Fall 2007

Security analysis of a cyber-physical system

Han Tang

Department:

SECURITY ANALYSIS

OF A

CYBER-PHYSICAL SYSTEM

by

HAN TANG

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN COMPUTER ENGINEERING

2007

Approved by

_____	_____
B. McMillin, Advisor	M.L. Crow

_____

A. Miller

ABSTRACT

Cyber-Physical Systems (CPSs) are an integration of computing and physical processes. Information flow is an inherent property of CPSs and is of particular interest at their cyber-physical boundaries. This thesis focuses on discovering information flow properties and proposes a process to model the information flow in CPSs. A Cooperating FACTS Power System serves as a tangible example to illustrate modeling information flow using the proposed process. The proposed process can be used to model the information flow security, help analyze current information flow security requirements, and aid in the design of further security policies in a CPS.

ACKNOWLEDGMENTS

The author would like to express her gratitude to her advisor Dr. Bruce McMillin for his guidance, valuable advice and encouragement. Without his kind help and understanding, the author couldn't have gotten this far. Also, the author would like to thank thesis committee members Dr. Mariesa Crow and Dr. Ann Miller for their advice and participation in the defense, and to all the faculty and students in the Power Research Group at University of Missouri Rolla, led by Dr. Crow and Dr. McMillin, for those numerous instructive discussions and kind support of the research work. The author would also like to take this chance to thank Dr. Miller for her illuminating explanations of the security area during her classes.

Finally, the author would like to thank her husband Hai Lan and friends Cheryl and Randy Scott and Cathy and Alex Primm for their help and encouragement. With their help, the author can study hard and live happily in the peaceful town of Rolla.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Page

LIST OF TABLES

Page

NOMENCLATURE

Symbol	Description
ES	Event System
NF	Noninference security property
ND	Nondeducible security property
BNDC	Bisimulation-based Non Deducibility on Composition security property
Tr	System event traces

1. INTRODUCTION

Cyber-Physical Systems (CPSs) are integrations of computation with physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops, where physical processes affect computations and vice versa [18]. In the physical world, the events occur in real-time so discrete event clocks cannot be stopped to create a consistent state and concurrency is intrinsic. However, computing and networking technologies currently do not take those into consideration well. CPS applications include high confidence medical devices and systems, traffic control and safety, advanced automotive systems, process control, energy conservation, environmental control, avionics, instrumentation and critical infrastructure control systems (such as electric power, water resources, and communications systems). Besides inherited physical interactions and their concurrent computation nature, CPSs are usually network-centric systems [25].

Various issues in the study of CPSs need to be addressed. This thesis focuses on the security aspect of the CPS. Among the various security issues dealing with confidentiality, integrity and availability, this thesis focuses on the confidentiality of CPSs, especially on information flow security. The physical nature of a CPS tends to expose information flow through actions at the *cyber-physical boundary*.

Many CPSs consist of similar elements. In the Cooperating FACTS Power System (CFPS), an intelligent controller communicates with other intelligent controllers and makes decisions via distributed decision making. In the CFPS, an intelligent controller sits on lines of an electric power system to balance the power flow of the entire power system. Throughout this thesis the CFPS is used as the example to identify and model the information flow in a CPS. The CFPS serves as a real world example to show the applicability of the proposed process.

The family of Flexible AC Transmission System (FACTS) devices are power electronic-based controllers that can rapidly inject or absorb active and reactive power, thereby affecting power flow across transmission lines; a FACTS device changes the amount of power owing on a particular power line. The use of FACTS devices in a power system can potentially overcome limitations of the present manually/mechanically

controlled transmission system [3]. A FACTS Device (depicted in Figure 1.1) consists of an embedded computer that depends on a low voltage control system for signal processing, which, in turn, depends on a low and a high voltage power conversion system for rapidly switching power into the power line. Each FACTS device controls one power line (ControlledLine) and multiple FACTS devices interact with each other via exchanging messages over a network (Communication). The net effect of the FACTS devices and the power grid is that each power line and FACTS device is affected by other power lines and FACTS devices.

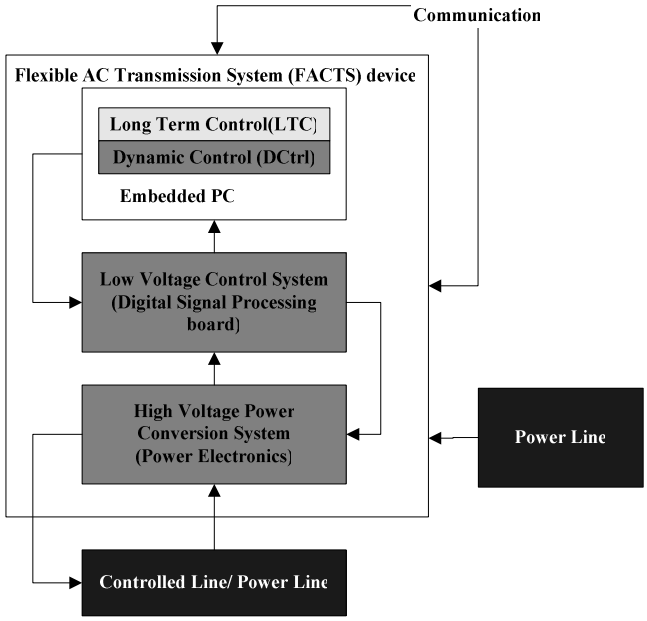

Figure 1.1 A FACTS device

The Unified Power Flow Controller (UPFC) device is a type of FACTS device [3][28] that can modify active power flow on a power line. In this thesis, the FACTS devices refer to the UPFC devices.

FACTS devices are primarily used when a cascading failure occurs within a Power System; one or more lines are lost due to a downed line or overloaded line and the resulting redirected power flow stresses the network. Too much power may flow over

lines of inadequate capacity and one-by-one the lines overload and trip out until a large portion of the Power System has failed [3]. FACTS device coordination is required to prevent cascading failures [1][3]. The FACTS devices themselves communicate over an interconnected computing network to reach agreement on how power should be routed or re-routed in the presence of a contingency. These Cooperating FACTS Devices (CFD) working together in the electric power network form the CFPS [28]. The FACTS devices behave autonomously, but they depend on information received from their participation in the CFPS to determine their responses. The CFPS uses a distributed maxflow algorithm [1] to rebalance power flow, which is done in the Long Term Control (LTC), running on different processors that are located in different UPFC devices to compute the decision and manipulate the power network by sending the power settings to Dynamic Control. The Dynamic Control then sets the Power Electronics to enforce the local power flow to an expected value which redistributes power flow at a regional or wider level within the power network. The LTC and Dynamic Control both sit in the Embedded PC as a portion of a FACTS device (shown in Figure 1.1). Each FACTS device must continually monitor not only its own behavior in response to system operating changes, but the response of neighboring devices as well.

Distributed computing management is different from a traditional centralized power network management system; the CFD manipulates the whole CFPS in a decentralized way, so that new security issues emerge. In [28], a broad investigation into the operational and security challenges that the CFDs face has been discussed. A general security analysis of FACTS has been given in the report which includes vulnerability of CFD and some available good practices based on those used for SCADA systems. An agent-based security framework has been suggested, while multiple levels of FACTS devices security issues and the confidentiality, integrity and availability of the electric power grid have been briefly analyzed. However, no approach has been proposed nor any concrete example described in the confidentiality of CFPS.

The North American Electric Regulatory Corporation (NERC) provides a basis to define permanent cyber security standards [34]. These provide a cyber security framework to identify and assist with the protection of Critical Cyber Assets to ensure reliable operation of the Bulk Electric System. Those requirements, stated in Standard

CIP-002-1 to CIP-009-1, address various security issues and require approaches to provide security in the Bulk Power System.

This thesis identifies the vulnerability of information flow in a CPS from analyzing the example system's execution sequence. After analyzing the potential information flow of the CPS, a process is proposed to model the information flow security to provide secure computing in the CPS. Finally, automatic checking tools are applied to check system behavior against the developed security property to prove the system's security.

2. BACKGROUND

2.1 INFORMATION FLOW SECURITY

A security model is used to describe any formal statement of a system's confidentiality, integrity and availability requirements [23]. Using information flow, principals can infer properties of objects from observing system behavior [32]. This is a potential hazard in the cyber-physical world so it requires more attention. To be more specific, inferring confidential information from the observable information flow is a potential source of critical information leakage; the information flow of CFPS needs to be carefully analyzed. Various security models that analyze multi-level security system behavior from the access control or execution sequence perspective have been discussed for decades to address the information flow problems of a system in the defense community. However, most of the related publications [21] [22] [23] [24] [27] have not been directly applied to CPSs. One of the reasons security models are less popular outside the defense area is due to the complexity. Considering the significance of the confidential information in critical infrastructure, it is worth introducing these models to address the information flow in the security analysis of critical infrastructure. Figure 2.1 shows a partial taxonomy of the security models discussed in [24]. Those models in grey have been used in this work to analyze the security of CFPS.

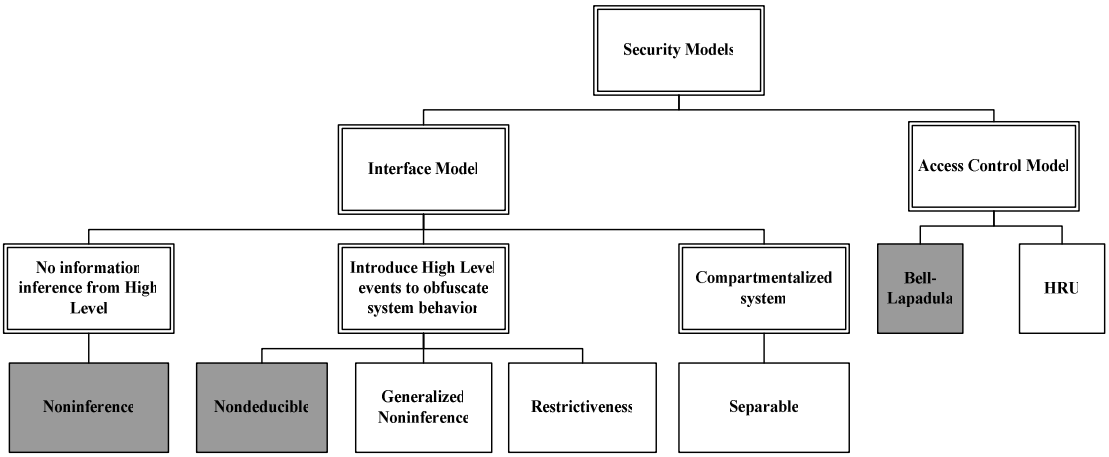

Figure 2.1 Partial taxonomy of the security models in [24]

Before defining the security models that has been used throughout this thesis, Table 2.1 is a list of convention:

Table 2.1 Convention used in formal description throughout this thesis

Symbol	Meaning
Tr	System traces
τ	A system trace
$\backslash_x$	System purge all traces in the domain of x
$E_1 \mid E_2$	Parallel composition of event E_1 and E_2
H	High-level security domain
L	Low-level security domain
I	Inputs
O	Outputs

2.1.1 Noninference Model. A system is considered secure if and only if for any legal trace of system events, the trace results from the legal trace purged of all high-level events is still a legal trace of the system [23][24][27].

$$NF(ES) \equiv \forall \tau \in Tr : \tau \backslash_h \in Tr \qquad (1)$$

Here, in order to make the security property easier to understand, an imaginary problem modeled after delivering pizzas to the Pentagon is constructed, the Pentagon-pizza shop example. There is a high-level set of events (experts arrive) that are supposed to be secret and a set of low-level events in which a pizza shop cooks and the Pentagon disposes of pizza. The events are depicted in Figure 2.2. The notation of system events are borrowed from [21]. A solid line refers to a low-level event and a dotted line refers to the high-level event.

