Spring 2009

Information flow properties for cyber-physical systems

Rav Akella

Department:

INFORMATION FLOW PROPERTIES FOR CYBER-PHYSICAL SYSTEMS

by

RAVI CHANDRA AKELLA

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN COMPUTER SCIENCE

2009

Approved by

Dr. Bruce McMillin, Advisor
Dr. Ann Miller
Dr. Sriram Chellappan

ABSTRACT

In cyber-physical systems, which are the integrations of computational and physical processes, security properties are difficult to enforce. Fundamentally, physically observable behavior leads to violations of confidentiality. This work analyzes certain noninterference based security properties to ensure that interactions between the cyber and physical processes preserve confidentiality. A considerable barrier to this analysis is the representation of physical system interactions at the cyber-level. This thesis presents encoding of these physical system properties into a discrete event system and represents the cyber-physical system using Security Process Algebra (SPA). The model checker, Checker of Persistent Security (CoPS) shows Bisimulation based NonDeducibility on Compositions (BNDC) properties, which are a variant of noninterference properties, to check the system's security against all potential high-level interactions. This work considers a model problem of invariant pipeline flow to examine the BNDC properties and their applicability for cyber-physical systems.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Bruce McMillin, for his constant encouragement and suggestions for improving the quality of my work and especially for giving this work, a clear direction. I would also thank Dr. Ann Miller and Dr. Sriram Chellappan for their valuable suggestions and ideas. I would like to acknowledge Dr. Thomas Weigert for his idea of an event-based approach to encode the system into process algebra, which, otherwise had been difficult to implement using value passing SPA. Some of the ideas in this manuscript developed during the course of research meetings with my labmates, for whose assistance I am also grateful. I would also like to acknowledge Intelligent Systems Center, Missouri S&T, Rolla, for funding my work. Not least, I am greatly indebted to my parents, Mr. Sarma Akella and Mrs. Sobha Rani Akella for their deep concern and care for me and my academic pursuits.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

1.1. DEFINITION OF CPS

Cyber-physical systems (CPSs) are integrations of computation with physical processes. Embedded computers and networks monitor and control physical processes, usually with feedback loops; physical processes affect computations, and vice versa [14]. Applications of CPS include high-confidence medical devices and systems, traffic control and safety, advanced automotive systems, process control, energy conservation, environmental control, avionics, instrumentation, and critical infrastructure control systems (such as electric power, water resources, and communications systems). These systems are interconnected both physically and in the cyber world so that an action in one part of the system is felt in other parts of the system. Timing [23], frequency [21], and security [22] are some properties of interest. Theoretical basis for this work is also based on security models outlined by McClean [16] [17] [18] and Zakinthinos et.al [24]. The physical nature of a CPS tends to expose information flow through actions at the cyber-physical boundary. This report focuses on the confidentiality properties of CPSs, especially in terms of information flow security. Keeping the actions of a CPS confidential can preserve its integrity.

1.2. METHODOLOGY FOR APPLYING INFORMATION FLOW SECURITY IN CPS

The security requirements of a CPS depend on cyber information flow, physically observable behavior, and the interactions among the cyber and physical components of the system. Commodity flow in a cyber-physical system refers to the flow of physical entity

through the physical components. If these semantics are understood, they actually represent information. Thus, the cyber and physical components may satisfy the requirements of a security policy individually; however their composition may not. When two systems are composed, the resulting systems' behavior depends on the interactions between the two component systems. This work addresses the cyber-physical interactions resulting from the compositon of a system.

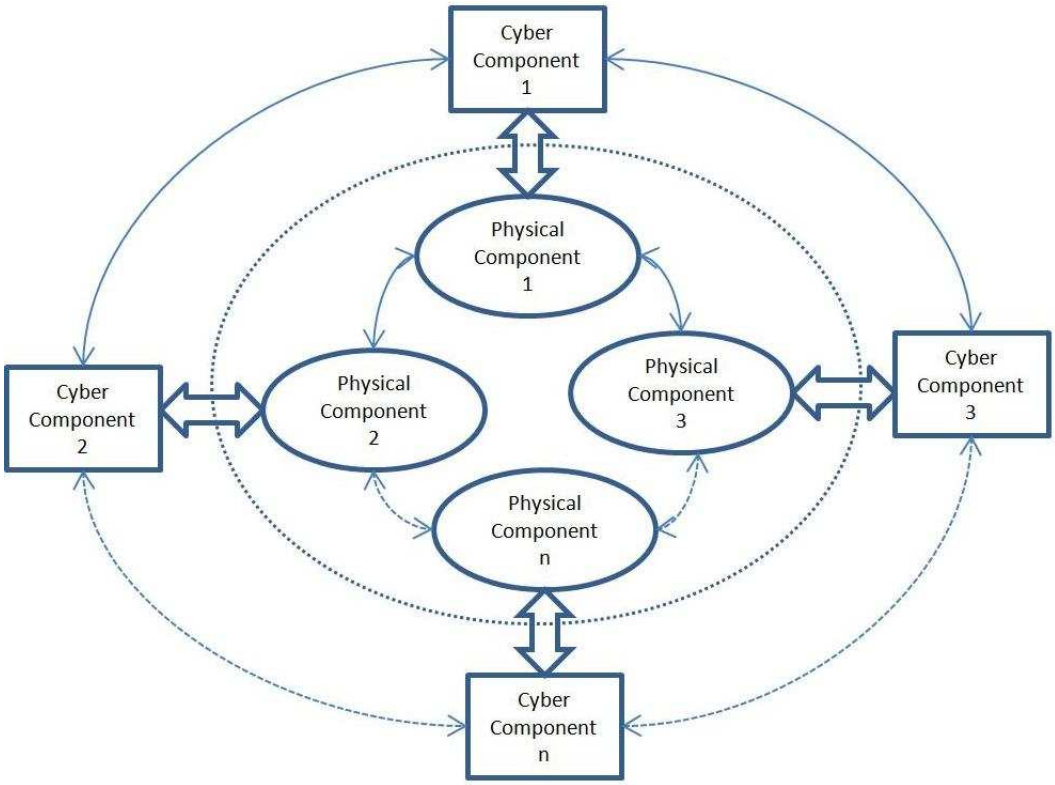

Figure 1.1. Cyber-physical Interactions

1.2.1. Establishing cyber-information flow security.

Figure 1.1 shows computational components that monitor the respective sections of a system. Setting aside their interactions with the physical components of the system, this constitution of computational components is just like a network of computer systems. The

system comprising the outermost ring in Figure 1.1 is assumed to be protected if it follows a set of predefined computer security policies such as encryption and authentication.

Goguen and Meseguer[11] and McClean[16] have comprehensively addressed information flow properties for a general class of deterministic and nondeterministic systems. These security properties are used to analyze the system to determine whether an observer might be able to deduce information about the system or interfere with the actions that take place in it. Classical models of information flow security like nondeducibility[12], noninterference[11] and noninference are concerned with preventing unauthorized information flows like downgrading of information through covert channels. Cyber-information flow security is achievable, if such security requirements are imposed on the system.

