Security Design against Stealthy Attacks on Power System State Estimation: A Formal Approach

Mohammad Ashiqur Rahman*

Florida International University, Miami, United States

Amarjit Datta

Tennessee Technological University, Cookeville, United States

Ehab Al-Shaer

University of North Carolina, Charlotte, United States

Abstract

State estimation is very important for securely, reliably, and efficiently maintaining a power grid. If state estimation is not protected, an attacker can compromise meters or communication systems and introduce false measurements, which can evade existing Bad Data Detection (BDD) algorithms and lead to incorrect state estimation. This kind of attack is stealthy and widely known as an Undetected False Data Injection (UFDI) attack. Attackers are limited by different attributes, in terms of knowledge, capabilities, resources, and attack targets, that are important to consider for realizing the potential attack vectors and, thereby, the security measures. In this paper, we present a formal framework for automatic synthesis of security architectures that will guard the grid against potential UFDI attacks on state estimation. In this approach, we first formalize UFDI attacks with respect to the grid topology, electrical properties, and different attack attributes. The solution to the model derives the attack vectors that exist in the formalized scenario. These attack vectors are considered by a second formal model, the security architecture synthesis model, to design the

☆This work was partially supported by National Science Foundation [grant number 165730].

*Corresponding author

Email address: marahman@fiu.edu (Mohammad Ashiqur Rahman)

security measures (*i.e.*, a set of measurements to be protected against false data injection). We enhance the performance of the security architecture synthesis mechanism by performing parallel and stepped execution of the formal models. We demonstrate the proposed mechanism through case studies and evaluate the performance of the proposed model by running experiments on different IEEE test systems.

Keywords: Smart Grid, False Data Injection, Security Architecture, Formal Model, Synthesis.

---

1. Introduction

State estimation is the process of approximating unknown state variables of a power system based on the measurements received from various meters strategically placed in the grid. The estimation process provides information pertinent to the power grid's condition, which is typically used in contingency analysis to control the power grid components and maintain the reliable operation even if some faults occur. The output is also utilized in optimal power flow management for economic efficiency. Cyber technology is increasingly used in power grids to efficiently and reliably control and monitor the power system [1]. The growing use of cyber infrastructure introduces numerous cyber-physical vulnerabilities in power grids [2, 3, 4]. It has been shown that cyber attacks against Supervisory Control and Data Acquisition (SCADA) systems can potentially cause significant damage to a power grid and its utilities [5]. Several researches explicitly show the ramification of the UFDI attacks on the power grid. The impact of UFDI attacks on the optimal operation of the grid is formally studied in [6, 7], where the results show that an adversary can increase the generation cost. How the UFDI attacks can compromise the energy markets is demonstrated in [8].

An adversary can compromise meters by introducing malicious measurements, which can lead to incorrect state estimation. There are Bad Data Detection (BDD) algorithms [9, 10] that detect bad measurements, principally based

on the squares of differences between observed and estimated measurements with some threshold values. If the differences are greater than the threshold values, the measurements are identified as bad data. Measurements are estimated based on the linear relationships among the measurements, potential noise, and the state variables.

An attacker can generate bad measurements and still remain undetected by evading the BDD algorithm [11]. As a result, states are estimated incorrectly. Stealthy attacks of this kind are known as Undetected False Data Injection (UFDI) attacks. However, an attacker usually must deal with different challenges (*i.e.*, attack attributes), such as limited capabilities, limited resources, specific targets, etc. Moreover, in order to launch a UFDI attack, an attacker needs to have necessary knowledge of the power grid (*i.e.*, connectivity among the buses and admittances of the connecting lines). Since the access to the grid information is usually restrictive, the attacker may suffer from limited information. Still, upon leveraging the relative associations of the states at the neighboring buses, one might be successful in launching a number of attacks despite limited information [12, 13]. Though previous works investigated UFDI attacks by considering some attack attributes, they did not consider a comprehensive list of attack attributes and, more importantly, interrelations among these attributes. Such an investigation requires simultaneous modeling of all these attributes. It is shown in [14, 15] that even when adversaries have perfect knowledge and capabilities, the grid can be defended against such UFDI attacks if a strategically chosen set of measurements is secured. The algorithms presented to identify such a measurement set were also shown to be equivalent to the NP-complete hitting set problem. Moreover, these algorithms are inflexible to consider various attack scenarios or the grid operators' resource limitation.

In this paper, we propose a formal framework for automatically synthesizing a security architecture (*i.e.*, a set of measurements that needs to be secured), with respect to an attack model, security requirements, and the grid operator's constraints. In our previous work [16, 17], we presented a formal model of identifying UFDI attacks on state estimation with respect to different attack

attributes. This model can be solved using an SMT (Satisfiability Modulo Theories) solver [18] to determine potential UFDI attacks. Here, we extend this verification model for generating attack vectors in a specific attack model and devise another formal model that employs these attack vectors to synthesize a list of measurements to be protected against data integrity breaches. The proposed model will allow a grid operator to take necessary security measures within his or her capabilities against adversaries with an expected set of attack attributes. We demonstrate the proposed security architecture synthesis model through a case study based on the IEEE 14-bus test system [19]. We evaluate the scalability of the proposed model by running experiments on various IEEE test systems. Since the exploration of the whole attack space for all potential attack vectors needs a significantly long time, we design an efficient mechanism to synthesize security architecture by performing parallel and stepped executions of the attack vector generation and security architecture synthesis models. The evaluation results exhibit high scalability of the synthesis mechanism.

The rest of this paper is organized as follows: In Section 2, we provide necessary background and our motivation. We present our security architecture synthesis framework in Section 4. We devise a scalable mechanism for efficient security architecture synthesis in 5. We briefly discuss the related work in Section 6. We conclude the paper with a discussion on the limitations of the presented study in Section 7, followed with the conclusion in Section 8.

2. Background and Motivation

The UFDI attack on state estimation according to the literature (*e.g.*, [11, 12]) is mainly based on the DC power flow model and we consider the same power flow model in this work.

2.1. DC Power Flow Model

In the DC power flow model, the power balance equations in a power system are expressed by assuming the impedance of a transmission line purely in terms

of its reactance [20]. The voltage magnitudes at all buses are taken as fixed at 1 per unit and only the phase angles are treated as the variables. Thus, the voltage phasor at bus i is expressed by $1\angle\theta_i$. Denoting the admittance of the line between buses i and j by Y_{ij}, the real power-flow (P_{ij}) across a transmission line is given by: $P_{ij} = Y_{ij}(\theta_i - \theta_j)$. Y_{ij} is the reciprocal of the reactance. The power-balance constraint that equates the algebraic sum of powers incident at every bus to zero creates a linear system of equations of the form: $[\mathbf{B}][\theta] = [\mathbf{P}]$.

