Spring 2013

Verification of information flow security in cyber-physical systems

Rav Akella

Department: Computer Science

VERIFICATION OF INFORMATION FLOW SECURITY IN CYBER-PHYSICAL SYSTEMS

by

RAVI CHANDRA AKELLA

A DISSERTATION

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

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

2013

Approved
Dr. Bruce McMillin, Advisor
Dr. Sriram Chellappan
Dr. Sanjay Madria
Dr. Wei Jiang
Dr. Jonathan Kimball

ABSTRACT

With a growing number of real-world applications that are dependent on computation, securing the information space has become a challenge. The security of information in such applications is often jeopardized by software and hardware failures, intervention of human subjects such as attackers, incorrect design specification and implementation, other social and natural causes. Since these applications are very diverse, often cutting across disciplines a generic approach to detect and mitigate these issues is missing. This dissertation addresses the fundamental problem of verifying information security in a class of real world applications of computation, the Cyber-physical systems (CPSs).

One of the motivations for this work is the lack of a unified theory to specify and verify the complex interactions among various cyber and physical processes within a CPS. Security of a system is fundamentally characterized by the way information flows within the system. Information flow within a CPS is dependent on the physical response of the system and associated cyber control. While formal techniques of verifying cyber security exist, they are not directly applicable to CPSs due to their inherent complexity and diversity. This Ph.D. research primarily focuses on developing a uniform framework using formal tools of process algebras to verify security properties in CPSs. The merits in adopting such an approach for CPS analyses are three fold- i) the physical and continuous aspects and the complex CPS interactions can be modeled in a unified way, and ii) the problem of verifying security properties can be reduced to the problem of establishing suitable equivalences among the processes, and iii) adversarial behavior and security properties can be developed using the features like compositionality and process equivalence offered by the process algebras.

ACKNOWLEDGMENTS

First of all, I would like to express my deep gratitude to my advisor, Dr. Bruce McMillin. His constant encouragement right from the beginning, time and concern for my research progress have been valuable to this work. He has provided me with valuable opportunities to discuss and refine my thoughts in this area of research. I will cherish the culture, thought process, critical thinking and professionalism that he has imparted in me.

I thank all my advisory committee members for their valuable services in spite of their busy schedules. Some of the ideas in this dissertation have been developed during the course of research meetings with my labmates to whom I am grateful - Thoshitha Gamage, Thomas Roth, Stephen Jackson, Li Feng and Derek Ditch. I greatly acknowledge the Free Renewable Electric Energy Delivery and Management (FREEDM) systems center for funding my work under the grant NSF EEC-0812121. I do not want to miss this opportunity to thank the staff in the computer science department and administrative offices at S&T who have spent their time and energy for me in the background.

My life in the department and in Rolla has been greatly spiced up by the company of Brijesh Chejerla, Ravi Arvapally, Deepak Somayajula, Rakesh Gudavarthy and a big list of friends from this wonderful community at S&T to whom I am thankful. A very special thanks to this person who has been a great influence on me- Neelanjana Dutta-for everything that you are! A very special mention goes to my brother Raja Akella, Aruna Vadina, Ani and Honey for their love and concern. Finally, I dedicate this dissertation to my Dad & Mom- love you!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

CPS	Cyber-Physical System
SPA	Security Process Algebra
FREEDM	Free Renewable Electric Energy Delivery and Management
SST	Solid State Transformer
DRER	Distributed Renewable Energy Resource
DESD	Distributed Energy Storage Device
LOAD	House Load
DGI	Distributed Grid Intelligence
IEM	Intelligent Energy Management

1. INTRODUCTION

Critical infrastructure refers to a wide range of systems that deliver critical services to the society in a reliable, safe, dependable and secure fashion. The complexity of critical infrastructure is dramatically increasing. Electric power transmission and distribution, air traffic control, and cruise control for automobiles, among many other systems, will soon become "smart grids" for managing electric power, automated air traffic management for aircraft routing, and "smart" cruise control for automobiles. Unlike their predecessors, these modern systems include not only physical components, but also software. These integrated systems are examples of Cyber-Physical Systems (CPSs). Formally, CPSs are integrations of computation with physical processes [1]. Potential CPSs include high-confidence medical devices and systems, traffic control and safety, advanced automotive systems, process control, energy conservation, environmental control, avionics and instrumentation.

The complexity of a CPS results from the highly networked and distributed nature of the physical and cyber components within. Smart Grid systems potentially include thousands of power lines, transformers, meters, power flow controllers, and communicating embedded computers. Air traffic control is comprised of airplanes with communicating embedded computers. Smart cruise control consists of embedded computers in automobiles that potentially communicate with surrounding automobiles on the highway. All these components interact with each other, sometimes in unpredictable ways. Reliance on the critical infrastructure means reliance on the physical components and cyber components, and their interactions. A significant concern is the increased number of places within a CPS that are vulnerable to attack and/or failure. The vulnerability of critical infrastructure to cyber and physical failures was apparent in the 2003 U.S. blackout [2] and the recent findings

of cyber malware like the Stuxnet worm [3] [4] that compromise supervisory control and data acquisition (SCADA) systems. New attack models that are infrastructure dependent are discussed in literature, like the false data injection attacks in SCADA systems [5].

1.1. UNDERSTANDING SECURITY REQUIREMENTS FOR A CPS

Security of a system is generally described in terms of availability, integrity, and confidentiality. Availability refers to the ability to use the information or resource as desired. Integrity refers to the prevention of unauthorized or unintended change of data or resources. Confidentiality is the confinement of information or resources to the trusted entities within a system [6]. All three aspects of security are necessary in a CPS. Conventional concepts like authentication, access control and cryptographic methods don't always offer secure ways of dealing with information, especially when computation involves physical entities as in CPSs. An attacker who is an insider or who can compromise specific cyber components, has access to more information by associating the received cyber information with his physical observability. Hence, it is important to ensure the confidentiality within the system to prevent the attacker from obtaining critical information, that could be used to perform availability and integrity attacks. Understanding, modeling, and ensuring security in a CPS is a significant challenge because any analysis must account for both the cyber and physical components of a system and their interactions. There is an interdependence of events within the cyber and physical domains often leading to more fundamental security issues. The primary challenge, particularly when considering cyber-physical interactions, is to find a uniform semantic basis for the analysis. Confidentiality, in particular, has received lesser attention in the context of CPSs due to the lack of a unified theory. Information flow policies provide an appealing semantic basis in

quantifying CPS interactions since they define the way information moves throughout a system [6]. The security analysis of a CPS, therefore, requires an analysis of the cyber components, the physical components, and, most importantly, the interactions among them. Typically, these components interact as shown in Figure 1.1.

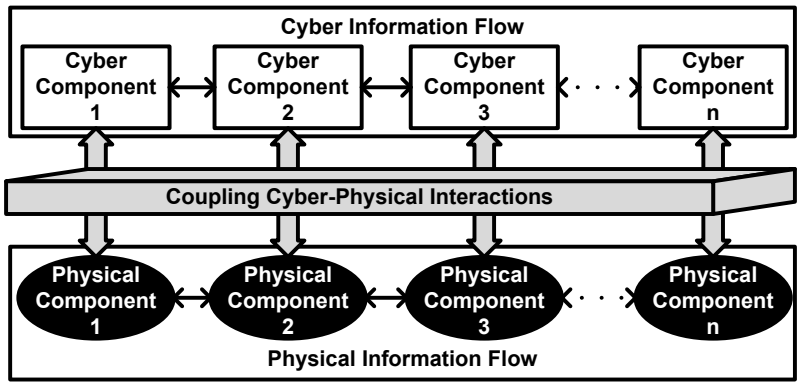

Figure 1.1. Cyber-physical interactions

1.1.1. Cyber Information Flow. The purely cyber portion of a CPS includes interconnected cyber components that exchange data to compute actions or responses. Cyber systems are vulnerable to worms, viruses, denial-of-service attacks, malware, phishing, and user errors that compromise integrity and availability. Confidentiality is a prime aspect of many cyber systems. A great deal of analysis is concerned with ensuring data confidentiality through well-known mechanisms such as cryptography. Beyond this, information flow properties for a general class of deterministic and non-deterministic systems have been addressed [7] [8]. These classical models of information flow security are concerned with quantifying information that is downgraded via covert channels to observers. The complexity of securing cyber information space in CPSs lies in tying the cyber components to the physical system and thereby, the cyber-physical interactions.

1.1.2. Physical Commodity Flow. The purely physical portion of a CPS includes interconnected physical components that perform or control certain physical actions like monitoring physical commodity flow. Commodity refers to the main resource that is transported over an infrastructure; for example, gas is the commodity transported over a gas pipeline system and power is the commodity distributed over an electric power grid. The infrastructures are usually controlled at specific geographic locations and consist of several physical components such as power lines, buses, pipes, and joints. Commodity flow changes due to usage or control settings and are governed primarily by the laws of physics and the topology of the physical system. Physical commodity flow is governed by the concept of invariant flow of a physical entity. For example, the flow in a gas pipeline changes in accordance with the laws of gas flow, and power flowing through every branch of a power grid varies according to Kirchhoff's laws. Physical systems have a separate set of vulnerabilities that expose them to physical attacks that can affect both availability and integrity. A CPS is inherently exposed to the outside world due to its physical nature; automobile and aircraft movements can be observed, pipeline and electric power flow can be measured. These observations yield *information* about the system and its underlying control processes and settings. The interconnection topology of a system, coupled with the observations from physical flow, provide information. In general, however, it is difficult to prevent unauthorized or undesirable information flow within a physical system.

1.1.3. Cyber-Physical Interactions. Cyber-physical interactions result from the coupling of the information and commodity flows in a CPS (represented by the center plane in Figure 1.1). In a CPS, cyber processes interact with physical components by reading their physical states and actuating the controlled physical components. Clearly, there is an interdependence of actions in the cyber and physical domains. Vulnerabilities are a natural consequence of this interdependence; an

action in one part of the system is causally felt in other parts of the system, leading to information flow leakage at the cyber-physical boundary. In other words, an observation about commodity flow could permit an observer to infer sensitive cyber actions.

Security considerations for a CPS, therefore, depend on cyber information flow, physically observable behavior, and the interactions among the cyber and physical components of the system. Due to infrastructure interdependencies [9] [10], a compromise in the security of one system may threaten another system. For example, a failure of security at a fuel plant that depends on a gas pipeline to generate power for the electric power grid affects the security of the power grid. Timing, security [11] and frequency [12] are key properties that have an impact on the confidentiality of a system. Information flow analysis is a fundamental concept in theory, that is generally used to reason about the security violations due to the way information moves within the system. A theoretical basis for information flow analysis is based on security models outlined by McClean [8] [13] [14] and Zakinthinos et al. [15]. The complexity of CPS interactions however exceeds the ability of informal information flow analysis methods.

1.2. CONTRIBUTIONS OF THIS WORK

Automating the process of verifying information flow properties in CPSs requires specification of the system behavior within a formal framework. Process algebras [16] provide both a rigorous system specification and associated verification (model checking) procedures, but have been rarely used because of their complexity. However, formal methods researchers have made automated theorem proving and model checking techniques more capable and efficient in the recent years, to formally verify many properties of distributed and real systems. The significant contribution of this work is to develop a formal framework that satisfies the following objectives:

- It should provide a robust specification of a real system that captures the distributed, concurrent, asynchronous and mobile nature of interactions among various processes.

- It should provide a semantically uniform way of bridging the diverse cyber and physical spaces.

- It should facilitate the automatic verification of information flow security properties.

This work primarily adopts the process algebraic approach to accomplish the above objectives. The main findings discussed in this dissertation have been reported in the following publications:

1. In [17], the specification of a CPS was presented in a basic process algebra, the Security Process Algebra (SPA). Process algebraic techniques like bisimulation were used to define information flow security properties in SPA. Manual proofs were presented to verify non-deducibility properties for different cases of an attacker who can break the confidentiality within the system.

2. In [18], a general approach was laid out to automate the verification of information flow properties for any CPS. A test CPS was used to demonstrate the process and the results of automatic verification of non-deducibility on the test CPS was presented.

3. In [19], information flow properties were developed and verified in an advanced process algebra, the π-calculus. It was shown that the verification using π-calculus was more robust because of its rich features to model complex aspects of distributed computation, and the ability to define a wide variants of basic information flow properties.

4. The process algebraic verification of security properties in CPSs was demonstrated in this dissertation with the help an advanced smart grid architecture that was published in [20] [21].

5. Prior work that formed the basis of this Ph.D research can be found in [22], [23], and [24].

The remainder of this dissertation is organized as follows. In Section 2, a study of existing CPS security methods is presented and the case for information flow analysis is established. Section 3 includes a discussion of formal specification and verification techniques that could be adopted for CPS analyses. In Section 4, information flow properties that are attractive to CPS architectures are described. The essence of this work is explained with the help of a running example of a distributed CPS introduced in Section 5. Section 6 illustrates the approach involved in specification and verification of information flow properties in CPSs. In Section 7, an advanced process algebra, the π-calculus, is adopted to address some of the challenges encountered using conventional process algebras for CPS analyses. It will be shown that the π-calculus facilitates a better reasoning of security in CPSs due to its rich features and tool support for automatic verification illustrated in Section 8. Finally, the key findings of research performed towards this dissertation are discussed in Section 9.

2. STATE OF THE ART IN CPS SECURITY

The existing work on CPS security can be classified into the following categories based on what they address- i) standards for CPS security, ii) impact of cyber security on the secure operations of the physical system, iii) impact of physical system security on secure cyber operations, and iv) impact of cyber-physical interactions on CPS security.

2.1. STANDARDS FOR CPS SECURITY

The National Institute of Standards and Technology (NIST) has released guidelines to address the growing concerns of smart grid security [25]. The report presents recommendations on architectures, requirements and strategies that lead to improved security, privacy and reliability of smart grid. Some of the reasons that pose additional risk to the grid include increased complexity of the grid due to interconnected networks leading to increased entry points for potential adversaries and the impact of coordinated cyber-physical attacks. Confidentiality is an increasing concern to protect i) the privacy of the consumer, ii) the electric market information and iii) the power company. Detection of covert channels and eliminating them during runtime is extremely necessary. The use of formal methods-based techniques like information flow analysis is suggested.

The North American Electric Regulatory Commission (NERC) has defined cyber security standards [26] for electric energy infrastructure. These standards are intended to provide a cyber security framework to identify risks, help secure critical cyber assets, and ensure that the electric power grid operates reliably. In particular, Standards CIP-001-2 through CIP-009-4 address security issues in the bulk electric

system such as the identification of critical cyber assets and their physical protection, reporting of sabotaged behavior, incident reporting and recovery plans, security management control, etc.

A Department of Energy (DOE) publication [27] discussed existing cyber security standards, focusing specifically on control systems used in critical infrastructures. The standards help identify requirements for secure communication protocols and systems. Philips et al. [28] have conducted a broad investigation of the operational and security challenges of an advanced power grid involving Unified Power Flow Controller (UPFC) devices. Unlike SCADA systems, however, coordinated UPFC devices manipulate a smart power grid in a decentralized manner so that new security issues emerge. The authors discussed best practices, policies and risk assessment at the control level to achieve confidentiality, integrity, and availability in a cooperative UPFC power network.

The standards in place are insufficient to ensure the security of CPSs given the evolution of a wide range of threats exploiting the vulnerabilities of these systems. The Department of Homeland Security (DHS) conducted a broad investigation of different critical infrastructures to assess the current state of the art and identify challenges for cyber-physical systems security [29]. The report includes best practices, vulnerabilities and major factors affecting the system security of six critical infrastructures- electric energy, chemical, transportation, water, healthcare and commercial facilities. Two key challenges were identified to be essential for protecting cyber infrastructures - i) appropriate integration, protection, detection, and response mechanisms to construct CPSs that are resilient to both accidental failures, malicious attacks or manipulations, and surreptitious monitoring, and ii) verification and validation of interconnected and interacting control system components for the overall process by developing models, theories, and tools that account for a system's cyber and physical components in an integrated, unified way. "The unique security needs

of cyber-physical systems must be documented by formal requirements capture tools that can track both the discrete and continuous aspects of cyber-physical systems. In essence, what is not understood well cannot be built and verified correctly." A generalized theory is needed to address the security challenges emerging from the increasing complexity of CPSs.

2.2. FOCUS ON CYBER SECURITY FOR SECURE OPERATIONS OF THE PHYSICAL SYSTEM

Compromised cyber components can be used to send undesirable control command or data to the physical system, thereby compromising the desired operation of the physical system. The Stuxnet worm (W32.Stuxnet) [3] [4] serves as an extreme case of such attacks. The Stuxnet is a rootkit that divulges information and subverts industrial systems by targeting Siemens programmable logic controllers. The attack was undetectable since the worm fakes the control signals so that no false alarms are raised when a system is infected. Therefore, protection of the cyber domain and analysis of impact on the physical system due to compromised cyber component(s) are both necessary to secure operations of the physical system.

Holstein et al. [30] discussed SCADA cyber security to mitigate known vulnerabilities to attacks like the replay attack and the known-key attack etc. The goal was to protect communication packets, and to provide authorization and role-based access controls for interfacing control operations with energy management and distribution systems. They developed schemes for the protection of data based on the Advanced Encryption Standard (AES) to provide encryption and authentication, and cryptographic key management. While it is critical to adopt these practices, the possibility that information flows from the protected cyber domain to the observable physical network still remains.

McDonald et al. [31] have investigated control vulnerabilities associated with the energy sector. Their approach engages the Virtual Control System Environment (VCSE) that simulates and emulates the various elements of a control system dependent infrastructure like the cyber and physical components, human interfaces, etc. The VCSE tool set allows users to model any large infrastructure and assess its vulnerabilities. Different cyber attacks like the man-in-the-middle attack, rogue software attack, presence of unencrypted channels, etc. were simulated in this framework to validate the security of a given infrastructure. However, their approach requires a very detailed model of the infrastructure being analyzed.

Liu et al. [5] discuss the impact of *false data injection attacks* against state estimation in electric power grids. In power grids, state estimation is generally used to monitor the power system state by estimating unknown state variables based on the readings of meters placed at specific locations on the grid. The output of state estimation enables the system operators to identify potential operational problems in the grid and take decisions accordingly. During this process, bad data measurements are detected and removed to protect state estimation [32]. The authors show that it is possible to perform a false data injection attack in which the attacker injects malicious measurements that will bypass the existing bad data detection techniques and thereby, interferes with the state estimation process. It was assumed that the attacker can access the current power system configuration and manipulate the measurements of meters at physically protected locations such as substations. The attacker constructs an attack vector comprised of arbitrary measurements introduced at the meters he has access to. An efficient construction of such an attack vector was shown to be possible, given that the attacker has access to a matrix of measurements and state variables. It was shown that even if the attacker was constrained to specific meters or limited in the resources required to compromise the meters, construction of such attack vectors change the results of state estimation in arbitrary ways.

In a similar work by Sastry et al. [33], the attacker is assumed to obtain perturbed data (and not necessarily, the complete configuration as in [5]) to perform a deception attack on state estimator in SCADA for electric power systems. Control theoretic methods were applied to compute the attack vector for two widely used bad data detection schemes. The discussion on these bad data detection schemes is beyond the scope of this work. It was shown that it is possible to compute and stealthily inject false data into the SCADA system to alter the state estimation, considering the uncertainity associated with the partial or out-dated model available to the attacker.

