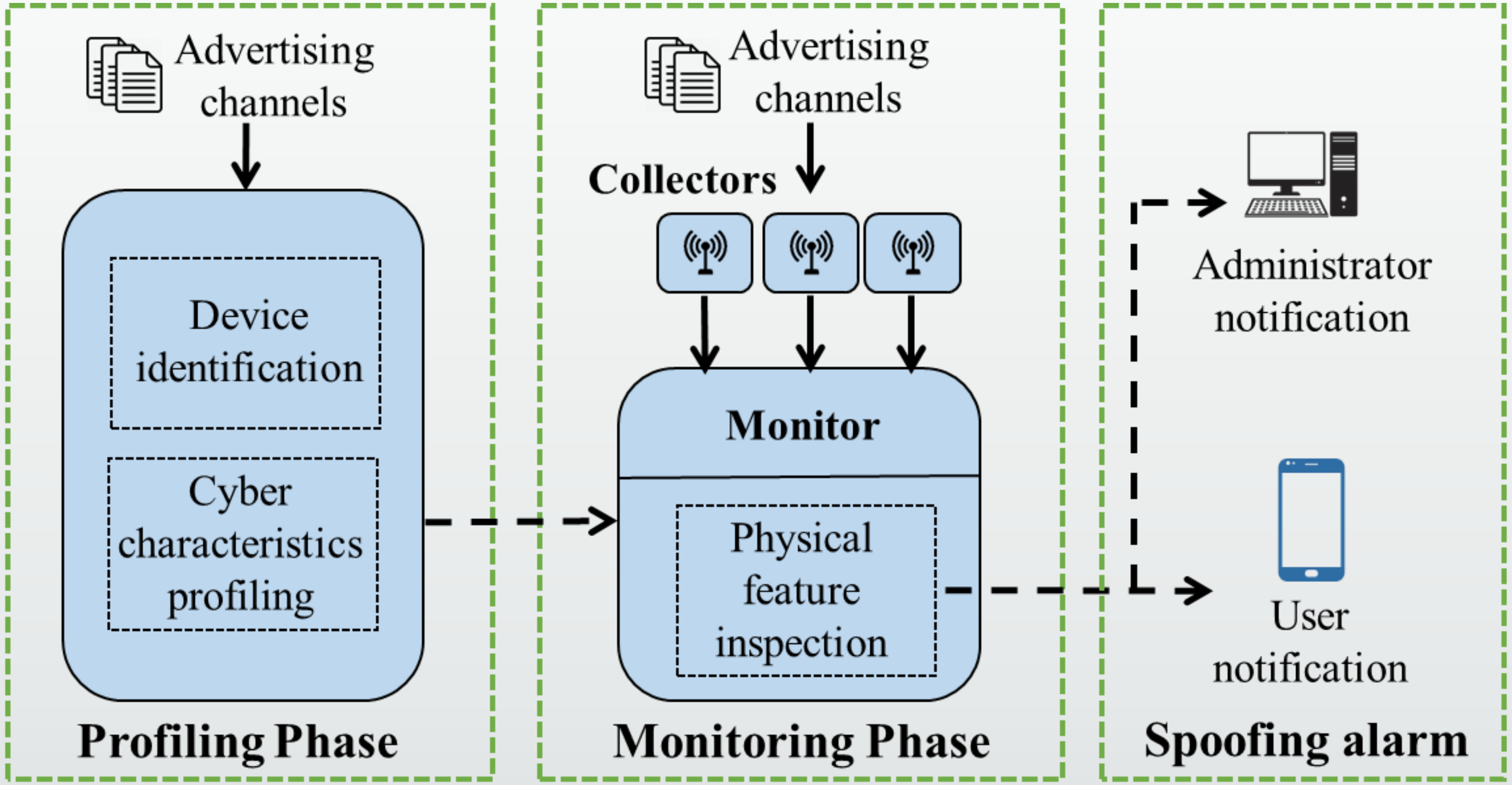


Detecting strong adversary (SDR)

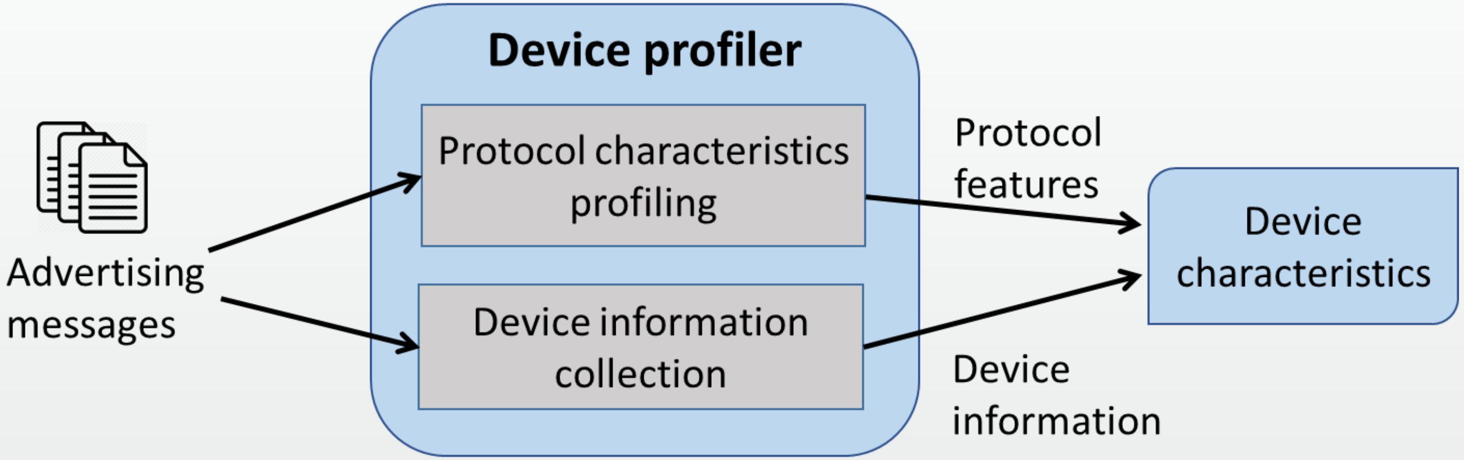
- Software Defined Radio (SDR) can mimic some physical feature (CFO, RSSI)
- Moving target defense
  - CFO and RSSI varies on the same collector from different channels
  - Re-assign the channels after a random period
  - Because of the randomness the attacker cannot mimic all these features



