Spring 5-11-2015

Enhanching Security in the Future Cyber Physical Systems

Kebina Manandhar

ENHANCING SECURITY IN THE FUTURE CYBER PHYSICAL SYSTEMS

by

KEBINA MANANDHAR

Under the Direction of Xiaojun Cao, PhD

ABSTRACT

Cyber Physical System (CPS) is a system where cyber and physical components work in a complex co-ordination to provide better performance. By exploiting the communication infrastructure among the sensors, actuators, and control systems, attackers may compromise the security of a CPS. In this dissertation, security measures for different types of attacks/ faults in two CPSs, water supply system (WSS) and smart grid system, are presented. In this context, I also present my study on energy management in Smart Grid.

The techniques for detecting attacks/faults in both WSS and Smart grid system adopt

Kalman Filter (KF) and χ^2 detector. The χ^2-detector can detect myriad of system faults/attacks such as Denial of Service (DoS) attack, short term and long term random attacks. However, the study shows that the χ^2-detector is unable to detect the intelligent False Data Injection attack (FDI). To overcome this limitation, I present a Euclidean detector for smart grid which can effectively detect such injection attacks. Along with detecting attack/faults I also present the isolation of the attacked/faulty nodes for smart grid. For isolation the Generalized Observer Scheme (GOS) implementing Kalman Filter is used. As GOS is effective in isolating attacks/faults on a single sensor, it is unable to isolate simultaneous attacks/faults on multiple sensors. To address this issue, an Iterative Observer Scheme (IOS) is presented which is able to detect attack on multiple sensors.

Since network is an integral part of the future CPSs, I also present a scheme for preserving privacy in the future Internet architecture, namely MobilityFirst architecture. The proposed scheme, called Anonymity in MobilityFirst (AMF), utilizes the three-tiered approach to effectively exploit the inherent properties of MF Network such as Globally Unique Flat Identifier (GUID) and Global Name Resolution Service (GNRS) to provide anonymity to the users. While employing new proposed schemes in exchanging of keys between different tiers of routers to alleviate trust issues, the proposed scheme uses multiple routers in each tier to avoid collaboration amongst the routers in the three tiers to expose the end users.

INDEX WORDS: Smart Grid, Water Supply System, Kalman Filter, False Data Injection Attack

ENHANCING SECURITY IN THE FUTURE CYBER PHYSICAL SYSTEMS

by

Kebina Manandhar

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

2015

ENHANCING SECURITY IN THE FUTURE CYBER PHYSICAL SYSTEMS

by

KEBINA MANANDHAR

Committee Chair: Xiaojun Cao

Committee: Anu Bourgeois

Raj Sunderraman

Yi Zhao

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

May 2015

DEDICATION

I would like to dedicate this dissertation to my father, Kedar Bahadur Manandhar, and my family back home and in the US, who have all been very supportive throughout these last five years.

ACKNOWLEDGEMENTS

I want to express my gratitude to my advisor Dr. Cao for his immense help in providing me with research directions and directing the publications. I would also like to thank my PhD committee for providing me with valuable suggestions on the overall form of this dissertation. I am thankful to all the group members in my research group, specially Yang for his guidance. Lastly, this dissertation would not have been possible without the support of my husband, Ayush. I want to thank him for his immense support these past five years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- AMF - Anonymity in MobilityFirst network
- AS- Autonomous Systems
- CPS - Cyber Physical System
- DMap - Direct Mapping
- DoS - Denial of service
- FDI - False Data Injection
- GOS - Generalized Observer Scheme
- GNRS - Global Name Resolution Service
- GUID - Globally Unique Flat Identifier
- HEM - Home Energy Management
- IOS - Iterative Observer Scheme
- KF - Kalman Filter
- MF - MobilityFirst
- NA - Network Address
- SCADA - Supervisory Control and Data Acquisition
- WSS - Water Supply System

PART 1

INTRODUCTION

A Cyber Physical System (CPS) tightly combines and co-ordinates its computational/-cyber elements together with physical elements [1, 2]. In many CPSs, sensors are deployed to enable the communication between cyber and physical elements by converting the information gathered from the physical world to cyber data. For example, the authors of [3] designed the CPS employing sensor techniques to monitor the algae growth in *Lake Tai, China*. Their design comprises of sensors and actuators to monitor the order of severity of the algae bloom and to dispatch salvaging boats. Similarly, the authors of [4] proposed a CPS approach that navigates users in locations with potential danger, which takes advantage of the interaction between users and sensors to ensure timely safety of the users.

Along with the growth of interest in CPS, the security aspects of the system have also attracted significant attention. Recent attacks on cyber physical systems, such as STUXNET [5] and the malicious attack on the sewage system in Queensland, Australia (2000) by an ex-employee, further highlights the vulnerability of such systems and emphasizes the need to study as well as understand the security aspects of a CPS. Besides pure cyber attacks as in the case of STUXNET, sensors are equally prone to attacks and manipulations. Attacks on sensor readings allow attackers to mask the actual attack performed on the CPS, causing the attack to remain undetected. Thus, it is important to detect attacks/faults and isolate the attacked/faulty nodes in a timely manner. This dissertation focuses on the security aspects of two cyber physical systems: Smart Grid system and Water supply system.

With the advent of new technologies, the secluded power grid system is being replaced by grid, which is a typical smart CPS having more embedded intelligence and networking capability. Power grid is an important infrastructural backbone having a deep impact on economy as well as our daily activities. Failures in power grid often lead to catastrophic

effects as the ones in New York (2003) and Mumbai (2012). Though both of these failures resulted due to the faults in the system, security failures can also result in similar consequences, if not worse. In smart grid systems, cyber and physical components work in a complex co-ordination to provide better performance and stability. Sensors are equipped throughout the system to monitor various aspects of the grid such as the meter and voltage fluctuations in these systems. The collected information from the sensor networks help in providing feedback to the physical power grid devices. Hence, such a CPS involves a two-way communication between the controller system and the physical components as shown in Figure 1.1. This provides user with many new functionalities but at the same time it also provides attackers with more attack possibilities.

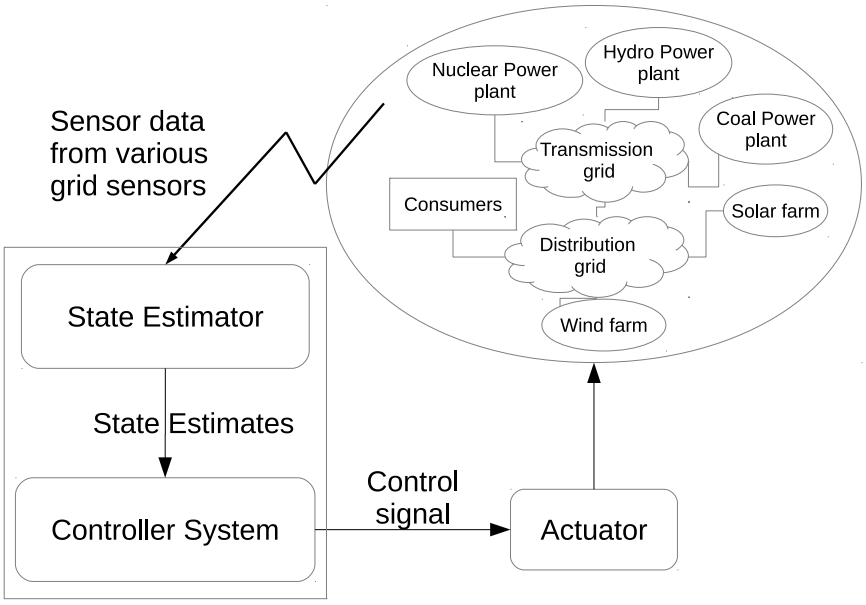

Figure (1.1) Block diagram for a smart grid system

In the following chapters, I present a security framework for smart grid using Kalman Filter (KF) and χ^2-detector. Since the χ^2-detector can not detect the statistically derived False Data Injection (FDI) attack, an additional detection technique using the Euclidean distance metric is also presented. A mathematical model is derived together with KF to detect possible attacks and faults on the smart grid system. The performance of the ex-

ploratory method, χ^2-detector, in identifying faults and random attacks is studied and the limitation of χ^2-detector in detecting the statistically derived False Data Injection attack is explored. I also present a new Euclidean detector to be coupled with KF and demonstrate the effectiveness of these approaches via extensive simulations and analysis on practical systems.

Along with detection, the isolation of attacked/faulty nodes in the system is also equally important. For attack isolation, I implement the Generalized Observer Scheme (GOS) based on Kalman Filter and χ^2-detector. GOS is effective in detecting attack on a single sensor. However, it fails to detect attacks/faults on multiple sensors at a given time. To address this issue I present a new Iterative Observer Scheme (IOS) to identify the sensors that are being arbitrarily attacked or are at fault in smart grids. The comparison between GOS and IOS shows IOS can effectively identify the attacks/faults on multiple sensors. Further, GOS employs m observers to identify a single misbehaving sensor in constant time. Compared to that, IOS uses the same amount of observers to detect all misbehaving nodes in approximately $O(log_\alpha m)$.

Similarly, Water supply system (WSS) is another sensitive area which can have dramatic public health and economical impacts when attacked. Water supply system deploys sensors and actuators to monitor water level and water velocity at a certain location and time. In this dissertation, I present a framework for attacks and faults detection in a water supply system by applying the Saint-Venant equations and Kalman Filter techniques. The Kalman Filter generates estimates for state variables using the mathematical model for the WSS and the data obtained from the sensor network deployed to monitor the WSS. A χ^2-detector is then employed to detect the discrepancies between the estimated data and the measured data, and trigger alarms. The χ^2-detector can effectively detect attacks like the DoS attack and random attack. FDI attack is also investigated on the sensors and elaborate corresponding defending approaches are discussed.

As discussed earlier network is an integral part of the future CPSs. With more and more static as well as mobile devices being connected to the network and with requirement for complex security protocols, it is no longer clear that the networking requirement for

the emerging and future technologies can be satisfied by making incremental changes in the current networking architecture. Hence, different possible future network architectures are being studied, for example Nebula [6], MobilityFirst [7]. Furthermore, in order to have a secure CPS it is crucial to have a secure networking system. Therefore, in this dissertation anonymity and security for the future internet architecture specifically for the MobilityFirst network is also studied.

With increasing number of smart devices attached to the network, keeping track of energy consumption and fluctuation becomes important as they enable the generation of energy signatures to track malicious activities. In this context, I also present our study on scheduling of electrical appliances based on their energy consumption patterns, in a building.

This dissertation is organized as follows. In Chapter 1 I discuss Kalman Filter. In Chapter 4 a security framework for smart grid using kalman filter and χ^2-detector is presented. Further, I show that χ^2-detector is unable to detect FDI attack and then present Euclidean detector that is able to detect FDI attack. In Chapter 5 two attack isolation schemes for smart grid are presented. These schemes are called GOS and IOS, and use filter banks that are based on KF. Chapter 7 further extends this framework to be used with WSS. In this chapter, I derive a mathematical model for WSS and implement KF to obtain the security framework. In Chapter 8 I present a scheme called Anonymity in MobilityFirst (AMF), which preserves user anonymity and provides security by implementing encryption and clever exchange of key for the MobilityFirst future internet architecture. In chapter 6 I present linear programming based load scheduling for energy management in smart grid. Finally I conclude in Chapter 9 with my findings.

PART 2

CYBER PHYSICAL SYSTEM

The term cyber physical system (CPS) refers to a new generation of intelligent systems with integrated computational and physical capabilities that allows interaction between computational and physical components [8]. In CPS the computational components (embedded computers and networks) monitor and control the physical processes. These processes are usually provided with feedback loops that provide feedback to the computational components [9]. The feedbacks help the computational components to monitor and control the physical processes.

While some form of CPSs are already in use [3][4], newer types of CPSs are emerging in many different areas. These areas range from autonomous vehicles and traffic systems to critical infrastructures like electric power distribution, oil and natural gas, water and waste-water distribution systems [10]. The disruption of these critical systems could have a significant health, safety and economic impact.

2.1 Security in Cyber Physical System

The tight integration between computational and physical systems introduces new security concerns in CPS. As stated in [11], the existing security approaches are either inapplicable, not viable, insufficiently scalable, incompatible, or simply inadequate to address the challenges posed by highly complex environments in a CPS. Since the CPS system are more complex and involves more devices than the traditional networks, attacks different from traditional cyber attacks that are unique to such systems are possible in CPS for example resonance attacks [12]. In such an attack, the compromised sensors or controllers force the physical system to oscillate at its resonant frequency. Data injection attacks are another class of attacks that are possible in CPS [13][14], in which the attackers compromise a set

of sensors and provide false data to the central system. These attacks are crafted to bypass the detectors and cause the system state to change which may cause the system to act in an unpredicted manner.

Similar to the traditional networks, the security goals for CPS include integrity, availability and confidentiality but their definition need to be adjusted for CPS [12]. Integrity refers to the trustworthiness of the data and refers to being able to prevent, detect and survive data integrity attacks in CPS. Availability refers to the ability of the system to be accessible and usable upon demand. The attack on availability such as DoS attack might have more severe impact on CPS since it also involves operation of time sensitive physical system. Confidentiality in CPS could refer to either user privacy in commercial CPS application or preventing adversary from eavesdropping on the communication channels between the sensors and the controller, and between the controller and the actuator [12].

2.2 Challenges in CPS security

Information security for traditional networking systems has developed mature technologies that can be used to prevent, detect and respond to attacks against control systems. However, the state-of-the-art research in computer security does not consider how attacks affect the estimation and control algorithms and ultimately, how attacks affect the physical world [12]. Moreover, most of the effort for protecting control systems focus on reliability i.e. fault detection and are wide open to malicious cyber attacks.

Much study in literature focus on the security of data communication from the physical components to the central controller or among different elements (e.g., sensors and actuators). For example, the authors in [15] propose an intrusion detection system to detect malicious nodes in the smart grid wireless network. Similarly, a distributed intrusion detection system is discussed in [16]. Recently, many emerging attacks specifically targeting communication and control systems in smart grid are exposed [15, 16, 17, 18, 19]. Similarly, deception and denial of service attacks against a networked control system are defined in [20]. Deception attacks compromise the integrity of either the control packets or measurements tampering the

sensors and actuators. DoS attacks compromise the availability of resources, for example by jamming the communication channel rendering the CPS unable to function properly. A more direct approach would be to tamper the physically unguarded monitoring sensors leading to the generation of false data. A general strategy to identify physical tampering is to deploy an estimator and a detector in the controller. The estimator compares the calculated estimates with the actual readings and verifies them [13],[14]. The detector triggers an alarm when the estimated states and measured states do not agree with each other. In other words, a significant difference between the estimated and measured states signifies either a fault or a possible attack on the system. However, the studies in [13],[14] show that a new type of attack called False Data Injection attack can be directed at the system if some system parameters are known to the attacker.

In this dissertation, I study security for two CPSs: smart grid and water supply system. For each of these CPSs the existing literature is discussed in depth at the beginning of each chapter. In addition to this, I also study security and privacy in MobilityFirst future internet architecture considering the importance of networking infrastructure in the CPSs.

PART 3

KALMAN FILTER

Since majority of our techniques discussed in this dissertation are based on Kalman Filter, we disucss Kalman Filter in this chapter. Kalman Filter is used to estimate the values of state variables of a dynamic system that is excited by stochastic disturbances and stochastic measurement noise. The Kalman filter is a set of mathematical equations that provides an efficient mechanism to estimate the state of a process such that the mean of the squared error is minimized [21]. The filter supports estimations of past, present and future states.

3.1 Principles of KF

The Kalman Filter [22, 21, 23] technique is used to obtain estimates for the state space vector x_t. Kalman Filter assumes the state at time t is evolved from the state at time $t-1$ according to the equation below.

$$x_t = Ax_{t-1} + Bu_{t-1} + w_{t-1} \tag{3.1}$$

where, A is the state transition model applied to the previous state x_{t-1}, B is the control input model applied to the control vector u_t and w_t is the process noise which is independent of the initial conditions. The observation equation for Kalman Filter can be written as:

$$y_t = C_t x_t + v_t \tag{3.2}$$

Here, y_t is the measurement vector, C_t is the observation model that maps state space into the observed space and v_t is the measurement noise which is independent of the initial conditions and process noise. Both w_t and v_t are assumed to be white Gaussian noise with

zero mean and standard deviation σ.

Kalman Filter can then be applied to compute state estimations $\hat{x}_t$. Let the mean and covariance of the estimates be defined as follows:

$$\begin{aligned}
\hat{x}_{t|t} &= E[x_t, y_0, ..., y_t] \\
\hat{x}_{t|t-1} &= E[x_t, y_t, ..., y_{t-1}] \\
\rho_{t|t-1} &= \Sigma_{t|t-1} \\
\rho_{t|t} &= \Sigma_{t|t-1}
\end{aligned} \tag{3.3}$$

Here, $\hat{x}_{t|t}$ is the estimate at time t using measurements up to time t, $\hat{x}_{t|t-1}$ is the estimate at time t using measurements up to time $t-1$. Similarly, $\rho_{t|t}$ is the covariance of the estimates at time t using data up to time t and $\rho_{t|t-1}$ is the covariance of the estimates at time t using data up to time $t-1$. Now, the iterations of Kalman Filter can be written as:

Time Update:

$$\begin{aligned}
\hat{x}_{t+1|t} &= A\hat{x}_t \\
\rho_{t|t-1} &= A\rho_{t-1}A^T + Q
\end{aligned} \tag{3.4}$$

The Equation (3.4) projects the state and covariance estimates at $t+1$ time step from t time step. Here, A is obtained from the state space model and Q is the process noise covariance matrix.

Measurement Update:

$$\begin{aligned}
K_t &= \rho_{t|t-1}C_t^T(C_t\rho_{t|t-1}C_t^T + R)^{-1} \\
\rho_{t|t} &= \rho_{t|t-1} - K_tC_t\rho_{t|t-1} \\
\hat{x}_t &= \hat{x}_{t|t-1} + K_t(y_t - C_t\hat{x}_{t|t-1})
\end{aligned} \tag{3.5}$$

Equation (3.5) represents the measurement updates of the Kalman Filter. K_t is the

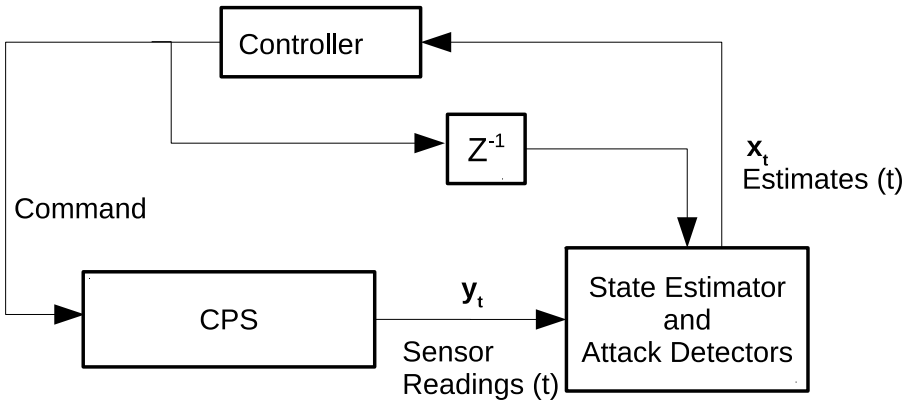

Figure (3.1) Cyber Physical System

Kalman gain and R is the measurement noise covariance matrix. The last two formulae in Equation (3.5) are used to generate a more accurate estimate by incorporating the measurements y_t. The initial condition is $x_{0|-1} = 0, \rho_{0|-1} = \Sigma$. We assume that the Kalman gain converges in a few steps and is in a steady state. Finally, a training period is assumed such that the filter knows the Kalman gain before the estimation, then,

$$\rho \triangleq \lim_{k\to\infty} \rho_{t|t-1},$$
$$K = \rho C^T (C\rho CT + R)^{-1} \tag{3.6}$$

Equation (3.4) can be further updated as:

$$\hat{x}_{t+1} = A\hat{x}_t + K[y_{t+1} - C(A\hat{x}_t)] \tag{3.7}$$

Thus, applying the equations above we can obtain the estimates for the state variables from the previous state variables and the measurements obtained.

3.2 KF applications

Since R.E. Kalman published his famous paper on Kalman Filter in 1960, Kalman filter has been the subject of extensive research and application. Some of the applications are discussed in this section. Applications of KF expands to wide range of areas such as weather forecasting, target tracking, economics and SCADA system state estimation. In [24] KF is used to forecast wind speed for wind power prediction. Application of KF in airborne radar tracking system is studied in [25]. Analysis of the extended Kalman filter applied to bearings-only target tracking is presented in [26]. Similarly, position and velocity tracking using KF is studed in [27]. In [28] monthly retail sale estimates revision using KF is studied. Application of Kalman Filter techniques to power system dynamic state estimation is studied in [29].

In this dissertation we use KF to predict next state of the smart grid system and the water supply system. For the smart grid system we consider two different models, first using sinusoidal voltage equation and second using DC Model. We use Saint-Venant equation to derive a model for the WSS system to be used with KF. Each of these models and their integration with KF is discussed in more detail in the upcoming chapters.

PART 4

DATA INJECTION ATTACK IN SMART GRID

In this chapter, we classify and comprehensively analyze the state-of-the-art studies in security of smart grid system. As data injection attacks are one of the major concerns in security of smart grid, we present different mechanisms to detect various types of data injection attack.

4.1 Data Injection Attack Overview

Smart grid relies on wired/wireless networking infrastructures to integrate the control system and physical power grid system. Thus, it is important to understand and defend against cyber attacks that emerge from both the networking and the control infrastructures. The addition of wired/wireless communication capabilities in the existing power grid system result in increasing complexity, and potentially more holes in security. The smart grid system can incorporate the traditional security measures (e.g. intrusion detection and firewall) to prevent rudimentary attacks like the ones in traditional data networks.

The existing studies on the security of smart grid, can be broadly categorized into three categories. The work in the first category deals with the wired/wireless networking security among cyber components in the smart grid [15][16][17][18][19][30][31]. The papers in the second category investigates the early detection of anomalies in the system. Smart grid is a real time system and faults/attacks must be handled as soon as possible. The early anomaly detection schemes [32][33] can pro-actively protect the system. The work in the third category applies the control theories in the security process using various state estimation and detection techniques [34] [35] [36].

A wireless mesh network architecture was proposed in [15] for the smart grid system and an intrusion detection scheme called smart tracking firewall was introduced. To over-

come the security pitfalls such as signal jamming and eavesdropping, the authors in [15] also investigated the anti-jamming, physical layer security technique coupled with smart tracking firewall. The proposed firewall consists of two agents: intrusion detection agent and response agent. The agents maintain two lists of misbehaving nodes called the black-list and the grey-list. These lists keep tracking the malicious nodes in the network. Another distributed intrusion detection scheme was discussed in [16], which deploys an intelligent module and an analyzing module along with an artificial immune system to detect and classify malicious data as well as possible attacks on the smart grid. In [17], secure estimation of the system states is discussed. The channel capacity requirement to ensure negligible information leakage to the adversary regarding the system states and control message is studied in this chapter. A message authentication scheme was proposed in [18] to achieve the mutual authentication among the smart meters in the smart grid using shared keys and hash based authentication techniques. Another signature based message authentication scheme was proposed in [19], which employs the multicast authentication to reduce the signature size and communication bandwidth at the cost of increased computation. As suggested in the paper, such authentication scheme is more desirable in the smart grid, where there is a limitation in the storage size and bandwidth. The work in [30] uses the data concatenation and random drop schemes to defend traffic analysis attack and the study in [31] is about defending the internet based load altering attack.

Unlike the papers discussed above that focus mainly on the protection of data communication in the smart grid, the authors in [32] presented an early warning scheme to predict/prevent anomalous events in advance. The proposed approach consists of detection, reaction, data recollection and alarm management components. Anomaly detection in the existing power grid substation was studied in [33], which presents an anomaly inference algorithm based on the combination of transaction-based model, hidden Markov model and feature-aided tracking.