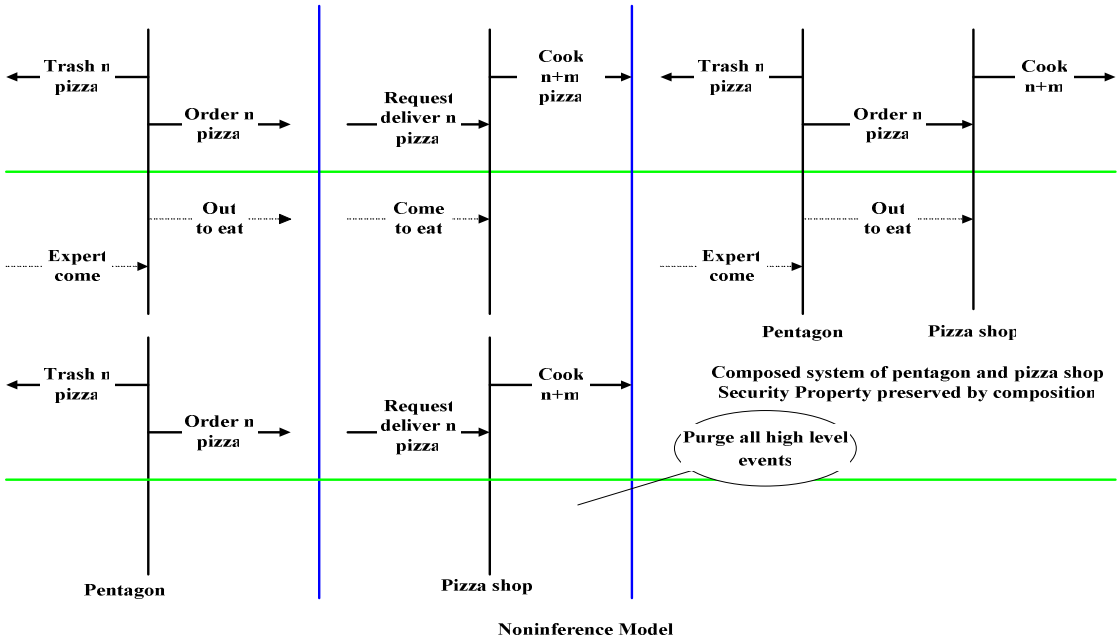

Figure 2.2 Pentagon-pizza shop example for noninference security property

Shown in Figure 2.2 are two systems, namely the Pentagon and the pizza shop. Each one has legal (allowed) sequences as follows:

$$Pentagon : \{Expert\_Come \mid Out\_to\_eat \mid Order\_n\_pizza \mid Trash\_n\_pizza,$$
$$Order\_n\_pizza \mid Trash\_n\_pizza\}$$
$$Pizza\_shop : \{Come\_to\_eat \mid deliver\_n\_pizza \mid cook\_n+m\_pizza,$$
$$deliver\_n\_pizza \mid cook\_n+m\_pizza\}$$

If only consider the Pentagon system, the high-level events are Expert come and the number of the people who go out_to_eat, the low-level events are $Order\_n\_pizza$ and $Trash\_n\_pizza$. From a more substantiative point of view, if the Pentagon trashes regular numbers of pizza boxes everyday, these low-level events happen no matter what the high-level events are and the observers will not be able to infer if there are any high-level events (like any experts coming to Pentagon who require ordering pizza). If a system shares the same property as this Pentagon system, it satisfies

the noninference security property as described in (1). The pizza shop is another example that satisfies the noninference security property for the same reason that purging the high-level events leaves the low-level (observable) events unchanged.

2.1.2 Nondeducible Model. A system is considered nondeducible secure if it is impossible for a low-level user, through observing visible events, to deduce anything about the sequence of inputs made by a high-level user. In other words, system is nondeducible secure if the low-level observation is compatible with any of the high-level inputs. [21][23][24]

$$ND(ES) \equiv \forall \tau_L, \tau_H \in Tr : \exists \tau \in Tr : \tau \setminus_h = \forall \tau_L \cap \tau |_{H \cap I} \qquad (2)$$

The Pentagon-pizza shop example is also used here (shown in Figure 2.3) to illustrate the nonduducible security property. In this figure, the possibility that the composed system doesn't satisfy the nondeducible property is illustrated as well.

In Figure 2.3, Pentagon and Pizza shop are still used as the systems to illustrate the nondeducible security property. Each of the system has legal (allowed) sequence as following:

$$Pentagon : \{Expert\_Come \mid Out\_to\_eat \mid Order\_lunch \mid Even\#,$$
$$Out\_to\_eat \mid Order\_lunch \mid Odd\#\}$$
$$Pizza\_shop : \{Come\_to\_eat \mid Cook\_lunch \mid Odd\#,$$
$$Cook\_lunch \mid Even\#\}$$

If the Pentagon system is considered in isolation from the low-level observation, the observer should not be able to infer Even# and Odd# are introduced by either 0,1 or more Expert come events. Any system sharing the same property as the Pentagon system, in which the low-level observation is compatible with any of the high-level inputs, satisfies the nondeducible security property defined in (2). However, the composability of the nondeducible security property needs to be pointed out as shown on the right side of Figure 2.3. Although the Pentagon system and the pizza shop system satisfy the nondeducible security property individually, when composed together, the composed system no longer satisfy the nondeducible security property since, when the observer

observes Even# from one side and Odd# from the other side, s/he will infer that there must be some high-level event(s) that caused the difference.

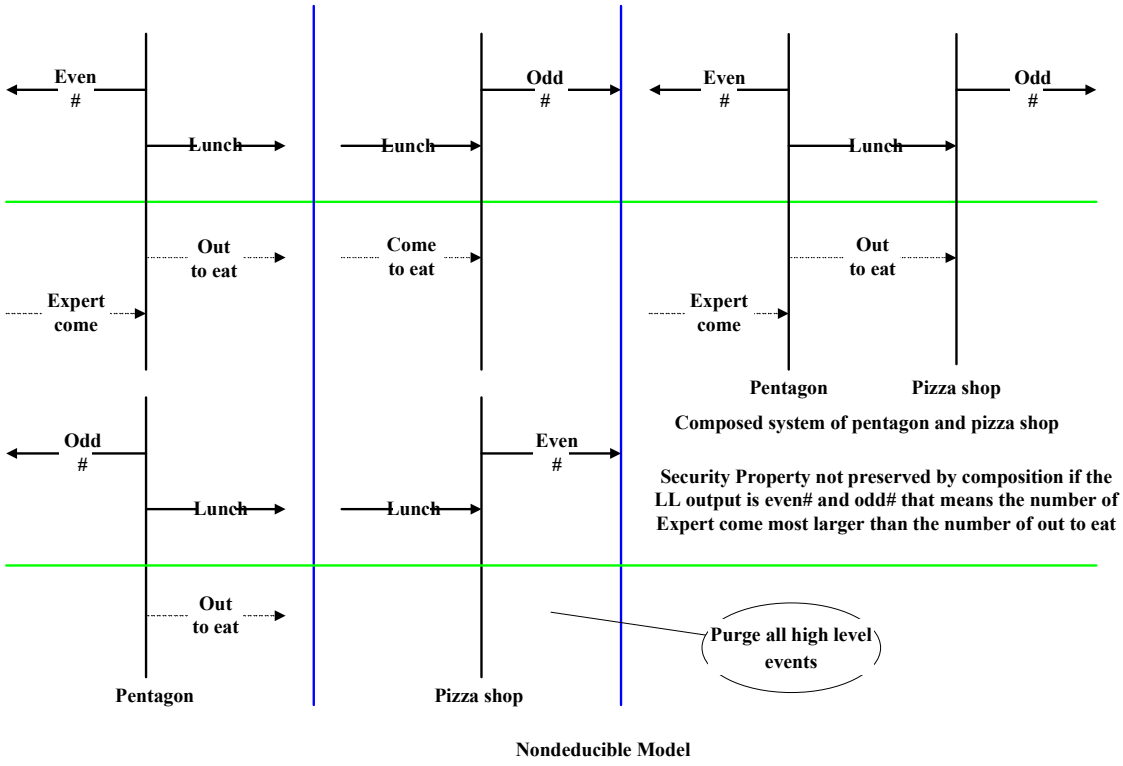

Figure 2.3 Pentagon-pizza shop example for nondeducible security property

2.1.3 Bisimulation-based Nondeducibility on Composition Model. A system is considered to have the Bisimulation-based Non-Deducibility on Composition (BNDC) property, if it can preserve its security after composition. [6][8] A system ES is BNDC if for every high-level process P, a low-level user cannot distinguish ES from $(ES \mid P) \setminus Act_H$ (ES composed with any other process P and purged high-level events). In other words, a system ES is BNDC if what a low-level user sees of the system is not modified by composing any high-level process P to ES.

Formally BNDC can be defined as: ES is BNDC if and only if

$$BNDC(ES) \equiv \forall \Pi \in \mathrm{E}_H, ES / \tau_H \approx_B (ES / \Pi) \setminus \tau_H \quad (3)$$

Note: here ES / τ_H means turn all the high-level events in ES to internal events. BNDC can be illustrated with a very similar Pentagon-pizza shop example as in Figure 2.3 by adding an internal event that leads to a high-level output. In this case, the system can be composed with any other system but from the observation point of view (bisimulation), the system satisfies the BNDC property.

2.1.4 Bell-LaPadula Model. Different from those security models mentioned above, the Bell-LaPadula model is an access control model which offers more tangible security rules that can be enforced during execution. In the Bell-LaPadula model [2], all entities are divided into subjects and objects. Subjects are active entities, while objects are passive containers for information. The Bell-LaPadula model sets up rules for untrusted subjects:

- Untrusted subjects may only read from objects of lower or equal security level
- Untrusted process may only write to objects of greater or equal security level

2.1.5 Applicability. The CFPS system fits within the multi-level security structure. To analyze the information flow of CFPS more effectively, the security models defined above are used. The noninference property might be too strong in some systems where the low-level inputs result in high-level outputs. However, the noninference model can be applied in this information flow analysis for the principle components of UPFC devices because no low-level input results in high-level outputs in the systems being analyzed. The nondeducible security property is used to analyze the system where high-level outputs are observable. According to [21], if an entire system is nondeducible secure, then no low-level user of that system will ever learn any high-level information through the system. The BNDC security model has the advantage that if systems satisfy the BNDC property, they are composable. Furthermore, the BNDC is compatible with noninference and nondeducible security properties. The Bell-LaPadula model is used to illustrate how vulnerabilities are introduced in other perspectives besides the interface models.

2.2 SECURITY PROCESS ALGEBRA (SPA) AND PERSISTENT SECURITY PROPERTY CHECKING TOOL – COPS

In order to formalize the security models described in last section, this thesis uses security process algebra (SPA) to formalize the behavior of the system and uses CoPS as an automatic tool to check the system's security property against security properties that can be checked by CoPS.

2.2.1 SPA. Security Process Algebra (SPA, for short) is an extension of Calculus of Communicating Systems (CCS) [26] - a language proposed to specify concurrent systems, that defines algebra consisting of operators for building systems using a bottom-up approach from smaller subsystems. The basic building blocks are atomic activities, called actions; unlike CCS, in SPA, actions belong to two different levels of confidentiality, thus allowing the specification of multilevel (actually, two-level) systems. The BNF Syntax of SPA to describe the system is [9]:

$$E ::= 0 \mid \mu.E \mid E_1 + E_2 \mid E_1 \mid E_2 \mid E \setminus L \mid E \setminus_I L \mid E / L \mid E[f] \mid Z$$

where 0 is the empty process, which cannot do any action; $\mu.E$ can do action μ and then behaves like E; $E_1 + E_2$ can alternatively choose to behave like E_1 or E_2; $E_1 \mid E_2$ is the parallel composition of E_1 and E_2, where the executions of the two systems are interleaved, $E \setminus L$ can execute all the actions E is able to do, provided that they do not belong to $L \cup \overline{L}$; $E \setminus_I L$ requires that the actions of E do not belong to $L \cap I$; E / L turns all the actions in L into internal τ's; if E can execute action μ, then $E[f]$ performs $f(\mu)$; finally, Z does what E does, if $Z \stackrel{def}{=} E$.

As an example of using SPA, consider an imaginary system, ES, that leaks information from a high-level security entity to the low-level. ES has no constraints on read or write (output or input) sequences; the system behavior can be described as:

$$ES = Action \mid Object(high, y) \mid Object(low, y) \setminus N;$$
$$Action = read(l, x).val(z) + write(l, y).W(y, z) \setminus N$$
$$Object(x, y) = \overline{R}(x, y).Object(0, init) + W(x, y).Object(x, y)$$

where N is the event set that ES does not allow. In the above description: Object refer to any security entities and it has parameter l which could be *high* or *low* to indicate the security level of object and parameter y to indicate the current status of y (in this example, a value is used to indicate the current state). *read* and *write* refer to the action that this system allowed. $\overline{R}$ and W refer to the real final output of reading result or the input of writing result.

Here is one possible sequence that leaked the information:

$$(Action \mid Object(0,0) \mid Object(1,5) \setminus N$$
$$\xrightarrow{read(0,1)} (read(0,1).\overline{R(1,5)} \mid Action \setminus N$$
$$\xrightarrow{write(0,1)} (write(0,0).W(0,5) \mid Action) \setminus N$$
$$\xrightarrow{\tau} (Object(0,5) \mid Object(1,5) \mid Action) \setminus N$$
$$\xrightarrow{read(0,0)} (R(0,0).Val(0,5) \mid Action) \setminus N$$

This sequence can be interpreted as: a low level ($l = 0$) object read the high level ($l = 1$) object and get its status ($y = 5$) and write it to itself ($l = 0, y = 0 \rightarrow 5$), later any low level object can read this low level object and get the status ($y = 5$) which leaks the information.

2.2.2 CoPS. CoPS is an automatic checker of multilevel system's security properties [20]. In particular, CoPS checks the three security properties: Bisimulation-based Non-Deducibility on Composition (BNDC), Strong Bisimulation-based Non-Deducibility on Composition (SBNDC) and, Persistent BNDC (P BNDC) [6] [7] [8]. These are Non-Interference properties [24] which imply the Bisimulation-based Non-Deducibility on Composition [6] [8]. In this case, the CoPS is chosen to check the modeled behavior of CFPS to see if it satisfies the BNDC which is compatible with the noninference and nondeducible security properties.

The SPA discussed in the last section can be converted to code that is compatible with CoPS syntax and checked automatically by CoPS against security properties that reorganized in CoPS. The conversion takes several steps as:

CoPS has keywords as shown in Table 2.2.

