“On Survivability of Mobile Cyber Physical Systems with Intrusion Detection”

Alex Campbell, 2016-11-17
Presentation Contents

1. Introduction and Concepts
2. Problem Statement and Challenges
3. Literature Overview
4. Proposed Solution
5. Designing the System Model
6. Running the Simulation
7. Conclusion / Future Work
Introduction and Concepts

**General Concept:**

- Advancing technology leads to increased presence of **Cyber Physical Systems (CPS)**
- **Survivability** becomes more important.
- **Mobile CPSs** complicate the issue of **survivability**
Introduction and Concepts

Cyber Physical System

- “Systems that are built from, and depend upon, the seamless integration of computational algorithms and physical components” [1].

- Defining Characteristics [3]:
  - Cyber Capability in every component
  - Automated
  - Capable of Large-Scale Networking
  - Capable of optimization through dynamic reconfiguration

Introduction and Concepts

CPS Examples

- Process Control Systems
- Medical monitors
- Autonomous (self-driving) Vehicles
Introduction and Concepts

Mobile Cyber Physical System

- Subcategory of a **CPS**
- Inherently mobile
- Examples [4]:
  - Smartphone network
  - Environmental monitoring systems
- Applications [5]:
  - Traffic Measuring System
  - IoT

Problem Statement and Challenges

Problem:

“[Maximize the survivability of] a mobile cyber physical system (MCPS) comprising sensor-carried human actors, vehicles, or robots assembled together for executing a specific mission in battlefield or emergency response situations.”

- Maximize uptime of MCPS
- Mission critical scenarios
- Protect against malicious attacks, unauthorized intrusions
Problem Statement and Challenges

Challenges:

- Distributed architecture
- Large Scale
- Rough / Dangerous environmental Conditions
- Resource Constraints

Main point: scenarios include possibilities of:

- Compromised / captured nodes
- Inability to replenish nodes
Literature Overview

**Literature:** For survivability, design MCP systems that promote:

- Intrusion Prevention
- Intrusion Detection
  - Application-specific Intrusion detection
  - Anomaly-based detection
- Intrusion Tolerance
  - Static / Structural
    - Redundancy: component, path, data
    - Threshold Cryptography (cooperative decryption).
    - Decentralization
  - Dynamic / Responsive:
    - Self-Organization
    - Dynamic Routing
    - Forward / Backward Recovery
Survivability

Energy Consumption  Intrusion Protection

Both Energy Depletion and Security Failures constitute failure of an MCPS!
Proposed Solution

**Solution:** Perform a mathematical-model-based analysis to maximize **Survivability**

- Model an MCPS with **Dynamic Voting-based Intrusion Detection**
- Optimally balance intrusion detection energy conservation
Designing the System Model

**Reference System:** Distributed network of 128 nodes, where each node contains

- 600 MHz Analog Devices Blackfin DSP Processor
- 8MB flash memory
- 64MB SDRAM
- GPS Receiver
- 7.5 V battery
- Sensors (inertial, barometric, physiological, radiological, environmental).

**Purpose:** Detect nearby phenomena, transmit information to neighbors to perform localization and remote sensing (collect data without making physical contact with the object [Wikipedia].
Designing the System Model

**Attack Model:** Two Types:

- **Node Capture**
  - Defeats Authentication
  - Creates Insider Threats
- **Bad Data Injection**
  - Defeats integrity of data
  - Defended against by insiders

**Assumption:** When the system contains $\frac{1}{3}$ compromised nodes, the system has failed (Byzantine Fault Model). Once a consensus cannot be reached (due to fear of malicious nodes), the system has failed.
Designing the System Model

**Intrusion Detection Technique:** Dynamic voting-based intrusion detection

- Detection informed by location/distance data anomalies between neighbors
- A “coordinator” node is chosen amongst neighbors at random to prevent specific targeting by attackers
- Coordinator selects $m$ random nodes to participate in labeling nodes as good/bad
Designing the System Model

**Intrusion Detection Technique:** Dynamic voting-based intrusion detection

**Main Point:** Predict the number of good/bad nodes as a result of compromising events happening in the system, coupled with voting-based intrusion detection.
Designing the System Model

