5-7-2016

On Cyber-Physical Security of Smart Grid: Data Integrity Attacks and Experiment Platform

Song Tan
Georgia State University

ON CYBER-PHYSICAL SECURITY OF SMART GRID: DATA INTEGRITY ATTACKS AND EXPERIMENT PLATFORM

by

SONG TAN

Under the Direction of WenZhan Song, PhD

ABSTRACT

A Smart Grid is a digitally enabled electric power grid that integrates the computation and communication technologies from cyber world with the sensors and actuators from physical world. Due to the system complexity, typically the high cohesion of communication and power system, the Smart Grid innovation introduces new and fundamentally different security vulnerabilities and risks. In this work, two important research aspects about cyber-physical security of Smart Grid are addressed: (i) The construction, impact and countermeasure of data integrity attacks; and (ii) The

design and implementation of general cyber-physical security experiment platform.

For data integrity attacks: based on the system model of state estimation process in Smart Grid, firstly, a data integrity attack model is formulated, such that the attackers can generate financial benefits from the real-time electrical market operations. Then, to reduce the required knowledge about the targeted power system when launching attacks, an online attack approach is proposed, such that the attacker is able to construct the desired attacks without the network information of power system. Furthermore, a network information attacking strategy is proposed, in which the most vulnerable meters can be directly identified and the desired measurement perturbations can be achieved by strategically manipulating the network information. Besides the attacking strategies, corresponding countermeasures based on the sparsity of attack vectors and robust state estimator are provided respectively.

For the experiment platform: ScorePlus, a software-hardware hybrid and federated experiment environment for Smart Grid is presented. ScorePlus incorporates both software emulator and hardware testbed, such that they all follow the same architecture, and the same Smart Grid application program can be tested on either of them without any modification; ScorePlus provides a federated environment such that multiple software emulators and hardware testbeds at different locations are able to connect and form a unified Smart Grid system; ScorePlus software is encapsulated as a resource plugin in OpenStack cloud computing platform, such that it supports massive deployments with large scale test cases in cloud infrastructure.

INDEX WORDS: Cyber Security, Testbed, Smart Grid

ON CYBER-PHYSICAL SECURITY OF SMART GRID: DATA INTEGRITY ATTACKS AND EXPERIMENT PLATFORM

by

SONG TAN

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2016

ON CYBER-PHYSICAL SECURITY OF SMART GRID:
DATA INTEGRITY ATTACKS AND EXPERIMENT PLATFORM

by

SONG TAN

Committee Chair: WenZhan Song

Committee: Xiaolin Hu
Zhipeng Cai
Michael Stewart

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
May 2016

DEDICATION

This dissertation is dedicated to Georgia State University. This dissertation is also dedicated to my parents Yanping Tan and Xiuxiang Yu, and my dear girlfriend Mei Du, for their endless support, sacrifice, hard work, and love.

ACKNOWLEDGEMENTS

This dissertation work would not be possible without the help of many people. I want to express my gratitude to my advisor Dr.WenZhan Song for his continuous guidance, patience and support. It is not only his invaluable academic knowledge and methodologies, but also his passionate attitude and inspiration, that help me succeed in my future career. I also want to thank all my committee members, Dr.Xiaolin Hu, Dr.Zhipeng Cai and Dr.Michael Stewart, for their suggestions and help during my PhD study.

I'd also like to thank my fellow lab-mates and co-workers for helping me through valuable knowledge sharing and contributions. I made many great friends and co-workers at Georgia State University during my PhD endeavors. I'll especially thank Mingsen Xu, Lei Shi, Debraj De, Dan Huang, Liang Zhao and Goutham Kamath for their wonderful and memorable companionship with research works, productive discussions, support, and really great friendships.

Finally, I would like to thank all of my family members and all my friends for their unconditional support, love, patience and understanding.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- GSU - Georgia State University
- CS - Computer Science
- NIST - National Institute of Standards and Technology
- LMP - Locational Marginal Prices

CHAPTER 1

INTRODUCTION

In this chapter we first introduce the main background of the thesis: the *Smart Grid* and the *Cyber-Physical Security* of Smart Grid. Then we present our research focus.

1.1 Introduction

Power system is the most fundamental and complicated artificial system in human society. In United States, there are over 5,000 power plants, over 200,000 miles of high-voltage transmission, and over 5.5 million miles of distribution lines [1]. The traditional power system evolves very little over the past 50 years, which results in significant inefficiency and vulnerability. As reported in [1], the annual cost to U.S businesses of power outages and distribution loss is greater than $100 billion.

1.1.1 Smart Grid

With the help of the emerging information technology, the legacy power grid is evolved along the journey to Smart Grid. A Smart Grid is a complex cyber-physical intelligent power system which leverages the cyber infrastructure within power system for sensing, control, computation and communication, in order to achieve self healing, resilience, sustainability and efficiency. It holds great promise for revolutionizing the energy future to deliver cleaner and more efficient power, healthier air and lower carbon emissions [2]. Currently, the design architectures and implementation models for smart grid are still evolving and not finalized. One of the most well known common reference model of smart grid is proposed by the U.S National Institute of Standards and Technology (NIST) in [3]. A conceptual view of the NISTs smart grid reference model is depicted in Figure 1.1. The NISTs model is composed of seven domains: generation, transmission, distribution, customers, markets, operations, and service providers. The two-way electrical flows are

moving across the top four domains (power generation, transmission, distribution, and customer), which are controlled and managed by the bottom three domains (market, operations, and service providers) through communication flows. In addition, three typical customers are listed: Home Area Network (HAN), Building Area Network(BAN) and Industrial Area Network (IAN), where the Advanced Metering Infrastructure (AMI) takes place to monitor and manage the power and information flows through smart meters.

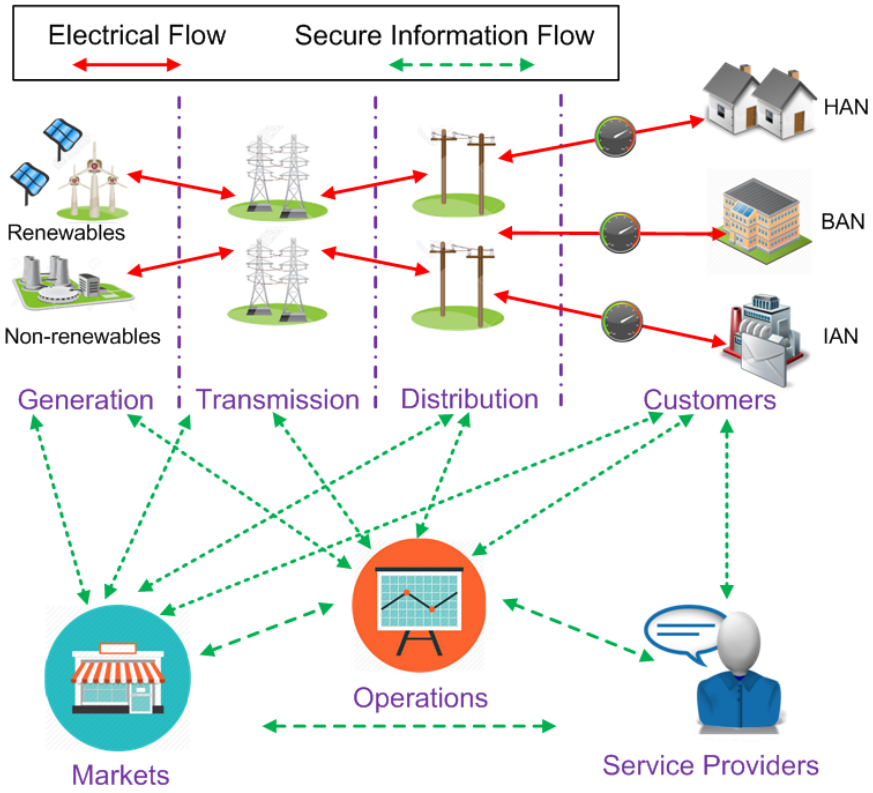


Figure 1.1 NIST reference model for smart grid

1.1.2 Cyber-Physical Security of Smart Grid

The Smart Grid vision is being realized through the implementation of cyber infrastructure overlaying the legacy physical power system. The cyber infrastructure enables the monitoring and control from millions of distributed end-points such as smart meters, sensors, automated control devices, all of which are interconnected by wired/wireless communication network. These features improve the efficiency, reliability, and sustainability of electric power system. However, the beauty

of the smart grid innovation comes with its danger: the integration and dependency upon cyber infrastructure would exceedingly increase the chances of cyber-physical threats and attacks. It might allows the attackers to intrude the communication network, acquire access to critical control routines, and even manipulate the meter measurements and system parameters to destabilize and perturb the physical power system in unpredictable ways, which could result in serious unstable power operating conditions, incorrect pricing scheme and even disastrous blackout.

The significance of cyber-physical security in smart grid lies in two folds. On the one hand, critical control processes such as state estimation, economic dispatch, load aggregation and demand response, etc, all rely on a secure and robust cyber infrastructure, which are indispensable to all aspects of smart grid. On the other hand, the cyber vulnerabilities may also enable adversaries to manipulate meter measurements, system parameters and price information, and even intrude and acquire direct access to these critical routines, to destabilize the grid in unpredictable ways. In the roadmap to secure control system proposed by Department of Energy and Department of Homeland security [4], energy control systems are subject to targeted cyber attacks. Potential adversaries have pursued progressively devious means to exploit flaws in system components, telecommunication methods, and common operating systems with intent to infiltrate and sabotage vulnerable control systems. Sophisticated cyber attack tools require little technical knowledge to use and can be found on the Internet, as can manufacturers technical specifications for popular control system equipment. As mentioned in [5], security issues are considered as one of the highest priorities for the smart grid design. Therefore, the cyber-physical security in smart grid has become a key concern with increasing urgency for the research community.

1.2 The Focus of This Work

The nature of threats and vulnerabilities are constantly changing, so application of best current practices for security is necessary but not sufficient. Therefore, for cyber-physical security research of Smart Grid, on one hand, it is essential to expose the potential new risks and explore corresponding countermeasures to mitigate the effects. One the other hand, it is also indispensable to design and implement experiment platforms to vividly evaluate and demonstrate these ideas.

In this work, two important research aspects about cyber-physical security of Smart Grid are addressed: (i) The construction, impact and countermeasure of data integrity attacks; and (ii) The design and implementation of general cyber-physical security experiment platform. For data integrity attacks: based on the system model of state estimation process in Smart Grid, firstly, a data integrity attack model is formulated, such that the attackers can generate financial benefits from the real-time electrical market operations. Then, to reduce the required knowledge about the targeted power system when launching attacks, an online attack approach is proposed, such that the attacker is able to construct the desired attacks without the network information of power system. Furthermore, a network information attacking strategy is proposed, in which the most vulnerable meters can be directly identified and the desired measurement perturbations can be achieved by strategically manipulating the network information. Besides the attacking strategies, corresponding countermeasures based on the sparsity of attack vectors and robust state estimator are provided respectively. For the experiment platform: ScorePlus, a software-hardware hybrid and federated experiment environment for Smart Grid, is presented. Compared with previous related works, ScorePlus incorporates both software emulator and hardware testbed, such that they all follow the same architecture, and the same Smart Grid application program can be tested on either of them without any modification. ScorePlus provides a federated environment such that multiple software emulators and hardware testbeds at different locations are able to connect and form a unified Smart Grid system. ScorePlus software is encapsulated as a resource plugin in OpenStack cloud computing platform, such that it supports massive deployments with large scale test cases in cloud infrastructure.

The rest of thesis is organized as follows. In Chapter 2, we survey the related works in the literature. In Chapter 3, we introduce the data integrity attack model and problem formulation. Then in Chapter 4 and 5, we respectively present the online data integrity attack [6] and leverage-point based data integrity attack [7], and the corresponding countermeasures based on the sparsity of attack vectors and robust state estimator are also provided. In Chapter 6, we present ScorePlus [8], a software-hardware hybrid and federated experiment environment for Smart Grid. Finally we conclude this work in Chapter 7.

CHAPTER 2

RELATED WORKS

In this chapter we discuss the related works in the literature. First we cover the relevant works in data integrity attacks of Smart Grid. Then we present related works in smart grid experiment platforms.

2.1 Data Integrity Attacks in Smart Grid

Data integrity attacks refer to the kind of cyber attacks in which an adversary controls a set of meters in state estimation process [9] and is able to alter the measurements from those meters. As a recent and appealing attack paradigm, quite a few of existing works have addressed the issue.

2.1.1 Attack and Defense Strategy of Data Integrity Attacks in Smart Grid

In [10], Liu *et al.* are firstly introduced the concept of data integrity attacks in Smart Grid. Assuming the attacker keeps the original power network topology data and parameter data intact, the authors shows that the attacker can inject errors to the meter measurement data in certain ways while without being detected by the existing bad data detector. Inspired by the work in [10], extensive further developments are made in [11] [12] [13] [14] [15], etc. Different undetectable attacks and defence strategies are presented. In [16], the authors at first introduce another class of malicious data attacks, called topology attack. The key innovation is that the manipulations of power network topology data are also considered.

2.1.2 Explore the Impact of Data Integrity Attacks in Smart Grid

Besides the pure attack and defense strategies of data integrity attacks, quite a few works also endeavor to investigate the impact of it, particularly on the operations of real-time electrical markets. Xie *et al.* in [17] firstly investigate the impact of integrity attacks on power market

through virtual bidding. In [18], Kosut *et al.* evaluate the proposed data attacks by their generated market revenues and the work is further studied by Jia *et al.* in pursuit of maximizing the revenues [19]. Yuan *et al.* in [20] show that the data integrity attacks can lead to increased system operating costs due to inordinate generation dispatch or energy routing. With the objective of controlling real-time Locational Marginal Prices (LMP) directly through data attacks, Tan *et al.* in [21] employ a control theory based approach to analyze the attack effect on pricing stability. Esmalifalak *et al.* in [22] novelly adopt a two-person zero-sum game approach to characterize the relations between attackers and defenders within electricity pricing. More recently, the authors in [23] and [24] have respectively proposed formal analytic frameworks to quantify the impact of data qualities on real-time LMP.

2.1.3 Remarks and Our Contributions

There are several issues with respect to the related works in data integrity attacks of Smart Grid. First, the above related works are based on the assumption that the attacker has full knowledge about the network information of targeted power systems, which includes network topology data and branch parameter data, etc. In fact, in any given power system, the network information is huge and highly secured, and more importantly, these information are dynamic since the network topology could be reconfigured in both normal situations and contingencies. Therefore, it is rather difficult for attackers to achieve complete awareness of network information in practice. Second, the above works usually consider the manipulations of meter measurement data, and the errors introduced by the attacker to the meter measurements have to be in the column space of the Jacobian matrix of the state estimation system model. It is also essential to explore the manipulations of network topology data and branch parameter data. Finally, even though the previous works have indicated the impact of data integrity attacks on real-time electrical market operations, the relationships between these two are not explicitly characterized or integrated into the attackers' objectives.

In the light of above issues, in Chapter 3, we introduce the data integrity attack model, and explicitly characterize the relationship between data integrity attacks and real-time electrical market

operations as a process simulator. A global simulation-based optimization problem is formulated such that the attackers can maximize its financial revenues from the constructed attacks. Then to reduce the required network information of targeted power system when launching attacks, we propose an online attack construction approach [6] in Chapter 4, by which the attacker is able to construct the desired attacks without the network information. Furthermore, a network information attacking strategy LPAttack [7], is proposed in Chapter 5, in which the most vulnerable meters can be directly identified and the desired measurement perturbations can be achieved by strategically manipulating the network information. Besides the attacking strategies, corresponding countermeasures based on the sparsity of attack vectors and robust state estimator are also provided in each chapter.

2.2 Experiment Platforms for Smart Grid

Smart Grid is an intelligent power system that involves various embedded devices for sensing, control, computation and communication. Validating the functionality, security and reliability of Smart Grid applications within such a system requires the modeling and emulation of both power networks and communication networks, as well as the interactions between them. The design and implementation of experiment environment for Smart Grid are challenging and have been studied for years. In this section, we conduct extensive survey about the previous related efforts, which can be summarized into two categories: real hardware testbed and software simulation.

2.2.1 Real hardware testbed approach

Real hardware testbeds are the platforms employing actual physical smart grid devices for the experiments. We further classifies this line of works into two subcategories: flat-out hardware platforms and hardware-in-the-loop platforms.

Flat-out hardware platform: The flat-out hardware platforms are the ones which consist of pure hardware devices. As a grid scale, the Korean government selected the whole Jeju Island to build the Smart Grid testbed to allow the testing of Smart Grid technologies and business

models [25]. In [26], the Idaho National Lab incorporates the actual Smart Grid components including power generators, storage batteries, and substations to facilitate the cyber security research of power transmission in Smart Grid. In [27], Renewable Energy Laboratory in Greece set up a central-controlled testbed consisting of PV-panels, battery banks and inverters to investigate the renewable integration issues. As a lab scale, the authors in [28] design SmartGridLab testbed, which consists of intelligent power switch, power generator, renewable energy sources, smart appliances, and power meter, in order to test distributed demand response algorithm in Smart Grid. In [29], Joyer *et al.* demonstrate a lab scale microgrid testbed, which is based on IEEE 1547 to serve as an interconnection standard.

Hardware-in-the-loop platform: In hardware-in-the-loop platforms, the hardware devices only serve as parts of platforms, and need to interact with other software simulations to conduct complete experiments. Hahn *et al.* in [30] employs devices like Programmable Logic Units (PLUs) and Intelligent Electronic Devices (IEDs) for communication networks and Real-Time Digtial Simulators for power network simulation. Stanovich *et al.* in [31] integrates hardware from energy field, such as Remote Terminal Unit (RTU), fiber optical cables within the testbed. Recently, the author in [32] employs devices like smart meters, phasor measurement units, phasor data concentrator, and hybrid vehicle charging system, as the essential components of microgrid testbed in lab.

2.2.2 Software simulation approach

The software simulation platforms for Smart Grid are entirely composed of software components, which can also be further classified into two subcategories: individual simulation platforms and co-simulation platforms.

Individual simulation platforms: Individual simulation platforms are those which encapsulate the simulation features into one process. In other words, it is one single simulator to complete the job. These platforms are usually focused on a particular aspect of interests for Smart Grid. In [33], Guo *et al.* implement an energy demand management simulator to predict the per-

formance and response of a self-adaptive demand management strategy. In [34], Molderink *et al.* design and develop a simulation environment from scratch to analyze control algorithms for various appliances, such as micro-generators, energy buffers and water heater, etc. In [35], Faria *et al.* describe Demsi, a simulator for demand response in the context of competitive electricity markets and intensive use of distributed generation. Energy service provider and demand side player are modeled and strategic decisions are evaluated. In [36], Narayan *et al.* propose GridSpice, a cloud-based simulation package for Smart Grid. Employing the well known distribution network simulator Gridlab-D [37] and the transmission network simulator Matpower [38], GridSpice is being developed iteratively with an ultimate goal of modeling the interactions between all parts of the electrical network, including generation, transmission, distribution, storage and loads. All the individual software platforms can complete their own tasks in the specific application domain, but they all just concentrate on the power network simulation. The communication network, as another critical component of Smart Grid, is not considered in these platforms. This is why the co-simulation platform comes to the picture.

Co-simulation platforms: Co-simulation (co-operative simulation) is a simulation methodology that allows individual components to be simulated by different simulation tools running simultaneously and exchanging information in a collaborative manner [39]. In [40], Hopkinson *et al.* present a federated simulation combining NS2, a discrete event network simulator with PSCAD, a continuous time power network simulator. In [41], Godfrey *et al.* simulate the Smart Grid using NS2 and OpenDSS, a power network simulator. In [42], Mallouhi *et al.* introduce a co-simulation testbed specifically for security analysis of SCADA system by employing PowerWorld simulator and OPNET. In [43] and [44], Lin *et al.* introduces a global event queue to synchronize NS2 and PSLF simulation.

The co-simulation approach typically requires iteratively running separate communication and power network simulations. The performance is affected by putting extra overhead of an intermediary of synchronization. Meanwhile, the interactions between communication and power system models are usually restricted to fixed synchronization interval. Mismatches can occur be-

tween the real dynamics and the simulated one, which exposes reliability issues of such systems. An improvement about this issue is to integrate one simulation component into the other, such that a single global clock and event queue is employed in the simulation engine. In [45], electric network is made into a component within OMNET++, a network simulator. In [46], the adevs simulation tools are integrated into NS2 to provide a hybrid modeling of the continuous time power system and discrete event communication system by the discretization of the continuous power dynamics. More recently, a few of co-simulation frameworks are developed to further improve the interoperability between multiple individual simulation platforms. [47] introduce FNCS, a co-simulation framework to incorporate multiple power system simulators (Matpower, GridLAB-D) and communication network simulator (NS3). [48] present Mosaik, which allows the Smart Grid users to combine thousands of simulated entities distributed over multiple simulator processes.

2.2.3 Remarks about related work

From the above literature review, we summarize the characteristics of the real hardware testbed approach and the software simulation approach for experiments in Smart Grid.

The real hardware testbed approach achieves high fidelity by employing dedicated devices as part of the platforms. Critical control programs, such as demand response algorithms, routing protocols, and security strategies, can be tested in real hardware testbeds and they could be directly migrated to the actual Smart Grid embedded devices. However, the substantial cost and resource needed to deploy these devices limits the repeatability of these efforts in a lab environment. Moreover, these testbeds cannot be accessed and shared remotely by the public research community and are difficult to scale when the test case becomes quite large.

The software simulations, on the other hand, achieve much better availability, usability and scalability. Within software simulation platforms, the models of various Smart Grid objects can be easily scaled and statistically analyzed. However, since software simulation typically abstracts the operating system, communication protocols and power dynamics into various mathematical simulation models, it can only duplicate the behavior and structure of the system, but not the execution environment of Smart Grid applications. Moreover, many of the recent smart grid cases

are hard to model because either they are in binary executable forms (e.g. malware codes), or evolve too rapidly (attack vectors), which makes the simulation development labor-intensive and error-prone.

2.2.4 Our approach: Software-Hardware Hybrid and Federated Experiment Environment

In light of the above issues, a much ideal approach is to combine the merits from both hardware and software platforms such that the users may examine the performance of Smart Grid applications under realistic communication and computation constraints in hardware platform, while evaluating the corresponding scalability in software platform at the same time.

ScorePlus bridges the gaps between real hardware approach and software simulation approach. The key advantages of ScorePlus are:

- First, ScorePlus employs both software emulator and hardware testbed, which expose the same and transparent interface to users. A Smart Grid application can be tested on either of them without any modification. With this integration, researchers may examine the performance of Smart Grid applications under realistic communication and computation constraints in hardware testbed, while evaluating the corresponding scalability in software emulator at the same time.

- Second, the federated architecture of ScorePlus enables each distributed software emulator or hardware testbed maintain its own autonomy and unique strengths, while all work together to make their resources available under a unified framework. This plug-and-play architecture greatly facilitates the scalability of distributed experiments.

- Finally, as far as we know, ScorePlus is the first Smart Grid experiment tool that has close integration within cloud infrastructure. Leveraging the customized resource plugin mechanism of OpenStack cloud computing platform, the ScorePlus software are equipped with high reusability to support massive deployment with large scale test cases.

ScorePlus also has its own limits: the power network model is static DC power flow model such that we cannot use ScorePlus to capture transient physical dynamics, frequency control, power

balance, and voltage regulations, etc. The strengths and limitations of our approach compared with related works are listed in Table 2.1. The ScorePlus codes are open source released at `https://sourceforge.net/projects/scorepluset/`.

Table 2.1 Summary of features for related works and ScorePlus

	Hardware testbed	Software Simulation	ScorePlus
Model fidelity	High	Low	High
Accessibility	Difficult	Easy	Easy
Scalability	Low	High	High
Code migration	Yes	No	Yes
Time step	Real time	Real time/discrete time	Real time
Frequency control, Power balance	Yes	Yes	No
Voltage regulations	Yes	Yes	No
Cloud Infrastructure integration	N/A	N/A	Yes

CHAPTER 3

DATA INTEGRITY ATTACKS PROBLEM FORMULATION

In this chapter, we present the problem formulation of data integrity attacks, in which we illustrate the attack model and reveal the intrinsic relations between data integrity attacks and real-time electrical market operations.

3.1 Preliminaries and System model

3.1.1 State Estimation and Bad Data Detection

In state estimation process, the control center collects real time measurements z from the deployed sensors and combines the network topology and parameter information to calculate the real time estimates of the unknown system variables x. Mathematically [9], let $x = (x_1, x_2, ..., x_n)^T$ and $z = (z_1, z_2, ..., z_m)^T$ denote state variables and meter measurements, respectively, where n is the number of unknown state variables, m is the number of meters, and $m \geq n$. The state variables are related to the measurements by $z = h(x) + e$, where e is the Gaussian measurement noise with zero mean and a covariance matrix $\sigma^2 I$. Under DC power flow model [9], the measurement model can be represented as:

$$z = Hx + e \tag{3.1}$$

where z is the bus power injection (power generation or load) and branch power flow measurements, H is an $m \times n$ full rank Jacobian matrix of the measurement model and x is the voltage phases at all buses. Then the estimated system states $\hat{x}$ and branch power flows $\hat{f}$ are given by:

$$\hat{x} = (H^T H)^{-1} H^T z, \qquad \hat{f} = F\hat{x} \tag{3.2}$$

where F is the sensitivity matrix of branch flows with respect to the voltage phases. With DC power flow model, since there also exists a linear bijection between nodal power injections and voltage

phases [49], then given a reference bus, we would have a l-by-n injection shift factor matrix S to denote the sensitivities of branch power flows with respect to the bus power injections [50], where l is the number of branches. Assume z contains the injection measurements at all buses and flow measurements across all the branches, denoted by z_{in} and z_f respectively, then we have:

$$z_f = S \cdot z_{in} + e, \qquad \hat{f} = S \cdot z_{in} \tag{3.3}$$

Bad data detector employs residual to detect the abnormalities in measurement data. From (5.5),

$$\hat{z} = H\hat{x} = Kz, \qquad \text{where } K = H(H^T H)^{-1} H^T \tag{3.4}$$

Then the measurement residual can be written as:

$$r = z - \hat{z} = (I - K)z \tag{3.5}$$

The detector fires an alarm when $\| r \|_2 > threshold$.

3.1.2 Real-Time Electrical Market

A combined two-stage (day-ahead and real-time) market is widely adopted by major U.S. Independent System Operators (ISO) to stabilize the power system and calculate Locational Marginal Prices (LMP) [51]. In the day-ahead market, given the projected system load levels L, the ISO obtains the optimal generation dispatch P^*, the vector of predicted power generation at each bus. Then P^* are sent to all generators as generation reference, and day-ahead payments are collected from customers at all buses.

In the real-time stage, the ISO obtains the actual system response through state estimation, including the estimated power injections $\hat{P}, \hat{L}$ and branch flows $\hat{f}$. Then the following linear program [51] is solved to find the associated real-time LMP λ, a vector whose ith element λ_i is the

LMP at bus i:

$$\begin{aligned} \underset{\Delta P,\ \Delta L}{\text{minimize}} \quad & \sum C_i^G \Delta P_i - \sum C_j^L \Delta L_j \\ \text{s.t.} \quad & (\tau): \ \sum \Delta P_i = \sum \Delta L_j \\ & \Delta P_i^{min} \le \Delta P_i \le \Delta P_i^{max} \\ & (\mu_b): \sum_i S_{bi} \Delta P_i - \sum_j S_{bj} \Delta L_j \le 0, \text{for } b \in \hat{C} \end{aligned} \tag{3.6}$$

where ΔP and ΔL are the vectors of incremental generation dispatch and load dispatch at buses, with fixed cost C^G, C^L respectively. $\Delta P_i^{min}, \Delta P_i^{max}$ are predefined lower and upper bounds, usually chosen as -2MW and 0.1MW in practice [51]. S_{bi} is element at bth row, ith column of matrix S in (3.3). Note that $\hat{C}$ is called *congestion pattern* [24], which denote the sets of branches whose estimated power flow exceeds the flow limit f_b^{max},

$$\hat{C} = \{b : \hat{f}_b > f_b^{max}\} \tag{3.7}$$

Then by solving (3.6), the real-time LMP at bus $i = 1, 2, ..., n$, is calculated:

$$\lambda_i = \tau - \sum_{b \in \hat{C}} S_{bi} \mu_b \tag{3.8}$$

where τ, μ_b are the corresponding dual variables in (3.6).

To clear the real-time market, the generator at bus i receives revenue $\lambda_i(\hat{P}_i - P_i^*)$, and the customer at bus j pays $\lambda_j(\hat{L}_j - L_j)$, where $\hat{P}_i$ and $\hat{L}_j$ are the estimated power generation and load at bus i and j from state estimation, respectively [51].

3.2 Problem Formulation

Suppose a malicious party wants to generate revenues from the real-time electrical market by compromising a subset of meters ζ_A, such that only measurements from meters in ζ_A can be modified. Note that the following strategies can also be applied to reducing customers' payments.

3.2.1 Constraints of Attacks

Firstly, since the attacker can only modify the measurements from meters in ζ_A, then the perturbed measurements has to be in the form:

$$\tilde{z} = z + a, \quad a \in \{a \in \mathbb{R}^m | a = \Psi c, \forall c \in R^m\} \tag{3.9}$$

where a is the attack vector, and Ψ is the diagonal matrix:

$$\Psi = diag(\psi_1, ..., \psi_m) \tag{3.10}$$

where ψ_i is a ***binary*** variable and $\psi_i = 1$ iff meter $i \in \zeta_A$.