Table 2.2 Keywords defined by CoPS

Keyword	Meaning
bi	Bind (agent) identifier
basi	Bind action set identifier
acth	Bind an action set to Act_H, the high level actions

- Identify security objects (defined as agent in CoPS using keyword *bi*)
- Identify objects' actions (defined as action set in CoPS using keyword *basi*)
- Classify security levels to each action and clarify high-level actions (defined as high-level actions in CoPS using keyword *acth*)
- Rewrite the system behavior with above identified items

In order to illustrate the syntax of CoPS the small imaginary system used in the last section is written into code that CoPS can interpret as following:

```
bi ES
  (Action |Obj_l0 | Obj_h5)\L

bi Action
  read_ll.rl0.'val_l0.Behavior +
  read_hh.rh5.'val_h5.Behavior +
  read_lh.rl5.'val_h5.Behavior +
  read_hl.rl0.'val_l0.Behavior +
  write_ll.'wl0.Behavior +
```

```
  write_lh.'wh0.Behavior +
  write_hl.'wl5.Behavior +
  write_hh.'wh5.Behavior

bi  Obj_l0
 'rl0.Obj_l0 + wl0.Obj_l0 + wl0.Obj_l5

bi  Obj_l5
 'rl5.Obj_l5  + wl5.Obj_l0 + wl5.Obj_l5

bi  Obj_h0
 'rh0.Obj_h0 + wh0.Obj_h0 + wh0.Obj_h5

bi  Obj_h5
 'rh5.Obj_h5 + wh5.Obj_h0 + wh1.Obj_h5

basi L
  rh0  rh5  rl0  rl5
  wh0  wh5  wl0  wl5

basi N
 val_h0 val_h5
 val_l0 val_l5
 read_hh read_hl read_lh read_ll
 write_hh write_hl write_lh write_ll


acth
 val_h0 val_h5
 rh0 rh5 wh0 wh5 read_hh
 write_hh write_hl
```

With the above code, the CoPS checks the behavior of the described system and finds it does not satisfy any recognized security properties, such as the BNDC. This is the same as the result in last section.

In the remainder of this thesis, information flow in CPSs will be discovered by using the SPA discussed to model system behavior and codes are written to check system behavior against security properties that are defined in CoPS. The later analysis of information flow problems resulting from system behavior is very similar to the small example discussed in this chapter.

3. INFORMATION FLOWS IN CYBER-PHYSICAL SYSTEM

Cyber-Physical Systems (CPS) are integrations of computation with physical processes. The embedded computers and networks used to monitor and control the physical processes, usually include feedback loops where physical processes affect computations and vice versa[5]. The cyber and physical interactions have the potential to leak information from the system to the outside world. In this section, the CFPS is used as an example to illustrate possible information flow in a CPS.

Lack of confidentiality of information flow can have catastrophic effects. As an example, consider an instance of the IEEE 118 bus system [3][19]. This is a highly stressed system with many lines near overload. There are critical lines that, if removed, will cause cascading failures throughout the system. From the analysis in [3][19], if line 4-5 is removed, line 5-11 will be overloaded and be tripped later, then line 7-12 will be overloaded and tripped, then other lines will be overloaded and lead to a cascading failure. If attackers know these critical lines together with a good guess of line capacity, they can carry out an effective attack causes a cascading failure of the system simply by physically removing a critical line. The confidential information leaked by information flow will assist or accelerate the attackers.

3.1 DEFINING INFORMATION FLOW IN CFPS

In the CFPS, decisions are made cooperatively and distributively. The decision-making information is what needs to be kept confidential. The internal settings and control operations of a single FACTS device or the CFDs are defined as confidential in [28]. Current work follows their definition of confidential information (as shown in Table 3.1, adapted from Table 2 in [28]) to analyze the information flow in the CFPS.

The CFPS is made up of 3 security levels (shown in Table 3.2). In the high-level domain, communication is done by the Long Term Control. In the medium-level domain, the Dynamic Control and Power Electronics have implicit communication with other FACTS devices. At the low-level security domain, the settings of the power line cause implicit communication in the power network. The implicit communication is done when the power setting of ControlledLine(s) is changed and the whole system's power flow

redistributes correspondingly as shown in Figure 3.1. This kind of communication is due to the interconnected nature of power networks. Failure of confidentiality in the system is defined as leakage of higher level (including the high-level and medium-level security domain) information, such as internal settings and control operations, to the low-level security domain.

Table 3.1 Confidential information in CFPS

Data	**Type**	**Source**	**Function**
Dynamic Control Feedback	Digital	Dynamic Control	Obtain and pass computed changes to prevent oscillations
Data Exchange with CFD neighbors	Analog and Digital (Ethernet)	Neighbor CFD	Data necessary to implement distributed max flow algorithm
Control	**Type**	**Source**	**Function**
Control Exchange with CFD neighbors	Digital (Ethernet)	Neighbor CFD	Information necessary for cooperative agreement on CFD changes

Table 3.2 Security levels in Cooperating FACTS Power System

Security	**Security entities**	**Reason**
High-level	Long Term Control Parameters of CFPS	Contains critical information for distributed control algorithm and calculated settings with a global view of the power grid
Medium-level	Dynamic Control DSP board Power Electronics	Contains settings received from high-level security entity and will generate local settings according to local control algorithms
Low-level	ControlledLine Local power network	Open access to some power lines or easy to obtain knowledge of part of the power grid

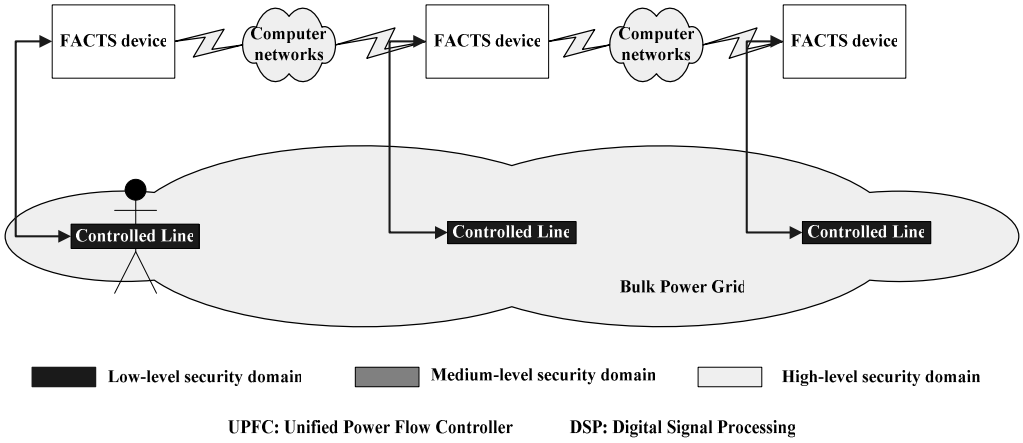

Figure 3.1 Architecture of CFPS

In order to demonstrate the information flow clearly, following assumptions are made:

Assumption 1: The message send by LTC is legitimate and correct. (The security of LTC itself is not taken into consideration in current work.)

Assumption 2: The communication network which the LTCs used to pass the maxflow algorithm messages is secure. In other words, the communication between LTCs located in different UPFC devices is considered to be secure.

Assumption 3: The power flow information of entire power network is secure, although some single power lines can be measured or a local topology is observable.

Assumptions 1 and 2 define the problem scope of this paper, which is confined to investigate the security of system information flow but not other security issues such as active attacks including maliciously changing the settings. Assumption 3 is made to analyze the system's information flow with the basic information that the possible attackers could find.

3.2 FINDING THE INFORMATION FLOW IN CFPS

A bottom-up approach is used to find and analyze the information flow of CFPS. The CFPS is decomposed to the level of single components which are used to aggregate the UPFC device. The information flow is analyzed at the component level first, then

those components are composed to build UPFC device. The information flow at the UPFC device level is further investigated to reflect the security of the system.

3.2.1 Information Flow of the Components in the UPFC. The principal components of a UPFC device which include the LTC, Dynamic Control, DSP board and Power Electronics are depicted in Figure 1.1. The information flow of a UPFC device is shown in Figure 3.2, where each component is considered a security entity. Figure 3.3 illustrates the information flow of the principle components building a UPFC device using the pictorial notation for the traces as introduced in [21]. Here, horizontal vectors represent inputs to and outputs from the system. The broken line represents the higher level events and the solid line represents the low-level events.

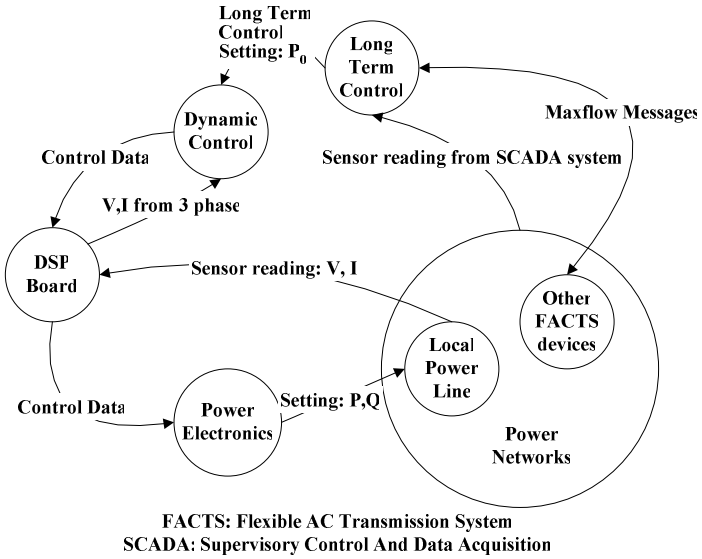

Figure 3.2 Information flow diagram of UPFC devices

A series of lemmas regarding the components of the UPFC device are proved as following. These are used to prove the property of noninference and other security properties of the composed system in later theorems.

3.2.1.1 DSP board. **Lemma 1,** the DSP operation is noninference secure.

Proof: Seen from Figure 3.3, the DSP board is a non-deterministic system which is built up from traces of the following form:{ { },e_1,e_3,e_4,e_1e_2,e_1e_3,e_1e_4,e_3e_4, $e_1e_2e_3$,

$e_1e_2e_4$,$e_1e_3e_4$,$e_1e_2e_3e_4$, …}(… stands for any interleavings of listed traces in the system), where e_1 is a Low-level Input (LI) event; e_2 is a High-level Output (HO) event; e_3 is a High-level Input (HI) event and e_4 is a HO event. This system satisfies the definition of noninference [24][25][27] because purging any legal trace of events not in low-level security domain, the result will either be e_1 or {} which are both legal traces of the system, i.e., DSP Board system itself is a noninference secure system where no information flows from the high level security domain interfere with (the interference used in this paper refer to the events from other domain than the observer belongs to, that can be observed by the observer) the low level security domain. ■

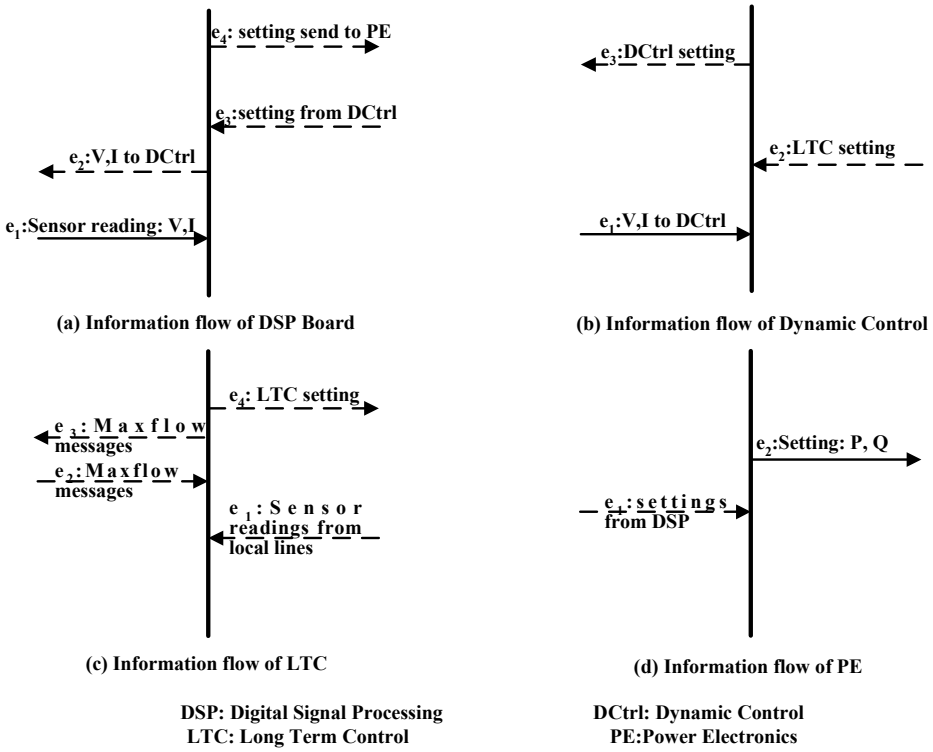

Figure 3.3 Information flow of principle components of UPFC

3.2.1.2 Dynamic Control. **Lemma 2,** the Dynamic Control operation is noninference secure.

Proof: the Dynamic Control system is a non-deterministic system, shown in Figure 3.3(b), that contains traces of the following form: {{},e_1,e_2,e_1e_3,e_1e_2,e_2e_3,$e_1e_2e_3$, …}, where e_1 is a LI event, e_2 is a HI event and e_3 is a HO event. When project any legal trace to the low-level security domain or purge any events that not in the low level security domain, the result will be either e_1 or {}, which are also legal traces. Therefore, the Dynamic Control system satisfies the noninference security model. ■

3.2.1.3 Long Term Control (LTC). The LTC system, which is a non-deterministic system shown in Figure 3.3(c), where all the events are high-level events. It's obvious that there is no interference between high-level security domain and the lower level security domain in LTC system. In other words, there is no information flow out of the high-level security domain. Proving this in the perspective of information flow is trivial.