Modeling the system in regards to intrusions and energy consumption:

\[
\begin{align*}
N_g & \rightarrow \text{TCP} \\
N_g & \rightarrow N_b \\
N_b & \rightarrow \text{TIDS} \\
N_b & \rightarrow N_e \\
N_e & \rightarrow \text{TFP} \\
N_e & \rightarrow \text{ENERGY}
\end{align*}
\]

- **Ng**: # Good Nodes
- **Nb**: # Bad Nodes
- **Ne**: # Nodes Evicted
- **TIDS**: Intrusion Detection Interval
- **energy**: Binary, 1=full energy, 0=exhaustion
- **λ**: Compromise Rate
- **Pfn**: P(false negative)
- **Pfp**: P(false positive)
- **TIDS**: Dynamic Voting Invocation Interval
- **TCP**: Good node get compromised. Rate: \(\lambda \times N_g\)
- **TIDS**: Evict Bad Node: \(\frac{N_b \times (1 - P_{fn})}{T_{IDS}}\)
- **TFP**: Evict Good Node: \(\frac{N_e \times P_{fp}}{T_{IDS}}\)
- **ENERGY**: Energy is Exhausted: \(\frac{1}{N \times T_{IDS}}\)
Designing the System Model

Equivalent Semi-Markov Model:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Ng</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Nb</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Ne</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>energy</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Designing the System Model

Modeling the system in regards to intrusions and energy consumption:

- **Important Concepts:**
  - Tokens = nodes in MCPS
  - Initialize 128 Good nodes
  - Pfn, Pfp, and λ are used as input parameters to the underlying markov chain.
  - Use to calculate expected values for each state at time $t$.
  - Use these expected values to solve for Pfn and Pfp at time $t$.
  - Adjust Transitions TIDS and TFP to model changes to Pfn and Pfp.
Designing the System Model

Solving for $P_{fn}$ and $P_{fp}$:

$$P_{fn} = \sum_{i=0}^{m-N_{maj}} \left[ \frac{N_b \binom{N_{maj} + i}{m}}{N_maj + i \binom{N_g + N_b}{m}} \right]$$

$$+ \sum_{j=0}^{m-N_{maj}} \sum_{k=N_{maj}-j}^{m-j} \frac{\binom{N_g}{k} \left( p_{fn} \right)^k \left( \frac{N_g - k}{m - j - k} \right) (1 - p_{fn})^{m-j-k}}{N_g + N_b \binom{m}{N_g + N_b}}$$

$$P_{fp} = \sum_{i=0}^{m-N_{maj}} \left[ \frac{N_b \binom{N_{maj} + i}{m}}{N_maj + i \binom{N_g + N_b}{m}} \right]$$

$$+ \sum_{j=0}^{m-N_{maj}} \sum_{k=N_{maj}-j}^{m-j} \frac{\binom{N_g}{k} \left( p_{fp} \right)^k \left( \frac{N_g - k}{m - j - k} \right) (1 - p_{fp})^{m-j-k}}{N_g + N_b \binom{m}{N_g + N_b}}$$

Probability of a false negative due to selecting a majority of bad nodes

Probability of a false negative due to:
1. Selecting a majority of good nodes that cast incorrect votes
2. Including some bad nodes

Probability of a false positive due to selecting a majority of bad nodes

Probability of a false positive due to:
1. Selecting a majority of good nodes that cast incorrect votes
2. Including some bad nodes
Designing The System Model

Calculate MTTF via Reward Assignments:

