THE LEAKY ACTUATOR: A PROVABLY-COVERT CHANNEL IN CYBER PHYSICAL SYSTEMS

Amir Herzberg
amir.herzberg@uconn.edu
University of Connecticut
Storrs, USA

Yehonatan Kfir
yehonatank@gmail.com
Bar-Ilan University
Ramat-Gan, Israel

ACM Workshop on Cyber-Physical Systems Security & Privacy (CPS-SPC), 2019
INTRODUCTION

- **Cyber Physical Systems (CPS)** - Smart systems that include networks of physical and computational components, all aimed to govern a physical process.
- **Examples**: Nuclear Plants, Power Generations, Water Plant, Transportations.
- Critical for our life
- Built from large number of devices:
  - Sensors, Actuators, Controllers...
Devices are chosen based on **sufficient specification** and **lowest cost**.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Device A</th>
<th>Device B</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Quality</td>
<td>✔️ High Quality</td>
<td>✔️ Sufficient Quality</td>
</tr>
<tr>
<td>Price</td>
<td>Expensive</td>
<td>Cheap</td>
</tr>
</tbody>
</table>
Devices are chosen based on **sufficient specification** and **lowest cost**.

<table>
<thead>
<tr>
<th></th>
<th>Device A</th>
<th>Device B</th>
<th>Malicious</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification</td>
<td>High Quality</td>
<td>Sufficient Quality</td>
<td>Sufficient Quality</td>
</tr>
<tr>
<td>Price</td>
<td>Expensive</td>
<td>Cheap</td>
<td>Very Cheap</td>
</tr>
</tbody>
</table>

**Supply Chain Attack:** Attacker can offer a malicious device with sufficient quality.

**Attacker Goal:** To cause damage, by deploying its own malicious device.
ATTACKER CHALLENGE - 1

In order to cause damage, **multiple devices should co-operate**.
ATTACKER CHALLENGE - 1

- In order to cause damage, **multiple devices should co-operate**.
- Regulation today requires **isolation inside the CPS**
- There is no **direct communication between** sensor and actuators.
ATTACKER CHALLENGE

How to communicate between malicious devices?
FEEDBACK CONTROL LOOP

Feedback control loops are the main method used to stabilize physical values in CPS.

Threshold-controller
- Actuator with two possible commands to increase / decrease the physical value: $U^{INC} / U^{DEC}$
- Two thresholds: $T_{high}, T_{low}$

When the sensor measurements reach $T_{high} / T_{low}$, the controller changes its output to decrease / increase the signal.
Feedback control loops are the main method used to stabilize physical values in CPS.

Threshold-controller
- Actuator with two possible commands to increase / decrease the physical value: $U^{\text{INC}} / U^{\text{DEC}}$
- Two thresholds: $T_{\text{high}}$, $T_{\text{low}}$

When the sensor measurements reach $T_{\text{high}} / T_{\text{low}}$, the controller changes its output to decrease / increase the signal.
Feedback control loops are the main method used to stabilize physical values in CPS. A threshold controller with two possible commands to increase / decrease the physical value: $U_{INC}$ / $U_{DEC}$. Two thresholds: $T_{high}$, $T_{low}$. When the sensor measurements reach $T_{high}$ / $T_{low}$, the controller changes its output to decrease / increase the signal.
FEEDBACK CONTROL LOOP

Feedback control loops are the main method used to stabilize physical values in CPS.

- Threshold-controller
  - Actuator with two possible commands to increase / decrease the physical value: $U_{INC}$ / $U_{DEC}$
  - Two thresholds: $T_{high}$, $T_{low}$

When the sensor measurements reach $T_{high}$ / $T_{low}$, the controller changes its output to decrease / increase the signal.
FEEDBACK CONTROL LOOP

- Widely used in: phase controller, current limiters, pH controllers.

Periodic Physical Process
- The process value continuously iterates and passes the thresholds: $T_{\text{high}}$, $T_{\text{low}}$.
- The actuator's input changes between $U_k$ and $U_{\text{DEC}}$ periodically.
- We denote the $i$th transition of the actuator's output by $i$. 

