Cross-Layer Detection of Sensor-based Deception Attacks on Cyber-Physical Systems

Nilanjan Banerjee.

Associate Professor, University of Maryland, Baltimore County

(http://www.csee.umbc.edu/~nilanb)
Cyber-physical systems are powerful systems with applications to several diverse domains.

Figure courtesy of Christoph Roser at allaboutlean.com
Alas! with great power comes great responsibility and threat.
Traditional CPS defense has focused on only one layer or level.

Computer Science

Control Engineering

Figure courtesy of Dino.korah (Wikipedia)

Figure courtesy of Prof. Alvaro A. Cardenas [1]
Cyber defenses for CPS need to work at multiple levels

1. Our method is holistic and combines data from sensors at multiple layers.

2. Combining sensor information from multiple layers allows the CPS system to operate in the event of attacks gracefully.
The key idea is to meaningfully combine information from multiple layers

1. **Observers** predict the value that should be emitted by a rogue sensor.

2. This value emitted is weighted by a **trust value** that is calculated by using **physical layer side channel analysis**.

3. A **weighted consensus** of the emitted value is the estimation from the rogue sensor fed into the actuator.
The components for the macro- and micro- detectors work together
Outline of the rest of the presentation

1. Micro-level detector

2. Macro-level consensus algorithm

3. Prototype implementation and evaluation
The micro-level detector is a side-channel monitor that uses power transients to determine if code on the sensors has been modified.

<table>
<thead>
<tr>
<th>#</th>
<th>Instruction Sequence</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>mov #33, r15</td>
<td>;2 cycles</td>
</tr>
<tr>
<td>-</td>
<td>mov.b #1, 0(r15)</td>
<td>;4 cycles</td>
</tr>
<tr>
<td>2</td>
<td>mov #304, r15</td>
<td>;2 cycles</td>
</tr>
<tr>
<td>4</td>
<td>mov.b -6(r4), r14</td>
<td>;3 cycles</td>
</tr>
<tr>
<td>7</td>
<td>mov r14, 0(r15)</td>
<td>;4 cycles</td>
</tr>
<tr>
<td>11</td>
<td>mov #312, r15</td>
<td>;2 cycles</td>
</tr>
<tr>
<td>13</td>
<td>mov.b -5(r4), r14</td>
<td>;3 cycles</td>
</tr>
<tr>
<td>16</td>
<td>mov r14, 0(r15)</td>
<td>;4 cycles</td>
</tr>
<tr>
<td>20</td>
<td>mov #314, r15</td>
<td>;2 cycles</td>
</tr>
<tr>
<td>22</td>
<td>mov @r15, -4(r4)</td>
<td>;5 cycles</td>
</tr>
<tr>
<td>27</td>
<td>inc.b -6(r4)</td>
<td>;4 cycles</td>
</tr>
<tr>
<td>31</td>
<td>inc.b -5(r4)</td>
<td>;4 cycles</td>
</tr>
<tr>
<td>35</td>
<td>mov #33, r15</td>
<td>;2 cycles</td>
</tr>
<tr>
<td>37</td>
<td>mov.b #0, 0(r15)</td>
<td>;4 cycles</td>
</tr>
<tr>
<td>-</td>
<td>jmp $-56</td>
<td>;2 cycles</td>
</tr>
</tbody>
</table>

1. The power transient for instruction groups can be unique.
Why does using power transients work for instruction identification?

1 The power transient for an instruction group is unique since the hardware utilization for the instruction is unique.
We use a learning approach to reverse engineer instruction sequences.

**Training**
- Capture power profiles
- Apply PCA
- Bin and average based on H/W utilization

**Testing**
- Capture/Window power profiles
- Estimate instruction boundaries
- Classify instructions
Our trained templates for each instruction class is based on the PCA components. The accuracy of classification using PCA and 1 Nearest Neighbor classifier is close to 100% with only the top 10 PCA components. This leads to considerable data reduction over storing the raw power transient signatures/templates.
In the testing phase we use a dynamic programming solution to estimate the instruction sequence.
In the testing phase we use a dynamic programming solution to estimate the instruction sequence.
In the testing phase we use a dynamic programming solution to estimate the instruction sequence.

Apply **dynamic programming** to estimate the sequence of instructions.
Example of instruction classification accuracy.