An attack/fault in the smart grid system is always reflected in the form of change in either voltage, current or phase [34]. The work in [34] proposed a control-theoretic adaptation

framework for the system level security of smart grid. The control-theoretic framework uses the state estimation technique to estimate the data from the remote terminal units and applies power security analysis tools to detect attacks on the system. However, the proposed distributed state estimation owns slightly larger data estimation error [34]. Similarly, the protection for the set of meter measurements or changes was discussed in [35] [36]. The identification and verification of set of sensor measurements that are required to be protected in order to detect existence of False Data Injection attack in the smart grid is discussed in [36].

As discussed above, much of the present work is based on various security techniques that are originally developed for securing the Internet data communication. While these techniques are effective in securing the Internet, these security techniques alone are not sufficient to deal with attacks in a more complex CPS such as the smart grid system. Particularly, the existing techniques did not address the new class of attack called False Data Injection attack [13]. This type of injection attack is undetectable by detectors used in the existing state-estimation security frameworks [13],[14]. Hence, this work presents a framework, based on a state space model derived from the voltage flow equations, to defend different types of attacks and faults including the False Data Injection attack. We show that the False Data Injection attack cannot be detected using a traditional combination of estimator and detector (i.e., KF and χ^2-detector). Then, we present a different detector based on Euclidean distance metric to detect the complicated False Data Injection attack on the power grid system.

4.2 Proposed Framework For Smart Grid using Kalman Filter

In this section, we present the detailed description of the security framework for smart grid using Kalman Filter (KF). The framework is capable of detecting various attacks including short termed and long termed random attacks along with the powerful False Data Injection attack on the power system. We develop a state space model (as shown in Section 4.2.1) from the 3-phase sinusoidal voltage equations, to integrate with the technique of

Kalman Filter. Without loss of generality, we assume the use of voltage sensors to measure the state variables (e.g., amplitude and phase of the voltage) in the framework. The sampling rate for the sensors is assumed to be around 16 samples per 60 Hz cycle i.e. about 960 samples per second for medium to low data rate production [37].

Figure 4.1 shows the security framework where Kalman Filter estimates the values for the state variables based on the system state and the data from the numerous sensor readings. The estimated values generated by KF and the observed values for the state variables are fed into the detector. The detector compares the two state vectors (consisting all the state variables). If the two differ from each other significantly and are above a certain precomputed threshold, the detector triggers an alarm to signify a possible attack on the smart grid. As the literature study shows, the χ^2-detector is a typical choice for the Kalman Filter estimators [38] when the residue of the KF equations follows gaussian distribution and $g(t)$, (as in Equation (7.17)), follows the χ^2 distribution [14]. Attacks such as DoS attack and random attack are readily detected by the KF and χ^2-detector combination. However, False Data Injection attack can bypass such detectors and may remain undetected [14]. Hence, we use an additional detector, based on the Euclidean distance, along with χ^2-detector. The Euclidean distance detector reconstructs the sinusoidal voltage signal from the state parameters and calculates the difference between the estimated and observed voltage signals. If the difference is larger than a precomputed threshold, the detector triggers an alarm.

4.2.1 State Space Model

Power system deploys sensors or meters such as phasor measurement units to measure the system state at various locations and time to ensure a smooth operation of the power system. These meters are able to measure current phase and amplitude [39]. The measurements obtained from these meters/sensors are the state variables that are reported to the central controller via the wired/wireless communication infrastructure. As stated in [13], the state variables may include bus voltage, angles and magnitudes. Therefore, the state space model should reflect these properties of the power system. The study in [34] indicates that

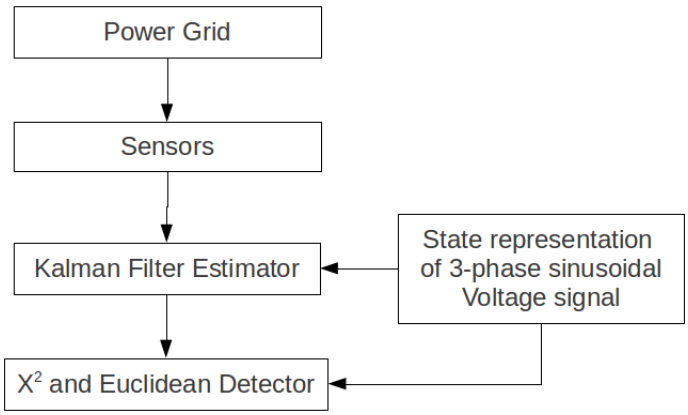

Figure (4.1) Security framework for the smart grid system

an attack or fault in the power system is always reflected in the form of change in either voltage, current or phase. Without loss of generality, we derive the state space model from the power grid voltage signal.

The voltage signal can be represented as a sinusoidal wave [40] as shown in Equation (8.1). The equation represents voltage as a function of amplitude (A_v), angular frequency wt and phase ϕ at discrete time. Equation (2) and (3) are mentioned here to represent the 3-phase voltage signal. For simplicity, we only consider Equation (8.1) in the process of developing the model.

$$V_1(t) = A_v cos(\omega t + \phi) \tag{4.1}$$

$$V_2(t) = A_v cos(\omega t + \phi - \frac{2\pi}{3}) \tag{4.2}$$

$$V_3(t) = A_v cos(\omega t + \phi - \frac{4\pi}{3}) \tag{4.3}$$

Equation (1) can be expanded as follows,

$$V_1(t) = A_v * cos\omega t * cos\phi - A_v * sin\omega t * sin\phi \tag{4.4}$$

Assuming the angular frequency is relatively constant over time, we consider amplitude

and phase as the variables in the state space representation. The equation then becomes,

$$V_1(t) = x_1 * cos\omega t - x_2 * sin\omega t \tag{4.5}$$

where, $x_1 = A_v * cos\phi$ and $x_2 = A_v * sin\phi$ are defined as the state variables. Assuming there is no additional delay in the system and considering random noise or small errors picked up by the system, we have Equation (7.4) representing the state equation over the time.

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + w(t) \tag{4.6}$$

Equivalently,

$$x(t+1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x(t) + w(t) \tag{4.7}$$

where, $x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$ and $w(t)$ is the process noise. Process noise represents the unmodeled system dynamics or the disturbance inputs in the system model.

The actual voltage signal for the current state using non stationary deterministic vector $[cos\omega t \ \ -sin\omega t]$ can be obtained using Equation (7.2) and can be written as shown in Equation (7.5), where $\gamma(t)$ represents the measurement noise.

$$y(t) = [cos\omega t \ \ -sin\omega t] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \gamma(t) \tag{4.8}$$

4.2.2 Kalman Filter

Figure 7.1 shows the control system with the KF embedded for the estimation of the state vector and detector for the detection of attacks or faults. As shown in Figure 7.1, $x(t)$ denotes the output of the state estimator that is fed to the controller and Z^{-1} is the control system feedback. The observations or sensor readings $y(t)$ are forwarded to the estimator at

a regular time interval denoted by Δt. At each time step Δt, the estimator of the system generates estimated readings based on the estimates $x(t-1)$ from previous time step and the real-time sensor readings $y(t)$.

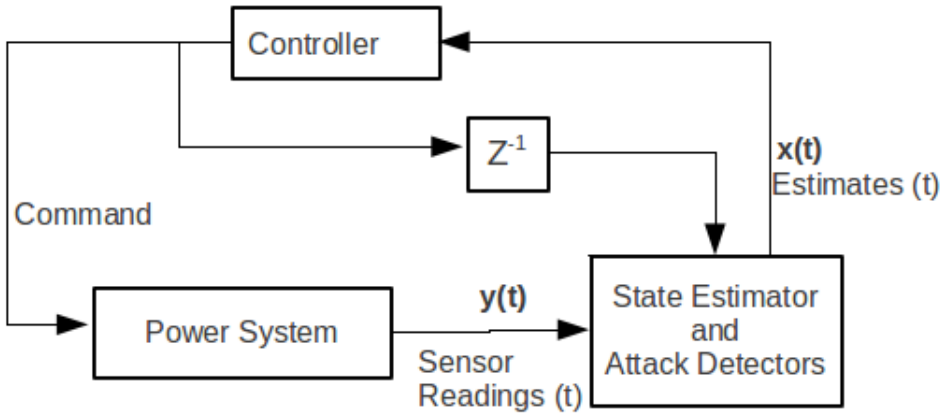

Figure (4.2) Power grid system

To apply the Kalman Filter technique, the state equation can be written as,

$$x(t+1) = Ax(t) + w(t) \tag{4.9}$$

where, A $= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

From Equation (7.5), the observation equation for Kalman Filter can be written as:

$$y(t) = C(t)x(t) + v(t) \tag{4.10}$$

Here, $y(t)$ is the measurement vector collected from the sensors, $C = [cos\omega t - sin\omega t]$, $v(t)$ is the measurement noise and assumed to be white Gaussian noise with mean 0 and standard deviation σ, which is independent of the initial conditions and process noise.

Kalman Filter can then be applied to compute state estimations $\hat{x}(t)$ using equations 4.9 and 4.10.

4.2.3 Generalization of the model

The State space model described in Section 4.2.1 can be generalized for power grid measurements. The voltage at any given bus can be obtained in the form of a sinusoidal wave (or phasor representation) using Kirchoff's Voltage Law (KVL) and/or Kirchoff's Current Law (KCL). Let us consider a 3-bus system as shown in the Figure 4.3 as an example. The voltage amplitude ($|V_i|$), phase (ϕ_i), active power (P_i) and reactive power (Q_i) are the variables in this system. Given a set of known initial values, the values for the unknown variables at each bus is obtained by solving Equation (4.11) [41]. These equations are produced by applying KCL at each node. Hence, by solving the power flow problem for the 3-bus system below, the voltage amplitude $|V_i|$ and phase ϕ_i at each bus are calculated.

$$
\begin{aligned}
&For\ any\ bus\ 'i'\ : \\
&P_i = \sum_{k=1}^{n} |V_i||V_k|(G_{ik}cos(\phi_i - \phi_k) + sin(\phi_i - \phi_k)) \\
&Q_i = \sum_{k=1}^{n} |V_i||V_k|(G_{ik}sin(\phi_i - \phi_k) - cos(\phi_i - \phi_k))
\end{aligned}
\tag{4.11}
$$

where, $|P_i|$ and $|Q_i|$ are the active and reactive power at bus i. $|V_i|$ and ϕ_i are the voltage magnitude and phase at bus i and $Y_{ik} = G_{ik} + jB_{ik}$ are the Y-bus elements.

As describe in Section 4.2.1, the state variables are $x_1 = A_v * cos\phi$ and $x_2 = A_v * sin\phi$. Assuming the system has reached the stable state, these values of $|V_i|$ and ϕ_i for bus i, obtained by solving the power flow equations, can be plugged in to obtain the initial values for the state variables at $t = 0$. Once the initial states are known, Equation (4.7) and (7.5) for the KF of bus i can be used to estimate values for next time step.

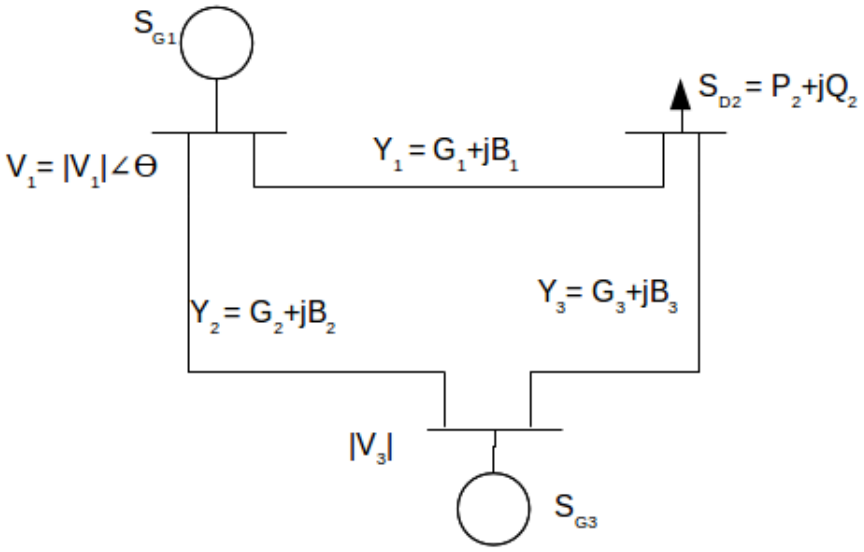

Figure (4.3) 3-bus system

$$
\begin{aligned}
&\textit{At bus } i: \\
&x_1(0) = |V_i| cos\phi_i \\
&x_2(0) = |V_i| sin\phi_i
\end{aligned} \tag{4.12}
$$

Equation (4.11) accounts for effect of all the generators and loads in the system at each bus. Hence, for any bus i, $|V_i|$ and ϕ_i obtained after solving this equation reflects the effect of all the system parameters. An attack/fault on any bus or branch in the system, is reflected in the form of change in the values of these variables. Since the KF described above uses the values of $|V_i|$ and ϕ_i obtained by solving Equation (4.11) as its initial state, any deflection in the values due to any attack/fault will cause the values of the state variables to deviate from the estimated values.

4.2.4 Attack Model

It is assumed that the attacker is able to control a subset of the sensor readings in the system. Three types of attacks are considered in this chapter: DoS attack, random attack and False Data Injection attack.

Denial of Service (DoS) Attack Denial of Service (DoS) attack is a form of attack where an adversary renders some or all the components of a control system inaccessible. DoS attack can be launched by jamming of the communication channels, flooding packets in the network, compromising devices to prevent data transfer etc. by the adversary [20]. DoS attack could be on sensor data, control data or both. DoS attack is modeled as the lack of availability of the sensor data.

Random Attack In this case, the attacks are not crafted to overcome the detection mechanism implemented by the central system. As describe in Equation (4.13), the attacker simply manipulates the sensor readings.

$$y'(t) = C(t)x'(t) + v(t) + y_a(t) \tag{4.13}$$

where $y_a(t)$ is the random attack vector generated by the attacker. When the system is under attack $y'(t)$ and $x'(t)$ denote observations and states. These random attacks could be generated at any point in time and could be a long term continuous attack or a short attack.

False Data Injection Attack In case of False Data Injection attack, it is assumed that the attacker knows the system model including parameters A, B, C, Q, R and gain K [42]. The attacker can also control a subset of sensors (S_{bad}). The attack model can then be described in Equation (4.14),

$$y'(t) = C(t)x'(t) + v(t) + \tau y_a(t) \tag{4.14}$$

where $\tau = diag(\gamma_1, ...\gamma_m)$ is the sensor selection matrix; $\gamma_i = 1$ if and only if $i \in S_{bad}$, otherwise $\gamma_i = 0$; and $y_a(t)$ is the malicious input from the attacker.

4.3 Attack/Failure Detector

The KF estimator calculates the following state of the system using the equations described in Section 5.3. As the meter readings for that state become available, the projected estimates and the actual meter readings are compared by the detector. If the difference between the two is above a precomputed threshold, an alarm is triggered to notify a possible attack or failure. As previously discussed, the framework presented in this chapter implements two types of detectors: The χ^2-detector and the detector implementing the Euclidean distance metric.

4.3.1 χ^2-detector

The χ^2-detector is a conventional detector used with Kalman Filter. As described in [38], the χ^2-detector constructs a χ^2 test statistics from the Kalman Filter and compares them with the threshold obtained from the standard χ^2 table.

Now, the residue z_{k+1} at time $k+1$ is defined as:

$$z(t+1) \triangleq y(t+1) - \hat{y}(t+1|t)$$

Equivalently,

$$z(t+1) \triangleq y(t+1) - C(A\hat{x}(t)) \tag{4.15}$$

Then, the χ^2-detector test consists of comparing the scalar test statistics given by:

$$g(t) = z(t)^T B(t) z(t) \tag{4.16}$$

where, $B(t)$ is the covariance matrix of $z(t)$. The χ^2 detector compares $g(t)$ with a pre-computed threshold obtained using the χ^2-detector-table [38] to identify a failure or attack. The χ^2 test is long-term test because, at each detection step, all integrated effects since system start time are considered. This property makes it very useful for the fault detection in smart grid which consists of sensors that are subject to soft failures like instrument bias shift. Another advantage of χ^2 detector is its computational complexity. The parameters required to perform the test are already generated by the Kalman Filter making it compatible with the KF. Furthermore, the threshold for the detector can be easily obtained from the χ^2-table making the threshold computation relatively easy. In our experiments, the threshold is chosen such that error rate is less than 5%.

False Data Injection attack is characterized by an attack sequence y_a such that $\limsup_{t\to\infty} ||\Delta x(t)|| = \infty, ||\Delta z(t)|| \leq 1, t = 0, 1, ...$ where, $||\Delta x(t)|| = x_a(t) - x(t)$, $||\Delta z|| = z_a(t) - z(t)$ and $x_a(t)$ and $z_a(t)$ are state variables and residue of the compromised systems [42]. This definition shows that the χ^2-detector may fail to detect False Data Injection attack on the sensors [14]. Thus, we introduce the Euclidean-based detector in the following section.

4.3.2 Detector implementing the Euclidean Distance Metric

Though χ^2 detectors have a high noise tolerance and work in most cases, attacks like False Data Injection attack fails to get detected [14]. This phenomenon is also visualized in the simulation results in the Section 4.4. The False Data Injection attack is a class of attack which is carefully crafted to bypass the statistical detector like χ^2-detectors. Thus to detect these types of attacks, we present an Euclidean-based detector, which calculates the deviation of the observed data from the estimated data. To apply the Euclidean detector, we first reconstruct the sinusoidal signals from the state estimates and then compare them with the measurements obtained from the sensors as shown in Equation (7.19),

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2} \tag{4.17}$$

where, p is the amplitude of the voltage signal and q is the amplitude of the estimated voltage signal.

If the difference between the two is greater than the threshold, as in case of χ^2 detector, an alarm is triggered. To minimize the false positives caused due to the noise, we set the threshold to 3σ (σ being the standard deviation of the noise from Section 5.3). As stated earlier, given the Gaussian noise with zero mean, setting the threshold to 3σ can filter out 99.73% false positives due to noise.

Under steady state the input signal can be reconstructed by applying the values of state variables in Equation (7.3). Similarly the estimated signal can be reconstructed using Equation (7.3) and $\hat{x}(t)$ from Equation (3.5).

Hence, the following corollary can be obtained from the definition of False Data Injection attack in Section 4.3.1.

Given $\limsup_{t\to\infty} ||\Delta x(t)|| = \infty$,

$$\lim_{t\to\infty} d(V_a(t), V(t)) = \infty$$

where, $d(V(t), V_a(t)) = \sqrt{(C(t)\hat{x}_a(t) - C(t)x'(t))^2}$.

As $||\Delta x(t)||$ tends to ∞, $d(V(t), V_a(t))$ approaches ∞ as well. Therefore, we can detect attacks and faults that results from the manipulation of the measured signal such as False Data Injection attack.

4.4 Implementation and Performance Evaluation

We implemented the KF estimator, χ^2-detector and Euclidean detector using Matlab. The experimental setup and the initial values are shown in Table 4.1. A 60Hz Sinusoidal voltage signal with random Gaussian noise is generated and fed to the Kalman Filter estimator as the input. Matlab function $randn()$ is used to produce normally distributed noise with mean value zero. The input signal and the resulting sinusoidal signal obtained using the state estimates are plotted in Figure 8.1-9 and Figure 11. Each of these figures contains

two graphs and shows the results of the simulation plotted against time. The top sub-graph shows how the amplitude varies with time for the input sinusoidal signal and the signal constructed using estimated state variables. In the bottom sub-graph, the value for $g(t)$ from Equation (7.17) are plotted against time. The straight horizontal line is the threshold obtained from the χ^2 table. For the Euclidean detectors, $d(p, q)$ from Equation (7.19) is plotted against time.

Table (4.1) Experimental setup

Frequency	60Hz
Amplitude	1 Volt
Sampling frequency	2 KHz
Initial value for $x_1(0)$	0
Initial value for $x_2(0)$	0
Initial covariance matrix $P(0\|0)$	Identity matrix

4.4.1 Attack/Fault detection using χ^2 detector

Figure 8.1 shows the simulation results using the χ^2-detector in the absence of attacks/faults for a certain period of time. It can be seen that the estimated values obtained from the KF estimator overlaps with the input signal denoting there is no difference between the estimated and the observed value. Hence for $g(t)$ obtained from the detector stays within the threshold. Since our simulations also consider the random noise in the system, there is a slight difference between the estimates and the input signal in the beginning. However, Kalman Filter works iteratively by correcting its estimates using both the state space model and the measurements obtained, the estimates gradually converge with the input signal.

In case of attacks, the estimated values do not match with the observed values and $g(t)$ exceeds the threshold as shown in Figure 4.5. As a result, the detector triggers an alarm signifying an attack/fault in the system. Similarly, Figure 4.6 shows a short-timed attack being detected by the framework. Figure 4.7 shows detection of DoS attack.

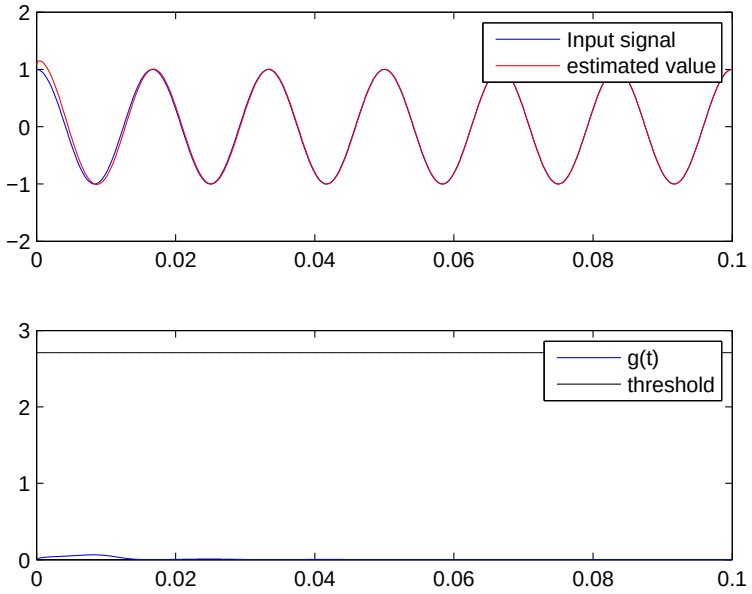

Figure (4.4) χ^2-detector when there is no attack/fault

4.4.2 False Data Injection Attack

False Data Injection attack injects fake sensor measurements that can fool the system implementing KF-estimator with χ^2-detector as described in [42]. The attack sequence can be obtained from Equation (4.18).

$$y_a(n+t) = y_a(t) - \frac{\lambda^{(i+1)}}{M} y^* \tag{4.18}$$

where, n is the dimension of state space, $y^* = Cv$, v=eigenvector of A, $|\lambda| >= 1$, $M = max_{k=0..n-1}||\Delta z(k)||$ and $\Delta z(k) = z'(k) - z(k)$. $z'(k)$ is the residue when the system is under attack.