![Diagram of pH Level with transitions labeled i=1, i=2, i=3]
LEAKY-ACTUATOR COMMUNICATION METHOD

Upon receiving a command, the actuator changes its output state with some random delay.

Actuator's delay influences the process, which is monitored by the sensor.

Attacker will use the delay for signaling:
- Fast / Slow response times can signal bits 0/1.

Water level after the same time, for different actuator's response times,
Upon receiving a command, the actuator changes its output state with some random delay. Actuator’s delay influences the process, which is monitored by the sensor. Attacker will use the delay for signaling:

- Fast response times, can signal bit 0/1.

Water level after the same time, for different actuator’s response times:

- Fast: $T_{high}$
- Slow: $T_{high}$

Uses a classifier, based on 8 measurable features of the process.
THE RECEIVER

The receiver measures a set of physical properties of the physical value \( z_k \).

Properties calculated over a set of \( \{z_k\} \):

- Starting at the first \( z_k \) that pass one of the thresholds \( T_{\text{high}} \), \( T_{\text{low}} \).
- Ends on the next threshold.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Threshold Passed</td>
<td>The last threshold passed by the physical process. The values of this feature can be ( T_{\text{high}} ) or ( T_{\text{low}} ).</td>
</tr>
<tr>
<td>Set Size</td>
<td>The number of samples in the set of ( z_k ).</td>
</tr>
<tr>
<td>Max ( z )</td>
<td>The maximal value of ( z_k ) in the set.</td>
</tr>
<tr>
<td>Min ( z )</td>
<td>The minimal value of ( z_k ) in the set.</td>
</tr>
<tr>
<td>Linear Approximation Coefficients</td>
<td>The linear approximation of ( z_k ) in the set. Formally, there are three coefficients in this feature. Two coefficients ( A_{1,0}, A_{1,1} ) represent the approximated function ( A_{1,1} \cdot x + A_{1,0} ) of the values ( z_k ), and a third coefficient ( err_1 ) represents the least-mean-square error of the approximated function.</td>
</tr>
<tr>
<td>( 2^{nd} )-order Polynomial Approximation Coefficients</td>
<td>The second order approximation of ( z_k ) in the set. Formally, there are four coefficients in this feature. Three coefficients ( A_{2,0}, A_{2,1}, A_{2,2} ) represent the approximated function ( A_{2,2} \cdot x^2 + A_{2,1} \cdot x + A_{2,0} ) of the values ( z_k ), and a fourth coefficient ( err_2 ) represents the least-mean-square error of the approximated function.</td>
</tr>
</tbody>
</table>

Table 1: Features used by the covert receiver classifier
ATTACKER CHALLENGES

How to communicate between malicious devices?

Delay
ATTACKER CHALLENGES

How to communicate between malicious devices?

Delay

Creates Anomaly in the CPS behavior...
ATTACKER CHALLENGE - 2

A lot of works on anomalies detections in CPS.

Communication Network Anomalies


Physical Anomalies – malicious sensor reporting / malfunctioning actuator

ATTACKER CHALLENGES

- How to communicate **between** malicious devices?
- How to avoid detection?

ATTACKER CHALLENGES

- How to communicate **between** malicious devices?
- How to avoid detection?

Covert Channel

Delay

Creates Anomaly in the CPS behavior...

Communication channels are critical for operating malwares.

"Covert" - using some "unmonitorred" channels

- Encoding information using light brightness ("Extended functionality attacks on IoT devices: The case of smart lights", Shamir et. al. 2016)
- Packet headers ("Embedding Covert Channels into TCP/IP", Murdoch et. al., 2005)
- Acoustic emissions of a motor ("Process-aware covert channels using physical instrumentation in cyber-physical systems", Krishnamurthy et. al. 2018)
- And more...

Monitoring the "unmonitorred" property, reveals the communication channel.