2.2. State Estimation and UFDI Attacks

The state estimation problem involves estimating n number of power system state variables $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$ based on m number of meter measurements $\mathbf{z} = (z_1, z_2, \cdots, z_m)^T$ [10]. In the case of a linearized (*i.e.*, DC) estimation model, the relationship between x and z is given by the equation:

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{e}, \text{ where } \mathbf{H} = (h_{i,j})_{m \times n}$$

$\mathbf{H}$ is known as the Jacobian matrix, while $\mathbf{e}$ is the vector of measurement errors. When the errors are distributed with zero mean, the state estimate $\hat{\mathbf{x}}$ is given as:

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

Here, $\mathbf{W}$ is a diagonal matrix whose elements are reciprocals of variances of the meter errors. Thus, estimated measurements are calculated as $\mathbf{H}\hat{\mathbf{x}}$. Measurement residual $||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}||$ is used to determine bad data. If $||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}|| > \tau$, it is assumed that bad data is present. Here, τ is a selected threshold value.

In the case of a UFDI attack, an adversary injects arbitrary false data $\mathbf{a}$ to the original measurements $\mathbf{z}$ following the relation $\mathbf{a} = \mathbf{H}\mathbf{c}$, which leads the BDD mechanism to failure [11]. Here, $\mathbf{c}$ is the added value to the original state estimate $\hat{\mathbf{x}}$ due to the injection of $\mathbf{a}$. Since $\mathbf{z} + \mathbf{a} = \mathbf{H}(\hat{\mathbf{x}} + \mathbf{c})$, the residual $||(\mathbf{z} + \mathbf{a}) - \mathbf{H}(\hat{\mathbf{x}} + \mathbf{c})||$ is still equal to $||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}||$.

2.3. Attack Model

The attack potentiality varies with the attack scenario, which is specified based on different attack attributes, as follows:

- *Accessibility.* An attacker may not have access to all of the measurements because the physical or remote access to the power grid substations can be restricted. Moreover, some measurements might be secured by applying necessary data integrity protective measures.

- *Resource.* An adversary may be constrained by cost or effort to mount attacks on vastly distributed measurements. In such cases, an adversary is constrained to compromising or altering a limited subset of measurements at a time. Sometimes, it is useful to represent this resource limitation with respect to buses. This is because, if the measurements required for the false data injection in an attack are distributed in many substations, *i.e.*, buses, then it would be harder for an attacker to inject false data to those measurements than to the set of measurements distributed in a small number of substations.

- *Knowledge.* To launch a UFDI attack, an adversary needs to know the grid topology (*i.e.*, the connectivity among the buses) and the electrical parameters of the transmission lines [11]. Partial knowledge restricts the attacker's ability to plan for an undetected attack.

2.4. Objective: Security against UFDI Attacks

If these attack attributes are not considered simultaneously (as in [11, 12]), the interrelation between these attack attributes cannot be analyzed properly. We address this challenge by providing a formal analytic framework that measures the attackability on state estimation given various attack attributes. In addition, we propose a formal method for synthesizing a security architecture, *i.e.*, measurements that need to be secured (data integrity protected), satisfying the given security requirement. The requirement primarily specifies the protection of state estimation from UFDI attacks with respect to an attack model.

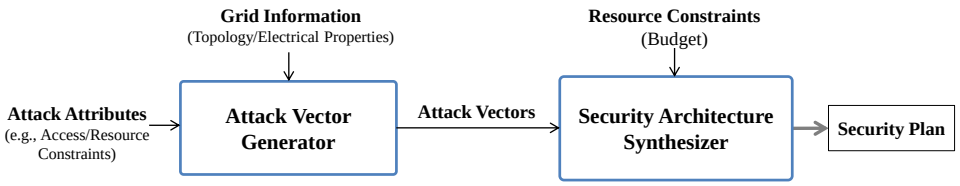

Figure 1: The security architecture synthesis framework to defend UFDI attacks against state estimation.

A few previous works (*e.g.*, [14, 15]) showed that UFDI attacks can be defended if a strategically chosen set of measurements is secured. However, they only considered the attack model where adversaries have unlimited knowledge, resources, or capabilities. Based on this worst-case scenario, the set of measurements to be secured may fall beyond the grid operator's resources. A large power grid usually has several hundreds of buses [21]. These buses are of widely distributed, often in remote areas. Moreover, the communication system and measurement devices still embrace legacy technologies, which often cannot support proper cryptographic measures [22]. Therefore, deploying proper security in a system with the legacy technologies and hardly-reached accessibility is non-trivial in terms of the existing technology capability as well as the deployment cost. That is why, the grid operator would love to keep the system (sufficiently) secured within a (limited) budget. Therefore, a security architecture within the grid operator's resource is required to defend UFDI attacks with respect to an expected level of attack potentiality.

3. Security Architecture Synthesis Framework

We present a schematic diagram of the security architecture synthesis framework in Figure 1. The framework in general has two modules. The first module, Attack Vector Generator, verifies a specified power grid for potential UFDI attacks in a particular attack model, thus generating all attack vectors that can successfully be launched. These attack vectors are taken as inputs to the second module, Security Architecture Synthesizer. This module is responsible for

synthesizing a security architecture, a list of measurements (or a set of buses) to be secured for mitigating the identified attack vectors, thus protecting the grid with respect to the attack model considered in the first module. The synthesis module also takes the grid operator's constraints (*e.g.*, the security deployment budget) as inputs so that the security architecture is designed within the operator's capacity. A security architecture specifies a set of measurements that needs to be data integrity protected so that false data cannot be injected in those measurements.

It is worth mentioning that each attack vector specifies a set of measurements that are required to be altered simultaneously during the attack campaign to remain undetected. There are many different combinations of the measurements to launch stealthy attacks. The security architecture synthesis framework needs the information of an attack vector to invalidate it. Therefore, to secure the bus system against stealthy attacks with respect to a set of attack attributes, the framework generates all the attack vectors in that attack model and provides a security architecture that mitigates the attacks according to the grid operators security requirements and resource constraints. The number of attack vectors is not an explicit input to the framework. The framework takes the bus system data and the attack attributes to generate the attack vectors.

In Section 4, we discuss the details of the framework's two modules, which primarily include formal modeling of various system properties, attack attributes, security requirements, and resource constraints. Since there are many alternative paths to launch a stealthy attack, a power grid with a few hundred buses can have a very large attack space that often cannot be handled by these formal models efficiently. In Section 5, we present a parallel and stepped approach of executing these formal models such that the framework can efficiently find a security architecture and has the capability to scale well with the grid size.