The derivation of the attack sequence [42] ensures that it bypasses the detector and increases the error in the state estimation. The second sub graph in Figure 4.8 shows the behavior of χ^2-detector under the False Data Injection attack. We can see the estimates

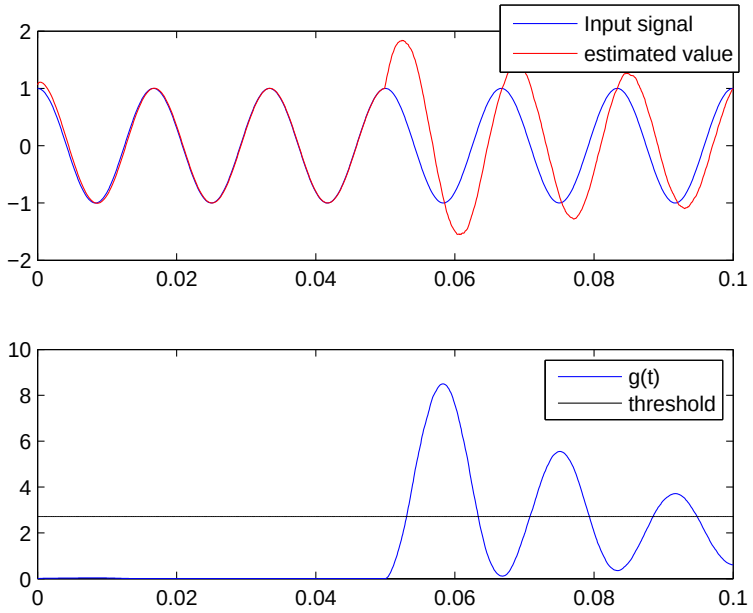

Figure (4.5) Continuous random attack detected using χ^2-detector

do not agree with the measured values in the top sub graph in Figure 4.8. However, $g(t)$ never exceeds the threshold. In other words, the graph shows that the statistical tests in χ^2-detector fails to detect the False Data Injection attack. We address this drawback in the next section, by using Euclidean detector, which can identify such attack by constantly monitoring the difference between the estimated values and the measured values.

4.4.3 False Data Injection attack detection using Euclidean Detector

The Euclidean detector compares the difference between the measured data and the estimated data based on the Euclidean distance metric as shown in Equation (7.19). Since the state variables only consider the time-invariant components of a sinusoid, the state variables remain relatively constant as described in Section 4.2.1. Thus, change in state variables could mean either attack or fault in the system. However, to avoid false alarms due to measurement or system errors, we set the threshold to 3σ as discussed in Section 4.3.2. Figure 5.1 shows the plot of the Euclidean Distance metric when there is no attack

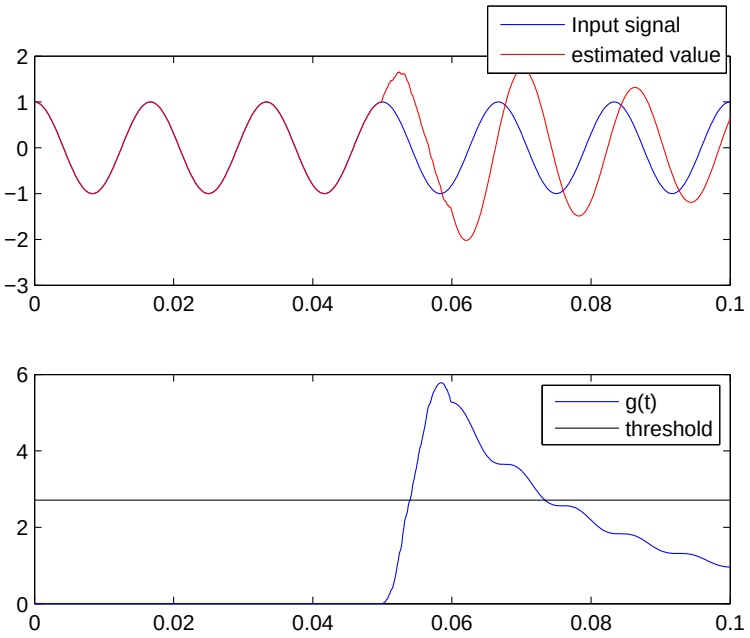

Figure (4.6) Random attack for a short period of time detected using χ^2-detector

in the system and the bottom sub graph in Figure 5.1 shows the plot when there is a False Data Injection attack in the system. When there is an attack in the system, the difference between the two curves exceeds the threshold, hence the False Data Injection attack can be detected by the Euclidean distance metric.

4.4.4 Load Change

In the model derived in this chapter, it is assumed the load in the system remains constant. In case there is change in load, there will be change in the voltage signal across the buses. If the load profile is known, then the change in voltage amplitude/phase caused due to the load change can be predicted. Assume there is a change in the voltage due to load change as shown in Figure 4.10. The parameters in the Kalman Filter can be adjusted to reflect the change in the voltage due to the load change. This allows us to obtain estimates for the state variables after the load change. Figure 4.10 shows that the estimates closely follow the signal with the load change. At time step 0.07, the random attack is detected by

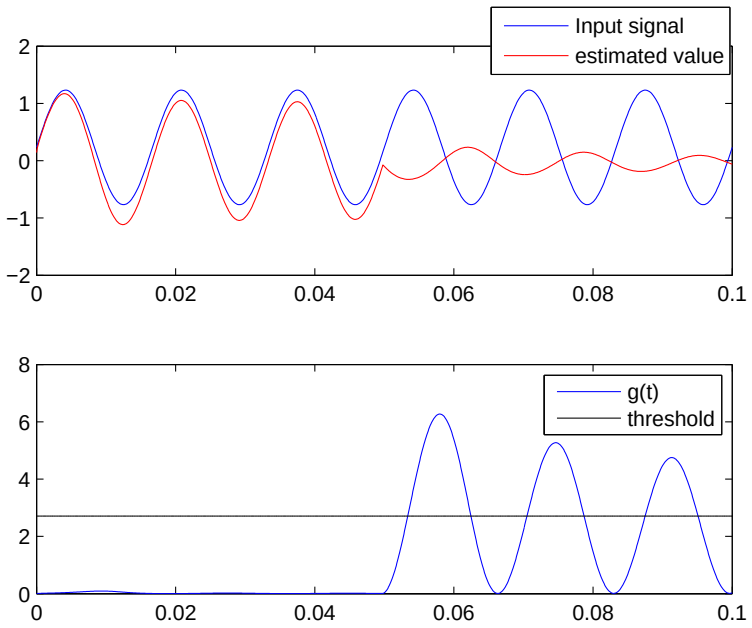

Figure (4.7) DoS attack detected using χ^2-detector

both the χ^2 detector and Euclidean detector in this scenario.

4.4.5 χ^2-detector vs. Euclidean detector

The probability of attack detection in both detectors are largely dependent on the value of the threshold. In the case of χ^2-detector, the threshold is obtained from the standard χ^2 table. Similarly, in the case of Euclidean detector, the threshold is obtained from the standard deviation of Gaussian distribution. In this experiment, we set the value of the thresholds in both detectors to filter 99% of noise. Thus the probability of false alarms due to noise is less than 1%.

In general, the Euclidean detector is more sensitive towards changes than χ^2-detector. As can be seen in Figure 4.11, Euclidean detector is faster in responding to the changes. Hence, if the noise parameters for the system are known, Euclidean detector gives better results. If the noise parameters are not known in advance, χ^2-detector is preferable since it handles the soft errors better. However, a disadvantage of the χ^2-detector relative to

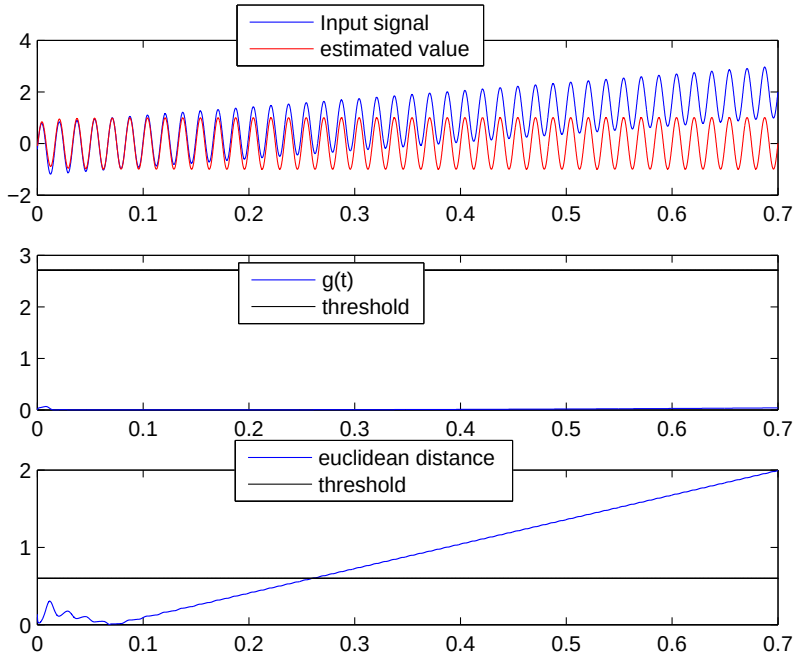

Figure (4.8) False Data Injection attack using χ^2-detector

Euclidean detector is its inability to detect False Data Injection attack. Figure 4.8 shows the reaction of χ^2 detector and Euclidean detector when the system is under the False Data Injection attack.

Euclidean detector reconstructs the signal from the state estimates and compares it with the measured signal whereas χ^2-detector only computes the residue vector. Since reconstruction requires more computation, Euclidean detector is more resource intensive than χ^2-detector.

4.4.6 Implementation of the security framework in IEEE 9-bus system

Figure 4.12 shows an IEEE 9-bus system with sensors to monitor the state parameters and the estimator/detector for bus 3. The 9-bus system is simulated using MATPOWER [43] package in MATLAB. The voltages and phases, obtained by solving the IEEE 9-bus power system in MATPOWER, are used as the state parameters in the Kalman Filter estimator.

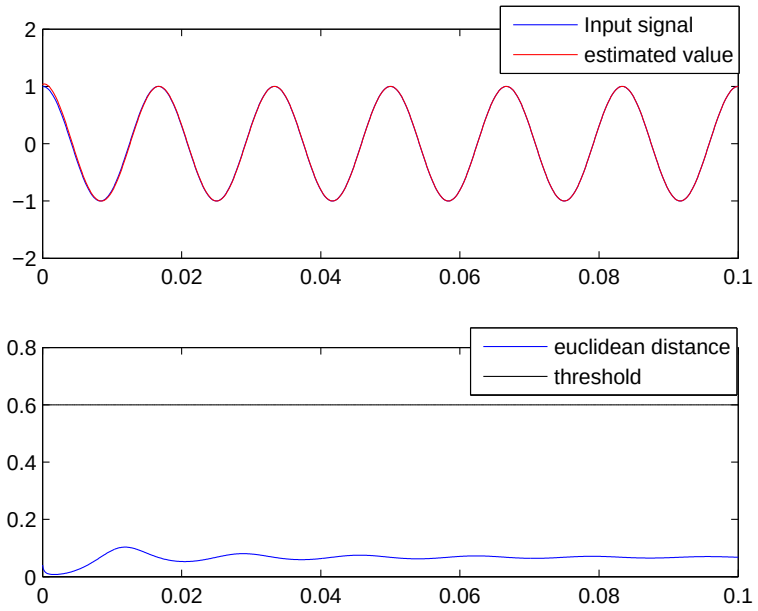

Figure (4.9) Euclidean detector when there is no attack/fault

Similar structure can be assumed for each bus in the system. For the simplicity, only bus 3 is discussed here. The attack sequence y_a is generated by the adversary as discussed in [42]. The sensors in the bus report their readings to the corresponding Kalman Filter estimators and Euclidean detectors. The successful detection of the False Data Injection attack on bus 3 is shown in the Figure 4.13.

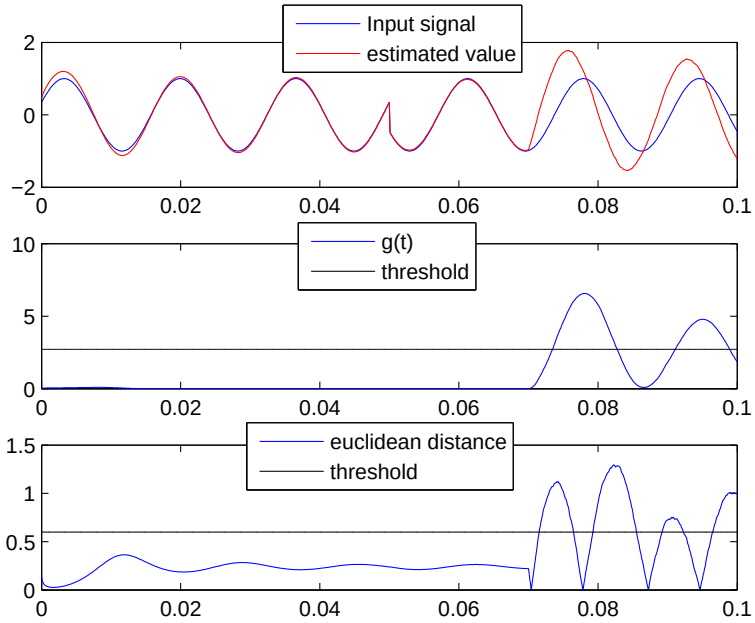

Figure (4.10) Change in voltage due to load change

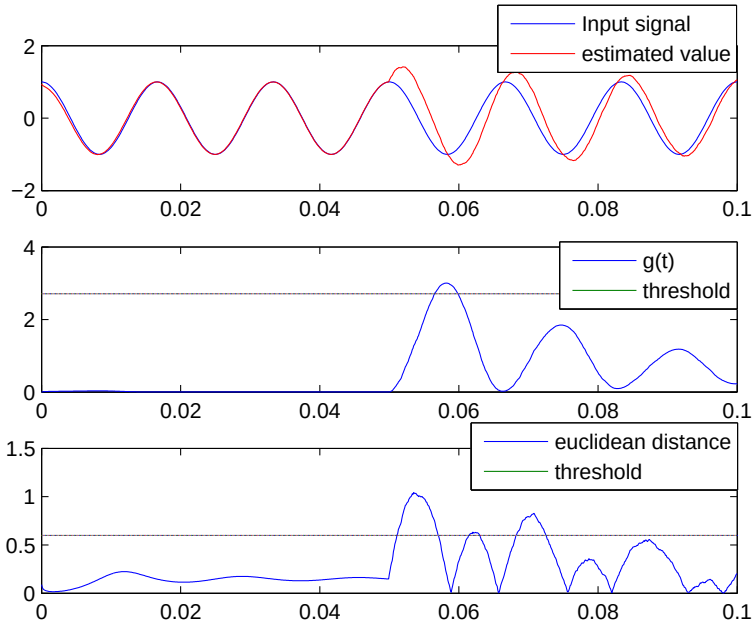

Figure (4.11) Performance of both detectors under the random attack

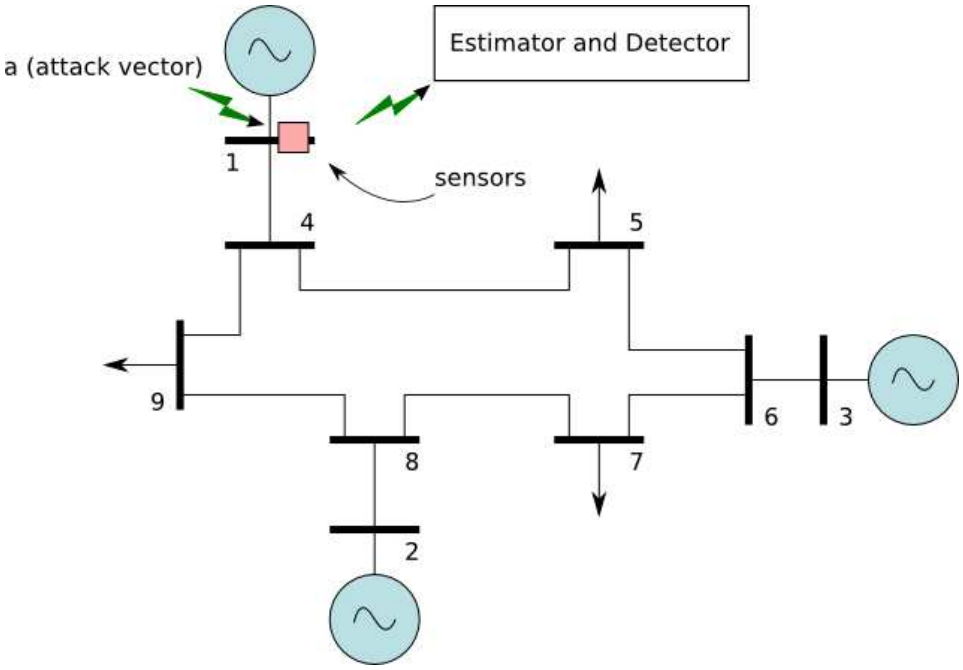

Figure (4.12) IEEE 9-bus system under False Data Injection attack

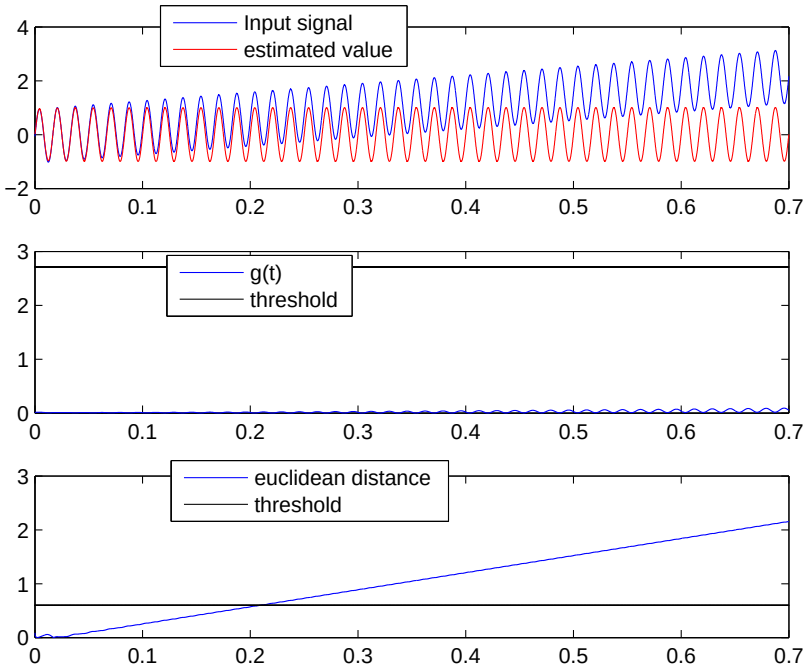

Figure (4.13) False data attack detection for bus 3 in IEEE 9-bus system

PART 5

ATTACK DETECTION AND ISOLATION IN SMART GRID

In this chapter, attack detection and isolation in smart grid is presented. Attack isolation provides mechanism to identify particular node that is under attack. Once the node is identified, necessary actions can be taken to recover the system from the attack.

5.1 Attack Isolation in Smart Grid Overview

While it is important to detect the attacks and faults in a CPS it is equally important to identify and isolate specific attacked or failed nodes. In cyber physical systems such as smart grids, attack isolation is still open and challenging [44] [45] [46] [47] [45] [48]. Fault Detection and Isolation (FDI) techniques have been used in control systems for identifying faults and the locations. FDI techniques can be broadly classified into model-based FDI and signal processing based FDI. Model-based FDI schemes or observer schemes depend on the measurement residual generated from the observers/filters such as Kalman Filter to facilitate decision making process [44]. Two observer schemes namely, Dedicated Observer Scheme (DOS) [45] and Generalized Observer Scheme (GOS) [46] can be used for fault detection and isolation in traditional control systems. GOS scheme is discussed in [49] and [48] for fault detection in doubly fed induction generators. Similarly, fault detection and isolation scheme for fractional order systems using the concept of GOS is proposed in [47]. In case of DOS, each observer is driven by a single instrument [45]. DOS is able to detect and isolate multiple faults at the same time. However, in order to implement DOS, it requires the system to be observable with any single output, which may not be feasible due to the complication or cost in a large system [48].

In the following sections we present a new Iterative Observer Scheme (IOS) to identify the sensors that are being arbitrarily attacked or are at fault in smart grids. The presented

IOS is based on the technique of Kalman Filter and χ^2-detector. With IOS, the readings from all the sensors are fed into Kalman Filter for attacks/fault detection. Once an attack or fault is perceived in the system, a smaller subset of sensors consisting of misbehaving sensor(s) are iteratively selected for processing each time to locate the attacked/faulty nodes. We also implement the Generalized Observer Scheme (GOS) based on the system model. Our study and comparison show that GOS identifies the attacks/faults on a single sensor while the IOS can effectively identify the attacks/faults on multiple sensors. In fact, GOS employs m observers to identify a single misbehaving sensor in constant time. Compared to that, our scheme uses the same amount of observers to detect all misbehaving nodes in approximately $O(log_\alpha m)$. In our implementation, we set the value of $\alpha = 2$ and the value of α can be changed depending on the specific systems. A larger α combined with parallel processing may actually further speed up our attacks/faults isolation process significantly. Without loss of generality, we assume the value of $\alpha = 2$ in the following sections.

5.2 System Model

Sensors such as PMUs that measure the current phase and amplitude [39] are used in the power system to measure the system state at various locations and time to ensure a smooth operation of the power system. The measurements obtained from these sensors are reported to the central controller via the wired/wireless communication infrastructure. As stated in [13], these measurements include bus voltage, angles and magnitudes. An attack or fault in the power system is always reflected in the form of change in either voltage, current or phase [34].

To model the power system, we use the DC power flow model.

$$P_{ij} = \frac{V_i V_j}{X_{ij}} Sin(\theta_i - \theta_j) \tag{5.1}$$

where V_i is the voltage magnitude and θ_i is the phase angle in bus i, X_{ij} is the reactance of transmission line between bus i and bus j. Since we are using DC power flow model, we

assume the amplitude of voltage in the buses to be close to unity and the phase difference between voltage in two buses to be small. Hence Equation (5.1) can further be simplified to obtain the following equation.

$$P_{ij} = \frac{\theta_i - \theta_j}{X_{ij}} \tag{5.2}$$

Now, the state-estimation objective is to estimate n phase angles θ_i via m real-time measurements. In power flow studies [50], only $n-1$ angles need to be estimated since the voltage phase angle (θ_i) of the reference bus is fixed and known. The state vector, θ, is defined as $\theta = [\theta_1, ..., \theta_n]^T$.

Similarly, the observation vector can be described as Equation (5.3).

$$z = P(\theta) + e \tag{5.3}$$

where $z = [z_1, ..., z_m]^T$ is the vector of measured active power in transmission lines, $P(\theta)$ is the non-linear relation between measurement vector z and state vector θ and $e = [e_1, ..., e_m]^T$ is the Gaussian measurement error vector with covariance matrix σ_e

The control center computes vector z from m active power measurement data. These measurements can either be the injected active power to bus i ($\sigma_j P_{ij}$) or the transmitted active power from bus i to bus j (P_{ij}).

Defining the Jacobian matrix $H \in R^m$ as

$$H = \frac{\delta P(\theta)}{\delta \theta}|_{\theta=0} \tag{5.4}$$

If the phase differences $(\theta_i - \theta_j)$ in Equation (5.3) is small, then the linear approximation model of Equation (5.4) can be obtained as:

$$z = H\theta + e \tag{5.5}$$

Further, the following equation for the system can be established to describe the states in

time series.

$$\theta_{t+1} = \theta_t + \omega_t \tag{5.6}$$

where θ_{t+1} is the state vector θ at time $t+1$.

5.3 Kalman Filter

In this section, we introduce the Kalman Filter (KF) [22] technique to obtain estimates for the state space vector x_t described in the above section. To apply the Kalman Filter technique, we use x_t to represent θ_t and the state equation can be written as,

$$x_{t+1} = Ax_t + w_t \tag{5.7}$$

where, $x_{t+1} = \theta_{t+1}$ and A is the unity matrix. The observation equation for Kalman Filter from Equation (5.5) can be written as:

$$y_t = C_t x_t + v_t \tag{5.8}$$

Here, y_t is the measurement vector z, $C_t = H_t$ and v_t is the measurement noise which is independent of the initial conditions and process noise. Both w_t and v_t are assumed to be white Gaussian noise with zero mean and standard deviation σ.

Kalman Filter can then be applied to compute state estimations $\hat{x}_t$.