1.2.2. Establishing commodity flow security.

Cyber-physical systems are typically composed of several physical components that carry a commodity. Examples of such systems include commodity transport networks like an electric power grid or a gas pipeline system spanning across a large geographical area. Such infrastructure is usually monitored at specific geographical sites. In a large composite network of powerlines or pipes, each monitored site is treated as a system component. Because each is meant to act as a local distribution point. Semantically, commodity flow is primarily governed by the laws of physics according to the design of the physical system. The innermost ring in Figure 1.1 captures these physical interactions, which are governed by the concept of invariant flow of the physical entity. For example, the pressure in a gas pipeline changes in accordance with the laws of gas pressure. Similarly, voltage across every branch of a power grid varies in accordance with Kirchhoff's law. With knowledge of the universal behavior of a system commodity, an observer can infer information flow. Commodity flow changes at any physical component whenever a control setting is modified by an actuator operating the component. An actuator is a mechanical or electrical device for controlling a mechanism or a system. Because of this physical interdependence among components in realtime, a change made to the commodity flow at one site cannot be restricted locally; it changes the commodity flow at other sites as well.

Modeling CPS for security with respect to commodity flow is difficult in real time because events occur dynamically and sometimes nondeterministically. For the same reason, establishment of the desired commodity flows is difficult using formal tools like nondeducibility and noninference security models. In other words, it is difficult to model the system without unauthorized or undesirable commodity flows because it is hard to capture the physical interactions.

1.2.3. Establishing the security of cyber-physical interactions.

The commodity in a cyber-physical system is controlled by the physical components, which are in turn monitored by the cyber elements. Cyber-physical interactions (illustrated by the middle ring in Figure 1.1) result from coupling information with the commodity flows of a cyber-physical system. To meet the requirements of a system, cyber processes interact with the physical components by interfacing with the control systems, like actuators that operate directly on the physical equipment. An example of such an industrial control process is the supervisory control and data acquisition (SCADA). Messages sent from the cyber level or received at the physical level undergo a conversion from digital information to electrical or actuating signals. These messages could be commands issued by the cyber elements defined to set certain control settings or parameters on the physical network. Clearly, the cyber and physical frameworks are interdependent. In section 1.2.2, protection of the commodity flow in a composite physical system were discussed. When cyber processes are developed for a physical system, vulnerabilities can be expected due to this mutual interdependence. In other words, a compromise in the commodity flow could compromise information flow, and vice-versa. This work identifies, such compromises of information flow security in specific cyber-physical infrastructures by validating them with properties known to secure information flow.

Individual cyber processes control various sections of a composite physical infrastructure that behaves as a unified system with laws of physics inherent in its constitution. First, given the actuating points on the physical system, the whole physical process cannot

practically be discretized to act as distinct sites independent of each other. Also, the complexity increases because the composite cyber framework (the ring containing the cyber components in Figure 1.1) cannot capture exactly the events associated with the composite physical framework (the ring containing the physical components in Figure 1.1)of the cyber-physical system. Such precise capture is impossible because the cyber process controlling a specific site does not consider the physical dependence of commodity flow at this site on neighboring sites. Either the neighboring sites must accommodate a rearrangement of the commodity flow, or the cyber process cannot be allowed to make a change. With distributed coordination among the cyber processes to make decisions regarding allowable flow readjustments among the sites, the desired flows in a system can be established. For example, two of many monitoring sites on the critical infrastructure (like different companies operating on the pipelines running through different geographical zones) could perform a trade, so that they share the commodity between themselves accounting for flow stabilization without impacting other other sites. However, the security of the information flow must also be ensured during interactions occurring in cyber-physical systems.

Information flow is often characterized by system behavior or, more significantly, by the behavior of the system objects. Information flow, is due to the impact of one object in the system on another. For example, to establish whether information flows from object A to object B, it is sufficient to establish that A's behavior has an impact on B's behavior and vice-versa. The presence or absence of information flow can thus be tested. For any pair of system behaviors that differ only in the behavior of object A, object B on observing the system, cannot distinguish between these two behaviors and vice-versa, then we can say that an equivalence exists between the behaviors of A and B. For deterministic systems, such equivalences can be made precisely because the behaviors are predictable and sometimes obvious. With cyber-physical processes, many of which are nondeterministic, this is not the case. Capturing this notion of the equivalence of information flow is difficult, and information flow cannot be precisely analyzed due to the nondeterministic nature of these processes. Ideally the physical processes could be controlled by restricting the sys-

tem to allowable information dictated by the cyber framework. This needs an exhaustive simulation or a history-based validation for every event in the system. Capturing such events for all possible behaviors of a cyber-physical system is tedious and it is difficult to build a simulation engine that records every atomic behavior of the system.The problem therefore, requires a different approach.

1.3. MODEL-CHECKING APPROACH TO ANALYZE SECURITY IN CPS

Checking confidentiality in a CPS requires exhaustive investigation of all possible system behaviors to detect any insecure interactions (i.e., any interactions that do not satisfy desired security properties). Therefore, a model-checking approach has been adopted. A big challenge in model-checking properties of a CPSs is to capture the semantics of both the physical and cyber system precisely. In particular, the semantics of physical interactions must be captured in a way that is meaningful to a model checker. As an additional challenge, due to the inter-connectivity of a CPS, information flows through multiple sources. This report captures the physical semantics of a CPS through physical laws of invariance, represents the continuous nature of the physical system as an event-based discretized system, and model check confidentiality properties (in particular, Bisimulation based nondeducibility on Compositions or BNDC) using process algebra for the combined CPS. The main objective of this work is to propose a method to perform a run-time security analysis of the physical system using sophisticated model-checking tools. The model CPS used here is a natural gas transport system that contains a rich interaction of physical flows, physical actions, and cyber actions.

1.4. MODEL PROBLEM: PIPELINE FLOW

The natural gas transport system which is a critical infrastructure is a commodity transport system consisting of a network of pipes. On a subset of these pipes are remote