The security architecture synthesis is a proactive decision-making process. The framework is designed to run offline, allowing the grid operator to analyze possible security architectures in various attack scenarios, security requirements, and business objectives, and implement an appropriate security plan.

4. Modeling of Security Architecture Synthesis against UFDI Attacks

In this section, we present the proposed security architecture synthesis model, starting with a discussion about the basic physical (power) model and the modeling parameters. We also demonstrate the security architecture synthesis using an example case study and evaluate its performance.

4.1. Power Flow Equations

According to the DC power flow model, the admittance of a line or branch is computed from its reactance. Each line connects two end-buses and, according to the current flow direction, these two buses are identified as from-bus and end-bus. The two end-buses of line i are denoted using lf_i (from-bus) and lt_i (to-bus), where $1 \leq i \leq l$, $1 \leq lf_i, lt_i \leq b$, and b is the number of buses. The admittance of line i is denoted by ld_i.

Each row of $\mathbf{H}$ corresponds to a power equation. The first l rows correspond to the forward line power flow measurements. The second l rows are the backward line power flow measurements, which are the same as the first l except the directions of the power flows are opposite. We use P_i^L to denote the power flow through line i, while P_j^B to denote the power consumption at bus j, and θ_j to denote the state value (*i.e.*, the voltage phase angle at bus j). Then, we have the following relation between the line power flow of line i (P_i^L) and the states at the connected buses (lf_i and lt_i):

$$\forall_{1 \leq i \leq l} \quad P_i^L = ld_i(\theta_{lf_i} - \theta_{lt_i}) \tag{1}$$

Equation (1) specifies that power flow P_i^L depends on the difference of the connected buses' phase angles and the line admittance.

The last b rows of $\mathbf{H}$ correspond to the bus power consumptions. The power consumption of bus j is the summation of the power flows of the lines connected to this bus. Let $\mathbb{L}_{j,in}$ and $\mathbb{L}_{j,out}$ be the sets of incoming lines and outgoing lines of bus j, respectively. Then, the following equation represents the power

consumption at bus j:

$$\forall_{1\leq j\leq b} \quad P_j^B = \sum_{i\in\mathbb{L}_{j,in}} P_i^L \; - \sum_{i\in\mathbb{L}_{j,out}} P_i^L \tag{2}$$

Basically, state estimation with the DC flow model reduces to finding the voltage phase angle (θ) at each bus by solving an overdetermined linear system of equations given the measurement configuration and line parameters in a weighted least square sense as stated in Section 2.

4.2. UFDI Attack and Security Modeling Parameters

We use cx_j to denote whether state x_j ($1 \leq j \leq n$) is affected (*i.e.*, changed to an incorrect value) due to false data injection. Note that, in the DC model, each state corresponds to a bus. Thus, n is equal to b. Parameter cz_i denotes whether measurement z_i ($1 \leq i \leq m$) is required to be altered (by injecting false data) for the attack. If any measurement at bus j must be changed, cb_j becomes true.

In this work, we model incomplete information with respect to line admittance only and use the variable bd_i to denote whether the attacker knows the admittance of line i. In the DC model, two measurements can be taken (*i.e.*, recorded and reported by meters) for each line: the forward and backward current flows. For each bus, a measurement can be taken for the power consumption at the bus. Therefore, for a power system with l number of lines and b number of buses, there are $2l + b$ number of potential measurements (z_is). Though a significantly smaller number of measurements are sufficient for state estimation, redundancy is provided to identify and filter bad data. We use mz_i to denote whether potential measurement z_i is taken. Note that though m is often used to represent the taken measurements, in this model m represents the maximum number of potential measurements (*i.e.*, $2l + b$). The attacker may not be able to alter a measurement due to inaccessibility or existing security measures. We use az_i to denote whether measurement z_i is accessible to the attacker. We also use sz_i to denote whether the measurement is secured.

4.3. Modeling of UFDI Attack Vector Generation

4.3.1. Changes in States

First, we present the formalization of changes in estimated states. The attack on state x_j specifies that the phase angle at bus j is changed. This condition is formalized as follows:

$$\forall_{1 \leq j \leq n} \quad cx_j \rightarrow (\Delta\theta_j \neq 0) \tag{3}$$

From Equation (1), it is obvious that a change of P_i^L is required based on the changes in state x_{lf_i} (θ_{lf_i}) and/or state x_{lt_i} (θ_{lt_i}). In the case of false data injection, P_i^L, θ_{lf_i}, and θ_{lt_i} are changed to ${P'}_i^L$, ${\theta'}_{lf_i}$, and ${\theta'}_{lt_i}$, then Equation (1) turns into the following:

$${P'}_i^L = ld_i({\theta'}_{lf_i} - {\theta'}_{lt_i})$$

The subtraction of Equation (1) from the above equation represents whether there are changes in the measurements and the states. The following is the resultant equation:

$$\Delta P_i^L = ld_i(\Delta\theta_{lf_i} - \Delta\theta_{lt_i})$$

In this equation, $\Delta P_i^L = {P'}_i^L - P_i^L$, $\Delta\theta_{lf_i} = {\theta'}_{lf_i} - \theta_{lf_i}$, and $\Delta\theta_{lt_i} = {\theta'}_{lt_i} - \theta_{lt_i}$. If $\Delta\theta_{lf_i} \neq 0$ (or $\Delta\theta_{lt_i} \neq 0$), then it is obvious that state x_{lf_i} (or x_{lt_i}) is changed (*i.e.*, attacked). The above relation for line i holds only if the line is taken in the topology. We formalize this constraint as follows:

$$\forall_{1 \leq i \leq l} \quad ml_i \rightarrow (\Delta P_i^L = ld_i(\Delta\theta_{lf_i} - \Delta\theta_{lt_i})) \tag{4}$$

4.3.2. False Data Injection to Measurements

An attacker needs to alter a set of measurements to launch an attack. This false data injection depends on the changes that are required on different power flows or consumptions. If $\Delta P_i^L \neq 0$, then it specifies that measurements (*i.e.*, i and $l + i$) corresponding to line i, when taken (*i.e.*, mz_i and mz_{l+i}), are required

to change. Similarly, the power consumption measurement at bus j is required to change when $\Delta P_j^B \neq 0$. These are formalized as follows:

$$\begin{aligned} &\forall_{1\leq i\leq l}\ (\Delta P_i^L \neq 0) \rightarrow (mz_i \rightarrow cz_i) \wedge (mz_{l+i} \rightarrow cz_{l+i}) \\ &\forall_{1\leq j\leq b}\ \ (\Delta P_j^B \neq 0) \rightarrow (mz_{2l+j} \rightarrow cz_{2l+j}) \end{aligned} \tag{5}$$

Conversely, measurement i is altered, only if it is taken and the corresponding power measurement is required to change. The constraint is formalized as follows:

$$\begin{aligned} &\forall_{1\leq i\leq l}\ \ cz_i \rightarrow mz_i \wedge (\Delta P_i^L \neq 0) \\ &\forall_{1\leq i\leq l}\ \ cz_{l+i} \rightarrow mz_{l+i} \wedge (\Delta P_i^L \neq 0) \\ &\forall_{1\leq j\leq b}\ \ cz_{2l+j} \rightarrow mz_{2l+j} \wedge (\Delta P_j^B \neq 0) \end{aligned} \tag{6}$$

4.3.3. Attack Attributes

Now, we formalize the attack attributes that constrain the success of a particular UFDI attack.