<table>
<thead>
<tr>
<th>Instruction Sequence</th>
<th>Predicted Classification Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Power Pin 1</td>
</tr>
<tr>
<td>pop_mem_reg</td>
<td>88</td>
</tr>
<tr>
<td>add_mem_mem_nosub</td>
<td>85</td>
</tr>
<tr>
<td>inc_reg_const_ind_nosub</td>
<td>100</td>
</tr>
<tr>
<td>mov_mem_ind_nosub</td>
<td>98</td>
</tr>
<tr>
<td>add_reg_reg</td>
<td>99</td>
</tr>
<tr>
<td>sub_mem_mem_sub</td>
<td>92</td>
</tr>
<tr>
<td>dec_const_reg</td>
<td>99</td>
</tr>
<tr>
<td>mov_ind_reg_nosub</td>
<td>100</td>
</tr>
<tr>
<td>subc_imm_reg_sub</td>
<td>95</td>
</tr>
<tr>
<td>bit_mem_mem_nosub</td>
<td>96</td>
</tr>
<tr>
<td>cmp_mem_mem_sub</td>
<td>95</td>
</tr>
<tr>
<td>xor_reg_const_ind_nosub</td>
<td>56</td>
</tr>
<tr>
<td>inc_const_reg</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>92</td>
</tr>
</tbody>
</table>

1. The accuracy of determining the sequence of instructions is close to 94%.
2. Certain power pins show better accuracy than others.
3. Majority vote does not always lead to better results.
How does this tie to the cross-layer estimation of anomalies?

1. The power monitors determine the degree to which the firmware running on the sensor is different from the one that is supposed to run.

2. Based on the above, the values $p_m$ (trustworthiness is estimated).

3. The observations for sensor $j$ at sensor $i$ are values that sensor $i$ predicts should be the value sensor $j$.

4. System runs a trust-based consensus to estimate the value at sensor $j$. 
A example to illustrate how this works

System can operate in the presence of attacks

Power monitors

Sensors

Observations

Actuator

Trust based consensus

Sensor output

Rogue sensor

Trust-based Consensus

Closed Loop System

\( \rho_m \) \( \rho_j \) \( \rho_i \)

\( O_i \) \( O_j \) \( O_m \)

\( i\hat{x}_i \) \( i\hat{x}_j \) \( i\hat{x}_m \) \( i\hat{x}_c \)
Our experimental setup comprise of temperature sensor testbed

![Temperature Control Testbed](image)

![Micro-level Analysis Setup](image)

### Scenario Timeline

<table>
<thead>
<tr>
<th>Launch Time (s)</th>
<th>Sequence #1</th>
<th>Sequence #2</th>
<th>Sequence #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>$S_5$</td>
<td>$S_6$</td>
<td>$S_6$</td>
</tr>
<tr>
<td>300</td>
<td>$S_1$</td>
<td>$S_3$</td>
<td>$S_4$</td>
</tr>
<tr>
<td>480</td>
<td>$S_5$</td>
<td>$S_5$</td>
<td>$S_2$</td>
</tr>
<tr>
<td>660</td>
<td>$S_2$</td>
<td>$S_1$</td>
<td>$S_3$</td>
</tr>
</tbody>
</table>

1. Emulate attack scenarios
If only macro-level inference is used, system is unstable after some time.

Unstable system, effect of attack if no action taken. Here false temperature is used to close the loop and quickly exits the safe region (gray ±6° F).

If the consensus value of all six sensors is used without any micro data, once the third sensor is attacked then the temperature control becomes unstable and exits the safe region.
The cross-layer approach can tolerate a large number of compromised sensors.

Here the micro-level measurements are used to weight the consensus calculations, even if four sensors are compromised, the temperature stays within the safe region.

Shows three different experiments all using the weighted trust-based consensus, Here different combinations of sensors were attacked at the same intervals. They all show similar acceptable behavior.
Future work

1. Instruction classification is a still an open problem. Being able to classify individual instructions is complicated.

2. Finding differences in code instruction sequences while being cognizant of semantic information in code is challenging.

3. Scalability of the system with hundreds of sensors need to be evaluated.
Cyber-Physical Systems (CPS) Security

A COLLABORATION BETWEEN UMBC AND UNITED STATES NAVAL ACADEMY (USNA)

UMBC Collaborators –
  Prof. Ryan Robucci, Prof. Chintan Patel,
  Prof. Nilanjan Banerjee, Prof. Anupam Joshi
Ph.D. Student Collaborators –
  Deepak Krishnankutty, Brien Croteau, Zheng Li

USNA Collaborators –
  Prof. Kiriakos Kirikadis, Prof. Tracie Severson,
  Prof. Erick Rodriguez-Seda

This work has been supported in part by the U.S. Office of Naval Research under Awards N00014-15-1-2179 and N0001417WX01442
Publications


Cross-Layer Detection of Sensor-based Deception Attacks on Cyber-Physical Systems

Nilanjan Banerjee

Associate Professor, University of Maryland, Baltimore County

(http://www.csee.umbc.edu/~nilanb)