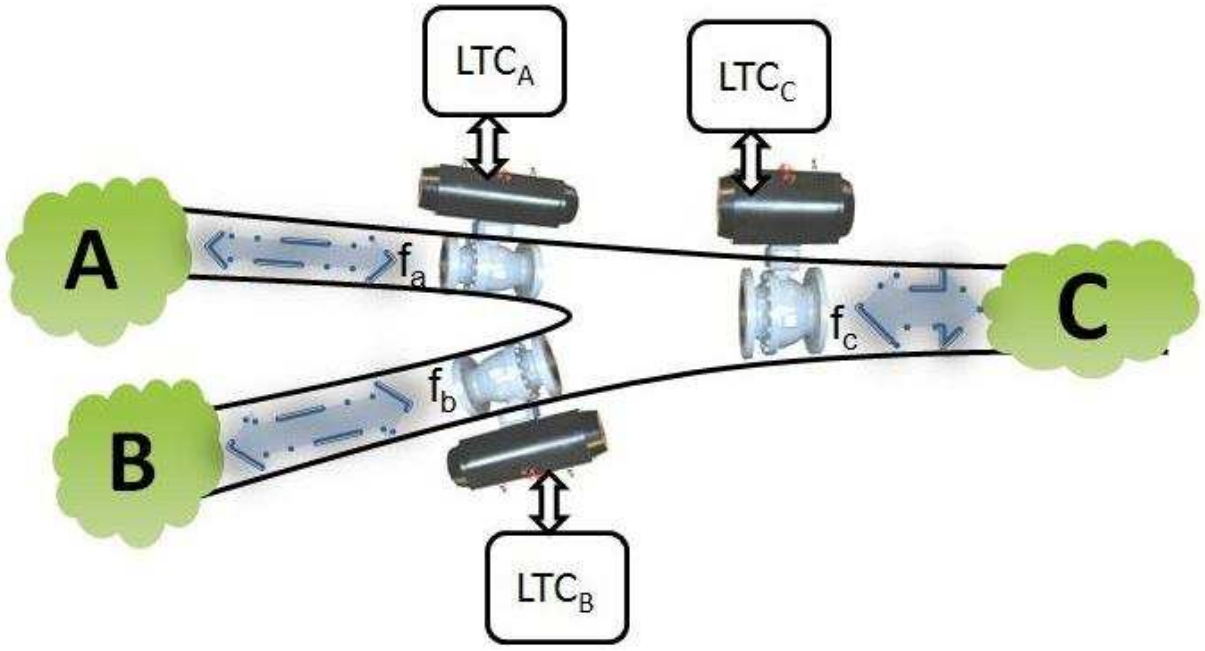

Figure 1.2. Segment of the pipeline system under invariance of physical flow

terminal units (RTUs), used to monitor and effect changes to the state of the gas(such as pressure) within the pipe to which they are attached. The RTUs are controlled by a distributed entity (consisting of multiple computational processes that communicate via cyber message passing) called the long-term control(LTC). The LTC is responsible for interfacing between the cyber and physical frameworks. Each process of the LTC has the ability to observe the state of the gas only within the subset of pipes under its control. Figure 1.2 shows a sample topology. Three operators control subnetworks A, B, and C respectively, with C receiving supply from the gas distribution point. Assume that each LTC can execute certain commands like raise and lower flow, which respectively, raise and lower the flow of the commodity (gas) within its operating pipe. Any of the two commands that LTC at A executes on its pipe will necessarily have an affect on its neighboring subnetworks and as such will constitute an observable event(in the form of a flow change) at both B and C. An invariant on this relationship due to the physics of the

system is that

$$f_c = f_a + f_b \tag{1}$$

where f_a, f_b, and f_c represent the flow of the commodity(gas) in the pipes controlled by operators at A, B, and C respectively [1].

[1]For the sake of simplicity in constructing the invariant, this analysis is limited to incompressible flow

2. BACKGROUND

2.1. SPA

Security process algebra (SPA, for short) [7] [10] is an extension of the calculus of communicating systems (CCS) [19], a language proposed to define concurrent systems. It defines algebra consisting of operators for building systems using a bottom-up approach from smaller subsystems. The basic building blocks are atomic activities, called actions. Unlike CCS, SPA includes actions belonging to two different levels of confidentiality, thus allowing the specification of multilevel systems. The BNF syntax used by SPA to describe a system is [10]:

$$E ::= 0|\mu.E|E_1 + E_2|E_1|E_2|E\backslash L|E\backslash_I L|E/L|E[f]|Z$$

where 0 is the empty process, which can perform no action. The SPA operations used in this work are summarized in Table 2.1. The process $\mu.E$ can perform action μ and behaves like E. The process $E_1 + E_2$ can alternatively choose to behave like E_1 or E_2 and $E_1|E_2$ is the parallel composition of E_1 and E_2, where the executions of the two systems are interleaved. The process $E\backslash L$ can execute all the actions E is able to do, provided that they do not belong to $L \cup \bar{L}$ (where $\bar{L}$ refers to the output); The process $E\backslash_I L$ requires that the actions of E do not belong to $L \cap I$; E/L turns all the actions in L into internal τ's. If E can execute action μ, then $E[f]$ performs $f(\mu)$; Z does what E does, if $Z \equiv_{def} E$. Following typical notation, $\tau \in Tr$ are system traces, $\tau\backslash_x$ is a trace purged of all events in the domain of x, $\tau\ |_x$ is a trace restricted to all events in the domain of x, and $E_1|E_2$ is the parallel composition of event E_1 and E_2. Additionally, $High$,Low are used to represent high-level and low-level security domains with high-level and low-level user in each domain. Also, the symbols I and O are used for inputs and outputs respectively.

SPA operation	Description
$E1 \mid E2$	Parallel Composition of E1 and E2 where executions of the two systems are interleaved
$E \backslash L$	Restrict the system E of all actions belonging to set L
$E1 + E2$	Non-deterministic choice to behave like E1 or E2
$\mu.E$	Performs action μ and behaves like E

Table 2.1. Operations in SPA

2.2. COPS

CoPS is an automatic checker of a multilevel system's security properties [3]. The keywords used in CoPS are explained in Table 2.2.In particular, CoPS checks three non-interference [18] based security properties: bisimulation-based nondeducibility on composition (BNDC), strong bisimulation-based nondeducibility on composition (SBNDC) and, persistent BNDC (P_BNDC) [7] [9] [8].

Keyword	Description
bi	to declare an agent (BInd agent)
basi	to Bind Action Set Identifier
acth	to declare the high actions set (ActH)

Table 2.2. Keywords used in Cops Syntax

2.3. BISIMULATION-BASED NONDEDUCIBILITY ON COMPOSITION

A system is considered to have the Bisimulation-based NonDeducibility on Composition (BNDC) property if it can preserve its security after composition [7]. A system ES is BNDC if for every high-level process P, a low-level user cannot distinguish ES from

$(ES|P)\backslash H$. The term $(ES|P)\backslash H$ stands for the process ES composed with any other process P and purged of all high-level events. In other words, a system ES is BNDC if what a low-level user sees in the system is not modified by composing any high-level process P with ES. Formally,

$$BNDC(ES) \equiv \forall \pi \in E_H, ES\backslash H \approx_B (ES|\pi)\backslash H \tag{2}$$

where $ES\backslash H$ changes all the H events in ES into internal events. A system is BNDC-preserving if the above property holds for all possible behaviors of the system.

2.4. PREVIOUS WORK

Philips et al. [20] performed a broad investigation of the operational and security challenges that SCADA systems typically encounter. Their report includes vulnerability and some available best practices for deploying and maintenance of SCADA systems. It also briefly analyzes multiple levels of flexible AC transmission systems (FACTS) device security issues and the confidentiality, integrity, and availability of a hypothetical electric power grid. However, it proposes no approach nor offers any concrete example for the confidentiality of CFPS.

The North American Electric Regulatory Commission (NERC) provides a basis to define permanent cyber security standards [1]. These standards provide a cyber security framework that identifies and assists with the protection of critical cyber assets to ensure reliable operation of the electric system. They are published in standards CIP-002-1 through CIP-009-1, and they address various security issues in the power system.

Holstein et al.[6] discuss cryptographic issues relating to interfacing SCADA operations with energy management and distribution in terms of protecting the communication packets, authorization, and access controls.

A 2005 Department of Energy publication[2] discusses existing cyber-security standards focusing specifically on control systems for critical infrastructure. The standards it

proposes helps to identify the requirements of a system and develop secure communication protocols and systems.