5.4 Attacks/Failures Detector and Identifier

After the KF estimator calculates the next state of the system and the sensor readings are available, the detector compares the estimates and the actual sensor readings to detect any disagreement.

5.4.1 χ^2-detector

The χ^2-detector is a conventional detector used with Kalman Filter. As described in [38], the χ^2-detector detector constructs a χ^2 test statistics from the Kalman Filter and compares them with a pre-computed threshold. Now, let us define the residue r_{k+1} at time $k+1$ as:

$$r_{t+1} \stackrel{\Delta}{=} y_{t+1} - \hat{y}_{t+1|t} \tag{5.9}$$

From Equation (5.8) and (5.9), we get,

$$r_{t+1} \stackrel{\Delta}{=} y_{t+1} - C(A\hat{x}_t) \tag{5.10}$$

Then, the χ^2-detector test consists of comparing the scalar test statistics given by:

$$g(t) = r_t^T B_t r_t \tag{5.11}$$

where B_t is the covariance matrix of r_t. The χ^2 detector compares $g(t)$ with a pre-computed threshold obtained using the χ^2-detector-table [38] to identify a failure or attack. χ^2 test is long-term test because, at each detection step, all integrated effects since system starting time are considered. This property makes it very useful for the fault detection in the smart grid, which consists of sensors that are subject to soft failures like instrument bias shift. Another advantage of χ^2 detector is its computational straightforwardness. The parameters required to perform the test are already generated by the Kalman Filter, making it compatible with the KF. Furthermore, the threshold for the detector can be conveniently obtained from the χ^2-table.

5.4.2 Filter Bank for Attacks/Faults Isolation

Generalized Observer Scheme (GOS) To handle the problem of attacks and faults isolation, filter banks can be used to estimate the dynamic behaviors of the system [48]. In

this chapter one of the filter banks, Generalized Observer Scheme (GOS) [46], is implemented. The filter bank consists of many observers. Each observer is supplied with different subsets of output and all of the input of the system.

Assume that the power grid system consists of m observers. The GOS is designed such that the i^{th} observer is driven by the input u and all of the m outputs except y_i. As described in Figure 5.1 the i^{th} observer is invariant to the fault of the measurement of y_i. Therefore, the residual vector r_i depends on all but the i^{th} sensor reading. Each residual vector r_i is composed of m elements $r_i = [r_{i1}, r_{i2}, ..., r_{im}]$ where $r_{ii} = 0$. These residual vectors are fed to the decision unit (the χ^2-detector in our implementation) which detects and isolates attacks or faults that are identified.

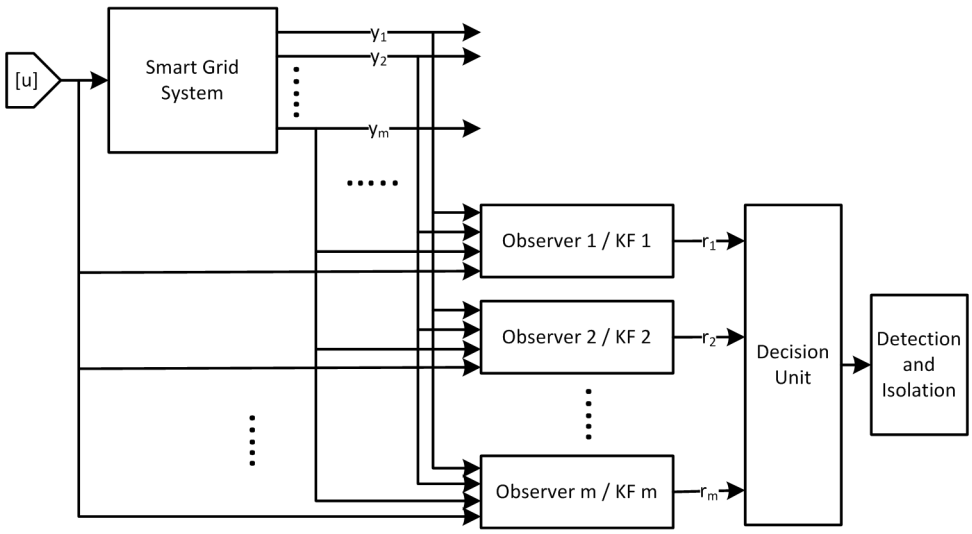

Figure (5.1) GOS using Kalman Filter

Iterative Observer Scheme (IOS) As to be shown in Section V, the GOS scheme can only detect attacks/faults on one sensor at a time. Hence, we present an Iterative Observer Scheme (IOS) to detect and isolate multiple attacked nodes. In step 1, we start with a single Kalman Filter and χ^2-detector combination to check for an attack. This takes all the m outputs into account. If an attack is found, we proceed to step 2. In this step,

we divide the m outputs into two subgroups and use them to drive two observers. The first observer is driven by the input u and first half of the m outputs. Similarly, the second observer is driven by the input u and second half of the m outputs. Hence, we check for attacks in outputs 1 to $\lceil m/2 \rceil$ and $\lceil m/2 \rceil + 1$ to m separately. This process is repeated iteratively for each half of the subset whenever the attack/fault is detected. The iteration terminates when the size of the subset becomes less than or equal to user defined cut off parameter λ such that $\lambda \geq 1$.

In this chapter, for the simplicity, every time an attack/fault is encountered, the size of the subset is divided by half and the detection operation is performed on each half repeatedly. In the absence of attacks/faults, we discontinue the isolation process on that subset. In our implementation, the algorithm terminates when $\lambda = 2$. It can be seen that for m measurements, maximum of $2m-1$ computations are required and there are at most $log_2(m)$ steps. However, it is worth noting that the division of a subset into smaller subsets can be carried out differently depending on the specific scenarios such as the physical distribution of the sensors. Generally, if one divides the set into α subsets in each iteration, the maximum steps will be $O(log_\alpha m)$. Clearly, parallel processing techniques can be employed to speed up for quick real-time isolation.

5.5 Implementation and Performance Evaluation

The Kalman Filter estimator, the χ^2-detector and observers are implemented using Matlab. MATPOWER package was used for solving the IEEE 9-bus and IEEE 14-bus systems [43]. Particularly MATPOWER was used to obtain matrix H which is used in the state estimation. Matlab function randn() was used to produce normally distributed noise with mean value zero. The threshold was obtained from the χ^2 table.

5.5.1 Fault detection and isolation using GOS

Figure 5.3 shows the detection of an attack on a single bus (Bus 5) using GOS scheme in IEEE 9-bus system. As it can be seen in the figure, since Observer 5 is not affected by

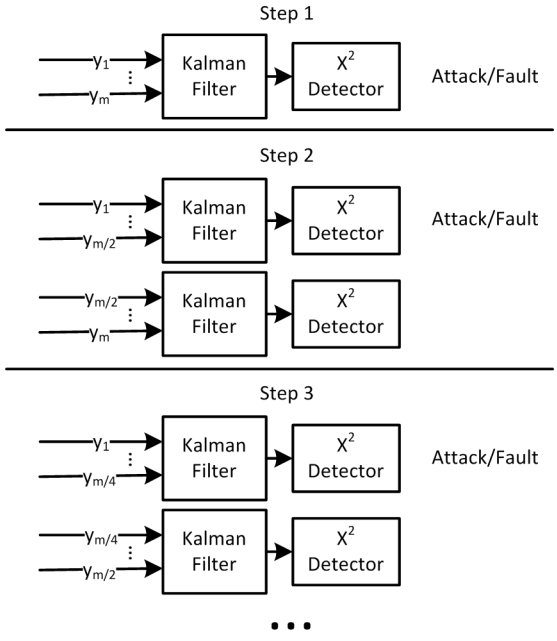

Figure (5.2) Iterative observer scheme using Kalman Filter showing first three iterations.

measurement y_5, the residue function $g(t)$ does not exceed the threshold denoted by the straight line. However, the remaining eight observers are affected by the attack. Hence, GOS shows that there is an attack on Bus 5.

In Figure 5.4, two attacks occur at the IEEE 9-bus system. It can be seen that the residue function $g(t)$ exceeds the threshold in all the GOS observers. Hence, GOS is unable to detect simultaneous attacks/faults in the system.

5.5.2 Fault detection and isolation using IOS

Figure 5.5 shows the result of the Iterative Observer Scheme (IOS) in IEEE 9-bus system. The graph on the top row shows the results after applying KF and χ^2-detector on the entire system. Since the residue function $g(t)$ exceeds the threshold, an attack is detected in the Step 1. Following this, the scheme divides the 9 outputs into two halves: $1-5$ and $6-9$ in Step 2. These two subsets of measurements are fed into two separate observers. The residue functions obtained from the observers are plotted in the graphs in the second row

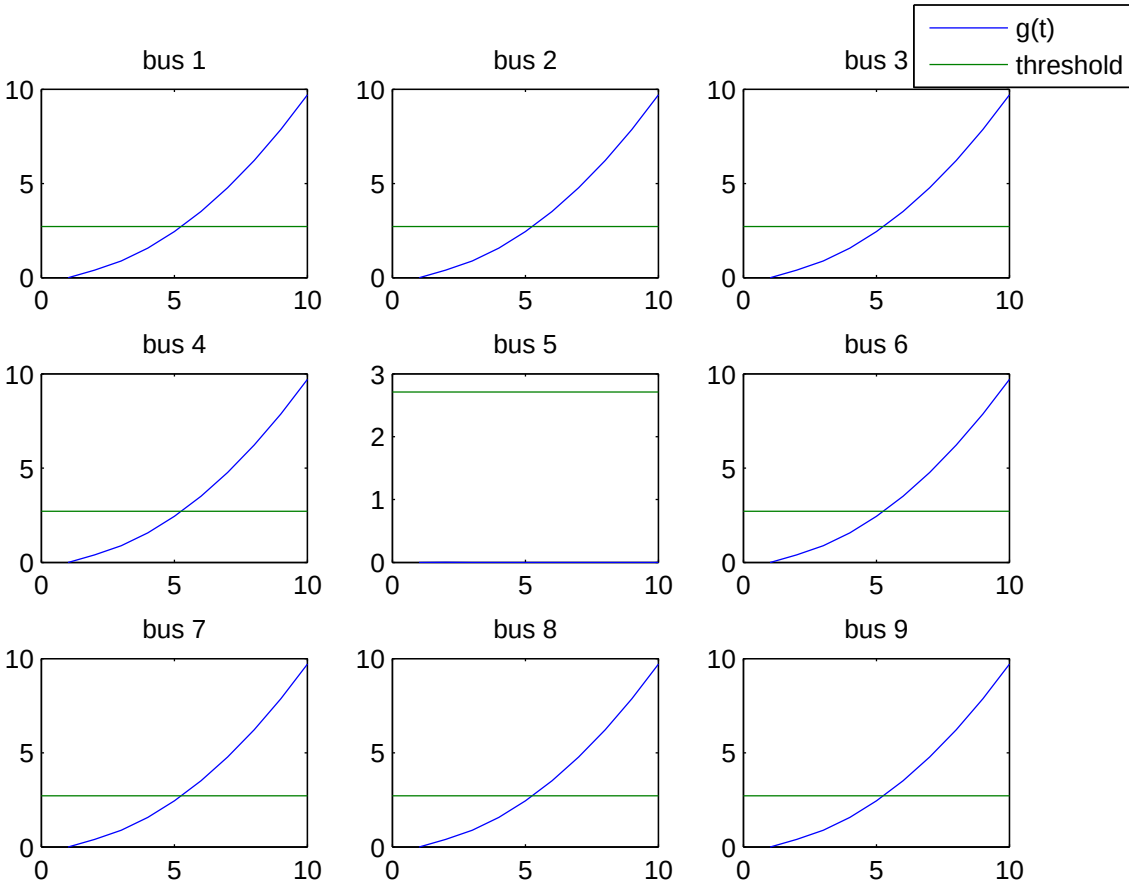

Figure (5.3) Simulation result showing detection of attack on bus 5 using GOS in IEEE 9-bus system.

of this figure. Again, it can be seen that there are attacks on both of these subsets. As a result, the scheme splits these subsets again into two halves each, thus creating four subsets: $1-3$, $4-5$, $6-7$ and $8-9$. When these subsets are fed to the observers, we obtain the residue functions as shown in the last row of the figure. It can be seen that $g(t)$ of the first and the last figure in this row (buses $1-3$ and buses $8-9$) exceed the threshold. This shows that there are attacks on these sets of buses. Similarly, the simulation graphs resulting from applying IOS on IEEE 14-bus is shown in Figure 5.6. In this figure, buses 3, 8 and 11 are under attack.

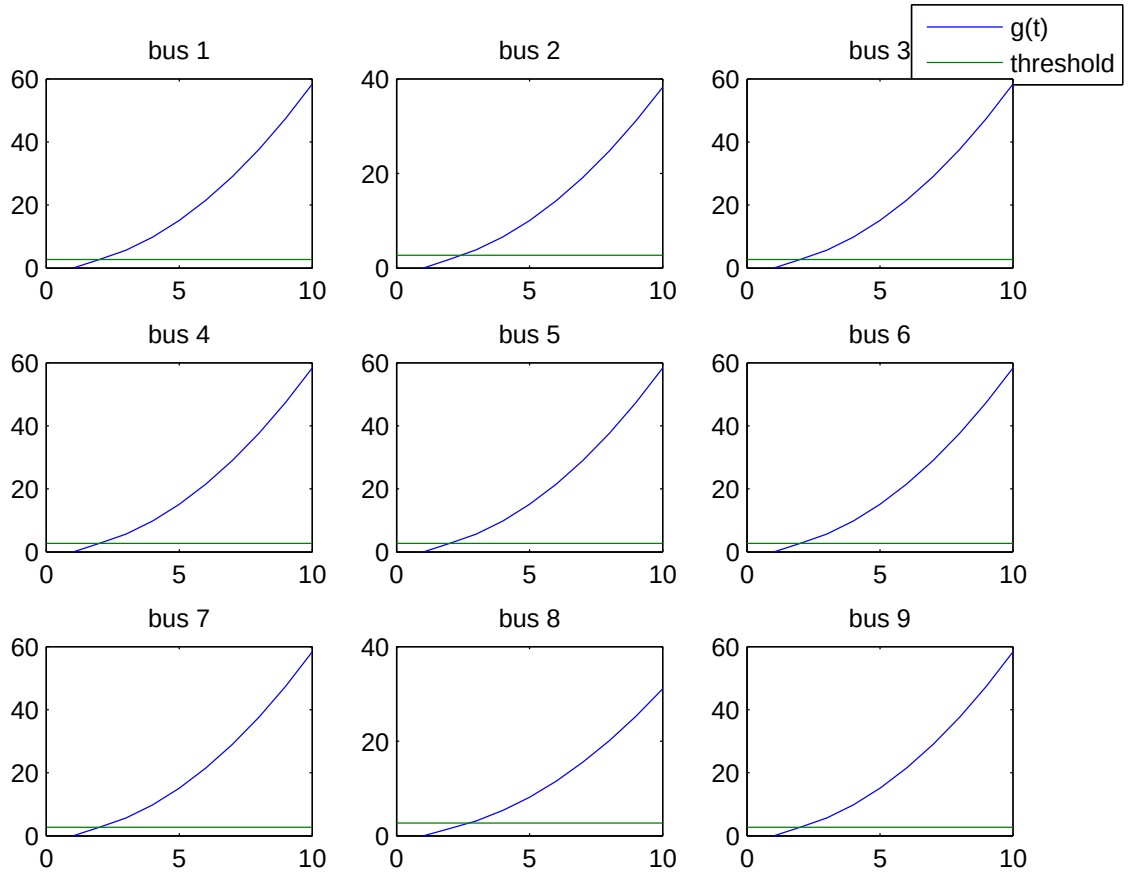

Figure (5.4) Simulation results showing failure in detection of attack using GOS in IEEE 9-bus system when there were attacks on two separate buses- Bus 5 and 8.

5.5.3 Comparison of GOS and IOS

	GOS	IOS
Detects multiple simultaneous attacks	No	Yes
Number of computations in worst case	m	$2m-1$
Number of computations in best case	m	$2\log m-1$

Table (5.1) Comparison of GOS and IOS.

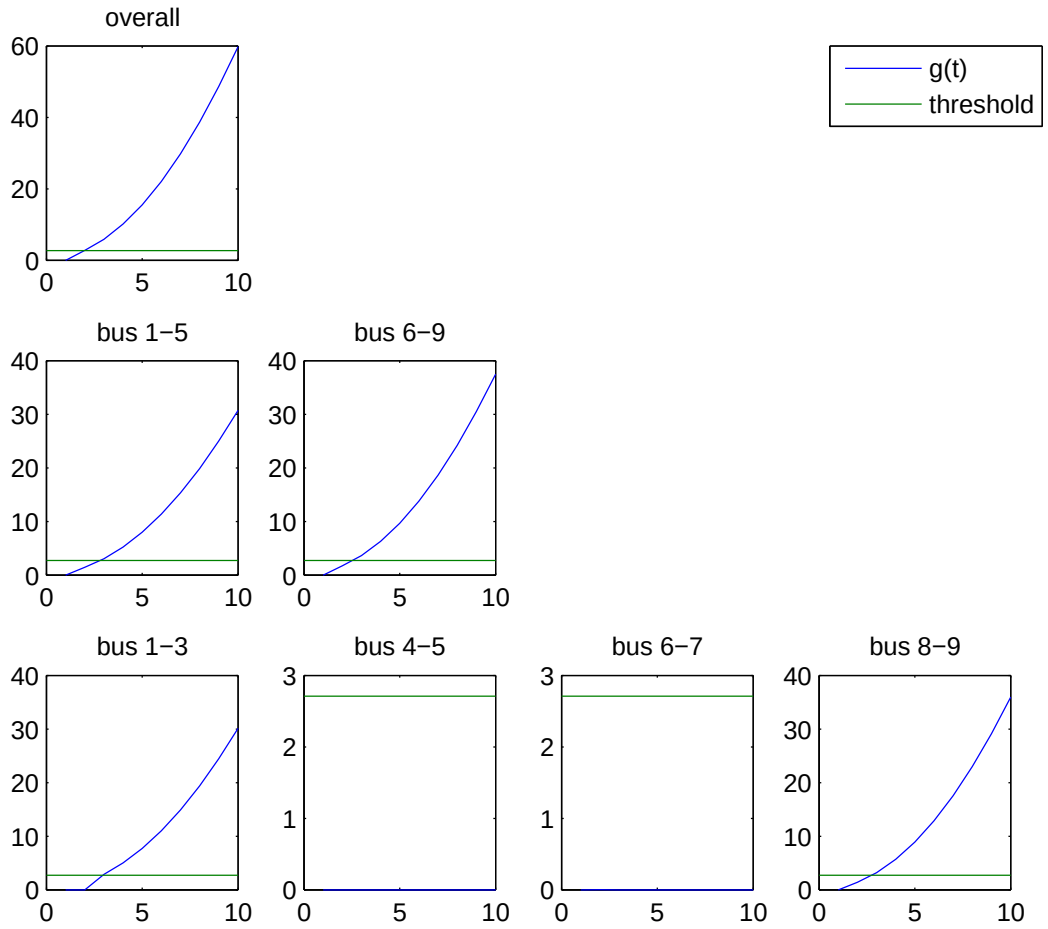

Figure (5.5) Simulation results showing attack on bus 2 and bus 8 for IEEE 9-bus using IOS.

Table 5.1 compares the GOS and IOS schemes based the number of computations required for the best case and the worst case scenarios. In case of GOS, since it only detects a single attack at a time, the best case and the worst case are the same. GOS uses m observers and assuming each observer does unit computation, the total number of computations required is m for all cases. For IOS, in worst case all the buses are under attack. Hence all of the subsets are checked for attacks in each iteration resulting in total $2m - 1$ computations. In the best case, only a single bus is attacked. In this scenario, only two subsets are checked

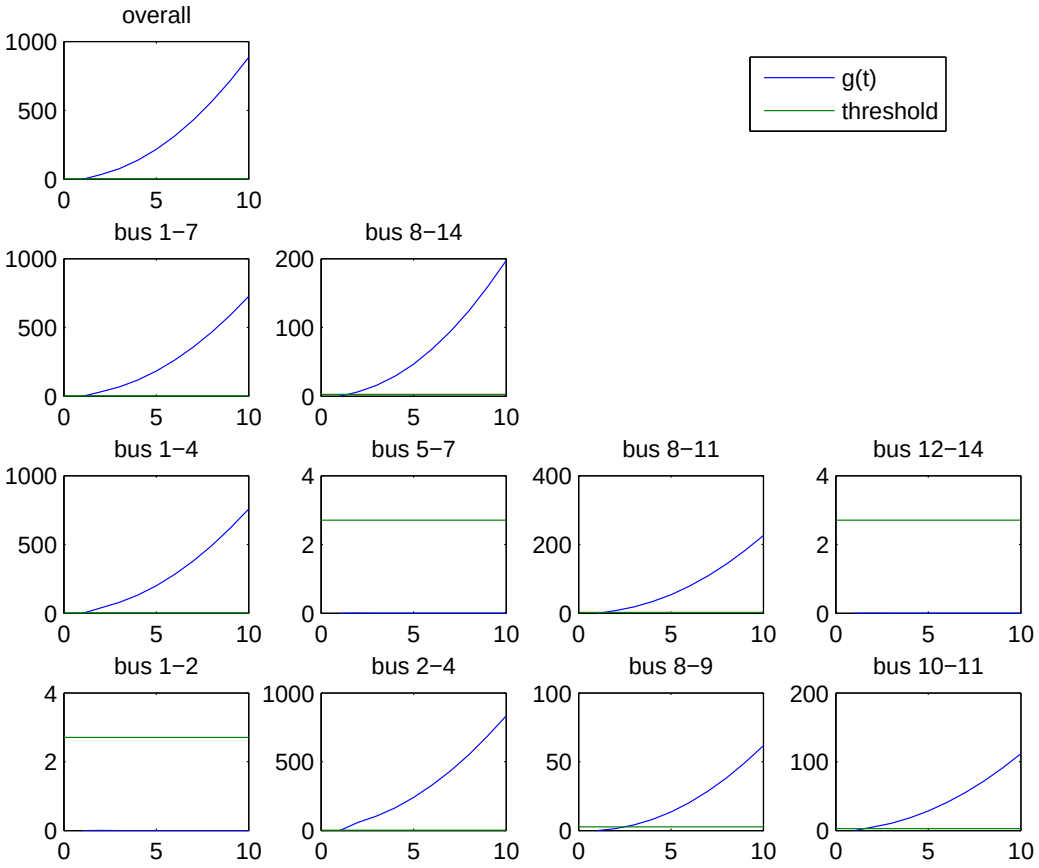

Figure (5.6) Simulation results showing attack on bus 3, 8 and 11 for IEEE 14-bus using IOS.

for attacks in each iteration except the first. Thus the total number of computations required is $2\log m - 1$.

PART 6

LOAD SCHEDULING IN SMART GRID

In this chapter, we present our study on load scheduling for a building in smart grid. Scheduling is important as it affects the overall energy signatures, which in turn, relates to malicious activities in smart grid. The scheduling algorithm proposed in this chapter maximizes the user comfort given the power and cost budgets.

6.1 Load Scheduling overview

Electricity, has become one of the most valuable commodities due to the increasing number of electrical appliances embedded in our daily lives. Given the energy challenges in the world, the need for optimizing energy consumption by these appliances, has grown. This need, not only relates to the commercial industries but also extends to the residential sector [51]. The advent of smart grids enables real-time energy consumption reading, load shifting and energy prices, which also promotes the consumers to optimize/manage the operation schedule of these appliances to reduce the electricity cost.

