Identifying and Analyzing Implicit Interactions in Critical Infrastructure Systems

Jason Jaskolka
Collaborator: John Villasenor (UCLA)

Department of Systems and Computer Engineering
Carleton University, Ottawa, ON, Canada
jaskolka@sce.carleton.ca

January 25, 2018
Acknowledgement & Disclaimer

Acknowledgement

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number, 2015-ST-061-CIRC01.

Disclaimer

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.
Critical Infrastructure

Analysis Framework

Concluding Remarks

Cybersecurity Challenges in Critical Infrastructure
Implicit Component Interactions
Research Problem
Cybersecurity Challenges in Critical Infrastructure

- Ubiquitous and pervasive
- Large, complex, and rapidly growing
- Mix of legacy systems and new technologies
- Numerous components or agents and even more interactions, some of which may be:
  - Unfamiliar, unplanned, or unexpected
  - Not visible or not immediately comprehensible

- Software/Hardware from third-party suppliers
- Cyber-attackers are far more sophisticated and have access to far more powerful tools than in the past

Implicit Interactions
Definition (Implicit Interaction)
An interaction among system components that may be unfamiliar, unplanned, or unexpected, and either not visible or not immediately comprehensible by the system designers.

- Can indicate unforeseen design flaws allowing for these interactions
- Constitute linkages of which designers are generally unaware
  \[\Rightarrow\text{ security vulnerability}\]
- Can be exploited to mount cyber-attacks at a later time
  - Potential for unexpected system behaviors
  - Example: Gain unauthorized access to information
Assuring safety, security, and reliability of critical infrastructure systems is becoming a top priority.

Shortcomings in development of formal methods and tools for determining whether such systems are protected from cyber-threats [Bennett 2015]

Ability to detect undesirable interactions among system components is needed [Jackson and Ferris 2012]

Research Challenge

Develop a rigorous (formal methods-based) approach to better understand, identify, analyze, and mitigate implicit interactions in critical infrastructure systems.
Why Formal Methods?

According to the DHS Cybersecurity Research Roadmap [DHS 2009]

“Formal verification and other analytic tools that can scale will be critical to building systems with significantly higher assurance than today’s systems.”

“In particular, theories are needed to support analytic tools that can facilitate the prediction of trustworthiness, inclusion modeling, simulation, and formal methods.”

“The potential utility of formal methods has increased significantly in the past four decades and needs to be considered whenever it can be demonstrably effective.”
Proposed Approach for Solving the Problem

Research Goal

Develop an analysis framework to enhance our understanding of how and why implicit interactions can exist and to identify system deficiencies in critical components to enable a better assessment of risks being taken by using such components to build critical infrastructure systems.

1. Model critical infrastructure systems using a mathematical framework
2. Formulate and identify the existence of implicit interactions
3. Analyze existing implicit interactions
4. Mitigate the existence of and/or minimize the threat posed by implicit interactions
Illustrative Example: Manufacturing Cell
Illustrative Example: Manufacturing Cell
Illustrative Example: Manufacturing Cell

Control/Coordination Agent

Storage Agent

Handling Agent

Processing Agent
Illustrative Example: Manufacturing Cell

1. Start
2. Load
3. Loaded
4. Prepare
5. Status
6. Unload
7. Unloaded
8. Setup
9. Status
10. Ready
11. Process
12. Processed
13. End
Approach for Solving the Problem

1. **Model critical infrastructure systems using a mathematical framework**

2. Formulate and identify the existence of implicit interactions

3. Analyze existing implicit interactions

4. Mitigate the existence of and/or minimize the threat posed by implicit interactions
An Algebraic Modeling Framework
Communicating Concurrent Kleene Algebra (C²KA)

- Formalism for modeling distributed multi-agent systems
- Captures communication and concurrency of agents at an abstract algebraic level
- Expresses influence of stimuli on agent behavior in open systems as well as communication through shared environments

Three levels of specification:
1. Stimulus-Response Specification
2. Abstract Behavior Specification
3. Concrete Behavior Specification
Communicating Concurrent Kleene Algebra (C²KA)