McDonald et al. [15] investigates SCADA vulnerabilities associated with the energy sector and propose a new approach to overcome such vulnerabilities. Their approach offers an environment called virtual control system environment (VCSE) that supports simulated, emulated, and physical components for investigative analysis of SCADA security. This environment requires an exhaustive modeling of the infrastructure on which it is meant to be implemented. Its vulnerability assessment is limited to then known security threats like man-in-the middle attacks; thus, to secure a system from a new mode of attack, new simulations must be performed to capture new scenarios. A focus on information flow in such CPSs mitigates attack-specific computations by generalizing the secure information flow properties applicable to a generic class of CPS.

Tang et.al[22][23] explored properties like BNDC in FACTS power System using process algebra. This work however, does not generalize those properties to be applicable to any CPS. This thesis is an attempt to generalize the approach to analyze CPS with respect to certain security properties. This thesis also identifies the vulnerability of information flow in a CPS by analyzing the invariant flows in commodity transport system.

Akella et.al [5] [4] discusses the formal methodology of applying security properties: noninterference, inference and nondeducibility to different cyber-physical systems like gas networks, smart house, steel foundry etc. The applied properties and the methodology involved formed a basis for elaborating this thesis.

This work proposes two approaches to analyze the information flow properties of a cyber-physical system. With one approach, we use formal methods in which the events or associated actions of the system are identified, their interactions are manually verified with a known security property as explained in section 3.2. Another approach makes use of sophisticated tools like a model-checker to automatically verify various behaviors of the system for potential violation of a chosen security property. The approach to use these automatic tools is presented in section 4. The use of former approach could often be

cumbersome, due to a CPS‘s complex interactions between several several components. The latter approach is thus suggested; however, the goal of the two approaches is the same. Application of these tools to a general class of cyber-physical systems is explained through a model problem involving natural gas transport system. We use Security Process Algebra (SPA) to model the system and a model-checker called CoPS [3] to validate the system for a chosen security property, BNDC.

3. SYSTEM DESCRIPTION AND ANALYSIS

3.1. SPA ANALYSIS OF GAS PIPELINE SYSTEM

A simple example is used here to demonstrate how this information flow is captured and how we it can be modeled using SPA. A gas pipeline system is shown in Figure 1.2. Actuators are used to automate process control across the sub-sections of the pipeline. An actuator can sense the true reading in the pipe and can also reset it to a desired value. Three possibilities of modeling the system are presented.

3.1.1. Single RTU in a physical system.

A single RTU sitting on a pipeline has control over the commodity (gas or liquid) flowing through the pipe. Also, whenever a eventual change takes place on the physical commodity, information is written to the system‘s cyber elements through the actuators. Such a system can be represented using CSP [13] as follows: Let e1 be an event of change in flow in a pipe segment with a single actuator. Let r1 be change in the reading of the pipeline due to the triggered event e1 (this operation can be represented as e1 $\Leftrightarrow$ r1). An eventual change of reading takes place at the cyber-level once this event is triggered. Because this is an independent process, it does not interfere with the remaining sections of the pipeline. Such a system can be simply represented as “e1 $\rightarrow$ stop $\setminus$ r1”. The following model encodes this system in SPA.

$$PipeLine\_Single = (Behavior1|Object(0, flow)|Object(1, reading))\backslash L$$

$$
\begin{aligned}
Behavior1 = M\_Change(l,x).(if(l == x).then.\\
change(x,y).val(l,y).Behavior1)\\
Else.Behavior1\\
Object(x,y) = \overline{change}(x,y).Object(x,init)
\end{aligned} \tag{3}
$$

In the above SPA syntax, cyber changes and physical changes are represented by two objects called reading and flow respectively. High-level and low-level objects are represented by 1 and 0 respectively. The term *init* stands for the initial reading of the object. Here, $M\_Change(l,x)$ represents events that change the subject of security level l to an object of security level x. The term y is the value (or state) of the object. This SPA describes the behavior of the pipeline system with a single actuator and its possible executions. Similar notation has been used for SPA syntax throughout the thesis.

3.1.2. SPA analysis with no communication among actuators. Figure 1.2 shows a segment of the pipeline system with multiple RTUs. This can be represented as an event based system in which e1 represents change of flow at A, e2 represents change of flow at B, and e3 represents change of flow at C. The behavior of the system can then be captured in SPA as follows:

$$e1 \rightarrow (e2 \rightarrow stop) \sqcap (e3 \rightarrow stop) \sqcap (e2||e3 \rightarrow stop)\backslash(r2 \sqcap r3 \sqcap (r2||r3)) \tag{4}$$

where '$\sqcap$' indicates the nondeterministic choice between the processes. Similarly, e2 and e3 occur initially. The readings at the cyber level then change nondeterministically as

$$r1 \rightarrow (r2 \sqcap r3 \sqcap (r2||r3)) \tag{5}$$

Information flow cannot be measured precisely in this case because A is interfering with B. Such a system can be represented using SPA as below and its encoding into the

CoPS framework is shown in Appendix:

$$PipeLine\_Multiple_{no\_commn} = (Behavior2|Object(0, flow)|Object(1, reading))\backslash L$$

$$\begin{aligned} Behavior2 = {} & M\_Change(l, x, c).(if(x! = l, c == 1).then. \\ & change(x, y).val(l, y).Behavior2).Else \\ & if.(x < l, c == 0).then. \\ & change(x, z).val(l, y).invariant(y, z).Behavior2 \\ & Else.Behavior2 \\ Object(x, y) = {} & \overline{change}(x, y).Object(x, init) \end{aligned} \tag{6}$$

3.1.3. SPA analysis with communication among actuators. The implicit flow of information can be minimized by regulating the physical flow through messages passed among the actuators or cyber components. For example, Pipe C can be stopped from interfering with actions from pipes A and B using the following CSP model.

$$e1 \rightarrow (e1 \rightarrow stop||e2 \rightarrow stop)\backslash(r1||r2) \tag{7}$$

This equation express the desired functionality of the system; practically, however, such a model is too restrictive. At the cyber level, the readings could vary as r1||r2 according to flow adjustments at A and B. The case of message-passing between pipes A and C or pipes B and C is similar. Such a representation, could provide a theoretical basis for capturing the function of the system. First, the information flow could be analyzed in relation to the commodity flow. The SPA model presented below captures the notion of a message-passing between A and C.

$$PipeLine\_system = (Behavior|Object(0, flow)|Object(1, reading))\backslash L$$

$$\begin{aligned} Behavior = {} & M\_Change(l, x).(if(x|l).then. \\ & change(x, p).val(l, q).invariant(p, q).Behavior)+ \\ & change(l, q).val(x, p).invariant(p, q).Behavior \\ & Else.Behavior \\ Object(x, y) = {} & \overline{change}(x, y).Object(x, init) \end{aligned} \tag{8}$$

In the above models, $M\_Change(l, x)$ stands for events that the subject of security level l changes to an object of security level x. The term y is the value (or state) of the object. $x|l$ represents object x being able to communicate with object l to perform a coordinated activity. The above model is generic in that it captures the behavior of the system for all possible pairs of coordinating users. The term *invariant()* captures the atomic actions of recomputing the physical flows as governed by invariant laws (of physics). For example, *invariant(p,q)* would compute the new values of p,q by considering their physical dependencies on the system. Also, the variable c stands for 'controlled;' it is assumed to be 1 if object l controls or is controlled by object x directly.