Different aspects of Home Energy Management (HEM) systems have been studied in the literature. Most of these studies focus on algorithms to effectively schedule the usage of these household appliances to minimize the electricity cost. For example, in [52] the collective impact of several different HEMs based on sharing appliances is studied. The study in [53] investigates the issue where all consumers might operate the appliances during the lowest electricity price. In [54], the authors demonstrate a HEM scheme with a smart thermostat control which allows the consumers to adjust its usage during the demand response (DR) periods. Similar work is presented in [55] where the authors have used sensor networks and taken other factors like lighting into consideration. In [56], the authors present their analysis on how the implementation of Smart HEMs could contribute to energy savings

for the residents. Optimizing HEM by integrating additional diverse energy sources like photovoltaic (PV), thermal energy and fuel cells using dynamic programming have been studied in [57].

In addition to optimizing the energy cost, the relation between user comfort and energy consumption is recently drawing much interest. Authors in [58] present three algorithms to reduce the peak loads of the home electronic appliances and rank these algorithms based on the inconvenience caused to the consumers. Their studies show that the different appliances are best scheduled with different algorithms. Authors in [59] extend this idea by allowing consumers to set different weights to different tasks. Their study on correlating the consumer inconvenience with the potential savings in the electricity cost shows that the savings potential decreases as more comfort is sacrificed. The energy saving versus the user requirements are translated into optimization of a constrained boolean satisfiability problem in [60], where the user specifications are converted into boolean expressions. A three layer architecture has been proposed in [61] to satisfy the maximum available electrical power constraint and to maximize user satisfaction criteria. The Linear Programming (LP) model developed in [62] maximizes user comfort by minimizing temperature difference between desired and indoor temperature. However, none of the above works consider the interruption of loads and possible blocking the appliances' energy request for a household HEM or building office.

In this chapter, we present a Linear Programming (LP) model, to maximize the user comfort in a residential HEM systems as well as a small office building via scheduling. Our model for HEM assumes that loads like washer/dryer and EV are deferrable as well as interruptible. Similarly, for office buildings, loads such as batch printing/photocopying are considered deferrable and interruptible. An iterative greedy algorithm (CPG) is also proposed.

The rest of the chapter is organized as follows. In Section 6.2, we classify the different appliances into different categories. Section 6.3 provides the problem definition. In Section 6.4, the linear programming model is described. The Comfort Prioritizing Greedy (CPG) algorithm is introduced in Section 6.5. In Section 6.6, a bin-packing algorithm is discussed.

Section 6.7, the simulation results for both HEM and small office are shown.

6.2 Appliance Classification

In this chapter we classify the appliances in a building (house or office) into different categories based on various attributes of the appliances. These categories are described in the list below.

- *Non deferrable and non interruptible load*: includes appliances with a fixed start time that cannot be changed and should not be interrupted; for example, refrigerator, lighting, servers.

- *Deferrable and interruptible load*: includes appliances with a flexible start time and can be interrupted. The only constraint is that the operation should start and end within a given start and end time. We consider washer/dryer and dishwasher to be interruptible load as it is possible to interrupt them at any given time and start it again at a later time. In case of small office building, batch printing/photocopying jobs are considered to be deferrable and interruptible.

- *Deferrable and non-interruptible load*: includes appliances that has a flexible start time as long as it is within a given time frame and should not be interrupted. For interruptible appliances, a minimum gap between interruptions can be set to disallow frequent interruptions.

- Non deferrable and interruptible load: includes appliances that has a fixed start time and can be interrupted.

6.3 Problem Definition

In this work, we model the load scheduling problem as an optimization problem. Given the power and budget constraints, indoor temperature, outdoor temperature, desired temperature, the amount of power required to maintain the desired temperature and various

other loads that require power allocation, our goal is to maximize the user comfort at any given time t. It is assumed that the user is most comfortable when the room temperature is close to the desired temperature. Hence, if $\delta = |x_t^{in} - desiredtemp|$, the user is most comfortable when $\delta \simeq 0$ and less comfortable with the increasing values of δ. In other words, the problem can be formulated as a LP problem whose goal is to minimize the temperature difference between the indoor and desired temperature at time t by allocating power to HVAC.

6.4 LP model

In this section, we present the LP model for the problem described above. We assume the power required by HVAC to obtain the desired temperature is calculated in advance and is available for the HEM to use. As described in [62], the power required by HVAC to reach the desired temperature when outdoor and indoor temperatures are known can be obtained using Equation (6.1). Hence, we assume the power required to maintain the indoor temperature at x_t^{in}, when the outdoor temperature is x_t^{out}, to be formulated as Equation (5). The value of ϵ_t for each hour is known in advance.

$$x_{t+1}^{in} = x_t^{in} + \alpha(x_t^{out} - x_t^{in} + GP_t^{HVAC} + c + \omega(t)) \tag{6.1}$$

where, α, G and c are the parameters estimated using historical data, x_{t+1}^{in} is the indoor temperature at time $t+1$, x_t^{in} is the indoor temperature at time t, x_t^{out} is the outdoor temperature at time t and P_t^{HVAC} is the power consumed by HVAC to achieve the desired temperature.

To model the power consumption by Electric Vehicle, HVAC, and deferrable and interruptible loads (washer, dryer and dishwasher), we define the variables in Table 6.1. It is assumed that maximum power consumption for each load type P_{max}^{HVAC}, P_{max}^{VE}, P_{max}^{INTR} is known. Then the objective function of our proposed linear programming model is to minimize the temperature difference i.e. to minimize $|x_t^{in} - desiredtemp|$. This absolute formula

can be replaced with linear equations in Eq.(2), (3) and (4).

$$\text{minimize} \quad \sum_{0}^{T_{n-1}} X'_t \tag{6.2}$$

$$\text{subject to} \quad x_t^{in} - desiredtemp < X'_t \tag{6.3}$$

$$-(x_t^{in} - desiredtemp) < X'_t \tag{6.4}$$

$$P_t^{HVAC} = \epsilon_t(x_t^{out} - x_t^{in}) \tag{6.5}$$

$$0 \leq P_t^{HVAC} \leq P_{max}^{HVAC}. \tag{6.6}$$

$$\sum_{T_s^{VE}}^{T_e^{VE}} P_t^{VE} = Q \tag{6.7}$$

$$0 \leq P_t^{VE} \leq P_{max}^{VE}. \tag{6.8}$$

$$\sum_{T_s^{INTR_i}}^{T_e^{INTR_i}} P_t^{INTR_i} = R_i \tag{6.9}$$

$$0 \leq P_t^{INTR_i} \leq P_{max}^{INTR_i}. \tag{6.10}$$

$$\sum_{0}^{T_{n-1}} z_t(P_t^{HVAC} + P_t^{VE} + P_t^{INTR_i}) \leq B \tag{6.11}$$

$$0 \leq P_t^{HVAC} + P_t^{VE} + P_t^{INTR_i} \leq P_t^{max} \tag{6.12}$$

6.5 Comfort Prioritizing Greedy (CPG) Algorithm

This section presents an iterative greedy algorithm to solve the optimization problem described in Section 6.3. We begin by defining a threshold temperature which is different from the desired temperature, and is acceptable indoor temperature for the user. For instance, if the desire temperature is $74°F$ then $76°F$ might be a reasonable acceptable temperature. We use threshold temperature to ensure devices other than HVAC will not starve. Since we are trying to optimize the user comfort, our first priority is to assign power to HVAC such

Algorithm 1 The Comfort Prioritizing Greedy (CPG) Algorithm

sort the list of time slots according to the price profile available
define a threshold temperature which the user would like to achieve in say γ
Total power assigned for vehicle charging $P_{total}^{VE} = 0$
Total power assigned to the interruptible load$P_{total}^{INTR} = 0$
Total budget spent $cost = 0$;
for Each timeslot t, starting from the least cost time slot **do**
    if $cost < B$ **then**
        assign power to P_t^{HVAC} which is required to achieve the threshold temperature γ
        update the total power available $P_t^{available} = P_t^{max} - P_t^{HVAC}$
        $cost+ = z_t * P_t^{HVAC}$
        $x_t^{in} = \gamma$
    end if
end for
for Each timeslot t, starting from the least cost time slot **do**
    if $T_s^{VE} \leq t \leq T_e^{VE}$ AND $P_{total}^{VE} < Q$ AND $P_{available} > 0$ AND $cost < B$ **then**
        if $P_t^{available} >= P_{max}^{VE}$ **then**
            $P_t^{VE} = P_{max}^{VE}$
            $P_t^{available} += P_{max}^{VE}$
            $cost+ = z_t * P_{max}^{VE}$
        end if
    end if
end for
for Each timeslot t, starting from the least cost timeslot **do**
    if $T_s^{INTR} \leq t \leq T_e^{INTR}$ AND $P_{total}^{INTR} < R$ AND $P_{available} > 0$ AND $cost < B$ **then**
        if $P_t^{available} >= P_{max}^{INTR}$ **then**
            $P_t^{INTR} = P_{max}^{INTR}$
            $P_t^{available} += P_{max}^{INTR}$
            $cost+ = z_t * P_{max}^{VE}$
        end if
    end if
end for
for Each timeslot t, starting from the least cost timeslot **do**
    if $P_t^{available} > 0$ AND $cost < B$ **then**
        if power is adequate and budget is available **then**
            Adjust the temperature to *desiredtemp*
        else
            Adjust the temperature to reach closest to the desired temperature
        end if
    end if
end for

Table (6.1) Variable list

Variable	Description
x_t^{in}	Indoor temperature
x_t^{out}	Outdoor temperature
$desiredtemp$	Desired temperature
z_t	Hourly electricity price
P_t^{HVAC}	Power consumed by HVAC
P_t^{VE}	Power consumed by electric vehicle
$P_t^{INTR_i}$	Power consumed by i^{th} interruptible load
P_{max}^{HVAC}	Maximum power consumption of HVAC
P_{max}^{VE}	Maximum power consumption of electric vehicle
$P_{max}^{INTR_i}$	Maximum power consumption of i^{th} interruptible load
Q	Desired charging amount of electric vehicle
R	Total power required by the interruptible load
P_t^{max}	Total power consumption within hour t
B	Total budget for the day
T_s^{VE}	Starting time for charging the vehicle
T_e^{VE}	Ending time for charging the vehicle
$T_s^{INTR_i}$	Starting time of i^{th} interruptible and deferrable load
$T_e^{INTR_i}$	Ending time of i^{th} interruptible and deferrable load

that indoor temperature can achieve the threshold temperature, in case of budget or power shortage.

In the proposed CPG algorithm as shown in Algorithm 1 and the flowchart in Figure 6.1, the time slots are sorted in an ascending order of the hourly electricity price (z_t) (Line 1 in Algorithm 1). Lines 2-6 are used to initialize required variables. Then, the following four iterations are carried out.

- *First Iteration* (Lines 7-13 in Algorithm 1): starting from the time slot with the lowest price, power is assigned to HVAC to achieve the given threshold temperature.

- *Second iteration* (Lines 14-22): starting from the time slot with the lowest price, power is assigned to charge the electric vehicle without exceeding the maximum power constraint and the budget constraint.

- *Third iteration* (Lines 23-31): starting from the time slot with the lowest price, power

is assigned to the remaining deferrable and interruptible loads without exceeding the maximum power and budget constraint. If at any point power cannot be assigned in these iterations because the budget constraint is exceeded then the threshold temperature is changed (increased in summer and decreased in winter) by $1°F$ and the algorithm is executed again from the beginning. Similarly, if any of the appliances cannot be scheduled because the power constraint is exceeded then the threshold temperature is changed and the algorithm is executed again.

- *Fourth iteration* (Lines 35-40): if power and budget is available then, the algorithm goes back and assigns power to HVAC so that the desired temperature can be obtained. The algorithm stops after completion of the last iteration or if the budget constraint is exceeded before the last iteration is completed.

6.6 Bin-packing Algorithm

In this section, we apply the standard technique of bin-packing to our load scheduling problem. Here, we consider the maximum power available at each hour to be the bin and try to pack as many appliances as possible in each of these bins without exceeding the budget constraint. For each time slot, if budget is available, assign the power required to achieve desired temperature to the HVAC. If there is space remaining in the bucket, i.e., the total power limit for that hour is not exceeded then assign power to the electric vehicle. Next, if there is still space then we assign power to the deferrable and interruptible loads for that time slot. The above power assignment is repeated for each time slot. The detailed steps are listed in Algorithm 2. Note, there can be appliances that cannot be scheduled while using this algorithm if the budget constraint is exceeded.

Lines(1-3) in the Algorithm 2 are used to initialize the required variables. Then, for each time slot, lines (5-7) are used to assign power to HVAC, and update available power and budget. Lines (8-14) assign power to the electric vehicle, and update available power and budget. If maximum power constraint is not exceeded for that time slot and budget is

available then lines (15-21) assign power to the interruptible loads.

Algorithm 2 The Bin-packing algorithm

1: Total power assigned for vehicle charging $P_{total}^{VE} = 0$
2: Total power assigned to the interruptible load$P_{total}^{INTR} = 0$
3: Total money available $cost = B$
4: For Each timeslot t if $cost > 0$,
5: assign power to P_t^{HVAC} which is required to achieve the desired temperature $desiredtemp$
6: update the total power available $P_t^{available} = P_t^{max} - P_t^{HVAC}$
7: total budget available $cost- = P_t^{HVAC} * z_t$
8: **if** $T_s^{VE} \leq t \leq T_e^{VE}$ AND $P_{total}^{VE} < Q$ AND $P_{available} > 0$ And $cost > 0$ **then**
9: **if** $P_t^{available} >= P_{max}^{VE}$ **then**
10: $P_t^{VE} = P_{max}^{VE}$
11: $P_t^{available} += P_{max}^{VE}$
12: $cost- = P_t^{VE} * z_t$
13: **end if**
14: **end if**
15: **if** $T_s^{INTR} \leq t \leq T_e^{INTR}$ AND $P_{total}^{INTR} < R$ AND $P_{available} > 0$ AND $cost > 0$ **then**
16: **if** $P_t^{available} >= P_{max}^{INTR}$ **then**
17: $P_t^{INTR} = P_{max}^{INTR}$
18: $P_t^{available} += P_{max}^{INTR}$
19: $cost- = P_t^{INTR} * z_t$
20: **end if**
21: **end if**

6.7 Simulation results

The simulation is performed using LPSolve for solving the LP formulations 6.3. The proposed CPG algorithm and the bin-packing algorithm are implemented using Java. The hourly price chart and outdoor temperature for 24 hours is shown in Figure 6.2 and Figure 6.3 respectively. We simulate a typical household with HVAC, and four appliances (Electric vehicle, washing machine, dryer, dishwasher) as well as an office building with HVAC and two deferrable and interruptible loads (batch jobs in photocopy machine and printers). The office appliances such as servers, workstations, projectors, printers, photocopy machine are considered as non deferrable and non interruptible loads. It is assumed that the batch printing and batch photocopying jobs have a start and an end time, and can be put in a

Table (6.2) Appliances and their time and power consumptions

Appliance	Power	Start time	Stop time	Total power
EV	3.3 kw	19:00	7:00	19.8
Washing machine	0.5 kw	8:00	17:00	0.5
Dryer	4 kw	17:00	20:00	4
Dish Washer	1.5	20:00	5:00	3

Table (6.3) The energy schedule for appliances in a household

Algorithm	EV	Washer	Dryer	Dishwasher
LP	1:00-6:00	17:00	18:00	2:00
Greedy	1:00-6:00	9:00	18:00	1:00
Bin-packing	1:00-6:00	9:00	-	1:00

queue.

The power consumption model for the household HEM is described in Table 6.2. This data is obtained from [63], where the maximum power in one hour is 15 KW and the desired temperature is set to $74°F$

When the budget is large and all the constraints are met, all three algorithms are able to accomodate all the appliances and the desired temperature as shown in Figure 6.4.

At the expense of denying any power to Dryer, the bin-packing algorithm does the best job in maintaining indoor temperature close to the desired temperature most of the day. The temperature spikes in Figure 6.5 are due to the rise in the outdoor temperature at that time. One can see that the LP modelling responds to the jump of outside temperature slightly slower while the CPG algorithm outperforms the other two in maintaining a smooth indoor temperature.

Figure 6.6 shows the performance of CPG algorithm when the threshold is $79°F$,$77°F$, $75°F$ respectively. When the threshold for the CPG algorithm is $79°F$ and $77°F$, all the appliances are scheduled. However, when the threshold was set to $75°F$ it can not schedule washer, dryer and dishwasher. Figure 6.6 shows that the performance is better when the

threshold is set to $77°F$ than when it is set to $79°F$. This imples that if the value of the threshold can be set closer to the desired temperature without blocking any of the appliances, then a better performance can be obtained.

The power consumption model for office includes the HVAC and two deferrable loads of batch printing, and batch photocopying. The maximum power in one hour is 20 KW and the desired temperature is set to $74°F$. Table 6.4 lists the power consumption, and start/end times for these jobs when the budget is enough. However, when the budget is limited to 200 cents, the bin-packing algorithm can not schedule the printer while the other two can accommodate all the jobs. Again, as shown in Figure 6.7, the overall performance of the CPG algorithm is slightly better that the other two.

Table (6.4) The energy schedule for appliances in an office

Appliance	Power	Start time	Stop time	Total power
Batch Printing	2.4 KW	18:00	7:00	7.2 KW
Batch Photocopying	1.5 KW	18:00	7:00	3 KW