Secondly, the attack should not be detected by the bad data detector. Based on (3.5), the new residual becomes $\tilde{r} = r + (I - K)a$. Based on triangular inequality,

$$\| \tilde{r} \|_2 \leq \| r \|_2 + \| (I - K)a \|_2 \tag{3.11}$$

Here we introduce a parameter ε, such that $\| (I-K)a \|_2^2 \leq \varepsilon$. The smaller ε is chosen, the less likely the attack will be detected. In the extreme case when $\varepsilon = 0$, the attack becomes *unobservable* [18].

3.2.2 Objective of Attacks

The objective of the attack is to maximize revenues from the real-time electrical market. From the end of Section II (B), we can see that the generator at bus i receives revenue $\lambda_i(\hat{P}_i - P_i^*)$ in normal situation. We analyze λ_i and $\hat{P}_i - P_i^*$ separately.

First, from (3.6) and (3.8), it suggests that given a shift factor matrix S, the real-time LMP λ depends only on the ISO's congestion pattern observation [24], i.e, $\hat{C}$. Meanwhile, since the ISO determines $\hat{C}$ through the estimated branch flows $\hat{f}$ as in (3.7), and $\hat{f}$ are solely determined by the power injection measurements within $\tilde{z}$ as in (3.3), therefore, we can see that $\tilde{z}$, $\hat{f}$, $\hat{C}$ and real-time LMP λ form a Markov chain, such that given a tuple of (a, z, S), there is a single corresponding λ. In other words, the LMP λ is essentially a function of (a, z, S). So from now on, we denote LMP

as $\lambda(a, z, S)$. We abstract the complex routine of $\lambda(a, z, S)$ as a simulator, and the flow chart of the simulator is shown in Figure 3.1.

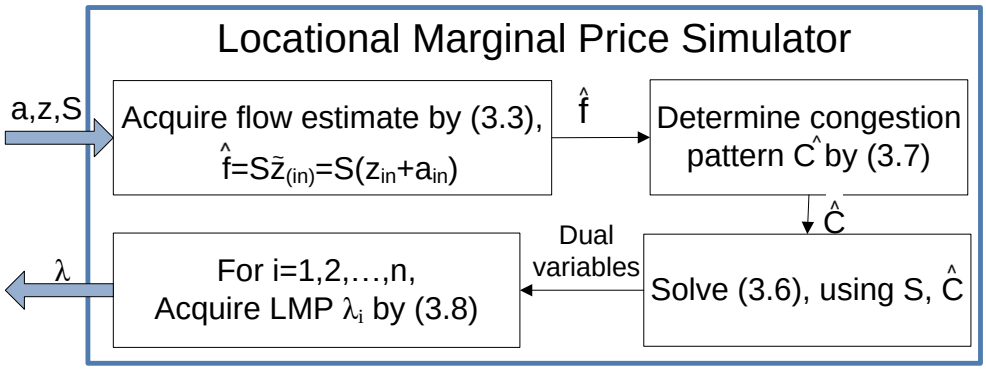


Figure 3.1 Locational Marginal Price Simulator $\lambda(a, z, S)$

Second, since the estimated power generation in real-time stage should match the optimal dispatch in day-ahead stage under normal situations [18] [19], then according to equation (3.4), for each bus i,

$$\hat{P}_i - P_i^* = K_i(z + a) - P_i^* = K_i a \tag{3.12}$$

where K_i is the corresponding row in matrix K to generation at bus i. Therefore, assume the attacker wants to make revenues from generations at buses within a target set $\mathbb{B}$. Then the total revenue from attack vector a is:

$$\mathcal{V}(a) = \sum_{i \in \mathbb{B}} \lambda_i(a, z, S) K_i a \tag{3.13}$$

3.2.3 Construct Attacks Against Real-Time Electrical Market

From all the above, the problem of constructing data integrity attacks against real-time electrical market can be formulated as a simulation-based global optimization problem P1:

$$\text{(P1):} \qquad \max_{a} \quad \mathcal{V}(a) = \sum_{i \in \mathbb{B}} \lambda_i(a, z, S) K_i a \tag{3.14}$$

$$\text{s.t.} \qquad a = \Psi c, \quad \forall c \in R^m \tag{3.15}$$

$$\| (I - K)a \|_2^2 \leq \varepsilon \tag{3.16}$$

3.2.4 Remarks

To solve P1, several facts are worth pointing out.

First, the objective function of P1 is based on complex simulator process in Figure 3.1. Only function values are available and there is no algebraic model to analyze differentiability and Lipschitz condition. Therefore, derivative-free optimization methods have to be employed [52].

Second, as in (3.3) (3.4) (3.6) (3.11) (3.13), the attacker will need the following knowledge to solve P1 accurately:

- 1) The meter measurement z.
- 2) The elements of matrix H, S and K.

The elements of matrix H, S and K depends on the detailed knowledge of **network information**, including network topology and branch parameters, such as the exact position of circuit breaker switches, transformer tap changers and power line admittances, etc. In fact, in any given power network, the network information is huge and highly secured, and these information could be dynamic since the topology can be reconfigured in both normal situations and contingencies. Therefore, it is rather difficult for the attackers to achieve complete awareness of network information in practice. A good question would be whether it is possible to construct available attacks when network information is not completely available.

CHAPTER 4

ONLINE DATA INTEGRITY ATTACKS IN SMART GRID

4.1 Online Construction of Data Integrity Attacks without network information

In this section, we consider the strategy to construct data integrity attacks without network information, in other words, when matrix H, S and K are unknown to the attacker. The measurement z, the cost coefficient C^G, C^L, and flow limit f^{max} are known to the attacker. By examining P1, we can see that the attacker need to specifically deal with $\| (I - K)a \|_2^2 \leq \varepsilon$ in the constraint, and $\lambda_i(a, z, S)$ and $K_i a$ in the objective.

4.1.1 Constraint $\| (I - K)a \|_2^2 \leq \varepsilon$

The constraint $\| (I - K)a \|_2^2 \leq \varepsilon$ determines whether or not the attack can be detected by the bad data detector. Since K is unknown, to safely launch an attack, the attacker would need to consider the extreme case of constraint $\| (I - K)a \|_2^2 \leq \varepsilon$, which is when $\varepsilon = 0$. In other words, if we could construct a vector a which has $\| (I - K)a \|_2^2 = 0$, then such an a would always satisfy the constraint and the bad data detector can never detect it. Note from (3.4) that when $a = Hv$, $\forall v \in R^n$, we always have $\| (I - K)a \|_2^2 = 0$. This is the so called unobservable attack [18]. Therefore, to satisfy the constraint, we just need to construct a vector a, which always lies in $\mathbb{R}(H)$, the column space of matrix H. The last question would be how to determine $\mathbb{R}(H)$ when H is unknown and even dynamic?

Inspired by [53], we can directly estimate and track the subspace $\mathbb{R}(H)$ using the measurement z. Let z_t denote the measurement vector at each time t, from (5.4):

$$z_t = Hx_t + e_t \tag{4.1}$$

To estimate $\mathbb{R}(H)$, at each time t, we aim at minimizing the following loss function,

$$J_t = \underset{J \in R^{m \times n}}{\arg\min} \sum_{i=1}^{t} \rho^{t-i} u_i(J), \tag{4.2}$$

where forgetting factor $0 \ll \rho \leq 1$ controls the memory and tracking ability, J is the estimated subspace with rank n since H is always full rank, and

$$u_i(J) = \min_x \| (z_i - Jx) \|_2^2, i = 1, ..., t \tag{4.3}$$

To solve (4.2), we alternate between the coefficient estimation and subspace update at each time t. Specifically, the coefficient vector is estimated by:

$$\begin{aligned} x_t &= \arg\min_x \| (z_t - J_{t-1}x) \|_2^2 \\ &= (J_{t-1}^T J_{t-1})^{-1} J_{t-1}^T z_t \end{aligned} \tag{4.4}$$

where J_0 is a random initialization. Then J_t is solved from:

$$J_t = \arg\min_J \sum_{i=1}^{t} \rho^{t-i} \| (z_i - Jx_i) \|_2^2 \tag{4.5}$$

where $x_i, i = 1, ..., t$, are estimated from (4.4).

Note for all row $h = 1, 2, ..., m$, the objective function in (4.5) can be rowwise decomposed [54] as $J_t = [J_1^t, J_2^t, ..., J_m^t]^T$:

$$\begin{aligned} J_h^t &= \arg\min_{J_h} \sum_{i=1}^{t} \rho^{t-i} (z_i(h) - x_i^T J_h)^2 \\ &= J_h^{t-1} + (z_t(h) - x_t^T J_h^{t-1})(W^t)^\dagger x_t \end{aligned} \tag{4.6}$$

where $W^t = \rho W^{t-1} + x_t x_t^T$ and $\dagger$ means pseudoinverse. Equation (4.6) is the classical formulation of Recursive Least Square (RLS) estimation with forgetting [55]. Based on RLS updating formula,

we further have:

$$(W^t)^\dagger = \rho^{-1}(W^{t-1})^\dagger - (\beta^t)^{-1}\alpha^t(\alpha^t)^T \tag{4.7}$$

$$\beta^t = 1 + \rho^{-1}x_t^T(W^{t-1})^\dagger x_t, \;\; \alpha^t = \rho^{-1}(W^{t-1})^\dagger x_t \tag{4.8}$$

We summarize the subspace estimation and tracking process for $\mathbb{R}(H)$ in Algorithm 5. At

Algorithm 1 Subspace Estimation and Tracking for $\mathbb{R}(H)$

Input: A sequence of real-time measurements $z_t, t = 1, 2, ...$
Initialize: An $m \times n$ random matrix J_0, and a diagonal matrix $(W^0)^\dagger = \delta I, \delta \gg 0$
for t=1,2,... **do**
  $x_t = (J_{t-1}^T J_{t-1})^{-1} J_{t-1}^T z_t$
  $\beta^t = 1 + \rho^{-1}x_t^T(W^{t-1})^\dagger x_t,$
  $\alpha^t = \rho^{-1}(W^{t-1})^\dagger x_t$
  $(W^t)^\dagger = \rho^{-1}(W^{t-1})^\dagger - (\beta^t)^{-1}\alpha^t(\alpha^t)^T$
  for h=1,2,...,m, in parallel **do**
    $J_h^t = J_h^{t-1} + (z_t(h) - x_t^T J_h^{t-1})(W^t)^\dagger x_t$
  Form J_t as $J_t = [J_1^t, J_2^t, ..., J_m^t]^T$

each time t, the attacker acquires the current estimated subspace J_t from Algorithm 5. Then the most conservative approach to replace the constraint $\| (I - K)a \|_2^2 \leq \varepsilon$ is:

$$a = J_t \cdot \eta, \quad \forall \eta \in R^n \tag{4.9}$$

Note that this constraint can be further relaxed in section D.

4.1.2 Objective $\lambda(a, z, S)$

To calculate $\lambda(a, z, S)$, the attacker should figure out the unknown matrix S. Assume z contains all the branch flow measurements and power injection measurements at all buses. Based on (3.3), let l denote the number of branches, for a particular branch flow measurement $z_f(j), j = 1, 2, ..., l$ in z_f, at each time t, we have:

$$z_f^t(j) = (z_{in}^t)^T S_j^t + e_t \tag{4.10}$$

where S_j^t is the jth row of matrix S at time t. Therefore, we can also estimate S_j^t through RLS with forgetting:

$$\begin{aligned} S_j^t &= \arg\min_{S_j} \sum_{i=1}^{t} \rho^{t-i}(z_f^i(j) - (z_{in}^i)^T S_j)^2 \\ &= S_j^{t-1} + (z_f^t(j) - (z_{in}^t)^T S_j^{t-1})(W^t)^\dagger z_{in}^t \end{aligned} \tag{4.11}$$

where $W^t = \rho W^{t-1} + z_{in}^t (z_{in}^t)^T$. Similar routines as in (4.7) (4.8) can be applied to find S_j^t. The process of estimating S is summarized in Algorithm 2. S_0 is initialized as a random $l \times n$ matrix.

Therefore, to calculate $\lambda(a, z, S)$, at each time t, the attacker first get S_t from Algorithm 2, then invoke simulator $\lambda(a, z_t, S_t)$ as in Figure 3.1.

Algorithm 2 Estimation for Shift Factor Matrix S

1: **Input:** Attack vector a, A sequence of real-time measurements $z_t, t = 1, 2, ...$
2: **Initialize:** A diagonal matrix $(W^0)^\dagger = \delta I, \delta \gg 0$
3: **for** t=1,2,... **do**
4: $\beta^t = 1 + \rho^{-1}(z_{in}^t)^T (W^{t-1})^\dagger z_{in}^t$,
5: $\alpha^t = \rho^{-1}(W^{t-1})^\dagger z_{in}^t$
6: $(W^t)^\dagger = \rho^{-1}(W^{t-1})^\dagger - (\beta^t)^{-1}\alpha^t(\alpha^t)^T$
7: **for** j=1,2,...,l, in parallel **do**
8: $S_j^t = S_j^{t-1} + (z_f^t(j) - (z_{in}^t)^T S_j^{t-1})(W^t)^\dagger z_{in}^t$
9: Form S_t as $S_t = [S_1^t, S_2^t, ..., S_l^t]^T$;

4.1.3 Objective $K_i a$

From (3.4), when H is unknown, K is unknown, so we cannot calculate $K_i a$ directly. However, the following Lemma 1 shed some light on the method to tackle this problem.

Lemma 1 *K in (3.4) is an orthogonal projector onto $\mathbb{R}(H)$.*

Proof 1 *Suppose $b \in R^n$, and let $\hat{b} = H\hat{x}$ be the orthogonal projection of b onto $\mathbb{R}(H)$. Then the residual $r = b - \hat{b} = b - H\hat{x}$ is orthogonal to $\mathbb{R}(H)$, hence, it is orthogonal to each of the columns*

of H. As a result, we have:

$$H^T(b - H\hat{x}) = 0 \implies H^T H\hat{x} = H^T b \tag{4.12}$$
$$\implies \hat{x} = (H^T H)^{-1} H^T b \implies H\hat{x} = H(H^T H)^{-1} H^T b$$
$$\implies \hat{b} = H(H^T H)^{-1} H^T b = Kb$$

Therefore, K is an orthogonal projector onto $\mathbb{R}(H)$.

Based on Lemma 1, we present the following theorem to calculate $K_i a$ when K is unknown.

Theorem 1 *Let matrix* $K = H(H^T H)^{-1} H^T$, *where* $H \in R^{m \times n}$ *with full rank n. Suppose there is another matrix* $J \in R^{m \times n}$ *also with full rank n, and* $\mathbb{R}(J) = \mathbb{R}(H)$. *Define matrix* K' *as:*

$$K' = J(J^T J)^{-1} J^T \tag{4.13}$$

,then $K = K'$.

Proof 2 *Since* $\mathbb{R}(J) = \mathbb{R}(H)$, *then:*

$$\forall x \in R^n, \quad \exists y \in R^n, \qquad s.t. \qquad Hx = Jy. \tag{4.14}$$

Based on Lemma 1, matrix K' *is also an orthogonal projector onto* $\mathbb{R}(H)$. *Therefore, for any* $u \in R^m$, *we can have:*

$$u = Ku + r, \quad u = K'u + r' \tag{4.15}$$

where $Ku, K'u \in \mathbb{R}(H)$, *and residual* r, r' *are orthogonal to* $\mathbb{R}(H)$. *So,*

$$(r - r')^T (Ku - K'u) = 0 \tag{4.16}$$

Since $r = u - Ku$, $r' = u - K'u$, *we further have:*

$$(K'u - Ku)^T (Ku - K'u) = 0 \tag{4.17}$$

which means for any $u \in R^m$, we have $Ku = K'u$. Then the orthogonal projector must be unique and $K = K'$.

Therefore, at each time t, the attacker can construct $K' = J_t(J_t^T J_t)^{-1} J_t^T$ using J_t generated from Algorithm 5, then use $K_i' a$ to replace $K_i a$ in the objective.

4.1.4 Summary

Based on Theorem 1, instead of using equation (4.9), we can further replace constraint $\| (I - K)a \|_2^2 \leq \varepsilon$ with $\| (I - K')a \|_2^2 \leq \varepsilon$. Therefore, the problem of attack construction without network information is formulated as P2:

$$\text{(P2):} \qquad \max_a \quad \mathcal{V}(a) = \sum_{i \in \mathbb{B}} \lambda_i(a, z, S) K_i' a_i \tag{4.18}$$

$$\text{s.t.} \qquad a = \Psi c, \quad \forall c \in R^m \tag{4.19}$$

$$\| (I - K')a \|_2^2 \leq \varepsilon \tag{4.20}$$

The attack construction process at each time t is summarized in Algorithm 3, where ν is a small constant.

Algorithm 3 Online Construction Of Data Integrity Attacks Against Real-Time Electrical Market

1: **Input:** A sequence of real-time measurements $z_t, t = 1, 2, ...$
2: **Initialize:** Launch Algorithm 1 and Algorithm 2;
3: **for** t=1,2,... **do**
4: Get J_t from Algorithm 1 and S_t from Algorithm 2;
5: **if** $\| J_t - J_{t-1} \|_F \leq \nu$ and $\| S_t - S_{t-1} \|_F \leq \nu$ **then**
6: Construct $K' = J_t(J_t^T J_t)^{-1} J_t^T$, update P2 objective;
7: Update the constraint $\| (I - K')a \|_2^2 \leq \varepsilon$ in P2;
8: Solving P2 using derivative-free optimization method. (In search process, evaluations of objective function invoke simulator routine $\lambda(a, z_t, S_t)$);
9: Based on solved vector a, modify the measurements of corresponding meters in ζ_A;

4.2 Countermeasure

In this section, we present the online defense strategy against the previously proposed attack. The defense strategy consists of two components: the attack detection component and the attack identification component. The detection component is responsible for indicating the existence of the attacks, and the identification component will be invoked afterwards to further identify the set of malicious meters. Only the real-time data streams of meter measurements are needed and no extra meter hardware investment is required.

4.2.1 Attack Detection

From the perspective of system operators, since the network topology and parameter information are highly secure in control center [9], they are the trustworthy baseline that can be employed to detect attacks. Note the shift factor matrix only depends on the network topology and parameter info, such that the matrix known to the system operator should be accurate at all times. We denote the true shift factor matrix as S_{true}. Meanwhile, as shown in Algorithm 2, the shift matrix can also be derived from the power flow measurements. Therefore, to detect the attack, the system operator could derive a corresponding shift factor matrix $\tilde{S}$ from a series of collected measurements $\tilde{z}$, then a discrepancy between the derived shift factor matrix $\tilde{S}$ and the S_{true} at hand will trigger an alarm to indicate the attack.

Specifically, suppose the control center collects a series of real time measurements $\tilde{z}$, which could be the manipulated measurements. Similarly as in (4.10), let l denote the number of branches, for a particular branch flow measurement $\tilde{z}_f(j)$, $j = 1, 2, ..., l$ in $\tilde{z}_f$, at each time t, we have:

$$\tilde{z}_f^t(j) = (\tilde{z}_{in}^t)^T \tilde{S}_j^t + e_t \tag{4.21}$$

where $\tilde{S}_j^t$ is the jth row of matrix $\tilde{S}$ at time t. As a result, we can also estimate $\tilde{S}_j^t$ iteratively

through RLS with forgetting:

$$\begin{aligned}\tilde{S}^t_j &= \arg\min_{\tilde{S}_j} \sum_{i=1}^{t} \rho^{t-i}(\tilde{z}^i_f(j) - (\tilde{z}^i_{in})^T \tilde{S}_j)^2 \\ &= \tilde{S}^{t-1}_j + (\tilde{z}^t_f(j) - (\tilde{z}^t_{in})^T \tilde{S}^{t-1}_j)(W^t)^\dagger \tilde{z}^t_{in}\end{aligned} \tag{4.22}$$

Note $W^t = \rho W^{t-1} + \tilde{z}^t_{in}(\tilde{z}^t_{in})^T$, and its pseudoinverse can be recursively updated as:

$$(W^t)^\dagger = \rho^{-1}(W^{t-1})^\dagger - (\beta^t)^{-1}\alpha^t(\alpha^t)^T \tag{4.23}$$

where

$$\beta^t = 1 + \rho^{-1}(\tilde{z}^t_{in})^T(W^{t-1})^\dagger \tilde{z}^t_{in}, \alpha^t = \rho^{-1}(W^{t-1})^\dagger \tilde{z}^t_{in} \tag{4.24}$$

Then derived shift factor matrix at time t can be formed as: $\tilde{S}_t = [\tilde{S}^t_1, \tilde{S}^t_2, ..., \tilde{S}^t_l]^T$.

After $\tilde{S}$ is derived, the discrepancy can be calculated as:

$$\gamma = \frac{\|S_{true} - \tilde{S}\|_F}{\|S_{true}\|_F} \tag{4.25}$$

Then if the discrepancy γ is greater than a tuned threshold κ, an alarm would be triggered and the attack identification process will be invoked.

4.2.2 Attack Identification

To further identify which measurements have been manipulated, suppose the control center collects a series of $\mathbb{N}$ real time measurements at continuous time stamps $\tilde{z}_t, t = 1, 2, 3..., \mathbb{N}$ and construct a m-by-$\mathbb{N}$ matrix $\tilde{Z}$ by columnwise stacking these measurement vectors together. From (3.9), we have:

$$\tilde{Z} = Z + A \tag{4.26}$$

where $Z = [z_1, z_2, ..., z_{\mathbb{N}}]$ is the matrix containing normal measurements and $A = [a_1, a_2, ..., a_{\mathbb{N}}]$ is the matrix containing all the attack vectors. It is known that power system measurements change gradually in continuous time interval [56], rendering Z typically low rank. Meanwhile, since the

attacker usually can only modify a limited number of meter measurements, such that matrix A tends to be sparse [57]. Therefore, the normal measurements and attack vectors can be recovered from:

$$\text{(P3):} \min_{Z,A} \quad \| Z \|_* + \omega \| A \|_1 \qquad \text{s.t. } \tilde{Z} = Z + A \tag{4.27}$$

where $\| \cdot \|_*$ denotes the nuclear norm, ω is a regularization parameter. The problem P3 is the well known sparse and low-rank matrix decomposition problem [58]. As long as we can recover sparse matrix A, then based on its rows that contain nonzero elements, we can identify which measurements have been attacked. As suggested by [59], P3 can be solved by employing Alternating Direction Method of Multipliers (ADMM) algorithm. The Lagrangian function of P3 is:

$$\begin{aligned}\mathbf{L}(Z, A, Q, \mu) = &\| Z \|_* + \omega \| A \|_1 + \langle Q, \tilde{Z} - Z - A \rangle \\ &+ \frac{\mu}{2} \| \tilde{Z} - Z - A \|_2^2\end{aligned} \tag{4.28}$$

where μ is positive number, Q is the Lagrangian multipliers, and $\langle \cdot \rangle$ denotes the Frobenius matrix product. For each iteration $k = 1, 2, 3, ...$ until convergence, Z is firstly updated as:

$$Z_{k+1} = U \mathbf{P}_{\frac{1}{\mu}}\{D\} V^T \tag{4.29}$$

where U, D, V^T are the singular value decomposition of matrix $(\tilde{Z} - A_k + \frac{1}{\mu} Q_k)$, and the operator $\mathbf{P}_a\{b\}$ is an elementwise applied soft thresholding function defined as:

$$\mathbf{P}_a\{b\} = \text{sign}(b) \cdot \max\{|b| - a, 0\} \tag{4.30}$$

Secondly, A is updated as:

$$A_{k+1} = \mathbf{P}_{\frac{\omega}{\mu}}\{\tilde{Z} - Z_{k+1} + \frac{1}{\mu} Q_k\} \tag{4.31}$$

Finally, the Lagrangian multiplier Q is updated as:

$$Q_{k+1} = Q_k + \mu(\tilde{Z} - Z_{k+1} - A_{k+1}) \tag{4.32}$$

4.2.3 Summary

From the above all, the procedure of online defense against data integrity attacks is illustrated in Figure 4.1. The attack detection process is monitoring the system using real-time measurement data stream all the time. Once an attack is detected, the attack identification process is launched to identify the potential malicious set of meters ζ_A. Then the measurements from meter set ζ_A will be removed from the measurement data stream used by the attack detection process. This procedure iterates until there is no attack indicated by the attack detection process. Algorithm 4 presents the

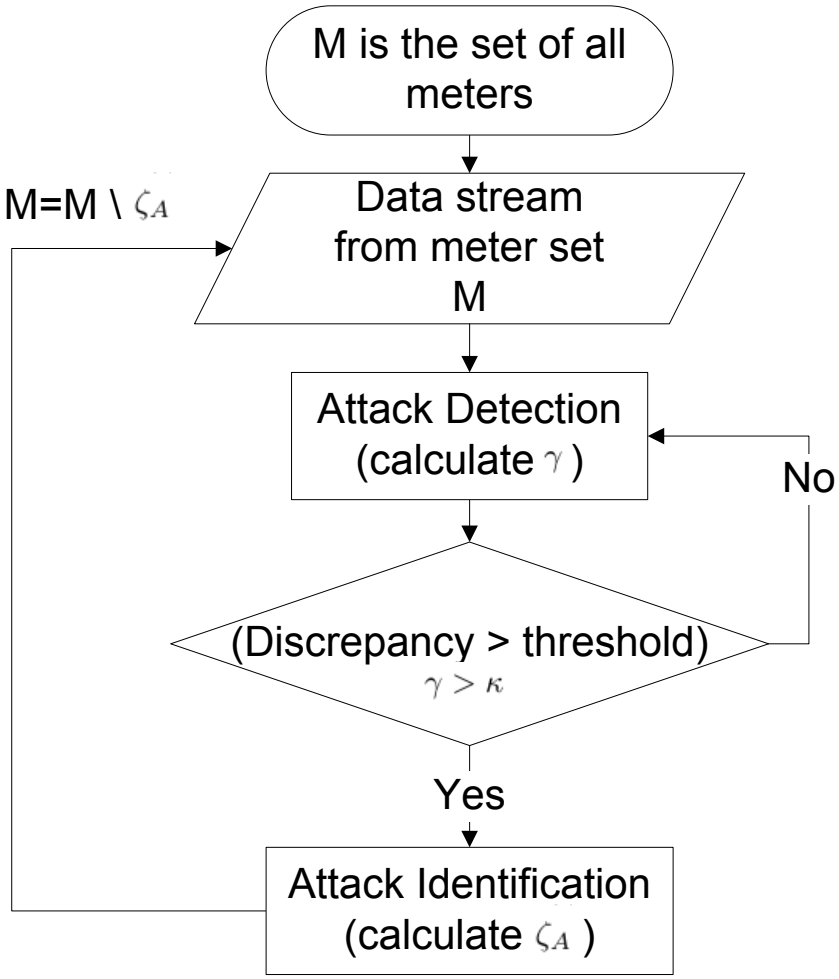


Figure 4.1 Flow Chart of Iterative Online Defense Process

implementation details. $\tilde{S}_0$ is initialized as a random $l \times n$ matrix, and κ is a tuned constant. $\mathbb{N}$ is the measurement buffer size, which determines how soon the system is able to identify the malicious

set of meters after an attack is detected.

Algorithm 4 Online Defense of Data Integrity Attacks Against Real-Time Electrical Market

M is the set of all measurement meters.
Input: A sequence of real-time measurements $\tilde{z}_t, t = 1, 2, ...$ from meter set M.
Initialize: A diagonal matrix $(W^0)^\dagger = \delta I, \delta \gg 0, k = 0$.
for t=1,2,... **do**
  $\beta^t = 1 + \rho^{-1}(\tilde{z}^t_{in})^T(W^{t-1})^\dagger \tilde{z}^t_{in}$,
  $\alpha^t = \rho^{-1}(W^{t-1})^\dagger \tilde{z}^t_{in}$
  $(W^t)^\dagger = \rho^{-1}(W^{t-1})^\dagger - (\beta^t)^{-1}\alpha^t(\alpha^t)^T$
  for j=1,2,...,l, in parallel **do**
    $\tilde{S}^t_j = \tilde{S}^{t-1}_j + (\tilde{z}^t_f(j) - (\tilde{z}^t_{in})^T \tilde{S}^{t-1}_j)(W^t)^\dagger \tilde{z}^t_{in}$
  Form $\tilde{S}_t$ as $\tilde{S}_t = [\tilde{S}^t_1, \tilde{S}^t_2, ..., \tilde{S}^t_l]^T$;
  if $\frac{\|\tilde{S}_t - \tilde{S}_{t-1}\|_F}{\|\tilde{S}_{t-1}\|_F} \leq \nu$ **then**
    Calculate $\gamma = \frac{\|S_{true} - \tilde{S}_t\|_F}{\|S_{true}\|_F}$
    if $\gamma > \kappa$ **then**
      (Concurrently start a separate and independent identification process as following.)
      Formulate matrix $\tilde{Z}$ by columnwise stacking most recent $\mathbb{N}$ measurements;
      Initialize $Z_0 = 0$, $Q_0 = 0$, $\mu = \frac{m \cdot \mathbb{N}}{4 \cdot \|\tilde{Z}\|_1}$, and $\omega = \frac{1}{\sqrt{\max(m, \mathbb{N})}}$
      while not converged **do**
        Update Z_{k+1} as in (4.29);
        Update A_{k+1} as in (4.31);
        Update Q_{k+1} as in (4.32);
        k=k+1;
      Based on the rows containing nonzero elements of recovered matrix A, assign the corresponding meters to the malicious set ζ_A.
      Update set M as $M = M \setminus \zeta_A$.
      Go to step 2.