- Recall that we want to optimize the MCPS Survivability
- Survivability is equivalent to the system’s expected lifetime, or MTTF
- Let Ri, reward assignment at state i, be:
  - Ri = 1 if the system is alive in state i
  - Ri = 0 if the system is dead in state i
  - System is dead when:
    - Place “Energy” does not have a token
    - The number of tokens when Nb > \( \frac{1}{3}(Nb + Ng) \)
      - Number of bad nodes comprises at least 1.3 of all nodes in system
- \( P_{fp}, P_{fn}, \) and \( T_{dis} \) all affect transition rates, and therefore MTTF
Designing the System Model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>Network size</td>
<td>128</td>
</tr>
<tr>
<td>$\bar{n}$</td>
<td>Number of neighbors within radio range</td>
<td>32</td>
</tr>
<tr>
<td>$p_{fn}$</td>
<td>Per-host false negative probability</td>
<td>[1–5]%</td>
</tr>
<tr>
<td>$p_{fp}$</td>
<td>Per-host false positive probability</td>
<td>[1–5]%</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Per-node capture rate</td>
<td>[1–24]/day</td>
</tr>
<tr>
<td>$T_{IDS}$</td>
<td>Intrusion detection interval</td>
<td>[0–700] s</td>
</tr>
<tr>
<td>$m$</td>
<td>Number of intrusion detectors per node</td>
<td>[3,11]</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Number of ranging operations</td>
<td>5</td>
</tr>
<tr>
<td>$E_t$</td>
<td>Energy for transmission per node</td>
<td>0.000125 J</td>
</tr>
<tr>
<td>$E_r$</td>
<td>Energy for reception per node</td>
<td>0.000005 J</td>
</tr>
<tr>
<td>$E_a$</td>
<td>Energy for analyzing data per node</td>
<td>0.00174 J</td>
</tr>
<tr>
<td>$E_s$</td>
<td>Energy for sensing per node</td>
<td>0.0005 J</td>
</tr>
<tr>
<td>$E_0$</td>
<td>Initial system energy</td>
<td>16,128 kJ</td>
</tr>
<tr>
<td>$P_{fn}$</td>
<td>System false negative probability</td>
<td>Eq. 1</td>
</tr>
<tr>
<td>$P_{fp}$</td>
<td>System false positive probability</td>
<td>Eq. 2</td>
</tr>
<tr>
<td>MTTF</td>
<td>Mean time to failure</td>
<td>Eq. 3</td>
</tr>
<tr>
<td>$N$</td>
<td>Maximum cycles before energy exhaustion</td>
<td>Eq. 4</td>
</tr>
<tr>
<td>$E_{IDS}$</td>
<td>Energy consumed per $T_{IDS}$</td>
<td>Eq. 5</td>
</tr>
</tbody>
</table>

\[ E_{\text{detection}} = m \times (E_t + \bar{n} \cdot E_r) + m \times (E_t + (m - 1) \cdot (E_r + E_a)). \] (8)
Running the simulation

**Theoretical Results:**

- $m = \#\text{nodes selected for voting}$
- Optimal Intrusion detection interval ($T_{IDS}$) is roughly 200 seconds
- Optimal $T_{IDS}$ value decreases as $m$ decreases: weaker intrusion detection means more invocations
- Optimally, $m = 5$. Best balance of energy exhaustion and security failure.
Running the simulation

- Use a simulation modeling library, SMPL, to:
  - Track node state (goodness, membership)
  - Schedule events
  - Monitor system failure based on events:
    - Security failure
    - Exhausted Energy
    - All nodes have been evicted

- Parameterize values:
  - $\lambda$ from 1/day to 1/10 minutes
  - $m$ from [3,11]
  - $T_{IDS}$ from 10s to 1280s

- Apply BMA for 95% confidence level and 10% accuracy:
  - 100 MTTF Observations
Running the simulation: SMPL Results

Fig. 7 Simulation and theoretical MTTF versus $T_{IDS}$ and $m$
Running the simulation

Remarks:

- Theoretical and Simulation plot shapes are very similar
- For both, MTF peaks near TIDS = 160s between 9000 and 11,000s
- $m = 5$ is the optimal value for $m$ in both cases
- The Mean Percentage Error (MPE) between the two is between 4.60 and 7.64%

Main Point: Survivability analysis methodology is validated due to similarities between results.
Conclusions and Future Work

- This paper demonstrated the feasibility of the authors’ survivability model for Mobile Cyber Physical Systems with voting-based intrusion detection.
  - Given known values for false alarm probabilities and node compromise rates, the model can determine the best intrusion detection interval and the best number of detectors to maximize MTTF.
- Future work may include discussions concerning design principles for intrusion detection protocols in both homogenous AND heterogenous MCPSs.