2.3. FOCUS ON PHYSICAL SYSTEM SECURITY FOR SECURE CYBER OPERATIONS

A deviation from the expected behavior of the physical system indicates a malfunction or security violation at the physical system (considering an application dependent error). The expected behavior of the physical system is calculated based on real-time guarantees of the embedded systems within or with the control signal estimates. So, it is possible to detect specific cyber attacks that considerably change the expected behavior of the physical system.

In [34], Mueller et al. detect the execution of unauthorized instructions in CPS-based real-time embedded systems by utilizing information obtained from static timing analysis. Timing analysis in hard real-time systems is a strict requirement to verify that all tasks meet their deadlines, failing which the application is considered incorrect. During static timing analysis, the aggregated cost of instruction blocks and architectural timing effects are considered, to calculate the bounds on execution times. Unauthorized code potentially takes an unusual amount of time to execute. The assumption is that during an attack, hardware parameters like memory latencies and processor frequencies remain unmodified and the time bounds on code sections

are determined prior to the schedule. They propose Worst Case Execution Times (WCETs) for specific code sections that safely bound the upper execution time; execution times above these bounds provide indications of a system compromise. Based on the granularity of time bounds on the application code, three techniques of intrusion detection are designed; Timed Return Execution that is used to monitor the execution of selected code sections at application-level checkpoints, Timed Progress Tracking that makes synchronous calls to the operating system scheduler at security checkpoints and assesses time bounds for longer code sections and Timed Address Execution Tracking that utilizes the asynchronous scheduler calls to validate time bounds for approximated code sections. These techniques can be employed to detect intrusions based on code injection attacks on the cyber control layer of a CPS. While such detection is necessary, the ability of an attacker to change the physical system behavior by injecting false data was not addressed.

In [35], Sastry et al. proposed a method to detect the cyber attacks that change the behavior of the underlying physical system in a process control system. In such systems, the state of the physical system is monitored by a network of sensors placed at specific locations on the physical system. The physical system is modeled as a composition of control input sequences, output sequences and output estimates calculated from the sensor data. Their argument is that "if it is known how the output sequence, $y(k)$ of the physical system at a time instance, k, should react to the control input sequence, $u(k)$, then any attack to the sensor data can be potentially detected by comparing the expected output $\overline{y(k)}$ with the received (possibly compromised) signal $\tilde{y(k)}$." Depending on the quality of the estimate $\overline{y(k)}$, there could be false alarms indicating something wrong with the state of the system. It was assumed that since the signal sent by the sensors to the control center lies within specific bounds of measurement, at any specific time the signal coming from the attack sensors can be made to fall within the same range. These attack models

also assume that the attacker has a knowledge of: (1) the exact linear control model of the physical system, (2) the time to detect an attack and the probability of a false alarm, and (3) the control command signals (range of $y(k)$).

2.4. FOCUS ON CYBER-PHYSICAL INTERACTIONS FOR CPS SECURITY

Cyber-physical events often reveal information about the physical system through interactions that make the physical actions observable. Such information can be used by an intelligent attacker to violate the confidentiality of the CPS.

In ubiquitous computing systems that are deployed in residential environments, privacy leaks exist even when all the sensor transmissions are encrypted. Srinivasan et al. [36] propose fingerprint and timing based snooping attack model under which the attacker needs only the timestamp and the unique RF waveform pattern (fingerprint) of each radio message. The temporal distance of the transmission patterns of each pair of sensors is calculated to cluster the sensors with minimum distance between them. Such a calculation will group the sensors belonging to each room of the house into a cluster, whose activity can then be monitored. They also suggest that use of attenuators and introduction of random delays on transmissions can protect against these fingerprint and timing based snooping attacks. Inference attacks based on the physical observability of events like turning on/off of a light, along with the proposed snooping attacks can significantly compromise the privacy of the residents.

Security analysis of CPSs based on unified cyber and physical behaviors of the system was performed in Sastry et al. in [37]. The authors investigated the vulnerabilities of the SCADA for the Gignac water canal system, brought by compromising the sensors and actuators. The authors developed a partial differential equation (PDE) system representing the SCADA control and the water flow in this network of canal pools. This PDE system is extended to include the physical behavior of the

system under specific attack models. The primary attack model is comprised of an adversary sending incorrect data from the sensors to affect the intended objective of automatic control methods. It was observed that the possibility of an adversary to withdraw water from the canal stealthily exists. The stealthy deception attack involves modifying the water level sensors to send false measurements to the SCADA controller such that the SCADA system does not respond to counter adversarial action of withdrawing water from the off-take. This can happen in the system, if the injected false data is close to the current actual reading and so the deviation is negligible to the controller. The authors showed how the stealthy water withdrawal by the attacker through sensor deception can be included in the PDE system, as a combination of switching signals to discretely open/close the off-take gates. False data can be injected accordingly for upstream and downstream flows of water in the canal pools. Such an attack can be extended to multiple pools by approximating the effect of water withdrawal at upstream/downstream canal pools and manipulating the upstream/downstream sensors to send false data to the SCADA controller.

This dissertation addresses a similar problem with special focus on analyzing the confidentiality violation due to the nature of cyber-physical interactions within the system. It is possible to obfuscate the critical operations with respect to the observer by injecting events, called compensating events, that nullify the causal effect of the critical operations [23]. A taxonomy composed of the security properties of the sensor network, the threat model, and the security design space for SCADA systems is discussed in [38]. Control theoretic analysis coupled with information security for secure control of CPSs was investigated in [39]. In [40], a unified critical systems ontology was developed that aids in the assessment and modeling of reliability, safety, liveness, fault tolerance, security, and human aspects of CPSs. A summary of the above literature is presented in Table 2.1.

Table 2.1. Summary of existing work in CPS security

Study of	Approach	Comments
Standards for CPS security [26] [27] [28] [29]	Identifies best practices and guidelines that suit different CPS architectures	Broader scope for further research exists
Impact of cyber security on the secure operations of the physical system [30] [31] [5] [33]	Suggests cryptographic, access control and other cyber security mechanisms to prevent attacks on physical system	False data injection attacks can be performed on the physical system due to compromised cyber component(s)
Impact of physical system security on secure cyber operations [34] [35]	Cyber attacks are detected based on deviation from secure and expected behavior of physical system	Relies on the physical system guarantees
Impact of cyber-physical interactions on CPS security [36] [37]	Showed how attacks can be performed exploiting the nature of CPS interactions	Information flow models can provide a better reasoning

2.5. DIRECTION TAKEN IN THIS WORK

Information flow policies provide an appealing semantic basis in analyzing CPS interactions, since they define the way information moves throughout a system. Information flow security guarantees that no information flows from a high-level security domain to a low-level security domain. Modeling of the CPS is necessary in order to verify if a given CPS satisfies an information flow model. A discussion of information flow models that are attractive to CPSs, is presented in Section 4. However, two shortcomings exist:

1) representation of the concurrent and distributed interactions within a CPS while capturing the discrete and continuous aspects is complex,

2) a generalized approach to verify information flow security is lacking primarily because information flow within CPSs is less studied.

This work addresses those shortcomings by proposing a novel direction using process algebras to model and verify security properties within a CPS.

3. FORMAL SPECIFICATION AND VERIFICATION METHODS

Specification of a system is a formal description of its expected behavior. A system specification can be used to check whether the real implementation of the system is behaviorally equivalent to its specification. For this reason, the specification should embrace the features and logic of the system as closely as possible to the real system. The specification is then verified for a particular property or policy. This area of formal methods is very well studied over decades to verify software as well as hardware for correct design, reliability and functionality. The advantage of undertaking such mathematical and computational rigor lies in enhanced reliability and assurance. Modeling and verification of systems is an emerging research aspect in the CPS community to verify that a given a CPS satisfies functional correctness and meets the expected criteria. However, representation of the concurrent and distributed interactions within a CPS while capturing the discrete and continuous aspects is complex.

Different approaches to model and formally analyze CPSs exist. A common approach to model CPSs is a treatment similar to hybrid systems. The behavior of a hybrid system is characterized by the continuous dynamics of the physical system and the discrete nature of the embedded control. The hybrid system behavior can be defined in terms of hybrid automata [41]. Hybrid automata abstract the continuous evolution of the physical system defined in terms of ordinary differential equations, and the discrete transitions which characterize the change of automaton state. The continuous change is defined in terms of $flows$ that characterize the control mode represented as differential equations. The discrete changes are represented using $jumps$ that define the control switch leading to a value at the conclusion of a discrete change. The automata consists of states that abstract the flows and transition

on conditions defined by jumps. The composition of hybrid automata was defined as the time synchronization of two automata over common events. The hybrid automata can be model checked to determine if any of the reachable states violated the defined correctness or security property [42]. Work in these lines has been recently performed in [43] and [44]. However, modeling communication among various distributed components of a CPS is difficult using hybrid automata. Moreover, the complexity of verification on the enormous state space exist. Modeling of communication and concurrency among various cyber and physical processes also remains a challenge.

A unified approach to deal with CPSs is necessary that can encompass the non-deterministic and concurrent nature and develop uniform semantics of cyber and physical processes of a CPS. Hybrid systems do not model all aspects of CPSs since the distributed nature of CPSs is ignored. Process algebras, by contrast provide attractive framework to define the concurrency, communication and non-deterministic nature in systems. Platzer [45] advocated that logical analysis of CPSs combines the logical verification approach of hybrid systems and of distributed hybrid systems. To this end, the use of process algebraic theory coupled with hybrid systems theory provides a unified framework to model and verify distributed CPSs.

Traditional process algebras like the Calculus of Communicating Systems (CCS) [16] and Communicating Sequential Processes (CSP) [46] capture non-determinism, communication, recursion, process abstraction and process divergence, but their treatment is different. For example, CSP provides different operators for communication and interleaving of events while CCS offers a single operator to perform both. However, communication happens in CCS semantics when the interleaved events are complementary, leading to an internal event. Such semantic differences coupled with different notions of equivalences that process algebras offer, help understand the behavior of systems.

Unfortunately, in some CPSs, process interactions could be dynamic meaning that processes might establish connections arbitrarily with other processes; thus the structure of communication is dynamic. In Section 3.1, a process algebra based on CCS, called the SPA is presented. SPA is useful to specify the behavior of distributed systems and to verify the security properties within. Due to the limitations discussed later, SPA cannot model all aspects of CPS interactions and hence, the π-calculus introduced in Section 3.2 is adopted.

3.1. SECURITY PROCESS ALGEBRA (SPA)

SPA [47] [48] is an extension of CCS - a language for specifying concurrent systems. It provides an algebra for defining larger systems from smaller subsystems in a bottom-up fashion. The basic building blocks are atomic activities called actions. Unlike in CCS, SPA actions belong to two different levels of confidentiality, permitting the specification of multilevel (actually, two-level) systems. The syntax for describing a system using SPA is:

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

In the above syntax, 0 is the empty process that cannot perform any action (specifically defines termination of a process); $\mu.E$ performs action μ and then behaves like E; $E_1 + E_2$ can alternatively choose to behave like E_1 or E_2; $E_1|E_2$ is the parallel composition of E_1 and E_2, where the executions of the two systems are interleaved; $E\backslash L$ executes all the actions that can be performed by E, provided that they do not belong to $L \cup \bar{L}$ (where $\bar{L}$ refers to the output); $E\backslash_I L$ requires that the actions of E do not belong to $L \cap I$; E/L transforms all the actions in L into internal actions; if E can execute action μ, then $E[f]$ performs $f(\mu)$; and finally, Z performs the actions that E performs if $Z \equiv E$. Following the customary notation, $\tau \in Tr$ is a system trace, $\tau\backslash_x$ 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 events E_1 and E_2. Additionally, $High$ and Low are used to represent H and L security domains containing H and L users, respectively. Also, the symbols I and O represent inputs and outputs, respectively. The operational semantics of the key operators in SPA are presented in Figure 3.1.

Prefix	$\frac{-}{a.E \xrightarrow{a} E}$	**Communication**	$\frac{E_1 \xrightarrow{a} E_1' \quad E_2 \xrightarrow{\overline{a}} E_2'}{E_1 \mid E_2 \xrightarrow{\tau} E_1' \mid E_2'}$
Parallel	$\frac{E_1 \xrightarrow{a} E_1'}{E_1 \mid E_2 \xrightarrow{a} E_1'}$	**Sum**	$\frac{E_1 \xrightarrow{a} E_1'}{E_1 + E_2 \xrightarrow{a} E_1'}$
	$\frac{E_2 \xrightarrow{a} E_2'}{E_1 \mid E_2 \xrightarrow{a} E_2'}$		$\frac{E_2 \xrightarrow{a} E_2'}{E_1 + E_2 \xrightarrow{a} E_2'}$
Restriction	$\frac{E \xrightarrow{a} E'}{E \backslash L \xrightarrow{a} E' \backslash L}$ if $a \notin L \cup \overline{L}$		

Figure 3.1. Operational semantics in SPA

SPA is used in the analysis presented in Section 6 to demonstrate the capability of process algebras in modeling CPSs. However, modeling the interaction among symmetric cyber processes that exhibit different behaviors was difficult. Additionally, using SPA, it was difficult to capture the dynamism involved in interactions between specific cyber processes that communicate to perform specific actions on the physical system. An advanced mechanism is required that will distinguish messages from symmetric processes based on the channels of communication. The case of multiple observers interacting to downgrade information from the system can be analyzed with this new mechanism. Hence the development of a π-calculus based information flow theory will be the next step.

3.2. THE π-CALCULUS

The π-calculus [49] is an enhanced version of CCS [16], that offers attractive formalism of processes establishing communication with other processes over dynamic channels. Such a dynamic communication can be represented using semantic operators like restriction (of a message to a channel), and scope extrusion through which the dynamically changing scope of the communication domain is abstracted through sending of channel names as messages to processes that did not have access to those channels before. Among the other novel features of π-calculus, notable are the ability to model communication among processes over dynamic links, and the notion of equality testing based on bisimulation. In π-calculus, an infinite set of *names* (m, n, p, q, r, etc.) are used for communication channels, and an infinite set of *variables* (x, y, z, etc.) are used to define the terms. The set of processes is defined by the grammar shown in Figure 3.2. The null process **0** does nothing. A composition P|Q behaves as processes P and Q running in parallel. Processes operate on channels to communicate with each other and with the outside world. The basic *interaction* is defined using $\bar{x} < N >$.P that defines an output process that is ready to output on channel x, or $x(m)$.P that defines an input process that is ready to receive a value over channel x. The replication !P behaves as an infinite number of copies of P running in parallel. The name restriction operator (νn.P) is a process that makes a new, private name n, and then behaves as P.

0	Null Process
P|Q	Parallel Composition
!P	Replication
νn.P	Name Restriction
$x(m)$.P	Message Input
$\bar{x} < N >$.P	Message Output

Figure 3.2. π-calculus syntax

A novel contribution of process algebraic theory is the ability to verify the equivalence of two systems. Different process algebras offer different notions of equivalence based on their structural operational semantics. Two processes are bisimulation equivalent if there exists a bisimulation relation including both the processes as a pair (formally introduced in Section 4.5). Verification of such equivalence among systems was studied independently by Paige et al. [50] and Kanellakis et al. [51]. However, to verify the equivalence of systems based on security, specification semantics to represent private communication among processes are required.

An extension of the π-calculus, the spi calculus, was introduced by Abadi and Gordon to specify and analyze cryptographic protocols [52]. Spi calculus extends the regular implication of restriction and scope extrusion using channels in the π-calculus, by providing operators using which messages transmitted over the channel can be encrypted/decrypted. The receiver of the encrypted message on the channel can perform a function on the received message can be decrypted using the shared key. Spi calculus can be used to analyze a variety of cryptographic protocols by defining the cryptographic operations using the channels in systems for distributed security. In spite of all the features the process algebras (CCS, CSP, π-calculus) and their variants offer, they cannot still be directly applied to CPS environments because of the inability to capture the continuous dynamics of a CPS.

The ϕ-calculus proposed by Rounds and Song [53] is an extension of the π-calculus to hybrid systems. A hybrid system can be defined using the ϕ-calculus as the communication among concurrent hybrid automata defining the CPS environment. In [54], Jifeng proposed a formal description language for hybrid systems based on CSP by developing trace semantics for the differential equations guiding the continuous system. A theory that bridges the process algebraic approach and hybrid systems theory for the purpose of security verification in CPSs is missing and

this work is an attempt in that direction. Recently, Wolfthusen et al. [55], models process control systems with adversarial behavior characterized as π-processes.

The CPS specification should entail the cyber and physical components and the interactions among them. Most importantly, the continuous nature (like flow invariance) of the physical network should be discretized to be represented as an interacting process with the cyber processes. Information flow models that are of interest to CPS security and their applicability within the process algebraic framework are discussed in Section 4.

4. INFORMATION FLOW MODELS

Information flow security guarantees that no information flows from a high-level security domain to a low-level security domain within a system. Information flow has been studied primarily for program and language security [56]. Various security models that analyze system security from the perspective of access control or execution sequence have been discussed for decades [13] [15]. However, this work has not been directly applied to CPSs. In a CPS, information flows between and within the cyber and physical components. Unfortunately, access control policies like the Bell-La Padula model [57] do not prevent information propagation because they do not control how information will be used. The possibility that confidential information may be inferred from the observable information flow represents a potential source of critical information leakage. Information flow models aid in understanding how information is leaked across the cyber-physical boundary and how this information may be downgraded by an observer within the system.

In the following sections, $\tau \in Tr$ is a system trace, $\tau\backslash_x$ is a trace purged of all events in the domain of x, $\tau\ |_x$ is a trace restricted to events in the domain of x, $E_1|E_2$ is the parallel composition of events E_1 and E_2, H and L are high-level and low-level security domains, respectively, and I and O are inputs and outputs. A legal or valid trace of a system is defined as an event trace, the order of which is such that the output events are always preceded by their corresponding input events. A H (L) event is one that occurs within the $H(L)$ domain. A L observation is a special case of a L event.

4.1. NON-INTERFERENCE

Non-interference [7] seems to be the natural way of defining the information flow behavior which states that the high level interactions does not interfere with the low level observability. In other words, information does not flow from H domain to L domain if the H behavior has no effect on what L observer can observe.

Let S be a set of subjects, Σ be a set of system states, O: a set of outputs, Z: a set of commands and $C = S \times Z$: a set of subject, command pairs indicating which subject executed which command. A state transition function $T : C \times \Sigma \rightarrow \Sigma$ specifies to which state the system transitions to when some command c is executed in a state σ, and an output function $P : C \times \Sigma \rightarrow O$ describes the output of executing c in state σ. Two functions, *proj* (projecton) and π (purge) are defined to formalize non-interference. The projection function is used to see the outputs that a subject can see for a given state of a system, e.g. $proj(S, Cs, \sigma_i)$ would result in the outputs from the command sequence Cs that subject S can see given state σ_i. The purge function has three forms:

- $\pi_G(Cs)$: This function will remove all elements in the command sequence Cs which involve subject S, for $S \in G$ where G is some subset of subjects.
- $\pi_A(Cs)$: This function will remove all elements in the command sequence Cs which involve action Z, for $Z \in A$ where A is some subset of actions.
- $\pi_{G,A}(Cs)$: This is a combination of π_G and π_A where the conditions of both functions must hold for an element to be removed.