4.3 Evaluation

In this section, we evaluate our proposed attacking strategies through IEEE bus benchmark system [60] with both synthetic data and real load data streams from the New York Independent System Operator [61]. All the numerical simulations are conducted in Matlab platform with software packages including @MATPOWER and patternsearch solver in Global Optimization Toolbox.

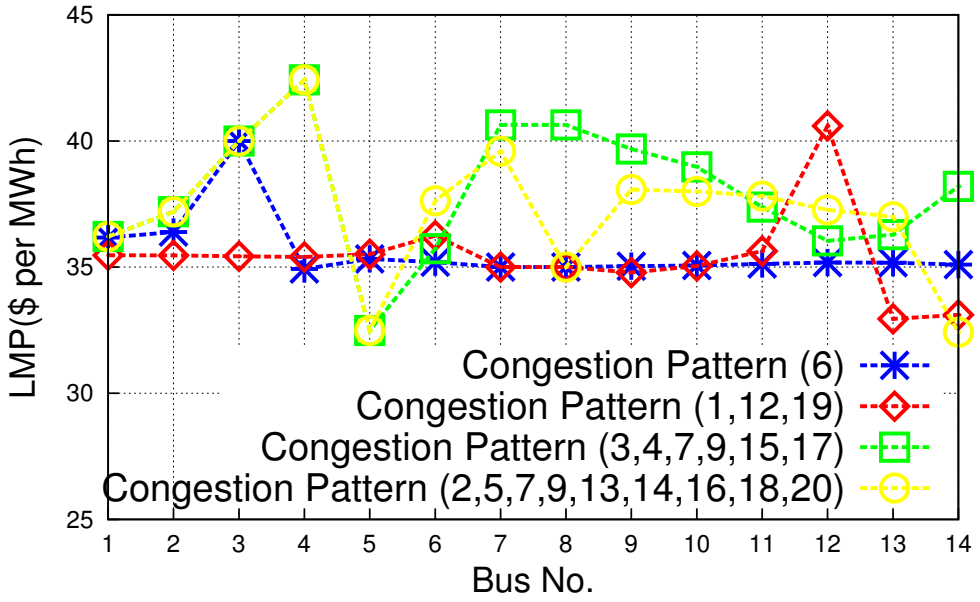


Figure 4.2 LMP at buses in different congestion patterns

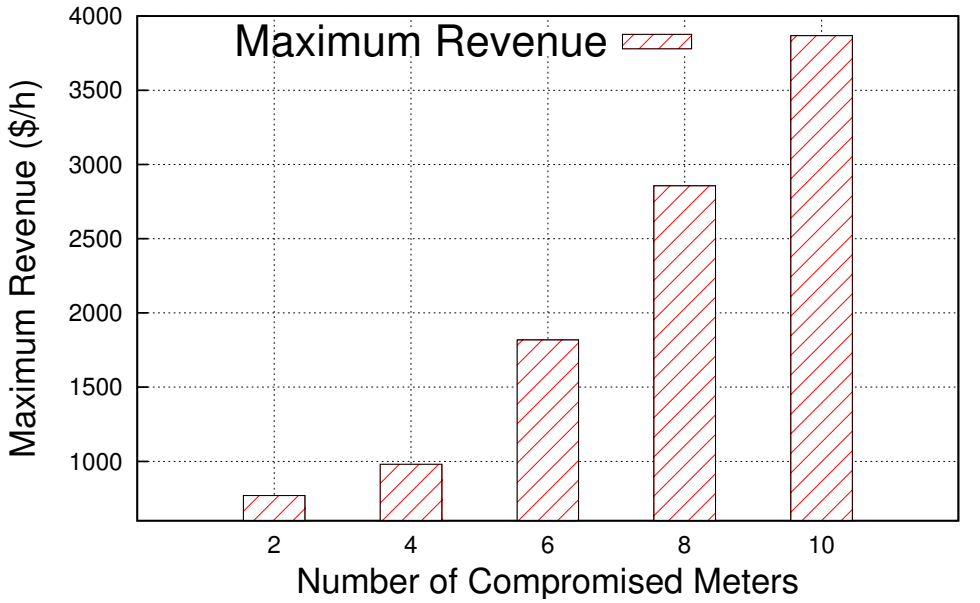


Figure 4.3 Maximum revenues with different numbers of compromised meters

4.3.1 Attack when network info is known

In this part, we examine the performance of P1 through IEEE14 bus system (14 buses and 20 branches), in which the network information is known. In this case, we use the synthetic load data that comes with MATPOWER. All power injection measurements and power flow measurements (in both directions for each line) are employed, such that $m = 54, n = 14$. We first examine the functionality of LMP simulator. Since the LMP at all buses totally depend on the congestion pattern, we directly plot the LMP under different congestion patterns in Figure 4.2. The congestion pattern includes the ID of branches whose power flows exceed the security limits. One interesting fact is that different congestion patterns could result in the same LMP at a particular bus, e.g, the LMP at bus 4 are the same in the last two congestion patterns, both of which have branch 7, 9 congested, and are incident with bus 4.

Table 4.1 Optimal attack vector a against IEEE14 with different sizes of ζ_A

size of ζ_A	optimal attack vector a
2	(0,63.0),(2,34.4)
4	(0,65.1),(2,32.0),(14,48.5),(34,-64.0)
6	(0,69.3),(2,32.0),(3,-48.0),(14,-48.0),(15,32.0),(34,-64.0)
8	(0,79.0),(2,32.0),(3,-32.0),(4,-48.0),(5,32.0),(14,52.3),(15,32.0),(34,-64.0)
10	(0,103.0),(2,32.0),(3,-48.0),(4,-64.0),(5,32.0),(14,68.0),(15,34.0),(17,32.0), (34,-64.0),(35,-33.0)

Then we demonstrate the optimal attack vectors in P1 when different number of meters are compromised. Table 6.2 lists the optimal attack vector a against IEEE14 system under different size of ζ_A. The notation (p, q) denotes the nonzero entries of vector a, and p is the index and q is the value. The corresponding maximum revenues under optimal attack vectors are given in Figure 4.3. We can see that the number of compromised meters has a significant impact on the revenues.

4.3.2 Online attack construction: when network info is unknown

In this part, we examine the performance of P2, in which the network information is unknown. Both IEEE14 bus and IEEE118 bus system are employed.

Data preparation For IEEE14 bus system, we incorporate the real-time load data streams from the New York independent system operator (NYISO) during a 48 hour period (10/01-10/02) in 2015. In NYISO, there are 11 load regions (CAPITL, CENTRL, DUNWOD, GENESE, HUDVL, LONGIL, MHKVL, MILLWD, NYC, NORTH, WEST). The load data are recorded for each region every 5 minutes. Therefore, for each region, there are load data at $12 \times 48 = 576$ continues time instances. Meanwhile, since there are exactly 11 load buses with IEEE14 bus system, the NYISO load data at each region can be directly mapped to each load bus in IEEE14 and generate 576 corresponding real-time measurements $z_t, t = \{1, 2, ..., 576\}$. Specifically, the following procedures are performed [62]:

- Map each load bus of IEEE14 bus system with one region of NYISO based on Table 4.2.
- Calculate ratio of the total loads from NYISO to the total loads of original IEEE14 buses system. Then divide each NYISO region load by the ratio and assign the resulted load to each load bus within IEEE14. The generation capacity in IEEE14 is not changed.
- Solve the system state x_t using power flow calculations based on the new loads and generate corresponding measurement z_t by the IEEE14 model.

Table 4.2 Mapping between NYISO Regions and IEEE14 Buses

Region Name	CAPITL	CENTRL	DUNWOD	GENESE	HUDVL	LONGIL	MHKVL	MILLWD	NYC	NORTH	WEST
IEEE14 Bus No.	2	3	4	5	6	9	10	11	12	13	14

For IEEE118 system, we leverage synthetic data generated from Monte Carlo simulation. In each Monte Carlo run, we use nonlinear state estimation model to generate measurement vector

at each time instance. State vectors at different time instances are assumed to be independent and identically distributed Gaussian random vectors with the mean equal to the operating states given in 118 bus data sheet.

Performance of Algorithm 1 and 2 To evaluate Algorithm 1 and 2, we use normalized errors to examine the performance of estimations for subspace $\mathbb{R}(H)$, matrix K, and S, which are defined as $\frac{\|(I-J_t \cdot J_t')H\|_F}{\|H\|_F}$, $\frac{\|K'-K\|_F}{\|K\|_F}$, and $\frac{\|S_t-S\|_F}{\|S\|_F}$, respectively. Meanwhile, to evaluate the performance of proposed algorithms in dynamic topology scenarios: For IEEE14 system, we disconnect bus 4 and bus 5 at time instance 200, and reconnect them at time instance 400. For IEEE118 system, we disconnect bus 15 and bus 33 at time instance 200, and reconnect them at time instance 400. Figure 4.4-4.11 show the normalized errors as the time goes in both cases. We can see that the algorithms are more sensitive to the topology changes in a smaller size of power system, in which the peak error will be larger but can be reduced more quickly.

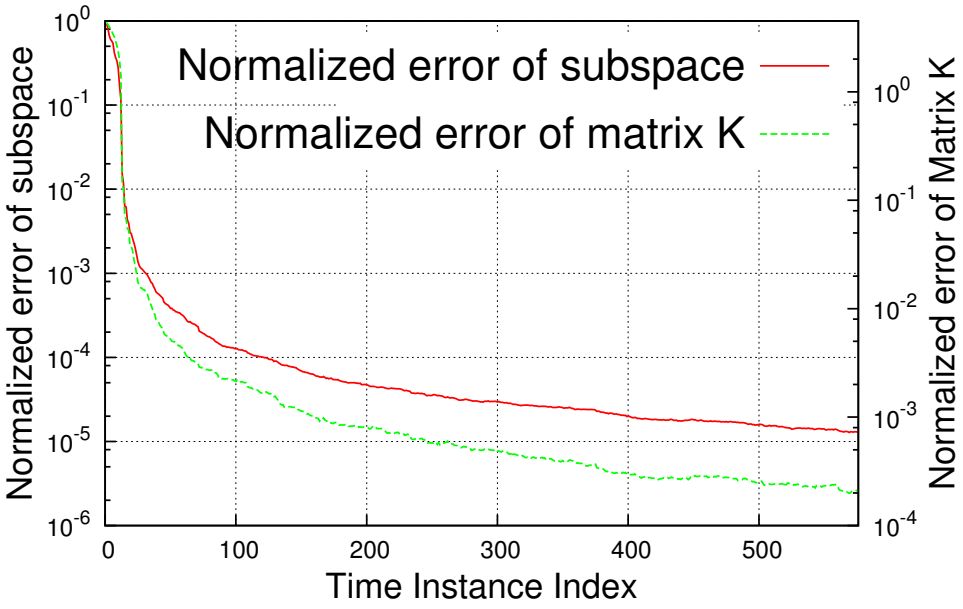


Figure 4.4 Subspace and matrix K estimation error in IEEE14 system

Performance of Algorithm 3 We evaluate the performance of Algorithm 3 from the perspective of attackers to see how much revenue can be generated. Since there will be errors in the

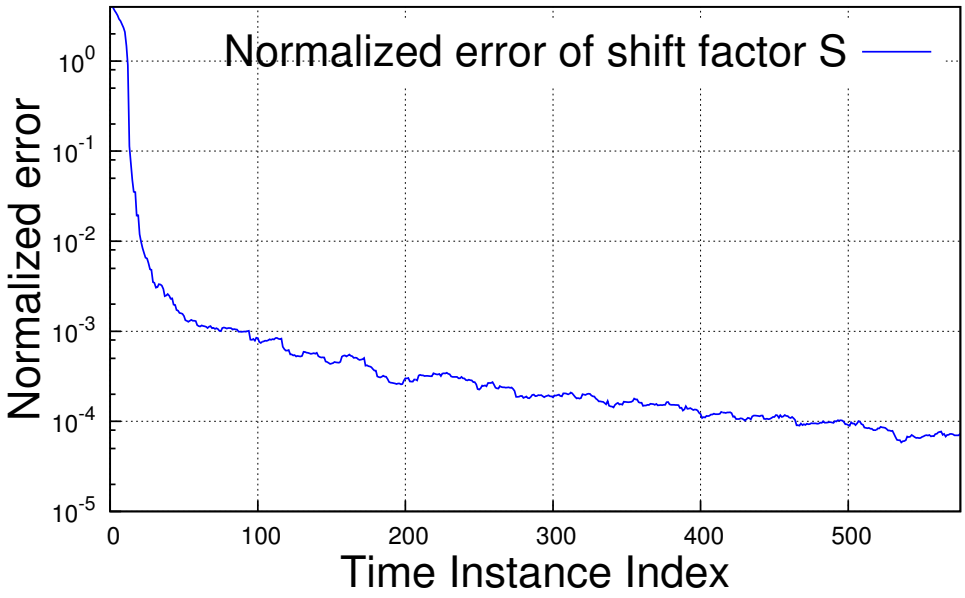


Figure 4.5 Shift factor S estimation error in IEEE14 system

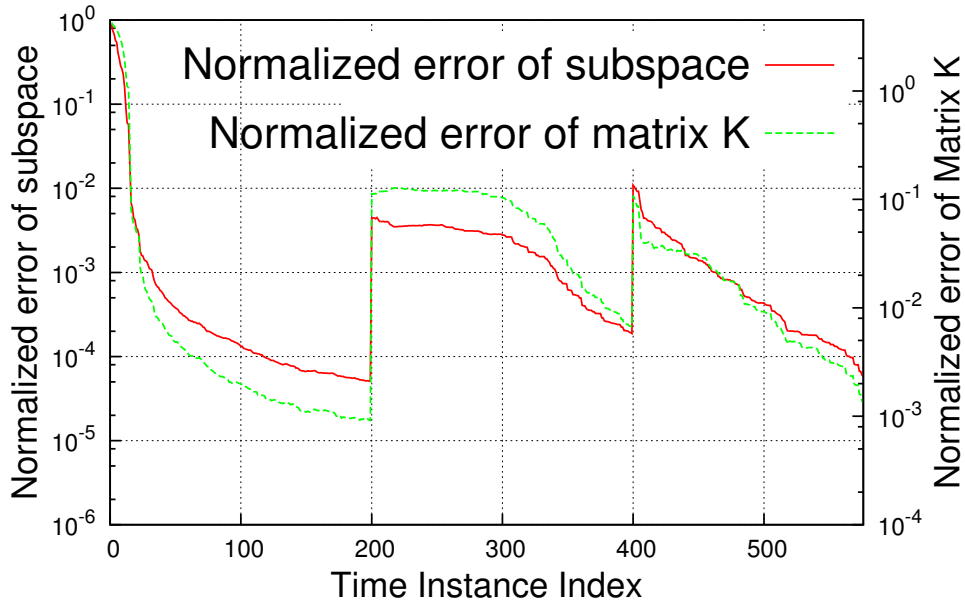


Figure 4.6 Subspace and matrix K estimation error in IEEE14 system with dynamic topology

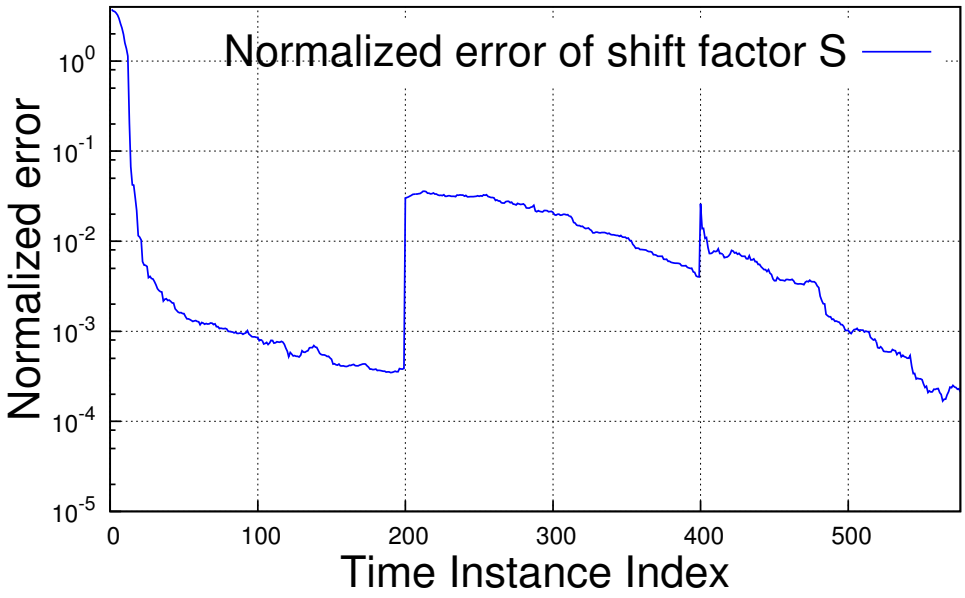


Figure 4.7 Shift factor S estimation error in IEEE14 system with dynamic topology

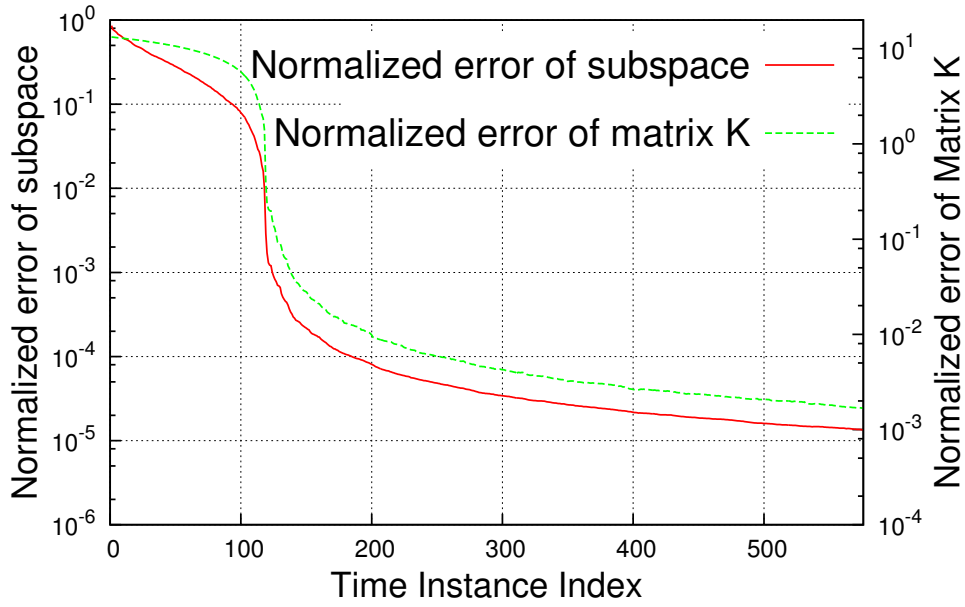


Figure 4.8 Subspace and matrix K estimation error in IEEE118 system

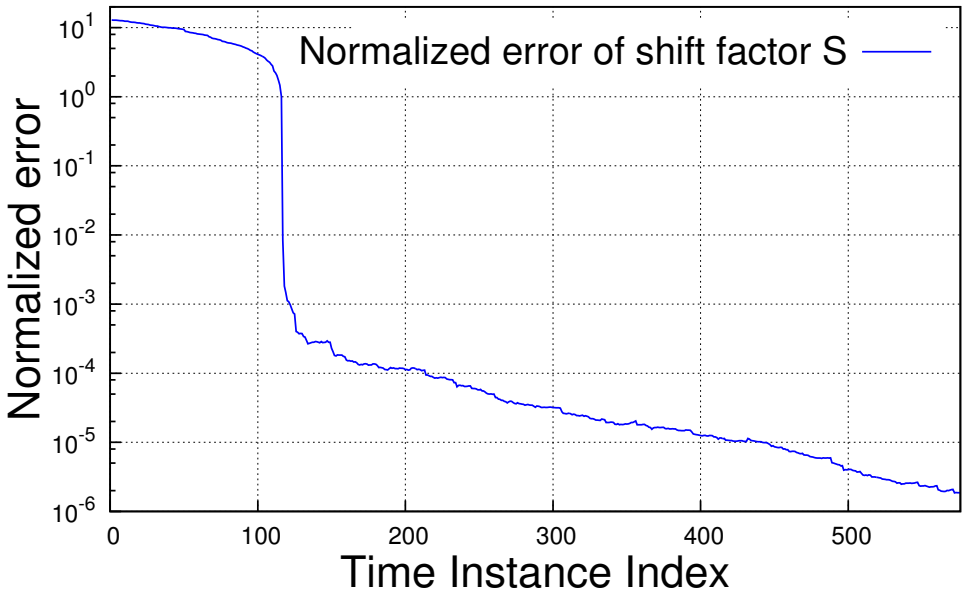


Figure 4.9 Shift factor S estimation error in IEEE118 system

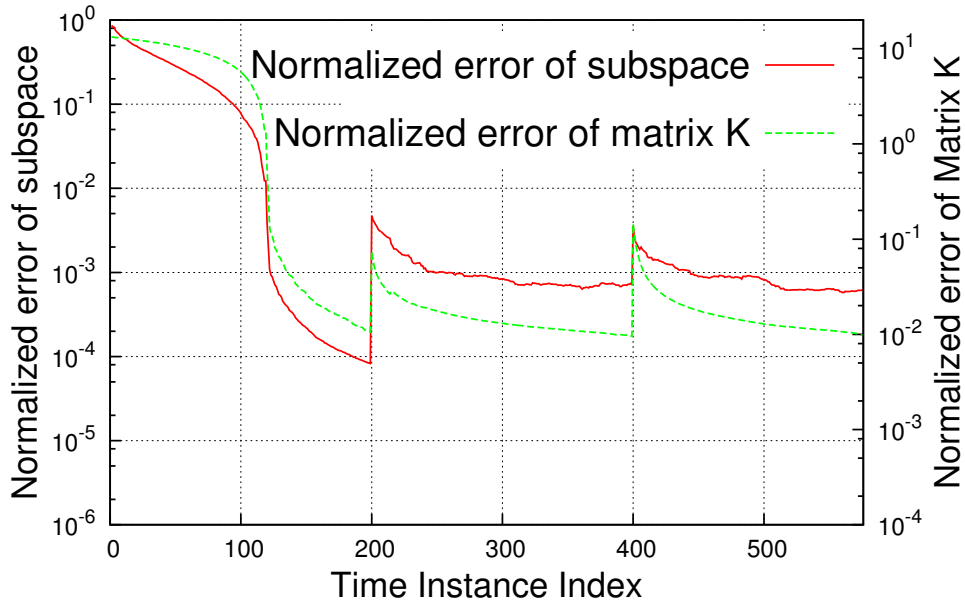


Figure 4.10 Subspace and matrix K estimation error in IEEE118 system with dynamic topology

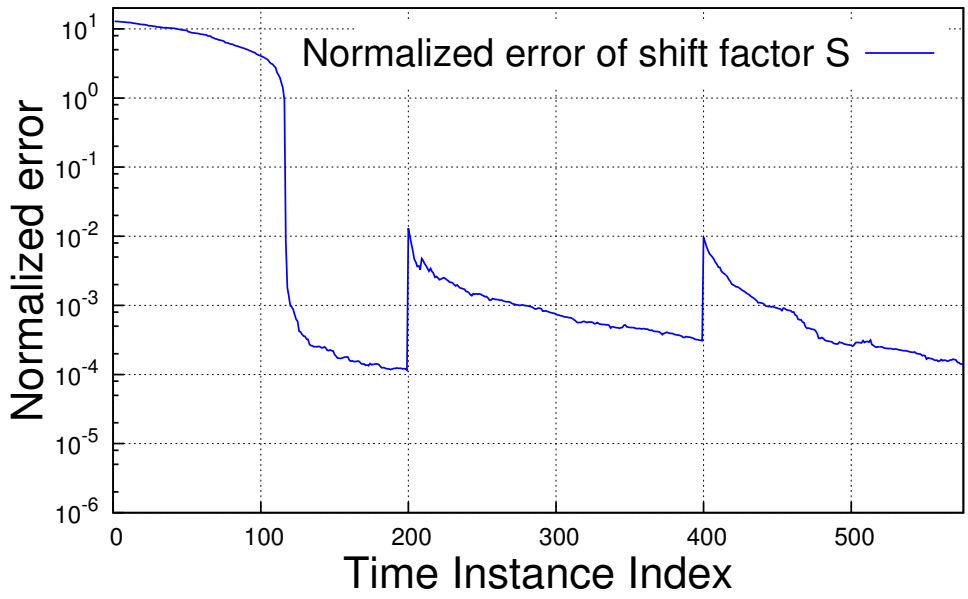


Figure 4.11 Shift factor S estimation error in IEEE118 system with dynamic topology

constructions of $\mathbb{R}(H)$, matrix K, and S, the bad data alarm is likely to be fired when the optimal attack vector from P2 is applied. Therefore, from the attacker's point of view, choosing the value of parameter ε in P2 would be critical. In IEEE14 system, the 0.05 significant-level bad data detector employs a chi-square distribution $threshold = \chi^2_{m-n,0.95} = \chi^2_{54-14,0.95}$. We present the corresponding real-time revenues under different ε in Figure 4.12 and 4.13 by employing the real load data from NYISO. The size of ζ_A is 10. In both cases, we plot the maximum revenues with known network information as a reference. When the network information is unknown, we can see that the revenues curve will start at a time point around 50 instead of 0. This is because in Algorithm 3, the normalized error of $\mathbb{R}(H)$, matrix K, and S can only become less than $\nu = 0.01$ until it collects certain amount of measurements. More importantly, the revenue curve is not continuous. The missing points in the curve are the time instances when the bad data alarm is fired due to the attack vector from P2. In that case, no revenue can be generated by the attacker. It can be seen that when ε is reduced from $threshold/2$, more time instances can generate revenues but the value of revenue is decreased correspondingly.

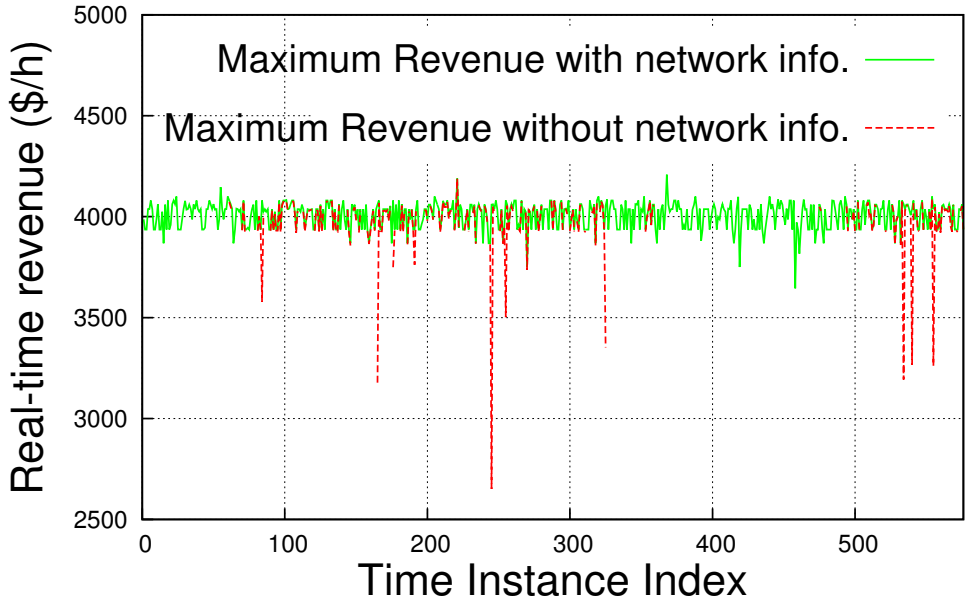


Figure 4.12 Real-time revenues with $\varepsilon = threshold$ in IEEE14

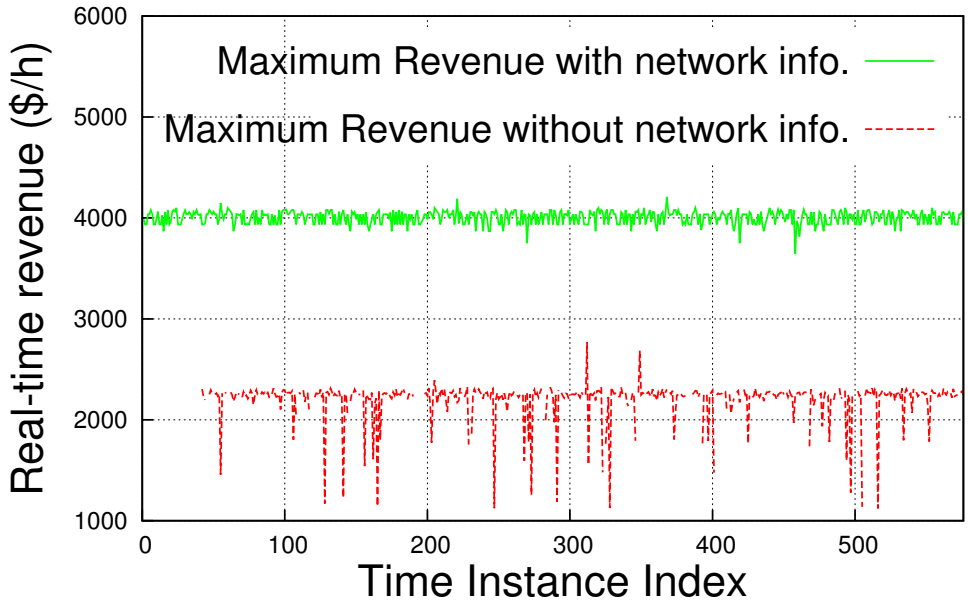


Figure 4.13 Real-time revenues with $\varepsilon = threshold/2$ in IEEE14

4.3.3 Countermeasure

In this subsection, we present the evaluation of countermeasure, which consists of both attack detection and attack identification in Algorithm 4.