System design



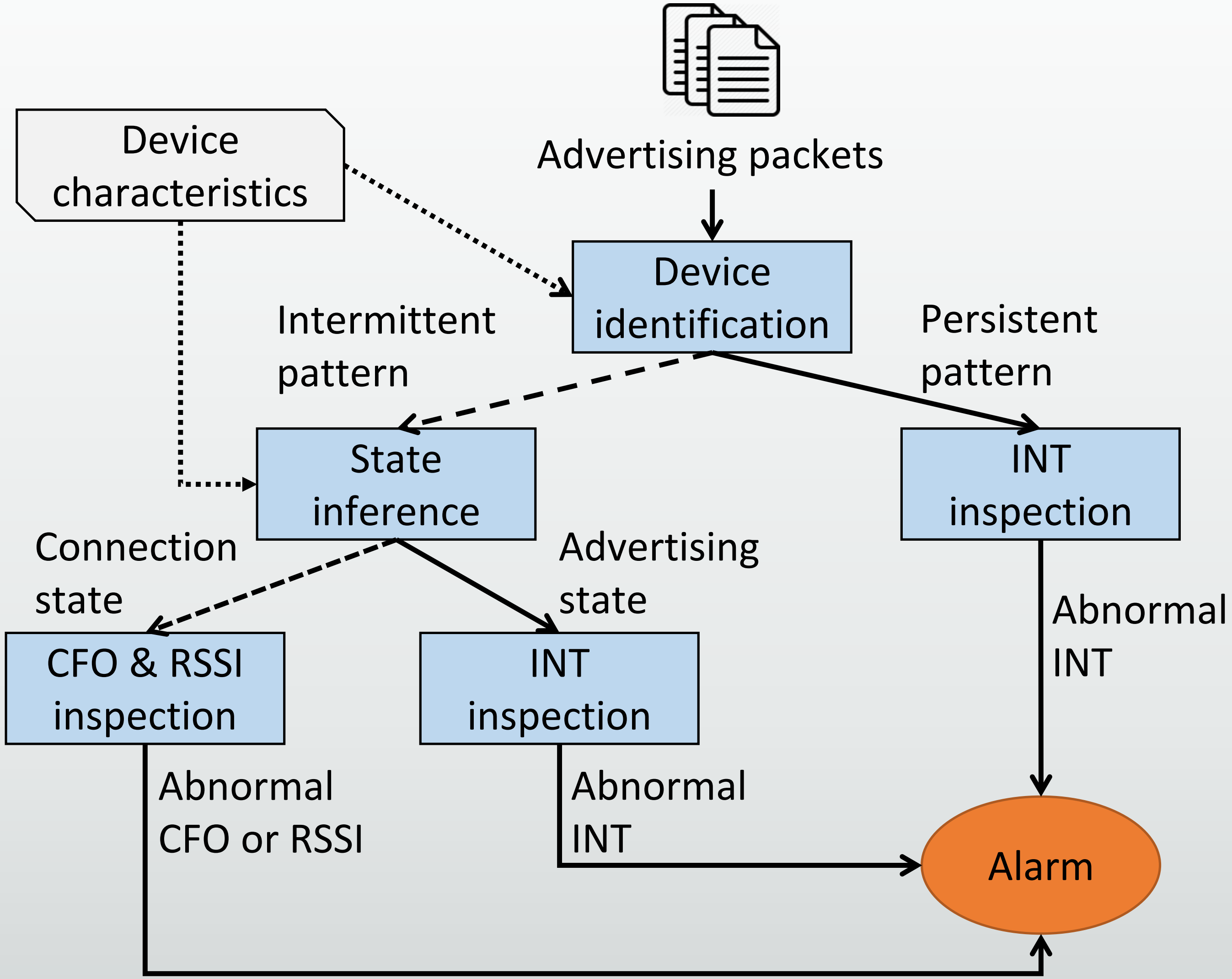
Architecture – profiling overview



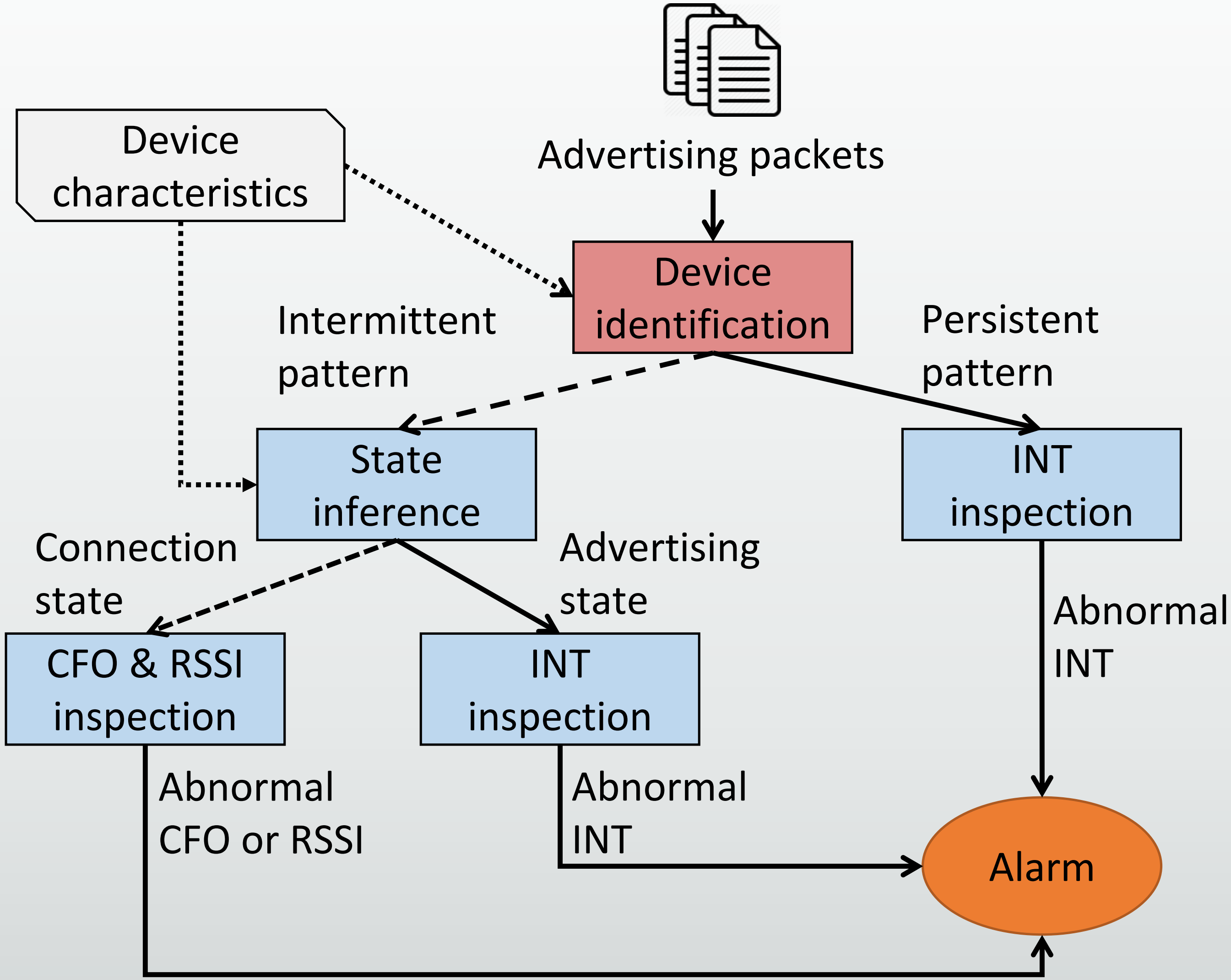
- Advertising pattern
- Advertising interval (INT)

Feature	Value
Device ID & Name	1, n097w
MAC Address	0xD1 76 A3 1A F4 7F
Advertising Data	0x06 09 4E 30 39 37 57
Advertising Pattern	Intermittent
Lower Bound of INT	1280 ms