The G users are non-interfering with G' users if for all valid command sequences, $proj(S, Cs, \sigma_i) = proj(S, \pi_{G,A}(Cs), \sigma_i)\ for\ S \in G'$. For a simple system with two groups, a H domain and an L domain,

$$NInt(ES) \equiv \forall \tau \in Tr, \tau \backslash_H = \tau \mid_L \tag{1}$$

This can be interpreted as follows: for all possible command sequences, all L projections are the same regardless if the H actions are purged out of the command sequence.

4.2. NON-INFERENCE

Conceptually, in a system that is non-inference secure, any observation is consistent either with H and L events, or with L events only. Thus, an observer cannot infer that any information is flowing from an H domain to an L domain. A system is considered secure if and only if for any legal trace of system events, the trace that results from the legal trace purged of all H events is still a legal trace of the system [13]. Formally, a system ES is said to be non-inference secure if

$$NInf(ES) \equiv \forall \tau \in Tr, (\tau \mid_L \in Tr) \mid_H = \phi \tag{2}$$

This imposes two conditions on the system. First, for any system trace, the restriction of the trace to L events alone is also a valid system trace. That is, just by observing L events, one cannot infer with certainty if a H event occurred. Second, purging H events from any system trace should yield a valid trace comprising of the L events.

4.3. NON-DEDUCIBILITY

Non-deducibility ensures that the low-level observability does not deduce the specific high-level inputs to the system [58]. Formally, a system ES is said to be non-deducible secure if

$$ND(ES) \equiv \forall \tau_L, \tau_H \in Tr : \exists \tau \in Tr : \tau \mid_L = \tau_L \wedge \tau \mid_{H \cap I} = \tau_H. \tag{3}$$

A system is considered non-deducible secure if nothing can be deduced about the sequence of input events, I, in the H domain based only on observation of events in the L domain. In other words, a system is non-deducible secure if a L observation is compatible with any H input event [14].

4.4. APPLICABILITY OF TRACE-BASED MODELS TO A CPS

Non-interference, non-inference and non-deducibility are trace based models in which one verifies the trace equivalence of system's execution with respect to a L observer on the same finite sequences. As the complexity of CPS increases, the possible sequence of events that needs to be verified for trace equivalence grows, increasing the complexity of confidentiality verification dramatically.

For CPS models, non-interference could be very restrictive since CPSs inherently involve the cyber or physical subjects in the H domain writing to the physical subjects in the H or L domain. For example, a cyber component issuing a $High$-level command to a physical component may result in an observable L event at the neighboring physical component due to physical flow invariant as shown in Figure 4.1. In this case, non-interference is violated.

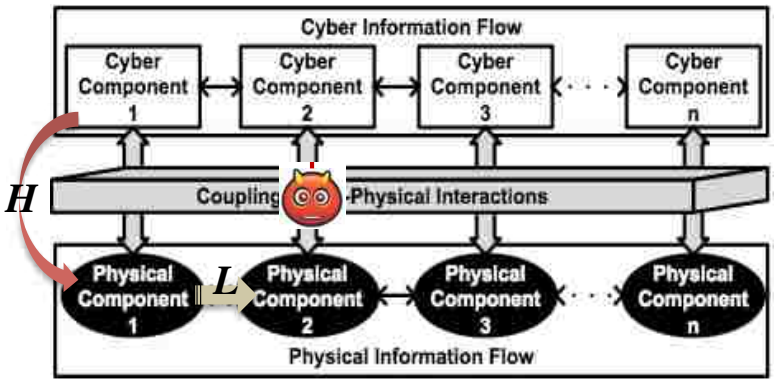

Figure 4.1. Non-interference for CPSs

Non-inference may be too strong in some cases, such as when a system cannot operate without a H event or where the L inputs result in H outputs. However, for the case shown in Figure 4.2, if the L events always occur even in the absence of the H event, then non-inference is satisfied.

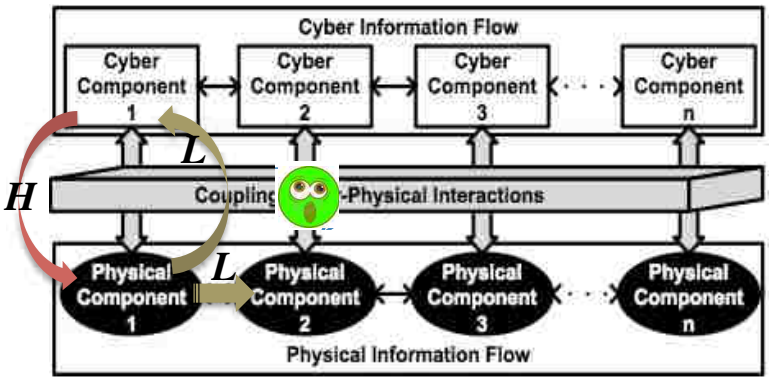

Figure 4.2. Non-inference for CPSs

Non-deducibility security models are used to analyze a system in which H outputs are observable. According to McLean [8], if an entire system is non-deducibile secure, then no L observation of the system will ever acquire any H information through the system. In Figure 4.3, due to the occurrence of two H input events simultaneously, an observer cannot deduce any information regarding the high level inputs from his low level observation.

While non-inference and non-deducibility seem to be attractive models to analyze the information flow within a CPS, their limitation to trace based systems can be overcome by applying them to different behaviors of a CPS. Verifying that a CPS preserves a specific information flow property, requires a formal description of the system.

To perform security analysis of a CPS, a specification system similar to SPA is necessary to specify the CPS while analyzing the interactions between different

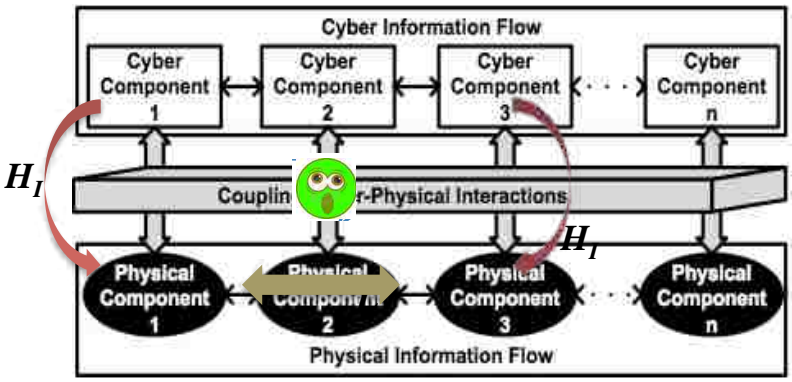

Figure 4.3. Non-deducibility for CPSs

security domains within, to determine potential violation of information flow properties. Different information flow properties were discussed in literature, that define how information may not flow between different security domains to preserve confidentiality. In SPA (CCS) and π-calculus, however, these properties are defined by exploiting the behavioral equivalence between processes as observed by the low-level observer. The bisimulation-based non-deducibility on composition (BNDC) property defined in Section 4.5 ensures that every state (characterized by different system behaviors) reachable by the system satisfies the basic non-deducibility property.

4.5. BISIMULATION-BASED NON-DEDUCIBILITY ON COMPOSITION

Checking confidentiality in a CPS requires an exhaustive verification of all possible system behaviors to detect the interactions that do not satisfy the desired security properties. A system is considered to satisfy bisimulation-based non-deducibility on composition (BNDC) if it can preserve its security after composition with other processes [47] [48] [59]. BNDC property uses weak bisimulation equivalence defined in Definition 1, to detect the behaviors that are not non-deducible secure on composition with a high level process.

Definition 1 (Weak Bisimulation)**.** *A binary relation $\mathcal{R} \subseteq \mathcal{E}X\mathcal{E}$ is a weak bisimulation if (P,Q) $\in \mathcal{R}$ implies, for all a $\in$ Act,*

- *if $P \xrightarrow{a} P'$, then there exists Q' such that $Q \xrightarrow{a} Q'$ and (P', Q') $\in \mathcal{R}$*
- *if $Q \xrightarrow{a} Q'$, then there exists P' such that $P \xrightarrow{a} P'$ and (P', Q') $\in \mathcal{R}$*

Two processes P, Q $\in \mathcal{E}$ are weakly bisimilar, denoted by $P \approx Q$, if there exists a weak bisimulation relation $\mathcal{R}$ containing the pair (P, Q).

A system ES is BNDC if, for every H process, Π, a L observation cannot distinguish ES from the process ES composed with any other process, Π. In other words, a system ES is BNDC if a L observation is not modified by composing any H process, Π with ES. Formally,

$$BNDC(ES) \equiv \forall \Pi \in E_H, ES\backslash H \approx_B (ES|\Pi)\backslash H \tag{4}$$

where $ES\backslash H$ changes all the H events in ES into internal silent actions and $\approx_B$ is a weak bisimulation relation. A system is BNDC-preserving if the above property holds for all possible system behaviors. A broad study of BNDC and its variants is presented in [60]. Verification of BNDC in a test CPS is presented in Section 6.

4.6. GENERAL APPROACH TO VERIFY INFORMATION FLOW PROPERTIES IN A CPS

A formal methodology to automate the process of verifying confidentiality of information flow within a CPS involves addressing three main issues below:

4.6.1. Representation of Cyber and Physical Processes and Their Interactions in a Computational Framework. A process algebra can be used to model the CPS as a composition of cyber and physical processes that communicate concurrently, if possible, in a synchronized manner. Each process is defined as

a sequence of events within the system which determine different states the system could transition into. In [22], SPA has been used to represent the physical actions interacting with the computational elements in a gas pipeline network. A similar approach has been taken in [17], to analyze information flow within a more complex smart grid that uses advanced distributed algorithms to manage the underlying physical resources like renewable energy sources, house loads, power storage, etc.

The physical network in a CPS forms a continuous time subsystem and the computational part forms a discrete event subsystem. Process algebras are not equipped to model the continuous-time nature of physical processes in a CPS. To capture the information flow of the combined system thereby forces the modeling of the continuous physical subsystem to be event-based, so that the physical events can be captured using process algebra. Physical events include 1) a local state change of the physical subsystem resulting from a cyber component controlling it (for example, a power flow controller increases/decreases voltage on the power line causing a flow change) or 2) a local physical state change brought by the dynamics of the physical network (for example, load on a power line increases/decreases as a stochastic process to which the power electronics react by making a setting). Invariance on physical flow can be modeled such that events that change the flow at various physical components are reflected in an aggregate flow that satisfies the invariant. The impact of physically observable behavior cannot be ignored to study information flow in CPSs. This forces the observable actions to be considered as events that are used as building blocks of process specification.

Cyber events within a CPS involve in distributed computation based on 1) communication with other cyber components or 2) communication with the physical component that it controls. Composition of cyber processes result in the transformation of complementary actions of the processes into internal silent actions in the composed process, defined by the SPA **Communication** operator in Figure 3.1.

The communication between physical processes is different from that between cyber processes; in the former case, the pair of processes make a synchronized physical change on the shared network (such as power transfer on the shared power bus from a component with high potential to a component with lower potential) and in the latter case, the pair of processes synchronize on complementary request/response type messages. Methods to counter interception of messages on the communication channel exist, validating the assumption that cyber processes can be securely composable as long as they perform complementary actions. By contrast, physical change between two physical processes is observable by an intermediary in the path of their direct communication, making it difficult to securely model such composition. Using the proposed approach, CPSs can be modeled in terms of SPA followed by bisimulation-based equivalence testing of the processes as outlined in Section 4.6.2.

4.6.2. Adequacy of Bisimulation-Based Non-deducibility Properties for CPS Models. According to the definition of BNDC (in Equation 4), the SPA specification of the system has to be composed with all the high-level processes (Π) that can be modeled using the high-level actions of the system. However, this would significantly increase the complexity of verification of BNDC property since it should be verified that $E\backslash H \approx_B (E|\Pi)\backslash H$, for all Π (Π could be defined as some combination of events from $\{H \cup \bar{H}\}$ reaching possibly any state $E' \in \xi$, where ξ is the set of all the states of the system). To avoid such universal quantification on Π, a strong form of BNDC called strong BNDC (or SBNDC) has been proposed in [60]. SBNDC states that iff for all E' reachable from E, if $E' \xrightarrow{h \in H} E''$, then $E'\backslash H \approx_B E''\backslash H$. This definition of SBNDC implies that the system before and after executing a high-level action remains indistinguishable.

Such a definition avoids the fact that the property should hold for all possible high-level processes within the system, transforming the bisimulation relation between E and $E|\Pi$ into a *bisimulation up to* H relation on E and $E\backslash H$ i.e., it will

be verified if $E \approx_{\backslash H} E\backslash H$ with the high actions replaced with silent internal action, τ. The *bisimulation up to H* relation for SBNDC ($\approx^0_{\backslash H}$) transforms the high-level events in E' into a sequence of $(\xrightarrow{\tau})^0$ or zero actions. An exhaustive study of BNDC and its variants is presented in [60]. The impact of silent internal actions on weak bisimulation relation is equivalent to that defined in CCS [16].

4.6.3. Testing for Bisimulation Equivalence of Processes. Two processes are bisimulation equivalent if there exists a bisimulation relation including both the processes as a pair. This problem of equivalence testing has been well studied in literature [50] [51] as a relational coarsest partition (RCP) problem: given a relation, R (in this case, *bisimulation up to H*) and an initial state, E over a global set of states ξ, find the coarsest stable refinement such that either $E \subseteq R^{-1}(E\backslash H)$ or $E \cap R^{-1}(E\backslash H) = \phi$; if such a stable partition cannot be found, then such a partition does not exist implying that *bisimulation up to H* relation does not exist between E and $E\backslash H$.

The remainder of this work demonstrates the specification and information flow verification challenges within CPS infrastructures by applying them to an example CPS (of Section 5) in Sections 6 and 7.

5. FREEDM: A TEST CPS

The Free Renewable Electric Energy Delivery and Management (FREEDM) [61] microgrid is a smart power grid architecture with advanced technologies of the Solid State Transformer (SST), Distributed Renewable Energy Resource (DRER), Distributed Energy Storage Device (DESD), and House Load (LOAD) powered with Distributed Grid Intelligence (DGI) to meet the goals of optimal energy management and reliability enhancement. FREEDM is a perfect example of a CPS, since it includes distributed physical and cyber components that communicate among themselves to control the system. The SST and power electronics that embed the DGI are referred to as an Intelligent Energy Management (IEM) (See Figure 5.1).

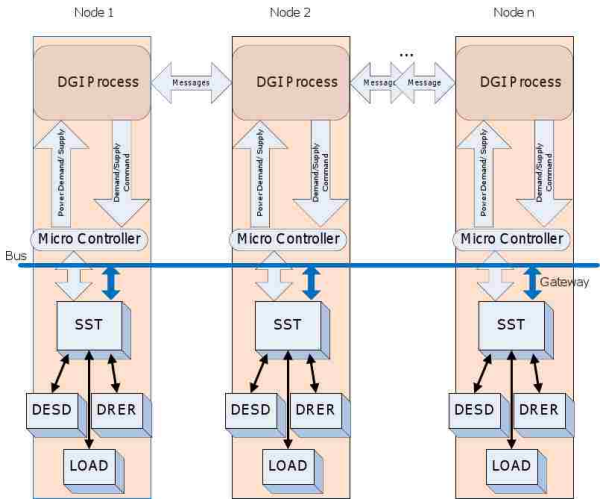

Figure 5.1. FREEDM microgrid with three nodes

Among various algorithms adopted by the DGI is a *Power Management* scheme [20], to efficiently balance power flow for optimal distribution of energy within the system.

5.1. DISTRIBUTED POWER MANAGEMENT SCHEME

The power management algorithm in the FREEDM system draws its origins from distributed load balancing schemes [62] in computer science, that are designed to normalize the loads of process execution among the peers of a distributed system. Intuitively, the nodes participating in a load balancing algorithm communicate their load changes with each other in an attempt to migrate the process execution task from a node with *Demand* to a node with *Supply*. The result of such a migration is that the nodes normalize their loads, thereby achieving a roughly balanced load computation. Every IEM computes the SST's actual load on the distribution grid and decides the state of a node as having *Supply* or *Demand* or *Normal* state of load. The algorithm consists of concurrent sub-processes with message passing communication among the IEMs on critical load changes. Each DGI maintains a (potentially out-of-date) *Load table* as shown in Table 5.1, to store information it receives about other nodes in the system. Load table updating strategies are adopted

Table 5.1. Load table maintained at each node

Node	**State**	**Node**	**State**	. .	**Node**	**State**
1	Supply	1	Supply	. .	1	Normal
2	Demand	2	Demand	. .	2	Demand
.	.	.	.	. .	.	.
.	.	.	.	. .	.	.
n	Supply	n	Supply	. .	n	Normal
At IEM 1		At IEM 2		. .	At IEM n	

to minimize cyber message traffic during frequent load changes. An IEM node, on entering into a *Supply* state, advertises a *Draft Request* message to the nodes in its load table that are in *Demand* state and waits for response. A *Demand* node, on

receiving a *Draft Request* message, responds to the sender by sending its demand cost with a special message called *Draft Age*. The *Supply* node, on receiving Draft ages from different *Demand* nodes, will compute a *Draft Standard* which is an optimized selection of the node it is going to supply power to by evaluation of factors like its own predicted need, economics and other optimization metrics. The *Supply* node, on computation of draft standard, sends a unique *Draft Select* message and initiates the power migration by making a set point on the *Gateway power* which is the local SST's individual contribution on to the shared power bus. On receiving the *Draft Select* message from the *Supply* node, the IEM which was in demand receives this power from the shared bus. The migration takes place in unit step size until the time the *Supply* node can supply to the *Demand* node or the *Demand* node meets its sufficient demand, or there is a change of load state in either of the nodes. The algorithm continues until all the nodes are in *Normal* state. A sample DGI trace involving a drafting node (which can *Supply*) and the source (which is in *Demand*) is shown below:

```
                                        DGI_Draft: Request bid from known loaded DGIs

DGI_Source: Respond to bid request
            if loaded                   DGI_Draft: Order the response messages arbitrarily

                                        DGI_Draft: Selects power to migrate based on cost
DGI_Source: Responds to select message
            and commands local SST
                                        DGI_Draft: Sends select message and commands
                                                   local SST
```

5.2. NEED FOR INFORMATION FLOW ANALYSIS OF FREEDM

The National Institute of Standards and Technology (NIST) identifies that confidentiality is an increasing concern in smart grids to protect i) the privacy of the consumer, ii) the electric market information and iii) the power company [25]. Reasons that pose additional risk include increased entry points for potential adversaries with increased complexity of the grid, and the impact of coordinated cyber-physical

attacks. Unrestricted information flow can potentially be used against the system for economic gains; for example, generators of renewable energy may withhold power to sell at a premium. The physical components that are exposed to any observer outside the CPS divulge some information regarding the system (physically observable behavior). For example, the operation of a wind turbine in a smart grid depends on its physical size, velocity of wind, etc., that are observable. Definitions of H and L domains change according to the physical location of the observer on the CPS; for example, an observer controlling a physical component knows more about the system than an external observer (with only physical observability). If a user has access to the *Load table* of his DGI, then he knows the demand states of his peers. Such a user can obtain critical information pertaining to the rest of the system by observing the cyber-physical interactions or faking his demand state, thereby violating confidentiality. To completely analyze the information flow, various cases of such observers should be accounted for, that reveal the extent of confidentiality violation within the system [17] as in Figure 5.2. Such models of information flow are discussed in Section 6.2.