Attack detection For the attack detection, corresponding to the two attack scenarios in Figure 4.12 and 4.13, we plot the resulted detection value γ when using both normal measurements and corrupted measurements in Figure 4.14 and 4.15. We can see that when the attack starts to generate revenue and cause data corruption, there would be a significant increase in value γ compared to its value in normal case. Therefore, using value γ can effectively indicate the existence of attacks.

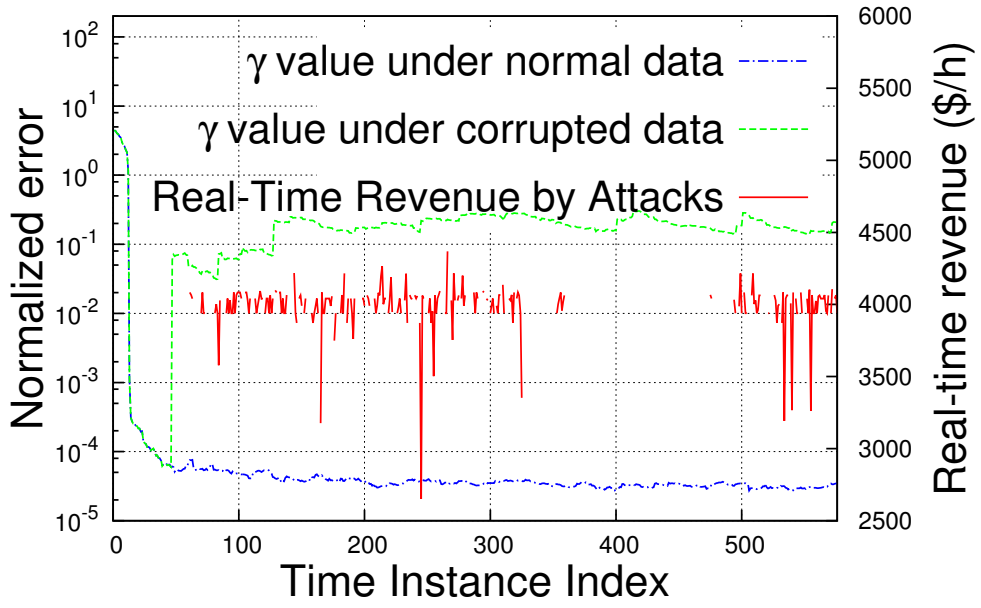


Figure 4.14 Attack detection with $\varepsilon = threshold$ in IEEE14

Attack identification To evaluate the performance of attack identification, we examine true positive rates and false alarm rates of malicious meter identification when different measurement buffer size $\mathbb{N}$ are employed. The measurement buffer size $\mathbb{N}$ determines how quickly the system would begin to identify the malicious meter sets after an attack is detected. From Figure 4.16, we can see that the buffer size $\mathbb{N}$ neither can be too small nor too large, which would result in either a

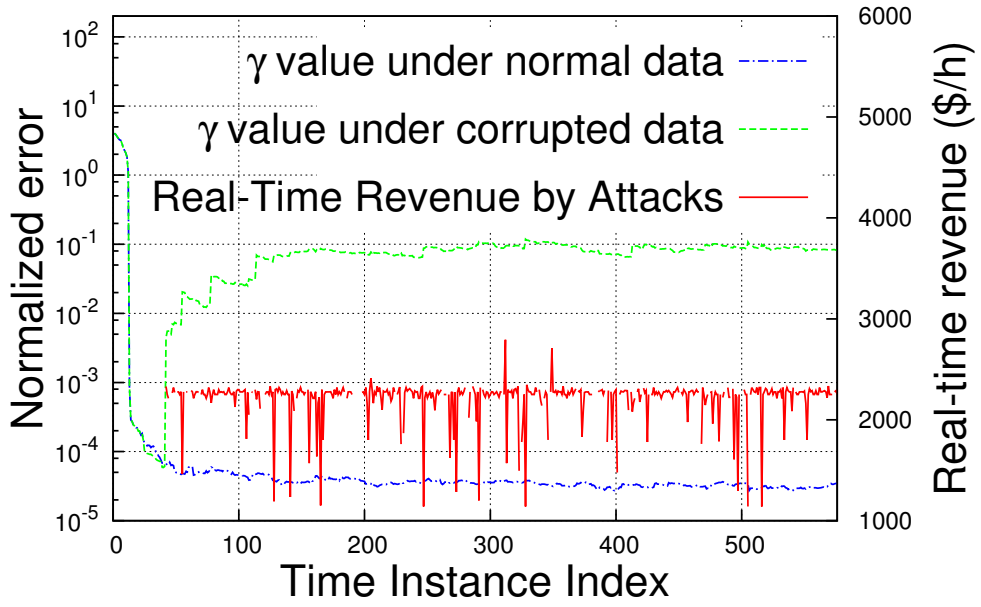


Figure 4.15 Attack detection with $\varepsilon = threshold/2$ in IEEE14

high false alarm rate or a poor true positive rate. A good trade off would be 96 in this case, where the formulated matrix $\tilde{Z}$ have slightly more columns than rows.

4.3.4 Online Computational Performance

The algorithm 1, 2, 3, and 4 are computationally intensive. Since the real time measurements are usually published every few minutes (5 minutes in PJM), our algorithms should be fast and responsive enough to adapt to the data generation speed. We evaluate the these algorithms within in Matlab 2013, on our testing machine (64 bits HP desktop with Inter(R) Core(TM) i7-5500 CPU@2.40GHz and 8GB memory). The *cputime* function in Matlab is employed to track the execution time. When receiving a new measurement vector z, the average time needed (seconds) to estimate $R(H)$ (Algorithm 1), estimate shift factor matrix S (Algorithm 2), calculate attack vector a (Algorithm 3), and detect and identify corresponding attack vector a (Algorithm 4), in both IEEE14 and IEEE118 bus systems, are listed in Table 4.3. From the table, we can see that due to the application of derivative-free optimization solver, the computation time increases significantly as the size of power network increases. However, in case IEEE118, Algorithm 3 is still responsive enough to generate the attack vector in real time. Moreover, since Algorithm 4 requires the

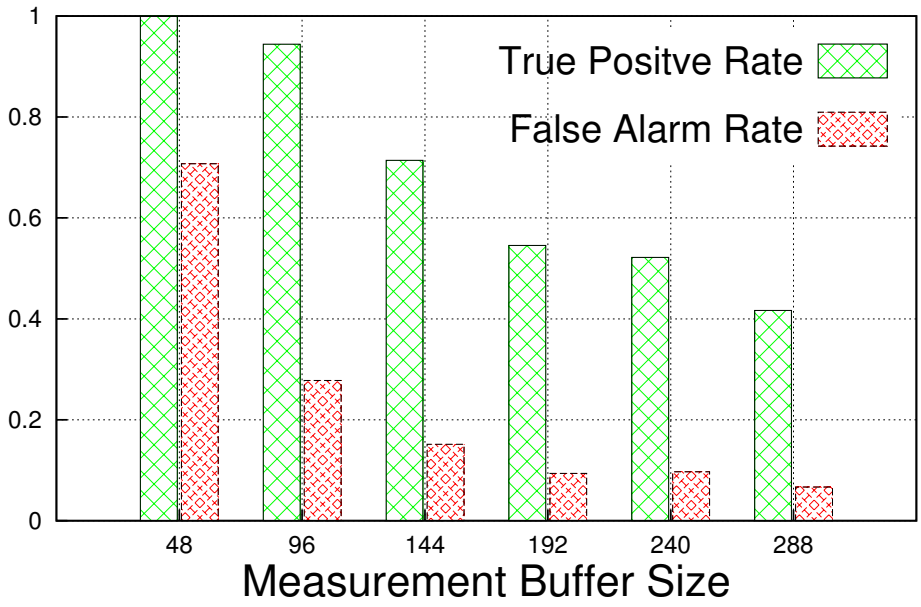


Figure 4.16 Attack identification with different measurement buffer size

buffering of several measurements before starting, it would be able to meet the online operation requirement.

Table 4.3 Computational Time of Algorithms 1-4 in Seconds

Case	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4
IEEE14 bus	0.0013	1.8989e-04	17.7542	33.4870
IEEE118 bus	0.0102	0.0027	101.546	219.772

4.4 Summary

In this chapter, we present online data integrity attacks against real-time electrical market. The online attack construction strategy is proposed when the attacker has no knowledge of power network information and our results show that the attacker could generate a fair amount of revenues through data integrity attacks. A corresponding online countermeasure is also presented to detect and identify the attacks.

CHAPTER 5

LPATTACK: LEVERAGE-POINT BASED DATA INTEGRITY ATTACKS IN SMART GRID

In this chapter, we present a novel class of malicious data attacks against Smart Grid state estimation, called LPAttack. Here LP represents *leverage points*, which are the outliers in the factor space of the regression model for state estimation [63]. Different from the previous works, we present a brand new approach to launching undetectable data attacks by strategically manipulating the network parameter data, such that leverage points are created within the factor space of the state estimation regression model. The key feature about leverage point is that the residual of the measurement corresponded with a leverage point will be very small even when it is contaminated with a very large error [63]. Based on this key feature, the attacker can freely introduce arbitrary errors into the corresponding meter measurements while without being detected by the existing bad data detection mechanism. Meanwhile, the existing leverage points can also be determined, such that the most vulnerable meters can be directly identified. The concept of leverage point is not new in power system state estimation, however, as far as we know, we are the **first** to explore the potentials of cyber attacks employing this feature. Our key contributions are:

- We present and rigorously prove the validity of the fundamental principles and strategies for launching LPAttack.
- We propose a potential countermeasure against LPAttack based on robust Schweppe-Huber Generalized-M estimator.
- We evaluate the LPAttack principles and countermeasure in IEEE test system, and examine in particular the effect of attacks on Locational Marginal Prices in real-time electrical market.

5.1 State Estimation and Bad Data Detection In Detail

State estimation: Figure 5.1 demonstrates the typical state estimation process in control center. State estimation takes three kinds of data as input:

- z: The meter measurement data, including power injections of buses and power flows of branches within power system.
- t: The network topology data, indicating the on and off status of various power network switches and circuit breakers between buses.
- p: The parameter data, typically including: 1) the branch susceptance data and 2) the variances of meter measurement errors, etc.

Typically, z, t and p are either telemetered data that are sent wirely/wirelessly from meters and sensors to control center, or kept in databases within data center.

After taking the input, the topology processing and observability analysis process would generate the regression model equation, in which:

- 1) The matrix H depends on the topology data t and the branch susceptance data in p;
- 2) The variances of each meter measurement error e_i in vector e, denoted by σ_i^2, are part of data in p.

Then the weighted least-square state estimator is used to solve the equation to get the best estimates of the unknown state variables x, which are the voltage magnitude and phase angle at each bus.

Mathematically, a precise definition of state estimation is given as follows [9]. Let $x = (x_1, x_2, ..., x_n)^T$ and $z = (z_1, z_2, ..., z_m)^T$ denote state variables and meter measurements, respectively, where n is the number of unknown state variables, m is the number of meter measurements, and $m \geq n$. Further let $e = (e_1, e_2, ..., e_m)^T$ denote meter measurement errors, which are assumed to be normally distributed with zero mean. The state variables are related to the measurements by:

$$z = h(x) + e \tag{5.1}$$

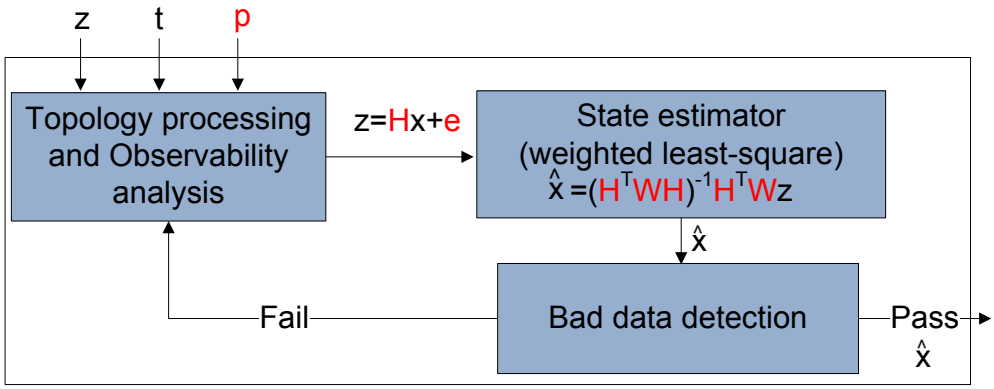


Figure 5.1 State estimation process in control center

where $h(x) = (h_1(x_1, x_2, ..., x_n), ..., h_m(x_1, x_2, ..., x_n))^T$, and $E(e) = 0$ and $cov(e) = W$, and W is defined as:

$$W = \begin{bmatrix} \sigma_1^{-2} & 0 & \cdots & 0 \\ 0 & \sigma_2^{-2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \sigma_m^{-2} \end{bmatrix} \tag{5.2}$$

where σ_i^2 is the variances of e_i.

The state estimation problem is formulated as the following Weighted Least Square (WLS) format:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \frac{1}{2} r^T W r \\ \text{subject to} \quad & z = h(x) + r \end{aligned} \tag{5.3}$$

For state estimation using standard DC power flow model [9], the Equation (5.1) can be represented by a linear regression model:

$$z = Hx + e \tag{5.4}$$

where H is an $m \times n$ full rank Jacobian matrix of the measurement model. Then the WLS state

estimator will give the following solution:

$$\hat{x} = (H^T W H)^{-1} H^T W z \tag{5.5}$$

Bad data detection: Measurement residuals are employed by the bad data detection techniques in Smart Grid to protect state estimation process against abnormality in measurement data, which are usually caused by nature or faulty sensors. The measurement residual is represented as:

$$r = z - \hat{z} = z - H\hat{x} \tag{5.6}$$

The objective function for bad data detection is defined as follows [9]:

$$J(\hat{x}) = \sum_{i=1}^{m} \frac{(z_i - H_i \hat{x})^2}{\sigma_i^2} = \sum_{i=1}^{m} \frac{r_i^2}{\sigma_i^2} \tag{5.7}$$

where H_i is the *ith* row of H. Then in the $J(\hat{x})$ test:

$$J(\hat{x}) = \begin{cases} \text{Bad data} , & \text{if } J(\hat{x}) > \varepsilon \\ \text{Good data} , & \text{if } J(\hat{x}) \leq \varepsilon \end{cases}$$

where ε is an empirical detection threshold defined in control center. If no bad data is detected, the state estimation resulted $\hat{x}$ would be accepted by the subsequent control processes. Otherwise, an alarm is fired and new data must be incorporated to start over the whole process.

5.2 LPAttack: Leverage-Point Based Attacks

We assume that the attackers can access and manipulate the data z, t, and p as needed to launch the attacks.

Let $\bar{z} = W^{\frac{1}{2}} \cdot z$, $\bar{r} = W^{\frac{1}{2}} \cdot r$, $\bar{H} = W^{\frac{1}{2}} \cdot H$ and $\bar{e} = W^{\frac{1}{2}} \cdot e$, from (5.4):

$$\bar{z} = \bar{H} \cdot x + \bar{e} \tag{5.8}$$

where $E[\bar{e}] = 0$ and $cov[\bar{e}] = I_m$. Then the WLS solution for x in (5.5) can be rewritten as:

$$\hat{x} = (\bar{H}^T \bar{H})^{-1} \bar{H}^T \bar{z} \tag{5.9}$$

and the residual,

$$\bar{r} = \bar{z} - \bar{H}\hat{x} = (I_m - K)\bar{z} \tag{5.10}$$

where I_m is the m dimensional identity matrix and K is the well known *hat matrix*[64], defined as:

$$K = \bar{H}(\bar{H}^T \bar{H})^{-1} \bar{H}^T \tag{5.11}$$

Since K is both symmetric ($K = K^T$) and idempotent ($K \cdot K = K$), then K_{ii} can also be written as:

$$K_{ii} = K_{ii}^2 + \sum_{j=1, j\neq i}^{m} K_{ij}^2 \tag{5.12}$$

It follows from the above equation that $0 \leq K_{ii} \leq 1$.

5.2.1 Principles of LPAttack

In the above regression model, the row vector $\bar{H}_i = (\bar{H}_{i1}, \bar{H}_{i2}, ..., \bar{H}_{in})$ defines a factor point in the n dimensional factor space of the regression. The outliers which are far away from the bulk of the factor points in this space are called ***leverage points*** and the corresponding measurements are called ***leverage measurements*** [63]. Geometrically, K_{ii} gives a measure of the distance from the factor point $\bar{H}_i$ to the bulk of the remaining $(m - 1)$ factor points. If K_{ii} is close to 1.0, it will be likely to behave as a leverage point.

$$r = z - \hat{z} = (I - K)z \tag{5.13}$$

A large value of K_{ii} will imply that there is a strong influence of the ith measurement z_i on its estimated $\hat{z}_i$, such that the estimated value is essentially determined by its measured value [9]. Thus we call the value of K_{ii} as ***the leverage of measurement*** z_i. As we can see from (5.10) and

(5.12), as K_{ii} becomes closer to 1, the residual r_i would be very small, no matter how much error is introduced into measurement z_i. Since the bad data detection process depends solely on the measurement residual, it would fail to reject the measurement data even when it is contaminated with a very large error. In other words, **the larger the attackers can increase the value of K_{ii}, the less likely the perturbation of measurement z_i can be detected by the bad data detection process.** This is the key idea of leverage-point attacks. Theorem 1 gives the general relationship between any set of measurements z and the values of K_{ii} when it can pass the $J(\hat{x})$ test.

Theorem 2 *Let ε be the threshold and $\sigma_{i=1,\dots,m}$ be the variances of errors in the $J(\hat{x})$ test. Given any set of measurements z, it is guaranteed to pass the $J(\hat{x})$ test when $\sum_{i=1}^{m}(1-K_{ii})\sum_{j=1}^{m}(z_j^2/\sigma_j^2) \leq \varepsilon$.*

Proof 3 *From (5.10), we have,*

$$\bar{r}_i = (\sum_{j=1,j\neq i}^{m} -K_{ij}\bar{z}_j) + (1-K_{ii})\bar{z}_i \tag{5.14}$$

From Cauchy-Schwarz inequality and (5.12),

$$\begin{aligned}
\bar{r}_i^2 &= ((\sum_{j=1,j\neq i}^{m} -K_{ij}\bar{z}_j) + (1-K_{ii})\bar{z}_i)^2 \\
&\leq [\sum_{j=1,j\neq i}^{m} K_{ij}^2 + (1-K_{ii})^2][\sum_{j=1}^{m}\bar{z}_j^2] \\
&= [K_{ii} - K_{ii}^2 + (1-K_{ii})^2][\sum_{j=1}^{m}\bar{z}_j^2] \\
&= (1-K_{ii})[\sum_{j=1}^{m}\bar{z}_j^2]
\end{aligned}$$

Since $\bar{r}_i = r_i/\sigma_i$ and $\bar{z}_i = z_i/\sigma_i$, then based on $J(\hat{x})$ test definition, we get:

$$\sum_{i=1}^{m}(1-K_{ii})\sum_{j=1}^{m}(z_j^2/\sigma_j^2) \leq \varepsilon \tag{5.15}$$

Theorem 2 suggests a basic requirement for the attacker to launch a successful attack. From a pragmatic point of view, it is more worth investigating how the perturbation of a particular mea-

surement z_i, can be marked by the only change of K_{ii}.

Theorem 3 *Suppose the original set of measurements z can bypass the $J(\hat{x})$ test. When one measurement z_i in z is perturbed into z'_i by the attacker, the new measurement set z' is guaranteed to bypass the $J(\hat{x})$ test if the attacker can change the original K_{ii} to K'_{ii}, which satisfies:*

$$K'_{ii} \geq K_{ii} + \frac{\sum_{j=1}^{m}(1 - K_{jj})(z_i'^2 - z_i^2)/\sigma_i^2}{\sum_{j=1, j\neq i}^{m}(z_j^2/\sigma_j^2) + z_i'^2/\sigma_i^2} \tag{5.16}$$

Proof 4 *Since the original z can pass the bad data detection process, from Theorem 1, we have,*

$$\sum_{i=1}^{m}(1 - K_{ii}) \sum_{j=1}^{m}(z_j^2/\sigma_j^2) \leq \varepsilon \tag{5.17}$$

In order to mark the perturbation of measurement z_i to z'_i by only changing K_{ii} to K'_{ii}, it must also satisfy,

$$[(\sum_{j=1}^{m} 1 - K_{jj}) + K_{ii} - K'_{ii}][(\sum_{j=1}^{m} \frac{z_j^2}{\sigma_j^2}) + \frac{(z_i'^2 - z_i^2)}{\sigma_i^2}] \leq \varepsilon \tag{5.18}$$

Compared with (5.17), we have:

$$(K_{ii} - K'_{ii})[(\sum_{j=1}^{m} \frac{z_j^2}{\sigma_j^2}) + \frac{(z_i'^2 - z_i^2)}{\sigma_i^2}] \leq -\sum_{j=1}^{m}(1 - K_{jj})\frac{(z_i'^2 - z_i^2)}{\sigma_i^2} \tag{5.19}$$

which is equivalent to:

$$K'_{ii} \geq K_{ii} + \frac{\sum_{j=1}^{m}(1 - K_{jj})(z_i'^2 - z_i^2)/\sigma_i^2}{\sum_{j=1, j\neq i}^{m}(z_j^2/\sigma_j^2) + z_i'^2/\sigma_i^2} \tag{5.20}$$

Theorem 3 demonstrates how much the attacker has to increase the value K_{ii} after the perturbation of a single measurement z_i. Note that in cases when the attacker wants to perturb multiple measurements, the attack can be conducted in a sequential manner and Theorem 2 is applied repeatedly in each step with the most updated K_{ii} and z. Also, since the maximum value of K_{ii} is 1, so when the calculated results indicate a required value which is greater than 1, that means only changing K_{ii} cannot mark the perturbation of z_i, and several other values of K_{jj}, $j = 1, ..., m, j \neq i$ should also be increased in order to meet Theorem 1.

One last question for the attacker would be how to actually increase the value of K_{ii}. The following theorem gives the answer.

Theorem 4 *Let K_{ii} be the ith diagonal element of hat matrix K defined in (5.11), then,*

$$(1 - K_{ii})^2 \leq \frac{\left\| \begin{bmatrix} \bar{H}_p \\ \bar{H}_f \end{bmatrix} \right\|_2^2}{\| \bar{H}_i^T \|_2^2}$$

where $\bar{H}_i$ is the ith row of $\bar{H}$, and $\bar{H}$ is partitioned as:

$$\bar{H} = \begin{bmatrix} \bar{H}_p \\ \bar{H}_i \\ \bar{H}_f \end{bmatrix}$$

Proof 5 *Since K is both symmetric ($K = K^T$) and idempotent ($K \cdot K = K$), then K is the orthogonal projector for col($\bar{H}$), which is the column space of $\bar{H}$. For any vector v, the projection of v gives the closest vector in col($\bar{H}$) to v, where closest is measured in Euclidean norm. That is,*

$$\| v - Kv \|_2^2 \leq \| v - u \|_2^2 \tag{5.21}$$

for all $u \in col(\bar{H})$.

Let $\hat{y} = Ke_i$ be the projection of e_i on the col($\bar{H}$), where e_i is the m dimensional vector with ith element equals to 1 and all the other elements are zeros. Then there exists a vector $\hat{t}$ such that

$$\hat{y} = \bar{H}\hat{t} = \begin{bmatrix} \bar{H}_p\hat{t} \\ \bar{H}_i\hat{t} \\ \bar{H}_f\hat{t} \end{bmatrix} \tag{5.22}$$

Note that $\hat{y}$ is the closest vector to e_i in col($\bar{H}$). That is, for any t,

$$\| e_i - \bar{H}\hat{t} \|_2^2 \leq \| e_i - \bar{H}t \|_2^2 \tag{5.23}$$

or equivalently,

$$\begin{aligned} &\| \bar{H}_p\hat{t} \|_2^2 + \| \bar{H}_f\hat{t} \|_2^2 + (1 - \bar{H}_i\hat{t})^2 \leq \\ &\| \bar{H}_p t \|_2^2 + \| \bar{H}_f t \|_2^2 + (1 - \bar{H}_i t)^2 \end{aligned} \tag{5.24}$$

Note that since $\hat{y} = Ke_i$, then with (5.22), we get $\bar{H}_i\hat{t} = K_{ii}$. Also set $t = \bar{H}_i^T / \| \bar{H}_i^T \|_2^2$, then $(1 - \bar{H}_i t)^2 = 0$. So from (5.24),

$$\begin{aligned} (1 - K_{ii})^2 &\leq \| \bar{H}_p t \|_2^2 + \| \bar{H}_f t \|_2^2 - \| \bar{H}_p\hat{t} \|_2^2 - \| \bar{H}_f\hat{t} \|_2^2 \\ &\leq \frac{\| \bar{H}_p\bar{H}_i^T \|_2^2}{\| \bar{H}_i^T \|_2^4} + \frac{\| \bar{H}_f\bar{H}_i^T \|_2^2}{\| \bar{H}_i^T \|_2^4} = \frac{\left\| \begin{bmatrix} \bar{H}_p \\ \bar{H}_f \end{bmatrix} \bar{H}_i^T \right\|_2^2}{\| \bar{H}_i^T \|_2^4} \\ &\leq \frac{\left\| \begin{bmatrix} \bar{H}_p \\ \bar{H}_f \end{bmatrix} \right\|_2^2 \| \bar{H}_i^T \|_2^2}{\| \bar{H}_i^T \|_2^4} = \frac{\left\| \begin{bmatrix} \bar{H}_p \\ \bar{H}_f \end{bmatrix} \right\|_2^2}{\| \bar{H}_i^T \|_2^2} \end{aligned}$$

From Theorem 4, it can be seen that the attacker can increase the value of K_{ii} by just increasing the l_2-norm of $\bar{H}_i^T$. Since $\bar{H}_i = 1/\sigma_i \cdot H_i$, then mathematically it gives three rules to increase the value of K_{ii}:

- **Rule 1**: Increase the absolute values of elements in H_i.
- **Rule 2**: Decrease the value of σ_i.
- **Rule 3**: Increase the number of non-zero elements in H_i.

5.2.2 Attacking Strategies in Smart Grid

In this part, we introduce the specific attacking strategies against the measurements in Smart Grid by applying the above principles.

Attacking Power Flow Measurement As shown in Figure 5.2, power flow measurement is the one placed between buses to monitor the power flow across the connecting branch. If an entry z_i of z is the measurement of the power flow from bus k to m, then $z_i = B_{km}(x_k - x_m)$, where B_{km} is the branch susceptance between bus k and m and x_k, x_m are the unknown voltage phase angles at bus k and m. The corresponding ith row of H is:

$$H_i = [0, ..., \underbrace{B_{km}}_{kth \text{ entry}}, 0, ..., 0, \underbrace{-B_{km}}_{mth \text{ entry}}, ..., 0] \tag{5.25}$$

If the attacker intends to alter the measurement from z_i to z_i' without being detected, he should

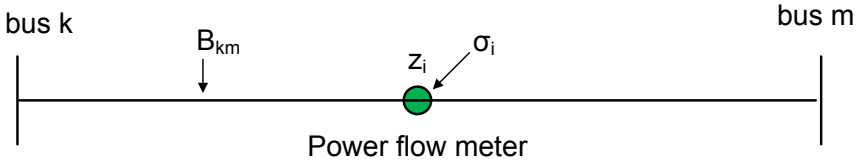


Figure 5.2 Power flow measurement

apply Theorem 3 first to figure out how much he has to increase the value of K_{ii}, then applies rule 1 and rule 2, which requires increasing the value of B_{km} and decreasing the value of σ_i.

Attacking Power Injection Measurement As shown in Figure 5.3, power injection measurement is placed at bus to monitor the power injection of the particular bus, typically from a load or synchronized generator. If z_i is the measurement of power injection at bus i, it is the sum of all the power flows along incident branches to that bus: $z_i = \sum_{j \in N_i} z_{ij}$, where N_i is the set of buses incident to i, e.g. , $k, m \in N_i$. Therefore, the corresponding ith row of H is the sum of all the row

vectors corresponding to the incident branch power flows, which is:

$$H_i = [0, ..., \underbrace{\sum_{j \in N_i} B_{ij}}_{ith \text{ entry}}, ..., \underbrace{-B_{ik}}_{kth \text{ entry}}, ..., \underbrace{-B_{im}}_{mth \text{ entry}}, ..., 0] \tag{5.26}$$

If the attacker intends to alter the measurement from z_i to z_i' without being detected, he applies

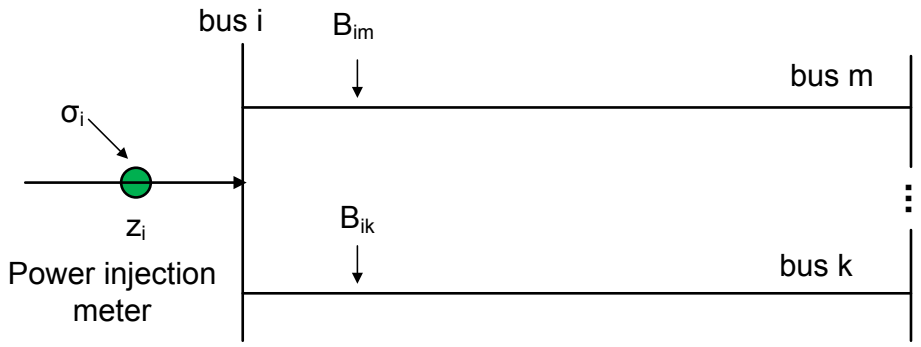


Figure 5.3 Attacking power injection measurement

Theorem 3 first to figure out how much he has to increase the value of K_{ii}, then he can apply rule 1 and rule 2 in this case. In addition, since the number of nonzero elements is related to the physical connections of buses in power network, rule 3 cannot be applied directly. However, it is still valuable since it suggests that the power injection measurements at buses with more incident branches are more vulnerable since they already have a large leverage in normal condition.