Due to the resource limitation, the attacker can inject false data to a limited number of measurements simultaneously. If T_{CZ} is the maximum number, then:

$$\sum_{1\leq i\leq m} cz_i\ \ \leq T_{CZ} \tag{7}$$

Another way of modeling resource limitation is with respect to the number of compromised buses or substations. Due to limited resources, an attacker can only access or compromise a limited number of substations (*i.e.*, buses) at a particular time. A substation is required to be accessed or compromised if a measurement residing at that substation must be altered. Therefore,

$$\begin{aligned} &cz_i \rightarrow cb_{lf_i} \\ &cz_{l+i} \rightarrow cb_{le_i} \\ &cz_{2l+j} \rightarrow cb_j \end{aligned} \tag{8}$$

Let T_{CB} be the maximum number of substations that the attacker can compromise. Then,

$$\sum_{1 \leq i \leq m} cb_j \ \leq T_{CB} \tag{9}$$

If the admittance of line i is unknown to the attacker, then it is not possible for him to determine the necessary changes that he or she needs to make in the power flow measurements of the line. This condition is formalized as follows:

$$(\Delta P_i^L \neq 0) \rightarrow ((mz_i \vee mz_{l+i}) \rightarrow bd_i) \tag{10}$$

The attacker usually cannot, with respect to physical or remote access, inject false data to all the measurements. If a measurement is secured (*i.e.*, data integrity protected), then although the attacker may be able to inject false data in the measurement, the false data injection will be unsuccessful. Hence, the attacker will only be able to change measurement z_i (here, $1 \leq i \leq m$) in order to attack, if the following condition holds:

$$cz_i \rightarrow az_i \wedge \neg sz_i \tag{11}$$

4.3.4. Attack Vectors

Let $\mathcal{M}_{UFDI}$ be the UFDI attack verification model, which is the conjunction of Equations (3) through (11). The solution to this model (*i.e.*, when $\mathcal{M}_{UFDI}$ is true), the assignments to the variables, particularly czs and cxs, represent an attack vector (let it be A_k), which specifies that a set of states (Ax_k) can be attacked if a set of measurements (Az_k) can be altered.

$$Az_k \rightarrow Ax_k \tag{12}$$

That means:

$$\bigwedge_{i \in Az_k} cz_i \ \rightarrow \bigwedge_{j \in Ax_k} cx_j \tag{13}$$

To find the next possible attack vector, we have to add the following con-

straint that basically discards the current attack vector from the attack space:

$$\neg((\bigwedge_{i \notin Az_k} \neg cz_i) \wedge (\bigwedge_{i \notin Az_k} cz_i) \wedge (\bigwedge_{j \in Ax_k} cx_j) \wedge (\bigwedge_{j \notin Ax_k} \neg cx_j)) \tag{14}$$

Let $\mathcal{A}$ be the set of attack vectors possible from a given attack model. Therefore, $|\mathcal{A}|$ is the number of potential attack vectors, *i.e.*, $A_k \in \mathcal{A}$, where $1 \leq k \leq |\mathcal{A}|$. We need to consider all $|\mathcal{A}|$ to design a security architecture with respect to this particular attack model, which is discussed in the following subsection.

4.4. Modeling of Security Architecture Synthesis

To synthesize a security architecture against a particular attack model, each corresponding attack vector needs to be considered. We must also consider the grid operator's resource limitation.

4.4.1. Secured Measurements and States

As shown in Equation (13), an attack vector specifies that if an attacker can inject false data in a set of measurements, he or she can attack the estimation of a set of one or more states. However, if one or more of the measurements are secured, then this particular attack will fail. In other words, if none of these measurements is secured, these states are vulnerable to a UFDI attack. It is worth reminding the readers that a secured measurement means it is data integrity protected, which does not allow an attacker to inject false data into the measurement (*i.e.*, alter the measurement).

An attack vector A_k exists only if none of the measurements in Az_k is secured, which specifies that each of the states in Ax_k is vulnerable to a potential UFDI attack. As we use sz_i to specify whether measurement i is secured, we have the following constraint:

$$\forall_{1 \leq k \leq |\mathcal{A}|} \ \neg(\bigvee_{i \in Az_k} sz_i) \rightarrow \bigwedge_{j \in Ax_k} cx_j$$

A state is secured only if it is not under any attack. If sx_j specifies whether state j is secured, then:

$$\forall_{1 \leq j \leq n} \quad sx_j \rightarrow \neg cx_j$$

4.4.2. Security Requirements and Constraints

While the attack attributes define the scale of the UFDI attacks that the grid operator wants to deal with, the security requirement specifies to what extent the operator wants to defend them. In other words, the security requirement specifies the extent of security, which is defined as the minimum number of states that the operator needs to keep secured against UFDI attacks. If T_{SX} specifies this number, then:

$$\sum_{1 \leq j \leq n} sx_j \geq T_{SX}$$

When T_{SX} is equal to n (*i.e.*, the number of states), this security requirement is the strongest.

The operator may have a requirement of securing a set of states. For example, if states 1, 2, 3, 4, and 6 must be kept secured, then:

$$sx_1 \wedge sx_2 \wedge sx_3 \wedge sx_4 \wedge sx_6$$

Due to the resource constraint, the operator can secure only a limited number of measurements. If T_{SZ} specifies this number of measurements, then:

$$\sum_{1 \leq i \leq m} sz_i \leq T_{SZ}$$

If one or more measurements are already secured (and considered while looking for the attack vectors), they will be included in the number of measurements to be secured.