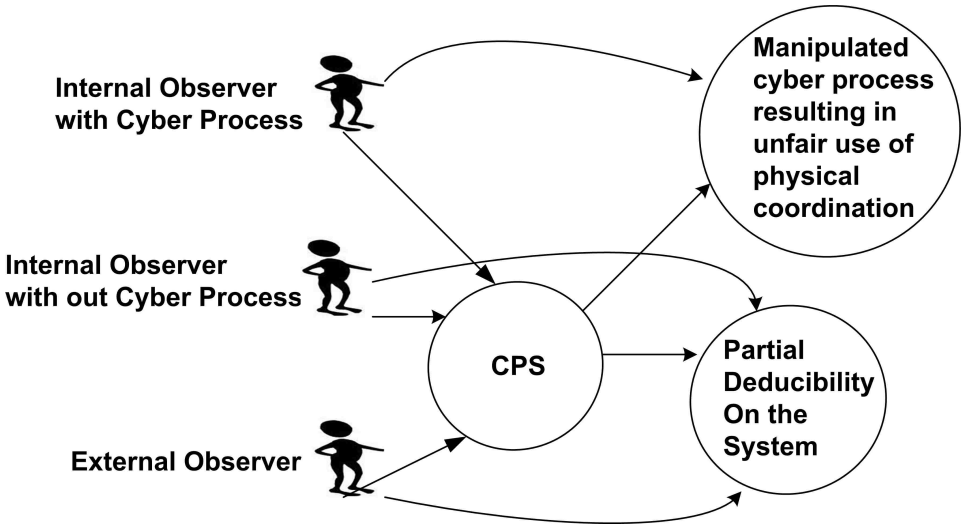

Figure 5.2. Different levels of confidentiality violation possible in a CPS

6. INFORMATION FLOW ANALYSIS USING SPA

A subnetwork of the FREEDM system with three nodes is depicted in Figure 6.1. The events in the system are $DRER$, $DESD$, $Load$, Bus, X_{SST}, $Gateway$ and $Utility$ which are the actions associated with DRER, state of DESD, house load, the total power on the shared power bus, strategy of the SST for local management at the node level and activity on utility grid respectively. For notational convenience, the events are italicized to distinguish them from their respective physical components (for example, DRER is the physical component while $DRER$ is a corresponding event). Event classification into H and L security domains differ in different scenarios.

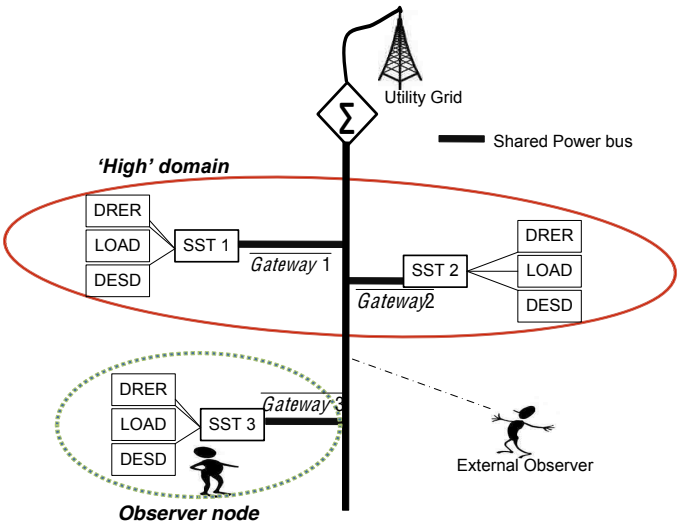

Figure 6.1. FREEDM subsystem with no DGI, two nodes and two observers

Lemma 1. *Power flow in the shared power bus is an invariant function of individual gateway loads of the participating nodes and the draw from, or contribution to, the utility grid.*

Proof. Assuming the utility grid to be an infinite source and sink of power, the power flow in the shared power bus of local grid can be expressed by Equation 5.

$$P_{Bus} = \sum_{i=1}^{n} P_{Gateway} + P_{Utility} \tag{5}$$

where n is the number of nodes and $P_{Utility}$ is the total power draw from or contribution to the utility grid. This is obvious since the flow in the subnetwork is preserved due to Kirchoff's current laws. The net demand or supply on the bus is compensated as a net draw from or contribution to the utility grid, respectively. □

6.1. MODELING OF FREEDM USING SPA

Each node without the DGI process is modeled as in Equation 6. The invariant on the bus shown in Equation 5 can be modeled as in Equation 7. The microgrid consisting of n such nodes can be modeled as in Equation 8.

$$\begin{aligned} Node_{noDGI} \cong (&DRER.\overline{DRER}.Node_{noDGI} + DESD.Node_{noDGI} \\ &+ Load.Node_{noDGI}).X_{SST}.(\overline{DESD}.Node_{noDGI} \\ &+ \overline{Load}.Node_{noDGI}).Gateway.Node_{noDGI} \end{aligned} \tag{6}$$

$$\begin{aligned} Bus \cong (&\overline{Gateway}_{Node\ 1}.Bus + \overline{Gateway}_{Node\ 2}.Bus + ... \\ &\overline{Gateway}_{Node\ n}.Bus) + Utility.Bus \end{aligned} \tag{7}$$

$$E \cong (Node\ 1_{noDGI}|Node\ 2_{noDGI}|...|Node\ n_{noDGI})|Bus \tag{8}$$

6.2. VERIFICATION OF INFORMATION FLOW USING SPA

Information flows in different ways depending on the observer's level of interaction with the system as shown below.

6.2.1. External Observer on the Physical System. The external observer can know visible information about the DRER like the size of the facility,

weather factors impacting the DRER output (represented by $DRER$), but not the output energy generated at any given instance of time ($\overline{DRER}$). As in Figure 6.1, the external observer could use an inductive pickup to obtain the reading on the shared power bus or the gateway at each node, since the power lines are physically visible and open. The following conclusions can be made on the information flow in the case of such an observer.

Lemma 2. *A single node in the system without DGI is BNDC-secure with respect to a low-level external observer with limited physical observability.*

Proof. Assuming that the low-level observer can only observe the visible DRER sources, the classification of events at any node as defined in Equation 6 is $Low = \{DRER\}$, $High = \{\overline{DRER}, DESD, Load, X_{SST}, \overline{DESD}, \overline{Load}, Gateway\}$. Restricting all the high level events within the node yields, $Node_{noDGI}\backslash H \cong DRER.Node_{noDGI}$. For any high level process Π, say Bus, the restriction on the composed system, $(Node_{noDGI}|\Pi)\backslash H \cong DRER.Node_{noDGI}$. Therefore, $Node_{noDGI}\backslash H \approx_B (Node_{noDGI}|\Pi)\backslash H$. □

Lemma 3. *A single node in the system without DGI is NOT BNDC-secure with respect to a low-level external observer that can read the gateway at the node.*

Proof. Assuming that the low-level observer can observe the visible DRER sources as well as the Gateway, the classification of events at any node as defined in Equation 6 is $Low = \{DRER, \overline{Gateway}\}$, $High = \{\overline{DRER}, DESD, Load, X_{SST}, \overline{DESD}, \overline{Load}\}$. Restricting all the high level events within the node yields, $Node_{noDGI}\backslash H \cong \{DRER\}$. For any high level process, $\Pi \equiv Bus$, the restriction of the composed system, $(Node_{noDGI}|\Pi)\backslash H \cong \{DRER, \overline{Gateway}\}$. Therefore, $Node_{noDGI}\backslash H \approx_B (Node_{noDGI}|\Pi)\backslash H$. □

An interesting observation in addition to Lemma 3 is that the process satisfies non-deducibility property. The observer might see a different output of gateway

($\overline{Gateway}$) every time a high level process, $\Pi \equiv Bus$ takes place within the system. However, the observer cannot deduce anything about the high level inputs to the system since a gateway change might be because of any of the high level inputs $\{\overline{DRER}, DESD, Load\}$ or a combination of them.

Theorem 1. *The physical system in FREEDM is BNDC-secure with respect to a low-level external observer as shown in Figure 6.1.*

Proof. From Lemmas 2 and 3, it follows that low-level observation on DRER is BNDC-secure while an additional observation on gateway at an individual node is not. However, when composed with the bus as in Equation 8, the system satisfies the BNDC property. Assuming that the low-level observer can observe the visible DRER sources as well as the Bus, the classification of events within the system as defined in Equation 8 is $Low = \{DRER^n_{i=1}, Bus\}$, $High = \{Node\ 1_{noDGI}, Node\ 2_{noDGI}\ ...\ Node\ n_{noDGI}, Utility\}$. Restricting all the high level events within the system yields, $E\backslash H \cong \{DRER^n_{i=1}, Bus\}$. For any high level process Π, say, $\overline{Gateway}_1.\overline{Gateway}_2$, the high-level restriction on the composed system, $(Node_{noDGI}|\Pi)\backslash H \cong \{DRER^n_{i=1}, Bus'\}$. Due to Lemma 1, observation of Bus is always consistent since $\sum_{i=1}^{n} Gateway + Utility = \sum_{i=1}^{n} Gateway' + Utility'$. Therefore, $E\backslash H \approx_B (E|\Pi)\backslash H$. □

Given that the observer can observe all the gateway loads, the observer can match every unique $\overline{Gateway}$ event with a corresponding Bus event, thereby divulging the confidentiality of the system. In that case, restricting all the high level events within the system yields, $E\backslash H \cong \{DRER^n_{i=1}, \overline{Gateway}^n_{i=1}, Bus\}$. For $\Pi \equiv Utility$ implying the case when a node draws from or sheds excess power to the utility, $(Node_{noDGI}|\Pi)\backslash H \cong DRER^n_{i=1}, \overline{Gateway}'_i, Bus'\}$ where Bus' is inconsistent with the event, Bus. In that case, the system is not BNDC-secure.

6.2.2. Internal Observer on the Physical System. If the nodes are not involved in the DGI power balancing process, the low-level internal observer as

shown in Figure 6.1, who is a part of the physical grid, can observe a change on the shared power bus, whenever a *Supply* node renders its excess generation to the utility grid or a *Demand* node 'absorbs' power from the utility grid. However, the observer cannot exactly know who performed the change. Therefore, it can be said that the system without the DGI process is non-deducible secure. This leads to the following Lemma 4.

Lemma 4. *The system without the DGI process is non-deducible secure.*

Proof. The change observed by the low-level observer on the shared power bus, Bus' could be due to any of the other nodes that are in *Demand* or *Supply*. The observer would be in doubt as to who performed the event or if more than one of the nodes performed it. The observer could not deduce the high level inputs to the system from the low level observation. This makes the system non-deducible secure. □

Theorem 2. *The physical system in FREEDM is BNDC-secure with respect to a low-level internal observer as shown in Figure 6.1.*

Proof. Assuming that the low-level internal observer, *IO* can observe the visible DRER sources as well as the *Bus*, the classification of events within the system as defined in Equation 8 is $Low = \{DRER^{n}_{i=1}, Node\ IO_{noDGI}, Bus, \overline{Gateway}_{Node\ IO}\}$, $High = \{Node\ 1_{noDGI}, Node\ 2_{noDGI}\ ...\ Node\ n_{noDGI}\}$. Restricting all the high level events within the system yields, $E\backslash H \cong \{DRER^{n}_{i=1}, Node\ IO_{noDGI}, Bus, \overline{Gateway}_{Node\ IO}\}$. For any high level process Π, say, $\overline{Gateway}_i.\overline{Gateway}_j$ where $i, j \neq IO$ the high-level restriction on the composed system, $E|\Pi)\backslash H \cong \{DRER^{n}_{i=1}, Node\ IO_{noDGI}, Bus, \overline{Gateway}_{Node\ IO}\}$. As with the case with external observer in Theorem 1, following the Lemma 1, observation of *Bus* is always consistent since $\sum_{i=1}^{n} Gateway + Utility = \sum_{i=1}^{n} Gateway' + Utility'$. Therefore, $E\backslash H \approx_B (E|\Pi)\backslash H$.

□

6.2.3. Internal Observer Without DGI, on the Physical System Composed With DGI. The system composed with power balancing process preserves non-deducibility. Intuitively, this is possible due to the invariance of physical flow (Equation 5). The nodes participating in power management process make their changes in such a way that the net power flow at the bus remains constant. This case was proved in an earlier work [24] using a gas pipeline system as test case. The nodes running the power management process, LB (defined in Equation 9) can be represented as in Equation 10.

$$LB \cong LB^S + LB^D \tag{9}$$

$$\begin{aligned} LB^S &\cong SendDraftRequest.LB^S + ReceiveDraftAge.LB^S + \\ &\quad ComputeDraftStandard.DraftSelect.LB^S.\tau.X_{SST}.LB \\ LB^D &\cong ReceiveDraftRequest.LB^D + SendDraftAge.LB^D + \\ &\quad AcceptDraft.LB^D.\tau.X_{SST}.LB \\ IEM &\cong (DRER.\overline{DRER}.IEM + DESD.IEM + Load.IEM).\tau.LB. \\ &\quad (\overline{DESD}.IEM + \overline{Load}.IEM + Gateway.IEM) \end{aligned} \tag{10}$$

$$E = (IEM\ 1|IEM\ 2|..|IEM\ n)|Node\ IO_{noDGI}|Bus \tag{11}$$

The system composed with the DGI process, E can be defined as in Equation 11. Assuming that the low-level internal observer, IO can observe the visible DRER sources, the classification of events within the system as defined in Equation 11 is $Low = \{DRER^n_{i=1}, Node\ IO_{noDGI}, \overline{Gateway}_{Node\ IO}\}$, $High = \{IEM\ 1, IEM\ 2\ ...\ IEM\ n\}$.

Theorem 3. *The system composed with the DGI process, as modeled in Equation 11 satisfies the BNDC property with respect to an internal observer without DGI.*

Proof. An internal observer without DGI cannot see the high-level message exchanges associated with the DGI processes. Given this, it is unaware of any power migration due to the power management algorithm. The high-level restriction on the system is $E\backslash H \cong \{DRER_{i=1}^{n},\ Node\ IO_{noDGI},\ Bus,\ \overline{Gateway}_{Node\ IO}\}$. The DGI power migration step between a *Supply* node, i and a *Demand* node, j is represented as $\Pi \equiv LB_i^S \underbrace{||}_{Migrate} LB_j^D$. The high-level restriction on the system composed with DGI, $(E|\Pi)\backslash H \cong \{DRER_{i=1}^{n},\ Node\ IO_{noDGI},\ Bus',\ \overline{Gateway}_{Node\ IO}\}$. However, Bus is consistent with Bus' due to the invariant defined in Equation 5. The total power on the bus connecting the three nodes (1, 2, IO) shown in Figure 6.2 is given by $P_{Bus} = P_{\overline{Gateway1}} + P_{\overline{Gateway2}} + P_{\overline{GatewayIO}}$. As a result of load balancing, if the migrated power from Node 1 to Node 2 is ζ units (such as watts), then $P'_{Bus} = (P_{\overline{Gateway1}} - \zeta) + (P_{\overline{Gateway2}} + \zeta) + P_{\overline{GatewayIO}}$. Power losses during migration are ignored for the sake of simplicity, leading to $P_{Bus} = P'_{Bus}$. The event Bus' could also be due to any process, (for instance $\overline{Gateway}_i.\overline{Gateway}_j$ as in Theorem 2) where $i, j \neq IO$. Therefore $E\backslash H \approx_B (E|\Pi)\backslash H$, making the system BNDC-secure. □

6.2.4. Internal Observer With DGI, on the System Composed With DGI. For an internal observer with DGI (Node 3) as shown in Figure 6.2, if Node 1 is in Supply state, it could be either supplying to Node 2 or selling power to the utility grid. On the other hand, if Node 2 is in Demand state, it is either receiving power from Node 1 or receiving from utility grid. Such an observer can infer about the global state of the system by analyzing the load table traces that are updated within its DGI process. A load table trace at every node (as shown in Table 5.1), can be represented in the trace model as a sequence of time varying tuples containing the state information. For example, $\Im_{\Delta t}$ = { (State(Node 1) at time t1,... State(Node n) at time t1), (State(Node 1) at time t2, .. State(Node n) at time t2), ..}. The

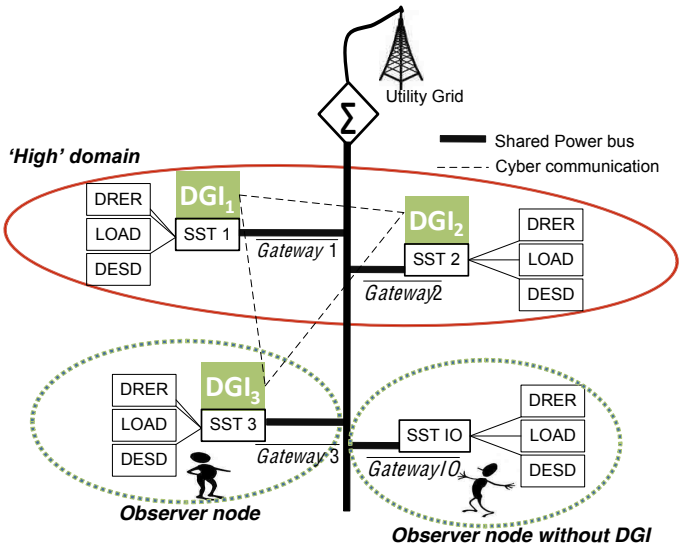

Figure 6.2. FREEDM subsystem with DGI, two nodes and two observers

observer's view of the system changes depending on the current demand state of the node, leading to different cases of information flow as below.

6.2.5. Observer in *Demand* State. From its load table trace, an observer can see the nodes that are in *Demand* state and *Supply* state. The quantity of information that is observable is more in this case, since it receives draft requests from all the nodes that are in *Supply* state. The observer in *Demand* state responds to the draft requests by sending its demand cost ($Draft\ Age$). If it receives a $Refusal$, it could be because the *Supply* node it responded to, has an inadequate matching cost to satisfy its requirement, or the *Supply* node has selected to draft with another *Demand* node which has a higher demand cost. In the case with only three IEMs, this doubt can be resolved as follows: If there is no other *Demand* node that the observer can see, then the *Supply* node does not have enough power to match its requirement. In this case, it can advertise a lesser cost until the time it succeeds. However, at the time it succeeds, it now has an estimate of the excess power the *Supply* node has, with which it can infer its *Load*.