5.2.3 Remarks

Several facts are worth pointing out from the above analysis.

- First, since σ_i is the standard deviation of measurement error e_i, a smaller σ_i indicates a higher accuracy of the measurement z_i. This implies that high accuracy measurement is more likely to become a leverage point and attacking a higher accuracy device will actually have better chance of success.

- Second, increasing the susceptance of branches should be the first choice of the attacker

since it can affect the branch flow measurement and the power injection measurements at incident buses simultaneously.

- Third, the existences of leverage points in actual power system are very common. For instance, the IEEE 118-bus system has 7.2% of branches with relatively large susceptance(13 over 179) and 28% of buses with at least 4 incident branches (33 over 118) [63]. This suggests that even without the explicit creations of leverage points, the attacker can still launch LPAttack against the corresponding measurements.

- Finally, even though the above attacking rules and strategies are analyzed from the pure mathematical perspectives, they are exactly coincident with the actual leverage point situations [9] in power system, which includes: injection measurement placed at a bus which is incident to branches with large susceptance value (apply rule 1 to injection measurement); flow measurement along branches with very large susceptance value (apply rule 1 to flow measurement); Using a very large weight for a specific measurement (applying rule 2 to both flow and injection measurement);injection measurements placed at a buses which are incident to a large number of branches (applying rule 3 to injection measurement).

5.3 Countermeasure

We propose the countermeasure strategy based on robust state estimator that does not require any investments of securing hardware devices. Since all the attacking strategies are based on the creation of leverage points, the straightforward countermeasure would be first evaluating the leverages of measurements to identify the leverage points, then discards the corresponding measurements before entering WLS state estimator. However, since the ubiquitous existence of leverage points in power system and the leverage measurements could be good when there is no cyber attacks, the above approach would destroy a large amount of useful information and could even make the system unobservable. Therefore, a better solution should be replacing the WLS with a more robust state estimator, which is designed to automatically detect the leverage points and suppress the influence of the corresponding measurements on the state estimation.

Inspired by the works in [65] [66], we present the countermeasure based on the robust Schweppe-Huber Generalized-M (SHGM) estimator. We modified SHGM such that it possesses good robustness and efficiency against LPAttack. ω_i is specifically designed as the penalty factor to suppress the effects of leverage measurements. The details are as follows:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \rho(r) = \sum_{i=1}^{m} \rho(r_i) \\ \text{subject to} \quad & z = h(x) + r \end{aligned}$$

where $z, h(x)$ are the same as in (5.1), and $\rho(r_i)$ is a function of the measurement residual r_i, which is defined as:

$$\rho(r_i) = \begin{cases} \frac{1}{2} r_i^2 / \sigma_i^2 & |r_i/\sigma_i| \leq a \cdot \omega_i \\ a \cdot \omega_i |r_i/\sigma_i| - \frac{1}{2} a^2 \cdot \omega_i^2 & otherwise \end{cases}$$

where a is a constant ranging from 1 to 3 and ω_i is defined as:

$$\omega_i = \min\{1, [\frac{1 - K_{ii}}{K_{ii}}]\} \tag{5.27}$$

Note that when a is infinity, the SHGM is equivalent to WLS.

We now need to derive an algorithm that finds the solution. To this end, we propose an algorithm based on numerically stable iteratively re-weighted least squares method [67]. Writing the KKT necessary conditions for a minimum of $\rho(r)$:

$$\begin{aligned} \frac{\partial \rho}{\partial x} = \frac{\partial \rho}{\partial r} \cdot \frac{\partial r}{\partial x} = 0 \Rightarrow \sum_{i=1}^{m} \frac{\partial \rho}{\partial r_i} \cdot \frac{\partial r_i}{\partial x} = 0 \\ \Rightarrow \sum_{i=1}^{m} \Upsilon(r_i) \cdot H_i = 0 \Rightarrow \sum_{i=1}^{m} \frac{\Upsilon(r_i)}{r_i} \cdot r_i \cdot H_i = 0 \end{aligned}$$

where $\Upsilon(r_i) = \frac{\partial \rho}{\partial r_i}$. Write the above in matrix form, we have:

$$H^T \cdot Q \cdot r = 0 \tag{5.28}$$

where Q is a m dimensional diagonal matrix and $Q_{ii} = \frac{\Upsilon(r_i)}{r_i}$, defined as:

$$Q_{ii} = \begin{cases} \frac{1}{\sigma_i^2} & |r_i/\sigma_i| \leq a \cdot \omega_i \\ \frac{a \cdot \omega_i}{r_i \sigma_i} \cdot \text{sign}(r_i) & otherwise \end{cases} \tag{5.29}$$

Also, from the first order Taylor approximation, we have:

$$h(x) \approx h(x^k) + H \cdot \bigtriangleup x^k \tag{5.30}$$

where k means the kth iteration. Since $r = z - h(x)$, $r^k = z - h(x^k)$, combined with (5.28), we get the equation in kth iteration:

$$H^T \cdot Q \cdot H \bigtriangleup x^k = H^T \cdot Q \cdot r^k \tag{5.31}$$

Note that matrix Q is keeping updated based on the residual r of current iteration. The algorithm is given in detail in Algorithm 5.

Algorithm 5 Re-weighted Least Square Solver

1: Initial guess x_0.
2: $k = 0$.
3: Calculate ω_i based on (5.27).
4: **while** true **do**
5: $r^k = z - h(x^k)$
6: Update Q based on (5.29).
7: $\bigtriangleup x^k = (H^T \cdot Q \cdot H)^{-1} H^T \cdot Q \cdot r^k$
8: $x^{k+1} = x^k + \bigtriangleup x^k$
9: **if** $\mid x^{k+1} - x^k \mid < \varphi$(threshold) **then**
10: return
11: **else**
12: $k = k + 1$

5.4 Evaluation

In this section, we validate the proposed attacking strategy and countermeasure using IEEE 14 test system. The one-line diagram of the test system is displayed in Figure 5.4. It is provided

with 12 power injection measurements and 13 power flow measurements in normal steady state. In the following, IN i denotes a power injection measurement at Bus i and FL $i-j$ denotes a power flow measurement from Bus i to Bus j. The negative measurement means the power flow direction is the opposite to the assumed one. We extract the configurations and parameters of the IEEE test systems from MATPOWER 4.0 [68], a MATLAB package for solving power flow problems.

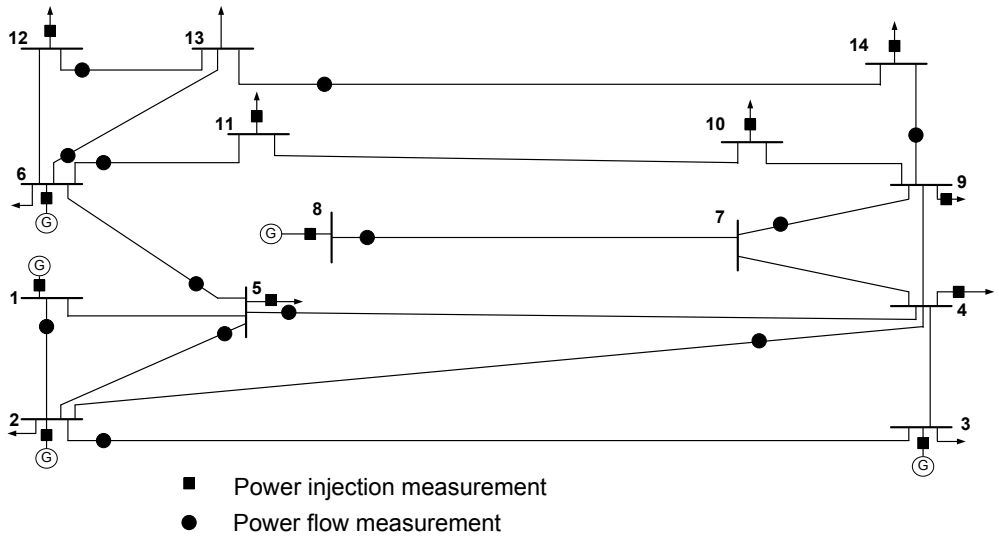


Figure 5.4 IEEE 14 bus test system

We first investigate the LPAttack principle in Theorem 2 by demonstrating the relationship between the perturbation of power measurement z_i and the corresponding required minimum increase of K_{ii}. Table 5.1 lists the results for several chosen power measurements in IEEE 14 test system. Note $\triangle z_i = \mid z_i' - z_i \mid$ and $\triangle K_{ii} = K_{ii}' - K_{ii}$. $\triangle z_i$ is set to be $0.5, 1, 2, 5, 10$ respectively. All numerical values of the measurements are in per-unit system with base value 100MVA(pu).

The results from Table 5.1 imply some interesting facts. First, as suggested by the bold numerical values, when the measurements like IN1 and FL1-2 are perturbed in a large magnitude($\triangle z_i = 5, 10$), the required increase of K_{ii} from Theorem 2 will result in a value greater than 1, which is impossible to achieve. In this case, it actually indicates that other diagonal values of matrix K should also be increased to satisfy Theorem 1 such that the measurement perturbation can be

Table 5.1 Leverage-Point Attack in IEEE 14 bus system

Meas.	IN1	FL1-2	IN4	IN5	IN6	FL4-5	FL5-6
Normal z_i	125.48	84.70	-27.39	-4.36	-6.42	-35.38	24.52
Original K_{ii}	0.3969	0.2171	0.6043	0.5536	0.5902	0.2179	0.1967
$\triangle K_{ii}(\triangle z_i$=0.5)	0.0637	0.0413	0.0140	0.0053	0.0034	0.0181	0.0126
$\triangle K_{ii}(\triangle z_i$=1)	0.1275	0.0864	0.0283	0.0101	0.0070	0.0364	0.0254
$\triangle K_{ii}(\triangle z_i$=2)	0.2553	0.1733	0.0567	0.0319	0.0151	0.0738	0.0518
$\triangle K_{ii}(\triangle z_i$=5)	**0.6395**	0.4380	0.0602	0.0348	0.0452	0.1914	0.1367
$\triangle K_{ii}(\triangle z_i$=10)	**1.2827**	**0.8908**	0.1143	0.0948	0.1156	0.4060	0.2976

marked. Second, we also discover that in general, when the original magnitude of measurement is large, the corresponding required $\triangle K_{ii}$ is relatively large. This indicates that the attack against measurements with smaller magnitude is easier to succeed. Finally, when the original value of K_{ii} is large, such as IN4 and IN5, the perturbations of corresponding measurements only require small increase in K_{ii}. This suggests that the attacks against measurements with larger K_{ii} are more prone to success. Figure shows the identified vulnerable meters in the IEEE 14 bus system.

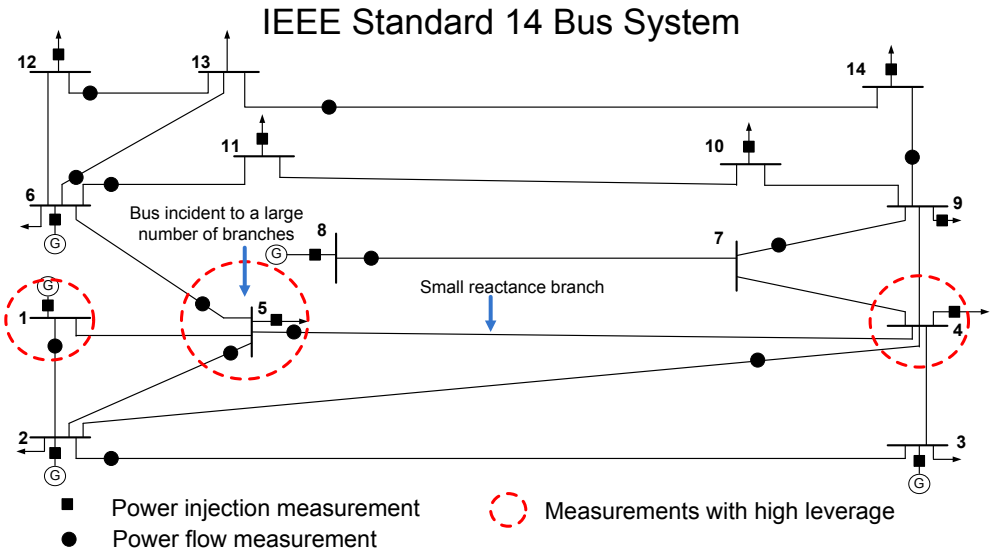


Figure 5.5 Most vulnerable meters in IEEE 14 bus system

Next we will study how the changes in parameters of branch susceptance and measurement error variances will affect the leverages of measurements. Figure 5.6 gives the result for IN5.

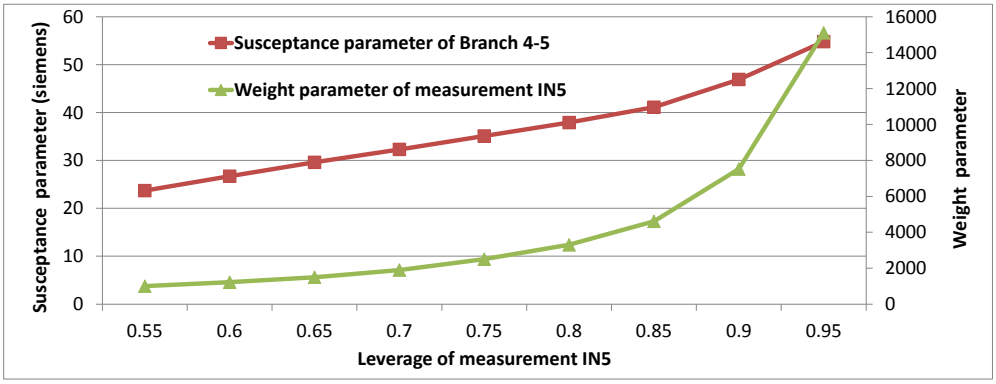


Figure 5.6 Relations between leverage of IN5 and value of parameters

Figure 5.7 shows the residuals after measurement perturbation in both WLS and our countermeasure. We can see that the residual in countermeasure is significantly larger than in WLS, and it can correctly fire the bad data detection alarm in most cases. The a in countermeasure is set to be 2.4 in our simulations.

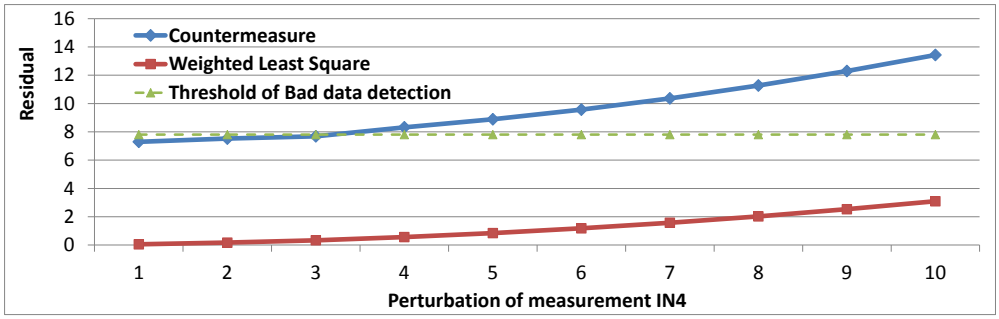


Figure 5.7 Residual in conventional WLS and our proposed countermeasure

Finally, the effect of LPAttack on power market, particularly on Locational Marginal Price (LMP), is examined. LMP is the core variable in market operations and obtained through the real-time pricing models [69]. The real-time pricing models are built on the power flow and network topology measurement results given by the state estimation process, thus our proposed leverage-point attacks would directly affect the LMPs by manipulating the power flow measurements. We adopt the Ex-post pricing model (e.g., in ISO New England, PJM, and Midwest ISO) in [70] and conduct the sensitivity analysis of LMP at each of the 14 buses with respect to perturbations in

different power flow measurements. Figure 6.22 shows the LMP sensitivities of all 14 buses with respect to the changes in three measurements: IN1, IN4 and FL5-6. The unit on vertical axis is ($/MWh)/(puMVA). We can see that the perturbation in measurements would have greater impact on the LMPs of nearby buses than other buses. For example, perturbation of IN4 has larger impact on LMPs of bus 4,5 and almost no impact on bus 13,14. Also, the perturbation in one measurement could yield either positive or negative sensitivities to LMPs at different buses.

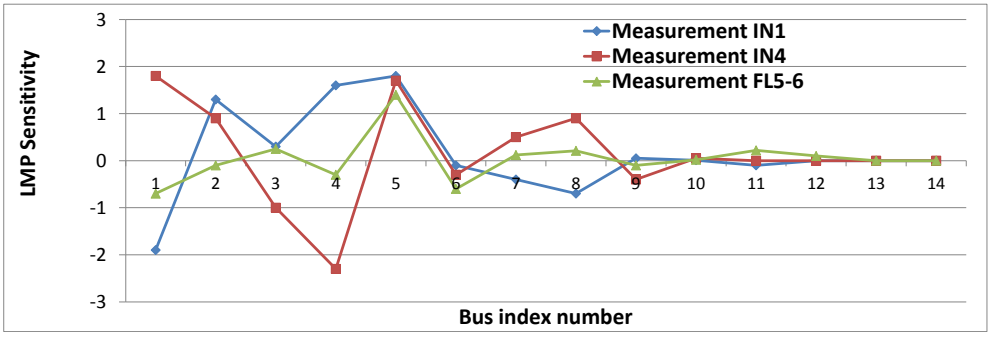


Figure 5.8 LMP sensitivities of all buses with respect to measurements

CHAPTER 6

SCOREPLUS: A SOFTWARE-HARDWARE HYBRID AND FEDERATED EXPERIMENT ENVIRONMENT FOR SMART GRID

In this chapter, we present ScorePlus [8], a Software-Hardware Hybrid and Federated Experiment Environment for Smart Grid.

6.1 Overall System Design

Figure 6.1 demonstrates the overall architecture of our platform. ScorePlus consists of Graphical User Interface (GUI), software emulator, and hardware testbed. The GUI connects with software emulator and hardware testbed remotely through Internet. Since the software emulator and the hardware testbed expose the same architecture and interfaces to the GUI, the user can run the same Smart Grid application test case on either of them without any migration issue. Meanwhile, by integrating the communication network and power network from both software emulator and hardware testbed, multiple software emulators and hardware testbeds are able to connect and interact through Internet. Upon specifying the connection interfaces between each other, communication and power flow could be established between each individual node, such as a virtual demander in software emulator and a real supplier in hardware testbed.

6.2 Software Emulator

As shown in Figure 6.2, the software emulator consists of Service Layer, Virtual nodes, Linux Ethernet Bridging, Communication Module, and Power Module. The Service Layer is essentially a socket server that provides various event handlers to the formatted messages from external interacting system. It is responsible for initializing the emulation case, collecting and forwarding the system request, and managing multiple emulation sessions, etc. The software emulator partially leverages our previous development in [71]. In this work, we specifically implement the network

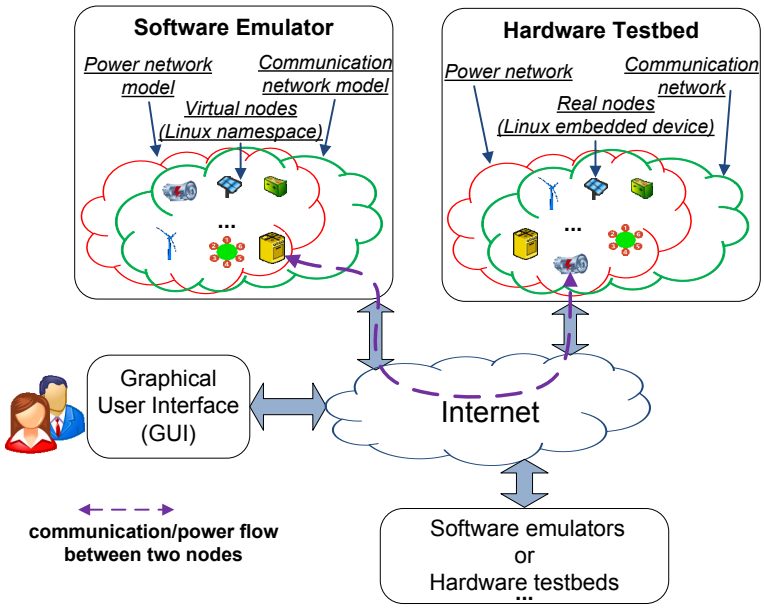


Figure 6.1 Overall architecture of ScorePlus

tunnels between virtual network interfaces and physical network interfaces, and upgrade the power network model through domain decompositions, and increase more message handles in the Service Layer to respond to external interactions, etc. All the improvements in the software emulator are intended to facilitate its integration with the hardware testbed.

6.2.1 Virtual Nodes: Light Weighted Virtualization

The emulation feature of software emulator is implemented using Linux network namespace techniques, which is a recent light weighted paravirtualization technique supported by mainstream Linux kernel. By calling the clone() system call, each created virtual node can have its own instance of Linux OS network stack and process space while sharing the same local file systems and hardware with other virtual nodes. From the perspective of codes running inside, each virtual node is just another piece of hardware platform controlled by Linux OS. Therefore, the virtual nodes can directly execute unmodified Smart Grid application codes from physical Linux-embedded devices, and vice versa. Figure 6.3 illustrates the software emulator scalability that about 180 virtual nodes can be created on a 64 bits HP desktop with Pentium(R) Dual-Core CPU E5700 @ 3.00GHz and 4GiB memory.

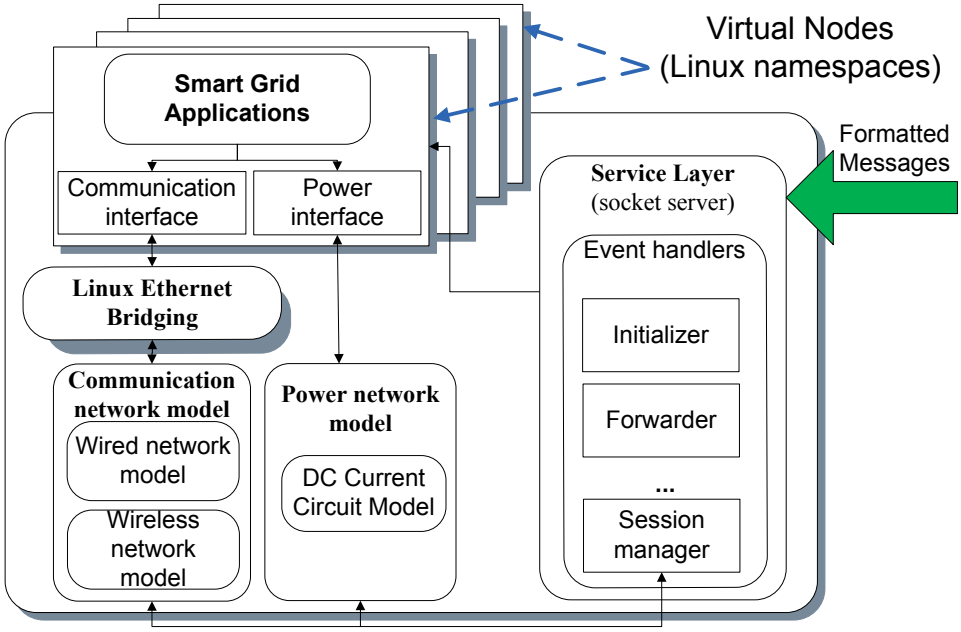


Figure 6.2 Design of Software Emulator

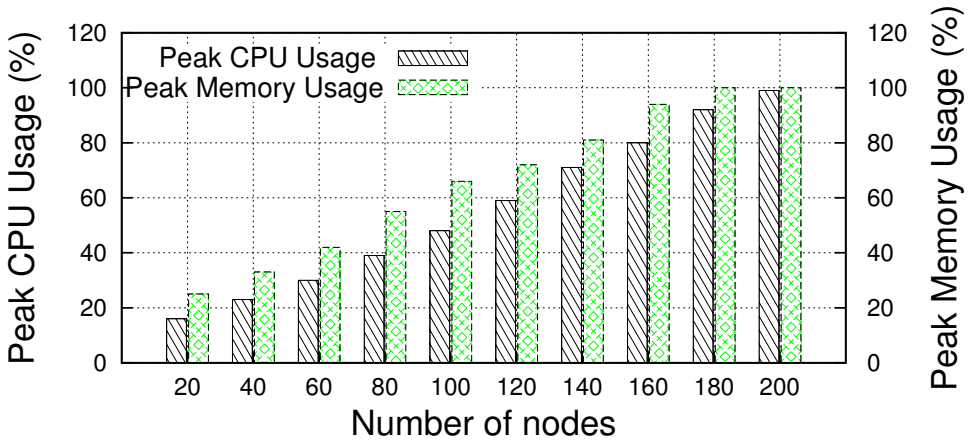


Figure 6.3 Scalability of Software Emulator

6.2.2 Linux Ethernet Bridging and Communication Module

All the virtual nodes in the emulator are equipped with virtual Ethernet interfaces and they are linked by Linux Ethernet Bridging. This approach provides underlying communication capabilities between each virtual node. Based on Linux Ethernet Bridging, the communication module in software emulator provides comprehensive support of various wired and wireless communication network models. All the models are simply the manipulations of the underlying communication infrastructure. More importantly, virtual Ethernet interfaces can be directly mapped to a physical Ethernet interface on the emulation host, such that all the traffic passing through that physical port can be transferred to the emulation environment. This enables the interactions between the software emulation environment with outside physical networks.

We employ the above key feature to achieve the integration of communication networks between the software emulator and hardware testbed, which will be presented in Section V.

6.2.3 Power Module

The power module in software emulator emulates the power flows analysis within Smart Grid and also provides implementations of pre-defined energy models. The power module receives initial power network topology, energy model configuration information and the connection interface information from service layer to formulate the power network model.

Power Network Model: General Description The power network model is a DC current model to emulate DC power flow analysis. Assume a power grid is composed of n nodes and b branches. Since the power network dynamics is subject to Kirchhoff's current and voltage laws (KCL and KVL), in order to calculate the voltages of all nodes, we apply nodal analysis to the grid and get the linear equations for the whole system:

$$AV = I \tag{6.1}$$

where coefficient matrix A is the $(n-1)\times(n-1)$ reduced nodal admittance matrix since we have chosen a reference node. Let $Nbr(i)$ represents the neighbour set of node i in the power network, we get:

$$a_{ij} = \begin{cases} \sum_{s\in Nbr(i)} g_{is} & i = j. \\ -g_{ij} & j \in Nbr(i) \\ 0 & otherwise \end{cases} \tag{6.2}$$

g_{ij} is the admittance between node i and node j, V and I are the unknown node voltage vector and the known nodal current injection vector, respectively. Figure 6.4 shows the data flow diagram of power module.

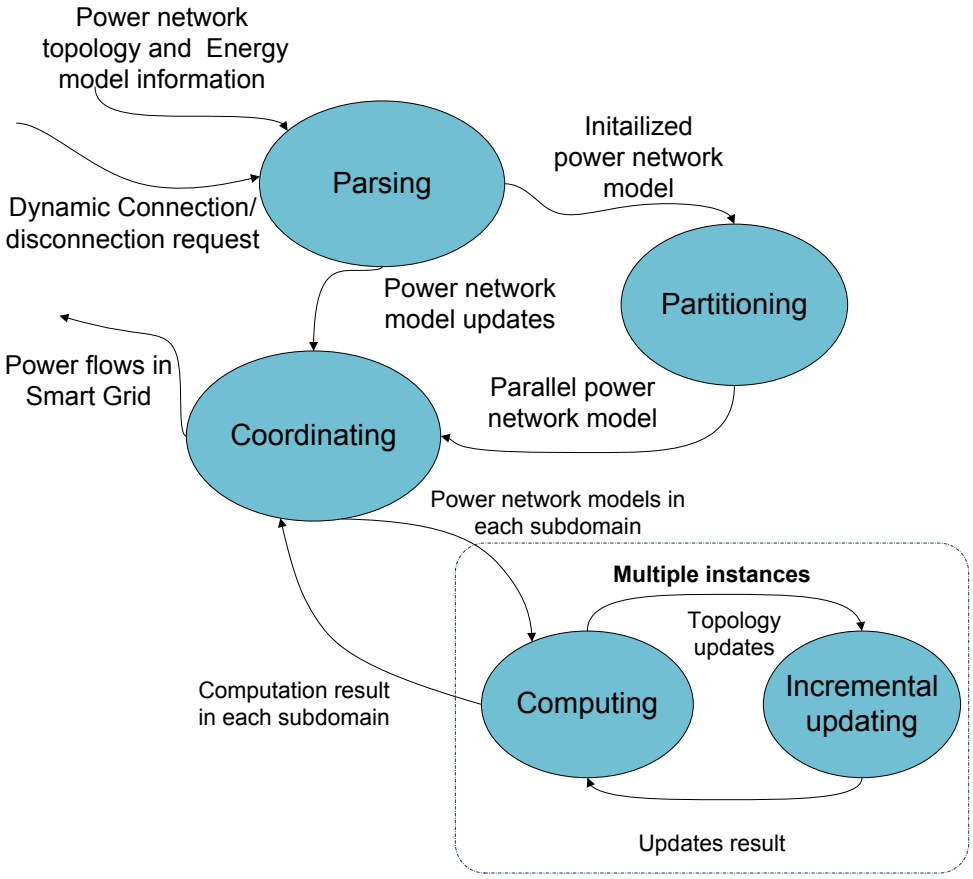


Figure 6.4 Data flow diagram of Power Module

Incremental updating Based on previous model, let's consider the situation when the power network topology changes. Suppose the power grid status changes, such as the admittance between node i and node j is changed by Δg_{ij}. So the new coefficient matrix $\widetilde{A}$ can be written as:

$$\widetilde{A} = A + U\Delta g_{ij}U^T \tag{6.3}$$

where

$$U = \begin{bmatrix} 0 & \cdots & 1 & \cdots & -1 & \cdots & 0 \\ & & i & & j & & \end{bmatrix}^T$$

Particularly, the changed admittance Δg_{ij} equals to $-g_{ij}$ when the branch is removed and $\Delta g_{ij} = g_{ij}$ when a new branch is added. Notice that [72]

$$\widetilde{A}^{-1} = A^{-1} - A^{-1}U\left(\Delta {g_{ij}}^{-1} + U^TA^{-1}U\right)^{-1}U^TA^{-1} \tag{6.4}$$

then we can get the $\widetilde{A}^{-1}$ with a much lower computation cost when we store previously computed A^{-1}.