3.2.1.4 Power Electronics. Lemma 3, the Power Electronics operation is not noninference secure.

Proof: the Power Electronics event system, shown in Figure 3.3(d), simply contains traces: {{}, e_1, e_1e_2, …}. When project any legal traces to the low-level security domain, the result will be either e_2 or {}, where e_2 is not a legal trace in this system. i.e., the power electronics system is not noninference secure. In this system e_1(HI) infers e_2(LO), which means if e_2 happens e_2 must happen before. ■

The causal relationship between e1 and e2 is where the information has been downgraded and passed to the lower security domain. This system is not secure not only in the perspective of interface models, but also in the view of access control models such as the Bell-LaPadula model [2] since there is information classified as higher level has been written to the low level domain, which violates the second rule of the Bell-LaPadula model.

3.2.2 Information Flow of the Composition of Components into the UPFC. The UPFC device is able to work only when all the components mentioned above compose together and work properly. In this section, the composed UPFC devices will be discussed with and without considering the internal events respectively. After the components are composed to form the UPFC device, the information flows between

components inside UPFC device are internal information flows (shown in Figure 3.4) and others are externals (shown in Figure 3.5).

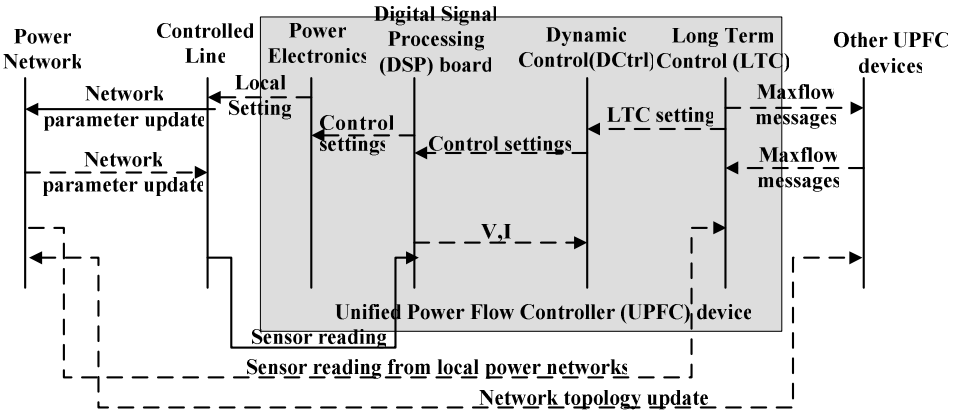

Figure 3.4 Information flow analysis at UPFC device level – internal and external flow

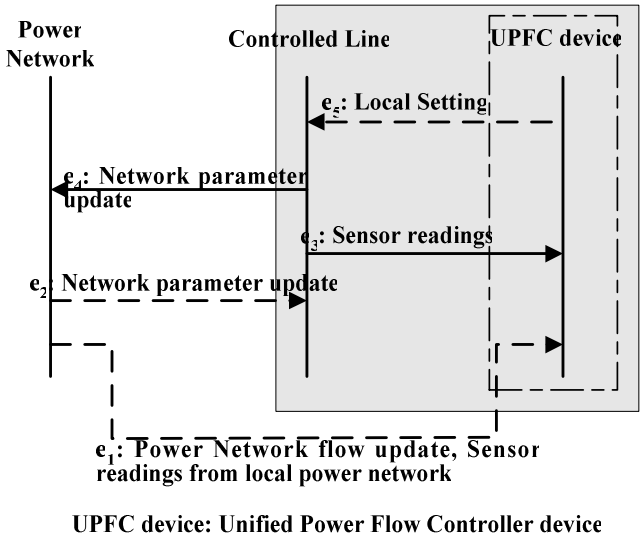

Figure 3.5 Information flow analysis at UPFC device level – external flow only

Theorem 1, Considering the external events only, the composition of DSP, Dynamic Control, LTC and Power electronics forming the UPFC device is noninference secure.

Proof: From Lemma 1, 2, the DSP and Dynamic Control are noninference secure. Connecting DSP and the Dynamic Control with the LTC, it is still noninference secure. The result of Lemma 3 does not invalidate the noninference secure property of these components composed with power electronics. Observing Figure 3.5 and taking the UPFC device without considering the internal events, it is a non-deterministic system that contains traces {{},e_1,e_3,e_5, e_1e_3, e_1e_5,e_3e_5,$e_1e_3e_5$, …}(The composed system's boundary is at UPFC device as shown in Figure 3.5). The projection of these external events traces for the UPFC device to the low-level domain is either {} or e_3 which are legal traces (the only observable low-level event – the sensor reading event can happen without the occurrence of any higher level events). That means the UPFC device, considering only the external events, is a noninference secure system. The UPFC device is noninference secure so that attackers cannot infer the higher level behavior simply from observing low-level events. ■

This noninference secure property proved in Theorem 1 is achieved without observation of power flow, in other words, the system boundary under consideration is the UPFC device itself but not the ControlledLine linked to the UPFC device. Since the attacker usually will not be able to attack the UPFC device itself due to the physical protection such as those required by CIP-006-1, the system boundary can stop at the ControlledLine. Usually the ControlledLine is more prone to be attacked due to its physical nature of open access.

Theorem 2, the system constructed of the UPFC device connected with the ControlledLine is nondeducible secure.

Proof: Observing the event system at ControlledLine from Figure 3.4, the system contains traces {{}, e_1e_4, e_2e_4, $e_1e_2e_4$, …}, where e_4 is LO event, both e_1 and e_2 are HI events. This system is not noninference secure because the projection of the legal trace to the low level domain ({e_4}) is not a legal trace. However, the system with the boundary at the ControlledLine satisfies nondeducible security property [24][25][27], because every high level input (either e_1, e_2 or both e_1 and e_2) are compatible with the low level output (e_4). ■

As shown in Figure 3.4, the changes of ControlledLine can be affected by the local settings from Dynamic Control or by the other LTC settings that propagate through

the power network. Even more, it could be affected by the topology change of power lines (such as a line trip), which triggers the redistribution of the power flow for the system. That is to say, by only observing the events interfering with the ControlledLine, no clue of where the information is from can be formed.

That the UPFC device (with the boundary at ControlledLine) satisfies the nondeducible security model seems to be a very favorable result, even during building the UPFC devices, a component which is not secure (as from Lemma 3 where the Power Electronics downgrades the information to a low-level domain), the system is still secure considering the external information flow interference. From the interface model point of view, the system is secure such that no confidential information is exposed through information flow. In the real system, however, the ControlledLine is observable, and this introduces a new vulnerability.

3.2.3 Information Flow at the Cyber-Physical Boundary. Given the results of previous sections, is this system really secure considering other types of inference? By measuring power flow in or out of the UPFC device, can the high-level actions be deduced? Due to the nature of the electric power network, its physical infrastructures are exposed outside and prone to be attacked easily. Taking the UPFC device as an example and considering only passive attacks such as attaching meters to measure the line voltage and current parameters, it is possible that these measured data could help to calculate the settings from the control devices of the Power System and infer the control operation accordingly. With a passive attack of using meters attached to the ControlledLine and with a reasonable amount of computation the "settings" of UPFC devices can be calculated with the computation model shown in Figure 3.6.

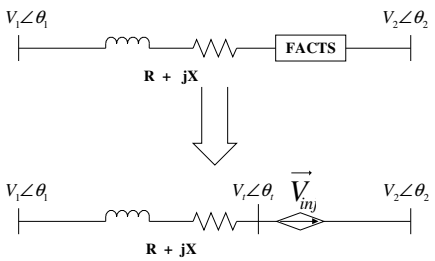

Figure 3.6 Computation model of ControlledLine and the FACTS devices

Theorem 3, the UPFC settings can be deduced by computation with the low-level observation.

Proof: In Figure 3.6, if take two measurement of three-phase instantaneous voltage and current information at both sides of the UPFC device ($V_t\angle\theta_t$ and $V_2\angle\theta_2$), using Kirchhoff's law, the injected voltage $\overrightarrow{V_{inj}}$ can be solved. The settings of UPFC from the Dynamic Control can be further calculated if $\overrightarrow{V_{inj}}$ is known. This means the local settings can be observed (compromised) even with the information flow analysis that has been done in previous paragraphs. ■

In summary, the selected CPS has information flow out of the system at the cyber-physical boundary. A proper way to catch and model this information flow needs to be addressed. In next section, a process to model the information flow in a CPS is proposed. The CFPS is used as example to illustrate the process.

4. PROPOSED PROCESS TO MODEL CPS'S INFORMATION FLOW

4.1 PROCESS DESCRIPTION

As expressed in the last section, the can be leaked to the outside through cyber-physical interactions. A process is proposed to model the information flow of a CPS.

The process of modeling information flow includes early steps of (1) eliciting security requirements by the misuse case and identifying nonfunctional requirements that tightly couple with the security requirements, (2) intermediate steps such as applying security models and modifying the models to suit the particular system, and (3) final steps of formally describing the system and checking system behavior against security properties. The entire process for modeling the information flow in a large system is shown in Figure 4.1. The ultimate goal of this thesis is to propose a feasible and effective process that can serve as a baseline to model the information flow security of a large CPS.

To illustrate the process and show its suitability for CPS, the CFPS continues to serve as the example. In the following sections, each step in this process is explained first in general, then corresponding work is done with the CFPS.

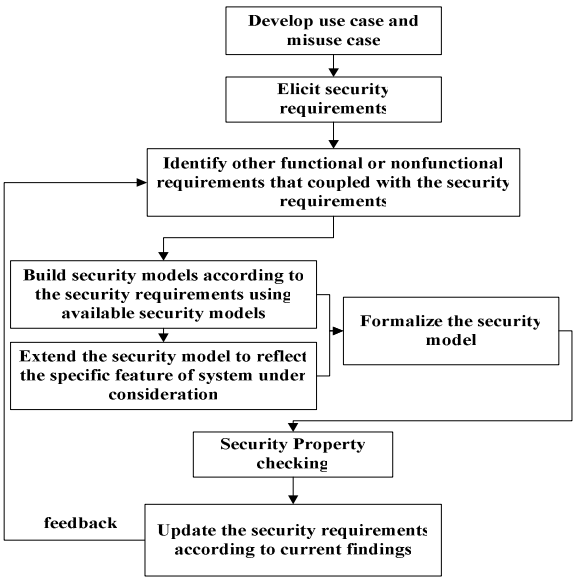

Figure 4.1 Process of modeling information flow security in Cyber-Physical System

4.2 STEPS OF THE INFORMATION FLOW MODELING PROCESS AND EXAMPLE OF CFPS

The process of modeling information flow security includes the following steps: security requirements elicit the misuse case, specify other non-functional requirements that have the potential to couple with the security requirements, analyze the elicited requirements using available security models and SPA, extend or modify the security model to adapt to the security of information flow; apply automatic checking.

4.2.1 Requirement Elicitation. Misuse case is used to elicit the requirement for securities as the first step of modeling the information flow of Cyber-Physical System.

4.2.1.1 Misuse case. A misuse case is the inverse of a use case [11][12][13] i.e., a function that the system should not allow. A use case is defined as a completed sequence of actions which gives increased value to the user. One could define a misuse case as a completed sequence of actions which results in loss for the organization or some speci_c stakeholder. A mis-actor is parallel to an actor, i.e., an actor who does not want the system to function, an actor who initiates misuse cases.

4.2.1.2 Misuse case of CFPS system. As mentioned, the misuse case can be used to describe the system's undesired behavior. Figure 4.2 is a diagram that uses the concept of misuse case and mis-actor to illustrate the information flow of the FACTS system. A current misuse case is shown in Figure 4.2, developed from group discussions by the Power Research Group at the University of Missouri, Rolla. However, other techniques, such as attack trees, can also be used to aid the generation of misuse cases to a system.

From Figure 4.2, it can be found that the use cases in the rectangle with the broken line are fundamental to both passive and active attackers. From Table 4.1 to Table 4.3, the same conclusion can be drawn namely, that the integrity and availability of the system is not independent of the confidentiality. Current work focuses on the confidentiality of the system. As shown in Table 4.1, SR 1.1.1, SR 1.2.1 and SR 1.3.1, physical protection to the device and the medium, needs to be applied. This thesis focuses only on the security requirement of the information flow of this system, which is mostly concerned with confidentiality (in Table 4.1).

