THE CHATTY SENSOR: 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

Annual Computer Security Applications Conference (ACSAC), 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…

- Operating in *Feedback Control Loops*
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: INC / 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.

- **Threshold-controller**
  - Actuator with two possible commands to increase / decrease the physical value: *INC / 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.

Threshold-controller

- Actuator with two possible commands to increase / decrease the physical value: INC / 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. Threshold-controller

- Actuator with two possible commands to increase / decrease the physical value: INC / 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.

- Widely used in: phase controller, current limiter, pH 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>Specification</td>
<td>High Quality</td>
<td>Sufficient Quality</td>
</tr>
<tr>
<td>Price</td>
<td>Expensive</td>
<td>Cheap</td>
</tr>
</tbody>
</table>
INTRODUCTION: DEVICE SELECTION

- Devices are chosen based on **sufficient specification** and **lowest cost**.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Device A</th>
<th>Device B</th>
<th>Malicious</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Quality</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Sufficient Quality</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Price</th>
<th>Device A</th>
<th>Device B</th>
<th>Malicious</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expensive</td>
<td>🍀</td>
<td>🍀</td>
<td>🍀</td>
</tr>
<tr>
<td>Cheap</td>
<td>🍀</td>
<td>🍀</td>
<td>🍀</td>
</tr>
</tbody>
</table>

- **Supply Chain Attack:**
  - Attacker offers a cheaper device, with sufficient specification.
  - OR: Attacker replaces benign devices, with malicious one.

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

- Successful, stealthy attack requires communication
  - e.g. from corrupt sensor to corrupt actuator:

```
INC

Water Level Low

Neglecting the command

Water Level is too Low

Alert!
```
ATTACKER CHALLENGE - 1

- Successful, stealthy attack requires communication
  - e.g. from corrupt sensor to corrupt actuator:

Without sensor-actuator co-operation, the attack easily detected
ATTACKER CHALLENGE - 1

- How to communicate **between** malicious devices?
  - Sensor to Actuator (S2A) – This work.
  - Actuator to Sensor (A2S) – Prev. work.
ATTACKER CHALLENGES

- How to communicate **between** malicious devices?
  - Sensor to Actuator (S2A) – This work.
  - Actuator to Sensor (A2S) – Prev. work.
- How to avoid detection?

[Diagram showing communication flow between devices and responses to IN/C, ACK, and neglecting commands.]
ATTACKER CHALLENGES

- How to communicate **between** malicious devices?
  - Sensor to Actuator (S2A) – This work.
  - Actuator to Sensor (A2S) – Prev. work.
- How to avoid detection?
For any time-step $k$, the sensor reports $z_k$.

The process value continuously iterates and pass the thresholds: $T_{\text{high}}, T_{\text{low}}$

Whenever $z_k$ passes a threshold, the controller switches the command $u_k \in \{\text{INC}, \text{DEC}\}$ to the actuator.

We denote the $i^{\text{th}}$ transition of the actuator’s output by $i$. 

CHATTY-SENSOR COMMUNICATION METHOD

Sensor encodes covert bits of information, on the **parity of the transition time-steps**:

- Transition at even / odd times will signal bit 0/1.

Assumption: Sensor and Actuator have a parity-synchronized clocks.

\[ k = 7, 12, 16, 21, 26, 31 \]

\[ M_t = 1, 0, 0, 1, 0, 1 \]
CHATTY-SENSOR COMMUNICATION METHOD

- Chatty-sensor influences the transition time-step.
  - Decreasing / increasing the reported value.
- For example:
  - Transition about to happen at $k=9$ -> but should be at $k=10$. 

$T_{\text{high}}$

$k = 9$

$M_i = 1$
Chatty-sensor influence the transition time-step.

- Decreasing / increasing the reported value.

For example:

- Transition about to happen at $k=9$ -> but should be at $k=10$.
- Chatty-sensor reduces the reported value at $k=9$ -> Transition now is at $k=10$.

$k = 10 \quad M_i = 0$
ATTACKER CHALLENGES

- How to communicate *between* malicious devices?

Transition Parity
ATTACKER CHALLENGES

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

Transition Parity

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

- A lot of works on anomalies detections in CPS.

- Communication Network Anomalies:
  - For example (one of many):

- Physical Anomalies – malicious sensor reporting / malfunctioning actuator
  - For example (one of many):
ATTACKER CHALLENGES

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

Transition Parity

Creates Anomaly in the CPS behavior…
ATTACKER CHALLENGES

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

Covert Channel

Transition Parity

Creates Anomaly in the CPS behavior...
COVERT CHANNELS

- “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)

\[ \Pr(D(\text{\red{\large \bullet}}) = \text{Mal.}) \approx \Pr(D(\text{\blue\square}) = \text{Mal.}) \]
**PROVABLE COVERT CHANNELS**

**IT Networks:** Provable channels were presented in the past:

**CPS Provable Covert Channel:**


Water level after the same time, for different actuator’s response times, \( u_k = u^{|NC} \)

Uses a classifier, based on 8 measurable features of the process.
How to (provably) avoid detection?

The provably-covert channel is based on two basic observations about sensors:

- The reported measurements has a random noise, derived from some (known) distribution.
- There are different benign types of sensors in the market:
  - Accurate (narrow noise distribution)
  - Noisy (wide noise distribution).

- Adding Noise at all time-steps
- Make sure to add positive / negative noise at the transition time
The Chatty-sensor uses an internal accurate sensor to measure the process.

At time-steps with transition: the chatty-sensor chooses whether to add positive or negative noise to the internal sensor.

All the other time-steps: The Chatty-sensor randomly chooses positive or negative noise to add.

- Internal benign sensor
- Chatty-sensor
**Problem [Encoding]:** Channel is noisy...

**Solution:** Error Correction Code. Sending encoded message $m$, $m=\text{ECC}(M)$.

\[ \Pr(D(\text{Mal.}) = \text{Mal.}) \approx \Pr(D(\text{Mal.}) = \text{Mal.}) \]
Problem [Encoding]: Channel is noisy...

Solution: Error Correction Code. Sending encoded message $m$, $m=ECC(M)$.

Problem [Indistinguishability]: If the encoded message is $m=00000000...$, the noise at the transitions will always be derived from $P_{\text{down}}$.

Solution: Sending pseudo-random bits, $b_i$, derived from $m_i$.

$\kappa \leftarrow \{0,1\}^l$ is a key, deployed at the sensor and actuator.

Assumption: Sensor and actuator have a synchronized $i$.

$$Pr(D(\text{Mal.}) = \text{Mal.}) \approx Pr(D(\varnothing) = \text{Mal.})$$
ATTACKER CHALLENGES

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

Pseudo-random Transition Parity

Evaluation
EVALUATION

- How good is the receiver in intercepting the chatty-sensor bits?

- **Theoretical**: Channel Capacity.

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

- **Channel Capacity** – highest information rate that can be achieved.
- Evaluated in a pH control process Simulink simulation.
- Based on real-world pH sensors noise.
  - $\sigma_{CS}, \sigma_{HQ}$ - chatty-sensor / internal high-quality sensor noise standard deviation.
- **Results**: About 0.12 bit of information on every transition.
  - 1 transition every 5 seconds = 1.44 bits per minute.
**EVALUATION**

- **Channel Capacity** – 0.12 bit per transition.
- **Bit-Error-Rate (BER)** – fraction of errors in the bits decoding.
  - Using repetition as error-correction-code: \(~10\% \text{ decoding errors}\), with repetition of 13.
- 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 from sensors to actuator.

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

In future works:
- Improving the BER – maybe by non-provable method.
QUESTIONS?