Power Network Model: Domain Decomposition Power network is generally a network of loosely coupled sub power networks. Each sub network is a relatively independent partition of the whole energy system and only few in-between connection lines join them together. Inside each sub network, we divide the nodes into two sets:

- Internal nodes: nodes that only have connections with the nodes inside the same sub network.
- Boundary nodes: nodes that have connections with the nodes in other sub networks.

The architecture of the power network is illustrated in Figure 6.5. Based on the previous analysis, we apply the Schur complement domain decomposition method [73] to our power network model. Specifically, suppose there are k sub networks, by grouping the internal nodes of each sub network and putting all the boundary nodes of the network in the back, we formulate the nodal analysis

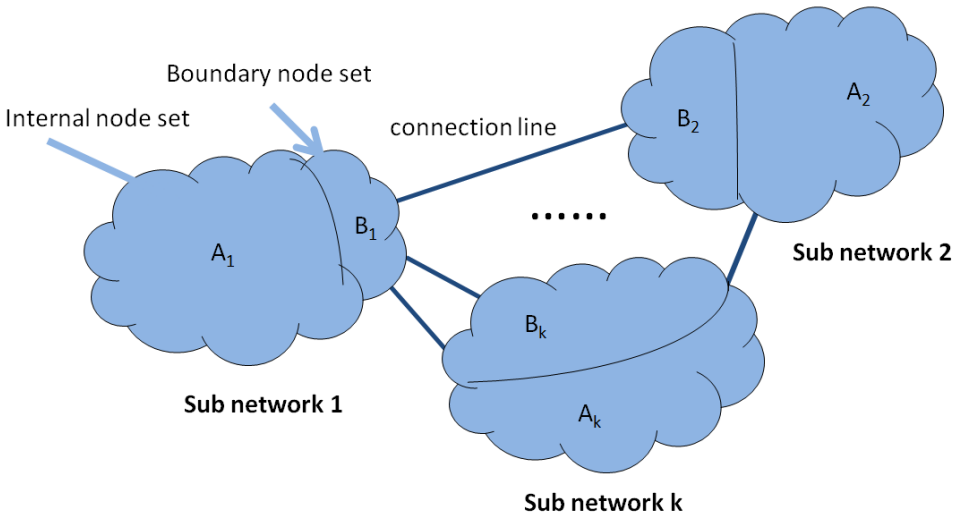


Figure 6.5 The general architecture of power network

model for the whole power network as the following:

$$\begin{bmatrix} Y_{A_1A_1} & 0 & \cdots & 0 & Y_{A_1B} \\ 0 & Y_{A_2A_2} & \cdots & 0 & Y_{A_2B} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & Y_{A_kA_k} & Y_{A_kB} \\ Y_{BA_1} & Y_{BA_2} & \cdots & Y_{BA_k} & Y_{BB} \end{bmatrix} \begin{bmatrix} V_{A_1} \\ V_{A_2} \\ \vdots \\ V_{A_k} \\ V_B \end{bmatrix} = \begin{bmatrix} I_{A_1} \\ I_{A_2} \\ \vdots \\ I_{A_k} \\ I_B \end{bmatrix} \tag{6.5}$$

Notice that B is the set of all boundary nodes in the whole network, consisting of $B_1, B_2, ..., B_k$. Therefore, Y_{A_iB} only has non zero entries in its submatrix $Y_{A_iB_i}$, for all $i = 1, 2, ...k$.

From (6.5), if the voltages for boundary nodes set V_B is known, then the voltages for the nodes in each sub network can be calculated as the following:

$$Y_{A_iA_i}V_{A_i} = I_{A_i} - Y_{A_iB}V_B, \forall i \in \{1, 2, ..., k\}. \tag{6.6}$$

Meanwhile, if we keep the corresponding part for the boundary node set B in equation (6.5), we can get:

$$\widetilde{Y_{BB}}V_B = \widetilde{I_B} \tag{6.7}$$

where

$$\widetilde{Y_{BB}} = Y_{BB} - \sum_{i=1}^{k} Y_{BA_i} {Y_{A_iA_i}}^{-1} Y_{A_iB} \tag{6.8}$$

$$\widetilde{I_B} = I_B - \sum_{i=1}^{k} Y_{BA_i} {Y_{A_iA_i}}^{-1} I_{A_i} \tag{6.9}$$

Define

$$x_i = Y_{BA_i} {Y_{A_iA_i}}^{-1} Y_{A_iB} \tag{6.10}$$

$$y_i = Y_{BA_i} {Y_{A_iA_i}}^{-1} I_{A_i} \tag{6.11}$$

for all $i = 1, 2, ...k$. Notice that x_i and y_i only requires local information for sub system i.

We employ the above key feature to achieve the integration of power networks between the software emulator and the hardware testbed, which treats each of them as a sub network of the whole power network. The details for that would be given in section V.

6.3 Hardware Testbed

Figure 6.6 shows the design of our hardware testbed, which follows the same architecture as the software emulator. Each node in the hardware testbed is a physical energy devices emulating an energy entity in Smart Grid system. Through communication interface and power interface, all the energy devices are connected by communication network and power network.

6.3.1 Overview of Energy Devices

The hardware testbed is composed of the following energy devices to emulate the energy entities in a Smart Grid system:

- 1 Supplier (This device emulates a general power generation. Output to the Smart Grid is up to 200mA. The Smart Grid voltage is typically 2.5V, but it may be higher or lower when source and load are unbalanced.)
- 5 Solar Panel Controller (Output is up to 30mA each, depending on light intensity.)
- 5 Wind Turbine Controller (Output is up to 30mA each, depending on wind intensity.)

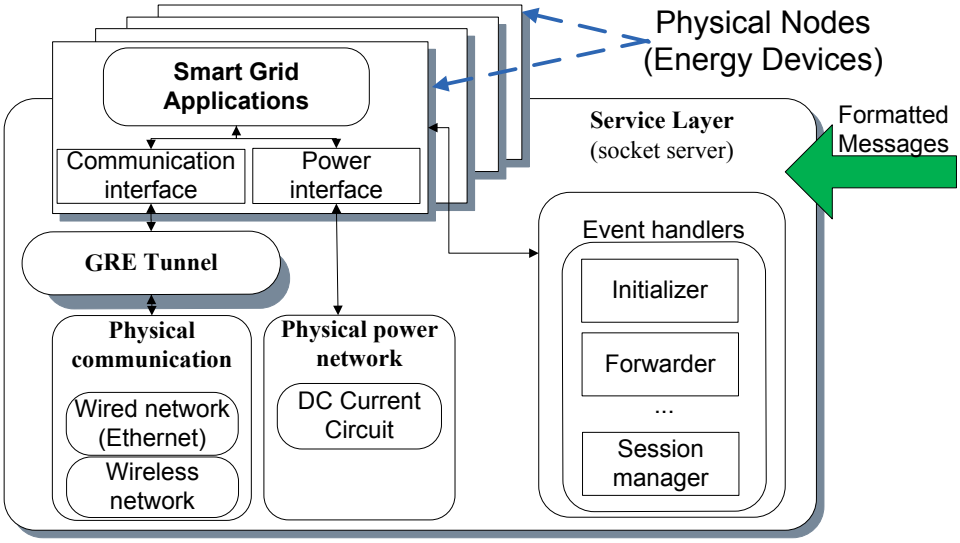


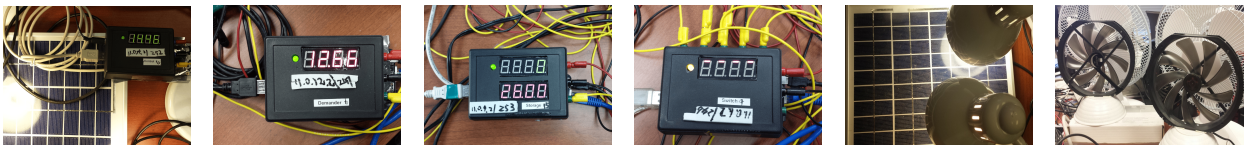
Figure 6.6 Design of Hardware Testbed

- 5 Storage (Capacitor bank with electronics. It can source or sink up to 50mA depending on the demands of the Smart Grid system.)
- 15 Demander (A load only, drawing up to 20mA each from the Smart Grid.)
- 5 Topology Switch (Has 6 ports and can switch current flow in multiple ways. Must be able to handle up to 300mA on all ports.)
- 1 Interface device, which serves as the energy tunnel when the hardware testbed is connected with the software emulator.

Figure 6.7 shows various energy devices, solar panel and wind turbine in use. The LCD display shows the current drawn and sourced by each energy device except the one in Topology Switch Device, which displays the connection status between all the 6 ports. Figure 6.7 shows the energy devices, the solar panel and the wind turbine when they are in use in the hardware testbed.

6.3.2 Energy Device Design Details

Each of the above energy device includes three boards: a Beagleboard [74], a Telosw board [75] and an Energy board. Figure 6.8 shows the process of remote access and configuration of these



(a) Solar Panel Controller (b) Demander (c) Storage (d) Topology Switch (e) Solar Panel (f) Wind Turbine

Figure 6.7 Hardware Testbed

energy devices from GUI by the user. The user specify the hardware configurations from the GUI and send requests to the Service Layer as formatted messages. Upon receiving and mapping these messages, the Service Layer forwards the corresponding request to the Beagleboard of the designated energy device. Then the control program in that Beagleboard then will communicate with its Telosw board, which ultimately interacts with the Energy board and put those configurations into effect, such as load of the demander, output of the renewable controller and connection status of the topology switch, etc. Likewise, the status of the energy devices are periodically queried and reported to GUI backward. The roles of the above three boards are summarized as the following:

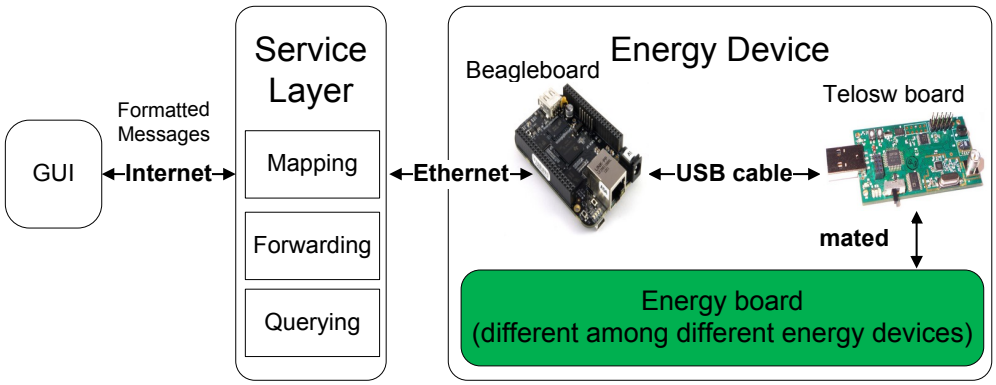


Figure 6.8 Remote access and configuration of energy devices

Beagleboard

- Interact with the Service Layer and control the behavior of each energy device correspondingly.
- Provide Linux-based environment in each energy device to test the Smart Grid applications.

- Enable the communications between each energy device, through both wired and wireless networks.

Telosw board

- Adjust of smart resistors to emulate power profiles.
- Monitor/measure power generation and consumption rate of each energy device.
- Provide I/O to mating Energy board for LED and display.

Energy board

- Provide different DC current circuit to fulfill the corresponding power requirements of different devices.

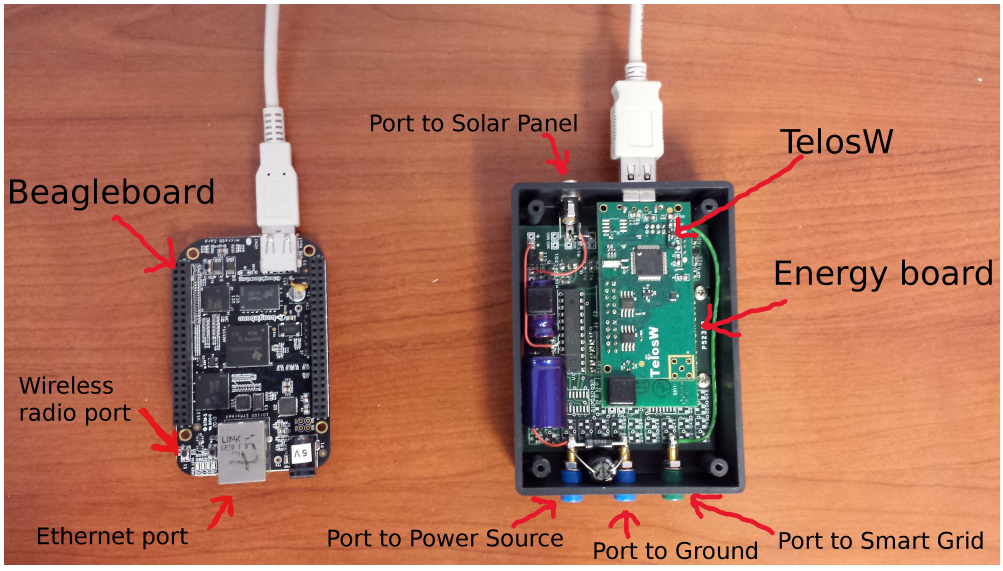


Figure 6.9 Details of Solar Panel Controller

Figure 6.9 shows the architecture of a Solar Panel Controller. The Beagleboard provides physical communication interfaces, such as the Ethernet port and mini USB to connect the wireless radio. It is connected with TelosW board through USB cable such that the programs running in the Beagleboard can directly access and control the energy profile through the TelosW. The TelosW is

mated to the Energy board, which are all enclosed in a black plastic container. The port at the top of the container is connected with a physical solar panel. There are three ports at the bottom of the container. The left one serves as the power source of the device and the right one is the actually power interface for Smart Grid system. The middle port is connected to the Ground.

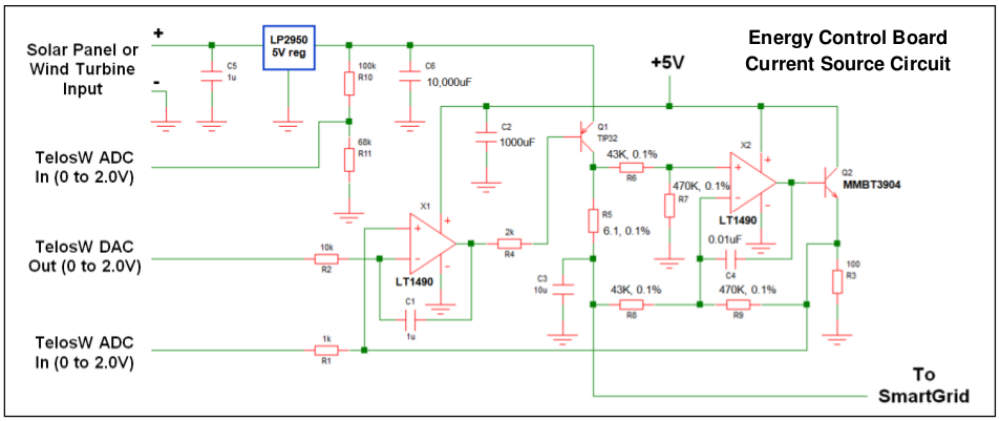


Figure 6.10 Schematics of Energy Board on Solar Panel Controller

Figure 6.10 shows the schematics for the Energy board on Solar Panel Controller. An analog voltage from the TelosW board indicates the amount of current output to the Smart Grid that is desired (0 to 2.0V corresponds to 0 to 30mA of desired output). An analog voltage to the TelosW board indicates the measured current output (0 to 2.0V corresponds to 0 to 30 mA of actual output current). The Solar Cell input to the Energy Control board will have a capacitor of 10,000uF (C6) for the purpose of temporarily storing energy from the generating source. 0.4 times the voltage on this capacitor will be supplied to the TelosW board for the purpose of indicating energy capability of the Solar Cell. The same device can be used for wind turbine control. Figure 6.11 shows a sample current output of a Solar Panel Controller device when the solar panel is exposed to a 50W 3-level light intensity lamp over time. Figure 6.12(a) shows the relationship between voltage input via current output in Controller device. Figure 6.12(b) shows the sreenshot of the ADC output voltage of the Solar Panel Controller in oscilloscope. The voltage output is very stable and no oscillation occurs.

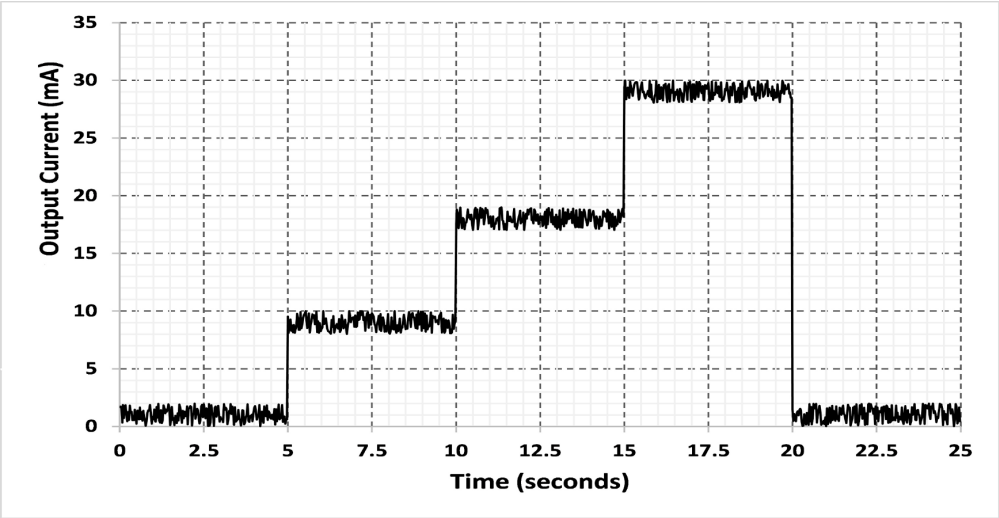


Figure 6.11 Sample Current Output of Solar Panel Controller

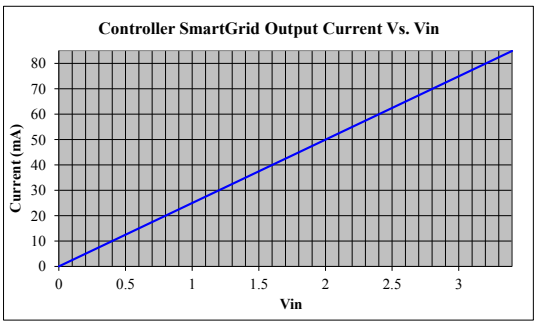
6.3.3 Power Network in Hardware Testbed: Dynamic Topology Configuration

The power network in hardware testbed connects all the energy devices through power interfaces through power cables. To facilitate the dynamic configurations of various power network topologies, we design the Topology Switch device as Figure (6.7(d)), which serves as the current hub for all the other energy devices. The Topology Switch can be accessed and configured dynamically through TelosW by the programs running inside the mated Beagleboard, such that the connection status between the 6 ports can be changed accordingly. Different power network topologies can be set up based on the users' requirements. The connection status is also visualized through the LCD. The Energy board of Topology Switch is shown in Figure 6.13.

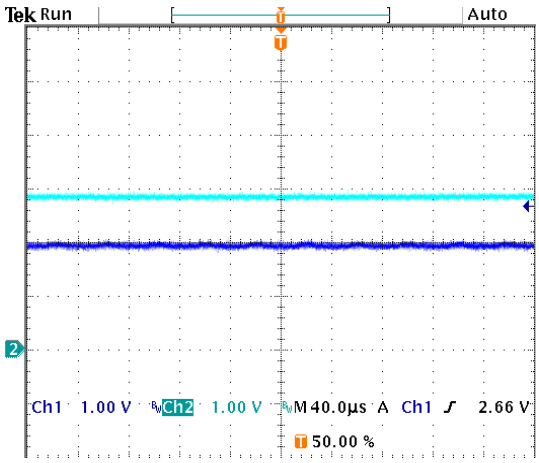
6.3.4 Communication Network in Hardware Testbed: Wire and Wireless Network

The communication network in hardware testbed are set up to emulate the wire and wireless communication in Smart Grid. The Generic Routing Encapsulation (GRE) tunnels are created between each energy device above the underlying communication network, such that the physical communication network can be unified with the ones in software emulator. For wired network, we use Ethernet to connect each energy device.

For wireless network, we are employing the WISP-2 outdoor antenna from ALFA Network,



(a) voltage input via current output



(b) ScreenShot in Tek Oscilloscope

Figure 6.12 Evalution of Solar Panel Controller

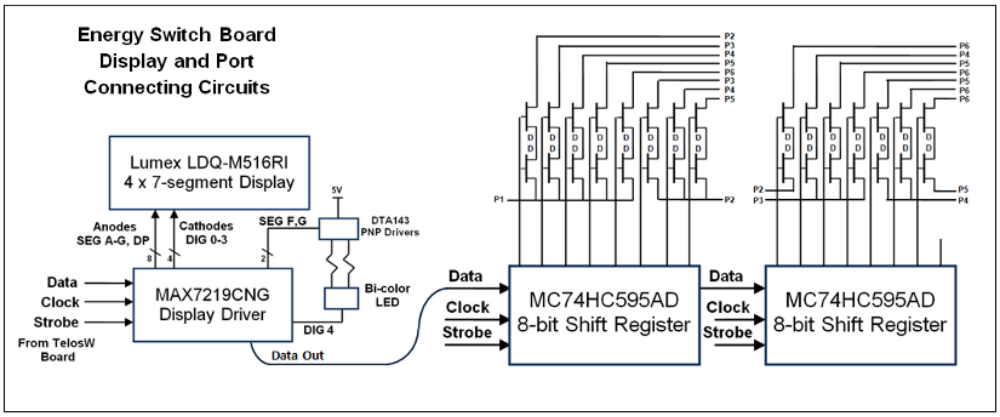


Figure 6.13 Design of Energy board for Topology Switch

Inc, which can be directly connected with Beagleboard through USB cable. WISP-2 is a Low-Cost IEEE 802.11n outdoor AP/CPE operating in 2.4GHz band that is also compliant with the standard IEEE 802.11b/g. In our indoor environment, the radio range of each node covers all the other nodes, so physically, all nodes can communicate with each other. To formulate a wireless mesh network with a specific topology, the iptables tool chain is employed within each energy device. Through ipables, we can filter the sent and received packets for each node, such that they can only communicate with the nodes as specified. Then the corresponding wireless network topology is formulated. The performance testing result from iperf tool are listed in Table 6.1.

Table 6.1 Performance of wireless network in hardware testbed

hops	1	2	3	4	5
bandwidth	1.05 Mbps	624 Kbps	416 Kbps	316 Kbps	242 Kbps
jitter	1.83ms	37.14ms	55.91ms	83.26ms	122.55ms

Table 6.2 Optimal attack vector a against IEEE14 with different sizes of ζ_A

size of ζ_A	optimal attack vector a
2	(0,63.0),(2,34.4)
4	(0,65.1),(2,32.0),(14,48.5),(34,-64.0)
6	(0,69.3),(2,32.0),(3,-48.0),(14,-48.0),(15,32.0),(34,-64.0)
8	(0,79.0),(2,32.0),(3,-32.0),(4,-48.0),(5,32.0),(14,52.3),(15,32.0),(34,-64.0)
10	(0,103.0),(2,32.0),(3,-48.0),(4,-64.0),(5,32.0),(14,68.0),(15,34.0),(17,32.0), (34,-64.0),(35,-33.0)

6.4 Integrating Software Emulators and Hardware Testbeds

ScorePlus supports scalable distributed experiments such that multiple software emulators and hardware testbeds running at different (local or remote) locations are able to connect and form a larger Smart Grid system. Here we only present the integration between software emulator and hardware testbed. Note integrations only among software emulators or only among hardware testbeds follow the same mechanism. In order to enable the interactions between the software emulator and the hardware testbed, both the communication network and power network within them should be integrated.

6.4.1 Integrating the Communication Network

The communication network emulated in software emulator runs in real time, so they can be connected to live physical networks. We build GRE tunnels between/among software emulation servers and the energy devices in hardware testbed. Figure 6.14 illustrates the details of how a virtual node in software emulator sends a packet to a physical node in hardware testbed. GRE tunneling is built between the software emulation server and the gateway machine of the hardware

testbed. The actual IP address of the two are in domain 131.96.x.0/24. When the packet reaches the edge of the network in software emulator, which is the physical network interface of emulation server, the tunnel entry in routing table would enable the encapsulation of the packet with GRE header and the tunnel destination, such that they can reach the hardware testbed environment. Also, by using this approach, all the ip addresses in the experiment environment are in 10.0.x.x/16 domain, and the physical networks can be abstracted as needed for the experiments.

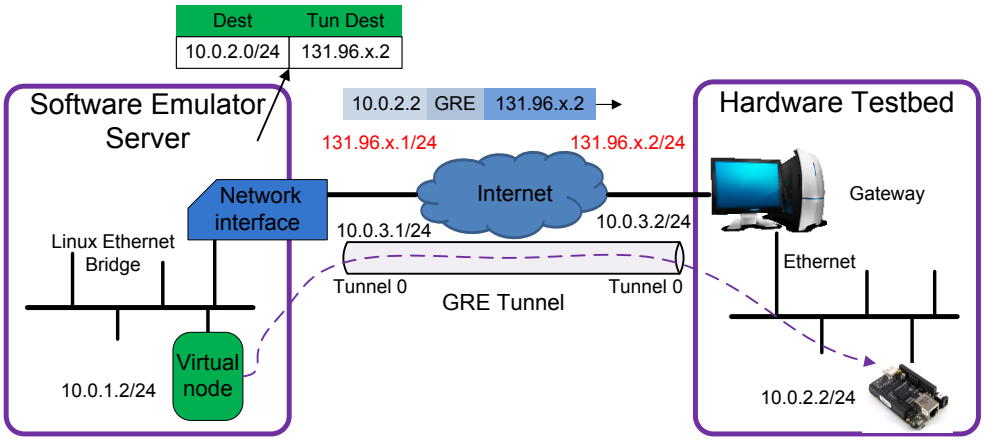


Figure 6.14 Communication between virtual node and real node through GRE tunneling

Figure 6.15 presents the performance of integrated communication network by showing the time elapsed for Open Shortest Path First (OSPF) routing protocol to converge. We employ different numbers of hardware testbed nodes and record time under different network delays set up in software emulator. As shown in Figure 6.15, when hardware testbed node is integrated within the communication network of the software emulator (the number of real nodes from 0 to 2), there is a dramatic increase in elapsed time. This indicates the GRE tunneling does impose a significant overhead. When the network delay in software emulator is increased to be comparable to the one in hardware testbed, which is 10ms this case, we see that the number of hardware testbed nodes involved in experiment doesn't really affect much the convergence speed of routing. The virtual and physical network gives about the same performance.

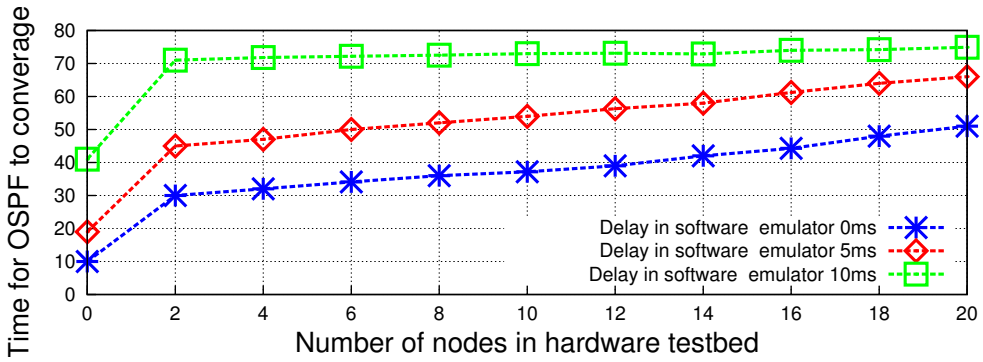


Figure 6.15 Time elapsed for OSPF to converge

6.4.2 Integrating the Power Network

The power network integration employs the domain decomposition model in Figure 6.5. Instances from either hardware testbeds or software emulators are treated as sub networks of whole power network.