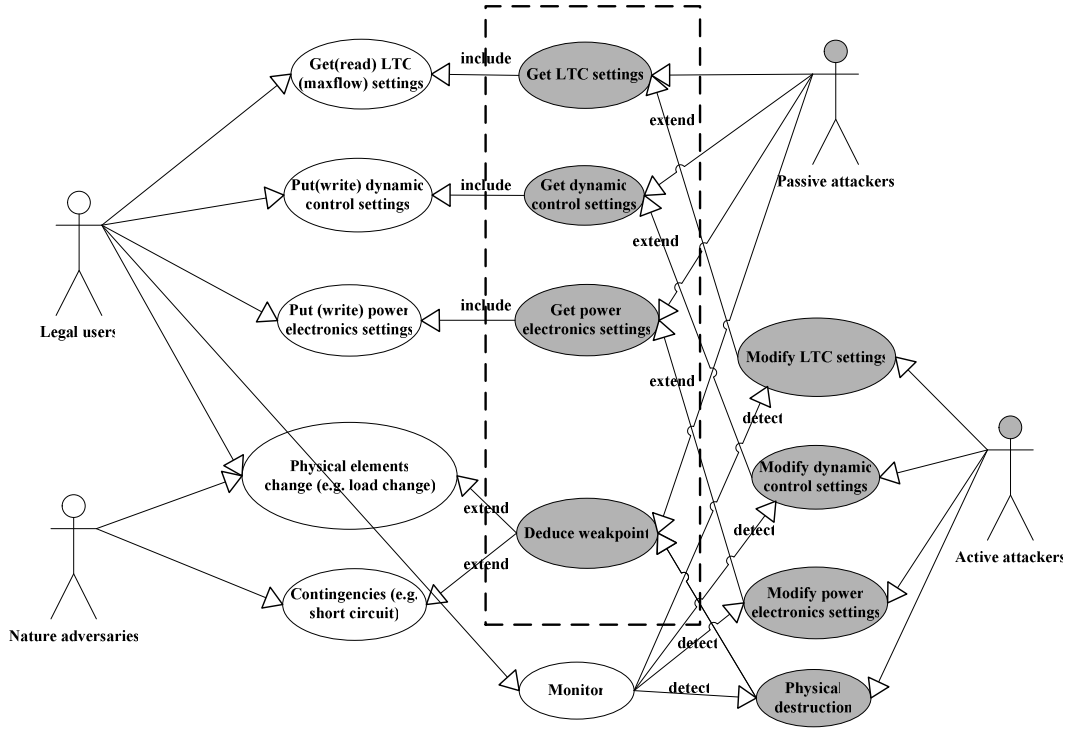

Figure 4.2 Misuse case of Flexible AC Transmission System

With the misuse case shown in Figure 4.2, some security requirements are elicited by considering the unveiled possible attacks, as shown in Tables 4.1, 4.2, and 4.3.

Table 4.1 Requirements for integrity

Security Requirements (Integrity): -
SR 2.1: The LTC's settings can not be changed
SR 2.2: The dynamic control(DCtrl)'s settings can not be changed
SR 2.3: The power electronics (PE)'s settings are confidential

Table 4.2 Some requirements for availability

Security Requirements (Availability): -
SR 3.1: Critical devices need physical protection and hardware backup

Table 4.3 Requirements for confidentiality

Security Requirements (Confidential): -
SR 1.1: The LTC's settings are confidential
SR 1.1.1: Physical protection to LTC and the media that the settings are sent through
SR 1.1.2: The LTC's control settings are confidential
SR 1.1.3: The LTC's control operation are confidential
SR 1.2: The dynamic control (DCtrl)'s settings are confidential
SR 1.2.1: Physical protection to DCtrl and the media that the settings are sent through
SR 1.2.2: The DCtrl's control settings are confidential
SR 1.2.3: The DCtrl's control operation are confidential
SR 1.3: The power electronics (PE)'s settings are confidential
SR 1.3.1: Physical protection to PE and the media that the settings are sent through
SR 1.3.2: The PE's control settings are confidential
SR 1.3.3: The PE's control operation are confidential
SR 1.4: No weak operation point of system can be deduced

4.2.2 Identify the Functional and Non-functional Requirements Behind the Misuse Cases. Identify the functional and non-functional requirements that couple with the current security requirement is important to achieve a complete specification of the security requirement. Table 4.4 shows a sample of the timing requirements of the CFPS. The system's information flow security cannot be achieved without other functional and non-functional requirements working properly.

The current process of finding the coupling of functional and nonfunctional requirements with the security requirement is by excluding those requirements that are not related to the security requirements. In practice, any requirement that affects the same system parameters or system states will be considered as coupling with the security requirements that have been identified. This is not an effective strategy, as it covers many functional and nonfunctional requirements. However, it is worthy in the design and

analysis phase of the critical infrastructure. The strategy of purging the non-security related requirements can be changed and investigated in the future.

Table 4.4 Sample of nonfunctional requirements[28][35]

Requirements: -
…
Real time constrains:
R x.1 The LTC's update rate of 10s
R x.2 The dynamic control (DCtrl)'s update rate is 1ms
R x.3 The power electronics (PE)'s update rate is 0.33s (300Hz)
R.x.4 The load change rate is 20ms (50Hz)
…

4.2.3 Security Analysis Using Available Security Models. Figure 4.3 shows the interaction between the FACTS device and the power system. Currently, the power system is modeled and represented by a simulation engine, which simulates an IEEE 118 bus power system. Attackers are also shown in Figure 4.3. However, only the passive attackers have been considered in modeling the system information flow security. In Figure 4.3 both the FACTS device and the Simulation Engine are high-level objects. However, the ControlledLine(s) are considered to be low-level objects due to their open physical nature.

Here, the analysis of the FACTS system's information flow contains two parts which are similar to those discussed in Section 3.2.2. The analysis is done at two security boundaries, one is at the physical boundary of the FACTS device and the other makes the ControlledLine the security boundary since the ControlledLine is more or less an opened line. The information flow is as shown in Figure 3.5, in Section 3.2.2.

The CFPS system is a nondeterministic system; noninference and nondeducible are the two security models that can be used to do a static check for the information flow. Two conclusions from Section 3.2.2 are listed here and will be analyzed next.

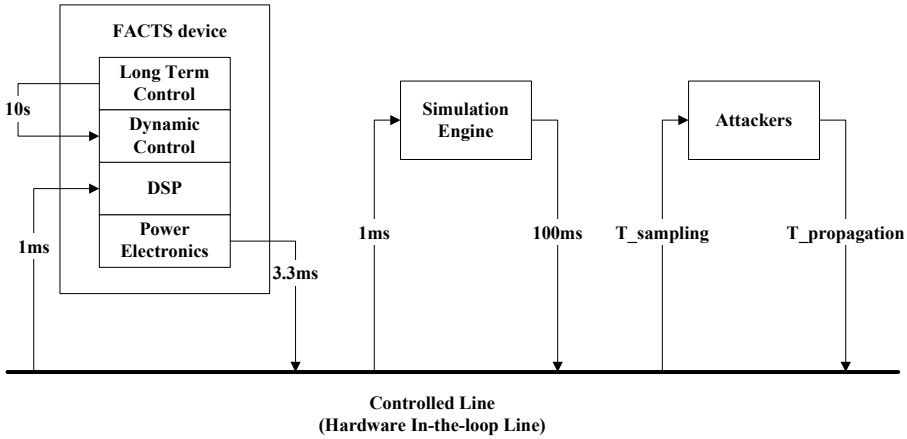

Figure 4.3 FACTS system interaction

- Conclusion 1: The UPFC device is noninference secure if taking the UPFC's physical boundary as the security boundary
- Conclusion 2: The UPFC device is nondeducible secure if taking the ControlledLine as the security boundary

The analysis from the events point of view has been given in Section 3.2.2. Here, in this step of the proposed process, the focus is on formal analysis using the SPA.

4.2.3.1 Security analysis of conclusion 1 using the noninference security model. A formal model can be applied to analyze Conclusion 1. Table 4.5 shows all the events that are allowed at the security boundary of UPFC devices.

Table 4.5 Events and allowed access

Events	Type	Implication
e_1	High-level subject (Power network) writes to high-level object (UPFC device)	High-level subject (UPFC device) reads from high-level object(Power network)
e_3	Low-level subject (ControlledLine) writes to high-level object (UPFC device)	Low-level subject (ControlledLine) reads from low-level object (local lines)
e_5	High-level subject (UPFC device) writes to low-level object (ControlledLine)	High-level subject (UPFC device) reads from high-level object (UPFC device)

Although Table 4.5 lists only the allowed events, the formal requirements should be able to capture both the illegal events and the invalid events. Equation (4) describes the behavior of the FACTS system if taking the physical boundary of the UPFC as the security boundary. The notion and value-passing SPA can be found in [15][16][27]. The analysis below follows the procedure that is described in Section 2.2.

$$UPFC\_Device_{no\_time} = (Behavior1 \mid Object(0, P_{CL}) \mid Object(1, P_{init})) \setminus L$$

$$Behavior1 = M\_read(l, x).(if\ (l == x)$$
$$then$$
$$r(x, y).Val(l, y).Behavior1)$$
$$else$$
$$Behavior1 + M\_write(l, x).(if\ (l >= x).then$$
$$write(l, z).\overline{W}(x, z).Behavior1$$
$$else$$
$$if\ (x == 1)then$$
$$write(l, z).\overline{W}(1, z).Behavior1$$
$$elseBehavior1$$

$$Object(x, y) = \overline{R}(x, y).Object(0, P, t) + W(x, y).Object(x, y) \qquad (4)$$

Here $M\_read(l, x) / M\_write(l, x)$ stand for events that subject of security level l read/write to an object of security level x. y and z are the values (or states) of the object. The above SPA describes the system behavior and possible executions.

Additional steps will be taken using the automatic checking tools to testify the above SPA described system satisfies predicates defined as the noninference security property, which is formalized in equation (1).

The FACTS system behavior can be shown as an access monitor for the UPFC devices, which is depicted in Figure 4.4.

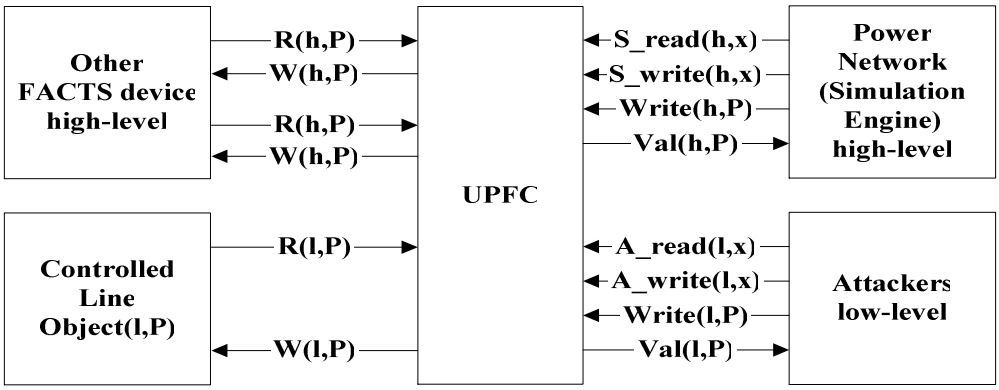

Figure 4.4 UPFC device security boundary at devices physical boundary

4.2.3.2 Formalize the security analysis of conclusion 2 shown in last section.

Similarly, the information from Conclusion 2 is formalized. Table 4.6 shows all the events that are allowed at the security boundary of ControlledLine.

Table 4.6 Events and allowed access

Events	Type	Implication
e_1	High-level subject (Power network) writes to high-level object (UPFC device)	High-level subject (UPFC device) reads from high-level object (Power network)
e_2	High-level subject (Power network) writes to low-level object (ControlledLine)	High-level subject (Power network) reads from high-level object (Power network)
e_4	Low-level subject (ControlledLine) writes to high-level object (Power network)	Low-level subject (ControlledLine) reads from low-level object (local lines)

The SPA to describe the CFPS which takes the security boundary at the ControlledLine is very similar to the behavior of the CFPS with the security boundary at UPFC device level. The SPA is shown as follows:

$$UPFC\_ControlledLine_{no\_time} = (Behavior2 \mid Object(0, P_{CL}) \mid Object(1, P_{init})) \setminus L$$

$$Behavior2 = M\_read(l, x).(if\ (l == x)then$$
$$r(x, y).Val(l, y).Behavior2)$$
$$elseBehavior2$$
$$+ M\_write(l, x).(if\ (l >= x)then$$
$$write(l, z).\overline{W}(x, z).Behavior2$$
$$else$$
$$if\ (x == 1)then$$
$$write(l, z).\overline{W}(1, z).Behavior2$$
$$elseBehavior2$$

$$Object(x, y) = \overline{R}(x, y).Object(0, P, t) + W(x, y).Object(x, y) \quad (5)$$

The system behaviors can be shown as Figure 4.5

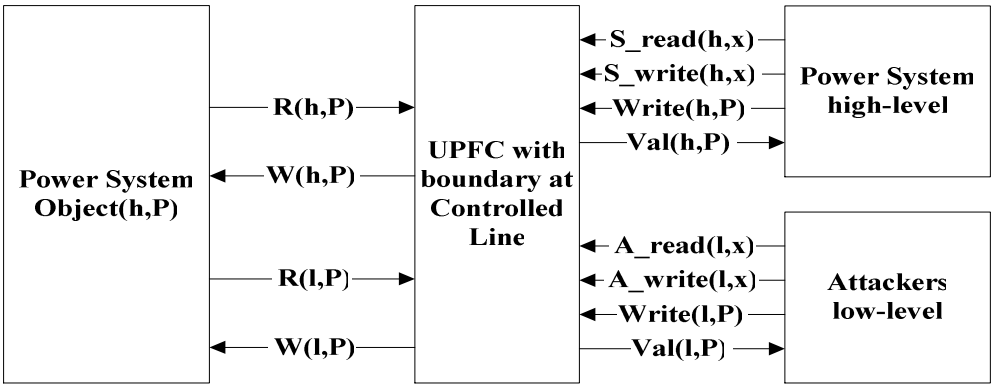

Figure 4.5 UPFC device security boundary at ControlledLine

From Section 3.2.2, intuitively, the FACTS system, which has a security boundary at the ControlledLine, satisfies the nondeducible security model. Here the SPA defined in this section needs to check against the nondeducible security model as defined in equation (2).