**Definition (C²KA)**

A **Communicating Concurrent Kleene Algebra (C²KA)** is a system \((S, K)\), where

- \(S = (S, \oplus, \odot, \circ, \n)\) is a *stimulus structure*
- \(K = (K, +, \ast, ;, \ast, \odot, 0, 1)\) is a CKA

- \((S, +)\) is a *unitary and zero-preserving left \(S\)-semimodule* with *next behavior mapping* \(\circ : S \times K \rightarrow K\)
- \((S_K, \oplus)\) is a *unitary and zero-preserving right \(K\)-semimodule* with *next stimulus mapping* \(\lambda : S \times K \rightarrow S\)

and where the following axioms are satisfied for all \(a, b, c \in K\) and \(s, t \in S\):

1. \(s \circ (a ; b) = (s \circ a) ; (\lambda(s, a) \circ b)\)
2. \(a \leq_K c \lor b = 1 \lor (s \circ a) ; (\lambda(s, c) \circ b) = 0\)
3. \(\lambda(s \circ t, a) = \lambda(s, (t \circ a)) \odot \lambda(t, a)\)
4. \(s = \circ \lor s \circ 1 = 1\)
5. \(a = 0 \lor \lambda(n, a) = n\)
Agent Specifications
Illustrative Example: Manufacturing Cell

Table: Stimulus-response specification of the Control Agent C

<table>
<thead>
<tr>
<th></th>
<th>start</th>
<th>load</th>
<th>loaded</th>
<th>prepare</th>
<th>done</th>
<th>unload</th>
<th>unloaded</th>
<th>setup</th>
<th>ready</th>
<th>process</th>
<th>processed</th>
<th>end</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLE</td>
<td>IDLE</td>
<td>IDLE</td>
<td>PREP</td>
<td>IDLE</td>
<td>IDLE</td>
<td>IDLE</td>
<td>IDLE</td>
<td>IDLE</td>
<td>IDLE</td>
<td>IDLE</td>
<td>IDLE</td>
<td>IDLE</td>
</tr>
<tr>
<td>PREP</td>
<td>PREP</td>
<td>PREP</td>
<td>PREP</td>
<td>PREP</td>
<td>PREP</td>
<td>PREP</td>
<td>INIT</td>
<td>INIT</td>
<td>PREP</td>
<td>PREP</td>
<td>PREP</td>
<td>PREP</td>
</tr>
<tr>
<td>INIT</td>
<td>INIT</td>
<td>INIT</td>
<td>INIT</td>
<td>INIT</td>
<td>INIT</td>
<td>INIT</td>
<td>INIT</td>
<td>INIT</td>
<td>INIT</td>
<td>INIT</td>
<td>INIT</td>
<td>INIT</td>
</tr>
<tr>
<td>PROC</td>
<td>PROC</td>
<td>PROC</td>
<td>PROC</td>
<td>PROC</td>
<td>PROC</td>
<td>PROC</td>
<td>PROC</td>
<td>PROC</td>
<td>PROC</td>
<td>PROC</td>
<td>PROC</td>
<td>PROC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>start</th>
<th>load</th>
<th>loaded</th>
<th>prepare</th>
<th>done</th>
<th>unload</th>
<th>unloaded</th>
<th>setup</th>
<th>ready</th>
<th>process</th>
<th>processed</th>
<th>end</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLE</td>
<td>load</td>
<td>n</td>
<td>prepare</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>PREP</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>setup</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>INIT</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>done</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>PROC</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>end</td>
<td>n</td>
<td>n</td>
</tr>
</tbody>
</table>

Control Agent C  \(\rightarrow\)  \(\langle\text{IDLE + PREP + INIT + PROC}\rangle\)
Storage Agent S  \(\rightarrow\)  \(\langle\text{EMPTY + FULL}\rangle\)
Handling Agent H  \(\rightarrow\)  \(\langle\text{WAIT + MOVE}\rangle\)
Processing Agent P  \(\rightarrow\)  \(\langle\text{STBY + SET + WORK}\rangle\)

Figure: Abstract behavior specification of the manufacturing cell agents
## Agent Specifications