Theorem 4. *The DGI power management process is not BNDC-secure with respect to an internal observer in* Demand *state.*

Proof. Let Π be a power migration step between IEM1 and IEM2 as shown in Equation 9. From its load table trace $\Im_t = \{(Supply, Demand)_t\}$, IEM3 initiates the high-level power balancing process Π' with IEM1. It advertises a cost, $\hat{Cost_3}$ and experiences a refusal, $\mathcal{R}$ that is a function of the advertised cost. Within the system defined in Equation 11, E, the following takes place:

$$IEM\ 1|IEM\ 2|IEM\ 3 \cong ([\Im_t.LB^S]_{IEM1} \underbrace{||}_{Migrate} [\Im_t.\hat{Cost_2}.LB^D]_{IEM2})|$$
$$([\Im_t.LB'^S]_{IEM1} \underbrace{||}_{Migrate} [\Im_t.\hat{Cost_3}.LB'^D]_{IEM3})$$

$$(IEM\ 1|IEM\ 2|IEM\ 3|\Pi)\backslash H \cong \{\Im_t, Bus', \overline{Gateway_3}\}$$

$$(IEM\ 1|IEM\ 2|IEM\ 3|\Pi|\Pi')\backslash H \cong \{\Im_t, Bus', \overline{Gateway_3}, \hat{Cost_3}, \mathcal{R}\}$$

The proof can easily be extended to n IEMs in the system to prove that $(E|\Pi)\backslash H \not\approx_B (E|\Pi|\Pi')\backslash H$. Hence the system is not BNDC-secure with respect to an internal observer in *Demand* state. □

Alternatively, the observer, on experiencing a $Refusal$ of its $Draft\ Age$, can bid a higher cost until the time it receives a $Draft\ Select$, meaning that it is selected by the *Supply* node to draft. In this case, cost of the other *Demand* node is divulged, along with interference of high level activity between the *Demand* node and the *Supply* node.

6.2.6. Observer in *Supply* State. The observer in *Supply* state can have information on the nodes that are in *Demand* state, with certainty. It initiates the $Draft\ Request$ to obtain the $Draft\ ages$ from the *Demand* nodes which include their respective demands. It is possible that the *Demand* nodes experience a refusal,

$\mathcal{R}$ since the observer is not actually ready for migration. The observer can continue this process of issuing fake draft requests resulting in the *Demand* node, not satisfying its request from any other IEMs in supply state. However, this case can be handled by not accepting any draft requests from the presumably supplying node after a certain number of refusals. Along with the low level physical observation and the demands advertised by the *Demand* nodes, the observer can infer critical information about DESD, Loads and strategy of SST at the *Demand* node.

Theorem 5. *The DGI power management process is not BNDC-secure with respect to an internal observer with DGI in* Supply *state.*

Proof. Based on its load table trace $\Im_t = \{(Demand, Demand)_t\}$, IEM3 initiates the load balancing process Π' with IEM1 and 2. Both the IEMs in *Demand* state respond with costs $\hat{Cost}_1$ and $\hat{Cost}_2$ respectively, revealing their demand. The proof is similar to Theorem 4.

$$(IEM\ 1|IEM\ 2|IEM\ 3)\backslash H \cong \{\Im_t, Bus', \overline{Gateway}_3\}$$

$$(IEM\ 1|IEM\ 2|IEM\ 3|\Pi')\backslash H \cong \{\Im_t, Bus', \overline{Gateway}_3, \hat{Cost}_1, \hat{Cost}_2\}$$

The proof can be extended to n nodes in the micro grid leading to the conclusion that $E\backslash H \ncong (E|\Pi')\backslash H$. This proves that the system does not satisfy BNDC property with respect to an internal observer in *Supply* state. □

6.2.7. Verification of a Single Node Involved in Power Migration Step. The system shown in Figure 6.3 is composed of three residential nodes (that include SST, DRER and DESD) with each running DGI and tied to the utility grid. However, the information flow analysis presented in the rest of this section is for a primitive FREEDM system that includes a single residential node running DGI and connected to the utility. It is verified whether this primitive system satisfies the SBNDC property with respect to a low-level observer. The events that take place

in FREEDM system (shown in Figure 6.3) can be classified for simplicity, into the following classes:

Events at a given cyber process (DGI, represented as *Cyber*):

RS: Read local physical state, *CO*: Compute,

IC: Issue command to a local physical controller (SST),

Send: Send message to peer cyber components,

$\overline{Recv}$: Receive message from peer cyber components.

Events at a given physical process (represented as *Physical*):

IN: Flow change preserving the invariant on the bus,

PO: Physical Observability,

SC: Local state change (due to stochastic usage of resources involving DRERs, DESDs, LOADs),

EC: Execute command from cyber component (DGI).

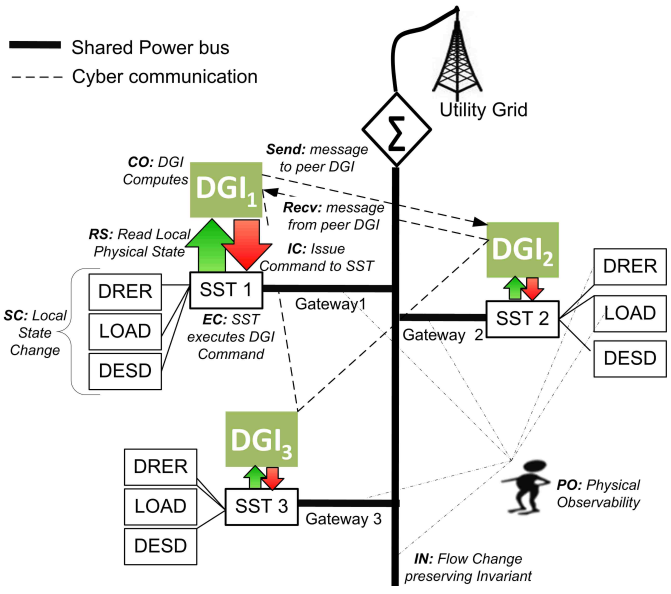

Figure 6.3. Events within the FREEDM system

IC, EC are complementary Input/Output pair. Similarly, SC and RS form a complementary I/O pair. For an observer internal to the system, the events can be classified as, $High = \{ RS, CO, IC, EC, SC, Send\}$ and $Low = \{PO, IN, \overline{Recv}\}$. The system can be built up in a bottom-up fashion as a composition of processes performing these events, as shown in Equations 12 through 16.

$$Cyber \cong RS.CO.IC.Cyber + RS.CO.Send.Cyber + \overline{Recv}.CO.IC.Cyber \quad (12)$$

$$Physical \cong PO.SC.IN.Physical + PO.EC.Physical \quad (13)$$

$$Node \cong Cyber|Physical \quad (14)$$

$$Invariant \cong (\overline{IN_1}.Invariant + \overline{IN_2}.Invariant...) + \tau.Invariant \quad (15)$$

$$System, E \cong (Node_1|Node_2...)|Invariant \quad (16)$$

The procedure to verify whether this system satisfies SBNDC is as follows: Unwinding E in terms of the events that define it and using the definition of composition, a possible state E' (such that $E \rightarrow E'$) can be defined as:

$$\begin{aligned} E' \cong\ & PO.\tau.IC.Node + \tau.CO.IC.Node + EC.\tau.\tau.PO.CO.IC.Node + \\ & PO.\tau.CO.Send.Node + \tau.CO.Send.Node + \\ & EC.\tau.CO.Send.\tau.PO.Node + \overline{Recv}.\tau.SC.\tau.PO.Node \end{aligned} \quad (17)$$

$$E'\backslash H \cong PO.\tau.Node + \overline{Recv}.\tau.Node \quad (18)$$

$$E' \approx^{0}_{\backslash H} PO.\tau.Node + \overline{Recv}.\tau.\tau.\tau.PO.Node \quad (19)$$

The bisimulation up to H on E' yields the relation shown in Equation 19. From Equations 18 and 19, $E' \not\approx^0_{\backslash H} E'\backslash H$. This proves that the generic CPS defined above does not satisfy SBNDC. This verification process can be automated by encoding the SPA formalism of the system into a model checking framework that is capable of verifying BNDC properties (E.g: CoPS [63]).

The trace that resulted in the failure of SBNDC is $\overline{Recv}.\tau.SC.\tau.PO.Node$ implying that on receiving a message, the $Node$ performs an internal action, a *State Change* event, followed by an internal action to maintain the $Invariant$ which is physically observable (for example, due to the voltage drop at a node on the power line, a small change is observed by a neighboring node sharing the physical line). In practice, such a physical change can be hidden through coordination with other nodes in which the following takes place: if one node makes a change, the other node(s) perform(s) a compensating event [24] [64]. This process E' can now be made SBNDC by adding a complementary $\overline{PO}$ event such that the effect of the PO event is nullified. Therefore, the system trace $\overline{Recv}.\tau.SC.\tau.PO.Node$ in Equation 17 can be modified as $\overline{Recv}.\tau.SC.\tau.PO.\overline{PO}.Node \cong \overline{Recv}.\tau.SC.\tau.\tau.Node$. The modified E' will have the following characteristics.

$$E'_{Modified}\backslash H \cong PO.\tau.Node + \overline{Recv}.\tau.Node \tag{20}$$

$$E'_{Modified} \approx^0_{\backslash H} PO.\tau.Node + \overline{Recv}.\tau.Node \tag{21}$$

Thus, $E'_{Modified}$ satisfies the SBNDC property. Similarly, it is verified for every E' reachable from E whether E' and $E'\backslash H$ belong to the bisimulation up to H relation. The possibility of multiple observers coordinating in an effort to deduce more information pertaining to the system is a challenging task that needs to be well addressed. Thus, the SBNDC property and its weak forms are sufficient for CPS verification (in many cases). This process reduces the problem of verifying the BNDC property on the SPA model of the CPS into verifying a bisimulation relation between

all the reachable states of the system. The problem of verifying the bisimulation equivalence between E and $E\backslash H$ is the next logical step to perform complete system evaluation. In Figure 6.4, part(a) demonstrates the RCP formulation of the primitive FREEDM system in Equations 17 to 19; part(b) demonstrates the RCP formulation of the modified FREEDM system in Equations 20 and 21.

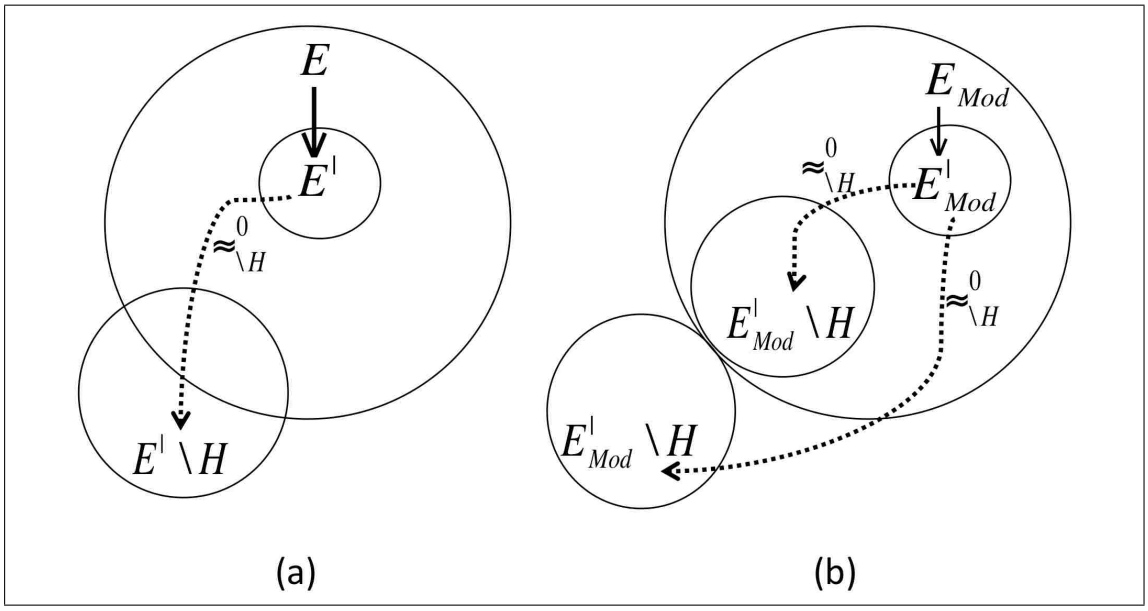

Figure 6.4. Relational coarsest partition formulation of FREEDM

In Figure 6.4(a), the partition including the state $E'\backslash H$ is unstable with respect to the partition containing E' (neither $E' \subseteq R^{-1}(E'\backslash H)$ nor $E' \cap R^{-1}(E'\backslash H) = \phi$) due to the transitions including PO event. This implies that there are some states of the system (brought about by high-level transitions) that are distinguishable with respect to the low-level observer's view ($E'\backslash H$). However, in the modified FREEDM system, as demonstrated in Figure 6.4(b), the partition including the state $E'_{Mod}\backslash H$ is stable with respect to the partition containing E'_{Mod}. This follows from the fact that the observer can only see either the states of the system brought about by the low-level

transitions or those states with high-level transitions transformed into internal τs (by the relation $\approx^0_{\backslash H}$). Thus this captures the idea that the low-level observer cannot distinguish between E'_{mod} and $E'_{mod}\backslash H$. Such a verification for coarsest partition over *bisimulation up to H* is performed for all the states $E' \in \xi : E \rightarrow E'$, with respect to $E'\backslash H$.

6.3. RESULTS WITH AUTOMATED VERIFICATION OF SBNDC ON FREEDM

Using the automated BNDC verifier, Checker of Persistent Security (CoPS) [63], the results obtained for the FREEDM system are tabulated in Table 6.1. The table shows whether each process of the FREEDM system satisfies SBNDC property or not. For each process, the graph generated indicates the state transitions of that process. Intuitively, these results suggest that DGI satisfies SBNDC properties assuming secure communications between various interacting processes. However, the components on the physical network that are inherently exposed fail to satisfy SBNDC and so is their composition with DGI. The modified microgrid in the example presented in Section 6.2.7 satisfies SBNDC due to the compensating event hiding the observable physical events.

The graph generated for each process in Table 6.1 suggests the complexity involved in exploring a large number of states (every E') to verify SBNDC. The verification of equivalence between E' and $E'\backslash H$, adds to the existing complexity. The states explored to verify the bisimulation equivalence using the Paige-Tarjan algorithm for equivalence testing [50] on the FREEDM system is 1339888. These results were generated using a computer with Intel Core 2 Duo processor having 2.4 GHz and 2GB memory running Mac OS X 10.5. The time taken was around 15 minutes and 30 seconds. The efficiency of equivalence testing can be improved by the

Table 6.1. Model checking results for the micro grid consisting of a single node

Process	**SBNDC**	**Generated Graph**	**Complexity**
MicroGrid: Node | DGI | Bus	No	V:3616, E:26543	15.5 Mins
MicroGrid (Modified)	Yes	V:5416, E:38641	13.9 Mins
Node: SST | DRER | DRES | Load	No	V:241, E:1120	-
DGI	Yes	V:5, E:8	-
Bus (Invariant)	No	V:3, E:3	-

development of algorithms that minimize the state space using advanced techniques of bisimulation and partial order reduction, as in [65].

The analysis using SPA has been a preliminary step in identifying a generic approach to model and verify CPSs. However, some limitations exist, particularly in modeling distributed message passing. In some CPSs like FREEDM, the communication links are dynamic in nature given the state-driven message passing initiated by the nodes. In SPA, there is no means to distinguish messages with same names originating from different nodes. To this end, the π-calculus has been adopted to overcome these limitations to model distributed communication.

7. INFORMATION FLOW ANALYSIS USING π-CALCULUS

In this section, the FREEDM system running DGI is modeled in the π-calculus framework to overcome the limitations incurred using SPA. Primarily, two aspects were ignored in the previous section- i) the private communication among peers based on demand state, and ii) distinguishing messages with same name originating from different peers based on channels of communication. In Figure 7.1, IEM3 which was in *Supply* initially involves in private communication with the *Demand* node, IEM1. However, IEM3 transitions from a *Supply* state to a *Demand* state, thereby establishing communication with a new *Supply* node, IEM2. Likewise, the communication structure changes based on the demand status of the nodes. This needs to be taken into account in the process specification of the system.

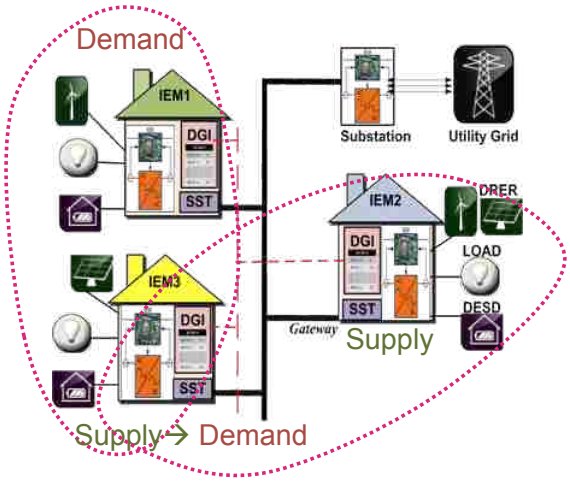

Figure 7.1. Need for scope extrusion in FREEDM

In π-calculus, the communication structure among existing processes can change over time. Dynamic communication (link mobility) among the peers can be represented using two features - i) restriction of a message to be sent or received over a

communication channel, and ii) scope extrusion. Scope extrusion allows the abstraction of the dynamically changing scope of the communication domain, by sending channel names as messages to processes that did not have access to those channels before. The π-calculus offers other useful features such as abstractions to model adversarial behavior, different notions of process equivalences to define different notions of security. These features are applied in the FREEDM system specification presented in this section.

7.1. MODELING OF FREEDM USING π-CALCULUS

Algorithm 1 presents the DGI power management algorithm that will be verified in composition with the system. In this section, an approach similar to [66] is considered, in which peer to peer algorithms were modeled using the π-calculus to verify their functional correctness.

Algorithm 1: DGI power management algorithm

```
{Variables: R:Renewable, L:Load, S:Storage}
Get Values: R/L/S/gateway
Compute Supply/Demand/Normal
if Demand then
   Broadcast ("Demand")
   if Recv ("Draft Request") then
      Send ("Draft Response")
   end if
   if Recv ("Draft Select") then
      Send ("Accept")
      Set R/L/S/gateway
   end if
end if
if Supply then
   Broadcast to Demand nodes ("Draft Request")
   if Recv("Draft Response") then
      Send ("Draft Select")
   end if
   if Recv ("Accept") then
      Set R/L/S/gateway
   end if
end if
```

In Figure 7.2, the modeling of a node running DGI is depicted as a composition of π processes. To facilitate modeling of distributed communication among various components within a node and with peer nodes, a number of channels are assumed to terminate/start from their respective ports. The ports e_1, e_2, e_3 represent the terminals of the dedicated channels used to communicate the demand states ($Normal$, $Supply$, $Demand$ respectively) of each node as messages sent or received by the DGI process. Similarly, m_1, m_2, m_3, m_4 serve as the terminal ports of the channels dedicated for the four kinds of power management messages. The s_l, s_g, s_r, s_s are the ports to communicate with the local devices (LOAD, SST, Renewable and Storage respectively). Unique identifiers (UUIDs) are assumed to be in place to label the message with the identifier.