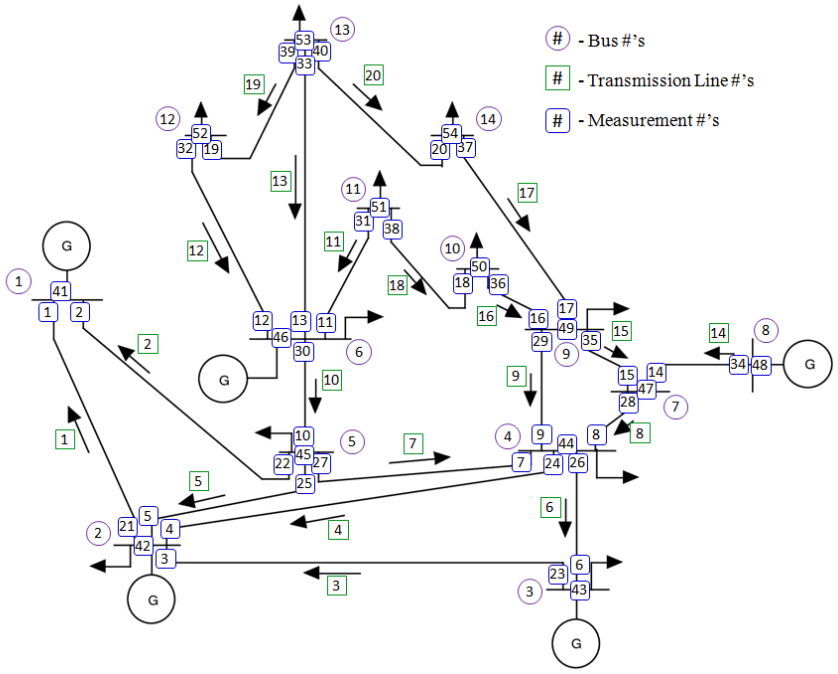

Figure 2: The diagram of the IEEE 14-bus test system.

4.5. Implementation

We encode the formalizations corresponding to our proposed UFDI verification model and security architecture synthesis model into SMT [23]. We write a program leveraging the *Z3 .Net API* [18] for this encoding. We use Boolean (*i.e.*, for logical constraints), integer (*e.g.*, mainly the configuration properties and thresholds), and real (*e.g.*, for power variables) terms for our encoding. For each of the model, a parser is built to take the inputs, *i.e.*, the system configurations and the constraints, to the model from a text file. By executing the model (in Z3), we obtain the verification (or synthesis) result as either satisfiable (*sat*) or unsatisfiable (*unsat*). The *unsat* results specifies that the problem has no attack vector (and/or security architecture) that satisfies the constraints. In the case of *sat*, we get the security architecture from the assignments to the variables, sz_is, which represent the set of measurements required to be secured.

Table 1: Bus System Input to the Example Case Study

```
# Topology (Line) Information
# (line no, from bus, to bus, admittance, knowledge?)
1 1 2 16.90 1
2 1 5 4.48 1
3 2 3 5.05 1
4 2 4 5.67 1
5 2 5 5.75 1
6 3 4 5.85 1
7 4 5 23.75 1
8 4 7 4.78 1
9 4 9 1.80 1
10 5 6 3.97 1
...  ...

# Measurement Information
# (measurement no, measurement taken?, secured?, can attacker alter?)
1 1 0 1
2 1 0 1
3 1 0 1
4 1 0 1
5 0 0 0
6 1 1 1
...  ...
21 1 0 1
22 0 0 0
23 1 0 1
24 1 0 1
25 1 0 1
26 1 0 1
...  ...
41 1 0 0
42 1 0 1
43 0 0 0
44 1 0 1
45 1 0 0
...  ...
```

4.6. A Case Study

In this example, we present a synthetic case study with respect to the IEEE 14-bus test system, as shown in Figure 2. The input corresponding to this ex-

Table 2: Adversary's and Operator's Constraints Input to the Example Case Study

```
# Attacker's Resource Limitation
10

# Operator's Security Requirement
# (This percentage of states needs to be secured)
100

# Operator's Resource Limitation
8
```

ample is shown in Tables 1 and 2. The input about the lines and measurements is partially shown due to space limitation. It is worth mentioning that measurement properties, attack attributes, and security requirements are arbitrarily selected in this case study and the studies presented latter of this paper.

The line information includes a set of data for each line: line number, end buses (from-bus and to-bus) of the line, a value indicating the line admittance, and the knowledge status. According to the inputs, the attacker has full knowledge about the system. Since this bus system has 14 buses and 20 lines, the maximum number of potential measurements is 54. For each of the measurements, the corresponding input includes (i) whether the measurement is taken for state estimation, (ii) whether the measurement is secured, and (iii) whether the attacker has the accessibility to alter the measurement. In this example, all measurements, except measurements {5, 10, 14, 19, 22, 27, 30, 35, 43, 49, and 52}, are taken. None of these measurements is secured. Due to limited resources, the attacker can attack only 10 measurements simultaneously. According to this attack model, the grid operator's security requirement is to defend all (*i.e.*, 100%) of the states against the UFDI attacks. However, the operator has limited resources that allow him to secure only a maximum of 8 measurements.

In this particular scenario of attack model, our UFDI attack vector generation model produces 85 different attack vectors. The security architecture synthesis model consequently provides a security architecture that mitigates all

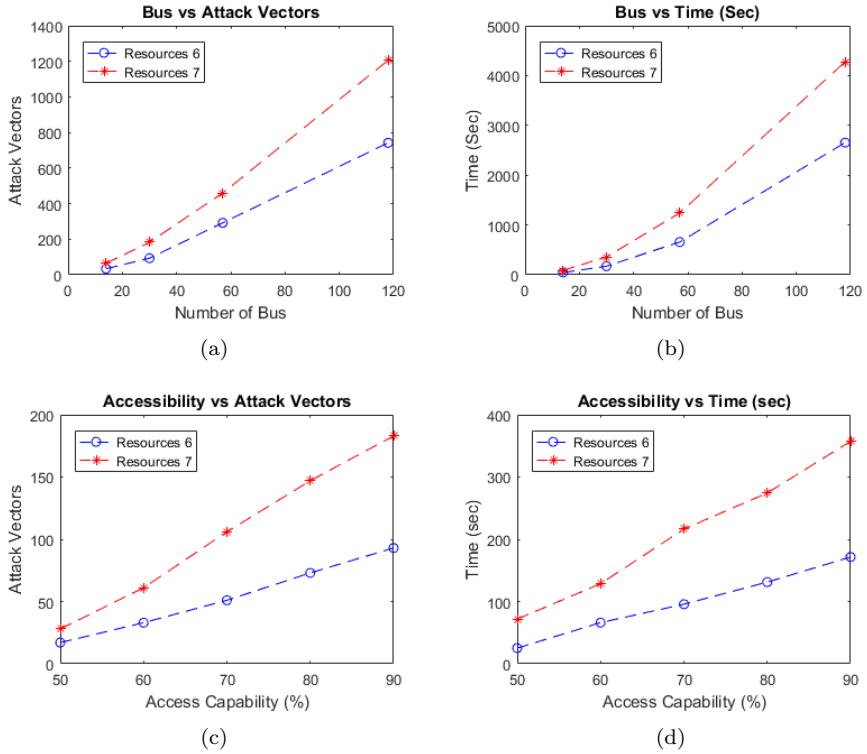

Figure 3: (a - b) The impact of the bus system's size on the number of attack vectors and the time to generate all these attack vectors and (c - d) the impact of the attacker's access capability on the number of attack vectors and the time to generate them.

of those attack vectors. According to this security solution, if measurements {8, 26, 34, 36, 41, 46, 53, and 54} can be secured, then no UFDI attack is possible in this given attack scenario. It is worth mentioning that if the grid operator cannot secure more than 6 measurements, then there is no security solution in this case.