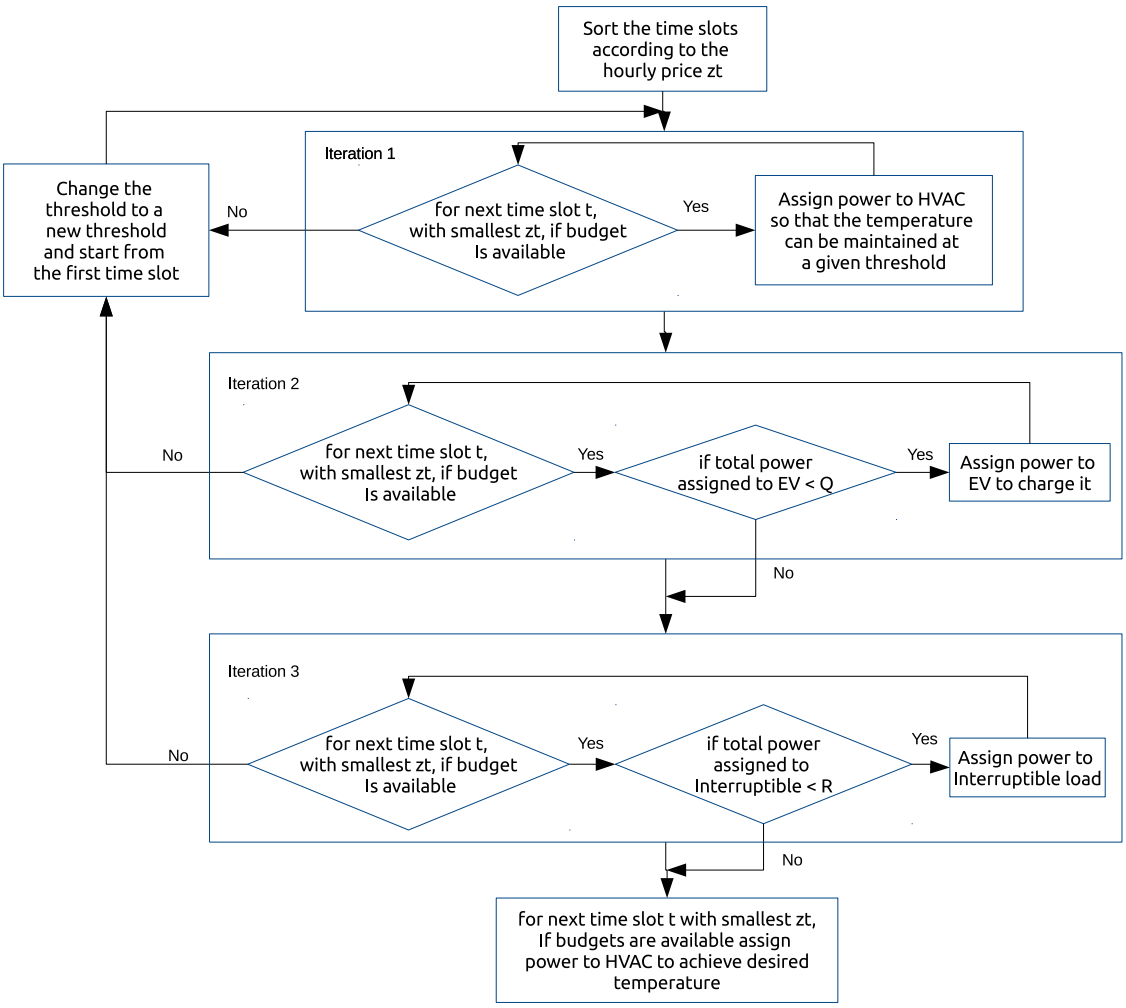

Figure (6.1) The flow chart describing the CPG algorithm

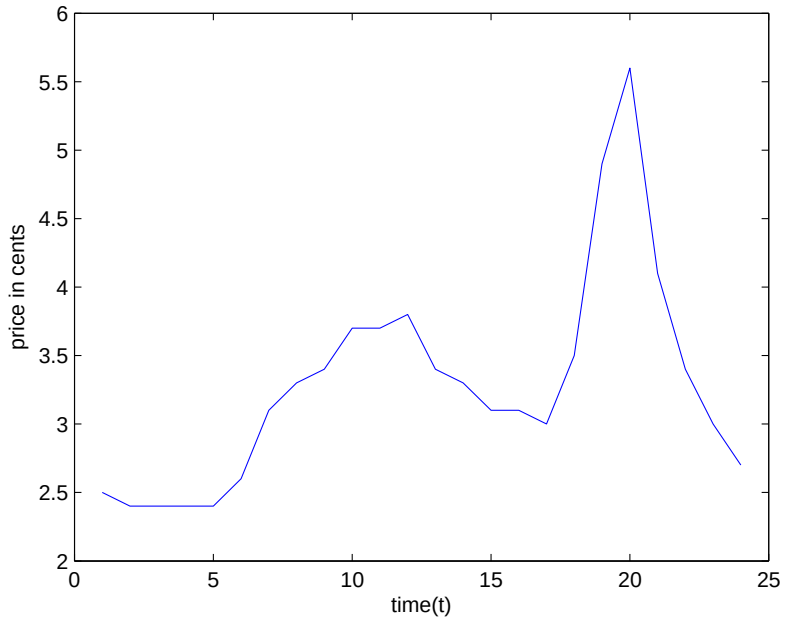

Figure (6.2) Hourly Price per KW power

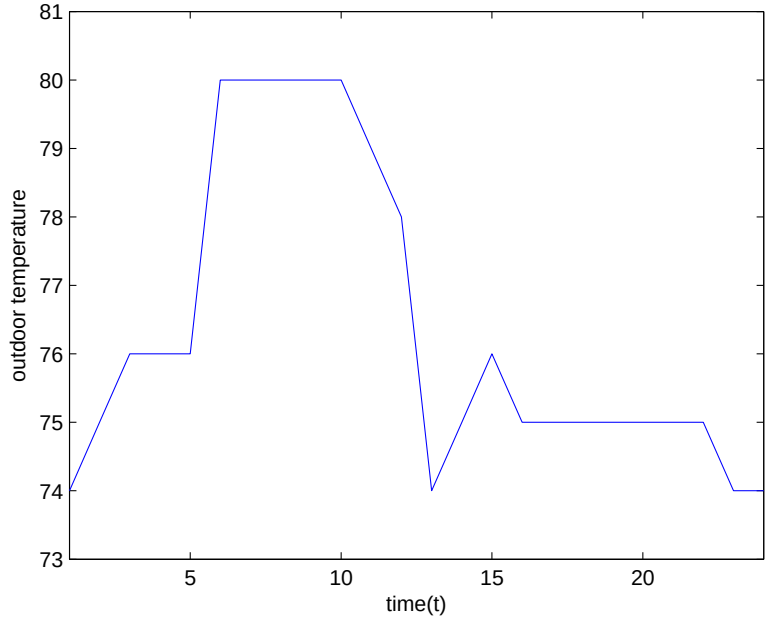

Figure (6.3) Hourly outdoor temperature

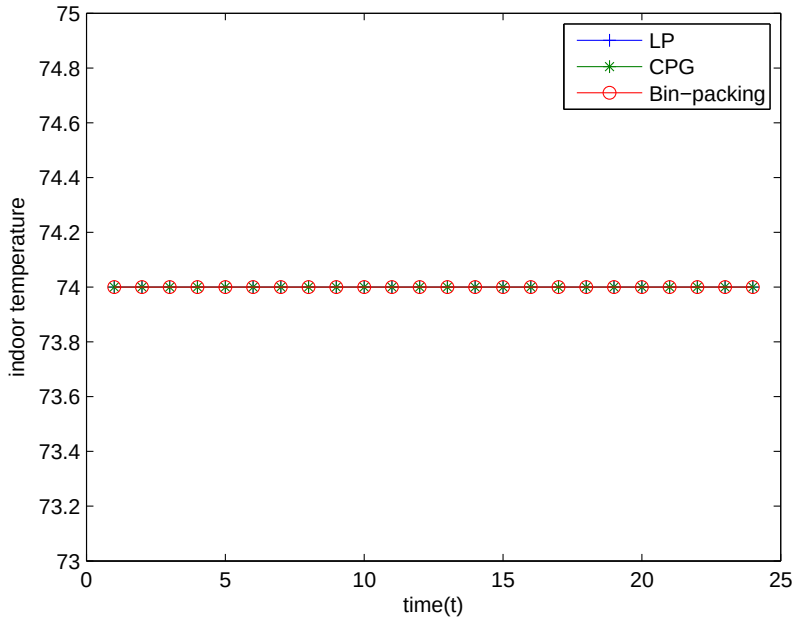

Figure (6.4) All three algorithms were able to achieve the desired temperature

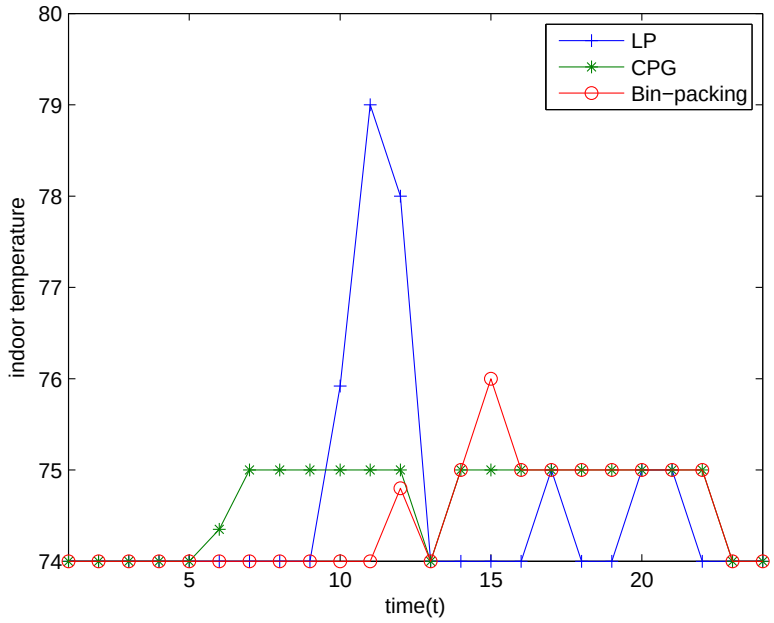

Figure (6.5) All three algorithms were able to achieve the desired temperature

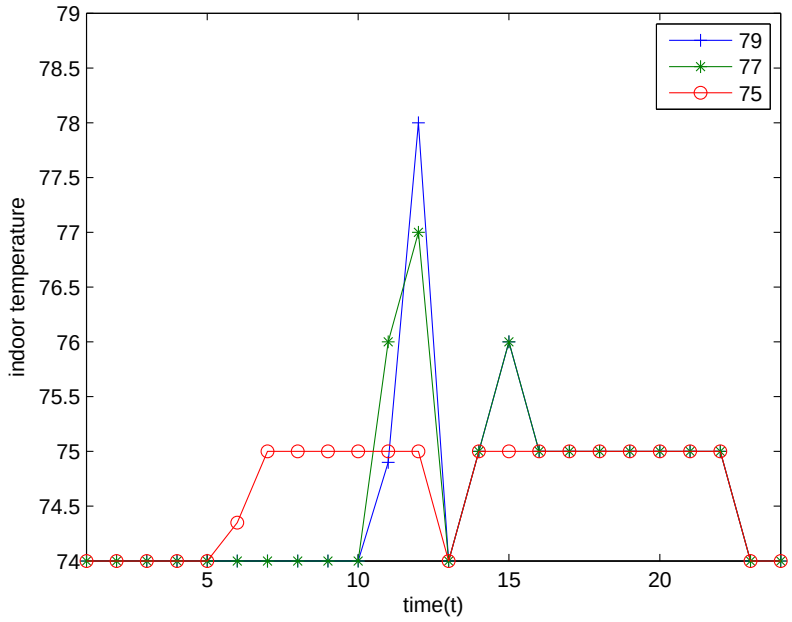

Figure (6.6) Indoor temperatures when threshold in the CPG algorithm is varied

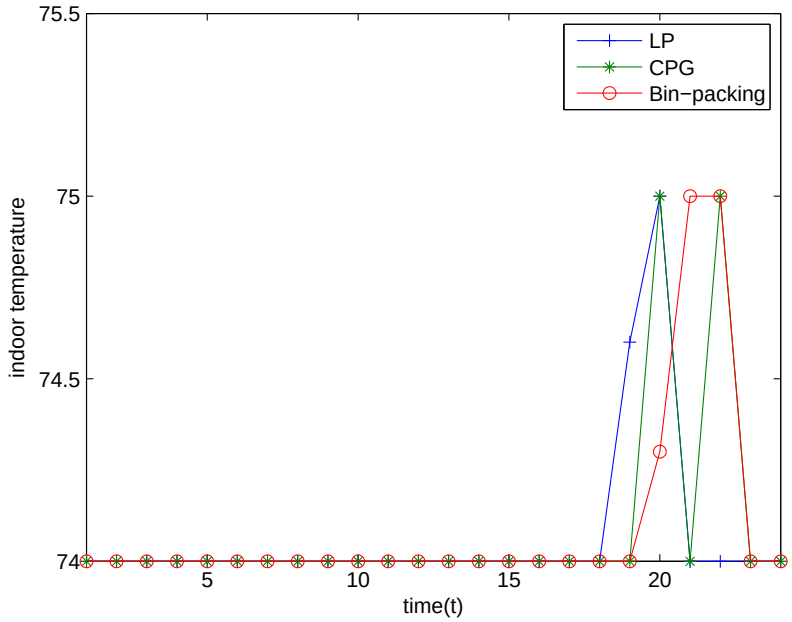

Figure (6.7) Indoor temperatures from three algorithms in a building office

PART 7

TOWARDS THE SECURED CPS: A CASE STUDY ON ATTACKS OF WATER SUPPLY SYSTEM

Water Supply system is another important CPS. In this chapter, we study different types of attacks on WSS and build a framework capable of detecting those attacks.

7.1 Attacks on WSS overview

Water supply system is one of the sensitive areas which can have dramatic public health and economical impacts when attacked. As new types of sensors have been introduced to detect the existing attacks, newer types of attack patterns to thwart this effort has also emerged. Thus in this chapter we look into the existing attack and detection models and present our technique.

The recent literature [64, 65, 66, 67, 68, 69] largely focus on detecting attacks targeted towards the quality of water in the supply system. In this regard, the attack detection can be classified into two groups. The first group deals with the deployment of different types of sensors to detect water quality and the second focuses on strategically placing the sensors in WSS.

Notable works in the first group include a taste sensor called *Electronic Tongue* [64]. As stated in the paper, the sensor is based on voltametric technique and is able to detect small changes in the chemical and bacterial compound in the water. Similarly, an advanced nanomaterial based sensor is presented in [65] for continuous and in-situ monitoring of various organic along with non-organic pollutants in water.

The second group deals with attack detection schemes primarily focusing on the sensor placement problem. The authors in [66] focus on minimizing the number of sensors as well as minimizing the effect of attacks on general public by efficiently positioning sensors in the

WSS. Data mining techniques are applied in [68] to find the optimal location for the sensor placement to determine drinking water quality. The authors of [67] proposed a technique to optimally choose online monitoring points to monitor municipal water supply system for prediction and early detection of contaminants. In both these groups, sensors are considered accurate/reliable and attacks on sensors to manipulate actual sensor readings are not considered. The authors of [69] used theory of switching boundary control of partial differential equations to model deception attacks on water SCADA system. The paper presented a stealthy deception attack that can evade detection by manipulating sensor measurements. However, it does not provide any specific mechanism to detect such an attack on SCADA system.

Though recent studies focus on the detecting attacks on the quality of water using sensors, there has not been adequate research in detecting attack on the sensors themselves in WSS. Our work addresses this issue.

7.2 Modeling the Water Supply System

We begin by modeling the WSS using the Saint-Venant equations. For a steady state water flow, the hyperbolic and continuous Saint-Venant model are then linearized and discretized for its direct application in the Kalman Filter (to be discussed in next section).

7.2.1 The Saint-Venant Model

The Saint-Venant equations are derived from the conservation of mass and momentum [70]. These equations are first order hyperbolic nonlinear partial differential equations and for one dimensional flow with no lateral inflow, these equations can be written as:

$$T\frac{\delta H}{\delta t} + \frac{\delta Q}{\delta x} = 0 \tag{7.1}$$

$$\frac{\delta Q}{\delta t} + \frac{\delta}{\delta x}\left(\frac{Q^2}{A}\right) + \frac{\delta}{\delta x}(gh_cA) = gA(S_b - S_f) \tag{7.2}$$

for $(x,t) \in (0,L)X\mathbf{R}^{+}$, where L is the length of the flow (m), $Q(x,t) = V(x,t)A(x,t)$ is the discharge or flow (m^3/s) across cross section, $A(x,t) = T(x)H(x,t).V(x,t)$ refers to velocity (m/s), $H(x,t)$ refers to water depth (m) and $T(x,t)$ refers to the free surface width (m), S_f(x,y) is the friction slope, S_b is the bed slope and g is the gravitational acceleration (m/s^2), These equations can be elaborated [71] in terms of water depth and velocity as:

$$T\frac{\delta H}{\delta t} + \frac{\delta(THV)}{\delta x} = 0 \tag{7.3}$$

$$\frac{\delta V}{\delta t} + V\frac{\delta V}{\delta x} + g\frac{\delta H}{\delta x} = g(S_b - S_f) \tag{7.4}$$

The friction is empirically modeled by the Manning-Stickler's formula:

$$S_f = \frac{m^2 V|V|(T+2H)^{\frac{4}{3}})}{(TH)^{\frac{4}{3}}} \tag{7.5}$$

where m is the Manning's roughness coefficient $(s/m^{\frac{1}{3}})$

7.2.2 Steady State Flow

There exists a steady state solution of the Saint-Venant equations under constant boundary conditions [70]. We denote the variables corresponding to the steady state condition by adding suffix 0. By excluding term containing δt and expanding equation (7.3), we obtain the following equation:

$$\frac{dV_0(x)}{dx} = -\frac{V_0(x)}{H_0(x)}\frac{dH_0(x)}{dx} - \frac{V_0(x)}{T(x)}\frac{dT(x)}{dx} \tag{7.6}$$

Solving (7.4) and (7.6), we get,

$$\frac{dH_0(x)}{dx} = \frac{S_b - S_f}{1 - F_0(x)^2} \tag{7.7}$$

with $F_0 = V_0/C_0$, $C_0 = \sqrt{gH_0}$. Here C_0 is the gravity wave celerity, F_0 is the Froude number. We assume the flow to be subcritical i.e, $F_0 < 1$ [70].

7.2.3 Linearized Saint-Venant Model

The linearized Saint-Venant model can be obtained from the steady-state flow characterized by V_0 and H_0 [70]. Let, $v(x,y)$ and $h(x,y)$ denote the first-order perturbations in water velocity and water level. Then,

$$V(x,t) = V_0(x,t) + v(x,t) \tag{7.8}$$

$$H(x,t) = H_0(x,t) + h(x,t) \tag{7.9}$$

The values of H and V are substituted in equation (7.3) and (7.4) and expanded in Taylor Series. We use T_0 in place of T to emphasize that it is uniform. As described in [70] neglecting higher order terms, a given term $f(V,H)$ of Saint-Venant model can be written as: $f(V,H) = f(V_0,H_0) + (f_V)_0 v + (f_H)_0 h$ in which, $()_0$ indicates steady state conditions. The linearized Saint-Venant equations can be obtained as the following [71],[72].

$$h_t + H_0(x)v_x + V_0(x)h_x + \alpha(x)v + \beta(x)h = 0 \tag{7.10}$$

$$v_t + V_0(x)v_x + gh_x + \gamma(x)v + \eta(x)h = 0 \tag{7.11}$$

where $\alpha(x), \beta(x), \gamma(x)$ and $\eta(x)$ are given by,

$$\alpha(x) = \frac{dH_0}{dx} + \frac{H_0}{T}\frac{dT_0}{dx} \tag{7.12}$$

$$\beta(x) = -\frac{V_0}{H_0}\frac{dH_0(x)}{dx} - \frac{V_0(x)}{T(x)}\frac{dT(x)}{dx} \tag{7.13}$$

$$\gamma(x) = 2gm^2\frac{|V_0|}{H_0^{\frac{4}{3}}} - \frac{V_0}{H_0}\frac{dH_0(x)}{dx} - \frac{V_0(x)}{T(x)}\frac{dT(x)}{dx} \tag{7.14}$$

$$\eta(x) = -\frac{4}{3}gm^2\frac{V_0|V_0|}{H_0^{\frac{7}{3}}} \tag{7.15}$$

7.2.4 Discretization

In order to discretize the linear equations generated in previous section, we use the Lax Diffusive Scheme [71] as listed below. The channel is divided into smaller segments of length Δx and a suitable time interval Δt is selected.

$$\frac{\delta v}{\delta t} = \frac{v_i^{k+1} - \frac{1}{2}(v_{i+1}^k + v_{i-1}^k)}{\Delta t} \tag{7.16}$$

$$\frac{\delta v}{\delta x} = \frac{(v_{i+1}^k + v_{i-1}^k)}{2\Delta x} \tag{7.17}$$

$$\frac{\delta h}{\delta t} = \frac{h_i^{k+1} - \frac{1}{2}(h_{i+1}^k + h_{i-1}^k)}{\Delta t} \tag{7.18}$$

$$\frac{\delta h}{\delta x} = \frac{(h_{i+1}^k + h_{i-1}^k)}{2\Delta x} \tag{7.19}$$

Given $(h_i^k, v_i^k)_{i=0}^I$, we want to compute $(h_i^{k+1}, v_i^{k+1})_{i=0}^I$. Here I is the total number of segments of length Δx. The updated equations for (h_i, v_i) are:

$$\begin{aligned} h_i^{k+1} \;=\; & \frac{1}{2}(h_{i+1}^k + h_{i-1}^k) \\ & -\frac{\Delta t}{4\Delta x}(H_{0(i+1)} + H_{0(i-1)})(v_{i+1}^k - v_{i-1}^k) \\ & -\frac{\Delta t}{4\Delta x}(V_{0(i+1)} + V_{0(i-1)})(h_{i+1}^k - h_{i-1}^k) \\ & -\frac{\Delta t}{2}\alpha_{i+1}v_{i+1}^k + \alpha_{i-1}v_{i-1}^k \\ & -\frac{\Delta t}{2}\beta_{i+1}h_{i+1}^k + \beta_{i-1}h_{i-1}^k \end{aligned} \tag{7.20}$$

$$
\begin{aligned}
v_i^{k+1} &= \frac{1}{2}(v_{i+1}^k + v_{i-1}^k) \\
&\quad -\frac{\Delta t}{4\Delta x}(V_{0(i+1)} + V_{0(i-1)})(v_{i+1}^k - v_{i-1}^k) \\
&\quad -\frac{g\Delta t}{2\Delta x}(h_{i+1}^k - h_{i-1}^k) \\
&\quad -\frac{\Delta t}{2}\gamma_{i+1}v_{i+1}^k + \gamma_{i-1}v_{i-1}^k \\
&\quad -\frac{\Delta t}{2}\eta_{i+1}h_{i+1}^k + \eta_{i-1}h_{i-1}^k
\end{aligned} \tag{7.21}
$$

We assume that Δx is very small then we can write that $h_{i-1} = h_i = h_{i+1}$ and $v_{i-1} = v_i = v_{i+1}$. The above equations will become:

$$
\begin{aligned}
h_i^{k+1} &= (1 - \frac{\Delta t}{2}\beta_i + \beta_i)h_i^k \\
&\quad +(\alpha_i - \frac{\Delta t}{2}\alpha_i)v_i^k
\end{aligned} \tag{7.22}
$$

$$
\begin{aligned}
v_i^{k+1} &= (\eta_i - \frac{\Delta t}{2}\eta_i)h_i^k \\
&\quad +(1 - \frac{\Delta t}{2}\gamma_i + \gamma_i)v_i^k
\end{aligned} \tag{7.23}
$$

7.2.5 Discrete Linear State-Space Model

From the discretized equations in previous section, state-space model can be formed as follows:

$$x(k+1) = Ax(k) + Bu(k) + w(k) \tag{7.24}$$

where, $x(k) = (v_0^k, ..., v_I^k, h_0^k, ..., h_I^k)^T$, with the applied control $u(k)$ in the form of discharge perturbation at the upstream end v_0^k and the discharge perturbation $w(k)$ at the downstream end v_I^k [70]. Here, w_k, x_0 are independent Gaussian random variables, and $x_0 \sim \mathcal{N}(0, \Sigma)$ and $w_k \sim \mathcal{N}(0, Q)$.

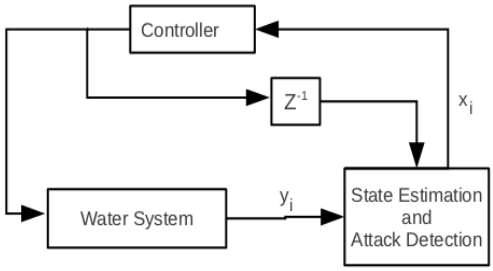

Figure (7.1) Kalman Filter

7.3 Detecting Attacks using Kalman Filter

In this section, we introduce the Kalman Filter [22] technique to obtain estimates for the state space vector $x(k)$ described in Section 7.2.5. Figure 7.1 shows the control system of the Kalman Filter with the sensor readings or observations from the water supply system namely, y_i. And x_i denotes the output of the control system that is fed to the controller. The observations (y_i) are forwarded to the central system containing estimator and detector at a regular time interval denoted by Δt. At each time step Δt, the estimator of the system generates estimated readings based on the reading of previous time step. These readings are used by the detector to detect the difference between the newly observed sensor readings and estimated readings.

7.3.1 The Kalman Filter

To apply the Kalman Filter technique, the observation equation for the above system can be written as:

$$y_k = Cx_k + \nu_k \tag{7.25}$$

Here, $y_k = [y_1^k, ..., y_m^k]^T \in \mathbf{R^m}$ is measurement vector collected from the sensors and y_i^k is the measurement generated by sensor i at time k. v_k is the measurement noise and assumed to be white Gaussian noise, which is independent of initial conditions and process noise.

Kalman filter can then be applied to compute state estimations $\hat{x}_k$ using observations

y_k.

7.3.2 Attack/Failure Detection

Since it is assumed that the noises in the system is Gaussian, we use χ^2 detector to compute the difference between the observed value from the sensors and the estimated values from the Kalman Filter as the following [14]:

The residue z_{k+1} at time $k+1$ is defined as:

$$z_{k+1} \stackrel{\Delta}{=} y_{k+1} - \hat{y}_{k+1|k}$$

equivalently,

$$z_{k+1} \stackrel{\Delta}{=} y_{k+1} - C(A\hat{x}_k + Bu_k) \tag{7.26}$$

$$g_k = z_k^T \mathcal{P} z_k \tag{7.27}$$

where $\mathcal{P}$ is the covariance matrix of z_k, the residue. The χ^2 detector compares g_k with a certain threshold to detect a failure or attack and triggers the alarm for potential attack or failure.

$$g_k > threshold, \tag{7.28}$$

where g_k is defined as:

$$g_k = g(z_k, y_k, \hat{x}_k, ..., z_{k-\tau+1}, y_{k-\tau+1}, \hat{x}_{k-\tau+1}) \tag{7.29}$$

The function g is continuous and $\tau \in \mathbf{N}$ is the window size of the detector [14].

7.4 Attacks and Defenses

Without loss of generality, we use level and velocity sensors in this work to design the framework. However, it must be noted that the framework is valid when any other types of sensors are used in the WSS. Without limiting ourselves to just level and velocity sensors, in this section, we list various attack/fault models that can be defended using the framework designed above.

1. WSS fault: Any system fault that can be detected by the sensors of the WSS, for example, water leakage, unintentional addition of contamination, will alert the detector and the alarm will be triggered.

2. Naive attack on the system: If the attacker simply adds contaminants to the water or steals water from the WSS, the sensors will report it to the central system. The detector will detect it immediately as the difference between estimated and actual observation will be large and will trigger the alarm.

3. Physical damage to the sensors: When the sensors are damaged physically, no measurements are obtained causing the system to detect it immediately. As in case of naive attack, the estimated reading will be different from the actual reading and will cause the alarm to trigger.

4. Random data injection: Let us assume that the attacker has control over some sensors and feeds some random values to mislead the system. As these randomly generated sequences will not correspond to the estimates generated by the Kalman Filter, the detector will detect the difference and report the attack.

7.5 Injection of False Data into the System

The attacks discussed above do not take statistical attack into account. The authors of [14] proposed False Data Injection attack that uses statistical analysis and has proved that a system with the Kalman Filter estimator can be attacked by generating an attack sequence

y_k^a. It is assumed that the attacker knows the matrices A, B, C, Q, R of the system along with observation gain K. The attack sequence changes the measurements obtained from the sensors to:

$$y'_k = Cx'_k + v_k + \Gamma y_k^a \tag{7.30}$$

where $\Gamma = diag(\gamma_1, ...\gamma_m)$ is the sensor selection matrix. γ_i is a binary variable whose value is 1 if the sensor i has been compromised, 0 otherwise. y_k^a is the input from the attacker. y'_k and x'_k denote the partially compromised system and are different from the original non-tampered values. Then the system dynamics changes as following:

$$\begin{aligned} x'(k+1) &= Ax'(k) + Bu'(k) + w(k) \\ y'_k &= Cx'_k + \nu_k + \Gamma y_k^a \\ \hat{x}'_{k+1} &= \hat{x}'_k + Bu'_k + K_k[y'_{k+1} - C(A\hat{x}'_k + Bu'_k)] \end{aligned} \tag{7.31}$$

The new residue and estimation error can be defined as

$$\begin{aligned} z'_{k+1} &= y'_{K+1} - C(A\hat{x}'_k + Bu'_k) \\ e'_k &= x'_k - \hat{x}'_k \end{aligned} \tag{7.32}$$

Also, the new error detection function can be defined as

$$g'_k = g(z'_k, y'_k, \hat{x}'_k, ..., z'_{k-\tau+1}, y'_{k-\tau+1}, \hat{x}'_{k-\tau+1}) \tag{7.33}$$

The alarm will be triggered when

$$g'_k > threshold \tag{7.34}$$

As stated in [14], if the attacker simply injects a large y_k^a, the residue z'_k will be large as well resulting in attack detection by the detector. However, if the vector $\hat{x}'_k, y'_k, z'_k$, has

the same statistical properties in the partially compromised system as those of the healthy system, then the attack can be successful.