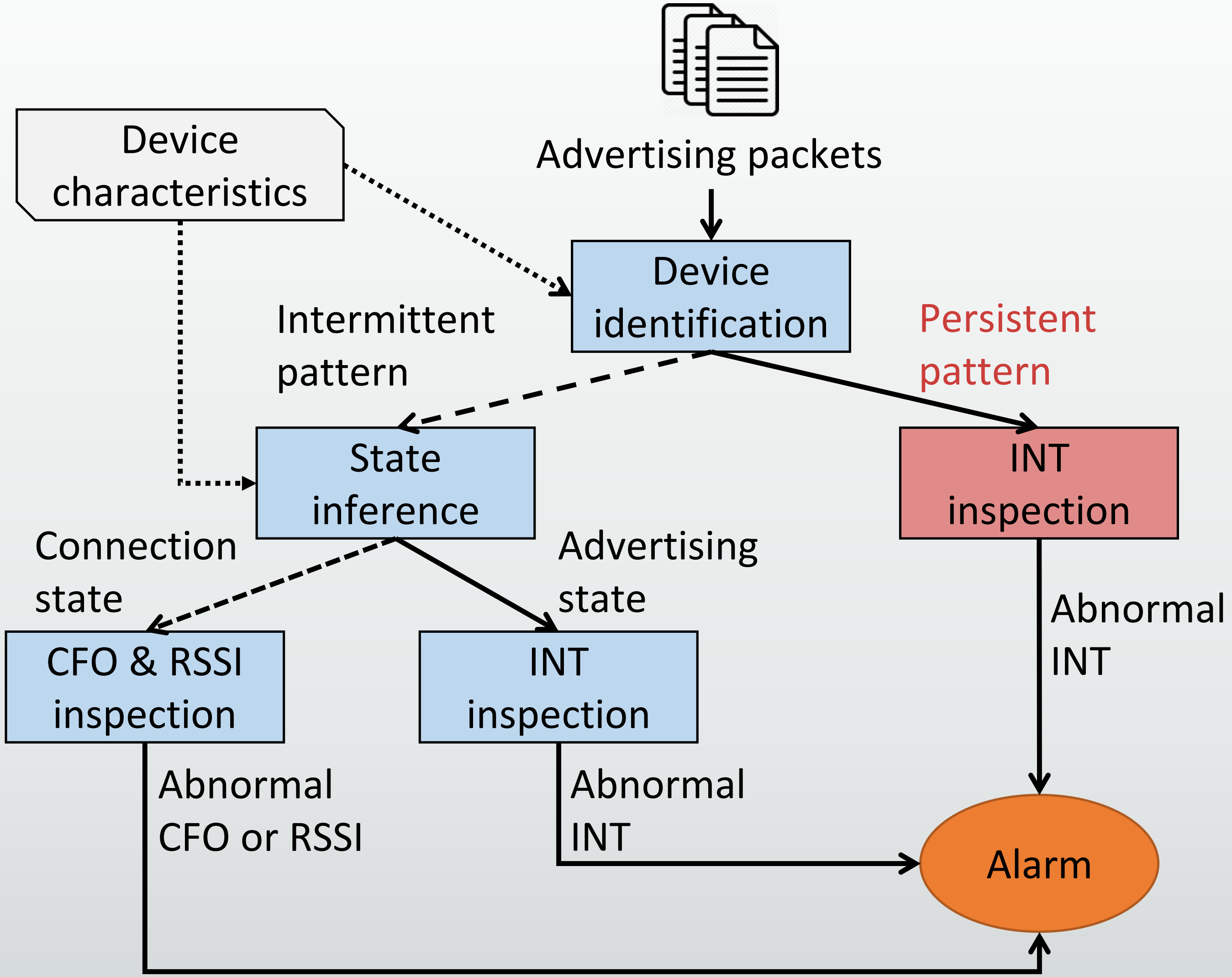
Architecture – monitoring overview



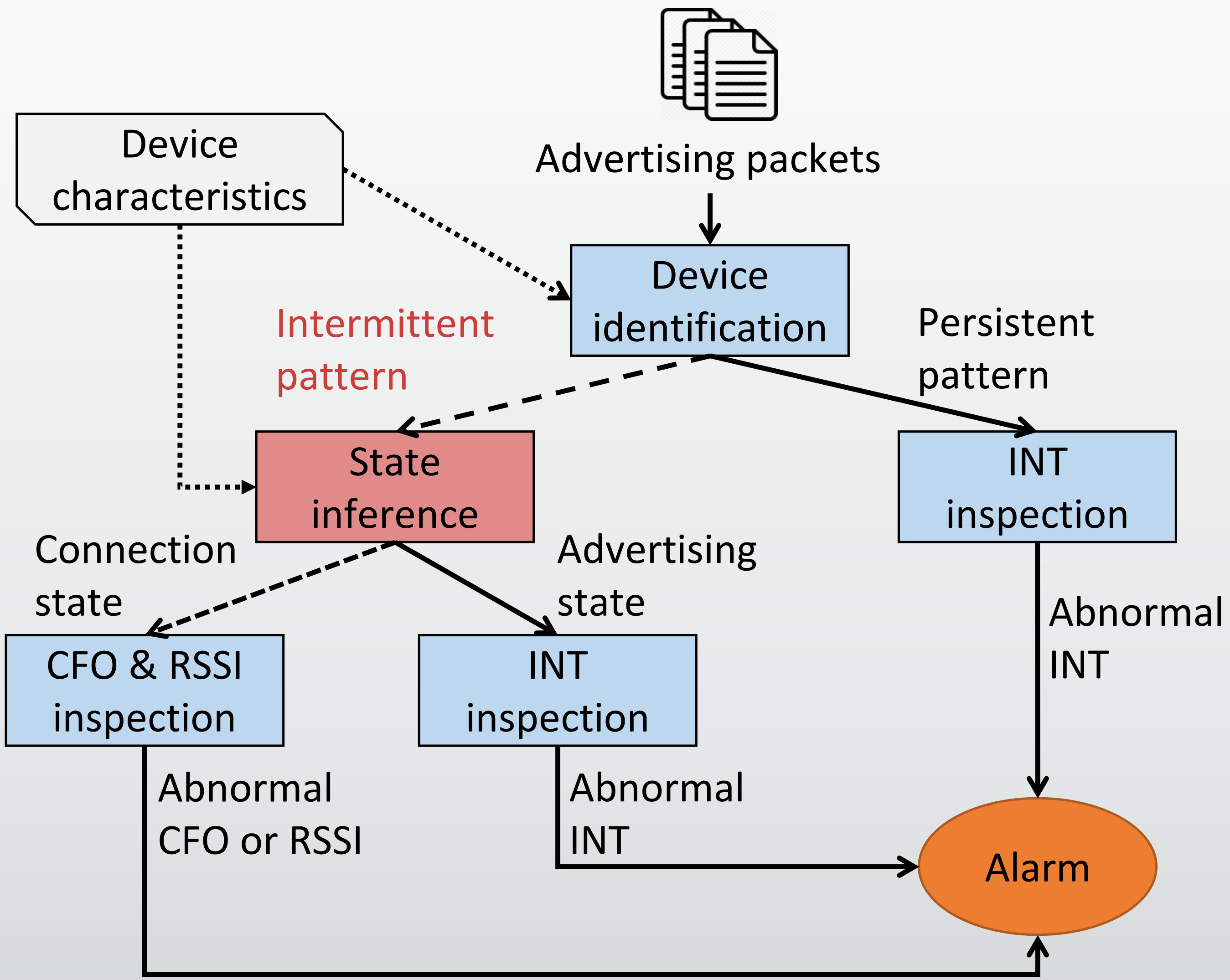
Architecture – monitoring overview



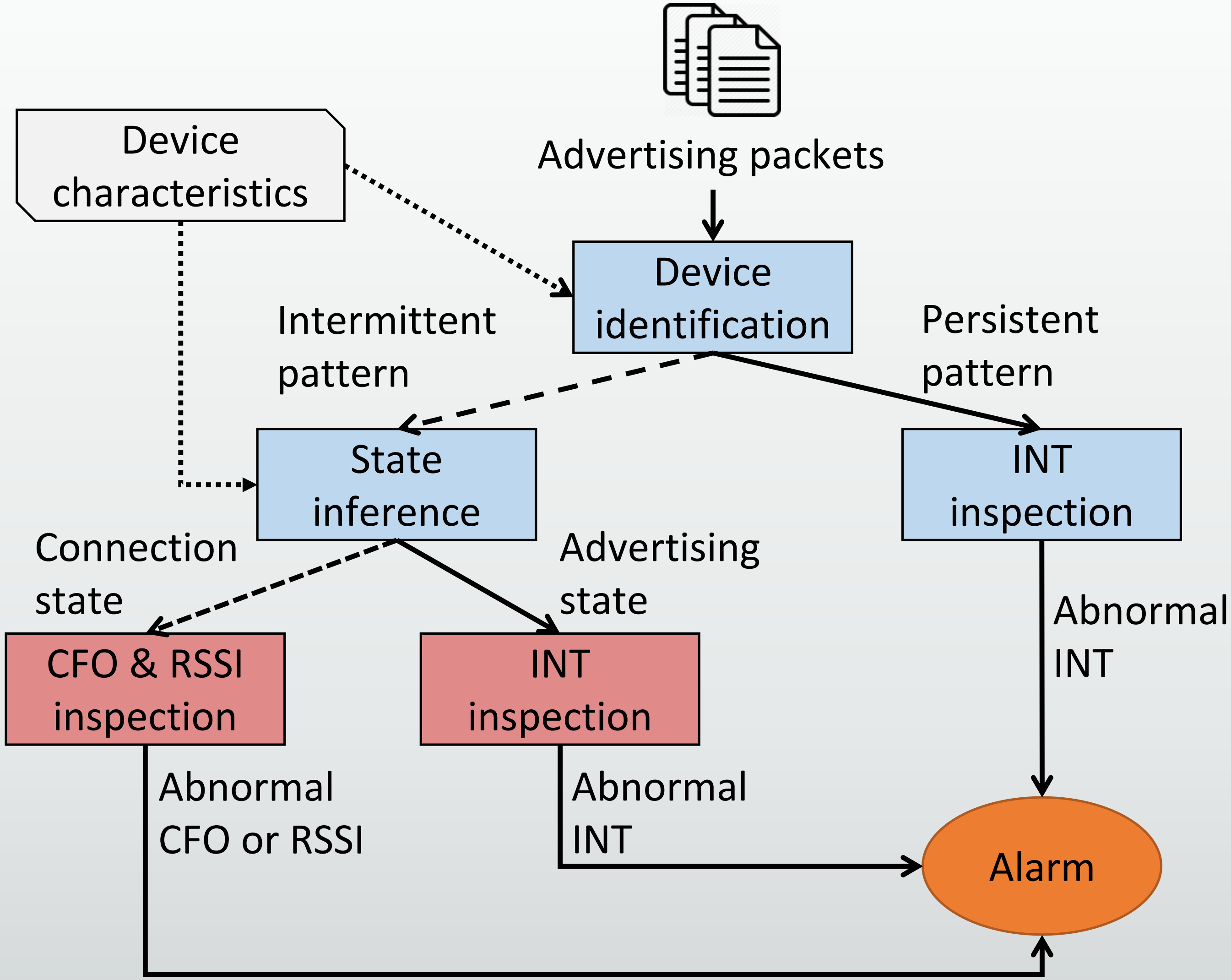
Architecture – monitoring overview