### Illustrative Example: Manufacturing Cell

<table>
<thead>
<tr>
<th>C →</th>
<th>S →</th>
<th>P →</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C</strong> →</td>
<td><strong>S</strong> →</td>
<td><strong>P</strong> →</td>
</tr>
<tr>
<td>IDLE def = state := 0</td>
<td>EMPTY def = status := 0</td>
<td>STBY def = skip</td>
</tr>
<tr>
<td>PREP def = state := 1</td>
<td>FULL def = status := 1</td>
<td>SET def = if (material = 1 ∧ state = 2 ∧ status = 0) → ready := 1</td>
</tr>
<tr>
<td>INIT def = state := 2</td>
<td></td>
<td>fi</td>
</tr>
<tr>
<td>PROC def = state := 3</td>
<td>WORK def = if (ready = 1) → part := PROCESS()</td>
<td>¬(ready = 1) → part := null</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fi</td>
</tr>
</tbody>
</table>

### Figure: Concrete behavior specification of the manufacturing cell agents
Approach for Solving the Problem

1. Model critical infrastructure systems using a mathematical framework
   - Communicating Concurrent Kleene Algebra (C²KA)

2. **Formulate and identify the existence of implicit interactions**

3. Analyze existing implicit interactions

4. Mitigate the existence of and/or minimize the threat posed by implicit interactions
Intended System Interactions

Control Agent (C)

Storage Agent (S)

Processing Agent (P)

Handling Agent (H)

\( P_{\text{intended}} \) denotes the set of intended system interactions
Illustrative Example: Manufacturing Cell

Intended System Interactions

\[ P_{\text{intended}} = \{ \]
\[ C \rightarrow S \quad S \rightarrow C \quad H \rightarrow S \quad S \rightarrow C \quad P \rightarrow S \quad H \rightarrow S \quad P \rightarrow S \quad C, \]
\[ C \rightarrow S \quad S \rightarrow C \quad H \rightarrow S \quad S \rightarrow C \quad P \rightarrow S \quad H \rightarrow S \quad C \rightarrow S \quad P, \]
\[ C \rightarrow S \quad S \rightarrow \varepsilon \quad H \rightarrow \varepsilon \quad P \rightarrow S \quad H \rightarrow S \quad C \rightarrow S \quad P, \]
\[ C \rightarrow S \quad S \rightarrow \varepsilon \quad H \rightarrow \varepsilon \quad P \rightarrow S \quad H \rightarrow S \quad P \rightarrow S \quad C, \]
\[ \ldots \]
\[ C \rightarrow S \quad S \rightarrow C \quad H \rightarrow S \quad S \rightarrow \varepsilon \quad P \rightarrow S \quad H \rightarrow S \quad C \rightarrow S \quad P, \]
\[ C \rightarrow S \quad S \rightarrow C \quad H \rightarrow S \quad S \rightarrow \varepsilon \quad P \rightarrow S \quad H \rightarrow S \quad P \rightarrow S \quad C, \]
\[ C \rightarrow S \quad S \rightarrow C \quad H \rightarrow S \quad S \rightarrow C \quad P \rightarrow S \quad H \rightarrow S \quad C \rightarrow S \quad P, \]
\[ C \rightarrow S \quad S \rightarrow C \quad H \rightarrow S \quad S \rightarrow C \quad \varepsilon \quad P \rightarrow S \quad H \rightarrow S \quad P \rightarrow S \quad C \} \]
Potential for Communication via Stimuli \( (A \rightarrow_S^+ B) \)

A has the **potential for communication via stimuli** with B if and only if

\[
\exists (n \mid n \geq 1 : A \rightarrow_S^n B)
\]

where

\[
A \rightarrow_S^n B \iff \exists (C \mid C \in A \land C \neq A \land C \neq B : A \rightarrow_S^{(n-1)} C \land C \rightarrow_S B)
\]

\[
A \rightarrow_S B \iff \exists (s, t \mid s, t \in S_b \land t \leq_S \lambda(s, a) : t \circ b \neq b)
\]

Definition (Existence of Implicit Interactions)