The algebraic condition to identify perfectly attackable systems is presented in [14]. As stated in the paper, the system described above is perfectly attackable iff A has an unstable eigenvalue and the corresponding eigenvector v satisfies the following conditions:

1. $Cv \in span(\Gamma)$, where $span(\Gamma)$ is the column span of Γ
2. v is the reachable state of dynamic system $e_{k+1} = (A - KCA)e_k - K\Gamma y^a_{k+1}$

Using the results from [14], the attack sequence can be generate using the following equation:

$$y^a_{n+i} = y^a_i - \frac{\lambda^{i+1}}{M} y^*, i = 0, 1, 2, ... \tag{7.35}$$

where, $|\lambda| \geq 1$ and $Cv \in$ span(Γ). There exists y^* such that $\Gamma y^* = Cv$ and M = max $||\Delta z_k||$. Following equation (7.35), the attacker can generate an attack sequence based on eigen decomposition of matrix A and matrix Γ.

7.5.1 Defense Against Data Injection Attack

As proposed in [14], the eigen decomposition could also be performed at the defender side on matrix A to find all the unstable eigenvector v to compute Cv. For each Cv, the 1's will indicate the compromised sensors needed by the attacker to perform a successful attack along direction v. Therefore, the defender can defend the system by deploying more redundant sensors along the direction of attack.

Another powerful defense mechanism is to implement the data encryption algorithms. The data from sensors to the estimator can be encrypted using either symmetric key or asymmetric key. The key can be periodically exchanged between the sensors and the central system which will make data injection more difficult.

7.5.2 Case Study

Using the Water Supply model and attack model described in sections above, a more comprehensive illustration is provided here. For the sake of simplicity, one water level sensor and one drifter is considered. Let the dimension of the state space be $n = 2$. Then, from equation (7.24) and (7.25) and assuming $\Delta t = 1$, we get:

$$X_{k+1} = \begin{bmatrix} 1 + \frac{\beta_i}{2} & \frac{\alpha_i}{2} \\ \frac{\eta_i}{2} & 1 + \frac{\gamma_i}{2} \end{bmatrix} X_k + w_k$$

where, $X_k = \begin{bmatrix} h_k \\ v_k \end{bmatrix}$

Also,

$$y_k = X_k + \nu_k \tag{7.36}$$

Considering the above system and substituting Manning's Coefficient $m = 0.025$, the matrix A will be: $A = \begin{bmatrix} 1 & 0 \\ 0 & 1.006 \end{bmatrix}$ and value of K will be: $K = \begin{bmatrix} 0.618 & 0 \\ 0 & 0.6193 \end{bmatrix}$

The eigenvector for the system with level sensor compromised will be $[0, 1]^T$ and with $n = 2$ the attack sequence can be designed to be:

$$y^a_{2+i} = y^a_i - \frac{\lambda^{i+1}}{M} y^*, \tag{7.37}$$

where i = 1,2,3,... and $\lambda <= 1$ and $M <= 1$

PART 8

TOWARDS THE SECURED CPS: A CASE STUDY ON ANONYMITY OF MOBILITYFIRST NETWORKING

With more and more mobile/static devices being connected to the current internet, the existing networking infrastructure may not be sufficient to handle these devices in future. With that in mind, researchers are studying possible network architectures for future internet. Further, since network infrastructure is an integral part of any CPS, it is important to secure the underlying network to build a secure CPS. Hence, in this chapter, we study anonymity and security of the nodes in the MobilityFirst future internet architecture.

8.1 Security in MF network overview

With the ever increasing demands of mobile and networking enabled devices, much recent research is conducted to incorporate these devices into a delay-tolerant, scalable, evolvable, secure, trustworthy and context aware architecture that can provide seamless mobility in future Internet. MobilityFirst (MF) network [7] is an architecture designed to facilitate the mobile as well as non-mobile devices in future Internet. MF Network uses Globally Unique Flat Identifiers (GUIDs) which are public key based self-certifying flat identifiers assigned to the devices in the network in order to provide strong authentication and security. As these GUIDs are flat, the identity of a device in MF Network is disassociated from its network location. Hence, when a device in MF Network moves from one network to another, its GUID remains the same whereas its network address (NA) changes. In other words, GUIDs are dynamically bound to routable network addresses using the Global Name Resolution Service (GNRS). GNRS employs a Direct Mapping scheme (DMap) to map the GUIDs to NA. DMap applies k hashing functions to produce a list of k network addresses and stores the GUID to NA mapping in the Autonomous Systems (ASs) that own these network

addresses [73] such that the mappings from GUID to NAs are proportionally distributed among ASs. This technique allows the hosting ASs to be derived locally from the identifier by any network entity.

Certain level of privacy is embedded in the MF network architecture owing to the fact that the GUIDs are separated from NAs. GUIDs are flat identifiers and do not disclose any location information. Nevertheless, GUIDs are long term, persistent identifiers of network identities, they can be easily exploited to track the end users. To overcome this problem, disposable GUIDs [74] can be used. However, disposable GUIDs are still mapped on to the NAs and these NAs give away coarse location information. Thus, varying the GUID can not provide the users with complete protection, as the users may have the same NA. In fact, the issues on anonymity and privacy in MF Network are wide open. The literature schemes (such as the ones described in Section II) are proposed for traditional Internet and cannot be directly applied to MF Network because MF Network does not use the same naming techniques and architectures as the traditional network. Hence, to provide anonymity and privacy to the users, it is necessary to make sure that the adversaries cannot eavesdrop and decode the information such as source, destination and the routing path of the packet.

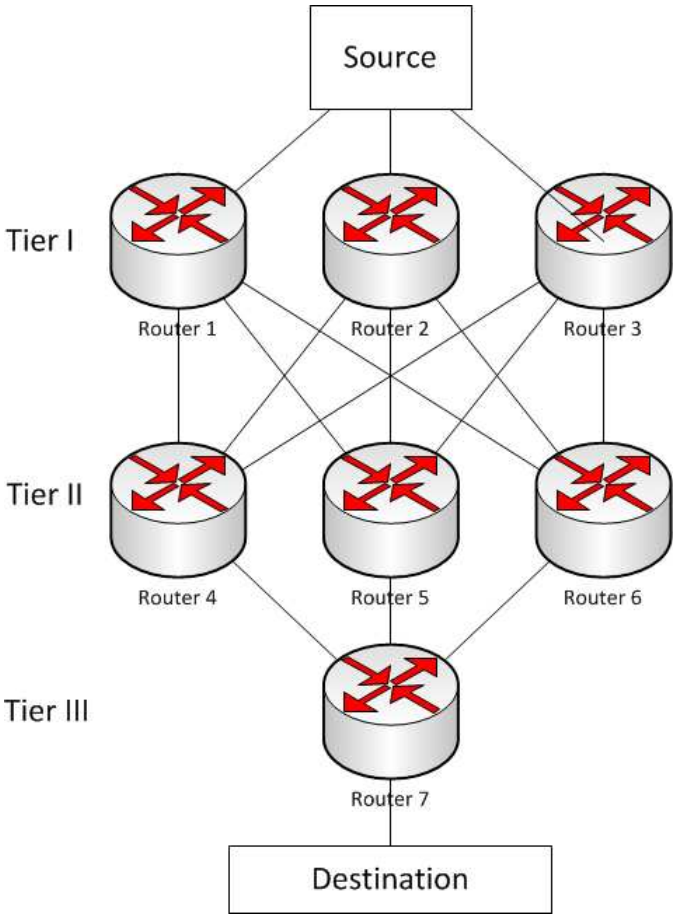

Figure (8.1) Virtual network topology in AMF

Sensitive user information can be collected by adversaries spying on the Internet connection. User location, websites visited, personal information etc. can be identified by doing active/passive network analysis in current Internet. To mitigate such privacy compromising in traditional Internet, much literature research has been conducted. For example, the studies in [75], [76], [77], [78], [79] suggest to use the cryptography and anonymous routing protocols to create a secure layer over the Internet to provide anonymity to the end users. Similarly, the use of pseudonyms and a trusted central server is investigated in [80], [81], [82], [83], [84], [85]. The work in [76] develops cryptography and secret sharing to provide identity anonymity, location anonymity, data and traffic anonymity in traditional network. The study in [77] proposes to integrate authenticated key exchange mechanism into the

routing algorithm to provide some anonymity. In [78] anonymous routing protocol that can preform MAC-layer and network-layer communications is discussed. The authors in [80] use the privacy-tag to hide actual location revealing address in case of mobile devices. The work in [81] provides anonymity by changing pseudonym using a trusted third party server and the effectiveness of frequently changing pseudonyms in case of vehicular network is discussed in [82]. The authors in [86] have proposed to use anonymity zone to provide anonymity to the source and the destination. Based on the onion routing technique, Tor is a practical tool that provides users with privacy and prevents traffic analysis in Internet [75]. However, as mentioned earlier, these literature schemes cannot take advantage of GUID and GNRS in MF Network and cannot directly be applied to MF Network due to the differences in architecture and naming schemes. Furthermore, in the case of Tor, it maintains a list of nodes that participate in the routing process and if adversaries can control many routers in the limited list, they may collude to decrypt the route exposing the end users

To overcome such vulnerability, the Anonymity in MobilityFirst (AMF) scheme utilizes the available GUID and GNRS services provided by the MF Network architecture along with intelligent key exchanges among the different tiers to setup a secure path between the two end users. The basic idea of AMF is shown in Figure 8.1, where the source sets up a route using three tiers of routers to connect to the destination. As to be elaborated in Section 8.2 and 8.3, Tier I has three random routers that can identify source and Tier II routers but are unaware of each other. Tier II also has three random routers that are unaware of each other and can identify Tier I routers and Tier III router. Tier III has one router which can identify Tier II routers and the destination router. Using the same hashing technique as in the case of identification of hosting ASs, the routers in the next tier are derived locally and route establishment proceeds only if all these multiple routers generate the same GUIDs as next tier routers.

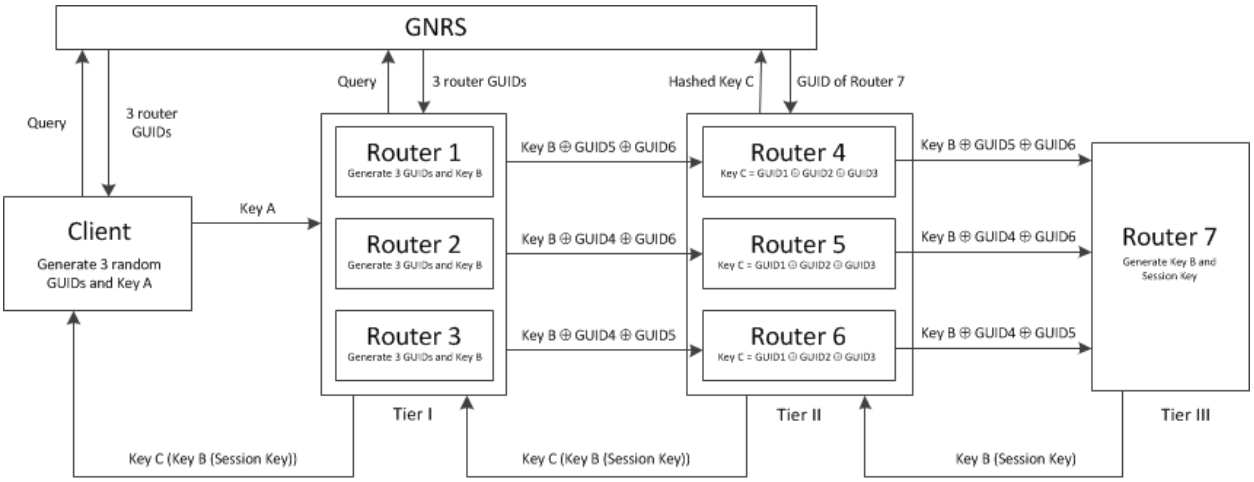

Figure (8.2) AMF Anonymity Architecture

8.2 System Model

Figure 8.1 shows the virtual connection topology of the AMF scheme. AMF maintains a list of available routers with their public keys. When data is to be transferred across the two users, it selects three routers and encrypts the data into three layers using the public keys of the routers selected. The encrypted data is then transmitted across the network. Three tiers are selected to ensure that the routers only have the knowledge of their immediate neighbors. In other words, the first tier only knows the source and not the destination, the third tier only knows the destination and not the source. The middle or second tier is used as the buffer between the first and the third tier. The first and second tiers have three random routers each. Three routers in these tiers are selected to ensure that they cannot collude to identify second and third tier routers without performing any GNRS query or work together to identify the source and the destination. The third tier has only one router since it does not need to identify any following tiers. These routers are identified in a random fashion and picked from the available pool of routers in the network which minimizes the chances of collusion between these selected routers thus removing the issues caused by trusting any node in the system. Specifically, the existing GNRS feature in MF Network architecture is

used and the techniques similar to generating k NAs using hashing function while storing the GUID to NA mapping can be used to identify multiple routers in each tier. Note that the router that performs the GNRS lookup will know the GUID of the network node that initiated the GNRS request and vice-versa. Thus, we cannot use this method for generating GUIDs for all three tiers as in the case of Tor [75]. The encrypted data is then transmitted across the network. Each router in the three tiers, upon receiving the packet, decrypts it and generates the address of next router and forwards it while providing the anonymity for the end users.

8.3 Anonymity in MobilityFirst (AMF)

Figure 8.2 presents the tiered AMF anonymity architecture in detail. To establish an anonymous session, the source starts by generating four random GUIDs, the first three are used to query GNRS to get three routers in the first tier. The fourth GUID serving as Key A is distributed to all Tier I routers. These routers hash the key four times to get four random GUIDs. The first three of them are used to obtain Tier II routers and the fourth random GUID is used as Key B. XORed Key B (each three routers in Tier II has its own XORed version of Key B) is distributed to Tier II routers. Tier II routers use XORed GUIDs of Tier I routers to obtain Key C. The Key C is hashed to get a single Tier III router. Tier III router performs XOR on different versions Key B obtained from Tier II routers to get the original Key B. After Key B is revealed to Tier III router, Tier III router generates a random session key, encrypts it with Key B and distributes it to all three Tier II routers. Tier II routers encrypt it again using Key C and transfer it to all three Tier I routers. Since Tier I routers do not have Key C, they will transfer the key to the source. As the source can generate both Key B by hashing Key A and Key C by XORing the Tier I GUIDs, the source will decrypt the session key to establish the session to connect with the destination. The following sub-sections describe how each tiers are generated and how they talk with each other in detail.

8.3.1 Client to Tier I

The client requesting anonymous connection initiates the communication by identifying the first tier routers. On one hand, choosing only two routers for the first tier will not help in identifying the misbehaving router that generates next tier router without following the protocol. On the other hand, choosing four or more routers unnecessarily complicate the scheme. Hence, we choose three routers such that if a router misbehaves and generates GUIDs that are not same as the other two, it can be identified. The client generates four GUID-length random strings of data and then feeds three of these strings into GNRS, from which the client (or source) gets three replies. The three routers that respond are identified by the client as Tier I routers. The fourth random string generated is Key A. This key is used as the seed to generate Tier II routers by Tier I routers. It must be noted that the routers in Tier I do not recognize each other and only communicate with client at this time.

8.3.2 Tier I to Tier II

After the establishment of route between client and Tier I routers, client sends Key A to the three Tier I routers. Each Tier I router hashes Key A for a set number of times to generate four more GUID-length strings. Again, following the same procedure as in the case of identifying Tier I routers by the client, the Tier I routers use GNRS to convert the first three hashed strings into GUIDs, thus generating the three Tier II addresses. The fourth random string generated by Tier I is Key B. Note that the client also knows Key A and can perform the same hash function to generate the four random strings including Key B. However, the client will refrain from performing the GUID lookups itself because Tier II routers should not know the identity of the client node. If only one Tier I router is used, the router could contact another router to collude with instead of performing GNRS lookup. Hence, it is important to have multiple ($>= 3$) Tier I routers. The route establishment can proceed only if all the three Tier I routers generate the same Tier II routers. And all Tier II routers must hear from all Tier I routers for the route setup. Key B is transferred from

the Tier I routers to the Tier II routers after XORing it with the GUIDs of the other Tier II routers. Since Key B is a randomly generated data and the Tier II routers are unaware of GUIDs of the source, destination and other Tier II routers, it is impossible for a Tier II router to derive the Key from the XORed GUIDs.

8.3.3 Tier II to Tier III

As discussed in the above sections, the Tier II routers have GUIDs of all the Tier I routers, thus Key C can be generated by XORing all of the GUIDs of Tier I routers. The XOR operation is commutative hence the order of XORing is irrelevant. Note that the original client can also generate Key C by XORing the GUIDs of Tier I routers. Key C is then hashed and fed through GNRS to obtain the GUID of the Tier III router. In the mean time, the Tier II routers transfer the XORed copies of the Key B to Tier III router, originally obtained by Tier II routers from Tier I routers. As can be seen in Equation (8.1), by XORing all of the XORed copies of Key B together, the Key B is revealed to the Tier II router.

$$\begin{aligned} &(Key\ B \oplus GUID5\ \oplus GUID6) \\ &\qquad \oplus \\ &(Key\ B \oplus GUID4\ \oplus GUID6) \\ &\qquad \oplus \\ &(Key\ B \oplus GUID4\ \oplus GUID5) \\ &= Key\ B \end{aligned} \tag{8.1}$$

8.3.4 Route Establishment

For the encrypted route establishment, a random session key is selected by the Tier III router. This session key is then encrypted with Key B and transmitted to Tier II routers. Which is then encrypted by Tier II routers using Key C and transmitted to Tier I routers. Although, Tier I routers know Key B, they cannot decrypt the session key as it is further

encrypted using Key C. Hence the session key is securely encrypted and transmitted to the original client. Finally, the client has both the keys: Key B and Key C, and is able to decode to reveal the session key. It must be noted that the client receives three encrypted copies of the session key and the route establishment will complete only if all three copies are the same. This assures that the session key was encrypted as required by the scheme. Once the session key is obtained, the client can randomly choose one of the Tier I routers and this router can randomly choose one of the Tier II router to setup the path along which the data traffic is routed. By employing the concept of disposable GUID, the session keys can also be changed frequently to prevent a single compromised key from revealing the contents of the entire conversation to further enhance the anonymity.

8.3.5 The Stepwise Procedure in AMF

The stepwise route establishment process, as described in Figure 8.3, is further elaborated in the steps below.

- Step 1: The client (or source) generates four random GUID length strings and chooses one of them to be Key A. The GNRS query using the remaining GUIDs allows the client to obtain GUIDs of three Tier I routers, which further enable the client to establish a connection with these three routers and transfer Key A.

- Step 2: In Tier I routers, Key A is hashed four times to get three random GUIDs and Key B. The GNRS query using the first three random GUIDs allows Tier I routers to obtain three Tier II GUIDs, which enable Tier I routers to establish connections with Tier II routers. To each router in Tier II, Tier I routers transfer Key B XORed with the GUIDs of the two remaining Tier II routers.

- Step 3: In Tier II routers, Key C is generated by XORing all 3 GUIDS of Tier I routers. Through hashing Key C and querying GNRS to get GUID of Tier III router, a connection with Tier III router can be established. Tier II routers then forward XORed Key B from Tier I routers to Tier III router without altering it.

- Step 4: In Tier III router, XOR operation is performed on all keys obtained from Tier II routers to get Key B. A random Session Key is generated and encrypted with Key B, which is then transmitted to Tier II routers.
- Step 5: Tier II routers encrypt the obtained session key using Key C and transmit it to Tier I routers.
- Step 6: Tier I routers transmit the key as it is to the Client.
- Step 7: The client (or source) obtains Key B by hashing Key A and obtains Key C by XORing the GUIDs of the Tier I routers. Now the client can use these keys to decrypt the session key and establish an anonymous connection with the destination using this session key.

8.4 Analysis

In order to ensure privacy and anonymity of the users, there should not be any data leakage between or amongst router layers. It is also very important to ensure that the GUIDs of routers in Tier I and Tier II are not exposed within the same layer. For example, if Tier II routers can identify each other, Key B can be decrypted and used to hijack the session. Thus, GUIDs of routers in the same layers should never be leaked between the routers in the same tier. Similarly, the random keys were used as the seed to deterministically derive GUIDs. It must ensured that the routers are unable to backtrack the hashes to obtain the key used as the seed. Once the seed is obtained, it can follow the same hashing technique to identify the participating routers in any layer and use it to hijack the session. However, since the order in which the Tier I routers pick the Tier II routers is not known to the Tier II routers only one of the three routers will be able to backtrack the seed. The route setup is a one-time process. Once the route is set up and the session keys are exchanged, the data communication can be preformed as in the case of normal Internet traffic. This implies the complexity of the scheme is defined by the complexity of the path setup procedure. In the scheme presented in this chapter, the client performs 4 hashes to generate 3 GUIDs and a

seed for the Tier I routers. Three GUID lookups are performed in parallel and as stated in [73], the total lookups take around 83ms 95% of the time. Once the Tier I routers are identified, the source transmits Key A to these routers. Tier I routers similarly perform 4 hashes (one of them is Key B) and 3 GUID lookups. Finally the Tier II routers generate the third GUID by performing XOR operation over the GUIDs of the Tier I routers and perform a GUID lookup. The Tier III router then generates the session key and encrypts the session key using Key B, which is further encrypted using Key C at Tier II and transferred all the way back to the client (or the source). The client can get the session key after decryption. Thus, the scheme requires at least 15 encryption, decryption and hashing functions and 7 GUID lookups to set up the routing path.

8.4.1 Comparison with Tor

In our scheme, the traffic is encrypted twice: using the GUID of the intermediate router and using the GUID of the final destination. In both cases, the routers only know their immediate neighbors. However, with Tor [75], the traffic is encrypted by the client three times, which is decrypted by the routers one after another and sent to the next hop. In addition, routers do not generally know which tier they belong to unless the last hop receives the plain text, in which case it is the third tier. In the AMF scheme each router, while unaware of the other routers in each tier, is aware of which tier it belongs to. Thus, Tier I routers are certain that they are talking to the client, but still unaware of who the destination is. A major advantage of the AMF scheme is that, the routers in different tiers are identified from a large number of routers in the Internet thus reducing the chance of collusion amongst routers in different tiers. However, the list of nodes participating in route establishment in Tor could be small and the adversaries may be able to take advantage by committing many nodes to this list.

8.4.2 An Illustrated Example

Figure 8.4 shows a snapshot of how the route is setup and communication is started between Alice's device A1 and Bob's device B2. A1 first queries GNRS to obtain the GUIDs of three routers in the MF Network. As described in Figure 8.3, the route is established between the routers within the network and an encrypted session key is returned to Alice. Alice encrypts the message to Bob once using Bob's public key and again with the session key before transmitting it through the network using the selected route. The last Tier router in the network decrypts the message using session key and transfers it to Bob. Bob then, decrypts the message with his private key. As one can see that, the anonymity and privacy of the conversation between Alice and Bob can be preserved since eavesdropping and traffic analysis cannot occur along the routing path.