Architecture – monitoring overview

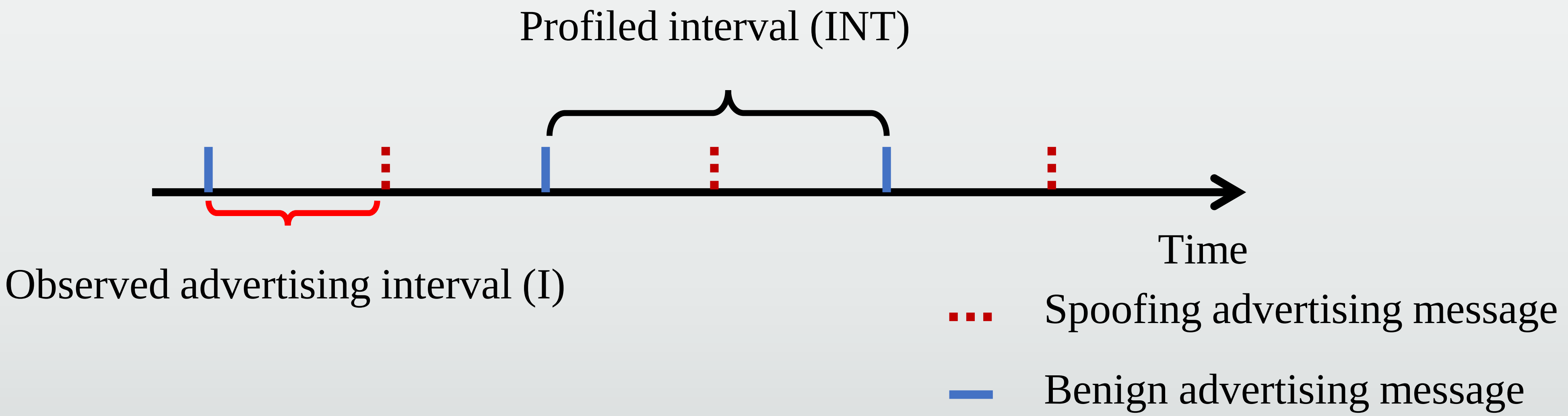


Architecture – monitoring overview



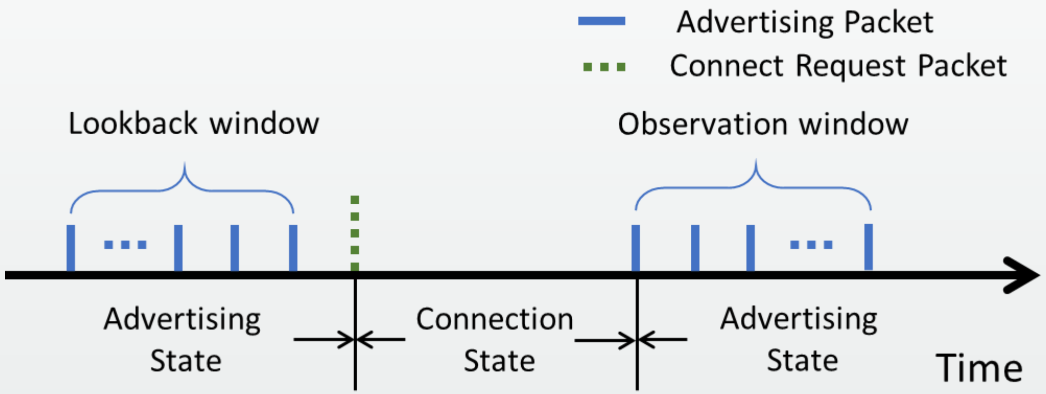
INT inspection

- Compare observed interval with the interval in the device's profile
  - Check whether I < INT



RSSI and CFO inspection

- Compare RSSI (CFO) feature of lookback window with observation window



- RSSI (CFO) feature of lookback window

$$f_c(x_i) = \frac{1}{\sigma_0\sqrt{2\pi}} \cdot e^{-\frac{(x_i-u_0)^2}{2\sigma_0^2}}$$