An *implicit interaction* exists in a system formed by a set \( A \) of agents, if and only if for any two agents \( A, B \in A \) with \( A \neq B \):

\[
\exists (p \mid p \Rightarrow (A \leadsto^+ B) : \forall (q \mid q \in P_{\text{intended}} : \neg \text{SubPath}(p, q)))
\]

where SubPath\((p, q)\) is a predicate indicating that \( p \) is a subpath of \( q \).
1. Determine the potential communication paths that exist from the system specification

- **Example**: Consider the manufacturing cell:

\[
\begin{align*}
&\text{P} \rightarrow \text{S}: \text{True} \\
&\text{P} \rightarrow \text{S} \quad \text{H} \rightarrow \text{S} \\
&\text{P} \rightarrow \text{S} \quad \text{C} \rightarrow \text{S} \\
&\text{P} \rightarrow \text{S} \quad \text{H} \rightarrow \text{S} \\
\end{align*}
\]

\[
\begin{align*}
&\text{H} \rightarrow \text{C}: \text{True} \\
&\text{H} \rightarrow \text{S} \quad \text{C} \\
&\text{H} \rightarrow \text{S} \quad \text{S} \rightarrow \text{E} \\
&\text{H} \rightarrow \text{E} \\
\end{align*}
\]
Identifying Implicit Interactions

2. Determine if a potential communication path is an implicit interaction

- **Example**: Consider the following potential communication paths:
  \[ H \rightarrow_S S \rightarrow_S C \text{ and } P \rightarrow_S C \rightarrow_S S \]
Identifying Implicit Interactions

- Control Agent (C)
- Storage Agent (S)
- Processing Agent (P)
- Handling Agent (H)

Diagram showing interactions between agents:
- C → S → C
- H → S
- C → P → H
- P → C
- C → P
Approach for Solving the Problem

1. Model critical infrastructure systems using a mathematical framework
   - Communicating Concurrent Kleene Algebra ($C^2KA$)

2. Formulate and identify the existence of implicit interactions
   - Potential for Communication

3. Analyze existing implicit interactions

4. Mitigate the existence of and/or minimize the threat posed by implicit interactions
Analyzing Implicit Interactions

- Provide a means for determining the interactions that have the potential to most negatively impact the system.

- **Severity**: a measure of the relative non-overlap between a possible interaction with the intended interactions of a system.
  - less overlap $\implies$ higher severity $\implies$ more unexpected

- **Exploitable**: a measure of the fraction of ways that a source agent can influence the behavior of its adjacent agents to eventually influence the behavior of the sink agent.
  - higher exploitability $\implies$ more ways to influence behaviors
Sample Tool Output

Identification & Severity

--------------------
ALL PATHS: S -> C
--------------------
SEVERITY = 0.00 S -> E P -> S H -> S C
SEVERITY = 0.50 S -> E H -> S C
SEVERITY = 0.50 S -> E P -> S C
SEVERITY = 0.33 S -> E H -> E P -> S C
SEVERITY = 0.33 S -> E H -> S P -> S C
SEVERITY = 0.00 S -> S C
--------------------
IMPLICIT PATHS: S -> C
--------------------
S -> E H -> S C
S -> E P -> S C
S -> E H -> E P -> S C
S -> E H -> S P -> S C
--------------------
ALL PATHS: P -> S
--------------------
SEVERITY = 0.33 P -> S H -> S C -> S S
SEVERITY = 0.50 P -> S C -> S S
SEVERITY = 0.33 P -> S C -> S H -> S S
SEVERITY = 0.50 P -> S H -> S S
--------------------
IMPLICIT PATHS: P -> S
--------------------
P -> S H -> S C -> S S
P -> S C -> S S
P -> S C -> S H -> S S
P -> S H -> S S

Attack Scenarios & Exploitability

IMPLICIT PATH = S -> E H -> S C
ATTACK SCENARIOS = {status}
EXPLOITABILITY = 1.00

IMPLICIT PATH = S -> E P -> S C
ATTACK SCENARIOS = {material, state, status}
EXPLOITABILITY = 0.75