With the above formal descriptions of both the FACTS behavior at the boundary of ControlledLine and the nondeducible property, the automatic property checking tools are ready to be applied to prove the security property of the FACTS system.

4.2.4 Beyond the Available Security Models. The security requirements are easy to couple with other kinds of requirements such as nonfunctional requirements, e.g. performance requirements (CPU burst can be encoded as '1' and CPU low usage can be encoded as a '0', which can make a covert channel). Various kinds of nonfunctional requirements can be coupled with the security requirements. This phenomenon occurs frequently in the cyber-physical world. In this case a security model that contains pure security considerations might only reflect one side of the problem. In order to add more perspectives to the problem, the security models selected to analyze the information flow are changed to include information about other requirements.

In the CFPS, the security requirement of information security has the potential of coupling with the real-time requirement of the system. However, the security models that are widely used do not always consider real-time or temporal behavior of the system. The analysis in the previous section, which uses the current available security models, cannot illustrate the possible security issues involving these temporal aspects. The system behavior with timing is shown in Figure 4.6.

Observe Figure 4.6, if the attacker passively attaches power flow meters to the low-level object (ControlledLine in the FACTS system) to log the line flow data, the attacker could observe some significant changes of the line flow at certain time intervals and infer the system update rate. For example, the following data gives a glimpse of a line flow log. Here, the data are based on lab data which is aiming at testing the load change and the FACTS device's response.

From this trace (shown in Table 4.7), it can be seen that the attacker gathers the line flow information every 5ms. In other words, it has a sampling rate of 200Hz. Observing the change rate of the line flow, the attacker can infer that after a significant line flow change (at 190505ms), at least every 5ms, there is a change that causes the line flow to drop. However, around every 100ms, the line flow will be balanced back to a higher setting. Knowledgeable attackers could start a brief analysis of the power system based on acquired information:

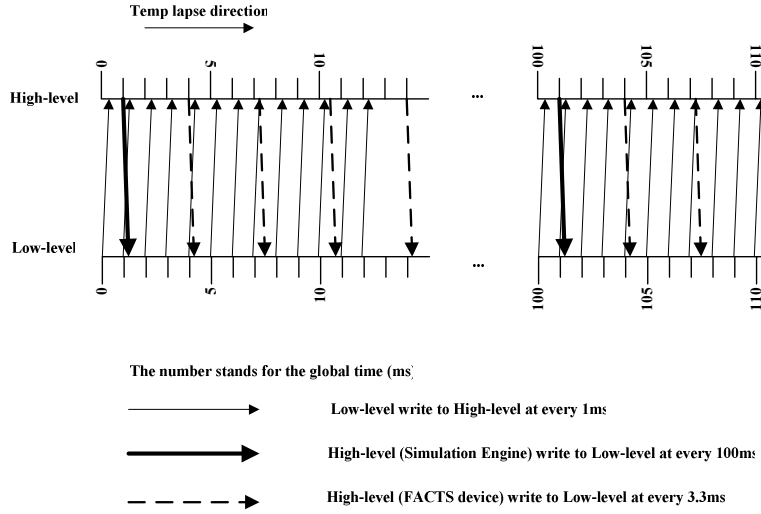

Figure 4.6 Intuitive analysis of system behavior with temporal consideration

- 190505 ms, some contingency happens (location not yet known) that causes the ControlledLine to have a flow change of around 20%
- At least every 5ms, the line flow drops by 2%, which means there is something withdrawing power flow from the ControlledLine at least every 5ms
- At least every 100ms, the line flow is changed by 6%, which means there is some other mechanism injecting power flow to the ControlledLine at least every 100ms

With the above observation and some guess work, the attacker obtains knowledge about the system response time with the FACTS device on, which is around 5-100ms.

The above analysis regarding the system's behavior, with temporal constraints taken into consideration, is based on some lab experience. A formal description needs to be given in order to use a model checking tool to prove the correctness of the security of information flow with timing considerations. Some literature [7][8][15] was introduced ways of adapting time in the security model. The security models built in Section 4.2.3 are also modified to reflect the temporal constraints of the system and show whether the coupling of nonfunctional requirements such as the real-time requirement, in this case, have violated the security requirement or not.

Table 4.7 Timestamped observation of ControlledLine

Time(ms)	Line flow (pu)
150000	-0.34248
150005	-0.3425
150010	-0.34254
…	-0.34252
…	-0.34252
	…
190505	-0.42768
190510	-0.42064
190515	-0.41765
190520	-0.41056
	…
190610	-0.42059
190625	-0.41751
190630	-0.41038
190635	-0.40723
	…

As in [8] and [15], time is represented by a tick to describe the system's time in a discrete manner according to the global clock. (e.g. system = write. . .system), where internal events will always follow write events and take a unit of time. In the current approach, to include the temporal constraints in the SPA, the FACTS system's behavior is chosen by extending the value passing SPA by one more value, the time interval. The line flow change observation is based on the information of ControlledLine, so the security boundary of the FACTS device was set to the ControlledLine. In the previous section, system behavior observed at ControlledLIne was found to be nondeducible secure. With temporal constraints, can a similar conclusion be reached?

$$UPFC\_ControlledLine_{time} = (Behavior2_t \mid Object(0, P_{init}, t) \mid Object(1, P_{init}, t)) \setminus L$$

$$Behavior2_{time} = M\_read(l, x, t).(if\ (l == x)then$$
$$R(x, y, t).Val(l, y, t).Behavior2_{time})$$
$$elseBehavior2_{time}$$
$$+ M\_write(l, x, t).(if\ (l >= x)then$$
$$Write(l, z, t).\overline{W}(x, z, t).Behavior2_{time}$$
$$else$$
$$if\ (x == 1)then$$
$$write(l, z, t).\overline{W}(1, z, t).Behavior2_{time}$$
$$elseBehavior2_{time}$$

$$Object(x, P, t) = \overline{R}(x, P, t).Object(x, P, t) + W(x, P', t).Object(x, P', t) \quad (6)$$

Figure 4.7 shows the CFPS behavior with timing constraints. After the formal expression of the system's execution sequence and the temporal constraints, the models can be used to feed in the model checking tools. As seen from the informal analysis, the conclusion has been drawn that the real-time constraints do affect the security properties. In this case, the security requirement on information flow needs to be updated (as shown in Table 4.8) with the real-time constraints to reflect the situation.

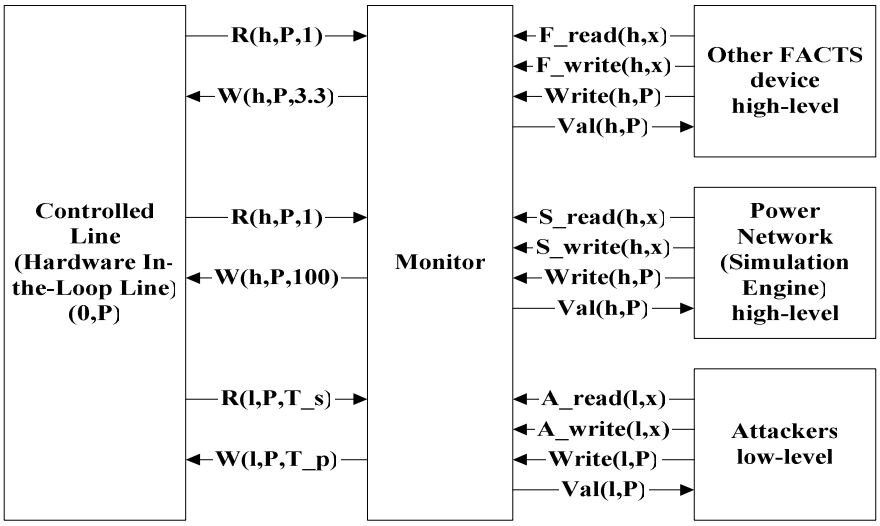

Figure 4.7 Behavior of FACTS considering timing constraints

Table 4.8 System requirement for confidentiality

Security Requirements (Confidential): -
SR 1.1: The LTC's settings are confidential
SR 1.2: The dynamic control (DCtrl)'s settings are confidential
SR 1.3: The power electronics (PE)'s settings are confidential
SR 1.4: No weak operation point of system can be deduced
…
Updated: SR 1.*: System operation time can not be deduced

4.2.5 Apply the Automatic Checking Tools. Applying model checking tools to the security models that are developed for the system is a significant step to prove the correctness of the current security requirements and to find new security needs based on the results of checking. In this thesis, the effort is mostly spent on preparing formal descriptions for current system behavior and the security models that can be fed to the selected checker or some other security property checking tools. However, if the security properties can be formalized as to which kinds are suitable for any model checking tools, those security properties can also be checked by available model checkers[28][29] other than CoPS. The following section will discuss formalizing the security properties identified in this section and feeding it to CoPS to get the result.

5. RESULTS

One of the most significant points in the proposed information flow modeling process for CPS is that the modeling process is not only aimed at describing the information flow model but also at providing a strategy to check the available model so that the result can be fed back to improve the security of a system at design time. The security property modeled following that process needs to be checked when the models are formalized. In this section, a persistent security property checking tool is applied to do the automatic checking. The correctness of the selected security models used to define the CPS is checked. The results from this formal checking can either prove the security of current CPSs or be valuable feedback to be added to or modify the security requirements of the system. As mentioned earlier, the SPA was chosen to formalize the security property and CoPS is chosen as the automatic formal security property checking tool. The security models described using SPA in Sections 4.2.3 and 4.2.4 were modified to be compatible with the CoPS syntax in this section and then fed to CoPS to get the result.

5.1 USING SPA TO DEFINE THE CFPS WITHOUT CONSIDERATION OF TIMING INFORMATION

Before considering any timing information in the CFPS, the system's behaviors modeled in Section 4.2.3 using SPA are rewritten using syntax provided by CoPS.

5.1.1 Security Boundary at UPFC Device Level. According to Conclusion 1 the system satisfies the noninference security property [27] considering the UPFC system's security boundary at the UPFC device level [36]. The system behavior is defined using SPA in Section 4.2.3 as shown in equation (4). Here system behavior is further modified to satisfy the syntax of CoPS as shown in Table 5.1 and fed into the CoPS to check against the security property of BNDC.

//this simulation is for the security boundary at UPFC device level
//without considering any timing issues. here value 0 means initial
//value, 1 means it could be set to a new value
bi UPFC_NT

```
(UPFC | LTC)\N
bi UPFC
(Behavior | HIL_h0 | HIL_l0)\L
bi Behavior
access_r_ll.(rl0.'val_l0.Behavior + rl1.'val_l1.Behavior) +
access_r_hh.(rh0.'val_h0.Behavior + rh1.'val_h1.Behavior) +
access_w_lh0.'wh0.Behavior +
access_w_lh1.'wh1.Behavior +
access_w_hl0.Behavior +
access_w_hl1.Behavior +
access_w_hh0.'wh0.Behavior +
access_w_hh1.'wh1.Behavior
bi HIL_l0
'rl0.HIL_l0 + wl0.HIL_l0 + wl1.HIL_l1
bi HIL_l1
'rl1.HIL_l1 + wl0.HIL_l0 + wl1.HIL_l1
bi HIL_h0
'rh0.HIL_h0 + wh0.HIL_h0 + wh1.HIL_h1
bi HIL_h1
'rh1.HIL_h1 + wh0.HIL_h0 + wh1.HIL_h1
bi LTC
a_r_hh.'access_r_hh.(val_h0.'put_h0.LTC + val_h1.'put_h1.LTC ) +
a_w_hh0.'access_w_hh0.LTC +
a_w_hh1.'access_w_hh1.LTC
basi L
rh0 rh1 rl0 rl1
wh0 wh1 wl0 wl1
basi N
val_h0 val_h1
val_l0 val_l1
access_r_hh access_r_hl access_r_lh access_r_ll
```

access_w_hh0 access_w_hh1 access_w_hl0 access_w_hl1
access_w_lh0 access_w_lh1 access_w_ll0 access_w_ll1
acth
a_r_hh a_r_hl
a_w_hh0 a_w_hh1 a_w_hl0 a_w_hl1
put_h0 put_h1
val_h0 val_h1
rh0 rh1 wh0 wh1 access_r_hh access_r_hl
access_w_hh0 access_w_hh1 access_w_hl0 access_w_hl1

5.1.2 Security Boundary at the ControlledLine Level. The ControlledLine is easier to attack compared to the UPFC device at the device boundary due to the physical security protection of the system. The UPFC system's security boundary is extended to the ControlledLine. Section 3.2.2 shows the UPFC system, taking the security boundary at the ControlledLine, and satisfying the nondeducible security property. The system's behaviors are described using SPA in Section 4.2.3 as shown in equation (5).

The above model has been converted into codes that are compatible with CoPS syntax as shown in Table 5.2. Those codes will be checked against the BNDC property. If this model satis_es the BNDC property, that means, the UPFC system can be composed with any other system that also satis_es BNDC to build a larger system.