- RSSI (CFO) feature of observation window

$$L_C = \frac{1}{N_0}\sum_{i=1}^{N_0} -\log f_C(x_i)$$

$$L_c > \Gamma\ ?$$

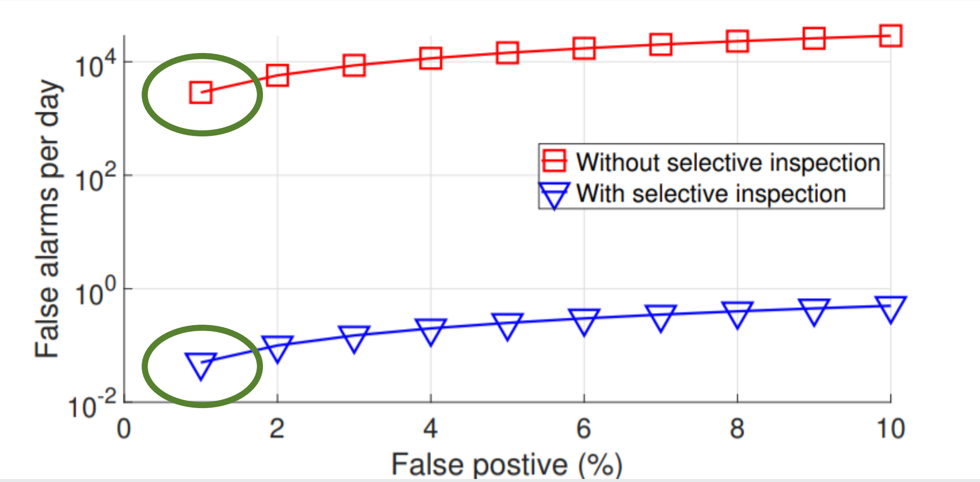
Effectiveness

Device ID	Device Name	Advertising Period (s)	Observation Window (s)	INT		CFO		RSSI		Overall	
				FP	FN	FP	FN	FP	FN	FP	FN
1	Nest Protect Smoke Detector	1.28	3.84	0.00	0.00	0.80	0.00	0.97	5.84	**1.76**	**0.00**
2	Nest Cam Indoor Camera	0.15	0.45	0.00	0.00	1.38	17.74	3.59	21.15	**4.92**	**3.69**
3	SensorPush Temperature Sensor	1.28	3.84	0.00	0.00	0.56	4.46	1.43	5.22	**1.98**	**0.23**
4	Tahmo Tempi Temperature Sensor	2.00	6.00	0.00	0.00	0.64	0.00	1.32	22.94	**1.95**	**0.00**
5	August Smart Lock	0.30	0.90	0.00	0.00	1.12	4.85	1.26	1.60	**2.37**	**0.08**
6	Eve Door&Window Sensor	1.28	3.84	0.00	0.00	0.77	8.17	1.64	1.46	**2.40**	**0.12**
7	Eve Button Remote Control	1.28	3.84	0.00	0.00	0.98	1.41	1.18	3.00	**2.15**	**0.04**
8	Eve Energy Socket	0.15	0.45	0.00	0.00	0.60	1.67	0.85	1.55	**1.44**	**0.03**
9	Ilumi Smart Light Bulb	0.10	0.30	0.00	0.00	0.88	14.28	1.48	15.73	**2.35**	**2.25**
	Average	0.87	2.61	0.00	0.00	0.86	5.84	1.52	8.72	**2.37**	**0.72**

- Low false positive (2.37%)
  - Only 1 false alarms in a week's heavy usage
- Low false negative (0.72%)
- Responsiveness (within 3 seconds)

Effectiveness of protocol feature

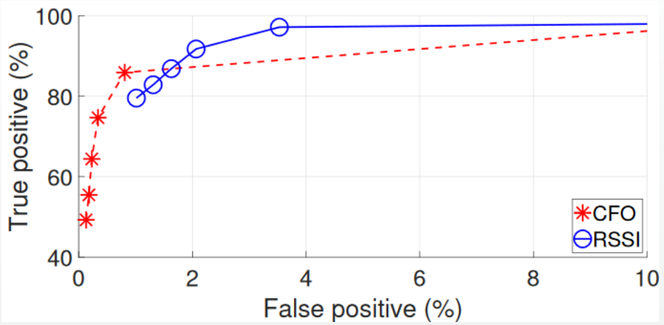
- The user uses a device 5 times per day
  - IoT device usage report[1]
- Reduce false positives significantly

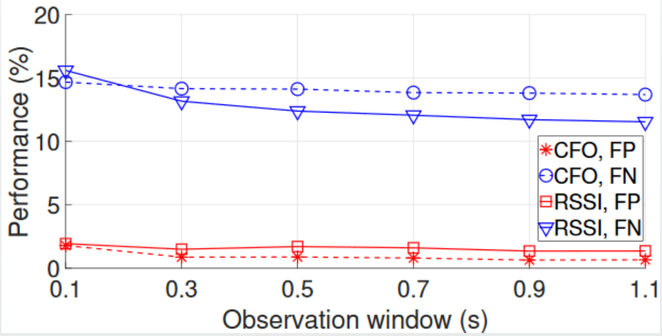


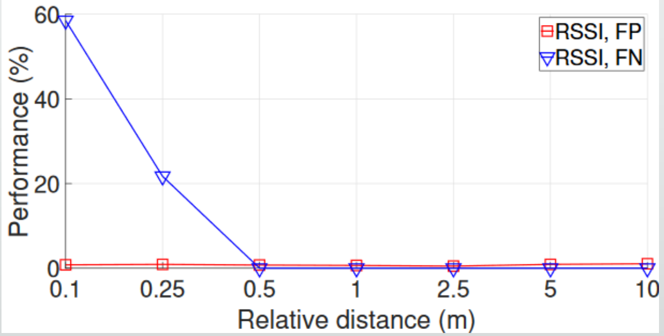
[1]. https://voicebot.ai/2018/04/02/smart-speaker-owners-use-voice-assistants-nearly-3-times-per-day/

Physical feature inspection with different parameters

- Different thresholds
  - Tradeoff between FP and FN
- Different observation window sizes
  - Use smaller size to be more responsive
- Impact of distance (RSSI)
  - 0.5 m is far enough to detect spoofing device