4.7. Performance Analysis

In this section, we present the results from the evaluation of the proposed security plan synthesis approach. We evaluate the scalability of our proposed attack vector generation model as well as the synthesis model by analyzing the time required for executing the model in different problem sizes. Problem size

depends primarily on the number of buses. We evaluate the scalability of our model based on different sizes of IEEE test systems, such as 14-bus, 30-bus, 57-bus, and 118-bus [19]. We run our experiments on an Intel Core i7 Processor with 8 GB memory under Windows 7 OS.

4.7.1. Evaluation Results for Attack Vector Generation Model

When the attacker has higher ability (*i.e.*, access capability or attack resources), the number of potential attack vectors becomes larger. In such cases, if the system size (*i.e.*, the number of buses in the system) is also large, the number of attack vectors becomes enormously high. Figure 3(a) shows the number of attack vectors varying the number of buses when there is no secured measurement. Figure 3(c) shows how greater access capability generates a larger number of attack vectors, with respect to the 30-bus system.

The time to generate all the attack vectors is proportional to the number of these vectors. That is, the larger is the number of attack vectors, the longer the time required for generating them. Figure 3(b) shows the time to generate all the attack vectors corresponding to Figure 3(a). The results demonstrate a super-linear growth in time with respect to the number of vectors. Similarly, Figure 3(d) shows the time to generate all the attack vectors according to the attacker's access capability (Figure 3(c)). The time requirement is linear with respect to the access capability.

4.7.2. Evaluation Results for Security Architecture Synthesis Model

We evaluate the scalability of our security architecture synthesis model by varying the number of buses. In Figure 4(a), we present the execution time of the synthesis model and find that the time requirement is linear with respect to the number of buses. In fact, the time is proportional to the number of attack vectors and the size of the attack vector. The overall time for generating the security design, *i.e.*, attack vector generation model plus security architecture synthesis model, is presented in Figure 4(b). We see that the time lies between the quadratic and linear orders. However, a security architecture and the time

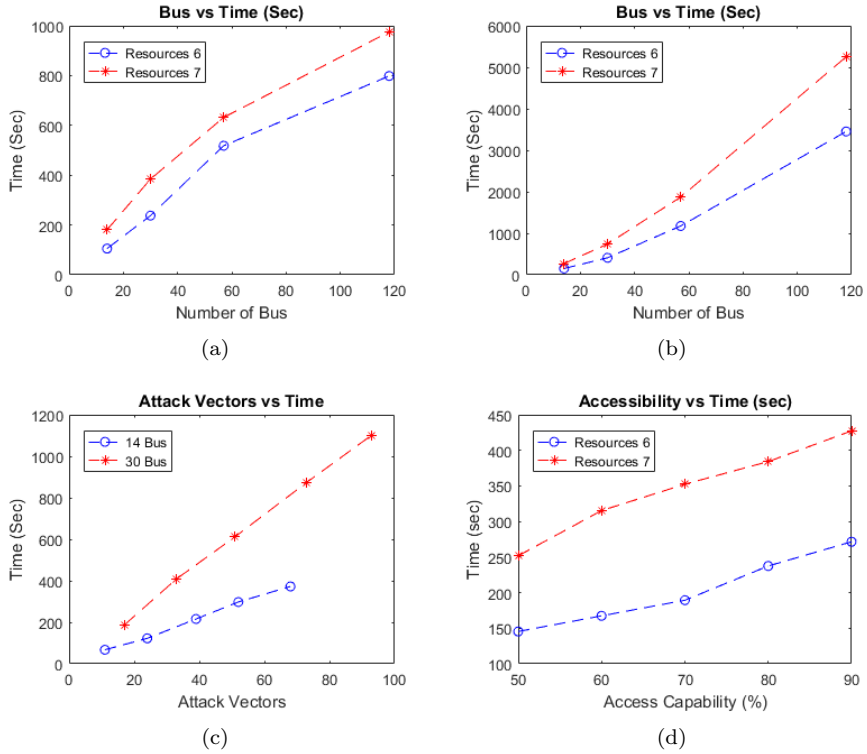

Figure 4: (a) The time to generate a security architecture (or attack mitigation plan) with respect to the number of buses, (b) the overall time to generate the security deployment architecture, (c) the impact of the number of attack vectors on the synthesis time, and (d) the impact of the grid operator's resource on the security architecture synthesis time.

to synthesize this depend on the given constraints, *e.g.*, the grid operator's resource limit. Figure 4(c) presents how the number of attack vectors impacts the security architecture synthesis time. As we can see, the execution time follows a linear growth with the number of attack vectors.

We analyze the impact of the grid operator's resource limit (for security deployment) on the security architecture synthesis time. The analysis result is shown in Figure 4(d). We observe that the synthesis time decreases rapidly with the increase in the resources because increasing the resources also increases the solution space (*i.e.*, the number of security architectures satisfying the security requirements) also increases. As a result, the time to search a security architec-

Algorithm 1 Exploration of the Attack Space by Each MPI Process

```
1: F := The attack vector synthesis model according to the input;
2: R := A random number to be used as the seed for exploring the attack space;
3: Done = FALSE;
4: 𝕊 = { };
5: while ¬Done do
6:     Result = findAttackVector(F, R);
7:     if Result ≠ NULL then
8:         shareResultWithOtherProcesses(Result);
9:         Ŝ = {Result} ∪ getResultsFromOtherProcesses();
10:        𝕊 = 𝕊 ∪ Ŝ;
11:        for each E ∈ Ŝ do
12:            F = F ∧ ¬E
13:        end for
14:    else
15:        Done = TRUE;
16:    end if
17: end while
```

ture decreases. However, if the available resources increase further, after some point, there may not be any more improvement.