//this simulation is for the security boundary at ControlledLine level
//without considering any timing issues
//here value 0 means initial value, 1 means it could be set
//to a new value
bi CL_NT
(CL | LTC)\N
//here consider the LTC objects and the internal events brought by LTC
bi CL
(Behavior | HIL_h0 | HIL_l0)\L

```
bi Behavior
access_r_ll.(rl0.'val_l0.Behavior + rl1.'val_l1.Behavior) +
access_r_hh.(rh0.'val_h0.Behavior + rh1.'val_h1.Behavior) +
access_w_lh0.'wh0.Behavior +
access_w_lh1.'wh1.Behavior +
access_w_hl0.Behavior +
access_w_hl1.Behavior +
access_w_hh0.'wh0.Behavior +
access_w_hh1.'wh1.Behavior
bi HIL_l0
'rl0.HIL_l0 + wl0.HIL_l0 + wl1.HIL_l1
bi HIL_l1
'rl1.HIL_l1 + wl0.HIL_l0 + wl1.HIL_l1
bi HIL_h0
'rh0.HIL_h0 + wh0.HIL_h0 + wh1.HIL_h1
bi HIL_h1
'rh1.HIL_h1 + wh0.HIL_h0 + wh1.HIL_h1
bi LTC
a_r_hh.'access_r_hh.(val_h0.'put_h0.LTC +
val_h1.'put_h1.LTC ) +
a_w_hh0.'access_w_hh0.LTC +
a_w_hh1.'access_w_hh1.LTC
basi L
rh0 rh1 rl0 rl1
wh0 wh1 wl0 wl1
basi N
val_h0 val_h1
val_l0 val_l1
access_r_hh access_r_hl access_r_lh access_r_ll
access_w_hh0 access_w_hh1 access_w_hl0 access_w_hl1
access_w_lh0 access_w_lh1 access_w_ll0 access_w_ll1
```

acth

a_r_hh

a_w_hh0 a_w_hh1 a_w_hl0 a_w_hl1

put_h0 put_h1

val_h0 val_h1

rh0 rh1 wh0 wh1 access_r_hh

access_w_hh0 access_w_hh1 access_w_hl0 access_w_hl1

5.2 USING SPA TO DEFINE THE CFPS WITH CONSIDERATION OF TIMING INFORMATION

Various researchers have worked on theoretical information flow property analysis for several years. However, the uniqueness of this work is in using a tangible example, the CFPS system, to illustrate the security properties that are developed from the theory. Furthermore, this work extends the model to consider the physical nature of the system. The physical nature of the system cannot be ignored since that is how the system works and some of the inherited physical nature will affect the cyber system in a CPS.

Currently, to the best of the author's knowledge, there is little literature [15] that describes a system's information property together with timing constraints. In order to include timing in the model, a special operation called "tick" is used. "Tick" does nothing but act as an atomic operation and represent the clock of the whole system moving by one unit of time.

Figure 5.1 lists the timing constraints of the CFPS system and also the corresponding number of ticks that had been used in the checking. The actual frequency ratio between the objects is 1000:330:1, however, in the model a reduced number of ticks is used to reduce the complexity of model checking. The pattern of the frequencies is kept close to this ratio, but is not exact.

After defining "tick" to represent the time lapse of the system, the models which use SPA can be modified. The UPFC system with both the security boundary at device level and the ControlledLine level have all been analyzed by adding the timing constraints as adding some "tick" after corresponding activities. The behavior of the

UPFC system is described in Table 5.3 to demonstrate the model of UPFC with the security boundary at the UPFC device. Another model of the UPFC system with the boundary at ControlledLine can be found as following.

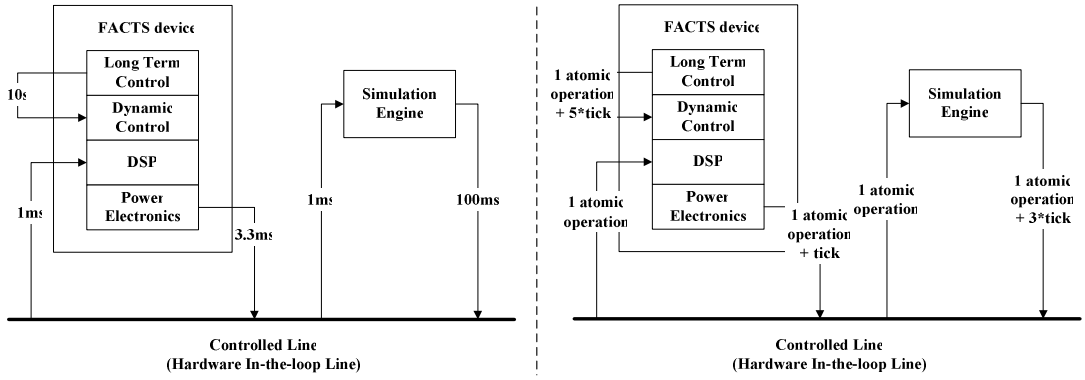

Figure 5.1 CFPS timing constraints and corresponding model to interpret the elapse of time

A SPA model of the UPFC system, which has the security boundary at the UPFC device, has the following timing constraints:

//this simulation is for the security boundary at UPFC device level
//considering any timing issues
//here value 0 means initial value, 1 means it could be set
//to a new value
bi UPFC_NT
(UPFC | LTC)\N
bi UPFC
(Behavior | HIL_h0 | HIL_l0)\L
bi Behavior
access_r_ll.(rl0.'val_l0.Behavior + rl1.'val_l1.Behavior) +
access_r_hh.(rh0.'val_h0.Behavior + rh1.'val_h1.Behavior) +
access_w_lh0.'wh0.tick.tick.tick.Behavior +
access_w_lh1.'wh1.tick.tick.tick.Behavior +

access_w_hl0.tick.tick.Behavior +

access_w_hl1.tick.tick.Behavior +

access_w_hh0.'wh0.tick.tick.tick.tick.tick.Behavior +

access_w_hh1.'wh1.tick.tick.tick.tick.tick.Behavior

bi HIL_l0

'rl0.HIL_l0 + wl0.tick.tick.tick.HIL_l0 + wl1.tick.tick.tick.HIL_l1

bi HIL_l1

'rl1.HIL_l1 + wl0.tick.tick.tick.HIL_l0 + wl1.tick.tick.tick.HIL_l1

bi HIL_h0

'rh0.HIL_h0 + wh0.tick.tick.tick.tick.tick.HIL_h0

+ wh1.tick.tick.tick.tick.tick.HIL_h1

bi HIL_h1

'rh1.HIL_h1 + wh0.tick.tick.tick.tick.tick.HIL_h0

+ wh1.tick.tick.tick.tick.tick.HIL_h1

bi LTC

a_r_hh.'access_r_hh.(val_h0.'put_h0.LTC + val_h1.'put_h1.LTC) +

a_w_hh0.'access_w_hh0.tick.tick.tick.tick.tick.LTC +

a_w_hh1.'access_w_hh1.tick.tick.tick.tick.tick.LTC

basi L

rh0 rh1 rl0 rl1

wh0 wh1 wl0 wl1

tick

basi N

val_h0 val_h1

val_l0 val_l1

access_r_hh access_r_hl access_r_lh access_r_ll

access_w_hh0 access_w_hh1 access_w_hl0 access_w_hl1

access_w_lh0 access_w_lh1 access_w_ll0 access_w_ll1

acth

a_r_hh a_r_hl

a_w_hh0 a_w_hh1 a_w_hl0 a_w_hl1

put_h0 put_h1
val_h0 val_h1
rh0 rh1 wh0 wh1 access_r_hh access_r_hl
access_w_hh0 access_w_hh1 access_w_hl0 access_w_hl1

SPA model of UPFC system, which has the security boundary at the ControlledLine, considering timing constraints:

//this simulation is for the security boundary at ControlledLine level
//without considering any timing issues
//here value 0 means initial value, 1 means it could be set
//to a new value
bi CL_NT
(CL | LTC)\N
bi CL
(Behavior | HIL_h0 | HIL_l0)\L
bi Behavior
access_r_ll.(rl0.'val_l0.Behavior + rl1.'val_l1.Behavior) +
access_r_hh.(rh0.'val_h0.Behavior + rh1.'val_h1.Behavior) +
access_w_lh0.'wh0.tick.tick.tick.Behavior +
access_w_lh1.'wh1.tick.tick.tick.Behavior +
access_w_hl0.tick.Behavior +
access_w_hl1.tick.Behavior +
access_w_hh0.'wh0.tick.tick.tick.tick.tick.Behavior +
access_w_hh1.'wh1.tick.tick.tick.tick.tick.Behavior
bi HIL_l0
'rl0.HIL_l0 + wl0.tick.tick.tick.HIL_l0 + wl1.tick.tick.tick.HIL_l1
bi HIL_l1
'rl1.HIL_l1 + wl0.tick.tick.tick.HIL_l0 + wl1.tick.tick.tick.HIL_l1
bi HIL_h0

'rh0.HIL_h0 + wh0.tick.tick.tick.tick.tick.HIL_h0

+ wh1.tick.tick.tick.tick.tick.HIL_h1

bi HIL_h1

'rh1.HIL_h1 + wh0.tick.tick.tick.tick.tick.HIL_h0

+ wh1.tick.tick.tick.tick.tick.HIL_h1

bi LTC

a_r_hh.'access_r_hh.(val_h0.'put_h0.LTC +

val_h1.'put_h1.LTC) +

a_w_hh0.'access_w_hh0.tick.tick.tick.tick.tick.LTC +

a_w_hh1.'access_w_hh1.tick.tick.tick.tick.tick.LTC

basi L

rh0 rh1 rl0 rl1

wh0 wh1 wl0 wl1

tick

basi N

val_h0 val_h1

val_l0 val_l1

access_r_hh access_r_hl access_r_lh access_r_ll

access_w_hh0 access_w_hh1 access_w_hl0 access_w_hl1

access_w_lh0 access_w_lh1 access_w_ll0 access_w_ll1

acth

a_r_hh

a_w_hh0 a_w_hh1 a_w_hl0 a_w_hl1

put_h0 put_h1

val_h0 val_h1

rh0 rh1 wh0 wh1 access_r_hh

access_w_hh0 access_w_hh1 access_w_hl0 access_w_hl1

5.3 RESULTS FROM THE CHECKER OF PERSISTENT SECURITY PROPERTY (COPS)

System behaviors are described in SPA and fed into CoPS to check against the security property of BNDC. The results are in Table 5.1. These results include the UPFC system which has the security boundary at the device level or at the ControlledLine.

Table 5.1 Results of applied CoPS against UPFC models described by SPA

System	Satisfy BNDC	# of states	Generated Graph	Time to check the property (s)	Composable
UPFC Device (No time constraints)	Yes	36	V: 34 E: 52	0.18	Yes
ControlledLine (No time constraints)	Yes	36	V: 34 E: 52	0.18	Yes
UPFC Device (With time constraints)	No	49	V: 47 E: 65	0.17	No
ControlledLine (With time constraints)	No	49	V: 47 E: 65	0.14	No

From the results listed in Table 5.1, conclusions can be drawn that for the security properties of UPFC system, without considering the timing constraints, whether the security boundary stops at the UPFC device or the ControlledLine, the UPFC system satisfies BNDC. This is a stricter result than those stated in Section 3.2.2, since Section 3.2.2 only claims the UPFC system with the security boundary at UPFC device level satisfies the noninference security property and with the security boundary at ControlledLine, satisfies nondeducible security property. However, as stated in [12], some systems that satisfy the nondeducible security property are not composable. This affects further consideration of the composed UPFC system with other systems to preserve security.

The current result is favorable since the UPFC system with the security boundary at ControlledLine not only satisfies the property of nondeducible but also satisfies the BNDC, which is a composable security property. The system satisfies the BNDC because the internal events brought by LTC have been taken into consideration. These internal events lead to e_4. Being more specific, the event system described in 3.2.2, shown in Figure 3.5, has been modified to allow e_4 to be a legal trace in the system by introducing the internal event τ. The system traces became {{}, $\tau.e_4$, e_1e_4, e_2e_4, $e_1e_2e_4$, …}. This system satisfies the BNDC since from the observation point of view the observed result is compatible with any high-level input even when composed with other systems [9]. Besides considering the composability in a CPS, timing constraints are also significant aspects. The UPFC system is also modeled in SPA with time taken into consideration. Table 5.1 provides those results that fed the SPA models to CoPS with timing. Unfortunately, UPFC system does not satisfy the security property of BNDC whether having the security boundary stop at the UPFC device or the ControlledLine. Besides not satisfying BNDC, the UPFC system with timing constraints is not composable.

The UPFC system with timing does not satisfy the BNDC security property. Intuitively, the divergence from BNDC by adding timing information to the UPFC system points out it is highly possible that timing constraints can be deduced or inferred by the observer since time lapse is a common event, which cannot be avoided in physical systems. It is something both trusted security domains and others can observe. An experiment is conducted to prove it is the pattern of timing constraints that introduces inference into the UPFC system. In this experiment, instead of classifying "tick" as a low-level event (naturally, it is a low-level event that can be observed by any level of security domain as long as a global clock exists), "tick" is classified as high-level event, and the CoPS tools has been rerun to check against the security property of BNDC. This experiment proves the initial guess that timing constraints introduce the possibility of inferences. One more item of security requirements need to be added to the system to demonstrate the need of removing the timing inference in the system. One possible solution to this problem is to introduce obfuscation into the system and mask the frequency pattern. Further research work needs to be conducted.

BNDC is important as CPSs are usually more or less composed of various physical and cyber systems. This fact shows the importance of composability to the security property, where composability means one or more composable secured systems. When composed together, their security properties will be preserved. In this way, no extra effort needs to be spent to prove the security of the system-of-system if every subsystem is secure and satisfies composable security properties.

