
Hocheol Shin¹, Yunmok Son¹, Youngseok Park¹,², Yujin Kwon¹, and Yongdae Kim¹

¹Korea Advanced Institute of Science and Technology
²NAVER Labs
Cyber-physical Systems and Sensors
Traditional Cyber Systems

External Network

Communication Interface

Control & In/Output Interface

System

Human Control

Other Systems
Cyber-physical Systems and Sensors
Sensor Spoofing and CPS Security
Sensor Spoofing and CPS Security

CPS Safety Features

CPS

Sensors

Actuators

Errors, Noises, Natural Disturbances

Surroundings
Sensor Spoofing and CPS Security

W/O CPS Security Features

CPS

Sensors

Controller

Actuators

Surroundings
Attack Vectors of Sensor Spoofing

**Regular Channel**: physical quantity sensed by the target
- Infusion Pump (Park et. al. WOOT’16)
- ABS (Shoukry et. al. CHES’13)

**Transmission Channel**: channel connecting sensor and backend
- Inject EMI to connecting wire
- Foo Kune et. al. S&P’13

**Sensor**:
- **Data/Signal Injection**
- **Transmission Channel**
- **Interference**

**Physical Quantities**

**Side Channel**: physical quantity *not* sensed by the target
- Expose gyroscope on a drone to acoustic noise of resonant freq.
- Son et. al. USENIX SEC’16
PyCRA

◆ **Physical Challenge-Response Authentication For Active Sensors Under Spoofing Attacks**
  - CCS’15
  - On detecting spoofing attempts against active sensors

◆ Claimed to be a generalizable & robust defense scheme

◆ Core Idea

![Diagram](attachment:PyCRA_diagram.png)

- Challenge with Randomness
- Response with Randomness
- Measured Entity
- No (negligible) Delay
PyCRA

◆ **Physical Challenge-Response Authentication For Active Sensors Under Spoofing Attacks**
  - CCS’15
  - On detecting spoofing attempts against active sensors

◆ Claimed to be a generalizable & robust defense scheme

◆ Core Idea

---

![Diagram of PyCRA](image)

Spoofing attack detected!
Outline

- PyCRA backgrounds
- Theoretical security analysis of PyCRA
- Experimental validation of the analytic study
- Open problem: “good” sensor spoofing defense
Passive Sensors

- Just receive incoming emissions
- Relatively short-distance applications
- Gas, PIR, microphone, camera, etc.

Active Sensors

- Emitter + passive sensor
- Measurement based on the difference of emitted and received signal
- Selective & remote sensing, ranging applications
- Radar, lidar, drop counter, etc.
PyCRA - Simple Detector

◆ Three schemes for spoofing detection
  - Simple detector, confusion phase, and $\chi^2$ detector
  - $\chi^2$ detector omitted for brevity

◆ Simple detector

[Step 1] Select a random time, $t_{\text{challenge}}$
[Step 2] Issue a physical challenge by setting $u(t_{\text{challenge}}) = 0$
[Step 3] If $y(t_{\text{challenge}}) > 0$, declare an attack
PyCRA - Confusion Phase

◆ Another trap to confuse the attacker
   1. Reduce the emitter signal under that of noise floor for a random period (Confusion Phase)
   2. Emitter signal is completely turned off to enter silent phase (LOW)
   3. Emitter is turned on again (HIGH)

- With a high probability, attacker will have a nonzero delay to distinguish confusion & silent phase
Attack Model

Victim System

- Active sensor system
- Both emitter and receiver cannot be shielded
- Analog-digital system: sampled and quantized

Attacker

- Can *transparently* receive victim emitter signal, and *transparently* inject spoofing signal
- Has more resources than the victim
- Trusted measured entity, non invasive, physical/computational delay
Sampling Race

◆ Fundamental vulnerability of PyCRA

Shared secret: timings of signal level changes

Emitter (Prover)  
Receiver (Verifier)  

Attacker Delay

Can know the secret soon after the challenge

Race exit condition

1. \( \exists \) Nontrivial lower bound of attacker delay
2. Victim’s time precision < lower bound of attacker delay

Secret has “very short” valid period
⇒ Arms race of time-precision
⇒ Sampling Race
Simple Detector Bypass

Spoofing detected

Spoofing not detected
Simple Detector Bypass

Victim’s sampling rate for robustness depends on the attacker’s agility

- But, what would be the maximum? How can we decide it?
- Is it practical to make all sensors super fast against uncertain attacks?
Simple Detector Bypass

- **Bypass condition**
  \[ T_A + t_{p,A} + t_{f,A} \leq T_V \]
Simple Detector Bypass

Bypass condition

\[ T_A + t_{p,A} + t_{f,A} \leq T_V \]

Same condition applies to bypass $\chi^2$ detector.