PROVABLE COVERT CHANNELS

“Provable-Covert” –
- No secret property
- Proving that it is impossible to detect the channel (under well defined assumptions)

\[
\Pr(D(\text{Mal.}) = \text{Mal.}) \approx \Pr(D(\text{Mal.}) = \text{Mal.})
\]

Provable channels were presented in the past, for IP networks:

How to (provably) avoid detection?
The provably-covert channel is based on two basic observations about actuators:

- The **response time is random**, derived from some (known) distribution.
- There are different **benign** types of actuators in the market:
  - Low response time (‘fast / high quality actuators’)
  - Long response time (‘slow actuators’).
LEAKY-ACTUATOR COVERT CHANNEL

A leaky actuator is using an internal fast actuator. It adds a pseudo-random delay from two possible delay distributions, \( P_0 \) and \( P_1 \). For transmitting bit \( b_i \), the leaky actuator chooses the added delay distribution to be \( P_{b_i} \). In random choice of \( b_i \), the delay distribution of the leaky actuator will be identical to that of the benign actuator.

<table>
<thead>
<tr>
<th>( b_i )</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{b_i} )</td>
<td>( P_0 )</td>
<td>( P_0 )</td>
<td>( P_1 )</td>
<td>( P_0 )</td>
<td>( P_1 )</td>
<td>( P_1 )</td>
</tr>
</tbody>
</table>
ATTACKER CHALLENGES

- How to communicate **between** malicious devices?
- **How to avoid detection?**

| Pseudo-random | Delay | Design | Receiver | Evaluation |
Deployed all its devices with a secret key $\kappa \in \{0,1\}^t$

$M$ – Message to send

$m$ – Encoded Message

$i$ – Transitions counter

Pseudo-random Bit Generator

\[ b_i = PRF_k(i) \oplus m_i \]
THE LEAKY ACTUATOR: BIT GENERATOR

Pseudo-random Bit Generator

**Inputs:** The message $M$, the key $\kappa$, and the transition index $i$.

**Output:** $b_i$ - a pseudo-random bit, based on the $i$th message bit $M_i$, and the output of the PRF $\text{PRF}_k(i)$.

Message $M$ encoded with error correction code $m$.

- Decreases bit error rate.
- First $I_{CAL}$ bits are all 0 – will be used for calibrating the sensor.
- The pseudo-random bit generator ensures that the delay is indistinguishable from random (from PRF property, see paper).

\[ \Pr(D(\text{Mal.}) = \text{Mal.}) \approx \Pr(D(\text{Mal.}) = \text{Mal.}) \]
THE RECEIVER

**Synchronization Assumption:** (relaxed in the paper) $i^T \equiv i^R$

**The Goal:** To identify when the delay is derived from $P_0$ and when from $P_1$.
- Detect $P_{b_i}$: Conclude $b_i \Rightarrow m_i^R = b_i \oplus PRF_k(i)$

**The Challenge:**
- The delay cannot be measured directly.
- The delay has an unknown impact on the physical process.

**The Solution:**
- Use the calibration period to train a classifier.
  - The first $K$ are all $0 \Rightarrow m = 0 \Rightarrow P_{b_i} = 0 \oplus PRF_k(i) \Rightarrow P_{b_i}$
  - Different delays present different impact on the physical process.
- Measure features of the physical process. Label them with the (known) calculated $P_{b_i}$
  - After calibration period, use the trained classifier to “guess” whether the delay was derived from $P_0$ or $P_1$. 

**Evaluation**
**(Receiver Design)**
EVALUATION

How good is the receiver in intercepting the leaky-actuator bits?

- **Theoretical**: Channel Capacity.

- **Practical**: Bit-error-rate of our receiver design.
EVALUATION: CHANNEL CAPACITY

Channel Capacity – highest information rate that can be achieved.

Evaluated two classifiers: KNN and Decision Tree (DT).

Different message length $|m|$ and calibrations periods $I_{CAL}$.

Results: About 0.5 bit of information on every transition.