3.2. MANUAL VERIFICATION OF BNDC PROPERTIES

This section presents the formal approach to determine whether a given system satisfies the BNDC property for the case of pipeline model. First, the system E, followed by the high-level process, Π and the set of high level actions, Act_H must be defined. The condition for BNDC can then be checked as E $\setminus$ $Act_H \cong$ (E | Π)$\setminus$ Act_H. Assuming that A and C coordinate in an attempt to secure information flow between them against B, two group of users can be defined; A and C belonging to *High* and B belonging to *Low* as

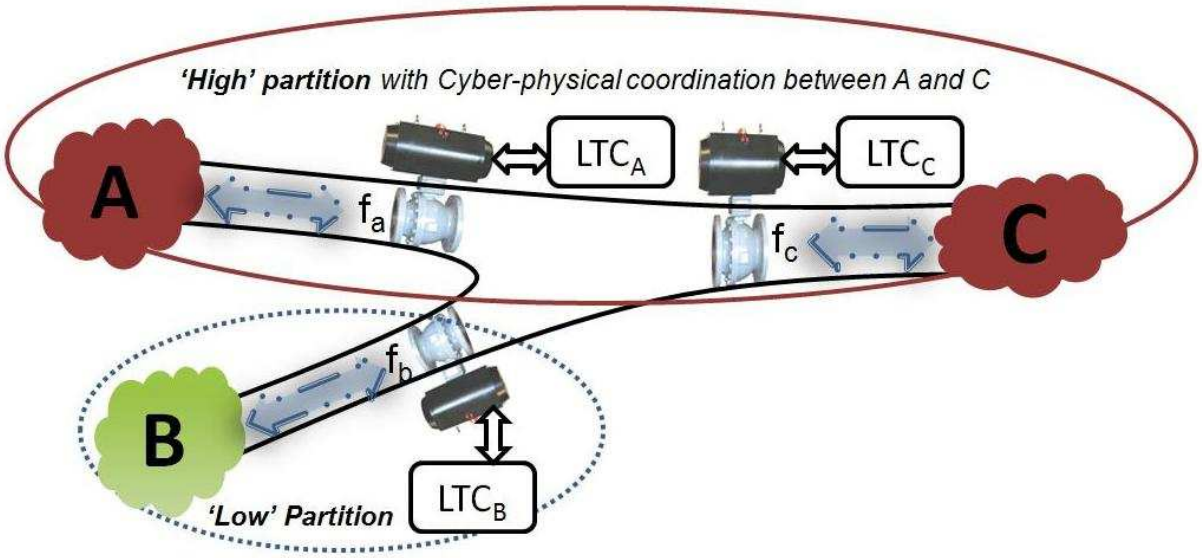

Figure 3.1. Segment of pipeline system showing security partitions

shown in Figure 3.1. The events in the system can be defined as h_A, h_C, and l_B and their respective outputs as $\overline{h}_A$, $\overline{h}_C$, and $\overline{l}_B$. Here, h_A represents a high level change made at A and leading to change at h_C (due to coordination) and eventually, B might experience the change in physical flow at A and C in the form of a low-level output, $\overline{l}_B$.

The system considered can be defined as a process $E = h_A.h_C.\overline{l}_B.\text{E} + l_B.(\overline{h}_A + \overline{h}_C).\text{E}$. This system definition captures the notion that a high-level change made at A or C will result in a causal low-level change at B due to the invariance property of physical flow. Alternatively, a substantial change made at B alone would result at A or C or both in the form of high level outputs $\overline{h}_A$ and $\overline{h}_C$. Considering the high-level process $\Pi = \overline{h}_A.\ \overline{h}_C.$ E and the high-level action set $Act_H\{h_A,\ h_C\}$, the operation $E\backslash\ Act_H \cong l_B.(\overline{h}_A + \overline{h}_C).\text{E}.$ $\overline{l}_B$ is executed only when a coordinated change is initiated between A and C. Composing the high-level process with the system results in all possible interleavings of this high level process with the system. Thus, $(E|\Pi)\backslash Act_H \cong \overline{l}_B.\text{E} + l_B.(\overline{h}_A + \overline{h}_C).\text{E}$. Although the high-level events h_A and h_C are rejected, the high-level process $\overline{h}_A.\ \overline{h}_C$ simulates the change at A and C that leads to a low-level change at B. In this case, $E\backslash Act_H \ncong (E|\Pi)\backslash Act_H$. This scenario fails to satisfy the BNDC property, indicating that

the system permits for changes at one group of users to be deduced by others as a result of the flow dependence in the composite system, expressed as in equation 1.

3.3. DISCRETE EVENT SYSTEM

Each event in the physical system must maintain the flow invariant (Equation 1) in the physical network. For example, to satisfy the invariant $f_a + f_b = f_c$, innumerable combinations of values for f_a, f_c and f_b are possible. By discretizing the values, the invariant physical flow in the network can be captured following the event that caused it. Also, discretization confines the security analysis to a limited number of states, thereby simplifying the encoding process for model-checking. In a way, the events are encoded along with these discrete values to define specific allowable system. Now, when a change event at a component results in a flow change there, e.g., at f_a, the system could have a finite set of values of f_b and f_c that satisfy the invariant. If three discrete values each

$\mathbf{f_a}$	$\mathbf{f_b}$	$\mathbf{f_c}$
0	0	0
k	0	k
k/2	0	k/2
0	k/2	k/2
k	k/2	3k/2
k/2	k/2	k
0	k	k
k	k	2k
k/2	k	3k/2

Table 3.1. Maintaining the flow invariant

are allowed for f_a, f_b and f_c (e.g., 0, k/2, k), then the system could fall in to any of nine possible states. This is reasonable because the task is to analyze the security with respect to change following an event, not to determine exactly the magnitude of change.

However, to provide a broader analysis, the state space could be relaxed by including more quantized values. For example, five values could be assigned in the previous case with in {0, k/4, k/2, 3k/4, k}. Three discrete levels capture the significant changes in the system and limit an explosion in the number of states. The possible system states resulting from three discrete values are shown in Table 3.1 for physical flow.

4. MODEL CHECKING CPS SECURITY: RESULTS

4.1. ASYNCHRONOUS PHYSICAL DOMAIN PROTECTION

One approach is to treat the physical changes as being protected and to asynchronize the actions of the cyber processes. Asynchronous physical domain protection model incorporates this approach. This is a system analogous to shared memory access in which only one user is allowed to access the system at a time to achieve concurrency control and enforce strict integrity. The idea is to allow the change brought about by the coordinated activity of A and C or activity initiated by B alone. Both these events cannot occur simultaneously. That way, high-level physical flow change can be treated as a result of a pre-operation from one of the partitions.