Furthermore, an approach of proving the security of the system-of-system is implied here. Formally proving that the subsystems satisfy some composable security properties, such as BNDC, then directly composing these systems with other systems that satisfy composable security properties, results in a system-of-system that should satisfy the security property. To better illustrate this process, Figure 5.2 shows the process of using formal checking tools to check the security property and compose the systems into systems-of-systems.

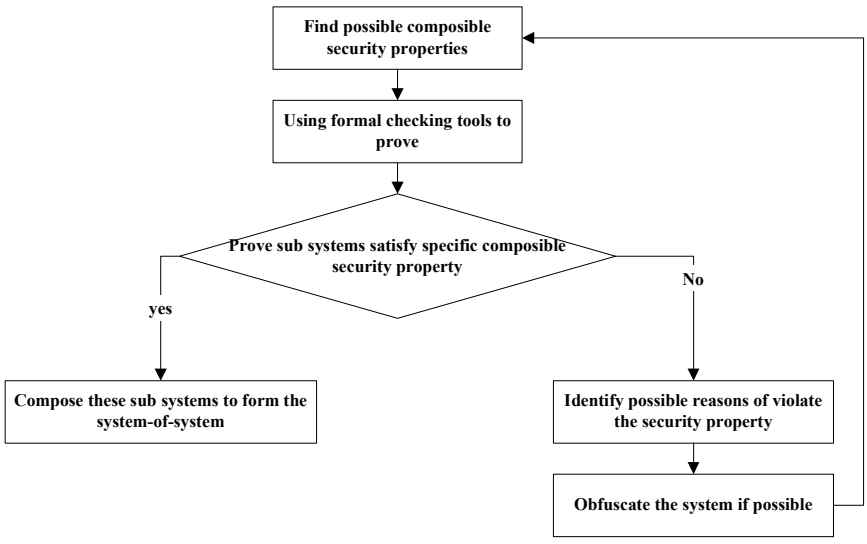

Figure 5.2 Process of using formal checking tool to prove the security property and compose systems which satisfy composable security properties into system-of-system

6. CONCLUSION

This thesis pointed out the importance of information flow security in a CPS, provided a process to model the information flow in a CPS, and suggested formalizing the system and using automatic checking tools to prove security properties.

6.1 CPS'S INFORMATION FLOW NEED TO BE CONSIDERED

This thesis analyzed the information flow in the CFPS. Under Assumptions 1, 2 and 3 described in Section 3.1, the UPFC local setting is confidential by considering the interface security models. However, the settings can still be deduced by mathematical computation with enough measurements taken from the ControlledLine(s), at the cyber-physical boundary. Meanwhile, UPFC control operations such as the Dynamic Control operation and Long Term Control operation cannot be inferred from observing the low-level behavior of CFPS. This is a promising result that shows considering the information flow of the CFPS, the confidentiality of the UPFC data setting and the control operations are not broken by inference or deducing information from information flow. This kind of self-obfuscation, in which the internal events of a system can obfuscate the system's behavior so that the external observer will not be able to deduce information from the system, not only appears in the power system but also in some other CPS such as oil pipeline systems, air traffic control systems and transportation systems. However, careful analysis is still needed at the cyber-physical boundaries since the cyber-physical interactions tend to leak the information to the outside world. This motivates a process or the modeling of the information flow of CPS.

6.2 A PROCESS TO MODEL INFORMATION FLOW IN CYBER-PHYSICAL SYSTEM IS POSSIBLE

The proposed process is suitable for a large system, which possibly has some other functional or non-functional requirements that mix with the security requirement. The current work leads to the conclusions and artifacts listed in Table 6.1.

Table 6.1 Conclusions and artifacts from the process of modeling information flow in Cyber-Physical System

Step 1: Elicit Information flow security requirements from misuse case **Conclusion**: It's possible and effective to use the misuse case to identify security requirements together with the system information flow model. System information flow security requirements can be elicited in this way. **Artifacts**: (1) Misuse case (2) System information diagram (3) Security requirement (CIA) (4) Information ow security requirements
Step 2: Identify functional or nonfunctional requirements related to the security requirements **Conclusion**: A strategy can be used to search the functional and nonfunctional requirements to find the possible requirement that couples with the system information flow security requirements. **Artifacts**: (1) Nonfunctional requirements list (temporal requirements) couple with the information flow security requirements
Step 3: Apply available security models and formal evaluation **Conclusion**: Available security properties and models that are widely used in the defense community can be used to formalize a large system as long as it can be broken into smaller subsystems which are composable **Artifacts**: (1) Formal description of the system behavior using value passing SPA (2) Formal description of the security models (noninference and nondeducible) using value passing SPA
Step 4: Extend security model according to the information analysis and formal evaluation **Conclusion**: Considering the system's temporal constraints, the security models used in step 3 have been modified to include the timing information of the system behavior

Table 6.1 Conclusions and artifacts from the process of modeling information flow in Cyber-Physical System (cont.)

Artifacts: (1) Formal description of the system behavior and temporal constraints using value passing SPA
step 5: Apply automatic tools to do the formal checking **Conclusion**: System behavior described in step 3 and 4 will be formalized using SPA and fed to the checking tool - CoPS, results will be fed back to revise the security requirements **Artifacts**: (1) Formal checking results

Furthermore, the results also show that formal checking tools, such as CoPS, are useful and efficient to prove the correctness of the security properties based on the available security requirements. If the correctness of a security property is proven by the tools, further security policies can be introduced accordingly. However, even if the security property is not validated by a formal checking tool, the results and checking process can uncover some potential security breach points and further aid the design of the system to provide better security.

6.3 FUTURE WORK

This thesis offers a concrete example of using the proposed process to model the Cyber-Physical System. More Cyber-Physical Systems need to be considered and various functional and non-functional requirements that coupling with the security requirements need to be identified and analyzed to further prove the wide application of this process. After modeling the information flow security of the Cyber-Physical Systems, possible solution as obfuscating the system need to be considered to secure the system. More work need to be done to prevent the system information flows through the cyber-physical boundaries.

BIBLIOGRAPHY

[1]Armbruster, A., Gosnell, M., McMillin, B. and Crow, M., "Power Transmission Control Using Distributed Max-Flow," *Procs of the 29th Annual IEEE International Computer Software and Applications Conference (COMPSAC)*, Edinburgh, Scotland, July 2005

[2] Bell, D. E. and LaPadula, L. J., "Secure Computer Systems: Mathematical Foundations," MITRE Corporation, 1973

[3] Crow, M., McMillin, B., and Atcitty, S., "An Approach to Improving the Physical and Cyber Security of a Bulk Power System with FACTS," ***EESAT Conferences 2005,*** http://www.sandia.gov/ess/Publications/pubs.html valid on 8-11-2007

[4] CoPS Homepage http://www.dsi.unive.it/~mefisto/CoPS/index.php valid on 7-1-2007

[5] Denning, D. Information Warfare and Security, Addison-Wesley, 1999

[6] Focardi, R., Gorrieri, R., "The Compositional Security Checker: A Tool for the Verification of Information Flow Security Properties," *IEEE Transaction on Software Engineering* VOL. 23, NO. 9, SEPTEMBER 1997

[7] Focardi, R., Gorrieri, R., and Martinelli, F., "Real-Time Information Flow Analysis," *IEEE Journal on Selected Areas in Communications*, VOL. 21, NO. 1, Jan. 2003

[8] Focardi, R., Gorrieri, R., and Martinelli, F. "Information Flow Analysis in a Discrete-Time Process Algebra," *In Proceedings of the 13th IEEE Computer Security Foundations Workshop (Csfw'00)* (July 03 - 05, 2000). CSFW. IEEE Computer Society, Washington, DC, 170

[9] Focardi, R. and Gorrieri, R. "A Classification of Security Properties for Process Algebras," J. Computer Security, vol. 3, no. 1, pp. 5–33, 1994/1995

[10] Goguen J. A. and Meseguer J. "Security Policies and Security Models," *Proc. of the IEEE Symposium on Security and Privacy (SSP'82)*, pag. 11-20. IEEE Computer Society Press, 2002

[11] Guttorm Sindre, Andreas L. Opdahl, "Eliciting Security Requirements by Misuse Cases," *Proceedings of the 37th International Conference on Technology of Object oriented languages and systems*. Sydney, pp 120-131, 20-23 Nov 2000

[12] Guttorm Sindre, Andreas L. Opdahl, "Capturing Security Requirements through Misuse Cases," *Proceedings of the 14th annual Norwegian informatics conference*. Norway, 2001

[13] Guttorm Sindre, Andreas L. Opdahl, "Templates for Misuse Case Descriptions," *Proceedings of the 7th international workshop on requirements engineering: Foundation for software quality.* Switzerland. June 2002

[14] Hoare, C. A. R. Communicating Sequential Processes. June 21, 2004

[15] Huang, J. and Roscoe, A. W. Extending noninterference properties to the timed world. *Proceedings of the 2006 ACM Symposium on Applied Computing* (Dijon, France, April 23 - 27, 2006). SAC '06. ACM Press, New York, NY

[16] IEEE Power Engineering Society FACTS Application Task Force, FACTS Applications, IEEE Publication 96 TP116-0, 1996

[17] Lamport, L. Specifying Systems -- The TLA+ Language and Tools for Hardware and Software Engineers Microsoft Research

[18] Lee Edward A. "Cyber-Physical Systems - Are Computing Foundations Adequate?" *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap* October 16 - 17, 2006 Austin, TX

[19] Lininger, A., McMillin, B. "Use of Max-Flow on FACTS devices," http://filpower.umr.edu/papers.htm valid on 8-3-2007

[20] Mantel, H. 2005. "The framework of selective interleaving functions and the modular assembly kit," In *Proceedings of the 2005 ACM Workshop on Formal Methods in Security Engineering* (Fairfax, VA, USA, November 11 - 11, 2005). FMSE '05. ACM Press, New York, NY, 53-62

[21] McCullough, D., "A hookup Theorem for Multilevel Security," *IEEE Trans. on Software Engineering* 1990

[22] McLean, J., "Security Models and Information Flow," *Procs. of the 1990 IEEE Computer Society Press*, 1990a

[23] McLean, J., "Security Models," *Encyclopedia of Software Engineering*, 1994

[24] McLean, J., "A general theory of composition for a class of 'possibilistic' security properties," *IEEE Trans. on Software Engineering*, 22(1):53--67, January 1996

[25] Midkiff, S.F. "Network-centric systems," *Pervasive Computing*, IEEE Volume 1, Issue 2, April-June 2002 Page(s):94 - 97

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

[27] O'Halloran, C., "A calculus of information flow," *Proc. European Symp. Research in Computer Security*, Toulouse, France, 1990

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

[29] Roscoe, A.W. The Theory and Practice of Concurrency. Prentice-Hall (Pearson) April 2004

[30] RT-SPIN homepage http://www-verimag.imag.fr/~tripakis/rtspin.html valid on 5-6-2007

[31] Ryan, M., Markose, S., Liu, X. F. ,McMillin, B. and Cheng, Y., "Structured Object-Oriented Co-Analysis/Co-Design of Hardware/Software for the FACTS Power System," *Procs. of the 29th Annual IEEE International Computer Software and Applications Conference (COMPSAC)*, Edinburgh, Scotland, July 2005

[32] Schneider, F. B. "Enforceable security policies," *ACM Trans. Inf. Syst. Secur.* 3, 1 (Feb. 2000), 30-50

[33] SPIN http://spinroot.com/spin/whatispin.html valid on 5-6-2007

[34] Standard CIP–002–1 through Standard CIP–009–1, ftp://www.nerc.com/pub/sys/all_updl/standards/sar/Cyber_Security_Standards_Board_Approval_02May06.pdf valid on 5-6-2007

[35] Sun, Y., Liu, X. F. and McMillin, B., "A Methodology for Structured Object-Oriented Elicitation and Analysis of Temporal Constraints in Hardware/Software Co-analysis and Co-design of Real-Time Systems," *Procs. of the 30th Annual IEEE International Conference on Computer Software and Applications (COMPSAC)*, Chicago, Sept. 2006

[36] Tang, H. and McMillin, B. "Analysis of the security of information flow in the Advanced Electric Power Grid using Flexible Alternating Current Transmission System (FACTS)," *First Annual IFIP WG 11.10 International Conference on Critical Infrastructure Protection* Dartmouth College Hanover, New Hampshire, USA March 19-21, 2007

[37] Zakinthinos, A. and Lee,E.S., "A General Theory of Security Properties," *Procs. of the 18th IEEE Computer Society Symposium on Research in Security and Privacy*, 1997

VITA

Han Tang was born on April 18, 1979, in Xi'an, Shaanxi, China. She earned the degree of Bachelor of Science in July 2001 and the degree of Master of Science in April 2004 in Electrical Engineering at Northwestern Polytechnical University. The degree of Master of Science in Computer Engineering was conferred upon her December 2007 at the University of Missouri at Rolla.

She was a project engineer working at the Schlumberger Beijing Geoscience Center for petrophysics interpretation software from 2004 to 2005. She was a software engineer and team leader in Simple Software Institute in China from 2001 to 2004. As a graduate research assistant at Northwestern Polytechnical University, she worked on several software projects related to power systems and researched network management system. Being a graduate research assistant at the University of Missouri at Rolla, she worked on a software-hardware co-designed project about the Flexible AC Transmission System (FACTS). Her academic interests include information security in distributed systems and formal security models for cyber-physical systems.