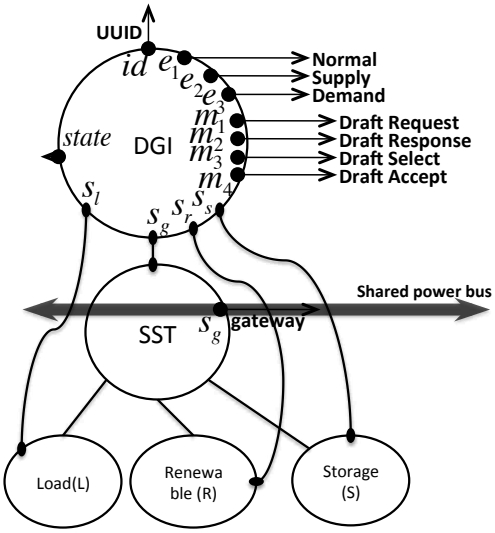

Figure 7.2. π-characterization of a cyber-physical process

The physical system can be similarly defined as a network of components that interact over channels. Equations 22 to 28 represent the formal specifications of the

FREEDM system. Equation 22 represents the modeling of the DGI power management algorithm of Algorithm 1. In Equation 23, Ω represents a power migration step between a pair of nodes (in supply and demand states respectively), initiated after successful DGI negotiations. The operation of physical components at each node (including SST, DRER, DESD and LOAD) is abstracted to that of sending and receiving a vector of corresponding power settings, as in Equation 24. The invariant acting on the bus can be developed inductively for n nodes in the system as in Equation 25. This invariant captures the invariance of power flow acting on the bus, following Kirchoff's current laws.

$$
\begin{aligned}
&DGI(id^x, \vec{m}, \vec{e}, \vec{s^x}, \vec{state^x}) \triangleq \\
&\quad s^x(\vec{s^x}).\tau.\bar{state^x} < \vec{state} > .DGI(id^x, \vec{m}, \vec{e}, \vec{s^x}, \vec{state^x}) \; + e^x(\vec{e^y}).\tau. \\
&\quad DGI(id^x, \vec{m}, \vec{e}, \vec{s^x}, \vec{state^x}) \; + \; state^x(SND).if \; SND = Supply \\
&\quad then \; \{^{SND}/_{Supply}\}.\bar{m_1^d} < m_1^x > .\Omega(id^x, id^d) + \; state^x(SND). \\
&\quad if \; SND = Demand \; then \; \{^{SND}/_{Demand}\}.\bar{e^x} < \vec{e_3^y} > .DGI(id^x, \vec{m}, \\
&\quad \vec{e}, \vec{s^x}, \vec{state^x}) \; + \; state^x(Demand).m_1^x(m_1^s).\bar{m_2^s} < m_2^x > .\Omega(id^x, id^s)
\end{aligned}
\tag{22}
$$

$$
\begin{aligned}
&\Omega(id^x, id^y) \triangleq \\
&\quad state^x(SND).if \; SND = Supply \; then \; \nu C_{xy}(m_2^x(m_2^y).\bar{m_3^y} < m_3^x > . \\
&\quad m_4^x(m_4^y)).\tau.\bar{s^x} < \vec{s^x} > .\mathbf{0} + state^x(SND). \\
&\quad if \; SND = Demand \; then \; \nu C_{xy}(m_3^x(m_3^y).\bar{m_4^x} < m_4^y >).\tau.\bar{s^x} < \vec{s^x} > .\mathbf{0}
\end{aligned}
\tag{23}
$$

$$
\begin{aligned}
&PHY(id^x, \vec{s^x}) \triangleq \\
&\quad \tau.s^x(\vec{s^x}).\bar{bus} < s_g^x > .PHY(id^x, \vec{s^x}) + \bar{s^x} < \vec{s^x} > .\tau.PHY(id^x, \vec{s^x})
\end{aligned}
\tag{24}
$$

$$INV(\vec{\delta}, n) \triangleq$$

$$\prod_{i \in n} bus(s_g^i).grid(u).INV(\vec{\delta}, n) + \bar{grid} < u > . \prod_{i \in n} \bar{bus}(s_g^i + \delta^i). \tag{25}$$

$$INV(\vec{\delta}, n), where\ |n| = \#Nodes.$$

$$INV(\vec{\delta}, 3) \triangleq$$

$$bus(s_g^1).bus(s_g^2).bus(s_g^3).grid(u).INV(\vec{\delta}, 3) \ + \bar{grid} < u > . \tag{26}$$

$$\bar{bus} < s_g^1 + \delta^1 > .\bar{bus} < s_g^2 + \delta^2 > .\bar{bus} < s_g^3 + \delta^3 > .INV(\vec{\delta}, 3)$$

$$Node(id^x, \vec{m}, \vec{e}, \vec{s^x}, \vec{state^x}) \triangleq \tag{27}$$

$$DGI(id^x, \vec{m}, \vec{e}, \vec{s^x}, \vec{state^x}) | PHY(id^x, \vec{s^x})$$

$$System, S \triangleq$$

$$Node(id^1, \vec{m}, \vec{e}, \vec{s^1}, \vec{state^1}) | Node(id^2, \vec{m}, \vec{e}, \vec{s^2}, \vec{state^2}) | ..... | \tag{28}$$

$$Node(id^n, \vec{m}, \vec{e}, \vec{s^n}, \vec{state^n}) | INV(\vec{\delta}, n)$$

7.2. INFORMATION FLOW PROPERTIES IN π-CALCULUS

Non-interference was previously studied in terms of bisimulation semantics, in the typed π-calculus framework in [67] and [68]. In this work, the non-deducibility property presented in π-calculus is based on the behavioral equivalence, the reduction closed barbed congruence [69]. The security requirement is to preserve the non-deducibility of cyber-physical activity involving the power migrations between various supply and demand nodes, with respect to an observer process that could be a legitimate node in the system as shown in Figure 7.3. The proofs presented rely on the idea of *context* which is fundamental in abstracting the environment with which the observer interacts. These concepts are formally defined in Definitions 2 and 4.

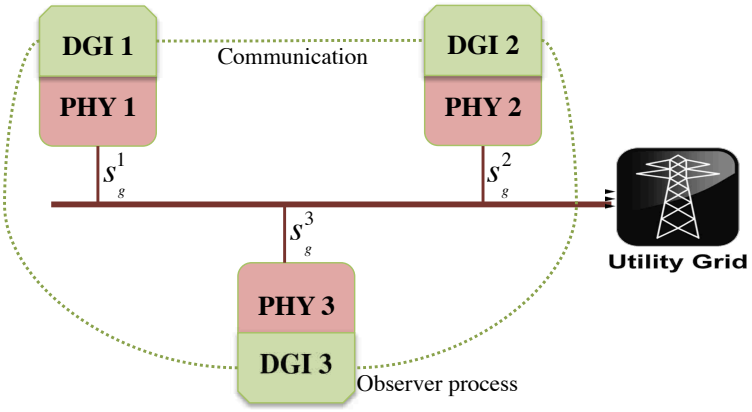

Figure 7.3. An observer process interacting with FREEDM

Definition 2 (Context). *A context, C is a subprocess given by the grammar in π-calculus, that defines the environment of the system in which the hole or place holder can be replaced with an observer process.*

$C :: [.] \mid \pi.C[.]+M \mid \nu aC[.] \mid C[.]|P \mid !C[.]$, *where P is a process within the system.*

Definition 3 (Barbed Bisimilarity). *Two processes, P and Q are barbed bisimilar, $(P \dot{\approx} Q)$, if*

1) $P \downarrow_\mu$ implies $Q \Downarrow_\mu$ and vice-versa.

2) $P \xrightarrow{\tau} P'$ implies $Q \xrightarrow{\tau} Q'$ and vice-versa.

In Definition 3, P $\downarrow_\mu$ implies that P can perform any input action with subject μ. Q $\Downarrow_\mu$ implies that Q performs the actions with subject μ with prefixed and suffixed number of τ actions ($Q \xrightarrow{\tau*} \xrightarrow{\mu} \xrightarrow{\tau*}$).

Definition 4 (Reduction Closed Barbed Congruence). *Two processes, P and Q are reduction closed barbed congruent, $(P \dot{\simeq}^c Q)$, if*

1) they are barbed bisimilar i.e., $P \dot{\approx} Q$ and

2) $C[P] \dot{\approx} C[Q]$, for every context C.

The definition 5 suggests that the system satisfies Basic π-ND if the observer cannot distinguish the system when composed with any high level process, for all contexts of the system.

Definition 5 (Basic π-non-deducibility (Basic π-ND))**.** *A process, P satisfies Basic π-ND if $P \dot{\simeq}^c P|\mathcal{H}$, where $\mathcal{H}$ is a high-level process composed of the names $\in H$.*

7.3. VERIFICATION OF INFORMATION FLOW USING π-CALCULUS

In Figure 7.3, a simple three node FREEDM system is shown. In order to verify if information flows from one node to the other during critical operations, an observer process that is a valid node within the system is considered. Intuitively, the observer should not be able to distinguish between the contexts that have and haven't performed the high-level power migration step. It needs to be established that $Node3 \dot{\simeq}^c Node3|H$ to prove that the system satisfies Basic π-ND (Definition 5) with respect to the observer.

7.3.1. Observer (Context) in the System Without DGI. An observer with no DGI cannot send or receive cyber messages within the system. Conceptually, Theorem 6 establishes that such an observer in a basic power system cannot deduce individual power settings.

Theorem 6. *The system without DGI is inherently secure with respect to a low-level observer.*

Proof. To prove that $Node3 \dot{\simeq}^c Node3|H$, it is required to establish the following as defined in Definition 4.

- $Node3 \dot{\approx} Node3|H$
- $C[Node3] \dot{\approx} C[Node3|H]$, for the context C, defined in Equation 29.

$$C :: [.]|S \tag{29}$$

The high-level process in this case is the sequence of settings on the bus, made by nodes other than the observer. Without DGI, the system with two nodes functions as:

$$\begin{aligned} S' \triangleq Node(id^1, \vec{s^1})|Node(id^2, \vec{s^2})|INV(\vec{\delta}, 3) := (\tau.\bar{bus} < s_g^1 > .S' + \tau. \\ bus(s_g^1).S')|(\tau.\bar{bus} < s_g^2 > .S' + \tau.bus(s_g^2).S')|INV(\vec{\delta}, 3) \end{aligned} \tag{30}$$

For $S' \xrightarrow{\bar{bus}<s_g^1+\delta^1>} \xrightarrow{\bar{bus}<s_g^2+\delta^2>} \hat{S'}$, and $Node3 \downarrow_{\bar{bus}<s_g^1>,\bar{bus}<s_g^2>}$ is $Null$. Similarly, $Node3|H := Node3|(\bar{bus} < s_g^1 + \delta^1 > |\bar{bus} < s_g^2 + \delta^2 >) \Downarrow_{\bar{bus}<s_g^1>,\bar{bus}<s_g^2>}$ is $Null$. Inserting the observer node in the context of Equation 29, the following are indistinguishable:

$C[Node3] := Node3|(\tau.\tau.grid(u).S' + \tau.bus(s_g^1).S')|(\tau.\tau.grid(u).S' + \tau.$
$bus(s_g^2).S')|\tau.\tau.bus(s_g^3).grid(u).INV(\vec{\delta}, 3)$

$C[Node3|H] := Node3|(\tau.\tau.grid(u).S' + \tau.bus(s_g^1 + \delta^1).S')|(\tau.\tau.grid(u).S' +$
$\tau.bus(s_g^2 + \delta^2).S')|\tau.\tau.bus(s_g^3).grid(u').INV(\vec{\delta}, 3)$

Therefore, $C[Node3] \downarrow_{\bar{bus}<s_g^1>,\bar{bus}<s_g^2>} \dot{\approx} C[Node3|H] \Downarrow_{\bar{bus}<s_g^1+\delta^1>,\bar{bus}<s_g^2+\delta^2>}$ and hence the conclusion, $Node3 \dot{\simeq}^c Node3|H$. $Node3$ does not observe a change in gateway value, s_g^3 due to the maintenance of the invariant on the bus, such that $s_g^1 + s_g^2 + s_g^3 + u = (s_g^1 + \delta^1) + (s_g^2 + \delta^2) + s_g^3 + u'$. □

Assuming nodes 1 and 2 are in supply and demand respectively, the following conclusions can be made regarding the observability of power migration between them. Different observations can be drawn depending on the demand state of the observer.

7.3.2. Observer (Context) in Supply State in the System With DGI. The observer in supply state, $Node3_s$ is first defined in Equation 31, and the context including this observer is then verified in Theorem 7.

$$
\begin{aligned}
Node3_s \triangleq Node(id^3, \vec{m}, \vec{e}, \vec{s^3}, \vec{state^3}) := \tau.\tau.\bar{bus} < s_g^3 > .\tau.\bar{state^3} \\
< \vec{state^3} > .Node3_s \ + e^3(\vec{e^2}).\tau.Node3_s \ + state^3(SND). \\
\{^{SND}/_{Supply}\}.\bar{m_1^2} < m_1^3 > .Node3_s \ + state^3(Supply).m_2^3 \\
(m_2^2).Node3_s
\end{aligned} \tag{31}
$$

Theorem 7. *The system is not secure with respect to a low-level observer in supply state.*

Proof.

$$
\begin{aligned}
S'' \triangleq Node(id^1, \vec{m}, \vec{e}, \vec{s^1}, \vec{state^1}) | Node(id^2, \vec{m}, \vec{e}, \vec{s^2}, \vec{state^2}) | INV(\vec{\delta}, 3) \\
:= \tau.\tau.\bar{bus} < s_g^1 > .\tau.\bar{state^1} < \vec{state^1} > .S'' \ + \tau.\tau.\bar{bus} < s_g^2 > .\tau. \\
\bar{state^2} < \vec{state^2} > .S'' \ + e^1(\vec{e^2}).\tau.S'' \ + state^1(Supply).state^2 \\
(Demand).\tau.\tau.S'' \ + state^2(Demand).\bar{e^2} < \vec{e_3^1} > .S'' \ + e^2(\vec{e^1}).\tau. \\
S''|INV(\vec{\delta}, 3)
\end{aligned} \tag{32}
$$

$$
\begin{aligned}
S''|\Omega(id^1, id^2) \triangleq Node(id^1, \vec{m}, \vec{e}, \vec{s^1}, \vec{state^1}) | Node(id^2, \vec{m}, \vec{e}, \vec{s^2}, \vec{state^2}) | \\
\Omega(id^1, id^2) | INV(\vec{\delta}, 3) := \tau.\tau.\bar{bus} < s_g^1 > .\tau.\bar{state^1} < \vec{state^1} > .S'' \\
+ \tau.\tau.\bar{bus} < s_g^2 > .\tau.\bar{state^2} < \vec{state^2} > .S'' \ + e^1(\vec{e^2}).\tau.S'' \ + \\
state^1(Supply).state^2(Demand).\tau.\tau.S'' \ + state^2(Demand). \\
\bar{e^2} < \vec{e_3^1} > .S'' \ + e^2(\vec{e^1}).\tau.S'' \ + \nu C_{12}(\tau.\tau.\tau.\tau.\tau.\tau.\tau.\tau). \\
(\tau.\tau.\bar{bus} < s_g^1 > + \tau.\tau.\bar{bus} < s_g^2 >).S''
\end{aligned} \tag{33}
$$

To verify whether $Node3_s \doteq^c Node3_s|\Omega(id^1, id^2)$ [1], it is required to establish the following as defined in Definition 4.

1) $Node3_s \dot{\approx} Node3_s|\Omega^{12}$

2) $C[Node3_s] \dot{\approx} C[Node3_s|\Omega^{12}]$, for the context C, defined in Equation 29.

Trivially, $Node3_s \dot{\approx} Node3_s|\Omega^{12}$, because $Node3_s$ does not interact with the process, Ω^{12}. Intuitively, Ω^{12} involves private communication among nodes 1 and

[1] $\Omega(id^1, id^2)$ will be used as Ω^{12} for simplicity.

2 and issue of commands to their respective physical subsystems, all of which are invisible to an external observer ($Node3_s$). However, in the context of the whole system the result is different as shown below in Equations 34 and 35. In Equation 34, $C[Node3_s] \downarrow_{\Omega^{12}}$ is $\{e^2(e^{\vec{3}})\}$ while $C[Node3_s|\Omega^{12}] \Downarrow_{\Omega^{12}}$ (in Equation 35,) is $\{e^2(e^{\vec{3}}), bus(s^3_{g'})\}$. $C[Node3_s|\Omega^{12}]$ captures the notion that $bus(s^1_g)$. $bus(s^2_g)$. $bus(s^3_g)$ takes place on the bus leading to an observable event $bus(s^3_{g'})$.

$$
\begin{aligned}
C[Node3_s] \triangleq S''|Node3_s := {} & \tau.\tau.\bar{bus} < s^1_g > .\tau.\bar{state}^1 < \vec{state}^1 > .C + \\
& \tau.\tau.\bar{bus} < s^2_g > .\tau.\bar{state}^2 < \vec{state}^2 > .C \ + e^1(e^{\vec{2}}).\tau.C \ + state^1 \\
& (Supply).state^2(Demand).\tau.\tau.C \ + state^2(Demand).\bar{e^2} < e^{\vec{1}}_3, e^{\vec{3}}_3 > \\
& .C \ + e^2(e^{\vec{1}}).\tau.C + \tau.\tau.\bar{bus} < s^3_g > .\tau.\bar{state}^3 < \vec{state}^3 > .C + \\
& e^3(e^{\vec{2}}).\tau.C
\end{aligned} \tag{34}
$$

$$
\begin{aligned}
C[Node3_s|\Omega^{12}] \triangleq S''|Node3_s|\Omega^{12} := {} & \tau.\tau.\tau.\tau.C + e^1(e^{\vec{2}}).\tau.C \ + state^2 \\
& (Demand).\bar{e^2} < e^{\vec{1}}_3, e^{\vec{3}}_3 > .C \ + \tau.\tau.\bar{bus} < s^3_{g'} > .\tau.\bar{state}^3 \\
& < \vec{state}^3 > .C \ + e^3(e^{\vec{2}}).\tau.C
\end{aligned} \tag{35}
$$

Intuitively, this implies that the observer is able to distinguish the case of a power migration between nodes 1 and 2 and non-occurence of power migration, by determining that node 2 has recently been in Demand state. Hence $Node3_s \dot{\not\approx}^c Node3_s |\Omega(id^1, id^2)$. □

7.3.3. Observer (Context) in Demand State in the System With DGI. The case in which the observer is in Demand state is similar to the case in which the observer is in Supply state. However, the evaluation context consists of the observer node in demand state, $Node3_d$.

$$
\begin{aligned}
Node3_d \triangleq Node(id^3, \vec{m}, \vec{e}, \vec{s^3}, \vec{state^3}) := \tau.\tau.\bar{bus} < s_g^3 > .\tau.\bar{state^3} \\
< \vec{state^3} > .Node3_d \ + e^3(\vec{e^2}).\tau.Node3_d \ + state^3(SND). \\
\{^{SND}/_{Demand}\}.\bar{e^3} < \vec{e_3^1} > .Node3_d \ + state^3(Demand). \\
m_1^3(m_1^1).\bar{m_2^1} < m_2^3 > .Node3_d
\end{aligned} \tag{36}
$$

Theorem 8. *The system is not secure with respect to a low-level observer in demand state.*

Proof. For the same reason as mentioned in Theorem 7, proving $Node3_d \dot{\approx} Node3_d|\Omega^{12}$ is trivial, because $Node3_d$ does not interact with the process, Ω^{12}.

$$
\begin{aligned}
C[Node3_d] \triangleq S''|Node3_d := \tau.\tau.\bar{bus} < s_g^1 > .\tau.\bar{state^1} < \vec{state^1} > .C \ + \\
\tau.\tau.\bar{bus} < s_g^2 > .\tau.\bar{state^2} < \vec{state^2} > .C \ + e^1(\vec{e^2}).\tau.C \ + state^1 \\
(Supply).state^2(Demand).\tau.\tau.C \ + state^2(Demand).\bar{e^2} < \vec{e_3^1}, \vec{e_3^3} > \\
.C \ + e^2(\vec{e^1}).\tau.C + \tau.\tau.\bar{bus} < s_g^3 > .\tau.\bar{state^3} < \vec{state^3} > .C \ + \\
e^3(\vec{e^2}).\tau.C
\end{aligned} \tag{37}
$$

$$
\begin{aligned}
C[Node3_d|\Omega^{12}] \triangleq S''|Node3_d|\Omega^{12} := \tau.\tau.\tau.\tau.C + e^1(\vec{e^2}).\tau.C \ + state^2 \\
(Demand).\bar{e^2} < \vec{e_3^1}, \vec{e_3^3} > .C \ + \tau.\tau.\bar{bus} < s_{g'}^3 > .\tau.\bar{state^3} \\
< \vec{state^3} > .C \ + e^3(\vec{e^2}).\tau.C
\end{aligned} \tag{38}
$$

$C[Node3_d] \downarrow_{\Omega^{12}}$ is $\{ e^2(\vec{e^3}),\ m_1^3(m_1^1)\}$ while $C[Node3_d|\Omega^{12}] \Downarrow_{\Omega^{12}}$ is $\{ e^2(\vec{e^3}),\ m_1^3(m_1^1),\ bus(s_{g'}^3)\}$. The context $C[Node3_d|\Omega^{12}] \Downarrow_{\Omega^{12}}$ captures the notion that $bus(s_g^1).bus(s_g^2).bus(s_g^3)$ takes place on the bus leading to an observable event $bus(s_{g'}^3)$. The observer is able to distinguish the case of a power migration between nodes 1 and 2 and non-occurence of power migration, by determining that node 1 is in Supply state and node 2 is in Demand state. Clearly, $Node3_d \dot{\not\approx}^c Node3_d|\Omega(id^1, id^2)$. □

7.3.4. Making the FREEDM System π-ND-secure. In the previous sections, it was proved that the FREEDM system is not π-ND when there are three

nodes in the system. However, if two other nodes (4 and 5) participate as in Figure 7.4 so that there are two Supply and two Demand nodes, the power migration between any pair of Supply and Demand nodes can be hidden as shown in Theorem 9.