5. Scalable Mechanism Design for Security Architecture Synthesis

As Figure 4(b) shows, the overall time for security architecture synthesis is large − over an hour just for 118 buses − even in a resource-constrained attack scenario. In a relaxed scenario, when an adversary has more capabilities, the attack space is much larger and the time to compute the attack vectors and to synthesize the security architecture significantly increases. Therefore, we propose mechanisms to explore the attack space in a scalable manner and thus efficiently synthesize the security architecture.

5.1. Parallelism for Efficient Attack Vector Generation

We develop a parallel mechanism to accelerate the generation of the attack vectors. The mechanism executes multiple MPI (Message Passing Interface) processes simultaneously and all these processes explore the attack space together. Each process runs the UFDI attack verification model with a random seed such that SMT searches in an arbitrary order. When a process gets an

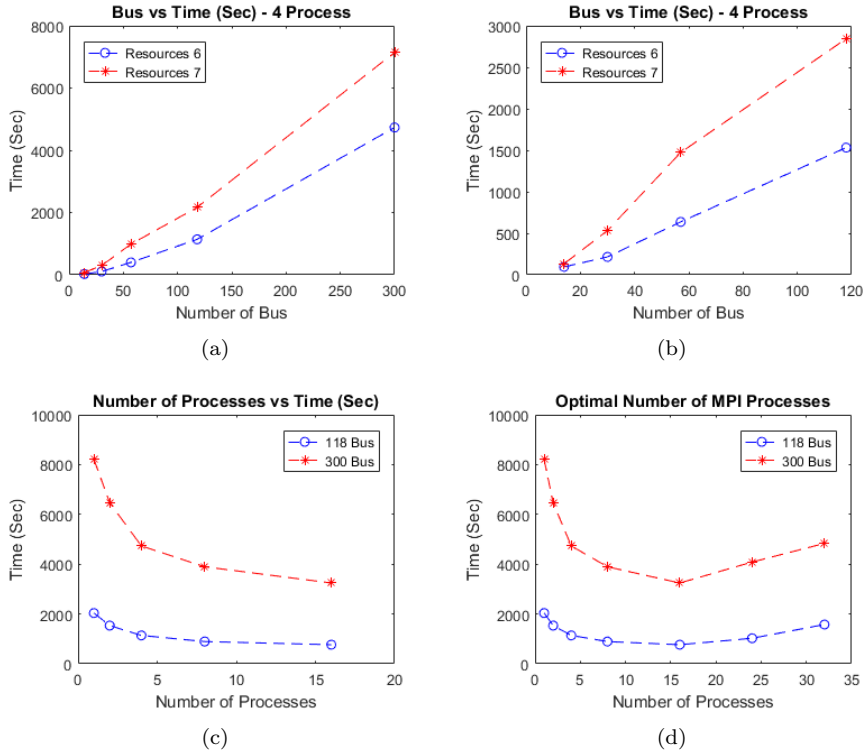

Figure 5: (a) The time for generating all the attack vectors when parallelism is applied (with respect to the bus system's size), (b) the overall time for security architecture synthesis, (c) the impact of the number of processes for generating all the attack vectors when parallelism is applied, and (d) the optimal choice of the number of processes to generate all the attack vectors.

attack vector, it informs other processes about the result so that the processes can exclude the attack vector from the search space. In this way, the attack vector search continues until each of the processes ends (when the verification model returns *unsat*, *i.e.*, a NULL attack vector). The corresponding set of operations executed by each process is presented in Algorithm 1.

5.1.1. Performance Analysis of Parallel Attack Space Exploration

We evaluate the efficiency of the parallel mechanism of the attack vector generation and the time for the security architecture synthesis. Figure 5(a) shows the required time for generating all the attack vectors with respect to the

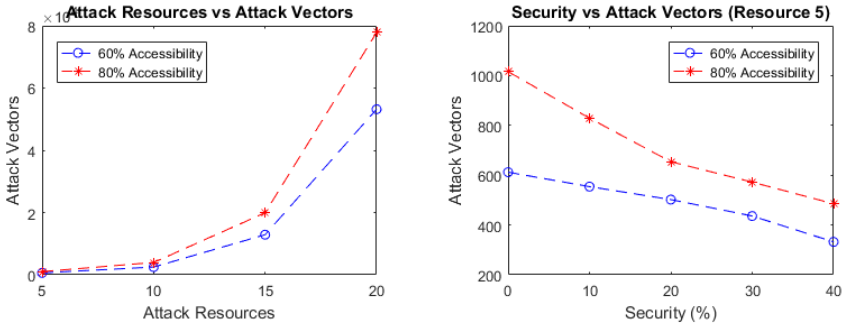

Figure 6: (a) The impact of the attacker's resource and (b) the impact of the security measurements on the number of attack vectors.

bus system's size. As we compare the result with that shown in Figure 3(b), the parallel mechanism reduces the attack vector generation time significantly, *e.g.*, less than half in the case of the 118-bus system, when we utilize only four parallel MPI processes. Even for the IEEE 300-bus system, it takes just over an hour. Figure 5(b) shows the total time for synthesizing the security architecture. As the use of parallelism significantly reduces the time for generating the attack vectors, it ultimately increases the time-efficiency of the synthesis process.

In the above experiments, four MPI processes execute in parallel to generate all the attack vectors. Figure 5(c) presents the impact of the number of MPI processes on the generation time. As the figure shows, the more is the number of processes, the shorter the execution time. However, the MPI processes need to communicate with one another for coordination, resulting in a cost, along with the overhead associated with executing multiple processes. Hence, there is a trade-off between the efficiency and the overhead. Figure 5(d) shows the impact of the number of processes on generating the attack vectors with respect to two problem sizes (*i.e.*, 118-bus and 300-bus systems). As the figure shows, while the time-efficiency keeps increasing with the number of processes, after some point, the efficiency reduces as the number of processes increases. Hence, there is an optimal choice for the number of processes for a problem size.

Table 3: Number of Attack Vectors in Different Scenarios

Attacker's Resource Limitation	**Attack Vectors**
10	85
12	358
14	1217
16	4314
18	13278
20	41802

5.2. Stepped Mechanism for Efficient Security Design

Although parallelism significantly improves the scalability of the security architecture synthesis, we can further improve the performance. The attack space depends on the attacker's resource − the more measurements the attacker can corrupt at a time, the more different attacks are possible. Figure 6(a) shows the impact of the attacker's resource on the number of attack vectors, for the IEEE 118-bus system. In this case, no secured measurement is considered. The results show that the number of attack vectors increases rapidly with the increase in the attacker's resource. For example, while there are 2,475 attack vectors at the resource limit of 10 measurements (in 60% accessibility), there are 12,870 attack vectors at the resource limit of 15 measurements. That is, for an increase of 5 measurements, the number of attack vectors increases by more than four times. The impact is larger when the accessibility is higher (80%). Table 3 shows the same for the IEEE 14-bus system when the attacker has 100% accessibility and there is no secured measurement. Although there are only 14 buses, the number of attack vectors increases from 4,314 (at the resource limit of 16 measurements) to 13,278 attack vectors for a resource increase of only 2 measurements. Figure 6(b) shows the impact of the secured (*i.e.*, data integrity protected) measurements on the attack space. Here the secured measurements are arbitrarily selected. As the figure shows, the more is the number of secured measurements (*i.e.*, reduced accessibility), the less the number of attack vectors. The reason for this is obvious − the secured measurements cannot be altered to launch stealthy attacks.