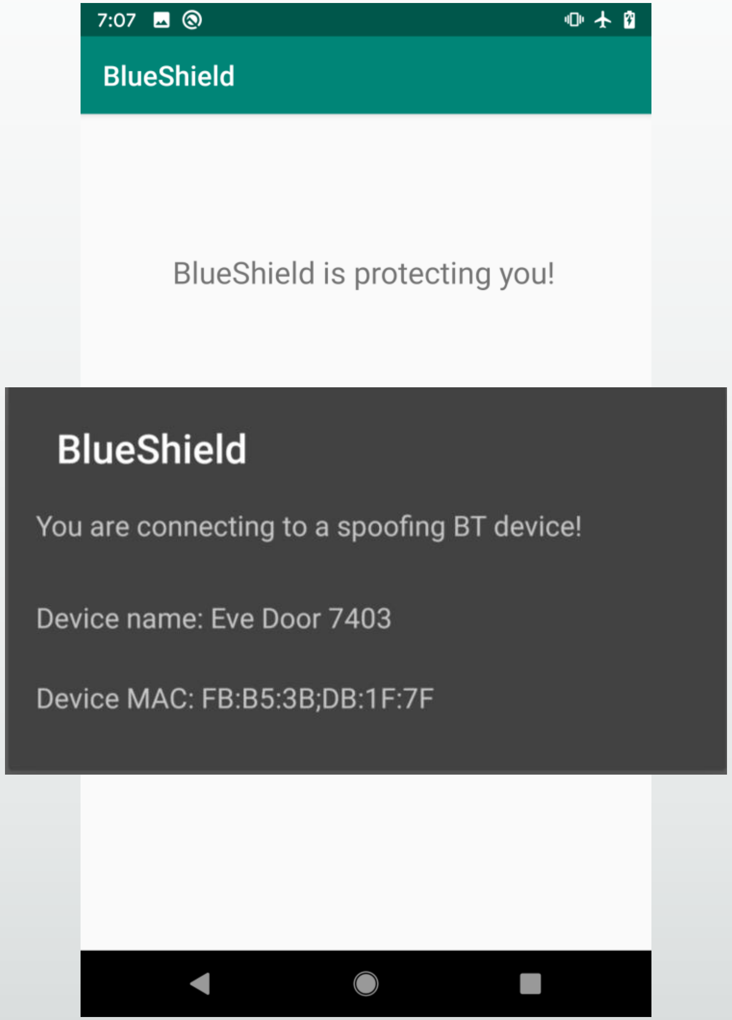




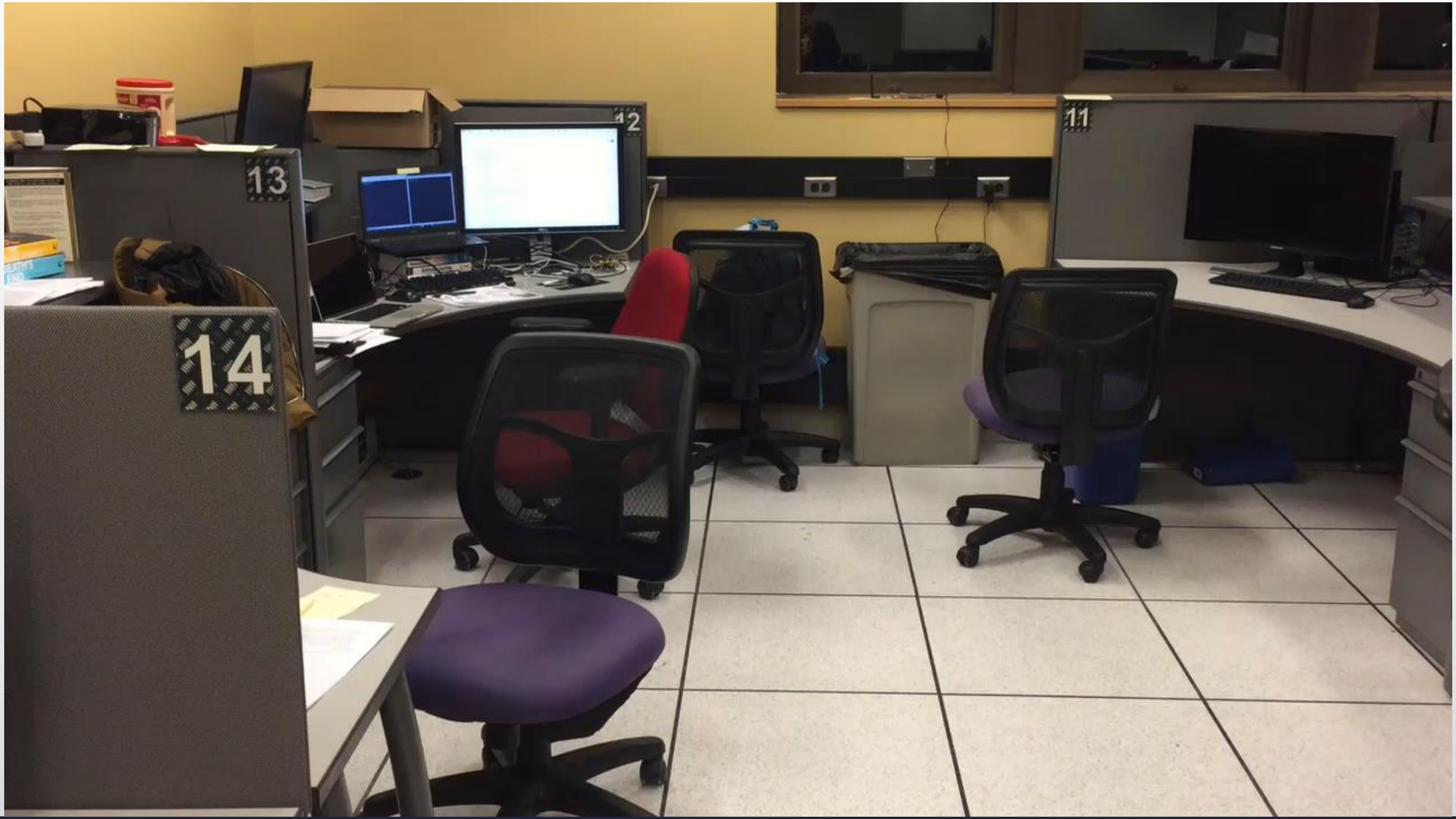


Administrator/user notification

- Notifying the administrator when spoofing attacks are detected.
- Notifying the user when the user is connecting to a malicious device (with an app installed).



Detection demo: (Eve) Door sensor



Summary

- BlueShield framework for detecting BLE spoofing attacks
  - Using robust **protocol and physical** features
  - Low-cost commodity hardware
  - Fully transparent to users
- Evaluation with 9 real-world devices
  - Low false positive and false negative rate
- Open source
  - https://github.com/allenjlw/BlueShield

Thank you! Questions?

This work was supported in part by ONR under Grant N00014-18-1-2674.

wu1220@purdue.edu