4.1.1. Protecting flow within *High* partition against *Low*. If two parties wish to perform a coordinated activity on a channel, their actions should be protected from being deduced by a third party. In our case, the pipe is strictly shared and constrained by invariance of physical flow. The discretization approach proves that such a phenomenon can be achieved where in BNDC property is satisfied. The idea here is to allow the system to transit to a new state in which B retains its discrete level and A and C rearrange their discrete levels such that the invariance of physical flow is preserved. This ensures that (low level) activity at B is never impacted by the (high level) activity between A and C. The terms high level and low level need not necessarily indicate a hierarchical distinction; they may simply represent two different groups or partitions. Securing the commodity flow between A and C (belonging to partition *High*) against B (belonging to partition *Low*) results in the configuration graphed in Figure 4.1. The CoPS modeling of this model is presented below. In CoPS, the system is defined in terms of agents or

processes interacting with one another. The keyword *bi* is used to bind the agent by declaring and defining it. The keyword *basi* is used to define a list of actions and bind it to a set identifier. It gives the ability to identify and label the actions as belonging to security partitions (*High* or *Low*). The keyword *acth* is used to declare the high actions set (ActH) of the system. Any set not defined will be treated as an empty set in CoPS.

```
//this simulation is for the commodity pipeline
//system assuming communication between A and C
//to protect their interactions against B
//Discrete values {'val1 = 0 ; 'val2 = k/2 ;'val3 = k}
//Invariant: A+B=C
//This code satisfies SBNDC Property in both the cases and also
//with compositionality
bi Action
   (Action1 | Action2)\N
bi Action1
    (A_Writes | C_Writes)
// Restricting B from all the high-level physical activity
bi Action2
  (B_Writes)\H
//Discretization of events in the physical system
bi State
 (State_1 + State_2 + State_3 + State_4 + State_5 + State_6 + State_7 +
 State_8 + State_9)\H
//Exploration of all possible sub-states the system can
//enter in to depending on the values of flows at A,C
//and B satisfying the physical property on invariance.
bi State_1
```

```
  'w_a.valA_1. 'w_b.valB_1.'w_c.valC_1.State_1
bi State_2
  'w_a.valA_2.'w_b.valB_1. 'w_c.valC_2.State_2
bi State_3
  'w_a.valA_3. 'w_b.valB_1.'w_c.valC_3.State_3
bi State_4
  'w_a.valA_1.'w_b.valB_2. 'w_c.valC_2.State_4
bi State_5
  'w_a.valA_2. 'w_b.valB_2.'w_c.valC_3.State_5
bi State_6
  'w_a.valA_1. 'w_b.valB_3.'w_c.valC_3.State_6
bi State_7
  'w_a.valA_3.'w_b.valB_2. 'w_c.valC_4.State_7
bi State_8
  'w_a.valA_3. 'w_b.valB_3. 'w_c.valC_5.State_8
bi State_9
  'w_a.valA_2.'w_b.valB_3.'w_c.valC_4.State_9
//Define the cyber and physical events performed by the
//operators at A,B and C
bi A_Writes
   (change_a.'w_a.State)
bi B_Writes
   (change_b.'w_b.State)
bi C_Writes
  (change_c.'w_c.State)
// all physical changes are classified as high level actions
basi H
'w_a 'w_c 'w_b
```

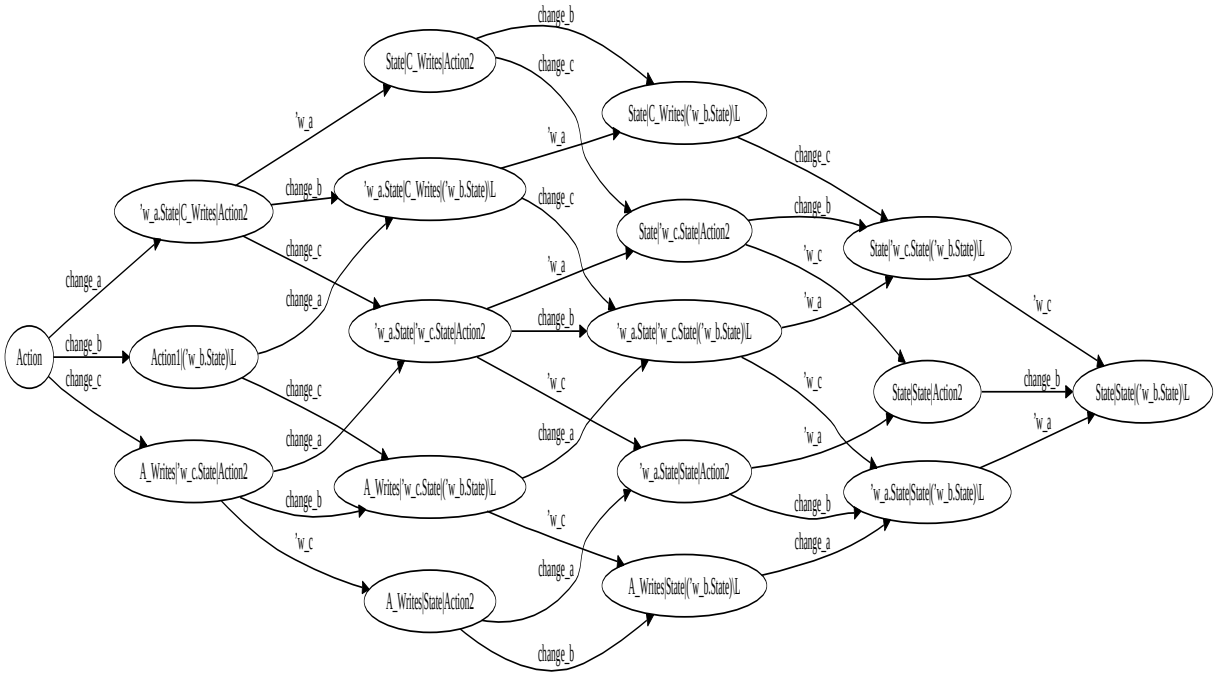

Figure 4.1. Protecting flow against *Low* partition $\{B\}$ from a physical change at *High* partition $\{A, C\}$

```
basi N
valA_1 valA_2 valA_3 //discrete values possible
valB_1 valB_2 valB_3
valC_1 valC_2 valC_3 valC_4 valC_5 // valC_4: 3k/2, valC_5: 2k
acth
change_a change_b change_c //readings at cyber level
'w_a 'w_b 'w_c
```

4.1.2. Bisimilarity Equivalence. Section4.1.1 established a secure flow between A and C (of one partition) against B (of another parton). To strengthen the notion of bisimilarity equivalence in their relationship, the system is model-checked to determine whether events at B are secure with respect to partition *High* (containing A and C). The resulting state graph in Figure 4.2 demonstrates that it is possible to get in to a state where only value at B changes with no impact on the flow at A or C. However, in the