<p>| $|m| = 10,000$ | $|m| = 50,000$ | $|m| = 100,000$ |
|----------------|----------------|----------------|</p>
<table>
<thead>
<tr>
<th>Classifier</th>
<th>$I_{CAL}$</th>
<th>$p$</th>
<th>$C$</th>
<th>Classifier</th>
<th>$I_{CAL}$</th>
<th>$p$</th>
<th>$C$</th>
<th>Classifier</th>
<th>$I_{CAL}$</th>
<th>$p$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1%</td>
<td>0.28</td>
<td>0.15</td>
<td></td>
<td>0.1%</td>
<td>0.12</td>
<td>0.46</td>
<td></td>
<td>0.1%</td>
<td>0.154</td>
<td>0.38</td>
<td></td>
</tr>
<tr>
<td>0.5%</td>
<td>0.12</td>
<td>0.47</td>
<td></td>
<td>0.5%</td>
<td>0.12</td>
<td>0.46</td>
<td></td>
<td>0.5%</td>
<td>0.13</td>
<td>0.44</td>
<td></td>
</tr>
<tr>
<td>kNN</td>
<td>1%</td>
<td>0.12</td>
<td>0.47</td>
<td>kNN</td>
<td>5%</td>
<td>0.11</td>
<td>0.48</td>
<td>kNN</td>
<td>1%</td>
<td>0.129</td>
<td>0.45</td>
</tr>
<tr>
<td>5%</td>
<td>0.11</td>
<td>0.49</td>
<td></td>
<td>10%</td>
<td>0.11</td>
<td>0.49</td>
<td></td>
<td>5%</td>
<td>0.128</td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td>10%</td>
<td>0.11</td>
<td>0.49</td>
<td></td>
<td>10%</td>
<td>0.11</td>
<td>0.49</td>
<td></td>
<td>10%</td>
<td>0.128</td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td>0.1%</td>
<td>0.2</td>
<td>0.26</td>
<td></td>
<td>0.5%</td>
<td>0.13</td>
<td>0.43</td>
<td></td>
<td>0.1%</td>
<td>0.154</td>
<td>0.38</td>
<td></td>
</tr>
<tr>
<td>0.5%</td>
<td>0.13</td>
<td>0.44</td>
<td></td>
<td>0.5%</td>
<td>0.11</td>
<td>0.48</td>
<td></td>
<td>0.5%</td>
<td>0.13</td>
<td>0.44</td>
<td></td>
</tr>
<tr>
<td>DT</td>
<td>1%</td>
<td>0.12</td>
<td>0.47</td>
<td>DT</td>
<td>1%</td>
<td>0.11</td>
<td>0.49</td>
<td>DT</td>
<td>1%</td>
<td>0.126</td>
<td>0.45</td>
</tr>
<tr>
<td>5%</td>
<td>0.11</td>
<td>0.49</td>
<td></td>
<td>5%</td>
<td>0.1</td>
<td>0.5</td>
<td></td>
<td>5%</td>
<td>0.126</td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td>10%</td>
<td>0.11</td>
<td>0.49</td>
<td></td>
<td>10%</td>
<td>0.1</td>
<td>0.5</td>
<td></td>
<td>10%</td>
<td>0.126</td>
<td>0.45</td>
<td></td>
</tr>
</tbody>
</table>
EVALUATION

- Channel Capacity – 0.5 bit per transition.
- Bit-Error-Rate (BER) – fraction of errors in the bits decoding.
  - Expansion ECC – Less than 0.1 bit per transition.
  - Reed-Muller ECC – Better results! ~0.13 bit per transition.
- We need better error-correction-codes for this channel [Future Work].
Choosing devices based on **specification** and **price** enables **provable** covert attacks.

As far as we know – this is the first **provable** covert channel in CPS.

Requires to improve defenses:
- Adding randomness to the channel (e.g. in the controller logic)
- Purchasing devices from different vendors.
- Monitoring power consumption of devices.

In future works:
- Complimentary channel from the **sensor to the actuator** (“Chatty-Sensor”).
- Extending the attack to additional control logics and physical processes.
QUESTIONS?