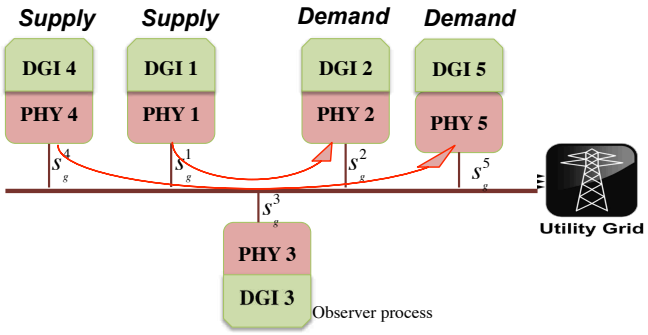

Figure 7.4. An observer process interacting with the 5-node FREEDM system

Theorem 9. *The system with five nodes is secure with respect to a low-level observer in supply state, when more than one power migrations occur in parallel.*

Proof. This proof shows that it holds true for the case of five nodes with two power migrations occurring in parallel since this is a base case. Since five nodes exist, Equation 32 would now include the invariant for five nodes, i.e., $INV(\vec{\delta}, 5)$ instead of $INV(\vec{\delta}, 3)$. In this case, $C[Node3_s] \downarrow_{\Omega^{12}}$ is $\{e^2(e^{\vec{1}}), e^2(e^{\vec{4}}), e^3(e^{\vec{2}}), e^3(e^{\vec{5}})\}$ and $C[Node3_s|\Omega^{12}] \Downarrow_{\Omega^{12}}$ is $\{e^2(e^{\vec{1}}), e^2(e^{\vec{4}}), e^3(e^{\vec{2}}), e^3(e^{\vec{5}}), bus(s^3_{g'})\}$. The observer at $Node3_s$ cannot deduce from the observation on the bus, $bus(s^3_{g'})$ if a power migration occurred between the pair of nodes (1, 2) or (4, 5) or both. In this case, $Node3_s \dot{\simeq}^c Node3_s|\Omega^{12}$. The same conclusion holds for the power migration between nodes 4 and 5, i.e., $Node3_s \dot{\simeq}^c Node3_s|\Omega^{45}$. The proof can be extended to the case of more than two power migrations occurring in the system consisting of more than five nodes. □

In the next section, the above results will be verified using automatic equivalence checkers.

8. AUTOMATIC VERIFICATION USING π-CALCULUS TOOLS

In this section, the validity of the manual proofs used in the non-deducibility analysis of the FREEDM system in the π-calculus framework is verified. Process algebraic equivalence checkers like Mobility WorkBench [70] and Proverif [71] are employed to automate the verification process. These tools accept the system specification in terms of π-calculus syntax and verifies certain equivalence properties. The MWB is a tool developed for polyadic π-calculus and verifies observational equivalence between processes. The π-ND property used in this analysis however uses a finer refinement of observational equivalence, the *reduction closed barbed congruence* (RCBC) and hence the capability of this tool is limited. However, it can be proved if the RCBC relation does not apply to two processes by proving that two processes are not weak observational equivalent. Proverif, on the other hand is more robust because it accepts the applied π-calculus specification of the system and provides in-built commands to verify a number of secrecy properties that are of interest to CPS analyses. Various analyses of the FREEDM specification is carried out using each of these tools. It is to be noted that the FREEDM system only serves as a test example of a complex distributed CPS and the modeling shown below provides the reader with how this analysis approach can be extended to other CPSs.

8.1. MWB

The Mobility Workbench (MWB) is an automated tool for manipulating and analyzing mobile concurrent systems specified in π-calculus. MWB proves that something is Barbed Bisimilar (BB, $\dot{\approx}$) but not Reduction Closed Barbed Congruent (RCBC, $\dot{\simeq}^c$). That is, if two processes P, Q are such that P $\dot{\approx}$Q , they need not be P $\dot{\simeq}^c$Q. However, the vice-versa is true. Therefore RCBC is a finer refinement of BB.

To verify the weak or observational equivalence, MWB is used as follows:

weq *Process1* *Process2*

The MWB encoding for the FREEDM system is presented in Appendix. The results obtained for a 3-node and 5-node system are summarized in Table 8.1.

Table 8.1. Basic π-ND results for observer in supply and demand states using MWB

Process 1	**Process 2**	**Comment**	**MWB**	
			$\dot{\approx}$	**Complexity**
S: Node1 | Node2	S$\|\Omega^{12}$ $\|Node3_S$	Power migration between two nodes with respect to an observer in supply state	**x**	28.48 mins
S: Node1 | Node2	S$\|\Omega^{12}$ $\|Node3_D$	Power migration between two nodes with respect to an observer in demand state	**x**	29.12 mins
S': Node1 |..| Node5	S'$\|\Omega^{12}\|\Omega^{45}$ $\|Node3_S$	Two consecutive power migrations in a 5-node system with respect to an observer in supply state	✓	49.3 mins
S': Node1 |..| Node5	S'$\|\Omega^{12}\|\Omega^{45}$ $\|Node3_D$	Two consecutive power migrations in a 5-node system with respect to an observer in demand state	✓	50.1 mins

8.2. PROVERIF

Proverif [72] is a tool, primarily intended to verify cryptographic protocols. It accepts as input, the protocol specification in applied π-calculus [73] and verifies claimed security properties. The tool helps uncover potential security violations in the implementation of cryptographic protocols in the presence of an attacker who may have complete control over the communication channels. An attacker with such capability, often referred to as a "Dolev-Yao" [74] attacker, can read, modify and inject messages into the system. In this work, this tool was used in a different way to reveal information flow violations in CPS environments by abstracting away, the

cryptographic primitives defined in the tool language. The verification of π-ND using Proverif is inspired from the privacy verification of electronic voting protocol, FOO92 in [75].

8.2.1. π-ND. The encoding of the FREEDM system in Proverif is shown in Figures 8.1 to 8.5. The observer is able to break the confidentiality of power migration among the nodes by distinguishing observer contexts in supply and demand states above. Additional properties that can be verified using Proverif are given in Figure 8.6.

```
free Set:bitstring [private].
free g1,g2:G [private].
free g3:G.
free delta1, delta2:netValue [private].
```

Figure 8.1. Defining variables and names to initialize the FREEDM Proverif script

```
(* Invariant process*)
fun Inv(G):netValue.
reduc forall g:G, value:netValue; sum(Inv(g), value)= value.
```

Figure 8.2. Proverif process defining the physical invariant of flow

8.2.2. Strong Secrecy. Strong secrecy is preserved when the attacker is unable to distinguish when the secret changes. With respect to the FREEDM system, the attacker should not distinguish between the cases when a supply node migrates a units of power in a single power migration step and b units of power. To perform such

```
(* DGI process1 in Supply state *)
let DGI_Supply =
    out(c, Status); (*Request sent to only demand nodes*)
    out(c, Request);
    in(ch, x_Response:bitstring);
    if x_Response = Response then
    out(ch, choice[Select, Reject]);
    in(ch, x_Accept:bitstring);
    if x_Accept = Accept then
    out(c_phy, Inv(g1));
    let y=sum(Inv(g1), delta1) in
    out(ch, Set).
```

Figure 8.3. Proverif process for a DGI node in supply state

```
(* DGI process2 in Demand state *)
let DGI_Demand =
    out(c,Status);
    in(c,x_Request:bitstring);
    if x_Request=Request then
    out(ch, Response);
    in(ch, x_Select:bitstring);
    if x_Select=Select then
    out(ch, Accept);
    out(c_phy, Inv(g2));
    let y=sum(Inv(g2), delta2) in
    out(ch, Set).
```

Figure 8.4. Proverif process for a DGI node in demand state

an attack, the attacker is assumed to detect power withdrawal from the utility and read the event of a spike in the power on the shared power bus. Strong secrecy can be verified in Proverif using the **noninterf** keyword as shown in Figure 8.6. It can be argued that strong secrecy is preserved in the FREEDM system since the migrations currently take place in fixed quantum of power. This property is particularly useful

to analyze the effect of partial information revealed to the attacker. For example, in the case of a series of power migrations between a supply-demand pair involving migratable quantum of non-uniform magnitude, the adversary can distinguish each of the series indicating that he obtains partial information. However, in a FREEDM system with 5 or more nodes, strong secrecy is always preserved unless there is a unique pair of nodes in Supply and Demand states, respectively.

```
(* The main process defining the system *)
process
        (!(DGI_Supply) | !(DGI_Demand) )
```

Figure 8.5. Proverif process defining FREEDM

8.2.3. Weak Secrecy. Weak secret refers to the information that the attacker may guess through passive or active observation of the system. A common example of weak secrets are human memorable passwords used in some protocols which are often values with low information entropy. The attacker may enumerate all the possible values and end up with the exact value though repeated trial. In Proverif, any name can be verified to determine if it is a weak secret, using the **weaksecret** keyword as shown in Figure 8.6.

```
query attacker(delta1).
query attacker(delta2).
weaksecret delta1.
noninterf g1, g2.
```

Figure 8.6. Using Proverif secrecy features on FREEDM

The time taken for the verification of all the properties is about 1.14 seconds and about 1.45 seconds for weak equivalence on a computer with Intel Core 2 Duo processor having 2.4 GHz and 2GB memory running Mac OS X 10.5. This proves that Proverif is far efficient compared to MWB. The results with Proverif are summarized in Table 8.2.

Table 8.2. Results of verification with Proverif

Property Verified	**Process/Name**	**3-node system**		**5-node system**	
		Result	**Complexity**	**Result**	**Complexity**
Basic π-ND	DGI_Supply, DGI_Demand	**x**	1.28 secs	✓	1.45 secs
Strong Secret	g1, g2	✓	1.14 secs	✓	1.14 secs
Weak Secret	delta1, delta2	✓	1.14 secs	✓	1.14 secs

9. CONCLUSIONS

This dissertation investigates the confidentiality properties in cyber-physical systems using formal methods. Modeling and verification of information flow properties for CPSs was discussed. First, the need for information flow analysis in CPSs has been established. Several information flow properties exist, of which non-inference and non-deducibility are appealing to distributed CPSs. Intuitively, an observer who is able to distinguish the states of the system before and after the execution of a critical event has more information about the system. The notion of bisimulation provides a way to check whether two processes are behaviorally equivalent. This property is instrumental in defining non-deducibility based properties using process algebras. However, specification of the system and its attributes precede the verification process. Modeling requires representation of the CPS and its diverse components to be represented under a uniform framework. The use of process algebras was demonstrated as a way of unifying the continuous and discrete aspects of CPS. Process algebras were applied to a test CPS, to illustrate the proposed modeling approach through which the behavior of a CPS, including the discrete distributed communication involving computation and the continuous flow dynamics of the underlying physical system can be represented in a unified semantic framework. The uniform representation of the CPS was later verified for known information flow properties. The analysis includes undesirable cases of information flow to an attacker in different contexts of the system operation.

The approach presented to analyze information flow in CPSs has three key steps outlined below.

1) Specification of the system with the cyber and physical components developed as communicating processes : The CPS was first represented

as a composition of Security Process Algebra processes. Aspects of process behavior such as concurrency, non-determinism, event transitions and communication had to be taken into account. Additionally, the events comprising the processes can be classified into H or L security domains using SPA. Due to the modeling limitations of SPA, π-calculus was later studied to specify the behavior of CPSs that involve link mobility and advanced distributed features such as sending messages over communication channels.

2) **Develop the continuous aspect as an interactive process that abstracts away the continuous nature through discretization of physical flow guided by an invariant :** In SPA as well as in the π-calculus, the continuous environment (power flow in the test CPS) was developed as a process composed of discrete events of change in value of flow that follows a physical invariant (Kirchoff's law in the test CPS). Such a process can interact with the rest of the processes that comprise the system, as a cyber process. This was sufficient in order to verify security properties instead of a more rigorous partial differential equation system since the events of change that cause an observable event are of the main concern.

3) **Represent the information flow model in terms of the equivalence verification capability offered by the chosen process algebra :** Using SPA, non-deducibility was verified on the test CPS by representing in terms of weak bisimulation equivalence (BNDC). This analysis revealed the events that caused non-deducibility to fail and the design was fixed by allowing this failure event to occur only when a compensating event takes place, thereby making it BNDC. Using π-calculus, non-deducibility was realized as a reduction closed barbed congruence relation, that was verified for the FREEDM system with three nodes and five nodes. The observer was able to distinguish between the

case of power migration between a pair of supply-demand nodes from the normal operation, causing basic π-ND to fail. However, it was verified that for the system with five nodes, basic π-ND is preserved.

Manual proofs were presented for all the cases studied using SPA and π-calculus. Automated tools were employed to verify the theorems presented, wherever applicable. The results obtained using the manual and automated approaches were the same, thereby justifying the correctness of the approach. Different notions of observational equivalence in π-calculus offer diverse variants of information flow models like the π-non-deducibility that are important to certain CPS infrastructures. The proposed approach makes it feasible to analyze new adversarial models and attack behaviors.

Using probabilistic and temporal reasoning, the attacker can deduce more information like the time of occurrence of the operation, duration of the operation [22], the frequency of occurrence, etc. This thought can be extended to the case where an attacker uses event history, or collaborates with other attackers to deduce critical information pertaining to the system. New mechanisms could be developed to find compensating events within the system, and schedule them, to nullify the effect of observable physical system responses such that the system preserves non-deducibility. More work is required to apply formal techniques to specify the behavior and verify information flow properties in CPSs. These aspects also present a challenge for future work in applying distributed system concepts to real systems and for the development of new paradigms for increased efficiency and reduced complexity of verification.

APPENDIX

A. MWB Encoding for a 3-node FREEDM System

```
agent DGI1(s1, s1g, state1, supply, e1) = s1(s1).t.'state1<state1>.t.'e1<supply>.
    DGI1(s1, s1g, state1, supply, e1)

agent DGI2(s2, s2g, state2, demand, e2) = s2(s2).t.'state2<state2>.t.'e2<demand>.
    DGI2(s2, s2g, state2, demand, e2)

agent PM(s1, s1g, s2, s2g) = (^z)t.'s1<s1g>.'s2<s2g>.PM(s1, s1g, s2, s2g)

agent DGI3(s3, state3, e1, e2, supply, demand) = s3(s3).t.'state3<state3>.DGI3(s3
    , state3, e1, e2, supply, demand) + e1(supply).e2(demand).DGI3(s3, state3, e1
    , e2, supply, demand)

agent PHY1(s1,s1g,bus) = t.s1(s1).'bus<s1g>.PHY1(s1,s1g,bus) + 's1<s1>.t.PHY1(s1,
    s1g,bus)

agent PHY2(s2,s2g,bus) = t.s2(s2).'bus<s2g>.PHY2(s2,s2g,bus) + 's2<s2>.t.PHY2(s2,
    s2g,bus)

agent PHY3(s3,s3g,bus) = t.s3(s3).'bus<s3g>.PHY3(s3,s3g,bus) + 's3<s3>.t.PHY3(s3,
    s3g,bus)

agent INV(s1g,s2g,s3g,bus) = 'bus<s1g>.'bus<s2g>.'bus<s3g>.INV(s1g,s2g,s3g,bus)

agent Node1(s1, state1, s1g, bus, supply, e1) = DGI1(s1, s1g, state1, supply, e1)
     | PHY1(s1,s1g,bus)

agent Node2(s2, state2, s2g, bus, demand, e2) = DGI2(s2, s2g, state2, demand, e2)
     | PHY2(s2,s2g,bus)
```

```
agent Node3(s3, state3, s3g, bus, e1, e2, supply, demand) = DGI3(s3, state3, e1,
    e2, supply, demand) | PHY3(s3,s3g,bus)

agent System(s1, state1, s2, state2, s3, state3, s1g,s2g,s3g,bus, e1, supply, e2,
     demand) = Node1(s1, state1, s1g, bus, supply, e1) | Node2(s2, state2, s2g,
    bus, demand, e2) | Node3(s3, state3, s3g, bus, e1, e2, supply, demand)| INV (
    s1g,s2g,s3g,bus)

agent System2(s1, state1, s2, state2, s3, state3, s1g,s2g,s3g,bus, e1, supply, e2
    , demand) = Node1(s1, state1, s1g, bus, supply, e1) | Node2(s2, state2, s2g,
    bus, demand, e2) | Node3(s3, state3, s3g, bus, e1, e2, supply, demand)| INV (
    s1g,s2g,s3g,bus) | PM(s1, s1g, s2, s2g)
```