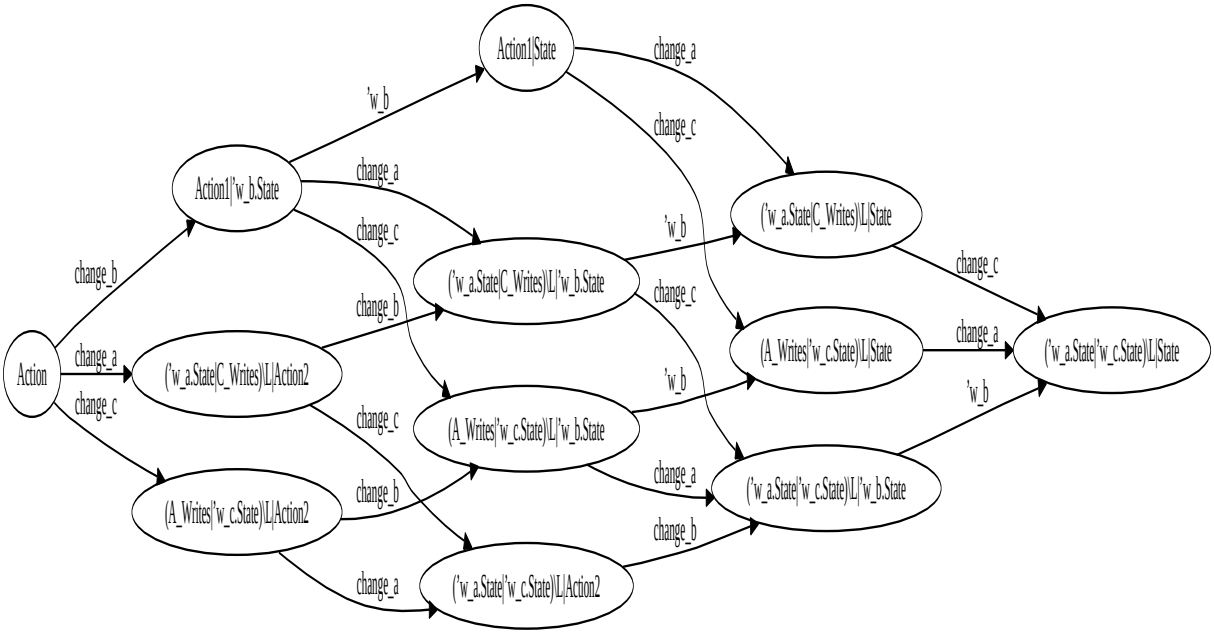

Figure 4.2. Protecting flow within *Low* partition $\{B\}$ against *High* partition $\{A, C\}$

physical system, it is not possible to change the level of B unless A or C is prepared to rewrite its flow. This limitation indicates that the discretization allows minute changes to occur in B. The final state shown in Figure 4.2 demonstrates that events producing a slight change in the value of flow at B will not be enacted by a physical invariance that changes the flow at A or C. The results of model-checking for both the cases are tabulated in Table 5.1. The CoPS modeling of this system is presented below.

```
//this simulation is for the commodity pipeline
//system to check if events changing flow at B
//would allow a causal flow change at other
//partition (includes A and C)
//Discrete values {'val1 = 0 ; 'val2 = k/2 ;'val3 = k}
//Invariant: A+B=C
//This code satisfies SBNDC Property in both the cases and also
//with compositionality
```

```
//val1 = 0 ; val2 = k/2 ; val3 = k
//invariant: A+B=C
bi Action
   (Action1 | Action2)\N
bi Action1
 (A_Writes | C_Writes)\H // Restrict A and C from performing any physical
                         // change (High-level)while B is writing the flow
bi Action2
  (B_Writes)
bi State
 (State_1 + State_2 + State_3 + State_4 + State_5 + State_6 + State_7
 + State_8 + State_9)\H
bi State_1
  'w_a.valA_1. 'w_b.valB_1.'w_c.valC_1.State_1
:
:
//Similarly defined other states..
bi A_Writes
   (change_a.'w_a.State)
bi B_Writes
   (change_b.'w_b.State)
bi C_Writes
  (change_c.'w_c.State)
// all physical changes are classified as high-level actions
basi H
'w_a 'w_c 'w_b
basi N
valA_1 valA_2 valA_3 //discrete values possible
```

```
valB_1 valB_2 valB_3
valC_1 valC_2 valC_3 valC_4 valC_5 // valC_4: 3k/2, valC_5: 2k
acth
change_a change_b change_c //readings at cyber level
'w_a 'w_b 'w_c
```

4.2. SYNCHRONOUS PARTITION PROTECTION MODEL

A refined model of Asynchronous physical domain protection model is the synchronous partition protection model in which the physical changes are classified as belonging to High and Low partitions and thereby allowing simultaneous operations of the two partitions. This essentially means that the system inherently considers allowable physical changes initiated by operators belonging to different partitions. We could have more levels of discretization to allow realistic values for flow at different operators. The results of model-checking for three-level discretized system and a five-level discretized system are tabulated in table 5.2.

```
//This code satisfies SBNDC Property in synchronous partition
//protection model with three-level discretization.
//val1 = 0 ; val2 = k/2 ; val3 = k
//invariant: A+B=C
bi Action
   (Action1 | Action2)\N
bi Action1
    (A_Writes | C_Writes)\L // Restrict A and C from performing
                            //  a low-level physical change
bi Action2
  (B_Writes)\H          // Restrict B from performing a
                        // high-level physical change
```

```
bi State
 (State_1 + State_2 + State_3 + State_4 + State_5 + State_6
 + State_7 + State_8 + State_9)
bi State_1
 'w_a. valA_1. 'w_c. valC_1.'w_b. valB_1.State_1
  //Similarly defined other sub-states
bi A_Writes
   (change_a.State)
bi B_Writes
   (change_b.State)
bi C_Writes
   (change_c.State)
basi L                  // Force physical change at B to
'w_b                    // Low-level partition
valB_1 valB_2 valB_3
basi H                  // Physical changes at A and C fall
'w_a 'w_c               // under High-level partition
valA_1 valA_2 valA_3
valC_1 valC_2 valC_3 valC_4 valC_5
basi N
valA_1 valA_2 valA_3 //discrete values possible
valB_1 valB_2 valB_3
valC_1 valC_2 valC_3 valC_4 valC_5 // valC_4: 3k/2, valC_5: 2k
acth
change_a change_b change_c //readings at cyber level
'w_a 'w_b 'w_c
```

5. RESULTS

The results of this work are summarized in Tables 5.1 and 5.2. Preservation of confidentiality, under the BNDC model, in commodity transport systems has been possible with discretization of the states following the invariant on physical flow.

They indicate that BNDC properties hold in the system with finite state transitions. Both type of systems discussed above satisfy BNDC once the physical property is captured. With more discretization levels in the synchronous partition protection model, states explored before reaching a stable state have increased six-fold by shifting from three-level discretization to a five-level discretized system. This result could mean that number of BNDC verified states increase with more finite discretization of physical events in the system. The statistics summarized in Table 5.1 and 5.2 indicate that it is practically possible to establish a protected flow in the pipeline system.

Asynchronous physical domain protection	**BNDC**	**Generated Graph**	**Compo -sable**
Protection of Flow against B due to a physical change at A,C	Yes	V:18 E:33	Yes
Protection of Flow against A,C due to a physical change at B	Yes	V:12 E:20	Yes

Table 5.1. Results of model-checking for Asynchronous physical domain protection

Synchronous partition protection	**BNDC**	**Generated Graph**	**Composable**
Three-level discretization	Yes	V:242 E:561	Yes
Five-level discretization	Yes	V:1458 E:3537	Yes

Table 5.2. Results of model-checking for Synchronous Partition Protection

6. CONCLUSIONS

The primary goal of this work is to illustrate a methodology to capture physical semantics of a commodity flow network in a cyber framework so as to establish a secure information flow in such systems. However, the greatest challenge was to model a CPS in which physical information is constrained by the invariant on the commodity flow in such CPSs. We could include this notion of invariant on the commodity flow with in the pipeline by discretizing the events causing the change of flow. With such a method, we could model the system as a deterministic state model with discrete values of flow within its physical components. Having done so, we were able to encode the infrastructure for a model checker to validate the BNDC properties in the system.