IMPLICIT PATH = S -> E H -> E P -> S C
ATTACK SCENARIOS = {status}
EXPLOITABILITY = 0.75

IMPLICIT PATH = S -> E H -> S P -> S C
ATTACK SCENARIOS = {status}
EXPLOITABILITY = 0.25

IMPLICIT PATH = P -> S C -> S S
ATTACK SCENARIOS = {start}
EXPLOITABILITY = 0.20

IMPLICIT PATH = H -> E P -> S C -> S S
ATTACK SCENARIOS = {}
EXPLOITABILITY = 0.00
Experimental Results

- For the manufacturing cell system:
  - 29 of the 65 total interactions are identified as *implicit interactions*

- Result of the potential for out-of-sequence messages or reads/writes from system agents
  - Due to cyber-attack or failure

- Demonstrates *hidden complexity and coupling* among agents
  - Potential for *unexpected system behaviors*

- Although the example is presented in the context of manufacturing, the analogous communication and dependencies are found in nearly all complex distributed systems
Approach for Solving the Problem

1. **Model critical infrastructure systems using a mathematical framework**
   - Communicating Concurrent Kleene Algebra ($C^2KA$)

2. **Formulate and identify the existence of implicit interactions**
   - Potential for Communication

3. **Analyze existing implicit interactions**
   - Classifying and Measuring Severity and Exploitability

4. **Mitigate the existence of and/or minimize the threat posed by implicit interactions**
A Comment on Mitigating Implicit Interactions

1. **Preemptive Approaches**
   - Eliminate potential for communication while maintaining overall system functionality
   - Introduce intermediate agents or modify agent behaviors

2. **Reactive Approaches**
   - Monitor communication and behavior to find suspicious activity
Impact of this Research

- **Enhances the understanding** of the hidden complexity and coupling in critical infrastructure systems
- **Formal foundation** upon which mitigation approaches can be developed
- **Basis for developing guidelines** for designing and implementing critical infrastructure systems that are resilient to cyber-threats
- **There is still more to be done!**
Where Do We Go From Here?

- **Refinements** to classification and measurement of severity and exploitability
  - Study impact of implicit interactions through simulation

- **Further articulate mitigation approaches**

- **Validate and evaluate** our approaches in real-world applications
  - Working with *USTRANSCOM* on analyzing a case study system
  - Obtain useful direction and feedback for transitioning our foundational research into practical outcomes from the project

- **Develop software tools** to automate the specification, identification, and analysis of critical infrastructure systems based on our mathematical foundations
Implicit interactions can pose a serious cyber-threat to critical infrastructure systems.

Aids in addressing potential security vulnerabilities at early stages of system development.

Provides vital information that can drive decisions on where and how to spend valuable resources in mitigation efforts.

Community engagement can enable contributions to emerging challenges in critical infrastructure cybersecurity.
J. Jaskolka and J. Villasenor.
An Algebraic Approach for Simulating and Analyzing Distributed Systems. 

J. Jaskolka and J. Villasenor.
Evaluating the Exploitability of Implicit Interactions in Distributed Cyber-Physical Systems. 
*ACM Transactions on Cyber-Physical Systems*, (Under Review).

J. Jaskolka and J. Villasenor.
An Approach for Identifying and Analyzing Implicit Interactions in Distributed Systems. 

J. Jaskolka and J. Villasenor.
Identifying Implicit Component Interactions in Distributed Cyber-Physical Systems. 

J. Jaskolka.
On the Modelling, Analysis, and Mitigation of Distributed Covert Channels. 
J. Jaskolka and R. Khedri.
A Formulation of the Potential for Communication Condition using C^2KA.

J. Jaskolka, R. Khedri, and Q. Zhang.
Endowing Concurrent Kleene Algebra with Communication Actions.

C. Bennett.
Feds Lack Method to Grade Critical Infrastructure Cybersecurity.
The Hill (Online), November 2015.

S. Jackson and T. L. J. Ferris.
Infrastructure Resilience: Past, Present, and Future.

A Roadmap for Cybersecurity Research.
Questions?
Thank You