Specifically, suppose there are already k connected instances of software emulators and hardware testbeds, and another instance $k+1$ joins in run time. Also assume that the $k+1$ instance connects with the instance set E directly, $E \subseteq \{1, 2, ..., k\}$. Then the composition process for each computation host i, $i = 1, 2, ..., k, k+1$, is executed as the following:

- If $i \in E$, then adjust the boundary node set B_i by adding the new boundary nodes connected with instance $k+1$ and also adjust the internal node set A_i by removing the corresponding boundary nodes connected with instance $k+1$. Compute x_i and y_i based on equations (6.10) and (6.11) respectively. Send the results to the coordinator host.

- The coordinator first reforms the boundary node set B by adding the new boundary nodes in $B_i, i \in E$ and B_{k+1}, then rebuilt Y_{BB} and I_B for the whole system. Secondly, it collects x_i and y_i from each host, and calculate V_B based on equations (6.7) (6.8) (6.9). Finally, it sends $Y_{BB}V_B$ back to each host.

- Each host i receives $Y_{BB}V_B$ from the coordinator and calculate V_{A_i} based on equation (6.6).

After the above process, each sub power network can set its updated status based on the calculated

results. If the sub network is a software emulator, we can update the status in our programs directly to achieve the resulting effects. However, if the sub network is a hardware testbed, physical changes must be made. To this end, we employ our specifically designed Interface device, which is essentially a combined supplier and demander with large capacity. When the calculated result indicates the hardware testbed sub network is requesting power from outside domains, the Interface device is set as a power supplier to provide corresponding power to the testbed. When the calculated result indicates the hardware testbed sub network is providing power to outside domains, the Interface device is set as a power demander to absorb power from the testbed correspondingly.

When one sub network is disconnected with the rest of the system, the steps are similar with the above except that instead of adding boundary nodes to the boundary set, the coordinator will remove the boundary nodes related with the exiting instances.

6.5 Deployment Plugin for ScorePlus in OpenStack Cloud Computing Platform

ScorePlus employs Linux containers (LXC) to achieve virtualizations in software emulator. As shown in Figure 6.3, a general PC can support test cases with at most 180 virtual nodes. As the scale of the test case grows, it becomes indispensable to deploy the software emulator in cloud computing infrastructure to increase its scalability. Moreover, when setting up multiple software emulation instances with different configurations, it would be convenient to just specify different parameters somehow without starting from scratch each time. To facilitate these deployment processes, we particularly implement a deployment plugin for ScorePlus in OpenStack cloud computing platform, which is essentially a *resource plugin* in Heat, the OpenStack Orchestration service.

6.5.1 OpenStack and Heat

OpenStack is a free and open-source cloud computing software platform, which is a global collaboration of developers and cloud computing technologists to produce the open standards for both public and private clouds. The OpenStack software consists of a group of interrelated projects to control various aspects of cloud infrastructure, such as authentication, orchestration, computing, networking, and storage etc.

Heat is the main project in OpenStack Orchestration program, which provides a template based orchestration to describe a cloud application. A Heat template allows instantiations of various OpenStack resource types, such as instances, floating ips, volumes, security groups, and users, etc. All the resources allocated for a Heat template are managed as a single *stack*. More importantly, Heat allows developers to add customized resource plugins, such that new resources can be integrated into the OpenStack frameworks to create cloud applications.

6.5.2 Implementation of Heat Plugin for ScorePlus

We design and implement a customized Heat resource plugin for ScorePlus in OpenStack, such that ScorePlus can be deployed and reused the same as any other resources in OpenStack cloud applications. In this way, ScorePlus can run on different computing, networking, and storage resources with different configurations, which are all based on the specifications in a Heat template. The resource plugin extends a base Resource class, and the lifecycle is managed by a series of rele-

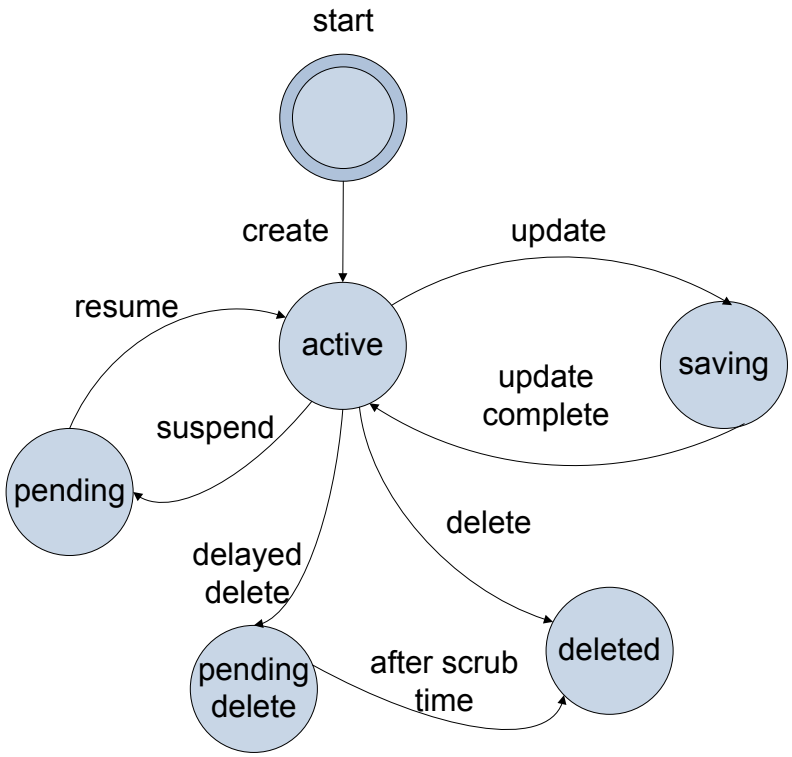


Figure 6.16 State Transition Diagram of ScorePlus Resource Plugin in OpenStack

vant handler methods, including create, update, suspend, resume and delete. Figure 6.16 illustrates the state transition diagram of ScorePlus resource plugin within Heat engine. Specifically, the following key methods extended from the heat.engine.resource.Resource class are implemented to handle the entire lifecycle:

- handle_create: Deploy ScorePlus on the dependent server and start the ScorePlus service socket.
- handle_update: Update current ScorePlus instance.
- handle_suspend: Pause the ScorePlus service socket process.
- handle_resume: Resume the ScorePlus service socket process.
- handle_delete: Remove ScorePlus related process and software on the dependent server.

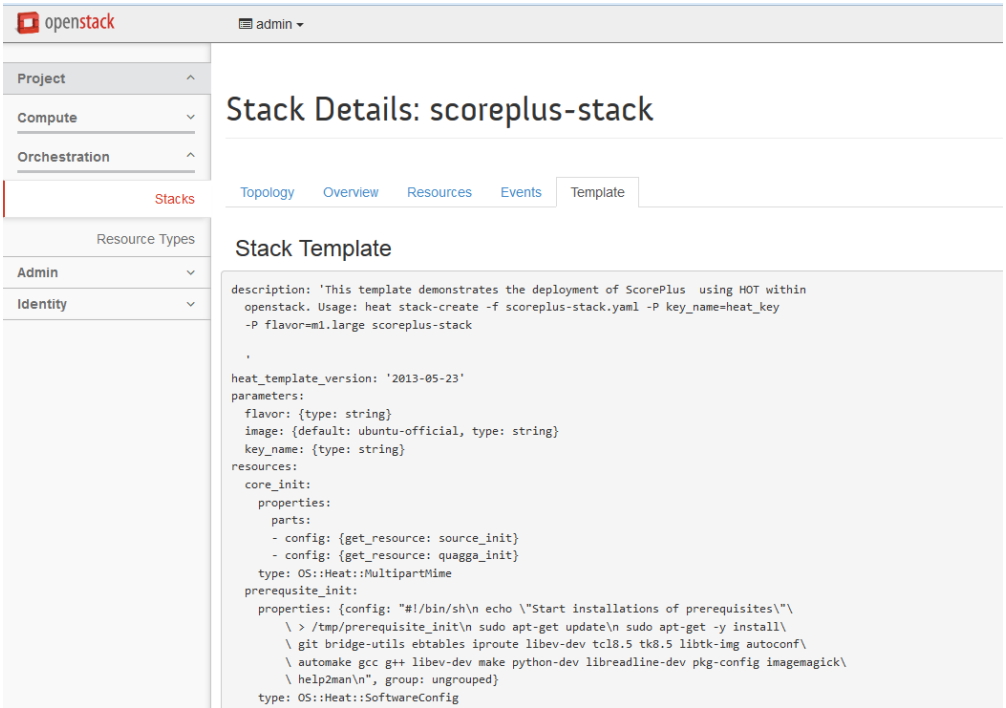


Figure 6.17 Heat Orchestration Template for ScorePlus

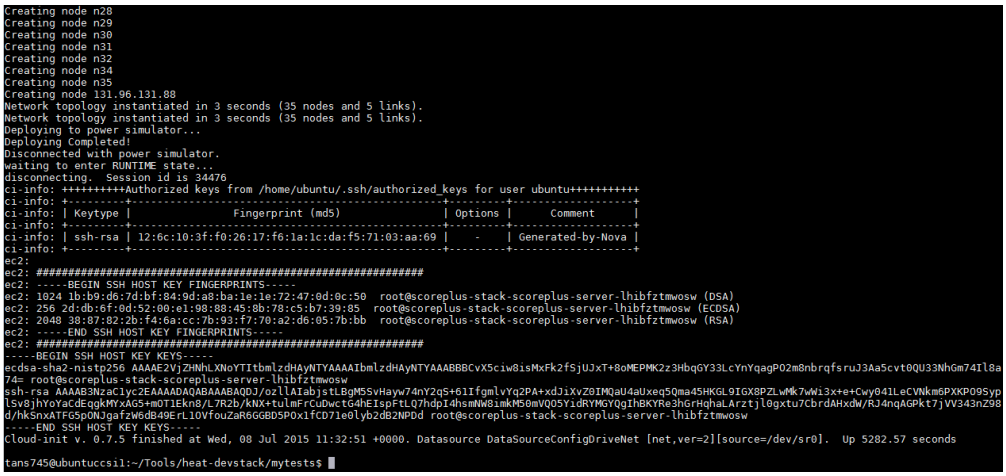


Figure 6.18 Openstack nova console output after ScorePlus deployment

6.5.3 Sample Use

Figure 6.17 shows a sample Heat stack template using ScorePlus resource plugin, which also includes other resources like OS::Nova::Server, OS::Heat::SoftwareConfig and OS::Heat::MultipartMime, etc. Figure 6.18 captures the corresponding console output after the deployment of this template in OpenStack. With respect to the dependencies between different resources, we can also see the dependency topology for the allocated resources from this template stack in Figure 6.19.

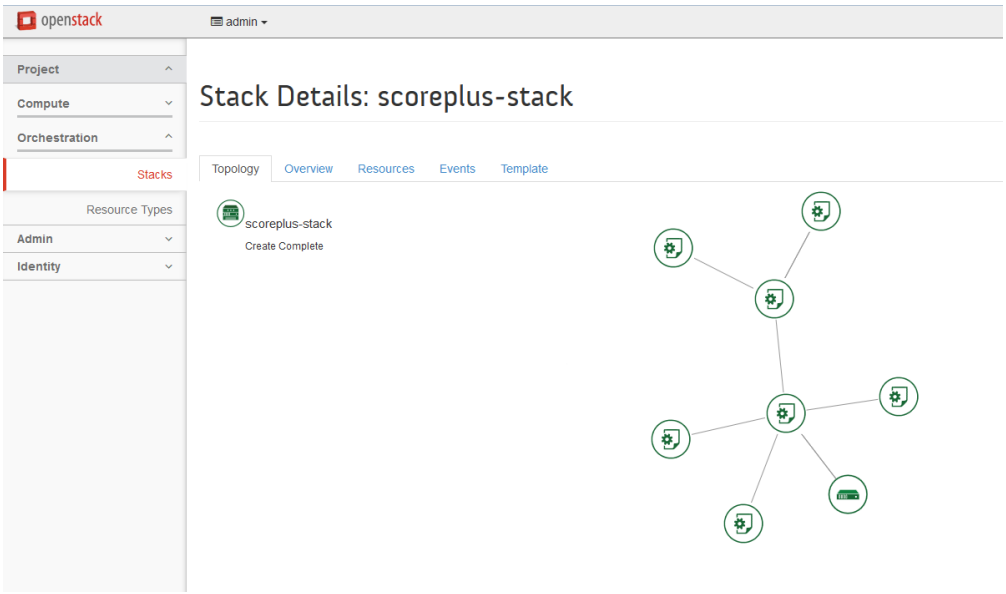


Figure 6.19 Resource Dependency Topology within ScorePlus Heat template

6.6 Evaluation and Experimentation

In this section, we demonstrate and evaluate the capabilities of ScorePlus to support cyber-physical analysis in Smart Grid, particularly in Microgrid [76]. The future Smart Grid is expected to be an integration of Microgrids featured by localized power generation, storage and consumption [77]. Microgrid works as an independent local power system that has the flexibility to connect (connecting mode) and disconnect (islanding mode) from the main grid as needed in order to minimize the energy cost and maximize the grid stability [78]. Figure 6.20 illustrates a typical Microgrid structure. Note that in ScorePlus we use Topology Switch to serve as circuit breakers.

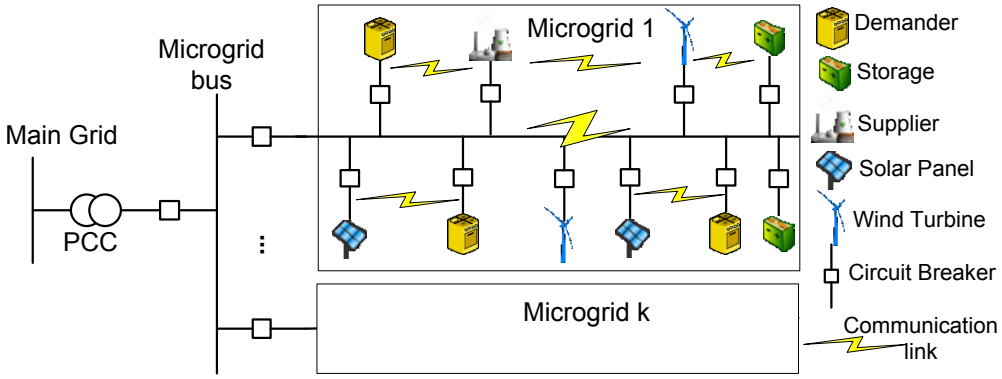


Figure 6.20 Typical Microgrid Structure

6.6.1 Experiment setup

The test case is created upon the AMI network test case in [79] and demand response model in [80]. The demand response model is also run in @GridLAB-D as a comparison for accuracy. We investigate the Microgrid demand response behavior in one virtual day. For each virtual time period h in the virtual day (2 real time seconds), $h = 0, 1, 2, 3, ...23$, each kind of nodes behaves as the following:

- **Supplier:** Providing power with maximum 2kw to the whole grid and broadcasting its hourly price [80] to all the topology switches. Calculating the bills for each demander based on the real time price and the collected energy consumption from Demanders.

- **Topology Switch:** Receiving the energy price information from the Supplier and then relay the messages to the demanders immediately.
- **Solar Panel:** When $h \in [0, 5] \bigcup [19, 23]$, setting its maximum output to 0. When $h \in [6 : 18]$, randomly setting its maximum output between 200w and 300w.
- **Wind Turbine:** Randomly Setting its maximum output between 100w and 200w for each hour.
- **Storage:** Its capacity is set to 1kwh with initial energy 0.1kwh; Sending its current energy residual to demanders and charges/discharges with maximum 100w as requested from Demanders.
- **Demander:** Either critical Demander or adjustable Demander. Critical Demander must work at 200w all the time. Adjustable Demander receives price from Topology switch and energy residual from Storage and adjust its setpoint based on the demand response model in [80].

An AMI communication network is set up between all the nodes, which consists of both Ethernet and IEEE 802.11 RF Mesh network with 54Mbps bandwidth, 0.1% packet loss rate. All the above node behaviors are implemented by the control programs running inside each node. Figure 6.21 shows Xterminal for a Demander node. We use command *ps aux* to check programs running inside. They include the OSPF routing process and the demander energy daemon process.

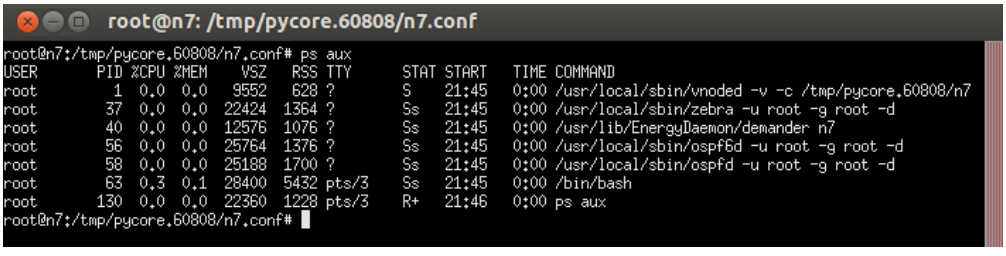


Figure 6.21 Smart Grid applications running in a Demander node

Five Microgrids are created as in Table II, running on four software emulation servers and one hardware testbed respectively.

Table 6.3 Number of nodes in each Microgrid

Microgrid ID	Type	Demander	Supplier	Switch	Solar	Turbine	Storage	Total
1	Software Emulator	11	1	5	4	4	5	30
2	Hardware Testbed	11	1	5	4	4	5	30
3	Software Emulator	22	2	10	8	8	10	60
4	Software Emulator	33	3	15	12	12	15	90
5	Software Emulator	55	5	25	20	20	25	150

6.6.2 Islanding Mode

In islanding mode, each Microgrid is working independently and self-sustained. We first examine Microgrid 1 and 2 for comparison. Figure 6.22 shows the real-time energy price and the adjustable Demander energy consumption over time, one in software emulator, one in hardware testbed and one in GridLAB-D as comparison for accuracy. We can see that the control program for the price responsive model in [80] tends to shift the energy consumption to the lower price period of the day. In addition, we can see the difference of the power profiles between the two adjustable Demanders. Theoretically, the two curves should be exactly the same since they are controlled by the same program as previously specified. However, since the software emulator is event-based processing, it can only update with a large discrete step size. As a comparison, due to the high ADC sampling rate of TelosW, the one in hardware testbed has the advantage of capturing the transient dynamics in a finer grain.

Figure 6.23 shows the renewable share of total energy consumption over time in all five Microgrids. We can see that generally, when the real time energy price increases, the renewable share increases correspondingly. In addition, as the size of the Microgrid increases, the renewable share become less sensitive to the price fluctuations since more renewable energy surplus can be distributed and stored by the demand response process.

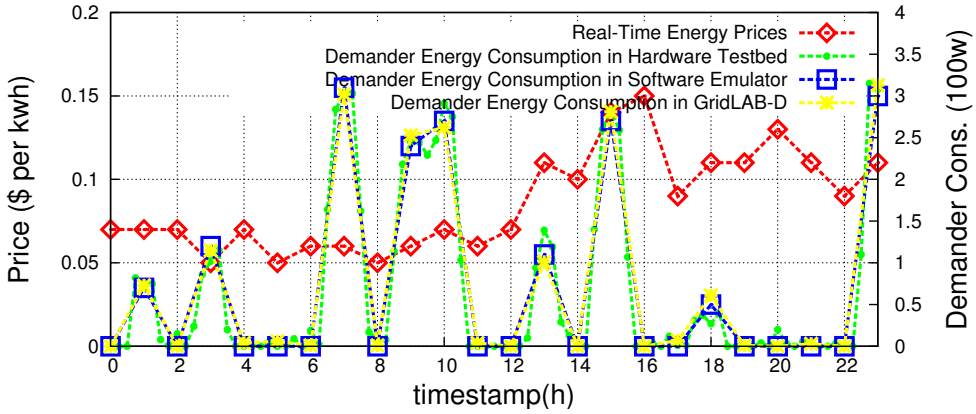


Figure 6.22 Real-Time Energy Price and Demander Energy Consumption

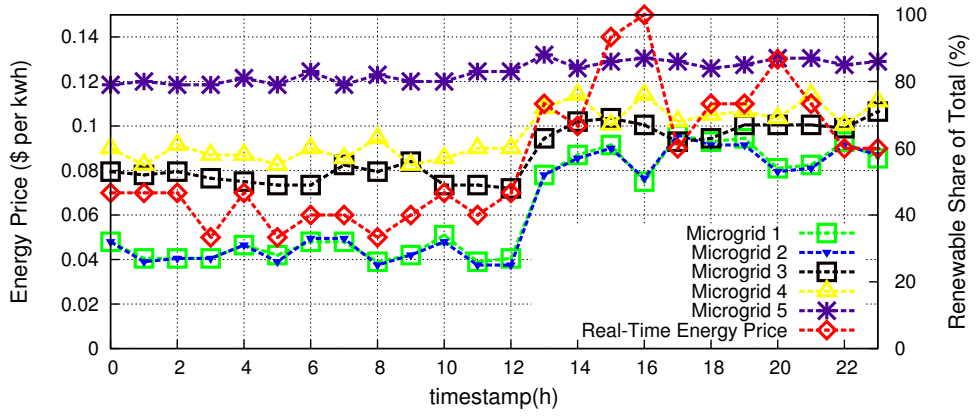


Figure 6.23 Renewable Share of Total Energy Consumption

6.6.3 Connecting Mode

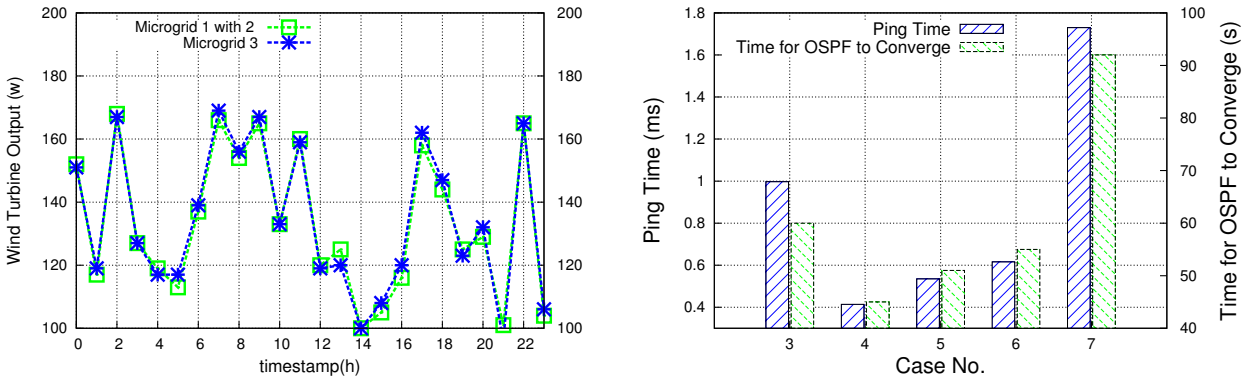
In connecting mode, Microgrid is connected with each other. We first examine our implementation for the integrations between different Microgrids. The software emulation servers and hardware testbed are connected with each other through Internet. Note that we specifically design the five Microgrids in Table II, such that Microgrid 1 connected with 2 is the same as Microgrid 3, and Microgrid 3 with 4 is the same as Microgrid 5. The study case No. and the Microgrids connected are listed in Table III.

Table 6.4 Cases in Connecting Mode

Case No.	1	2	3	4	5	6	7	8
Microgrid connected	1,2	3	2,3	4	3,4	5	1,2,4	1,3,4,5

Figure 6.24(a) shows the wind turbine output in case 1 and 2. The two curves conform with each other and the integration between software emulator with hardware testbed preserves the power profiles accurately. Figure 6.24(b) shows the ping time between two farthest nodes and the time needed for OSPF routing to converage in case 3, 4, 5, 6, 7. It indicates that for communication network, the integrations with software emulators introduce minor delays but major delays with hardware testbed, which is about 0.57ms difference in ping time.

Deployment in OpenStack For the multiple ScorePlus instantiations in case 8, we employ the ScorePlus resource plugin to deploy them all at one time in OpenStack. In particular, Neutron networking resources are integrated into the stack template, such that a private internal subnet is created to connect all 5 ScorePlus servers, all of which uses a common gateway router to communicate with external networks. The result networking topology of the multiple ScorePlus servers in OpenStack are illustrated in Figure 6.25.



(a) Wind Turbine Output (b) Network performance

Figure 6.24 Integrations between different Microgrids

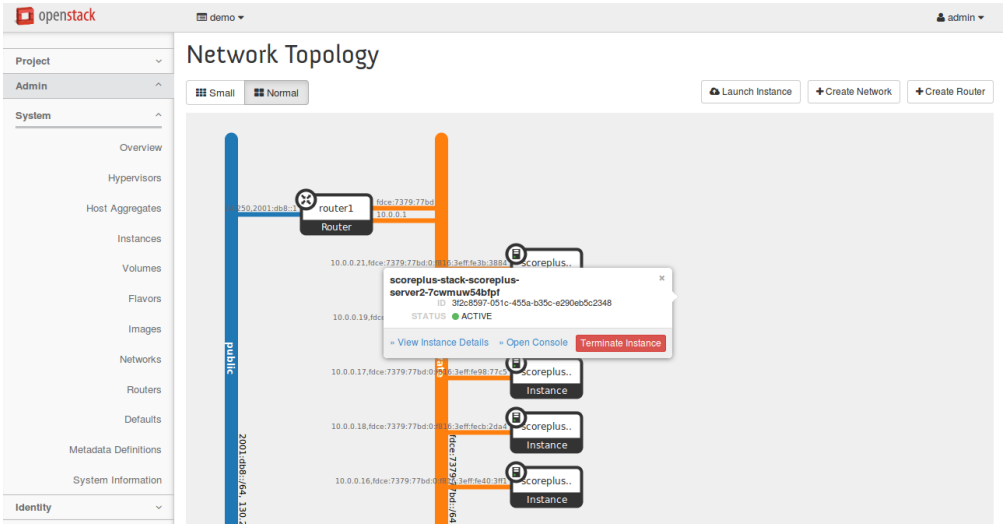


Figure 6.25 Networking topology of multiple ScorePlus servers after deployment in Openstack

Topology Attacks In connecting mode, the influence between different Microgrids becomes extremely important since the failure in one Microgrid could impact the stability of other Microgrids. To this end, we conduct contingency analysis for case 1 under topology attacks, in which attackers intend to disrupt the normal physical conditions of power system through malicious control of Topology Switch. As in Figure 6.26, suppose attackers inject a malicious control program to the Topology Switch n4 in house 3 and runs it at time h=7, such that it disconnects house 4 with house 5 as well as the hardware testbed. Only port 2, 3 and 4 are connected. In normal condition, the Topology Switch in house 3 is set such that port 1, 2, 3, 4, 5, 6 are all connected. So the

power needed in house 3 can be supplied from house 4, house 5 and the power network hardware testbed. After manipulating the connection status in the Topology Switch, the previous balanced power flow in the network is changed to unbalanced one, leading to a dramatic increase in the power flows (indicated by the red bold line in GUI). Meanwhile, since the hardware testbed is not providing power to the software emulator any more but the power generation is not aware of the attack at the moment, the extra power flow would be inevitably forwarded to the Storage devices in hardware testbed, which results in a sudden peak charging rate at the storage device, as marked by the red circle. The charging rate will reach a sudden peak around h=7 but return a higher but relative stable level afterwards since more energy surplus is being forwarded to Storage device. The same experiment is also conducted by replacing the hardware testbed with another software emulator. We can see that the hardware testbed could capture the power dynamics in a finer grain.