In order to apply this methodology to a large scale system, compositionality is fundamentally necessary. By verifying the properties for individual system components and composing them until we construct the massive system, we can achieve scalability with this approach. However, this could be computational intensive due to the enormous overhead incurred by model-checking. Application of this approach to large-scale systems will provide direction for further research in this area. Another interesting problem is the identification of a vulnerable node of a CPS, compromising which BNDC properties in the composite system fail. Such vulnerable node detection would enable us to select suitable nodes for coordination, thereby ensuring that the selected security property holds for the system. BNDC proved to be an important property for CPSs because of the inherent composition of various cyber and physical elements. The methodology adopted in this thesis provides a direction for future research in model-checking cyber-physical systems for information flow security.

7. APPENDIX

7.1. COPS ANALYSIS OF PIPELINE SYSTEM WITH NO COMMUNICATION AMONG ACTUATORS

```
//this simulation is for the commodity pipeline system assuming no
//communication between the actuators; It captures the behaviors of
//the system for every change performed by any of the operators

bi States
 ( A_writes | B_writes| C_writes )\H

bi A_writes
   w_a.'val_A.B_writes + w_a.'val_A.C_writes

bi B_writes
    w_b.'val_B.A_writes + w_b.'val_B.C_writes

bi C_writes
    w_c.'val_C.A_writes + w_c.'val_C.B_writes

basi N
val_A val_B val_C

acth
```

```
val_A val_B val_C
w_a w_b w_c
```

7.2. COPS ANALYSIS OF PIPELINE SYSTEM WITH COMMUNICATION AMONG ACTUATORS

```
//this simulation is for the commodity pipeline system assuming
//communication between the actuators, meaning the cyber components.
bi States
 (A_comm_C | A_comm_B | B_comm_C | A_writes | B_writes | C_writes)\H

bi A_comm_C
  w_a.'val_A.A_writes + w_c.'val_C.C_writes

bi A_comm_B
 w_a.'val_A.A_writes + w_b.'val_B.B_writes

bi B_comm_C
 w_b.'val_B.B_writes + w_c.'val_C.C_writes

bi A_writes
   w_a.'val_A.A_writes

bi B_writes
    w_b.'val_B.B_writes

bi C_writes
```

```
    w_c.'val_C.C_writes


basi N
val_A val_B val_C


acth
val_A val_B val_C
w_a w_b w_c
A_comm_C  A_comm_B  B_comm_C
```

BIBLIOGRAPHY

[1] *Standard CIP-002-1 through Standard CIP-009-1, ftp://www.nerc.com/pub/sys/all_updl/standards/sar/ Cyber_Security_Standards_Board_Approval_02May06.pdf.*

[2] *A Summary of Control System Security Standards Activities in the Energy Sector.* DOE, Office of Electricity Delivery and Energy Reliability, 2005.

[3] Cops. *http://www.dsi.unive.it/ mefisto/CoPS/index.php* (accessed December 30, 2008).

[4] Akella, R., and McMillin, B. Security in cyber-physical systems. *Technical report, Computer Science department, Missouri S&T* (2008).

[5] Akella, R., McMillin, B., and Service, T. Teaching of security in cyber-physical systems. In *12th National Colloquium for Information Systems Security Education* (2008).

[6] D.Holstein, J.Tengdin, J.Wack, R.Butler, T.Draelos, and P.Blomgren. *Cyber Security for Utility Operations.* Sandia National Laboratory, 2005.

[7] Focardi, R., and et al. The compositional security checker: A tool for the verification of information flow security properties. *IEEE Transaction on Software Engineering 23*, 9 (Sept. 1997).

[8] Focardi, R., Gorrieri, R., , and Martinelli, F. Information flow analysis in a discrete-time process algebra. In *Proceedings of the 13th IEEE Computer Security Foundations Workshop (Csfw'00)* (2000).

[9] Focardi, R., Gorrieri, R., , and Martinelli, F. Real-time information flow analysis. *IEEE Journal on Selected Areas in Communications 21*, 1 (Jan. 2003).

[10] Focardi, R., and Gorrieri, R. A classification of security properties for process algebras. *Computer Security 3*, 1 (1994/1995), 5–33.

[11] Goguen, J. A., and Meseguer, J. Security Policies and Security Models. In *Proc. of the IEEE Symposium on Security and Privacy (SSP'82)*, IEEE Computer Society Press.

[12] Goguen, J. A., and Meseguer, J. A Model of information. In *Proc. 9th Natl. Computer Security Conference* (1986).

[13] Hoare, C. A. R. *Communicating Sequential Processes*. Prentice Hall, 1985.

[14] Lee, E. A. Cyber-Physical Systems - Are Computing Foundations Adequate? In *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap* (October 16 - 17, 2006).

[15] McDonald, J., Conrad, N., Service, C., and Cassidy, H. *Cyber Effects Analysis Using VCSE:Promoting Control System Reliability*. Sandia National Laboratory, SAND2008-5954, 2008.

[16] McLean, J. Security models and information flow. In *Procs. of the 1990 IEEE Computer Society Press* (1990), IEEE Computer Society Press.

[17] McLean, J. *Encyclopedia of Software Engineering - Security Models*. 1994.

[18] McLean, J. A general theory of composition for a class of 'possibilistic' security properties. *IEEE Transactions on Software Engineering 22*, 1 (Jan. 1996), 53–67.

[19] Milner, R. *Communication and Concurrency*. Prentice Hall, 1989.

[20] Phillips, L. R., Baca, M., Hills, J., Margulies, J., Tejani, B., Richardson, B., , and Weiland, L. *Analysis of Operations and Cyber Security Policies for a System of Cooperating Flexible Alternating Current Transmission System (FACTS) Devices*. Sandia National Laboratory, SAND-2005-7301, 2005.

[21] Sun, Y., McMillin, B., Liu, X. F., and Cape, D. Verifying Noninterference in a Cyber-Physical System: The Advanced Electric Power Grid. In *Proceedings of the Seventh International Conference on Quality Software (QSIC)* (Portland, OR, October 2007), pp. 363–369.

[22] Tang, H., and McMillin, B. Analysis of the security of information flow in the Advanced Electric Power Grid using Flexible Alternating Current Transmission System (FACTS). In *Critical Infrastructure Protection* (2008), Springer, pp. 43–56.

[23] Tang, H., and McMillin, B. Security Property Violation in CPS through Timing. In *Proceedings of the 1st Workshop on Cyber-Physical Systems (part of ICDCS)* (2008), IEEE Computer Society Press.

[24] Zakinthinos, A., and Lee, E. A general theory of security properties. In *Procs. of the 18th IEEE Computer Society Symposium on Research in Security and Privacy* (1997).

VITA

Ravi Chandra Akella was born on November 19, 1984 in Andhra Pradesh in southern India. He completed his schooling there and received his Bachelors degree in information technology from Andhra University in 2007. Later, he obtained his Masters degree in computer science in May 2009 from Missouri University of Science and Technology (formerly, University of Missouri, Rolla) under the guidance of Dr. Bruce McMillin. His major interests are in distributed systems and security.