B. MWB Encoding for a 5-node FREEDM System

```
agent DGI1(s1, s1g, state1, supply, e1) = s1(s1).t.'state1<state1>.t.'e1<supply>.
    DGI1(s1, s1g, state1, supply, e1)

agent DGI2(s2, s2g, state2, demand, e2) = s2(s2).t.'state2<state2>.t.'e2<demand>.
    DGI2(s2, s2g, state2, demand, e2)

agent DGI4(s4, s4g, state4, demand, e4) = s4(s4).t.'state2<state2>.t.'e4<demand>.
    DGI4(s4, s4g, state4, demand, e4)

agent PM(s1, s1g, s2, s2g) = (^z)t.'s1<s1g>.'s2<s2g>.PM(s1, s1g, s2, s2g)

agent PM(s4, s4g, s5, s5g) = (^z)t.'s4<s4g>.'s5<s5g>.PM(s4, s4g, s5, s5g)

agent DGI3(s3, state3, e1, e2, supply, demand) = s3(s3).t.'state3<state3>.DGI3(s3
    , state3, e1, e2, supply, demand) + e1(supply).e2(demand).DGI3(s3, state3, e1
    , e2, supply, demand)
```

```
13 agent DGI5(s5, state5, e1, e2, supply, demand) = s5(s5).t.'state5<state5>.DGI5(s5
       , state5, e1, e2, supply, demand) + e1(supply).e2(demand).DGI5(s5, state5, e1
       , e2, supply, demand)

15 agent PHY1(s1,s1g,bus) = t.s1(s1).'bus<s1g>.PHY1(s1,s1g,bus) + 's1<s1>.t.PHY1(s1,
       s1g,bus)

17 agent PHY2(s2,s2g,bus) = t.s2(s2).'bus<s2g>.PHY2(s2,s2g,bus) + 's2<s2>.t.PHY2(s2,
       s2g,bus)

19 agent PHY3(s3,s3g,bus) = t.s3(s3).'bus<s3g>.PHY3(s3,s3g,bus) + 's3<s3>.t.PHY3(s3,
       s3g,bus)

21 agent PHY4(s4,s4g,bus) = t.s4(s4).'bus<s4g>.PHY4(s4,s4g,bus) + 's4<s4>.t.PHY4(s4,
       s4g,bus)

23 agent PHY5(s5,s5g,bus) = t.s5(s5).'bus<s5g>.PHY5(s5,s5g,bus) + 's5<s5>.t.PHY5(s5,
       s5g,bus)

25 agent INV(s1g,s2g,s3g,s4g, s5g, bus) = 'bus<s1g>.'bus<s2g>.'bus<s3g>.INV(s1g,s2g,
       s3g,s4g, s5g, bus)

27 agent Node1(s1, state1, s1g, bus, supply, e1) = DGI1(s1, s1g, state1, supply, e1)
        | PHY1(s1,s1g,bus)

29 agent Node2(s2, state2, s2g, bus, demand, e2) = DGI2(s2, s2g, state2, demand, e2)
        | PHY2(s2,s2g,bus)

31 agent Node3(s3, state3, s3g, bus, e1, e2, supply, demand) = DGI3(s3, state3, e1,
       e2, supply, demand) | PHY3(s3,s3g,bus)
```

```
33 agent Node4(s4, state4, s4g, bus, e1, e2, supply, demand) = DGI4(s4, state4, e1,
       e2, supply, demand) | PHY4(s4,s4g,bus)

35 agent Node5(s5, state5, s5g, bus, e1, e2, supply, demand) = DGI5(s5, state5, e1,
       e2, supply, demand) | PHY5(s5,s5g,bus)

37 agent System(s1, state1, s2, state2, s3, state3, s1g,s2g,s3g,bus, e1, supply, e2,
        demand) = Node1(s1, state1, s1g, bus, supply, e1) | Node2(s2, state2, s2g,
       bus, demand, e2) | Node3(s3, state3, s3g, bus, e1, e2, supply, demand)|
   Node4(s4, state4, s4g, bus, e1, e2, supply, demand) | Node5(s5, state5, s5g, bus,
        e1, e2, supply, demand)|
39 INV (s1g,s2g,s3g,s4g, s5g, bus)

41 agent System2(s1, state1, s2, state2, s3, state3, s1g,s2g,s3g,bus, e1, supply, e2
       , demand) = Node1(s1, state1, s1g, bus, supply, e1) | Node2(s2, state2, s2g,
       bus, demand, e2) | Node3(s3, state3, s3g, bus, e1, e2, supply, demand)|
   Node4(s4, state4, s4g, bus, e1, e2, supply, demand) | Node5(s5, state5, s5g, bus,
        e1, e2, supply, demand)|
43 INV (s1g,s2g,s3g,s4g, s5g, bus) | PM(s1, s1g, s2, s2g) | PM(s4, s4g, s5, s5g)
```

BIBLIOGRAPHY

[1] E. A. Lee, "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.

[2] United States Department of Energy, "U.S.-Canada power system outage taskforce, final report on 2003 Blackout in the United States and Canada: Causes and recommendations," 2004. [Accessed Dec. 2, 2011].

[3] N. Falliere, L. Murchu, and E. Chien, "W32.Stuxnet Dossier, Version 1.4," tech. rep., *Symantec Corporation*, February 2011.

[4] P. Kerr, J. Rollins and C. Theohary, "The stuxnet computer worm: Harbinger of an emerging warfare capability," *Congressional Research Service Report for Congress*, December 2010. [Accessed Dec. 2, 2011].

[5] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *ACM Transactions on Information Systems Security*, vol. 14(1), June 2011.

[6] M. Bishop, *Computer Security: Art and Science*. Addison-Wesley, 2003.

[7] J. A. Goguen and J. Meseguer, "Security policies and security models," in *Proc. of the IEEE Symposium on Security and Privacy (SSP'82)*, pp. 195–204, IEEE Computer Society Press, 2002.

[8] J. McLean, "Security models and information flow," in *Procs. of the 1990 IEEE Computer Society Press*, pp. 180–187, IEEE Computer Society Press, 1990.

[9] P. Pederson, D. Dudenhoeffer, S. Hartley, and M. Permann, "Critical infrastructure interdependency modeling: A survey of U.S. and international research," Tech. Rep. INL/EXT-06-11464.

[10] J. Min, W. Beyler, T. Brown, Y. J. Son, and A. Jones, "Toward modeling and simulation of critical national infrastructure interdependencies," in *Institute of Industrial Engineers (IIE) Transactions*, vol. 39, pp. 57–71, Special issue on Industrial Engineering and Operations Research in Homeland Security, 2007.

[11] H. Tang and B. McMillin, "Analysis of the security of information flow in the advanced electric power grid using flexible alternating current transmission system (FACTS)," in *Critical Infrastructure Protection*, pp. 43–56, Springer, 2008.

[12] Y. Sun, B. McMillin, X. F. Liu, and D. Cape, "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), pp. 363–369, October 2007.

[13] J. McLean, "Security models," in *Encyclopedia of Software Engineering* (J. Marciniak, ed.), John Wiley & Sons, 1994.

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

[15] A. Zakinthinos and E. Lee, "A general theory of security properties," in *Procs. of the 18th IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 94–102, 1997.

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

[17] R. Akella and B. M. McMillin, "Information flow analysis of energy management in a smart grid," in *SAFECOMP* (E. Schoitsch, ed.), vol. 6351 of *Lecture Notes in Computer Science*, pp. 263–276, Springer, 2010.

[18] B. McMillin and R. Akella, "Verification and protection of confidentiality in an advanced smart grid," in *The 45th Hawaii International Conference on System Sciences*, pp. 2169 –2175, 2012.

[19] R. Akella and B. McMillin, "Modeling and verification of security properties for critical infrastructure protection," in *The 8th Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW '12) (to appear)*, 2013.

[20] R. Akella, F. Meng, D. Ditch, B. McMillin, and M. Crow, "Distributed power balancing for the FREEDM system," in *The 1st IEEE International Conference on Smart Grid Communications*, pp. 7–12, October 2010.

[21] B. McMillin, R. Akella, D. Ditch, G. Heydt, Z. Zhang, and M.-Y. Chow, "Architecture of a smart microgrid distributed operating system," in *Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES*, pp. 1–5, March 2011.

[22] R. Akella, H. Tang, and B. M. McMillin, "Analysis of information flow security in cyber-physical systems," *International Journal of Critical Infrastructure Protection*, vol. 3, no. 3-4, pp. 157 – 173, 2010.

[23] T. Gamage, R. Akella, T. Roth, and B. McMillin, "Information flow security in cyber-physical systems," in *The 7th Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW '11)*, (New York, NY, USA), ACM, 2011.

[24] R. Akella and B. M. McMillin, "Model-checking BNDC properties in cyber-physical systems," *Computer Software and Applications Conference, Annual International*, vol. 1, pp. 660–663, 2009.

[25] The Smart Grid Interoperability Panel Cyber Security Working Group, "Introduction to NISTIR 7628 guidelines for smart grid cyber security," tech. rep., National Institute of Standards and Technology, September 2010. [Accessed January 10, 2012].

[26] Reliability standards, "Standard CIP-002-3 through Standard CIP-009-4," tech. rep., North American Electric Reliability Corporation, 2012. [Accessed January 10, 2012].

[27] "A summary of control system security standards activities in the energy sector," tech. rep., Office of Electricity Delivery and Energy Reliability, Department of Energy, Washington, DC, 2005. [Accessed January 10, 2011].

[28] L. R. Phillips, M. Baca, J. Hills, J. Margulies, B. Tejani, B. Richardson, , and L. Weiland, "Analysis of operations and cyber security policies for a system of cooperating flexible alternating current transmission system (FACTS) devices," Tech. Rep. SAND-2005-7301, Sandia National Laboratory, 2005.

[29] "Final report on workshop on future directions in cyber-physical systems security," tech. rep., Department of Homeland Security, January 2010. [Accessed January 10, 2011].

[30] D. Holstein and J. Tengdin and J. Wack and R. Butler and T. Draelos and P. Blomgren, "Cyber security for utility operations," Tech. Rep. NETL Project M63SNL34, Sandia National Laboratories, Albuquerque, New Mexico, January 2005. [Accessed January 10, 2011].

[31] J. McDonald and N. Conrad and C. Service and H. Cassidy, "Cyber effects analysis using VCSE:Promoting control system reliability," Tech. Rep. SAND2008-5954, Sandia National Laboratories, Albuquerque, New Mexico, January 2008. [Accessed January 10, 2011].

[32] L. Mili, T. Cutsem, and M. Pavella, "Bad data identification methods in power system state estimation, a comparative study," in *IEEE Transactions on Power Apparatus and Systems*, vol. 103(11), pp. 3037–3049, 1985.

[33] A. Teixeira, S. Amin, H. Sandberg, K. Johansson, and S. Sastry, "Cyber security analysis of state estimators in electric power systems," in *Proc. 49th IEEE Conf. on Decision and Control (CDC'10)*, pp. 5991–5998, 2010.

[34] C. Zimmer and F. Mueller, "Time-based intrusion detection in cyber-physical systems," in *International Conference on Cyber-Physical Systems*, pp. 109–118, April 2010.

[35] A. Cardenas, S. Amin, Y.-L. Huang, Z.-Y. Lin, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proc. 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS'11)*, pp. 355–366, 2010.

[36] V. Srinivasan, J. Stankovic, and K. Whitehouse, "Protecting your daily in-home activity information from a wireless snooping attack," in *Proceedings of the Tenth International Conference on Ubiquitous Computing*, pp. 202–211, 2008.

[37] S. Amin, X. Litrico, S. Sastry, and A. Bayen, "Stealthy deception attacks on water SCADA systems," in *In Proceedings of the 13th ACM international conference on Hybrid systems: computation and control (HSCC'10)*, pp. 161–170, ACM, 2010.

[38] A. Cardenas, T. Roosta, and S. S. Sastry, "Rethinking security properties, threat models and the design space in sensor networks : A case study in SCADA systems," *Ad Hoc Networks*, vol. 7, pp. 1434–1447, May 2009.

[39] A. Cardenas, S. Amin, and S. S. Sastry, "Secure control: towards survivable cyber-physical systems," in *First International Workshop on Cyber-Physical Systems (WCPS2008)*, pp. 495–500, IEEE, 2008.

[40] D. Fitch, S. Sedigh, B. McMillin, and R. Akella, "CPS-CSH cyber-physical analysis and design," in *The 7th Conference on Critical Information Infrastructures Security (CRITIS) (to appear)*, 2012.

[41] T. A. Henzinger, "The theory of hybrid automata," in *Proceedings of the 11th Annual Symposium on Logic in Computer Science (LICS)*, pp. 278–292, IEEE Computer Society Press, 1996.

[42] R. Alur and D. L. Dill, "A theory of timed automata," *Theor. Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[43] Warren A. Hunt Jr., "Modeling and verification of cyber-physical systems," in *National Workshop on High-Confidence Automotive Cyber-Physical Systems*, April 2008.

[44] M. Pluska and D. Sinclair, "Modelling and verification of cyber-physical system," in *20th European Meeting on Cybernetics and System Research*, 2010.

[45] A. Platzer, "Integrative challenges of cyber-physical systems," in *Workshop on Usable Verification*, November 2010.

[46] C. A. R. Hoare, "Communicating sequential processes," *Commun. ACM*, vol. 21, pp. 666–677, Aug. 1978.

[47] R. Focardi and R. Gorrieri, "The compositional security checker: A tool for the verification of information flow security properties," *IEEE Transactions on Software Engineering*, vol. 23, no. 9, pp. 550–571, Sept. 1997.

[48] R. Focardi and R. Gorrieri, "A classification of security properties for process algebras," *Computer Security*, vol. 3, no. 1, pp. 5–33, 1994/1995.

[49] R. Milner, J. Parrow, and D. Walker, "A calculus of mobile processes i," *Information and Computation*, vol. 100, no. 1, pp. 1 – 40, 1992.

[50] R. Paige and R. E. Tarjan, "Three partition refinement algorithms," *Society for Industrial and Applied Mathematics Journal of Computing*, pp. 973–989, 1987.

[51] P. C. Kanellakis and S. L. Smolka, "CCS expressions, finite state processes, and three problems of equivalence," *Information and Computation*, vol. 86, no. 1, pp. 43–68, 1990.

[52] M. Abadi and A. D. Gordon, "A calculus for cryptographic protocols: The spi calculus," in *4th ACM Conference on Computer and Communications Security*, pp. 36–47, ACM Press, 1997.

[53] W. C. Rounds and H. Song, "The ϕ-calculus- a hybrid extension of the pi-calculus to embedded systems," Tech. Rep. CSE-TR-458-02, Department of Computer Science and Engineering, University of Michigan, Ann Arbor, MI, 2002.

[54] H. Jifeng, "From CSP to hybrid systems," *A Classical Mind: Essays in Honour of C.A.R.Hoare*, pp. 171–189, 1994.

[55] T. McEvoy and S. Wolthusen, "A formal adversary capability model for SCADA environments," in *Critical Information Infrastructures Security* (C. Xenakis and S. Wolthusen, eds.), vol. 6712 of *Lecture Notes in Computer Science*, pp. 93–103, Springer Berlin / Heidelberg, 2011.

[56] A. Sabelfeld and A. Myers, "Language-based information-flow security," in *IEEE Journal on Selected Areas in Communications*, vol. 21(1), pp. 5–19, 2003.

[57] D. E. Bell and L. J. LaPadula, "Secure computer systems: Mathematical foundations," Tech. Rep. MTR-2547, MITRE Corporation, Bedford, MA, 1973.

[58] D. Sutherland, "A model of information.," in *Proceedings of the 9th National Security Conference.*, pp. 175–183, 1986.

[59] A. Bossi, R. Focardi, C. Piazza, and S. Rossi, "Bisimulation and unwinding for verifying possibilistic security properties," in *Proceedings of International Conference on Verification, Model Checking, and Abstract Interpretation*, vol. 2575, pp. 223–237, 2003.

[60] A. Bossi, R. Focardi, C. Piazza, and S. Rossi, "Verifying persistent security properties," *Computer Languages, Systems & Structures*, vol. 30, no. 3-4, pp. 231–258, 2004.

[61] A. Huang, "Renewable energy system research and education at the NSF FREEDM systems center," in *Power & Energy Society General Meeting, 2009. PES '09. IEEE*, pp. 1–6, July 2009.

[62] L. M. Ni, C.-W. Xu, and T. B. Gendreau, "A distributed drafting algorithm for load balancing," *IEEE Trans. Softw. Eng.*, vol. 11, no. 10, pp. 1153–1161, 1985.

[63] "Checker of Persistent Security (CoPS)." Tool available online at `http://www.dsi.unive.it/~mefisto/CoPS/index.php`, accessed December 30, 2011.

[64] T. T. Gamage, B. M. McMillin, and T. P. Roth, "Enforcing information flow security properties in cyber-physical systems: A generalized framework based on compensation," *Computer Software and Applications Conference Workshops*, pp. 158–163, 2010.

[65] A. Dovier, C. Piazza, and A. Policriti, "A fast bisimulation algorithm," Tech. Rep. UDMI/14/00/RR, Univ. di Verona, Univ. di Udine, 2000.

[66] R. Bakhshi and D. Gurov, "Verification of peer-to-peer algorithms: A case study," *Electron. Notes Theor. Comput. Sci.*, vol. 181, pp. 35–47, June 2007.

[67] S. Crafa and S. Rossi, "A theory of noninterference for the π-calculus," in *Trustworthy Global Computing* (R. De Nicola and D. Sangiorgi, eds.), vol. 3705 of *Lecture Notes in Computer Science*, pp. 2–18, Springer Berlin / Heidelberg, 2005.

[68] M. Hennessy, "The security pi-calculus and non-interference," *Journal of Logic and Algebraic Programming*, vol. 63, no. 1, pp. 3–34, 2005.

[69] D. Sangiorgi and D. Walker, *The π-Calculus - A Theory of Mobile Processes*. Cambridge University Press, 2001.

[70] B. Victor and F. Moller, "The mobility workbench - a tool for the pi-calculus," in *Proceedings of the 6th International Conference on Computer Aided Verification (CAV)*, (London, UK, UK), pp. 428–440, Springer-Verlag, 1994.

[71] B. Blanchet, M. Abadi, and C. Fournet, "Automated verification of selected equivalences for security protocols," in *Logic in Computer Science, 2005. LICS 2005. Proceedings. 20th Annual IEEE Symposium on*, pp. 331 – 340, June 2005.

[72] "ProVerif: Cryptographic protocol verifier in the formal model." Tool available online at `http://prosecco.gforge.inria.fr/personal/bblanche/proverif/`, accessed December 20, 2012.

[73] M. Abadi and C. Fournet, "Mobile values, new names, and secure communication," in *Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '01, (New York, NY, USA), pp. 104–115, ACM, 2001.

[74] D. Dolev and A. Yao, "On the security of public key protocols," *Information Theory, IEEE Transactions on*, vol. 29, pp. 198 – 208, Mar 1983.

[75] S. Kremer and M. Ryan, "Analysis of an electronic voting protocol in the applied pi calculus," in *Proceedings of the 14th European conference on Programming Languages and Systems*, ESOP'05, (Berlin, Heidelberg), pp. 186–200, Springer-Verlag, 2005.

VITA

Ravi Chandra Akella is originally from Andhra Pradesh, India. He completed his primary schooling from Little Angels High School, Visakhapatnam, India. He 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). His Master's thesis was titled "Information Flow Properties For Cyber-physical Systems." He was awarded a Ph.D. degree in Computer Science in May, 2013 by Missouri University of Science and Technology for his work on "Verification Of Information Flow Security In Cyber-physical Systems." His Master's and Ph.D. research was conducted under the guidance of Dr. Bruce McMillin. His doctoral studies were supported by a National Science Foundation Scholarship for his work with Free Renewable Electric Energy and Management Systems (FREEDM) Center. His major interests are in distributed systems, security, formal methods and cyber-physical systems.