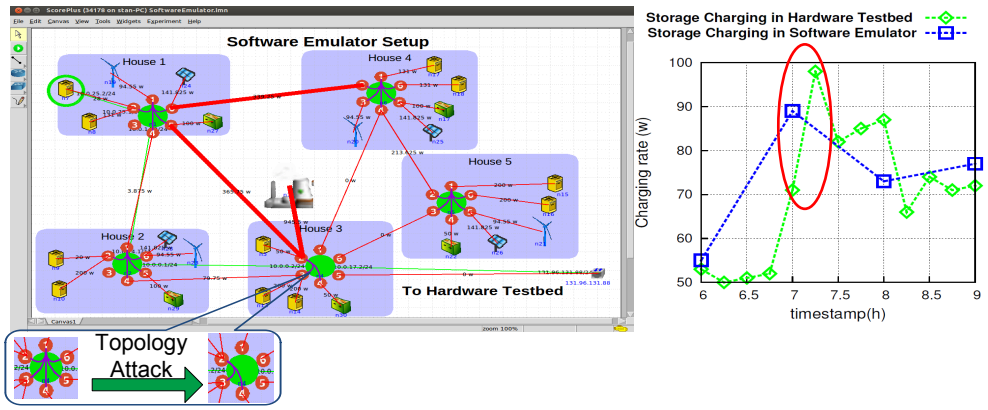


Figure 6.26 Cyber-physical attack on Topology Switch

6.6.4 Comprehensive Cyber-Physical Attacks

This testing case we created is based on the AMI network test case from American Electric Power Company[79] and the IEEE PES 37 bus distribution system test feeders [81]. Through this case, we further illustrate the advantages of our platform over software simulators: 1) the actual control program written in C language (either correct or malicious modified) can be directly run on each virtual node; 2) the real time cyber-physical impacts (altered system routing table entries and power flow perturbations) of the control programs can be demonstrated. Figure 6.27 shows the

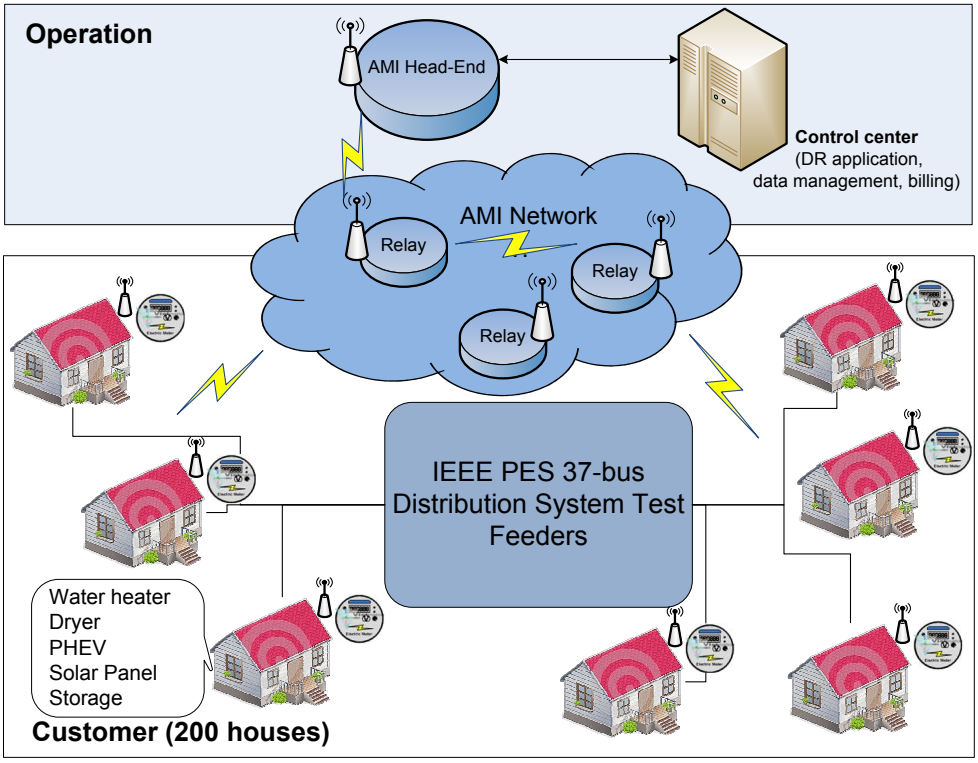


Figure 6.27 Comprehensive cyber attack case

building blocks of our experimental scenario:

- Operation layer: The control program in control center broadcasts real-time energy prices every 5 minutes and also collects meter reading data through AMI Head-End. Meanwhile, the control center also calculates the bills for each house, based on the real time price and the collected energy consumption data.

- Customer layer: The IEEE 37 bus distribution test feeders is set up to provide power for 200 residential houses. Each house is equipped with loads including a water heater, a dryer, a PHEV, a solar panel, and a storage. Moreover, a smart meter is employed to serve as the interface between the power network and AMI for each house. The program running in smart meter responses to the real time prices to adjust the setpoint of appliances within each house correspondingly based on the price-responsive control model in [80].

- AMI network: AMI enables communications and interactions between/within the operation

layer and the customer layer. The control center and AMI Head-End is connected through Internet. AMI Head-End, the relay nodes and the smart meters are formed as a IEEE 802.11 Radio Frequency Mesh network.

Suppose the customer under smart meter X wants to manipulate his energy bill without being caught. In order to achieve this, he launches a Distributed Denial-of-service attack to the bi-direction data flow within AMI, which consists of the energy consumption data from the smart meters to the AMI Head-End, and the energy price data from the AMI Head-End to the smart meters. For the energy consumption data, the attacker modifies the ones from smart meter X and his targeted neighbors, such that each targeted neighbor has an increase in the reported energy consumption compared to the actual consumption, and the smart meter X has a decrease equal to the total increase of its targeted neighbors in the reported energy consumption. In this way, from the perspective of utility company, the total energy provided still conforms to the total energy being billed. For the energy price data, the attacker modifies the price to a lower value, such that based on the demand response model, the actual energy consumption of each targeted neighbors will also increase. In this case, from the perspective of each targeted neighbor, the minor increase in the reported energy consumption data due to attack will become even less noticeable.

Specifically, as shown in Figure 6.28, the customer of smart meter X attacks three relay nodes at the same time: its own direct cluster head (Relay 2) and two neighbor cluster heads (Relay 1 and 3). Originally, Relay 1 and Relay 3 will directly interact with Relay 4 for the bi-direction data. We can see this from the result of *route* command in the terminal of Relay 1. To reach 192.169.0.4, which is the IP address of Relay 4, no intermediate gateway is needed and packets can be simply forwarded through interface *eth*0. However, after attack, there is one extra high priority entry in the routing table of Relay 1 such that the packets designated to 192.169.0.4 will be forwarded to 192.169.0.2 first instead of the original one hop reach. As a result, for Relay 2, besides the data packets of the 10 customers within its own cluster, it will also intercept the data packets of the other 20 customers within the clusters of Relay 1 and 3. By making the three Relay nodes working in concert to compromise the data, customer X could dramatically reduces its own reported energy usage. As shown in Figure 6.29, for smart meter X, even though the actual energy usage across

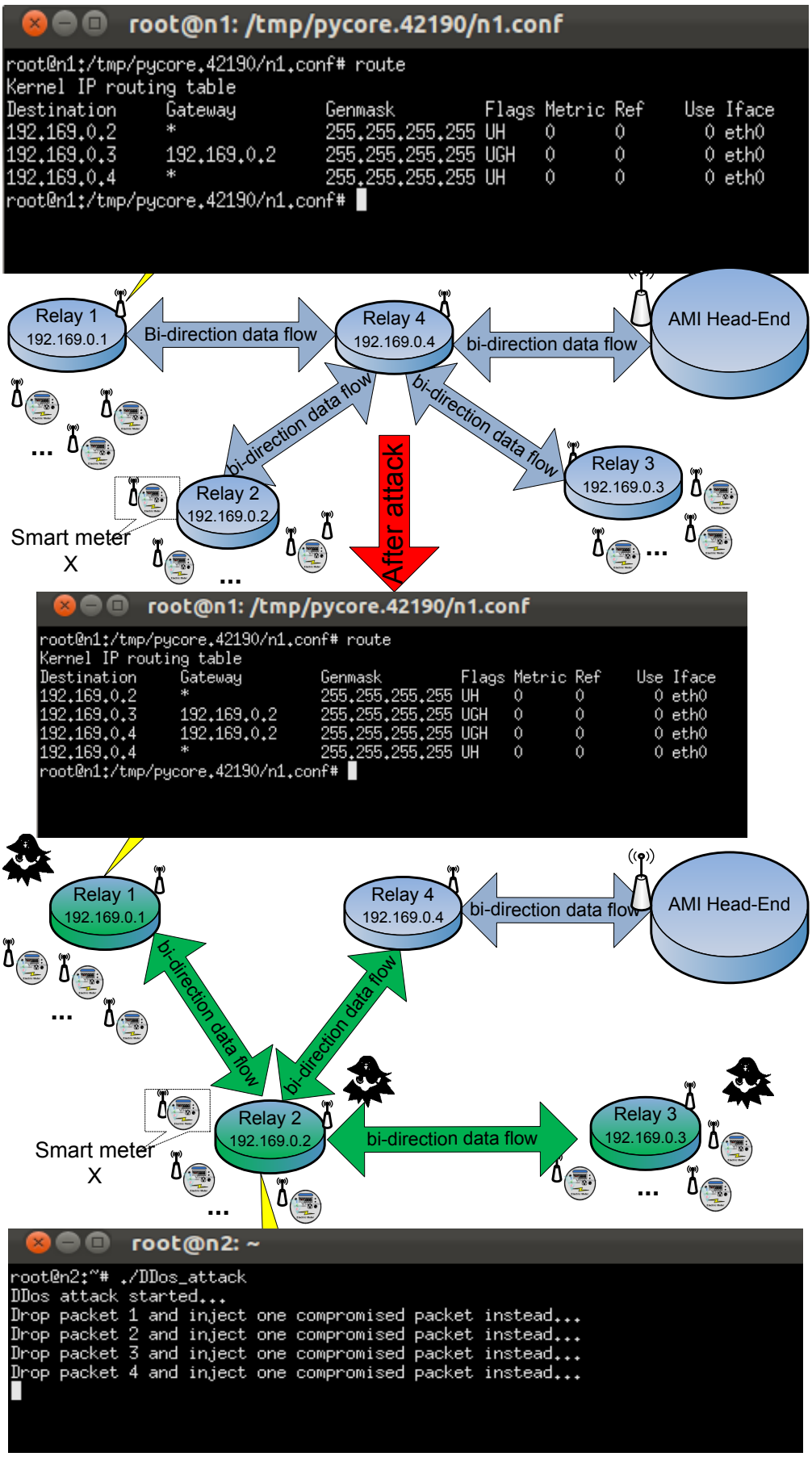


Figure 6.28 Potential attack within AMI

the day is 64kwh, the reported data is manipulated to 35kwh. The remaining 64 – 35 = 29kwh are evenly added to the other 29 customers' reported data. In this way, from the perspective of utility company, the total energy consumed still conforms with the total energy being billed. Moreover, from the perspective of each of the other 29 customers', since only 29/29 = 1kwh is added to their energy consumption, which usually results in about 0.1$ increase in their bills,it is very much likely that the customer will just let it go. Also note that since the energy price is modified to

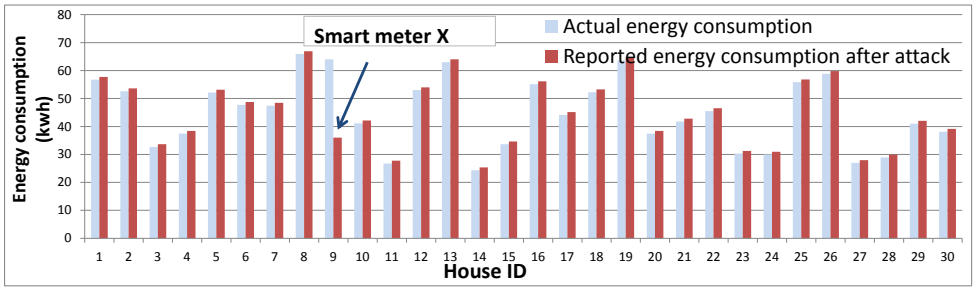


Figure 6.29 Actual energy usage and reported energy usage after attack

a lower value after the attack, the real power consumption paradigm of the attacked neighbors changes dramatically, compared to the normal situation when the correct real time energy price is given. As shown in Figure 6.30, the real power consumption of the attacked neighbors stays at a relatively higher level all the time after the attack and the demand response through real time pricing is not working any more. If more neighbors are involved in the attack, this will severely increase load of the system, which can result in a higher cost of power transmission or even an outage. An effective approach to detect this kind of attack is by monitoring the network traffic.

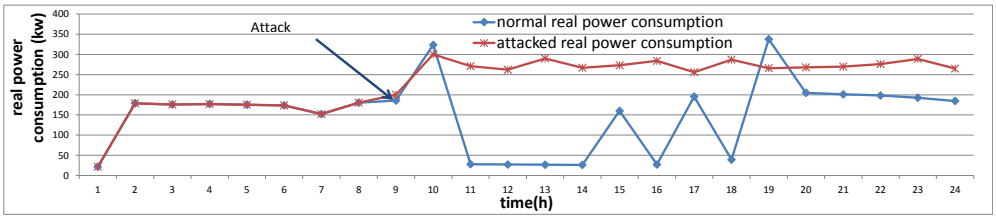


Figure 6.30 The total real power consumption of the attacked neighbors

As shown in Figure 6.31, since the routing path of the packets is changed and much more data

packets are forwarded to Relay 2, the throughput of Relay 2 will be increased unusually from the moment of attack. Also, the network traffic congestion at Relay 2 will result in an increase in the communication delay from Relay 1 to the AMI meter head.

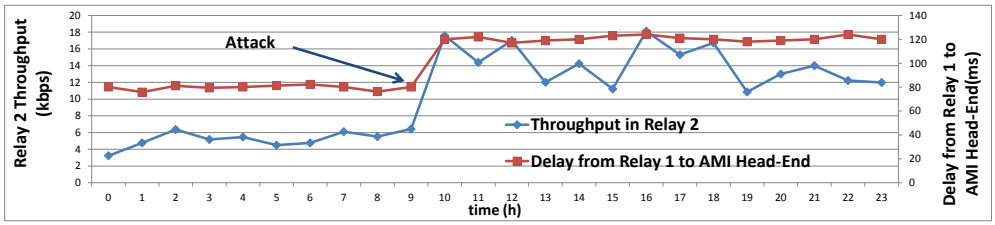


Figure 6.31 Throughput and Communication delay

6.7 Conclusion

In this chapter, we present the design, implementation and evaluation of ScorePlus, a software-hardware hybrid and federated experiment environment for Smart Grid. Our platform provides an extendable design paradigm for the creation of general cyber-physical testbeds. Testing cases such as distributed control algorithms, demand responses, and cyber-physical security issues, can all be evaluated in our platform. Future work could investigate the design of more hardware components like Phaser Measurement Unit and the decrease in system integration overhead. The ScorePlus codes including both software emulator and hardware testbed are open source released at `https://sourceforge.net/projects/scorepluset/`.

CHAPTER 7

CONCLUSIONS

This dissertation addresses two important research aspects about cyber-physical security of Smart Grid: (i) The construction, impact and countermeasure of data integrity attacks; and (ii) The design and implementation of general cyber-physical security experiment platform. For data integrity attacks: based on the system model of state estimation process in Smart Grid, firstly, a data integrity attack model is formulated, such that the attackers can generate financial benefits from the real-time electrical market operations. Then, to reduce the required knowledge about the targeted power system when launching attacks, an online attack approach is proposed, such that the attacker is able to construct the desired attacks without the network information of power system. Furthermore, a network information attacking strategy is proposed, in which the most vulnerable meters can be directly identified and the desired measurement perturbations can be achieved by strategically manipulating the network information. Besides the attacking strategies, corresponding countermeasures based on the sparsity of attack vectors and robust state estimator are provided respectively. For the experiment platform: ScorePlus, a software-hardware hybrid and federated experiment environment for Smart Grid, is presented. ScorePlus incorporates both software emulator and hardware testbed, such that they all follow the same architecture, and the same Smart Grid application program can be tested on either of them without any modification. ScorePlus provides a federated environment such that multiple software emulators and hardware testbeds at different locations are able to connect and form a unified Smart Grid system. ScorePlus software is encapsulated as a resource plugin in OpenStack cloud computing platform, such that it supports massive deployments with large scale test cases in cloud infrastructure.

Bibliography

[1] M. Gilstrap, A. Shravan, and K. DeCorla-Souza., "United states electricity industry primer," *U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability*, July 2015.

[2] S. Collier, "The emerging enernet: Convergence of the smart grid with the internet of things," in *Rural Electric Power Conference (REPC), 2015 IEEE*, April 2015, pp. 65–68.

[3] "Nist framework and roadmap for smart grid interoperability standards, release 1.0," *National Institute of Standards and Technology*, 2010. [Online]. Available: http://dx.doi.org/10.6028/NIST.SP.1108r1

[4] "Roadmap to secure control systems in the energy sector," *Department of Energy and Department of Homeland security*, 2008.

[5] G. Ericsson, "Cyber security and power system communication and essential parts of a smart grid infrastructure," *Power Delivery, IEEE Transactions on*, vol. 25, no. 3, pp. 1501–1507, July 2010.

[6] S. Tan, W. Song, M. Stewart, and L. Tong, "Construct data integrity attacks against real-time electrical market in smart grid," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2015.

[7] S. Tan, W.-Z. Song, M. Stewart, and L. Long, "Lpattack: Leverage point attacks against state estimation in smart grid," in *Global Communications Conference (GLOBECOM), 2014 IEEE*, Dec 2014, pp. 643–648.

[8] S. Tan, W. Song, S. Yothment, J. Yang, and L. Tong, "Scoreplus: An integrated scalable cyber-physical experiment environment for smart grid," in *IEEE International Conference on Sensing, Communication and Networking (SECON)*, 2015.

[9] A. Abur and A. Expósito, *Power System State Estimation: Theory and Implementation*, 2004.

[10] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, 2009.

[11] O. Kosut, L. Jia, R. Thomas, and L. Tong, "Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, 2010, pp. 220–225.

[12] R. B. Bobba, K. M. Rogers, Q. Wang, a. K. N. Himanshu Khurana, and T. J. Overbye, "Detecting false data injection attacks against dc state estimation," in *First Workshop on Secure Control Systems (SCS), IEEE*, 2010.

[13] S. Cui, Z. Han, S. Kar, T. Kim, H. Poor, and A. Tajer, "Coordinated data-injection attack and detection in the smart grid: A detailed look at enriching detection solutions," *Signal Processing Magazine, IEEE*, vol. 29, no. 5, pp. 106–115, 2012.

[14] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, and K. Poolla, "Smart grid data integrity attacks," *Smart Grid, IEEE Transactions on*, vol. 4, no. 3, pp. 1244–1253, 2013.

[15] Y. Huang, M. Esmalifalak, H. Nguyen, R. Zheng, Z. Han, H. Li, and L. Song, "Bad data injection in smart grid: attack and defense mechanisms," *Communications Magazine, IEEE*, vol. 51, no. 1, pp. 27–33, 2013.

[16] J. Kim and L. Tong, "On topology attack of a smart grid: Undetectable attacks and countermeasures," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 7, pp. 1294–1305, 2013.

[17] L. Xie, Y. Mo, and B. Sinopoli, "Integrity data attacks in power market operations," *Smart Grid, IEEE Transactions on*, vol. 2, no. 4, pp. 659–666, Dec 2011.

[18] O. Kosut, L. Jia, R. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *Smart Grid, IEEE Transactions on*, vol. 2, no. 4, pp. 645–658, Dec 2011.

[19] L. Jia, R. Thomas, and L. Tong, "Malicious data attack on real-time electricity market," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, May 2011.

[20] Y. Yuan, Z. Li, and K. Ren, "Quantitative analysis of load redistribution attacks in power systems," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 9, pp. 1731–1738, Sept 2012.

[21] R. Tan, V. Badrinath Krishna, D. K. Yau, and Z. Kalbarczyk, "Impact of integrity attacks on real-time pricing in smart grids," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer Communications Security*. New York, NY, USA: ACM, 2013, pp. 439–450.

[22] M. Esmalifalak, G. Shi, Z. Han, and L. Song, "Bad data injection attack and defense in electricity market using game theory study," *Smart Grid, IEEE Transactions on*, vol. 4, no. 1, pp. 160–169, March 2013.

[23] D.-H. Choi and L. Xie, "Sensitivity analysis of real-time locational marginal price to scada sensor data corruption," *Power Systems, IEEE Transactions on*, vol. 29, no. 3, pp. 1110–1120, May 2014.

[24] L. Jia, J. Kim, R. Thomas, and L. Tong, "Impact of data quality on real-time locational marginal price," *Power Systems, IEEE Transactions on*, vol. 29, no. 2, pp. 627–636, March 2014.

[25] K. smart grid institute, "Jeju Smart-Grid Testbed," 2010. [Online]. Available: http://smartgrid.jeju.go.kr/eng/

[26] I. N. Laboratory, "National scada test bed: Final fact sheet," 2009.

[27] D. Stimoniaris, D. Tsiamitros, T. Kottas, N. Asimopoulos, and E. Dialynas, "Smart grid simulation using small-scale pilot installations. - experimental investigation of a centrally-controlled microgrid," in *PowerTech, 2011 IEEE Trondheim*, june 2011, pp. 1 –6.

[28] W.-Z. Song, D. De, S. Tan, S. Das, and L. Tong, "A wireless smart grid testbed in lab," *Issue on Recent Advances in Wireless Technologies for Smart Grid, IEEE Wireless Communications Magazine*, 2012.

[29] J. B. Lobo and P. Idowu, "Laboratory scale microgrid test bed: Hardware implementation," *10th CMU workshop for smart grid testbed*, 2015.

[30] A. Hahn, A. Ashok, S. Sridhar, and M. Govindarasu, "Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid," *Smart Grid, IEEE Transactions on*, vol. 4, pp. 847–855, 2013.

[31] M. Stanovich, I. Leonard, K. Sanjeev, M. Steurer, T. Roth, S. Jackson, and M. Bruce, "Development of a smart-grid cyber-physical systems testbed," in *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES*, 2013, pp. 1–6.

[32] R. Barzegaran, "Attributes and design of resilient renewable microgrid laboratory," *10th CMU workshop for smart grid testbed*, 2015.

[33] Y. Guo, R. Li, G. Poulton, and A. Zeman, "A simulator for self-adaptive energy demand management," in *Self-Adaptive and Self-Organizing Systems, 2008. SASO '08. Second IEEE International Conference on*, oct. 2008, pp. 64 –73.

[34] A. Molderink, M. Bosman, V. Bakker, J. Hurink, and G. Smit, "Simulating the effect on the energy efficiency of smart grid technologies," in *Simulation Conference (WSC), Proceedings of the 2009 Winter*, dec. 2009, pp. 1530 –1541.

[35] P. Faria, Z. Vale, and J. Ferreira, "Demsi: A demand response simulator in the context of intensive use of distributed generation," in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, oct. 2010, pp. 2025 –2032.

[36] A. Narayan, "GridSpice-A Virtual Test Bed for Smart Grid," Tech. Rep., 2008.

[37] D. Chassin, K. Schneider, and C. Gerkensmeyer, "Gridlab-d: An open-source power systems modeling and simulation environment," in *Transmission and Distribution Conference and Exposition, 2008. T.No.x00026;D. IEEE/PES*, april 2008, pp. 1 –5.

[38] D. D. G. Ray D. Zimmerman, Carlos E. Murillo-S, "A MATLAB Power System Simulation Package," Tech. Rep., 1999.

[39] W. Li, X. Zhang, and H. Li, "Co-simulation platforms for co-design of networked control systems: An overview," *Control Engineering Practice*, vol. 23, pp. 44 – 56, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066113001925

[40] K. Hopkinson, X. Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury, "Epochs: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components," *Power Systems, IEEE Transactions on*, vol. 21, no. 2, pp. 548 – 558, may 2006.

[41] T. Godfrey, S. Mullen, R. Dugan, C. Rodine, D. Griffith, and N. Golmie, "Modeling smart grid applications with co-simulation," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, oct. 2010, pp. 291 –296.

[42] M. Mallouhi, Y. Al-Nashif, D. Cox, T. Chadaga, and S. Hariri, "A testbed for analyzing security of scada control systems (tasscs)," in *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, 2011, pp. 1–7.

[43] H. Lin, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili, "Power system and communication network co-simulation for smart grid applications," in *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, jan. 2011, pp. 1 –6.

[44] Y. Deng, H. Lin, S. Shukla, J. Thorp, and L. Mili, "Co-simulating power systems and communication network for accurate modeling and simulation of pmu based wide area measurement systems using a global event scheduling technique," in *Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES),2013*, May 2013, pp. 1–6.

[45] K. Mets, T. Verschueren, C. Develder, T. Vandoorn, and L. Vandevelde, "Integrated simulation of power and communication networks for smart grid applications," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2011 IEEE 16th International Workshop on*, june 2011, pp. 61 –65.

[46] J. Nutaro, P. Kuruganti, L. Miller, S. Mullen, and M. Shankar, "Integrated hybrid-simulation of electric power and communications systems," in *Power Engineering Society General Meeting, 2007. IEEE*, june 2007, pp. 1 –8.

[47] S. Ciraci, J. Daily, J. Fuller, A. Fisher, L. Marinovici, and K. Agarwal, "Fncs: A framework for power system and communication networks co-simulation," in *Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative*, ser. DEVS '14. San Diego, CA, USA: Society for Computer Simulation International, 2014, pp. 36:1–36:8. [Online]. Available: http://dl.acm.org/citation.cfm?id=2665008.2665044

[48] S. Lehnhoff, O. Nannen, S. Rohjans, F. Schlogl, S. Dalhues, L. Robitzky, U. Hager, and C. Rehtanz, "Exchangeability of power flow simulators in smart grid co-simulations with mosaik," in *Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), 2015 Workshop on*, April 2015, pp. 1–6.

[49] F. Wu, P.Varaiya, P.Spiller, and Oren.S, "Folk theorems on transmission access: Proofs and counterexamples," *Journal of Regulatory Economics*, vol. 10, no. 1, pp. 5–23, 1996.

[50] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, 2nd ed. Wiley-Interscience, 1996.

[51] A. Ott, "Experience with pjm market operation, system design, and implementation," *Power Systems, IEEE Transactions on*, vol. 18, no. 2, pp. 528–534, May 2003.

[52] L. Rios and N. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," *Journal of Global Optimization*, vol. 56, no. 3, pp. 1247–1293, 2013.

[53] B. Yang, "Projection approximation subspace tracking," *Signal Processing, IEEE Transactions on*, vol. 43, no. 1, pp. 95–107, Jan 1995.

[54] Y. Chi, Y. Eldar, and R. Calderbank, "Petrels: Parallel subspace estimation and tracking by recursive least squares from partial observations," *Signal Processing, IEEE Transactions on*, vol. 61, no. 23, Dec 2013.

[55] K. J. Astrom and B. Wittenmark, *Adaptive Control*, 2nd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1994.

[56] J. Hao, R. Piechocki, D. Kaleshi, W. H. Chin, and Z. Fan, "Optimal malicious attack construction and robust detection in smart grid cyber security analysis," in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*, Nov 2014, pp. 836–841.

[57] L. Liu, M. Esmalifalak, Q. Ding, V. Emesih, and Z. Han, "Detecting false data injection attacks on power grid by sparse optimization," *Smart Grid, IEEE Transactions on*, vol. 5, no. 2, pp. 612–621, March 2014.

[58] E. Candes and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009. [Online]. Available: http://dx.doi.org/10.1007/s10208-009-9045-5

[59] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, Jun. 2011. [Online]. Available: http://doi.acm.org/10.1145/1970392.1970395

[60] "Data sheets for ieee bus systems." [Online]. Available: http://shodhganga.inflibnet.ac.in/bitstream/10603/5247/18/19_appendix.pdf

[61] "New york independent system operator load data." [Online]. Available: http://www.nyiso.com/public/markets_operations/market_data/load_data/index.jsp

[62] G. Chaojun, P. Jirutitijaroen, and M. Motani, "Detecting false data injection attacks in ac state estimation," *Smart Grid, IEEE Transactions on*, vol. 6, no. 5, pp. 2476–2483, Sept 2015.

[63] L. Mili, V. Phaniraj, and P. Rousseeuw, "Least median of squares estimation in power systems," *Power Systems, IEEE Transactions on*, vol. 6, no. 2, pp. 511–523, May 1991.

[64] D. C. Hoaglin and R. E. Welsch, "The hat matrix in regression and anova," *The American Statistician*, vol. 32, no. 1, pp. 17–22, 1978.

[65] L. Mili, M. Cheniae, N. S. Vichare, and P. Rousseeuw, "Robust state estimation based on projection statistics [of power systems]," *Power Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 1118–1127, 1996.

[66] R. A. S. Benedito, L. F. C. Alberto, N. G. Bretas, and J. B. A. London, "Power system state estimation: Undetectable bad data," *International Transactions on Electrical Energy Systems*, 2013.

[67] P. J. Green, "Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 46, no. 2, pp. 149–192, 1984.

[68] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011. [Online]. Available: http://dx.doi.org/10.1109/tpwrs.2010.2051168

[69] W. Hogan, "Contract networks for electric power transmission," *Journal of Regulatory Economics*, vol. 4, no. 3, pp. 211–242, 1992. [Online]. Available: http://dx.doi.org/10.1007/BF00133621

[70] T. Zheng and E. Litvinov, "On ex post pricing in the real-time electricity market," *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 153–164, Feb 2011.

[71] S. Tan, W.-Z. Song, Q. Dong, and L. Tong, "Score: Smart-grid common open research emulator," in *The 3rd IEEE International Conference on Smart Grid Communications (IEEE SmartGridComm)*, 2012.

[72] B. Zhang, S. Sun, and Z. Yan, *Advanced Power Network Analysis*, T. U. Press, Ed., 2007.

[73] Y. Saad, *Iterative Methods for Sparse Linear Systems, Second Edition*, 2nd ed. Society for Industrial and Applied Mathematics, Apr. 2003.

[74] Beagleboard, *Beagleboard Reference Manual*, Beagleboard, 2009.

[75] G. Lu, D. De, M. Xu, W.-Z. Song, and B. Shirazi, "TelosW: Enabling Ultra-Low Power Wake-On Sensor Network," in *INSS 2010*, Jun. 2010.

[76] B. Hartono, Y. Budiyanto, and R. Setiabudy, "Review of microgrid technology," in *QiR (Quality in Research), 2013 International Conference on*, June 2013, pp. 127–132.

[77] H. Liang, B. J. Choi, W. Zhuang, X. Shen, A. Awad, and A. Abdr, "Multiagent coordination in microgrids via wireless networks," *Wireless Communications, IEEE*, vol. 19, no. 3, pp. 14–22, June 2012.

[78] A. C. R. Kamel and K. Nagasaka, "Detailed analysis of micro-grid stability during islanding mode under different load conditions," *Engineering, Vol. 3 No. 5, 2011, pp. 508-516*.

[79] R. Sarfi, B. D. Green, and J. Simmins, "AMI Network (AMI Head-End to/from Smart Meters)," August 2012.

[80] Hammerstrom, D. J., "Pacific Northwest GridWiseTM Testbed Demonstration Projects: Part I. Olympic Peninsula Project," Tech. Rep., October 2007.

[81] IEEE PES Distribution System Analysis Subcommittee's Distribution Test Feeder Working Group, "IEEE 37 Node Test Feeder," Tech. Rep., September 2010.