- Refer to the paper for details.
Confusion Phase Bypass

Assume timid attackers
- Stop spoofing with any anomalies in emitter signal
- Restart spoofing when emitter signal is fully restored
- Basically can be bypassed if simple detector can be bypassed
Experiment - Overview

◆ Conceptual experiment
  - Shows even crude circuitries can bypass 200kHz of sampling rate
  - PyCRA is valid when the race exit condition holds

◆ 200kHz as a criterion
  - Equivalent to 5μs of sampling interval
  - Given as sampling rate of high end MCU’s in PyCRA paper
  - More than enough for most sensing applications
Real-world IR drop counter for medical infusion pump (JSB-1200)

Drop counter
- Non-contact measurement of liquid volume

Experiment - Target Active Sensor
Experiment - Setup

**Experiment Process**

1. Arduino 1 turns off victim emitter @ T1
2. Arduino 2 turns off attacker emitter as soon as possible @ T2
3. Measure T2-T1
Experiment - Comparator

◆ Attackers only have to catch the *falling edge*

◆ Comparator
  - Compare $V_{\text{IN}}$ with $V_{\text{REF}}$
  - Binary output
  - Used the most basic type

![Comparator Diagram]

- $V_{\text{OUT}}$ transitions when $V_{\text{IN}}$ crosses $V_{\text{REF}}$
- $V_{\text{OUT}}$ is high when $V_{\text{IN}} > V_{\text{REF}}$
- $V_{\text{OUT}}$ is low when $V_{\text{IN}} < V_{\text{REF}}$
Experiment - Comparator

- Implemented with MCP602 op-amp IC
- Falling transition time reduced to 1/5
- Dedicated comparator ICs have much shorter switching time
Experiment - Overall Delay

◆ Reducing processor-side delay
  - Arduino analog read/write API: AnalogRead, AnalogWrite — 10kHz
    \( \rightarrow \) Too slow to achieve 5\( \mu \)s
  - Use of comparator \( \rightarrow \) faster digitalRead/Write
  - Used digitalRead/WriteFast library
Experiment - Overall Delay

◆ Reducing processor-side delay
  - Arduino analog read/write API: AnalogRead, AnalogWrite — 10kHz → Too slow to achieve 5μs
  - Use of comparator → faster digitalRead/Write
  - Used digitalRead/WriteFast library

◆ Overall delay of 2.8 μs achieved
  - Measured @ input terminals of two LEDs
  - Equivalent to 358kHz of sampling rate
  - 80% faster than 200kHz
Constraint on Manufacturing

◆ Adoption of high sampling rate $\rightarrow$ Resources!
  - More power & cost required
  - Hard to adopt sampling rate $>>$ required for sensing application

◆ Uneven Competition
  - A secure sensor must satisfy race exit condition which depends on possible spoofer implementation
Sample Rate and Additional Delay

◆ Attacker vs. victim cost
  - Attacker cost: attacker sampling rate + additional delay
  - Victim cost: victim sampling rate
  - Both depends on sampling rate
    → Equal competition when additional delay is negligible

◆ Plot bypass condition with sampling rate
  - \( T_A + t_{p,A} + t_{f,A} \leq T_V \)
  - \( F_A \leftarrow \frac{1}{T_A}, F_V \leftarrow \frac{1}{T_V}, C \leftarrow t_{p,A} + t_{f,A} \)
  - \( F_A \geq -\frac{1}{C^2} \times \frac{1}{F_V-1/C} - \frac{1}{C} \)
Near Linear PyCRA Security

PyCRA

- Not much adversarial penalty, near race exit point
- Near y=x (Equal)

Crypto

- Fast increase by powers of 2
- Trendline
- Keysize: 32/64/128/256 bit
Race Exit Points and Additional Delay

Additional delay & race exit point

- $F_A \geq -\frac{1}{C^2} \times \frac{1}{F_V^{-1/C}} - \frac{1}{C}$
- Reciprocal relation
- Small decrease in additional delay $\rightarrow$ Large increase in race exit point
Conclusion

◆ PyCRA, first work on generalizable and robust defense
  - Practicality should be implicitly assumed.
  - Under sampling race: not robust
  - After race exit: not practical for some applications → not generalizable
Conclusion

◆ PyCRA, first work on generalizable and robust defense
  - Practicality should be implicitly assumed.
  - Under sampling race: not robust
  - After race exit: not practical for some applications $\rightarrow$ not generalizable

◆ Requirements for “good” sensor spoofing defense solution
Conclusion

◆ PyCRA, first work on generalizable and robust defense
  - Practicality should be implicitly assumed.
  - Under sampling race: not robust
  - After race exit: not practical for some applications → not generalizable

◆ Requirements for “good” sensor spoofing defense solution

No solution satisfies all three properties: robustness, generalizability, and practicality

Defense of active sensor spoofing → Remains as an open problem