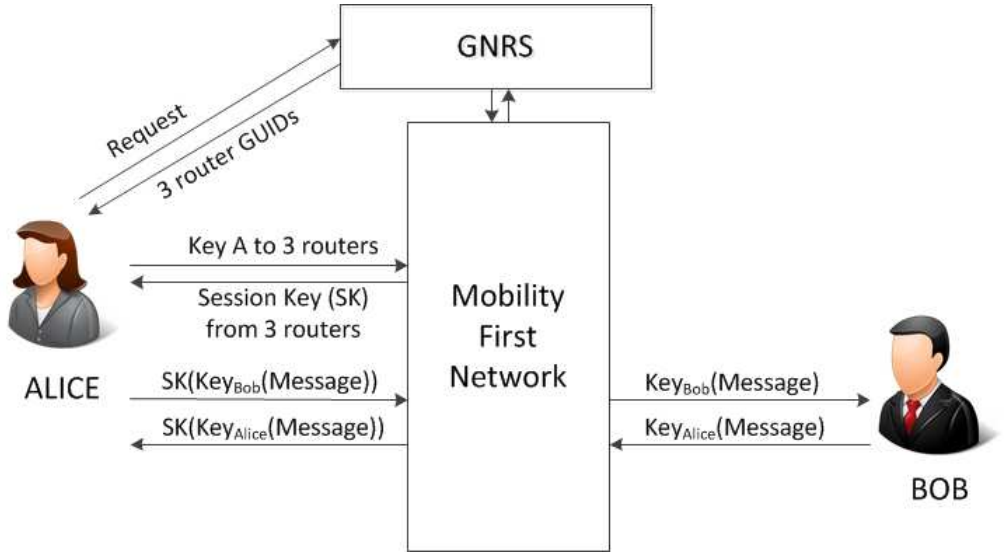

Figure (8.4) An AMF Example showing communication between Alice and Bob

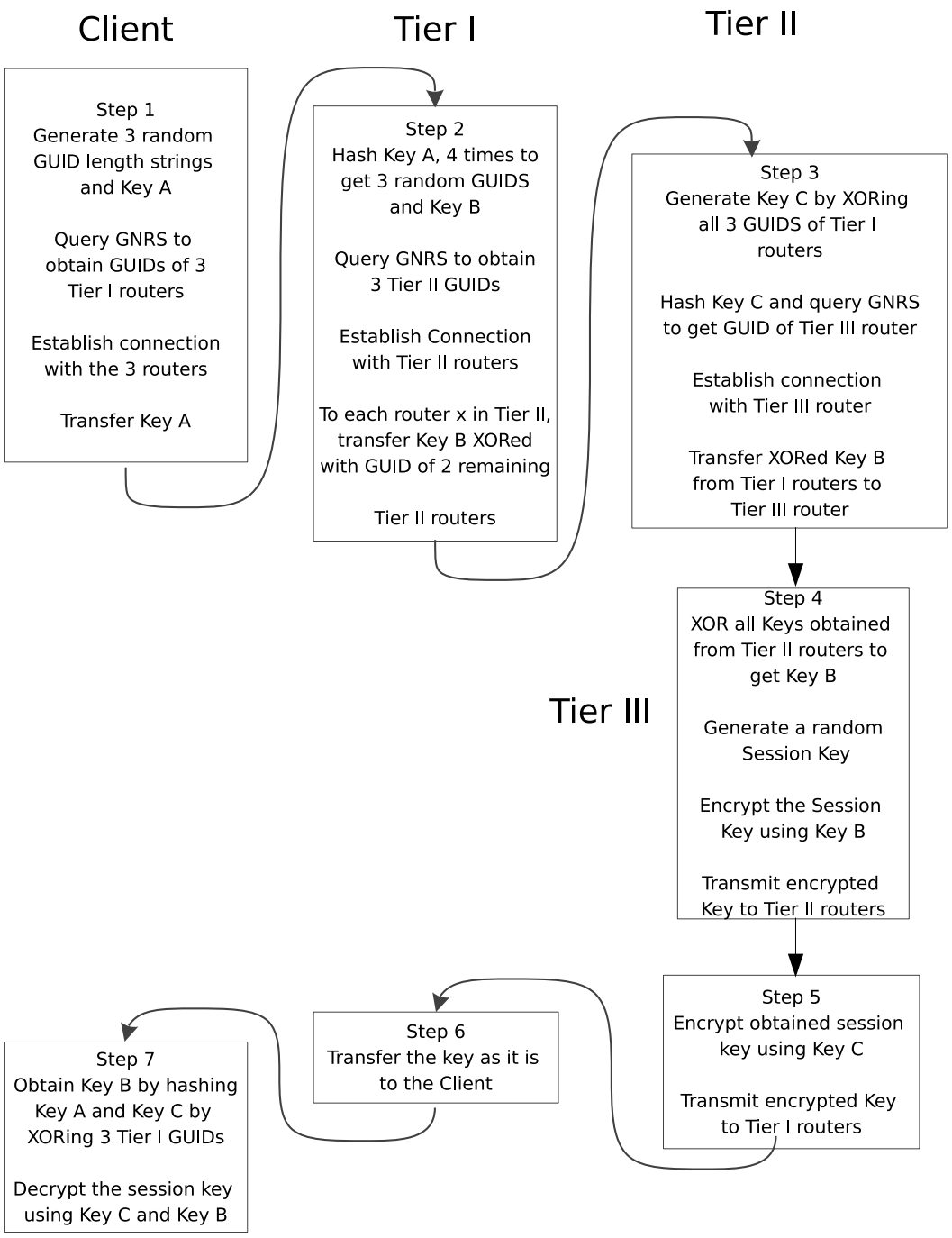

Figure (8.3) Stepwise description of AMF showing the key exchange and route establishment process

PART 9

CONCLUSIONS

A framework for smart grid system using Kalman Filter estimator together with the χ^2-detector and Euclidean detector has been designed. It has been shown that the χ^2-detector is efficient in detecting different types of faults and attacks like DoS attack and random attacks on the system. Further, to handle attacks on the system like False Data Injection which evades χ^2-detector, we have proposed Euclidean detector which uses Euclidean distance metric for detection. We have also shown that the false positives due to noise for the Euclidean detector can be reduced to less than 1% with the proper selection of the threshold. Our extensive simulation and analysis have demonstrated the effectiveness of the proposed Euclidean detector in detecting various types of attacks including the False Data Injection attack.

Two schemes using Kalman Filter and χ^2-detector have been proposed for attack/fault isolation. The first scheme uses the Generalized Observer Scheme (GOS) to identify a single sensor under attack or at fault. To identify attacks/faults on multiple sensor nodes, we proposed an Iterative Observe Scheme (IOS). IOS implements KF observers to identify the area under attack by splitting the set of sensors into subsets whenever the residue function exceeds the threshold. Our preliminary study shows, the IOS scheme requires $2m-1$ computations in the worst case and $2\log m-1$ computations in the best case, where m is the number of measurements in the system. GOS scheme requires m computations in both cases.

In order to be able to track malicious users in a smart grid monitoring energy consumption and fluctuations can be effective. Thus, in this dissertation scheduling and control of appliances including Heating, Ventilating and Air Conditioning (HVAC), Electric Vehicle (EV), and deferrable load such as washer/dryer, batch printer/photocopier machine have been studied. Given the temperature and price forecasts, a Linear Programming (LP) model

for the system is derived to maximize the user comfort. An iterative algorithm, namely, Comfort Prioritizing Greedy (CPG) was presented and compared with the bin-packing algorithm. It was shown that the proposed CPG algorithm can be effectively used to schedule the Home Energy Management (HEM) system.

Similalry, for the water supply system, a framework for attack/fault detection has been designed which is derived mathematically using the Saint-Venant equations for the shallow water systems. The Kalman Filter is then applied to these equations to obtain the state estimations for the WSS. The estimations obtained from the Kalman filter along with the actual observations are fed to a χ^2 detector and the difference between the two readings is compared with a given threshold. The detector is designed to trigger an alarm when the difference is greater than the threshold, hence detecting attacks or faults in the WSS. We have demonstrated various attacks can be detected by the framework. Furthermore, we have addressed a statistical attack model that could bypass the detection scheme and discussed possible defences against such an intelligent attack.

Finally, a scheme, namely, Anonymity in MobilityFirst (AMF), has been proposed to ensure anonymity in the future internet architecture. The AMF scheme employs a tiered approach while taking advantage of the core features offered by MobilityFirst Network to set up anonymous routing path and encrypted traffic delivery. It has been shown that with intelligent exchange of keys between different tiers of routers, user privacy in the MoblityFirst Network can be achieved without much overhead, which suggests that the proposed AMF can be included as a core service in MF Network.

REFERENCES

[1] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, June 2010, pp. 731 –736.

[2] E. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, May 2008, pp. 363 –369.

[3] D. Li, Z. Zhao, L. Cui, H. Zhu, L. Zhang, Z. Zhang, and Y. Wang, "A cyber physical networking system for monitoring and cleaning up blue-green algae blooms with agile sensor and actuator control mechanism on lake tai," in *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2011, pp. 732 –737.

[4] M. Li, Y. Liu, J. Wang, and Z. Yang, "Sensor network navigation without locations," in *INFOCOM 2009, IEEE*, April 2009, pp. 2419 –2427.

[5] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '11, New York, NY, USA, 2011, pp. 355–366. [Online]. Available: http://doi.acm.org/10.1145/1966913.1966959

[6] T. Anderson, K. Birman, R. Broberg, M. Caesar, D. Comer, C. Cotton, M. Freedman, A. Haeberlen, Z. Ives, A. Krishnamurthy, W. Lehr, B. Loo, D. Mazires, A. Nicolosi, J. Smith, I. Stoica, R. van Renesse, M. Walfish, H. Weatherspoon, and C. Yoo, "The nebula future internet architecture," in *The Future Internet*, ser. Lecture Notes in Computer Science, A. Galis and A. Gavras, Eds. Springer Berlin Heidelberg, 2013, vol. 7858, pp. 16–26. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38082-2_2

[7] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri, "Mobilityfirst future internet architecture project," in *Proceedings of the 7th Asian Internet Engineering Conference*, ser. AINTEC '11. New York, NY, USA: ACM, 2011, pp. 1–3. [Online]. Available: http://doi.acm.org/10.1145/2089016.2089017

[8] R. Baheti and H. Gill, "Cyber-physical systems," *The impact of control technology*, pp. 161–166, 2011.

[9] E. A. Lee and S. A. Seshia, *Introduction to embedded systems: A cyber-physical systems approach*. Lee & Seshia, 2011.

[10] A. Cardenas, S. Amin, and S. Sastry, "Secure control: Towards survivable cyber-physical systems," in *Distributed Computing Systems Workshops, 2008. ICDCS '08. 28th International Conference on*, June 2008, pp. 495–500.

[11] Y. Mo, T.-J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber physical security of a smart grid infrastructure," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195 –209, Jan. 2012.

[12] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, "Challenges for securing cyber physical systems," in *Workshop on future directions in cyber-physical systems security*, 2009.

[13] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security*, vol. 14, no. 1, pp. 13:1–13:33, Jun. 2011. [Online]. Available: http://doi.acm.org/10.1145/1952982.1952995

[14] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli, "False data injection attacks against state estimation in wireless sensor networks," in *2010 49th IEEE Conference on Decision and Control (CDC)*, Dec. 2010, pp. 5967 –5972.

[15] X. Wang and P. Yi, "Security framework for wireless communications in smart distribution grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 809 –818, Dec. 2011.

[16] Y. Zhang, L. Wang, W. Sun, R. Green, and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smart grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 796 –808, Dec. 2011.

[17] H. Li, L. Lai, and W. Zhang, "Communication requirement for reliable and secure state estimation and control in smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 3, pp. 476 –486, Sept. 2011.

[18] M. Fouda, Z. Fadlullah, N. Kato, R. Lu, and X. Shen, "A lightweight message authentication scheme for smart grid communications," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 675 –685, dec. 2011.

[19] Q. Li and G. Cao, "Multicast authentication in the smart grid with one-time signature," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 686 –696, Dec. 2011.

[20] S. Amin, A. A. Cárdenas, and S. S. Sastry, "Safe and secure networked control systems under denial-of-service attacks," in *Hybrid Systems: Computation and Control*. Springer, 2009, pp. 31–45.

[21] G. Welch and G. Bishop, "An introduction to the kalman filter," Chapel Hill, NC, USA, Tech. Rep., 1995.

[22] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal Of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[23] L. Kleeman, "Understanding and applying kalman filtering," in *Proceedings of the Second Workshop on Perceptive Systems, Curtin University of Technology, Perth Western Australia (25-26 January 1996)*, 1996.

[24] P. Louka, G. Galanis, N. Siebert, G. Kariniotakis, P. Katsafados, I. Pytharoulis, and G. Kallos, "Improvements in wind speed forecasts for wind power prediction purposes using kalman filtering," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 96, no. 12, pp. 2348–2362, 2008.

[25] J. B. Pearson and E. B. Stear, "Kalman filter applications in airborne radar tracking," *IEEE Transactions on Aerospace and Electronic Systems*, no. 3, pp. 319–329, 1974.

[26] V. J. Aidala, "Kalman filter behavior in bearings-only tracking applications," *IEEE Transactions on Aerospace and Electronic Systems*, no. 1, pp. 29–39, 1979.

[27] M. M. Olama, S. M. Djouadi, I. G. Papageorgiou, and C. D. Charalambous, "Position and velocity tracking in mobile networks using particle and kalman filtering with comparison," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 2, pp. 1001–1010, 2008.

[28] W. Conrad and C. Corrado, "Application of the kalman filter to revisions in monthly retail sales estimates," *Journal of Economic Dynamics and Control*, vol. 1, no. 2, pp. 177–198, 1979.

[29] Z. Huang, K. Schneider, and J. Nieplocha, "Feasibility studies of applying kalman filter techniques to power system dynamic state estimation," in *Power Engineering Conference*. IEEE, 2007, pp. 376–382.

[30] B. Sikdar and J. Chow, "Defending synchrophasor data networks against traffic analysis attacks," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 819 –826, Dec. 2011.

[31] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Distributed internet-based load altering attacks against smart power grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 667 –674, Dec. 2011.

[32] C. Alcaraz, C. Fernandez-Gago, and J. Lopez, "An early warning system based on

reputation for energy control systems," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 827 –834, Dec. 2011.

[33] C.-W. Ten, J. Hong, and C.-C. Liu, "Anomaly detection for cybersecurity of the substations," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 865 –873, Dec. 2011.

[34] H. Qi, X. Wang, L. Tolbert, F. Li, F. Peng, P. Ning, and M. Amin, "A resilient real-time system design for a secure and reconfigurable power grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 770 –781, Dec. 2011.

[35] S. Bi and Y. J. Zhang, "Defending mechanisms against false-data injection attacks in the power system state estimation," in *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, Dec. 2011, pp. 1162 –1167.

[36] R. B. Bobba, K. M. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. J. Overbye, "Detecting false data injection attacks on dc state estimation," in *First Workshop on Secure Control Systems (SCS 2010),CPSWEEK2010*, Apr. 2010.

[37] V. Sood, D. Fischer, J. Eklund, and T. Brown, "Developing a communication infrastructure for the smart grid," in *Electrical Power Energy Conference (EPEC), 2009 IEEE*, oct. 2009, pp. 1 –7.

[38] B. Brumback and M. Srinath, "A chi-square test for fault-detection in kalman filters," *IEEE Transactions on Automatic Control*, vol. 32, no. 6, pp. 552 – 554, Jun 1987.

[39] R. Wilson, "Pmus [phasor measurement unit]," *Potentials, IEEE*, vol. 13, no. 2, pp. 26–28, 1994.

[40] M. Djerf, "Power grid integration using kalman filtering," *Uppsala University, Signals and Systems Group*, no. 12003, p. 55, 2012.

[41] R. C. Dorf and J. A. Svoboda, *Introduction to electric circuits*. John Wiley & Sons, 2010.

[42] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," in *Preprints of the 1st Workshop on Secure Control Systems*, 2010, pp. 1–6.

[43] R. Zimmerman, C. Murillo-Sanchez, and R. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12 –19, Feb. 2011.

[44] J. George and I. Gregory, "Robust fault detection and isolation for stochastic systems," in *Proceedings of the American Control Conference (ACC)*, Jun. 2010, pp. 5421–5426.

[45] R. N. Clark, "A simplified instrument failure detection scheme," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-14, no. 4, pp. 558–563, Jul. 1978.

[46] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459 – 474, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/000510989090018D

[47] A. Aribi, M. Aouni, C. Parges, S. Najari, P. Melchior, and M. Abdelkrim, "Generalied fractional obsevers scheme to fault detection and isolation," in *Proceedings of the 10th International Multi-Conference on Systems, Signals Devices (SSD)*, Mar. 2013, pp. 1–7.

[48] D. H. Trinh and H. Chafouk, "Fault detection and isolation using kalman filter bank for a wind turbine generator," in *Proceedings of the 19th Mediterranean Conference on Control Automation (MED)*, Jun. 2011, pp. 144–149.

[49] B. Boulkroune, M. Galvez-Carrillo, and M. Kinnaert, "Combined signal and model-based sensor fault diagnosis for a doubly fed induction generator," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1771–1783, Sept. 2013.

[50] T. Overbye, X. Cheng, and Y. Sun, "A comparison of the ac and dc power flow models for lmp calculations," in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, Jan. 2004, pp. 9 pp.–.

[51] M. Pipattanasomporn, M. Kuzlu, S. Rahman, and Y. Teklu, "Load profiles of selected major household appliances and their demand response opportunities," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 742–750, Mar. 2014.

[52] T. Tazoe, J. Matsumoto, D. Ishi, S. Okamoto, and N. Yamanaka, "Novel scheduling method to reduce energy cost by cooperative control of smart houses," in *2012 IEEE International Conference on Power System Technology (POWERCON)*, Oct. 2012, pp. 1–6.

[53] Z. Zhao, W. C. Lee, Y. Shin, and K.-B. Song, "An optimal power scheduling method for demand response in home energy management system," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1391–1400, Sept. 2013.

[54] A. Saha, M. Kuzlu, and M. Pipattanasomporn, "Demonstration of a home energy management system with smart thermostat control," in *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES*, Feb. 2013, pp. 1–8.

[55] A. Barbato, L. Borsani, A. Capone, and S. Melzi, "Home energy saving through a user profiling system based on wireless sensors," in *The first ACM workshop on embedded sensing systems for energy-efficiency in buildings*. ACM, 2009, pp. 49–54.

[56] B. Asare-Bediako, W. Kling, and P. Ribeiro, "Home energy management systems: Evolution, trends and frameworks," in *Universities Power Engineering Conference (UPEC), 2012 47th International*, Sept. 2012, pp. 1–5.

[57] H. Tischer and G. Verbic, "Towards a smart home energy management system - a dynamic programming approach," in *Innovative Smart Grid Technologies Asia (ISGT), 2011 IEEE PES*, Nov. 2011, pp. 1–7.

[58] N. G. Dlamini and F. Cromieres, "Implementing peak load reduction algorithms for household electrical appliances," *Energy Policy*, vol. 44, no. 0, pp. 280 – 290, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0301421512000778

[59] K. Dittawit and F. A. Aagesen, "Home energy management system for electricity cost savings and comfort preservation," in *The 4th IEEE International Conference on Consumer Electronics–Berlin, Berlin, Germany*, 2014.

[60] F. Corno and F. Razzak, "Intelligent energy optimization for user intelligible goals in smart home environments," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 2128–2135, 2012.

[61] D. L. Ha, H. Joumaa, S. Ploix, and M. Jacomino, "An optimal approach for electrical management problem in dwellings," *Energy and Buildings*, vol. 45, no. 0, pp. 1 – 14, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778811005731

[62] Z. Yu, L. Jia, M. Murphy-Hoye, A. Pratt, and L. Tong, "Modeling and stochastic control for home energy management," *IEEE Transactions on Smart Grid*, vol. 4, no. 4, pp. 2244–2255, Dec. 2013.

[63] "Power consumption table," http://www.absak.com/library/power-consumption-table, Jan. 2008.

[64] M. Lindquist and P. Wide, "New sensor system for drinking water quality," in *Proceedings of Sensors for Industry Conference*, 2004, pp. 30 – 34.

[65] A. Vaseashta, E. Braman, P. Susmann, Y. Dekhtyar, and K. Perovicha, "Sensors for water safety and security," in *Sensors Applications Symposium (SAS), 2011 IEEE*, February 2011, pp. 302 –307.

[66] X. Ma, Y. Song, J. Huang, and J. Wu, "Robust sensor placement problem in municipal water networks," in *2010 Third International Joint Conference on Computational Science and Optimization (CSO)*, vol. 1, May 2010, pp. 291 –294.

[67] W. Wu, J. Gao, M. Zhao, Z. qian, X. Hou, and Y. Han, "Assessing and optimizing online

monitoring for securing the water distribution system," in *2007 IEEE International Conference on Networking, Sensing and Control*, April 2007, pp. 350 –355.

[68] A. Ailamaki, C. Faloutos, P. S. Fischbeck, M. J. Small, and J. VanBriesen, "An environmental sensor network to determine drinking water quality and security," *SIGMOD Rec.*, vol. 32, pp. 47–52, December 2003. [Online]. Available: http://doi.acm.org/10.1145/959060.959069

[69] S. Amin, X. Litrico, S. S. Sastry, and A. M. Bayen, "Stealthy deception attacks on water scada systems," in *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, ser. HSCC '10, New York, NY, USA, 2010, pp. 161–170. [Online]. Available: http://doi.acm.org/10.1145/1755952.1755976

[70] M. Rafiee, A. Tinka, J. Thai, and A. Bayen, "Combined state-parameter estimation for shallow water equations," in *American Control Conference (ACC), 2011*, July 2011, pp. 1333 –1339.

[71] M. Rafiee, Q. Wu, and A. Bayen, "Kalman filter based estimation of flow states in open channels using lagrangian sensing," in *Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009.*, December 2009, pp. 8266 –8271.

[72] X. Litrico and V. Fromion, "Infinite dimensional modelling of open-channel hydraulic systems for control purposes," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 2, December 2002, pp. 1681 – 1686 vol.2.

[73] T. Vu, A. Baid, Y. Zhang, T. Nguyen, J. Fukuyama, R. Martin, and D. Raychaudhuri, "Dmap: A shared hosting scheme for dynamic identifier to locator mappings in the global internet," in *Proceedings of the 32nd International Conference on Distributed Computing Systems (ICDCS)*, 2012, pp. 698–707.

[74] J. Lindqvist and M. Gruteser, "Privacy in MobilityFirst Architecture," http://mobilityfirst.winlab.rutgers.edu/documents/documents/Lindqvist.pdf.

[75] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 482–494, 1998.

[76] S. Chen and M. Wu, "Anonymous multipath routing protocol based on secret sharing in mobile ad hoc networks," in *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, vol. 1, 2010, pp. 582–585.

[77] R. Lu, Z. Cao, L. Wang, and C. Sun, "A secure anonymous routing protocol with authenticated key exchange for ad hoc networks," *Computer Standards and Interfaces*, vol. 29, no. 5, pp. 521 – 527, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0920548906001358

[78] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Mask: anonymous on-demand routing in mobile ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 9, pp. 2376–2385, 2006.

[79] A. Singh and L. Liu, "Trustme: anonymous management of trust relationships in decentralized p2p systems," in *Proceedings of the Third International Conference on Peer-to-Peer Computing*, 2003, pp. 142–149.

[80] R. Koodli, V. Devarapalli, H. Flinck, and C. Perkins, "Short paper: Location privacy with ip mobility," in *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks, SecureComm*, 2005, pp. 222–224.

[81] L. Hao, S. Lu, J. Tang, and A. Zhang, "A low cost and reliable anonymity scheme in p2p reputation systems with trusted third parties," in *Proceedings of the Global Telecommunications Conference, GLOBECOM*, 2008, pp. 1–5.

[82] M. Gerlach and F. Guttler, "Privacy in vanets using changing pseudonyms - ideal and real," in *Proceedings of the 65th Vehicular Technology Conference, VTC*, 2007, pp. 2521–2525.

[83] M. K. Reiter and A. D. Rubin, "Crowds: anonymity for web transactions," *ACM Transactions on Information System Security*, vol. 1, no. 1, pp. 66–92, Nov. 1998. [Online]. Available: http://doi.acm.org/10.1145/290163.290168

[84] A. Kobsa and J. Schreck, "Privacy through pseudonymity in user-adaptive systems," *ACM Transactions on Internet Technology*, vol. 3, no. 2, pp. 149–183, May 2003. [Online]. Available: http://doi.acm.org/10.1145/767193.767196

[85] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, 2008.

[86] X. Wu and E. Bertino, "An analysis study on zone-based anonymous communication in mobile ad hoc networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 4, pp. 252–265, 2007.