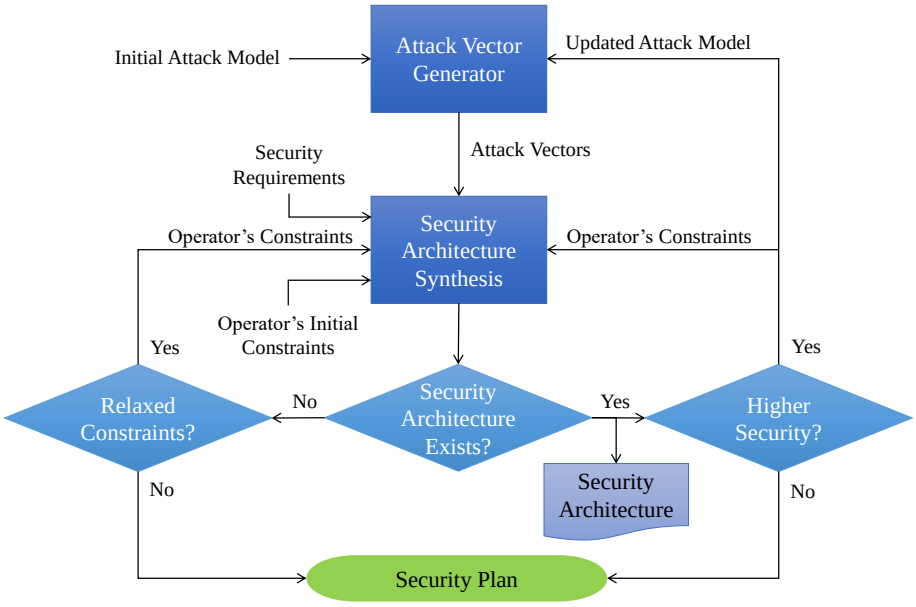

Figure 7: The flow chart of the stepped security design mechanism, especially to scale with large power grids.

Exploiting the resource and accessibility attack constraints, we devise a stepped design mechanism as shown in Figure 7 to find security architectures for larger problems in a scalable manner. The proposed mechanism includes the following steps:

1. Initially, an attack model is taken wherein the adversary's attack capabilities are smaller. In this case, the attacker's resource limitation is basically considered.

2. We find all potential attack vectors considering this specific attack scenario using the UFDI attack verification model (refer to Section 4.3).

3. According to these attack vectors, the synthesis model (refer to Section 4.4) is solved for a security architecture that includes a list of measurements that need to be secured to protect the system from these attack vectors Here, we may consider the operator's resource size smaller than

the original input, if there are further iterations required to grow the extent of the attack attribute(s) up to the expected attack scenario. If there is no security solution, we need to increase the resource size (i.e., relax the constraints) slowly until it reaches the operator's maximum capability.

4. If we find a security architecture for the expected, not the intermediate, attack model, then the process is finished. Otherwise, we update the attack model by increasing the attacker's resources and setting a smaller, but random, set of measurements taken from the security architecture as secured. Then, we return to Step 2 and continue the process.

It is worth mentioning that the design mechanism described above cannot ensure its completeness. That is, if the mechanism cannot find a security architecture within the grid operator's resources, it cannot conclude that there is in fact no solution within that resource limitation. In Step 4 of the mechanism, we initialize the grid with a part of the security architecture (*e.g.*, a set of measurements) that is synthesized in the previous cycle when the attacker has fewer resources. The reason of considering such a set as the given security is to decrease the search (solution) space, thus reducing the synthesis time. This initialization can end up with no security architecture, which might otherwise be found if a different subset were chosen or the whole solution space would be searched. This situation may occur only when the security architecture in the previous step (at fewer attack resources) is designed with more than necessary measurements, and all or some of these extra measurements are not in the optimal security plans. If the selected subset includes some of these redundant measurements and the operator does not enough resources to add necessary measurements to create a satisfiable security plan, the mechanism will fail. However, if the security architecture is provided in the previous step is an optimal one, then the mechanism will always be able to find a solution irrespective of the selected subset. This is because the attack vectors possible at a number of the attacker's resources is a subset of the attack vectors possible at an increased number of resources. Since at each step the security architecture synthesis model ensures

satisfiability, not the optimality (which is an NP-complete process [14, 15]), we may need to sacrifice some completeness to substantially increase the efficiency of synthesizing a security architecture.

5.3. Case Study

In this example, we consider the same IEEE 14-bus test system and the same input, except larger attack capabilities. The attacker has full knowledge about the system; all measurements, except measurements 5, 10, 14, 19, 22, 27, 30, 35, 43, 49, and 52, are taken; the attacker has access to all taken measurements; and none of the measurements are secured. Unlike the first example, the attacker has more resources to attack 30 measurements simultaneously. With respect to this attack model, the grid operator's security requirement is to use the resources to secure 100% of the states against any UFDI attack. The resource limitation of the grid operator allows a maximum of 13 measurements to be secured. In this case, the generation of all attack vectors takes over 2 hours without parallelism, but only 24 minutes when parallelism is applied. The synthesis process takes 5 minutes. Next, we apply the proposed design mechanism for the same security architecture synthesis. Here, the control steps are iterated for three times.

In the first step, we consider that the attacker can only attack 12 measurements and the grid operator can secure 8 measurements. The number of UFDI attack vectors received is 358 and the generation model produces them in 6 seconds. The corresponding security architecture is found as measurements {12, 17, 18, 34, 42, 44, 45, and 46} and the architecture is synthesized in 2 seconds. In the next iteration, we consider measurements {42, 44, 45, and 46}, an arbitrary subset of the security architecture found at the last iteration, as secured, while the attacker's capability as maximally 22 measurements for simultaneous alteration. Now, we receive 3511 attack vectors (in 90 seconds), while the security architecture as measurements {3, 15, 16, 40, 42, 44, 45, 46, 48, 51, and 53} (in 3 seconds). In this security architecture synthesis, we consider that the grid operator can secure a maximum of 11 measurements, which also includes the assumed secured measurements (*i.e.*, {42, 44, 45, and 46}).

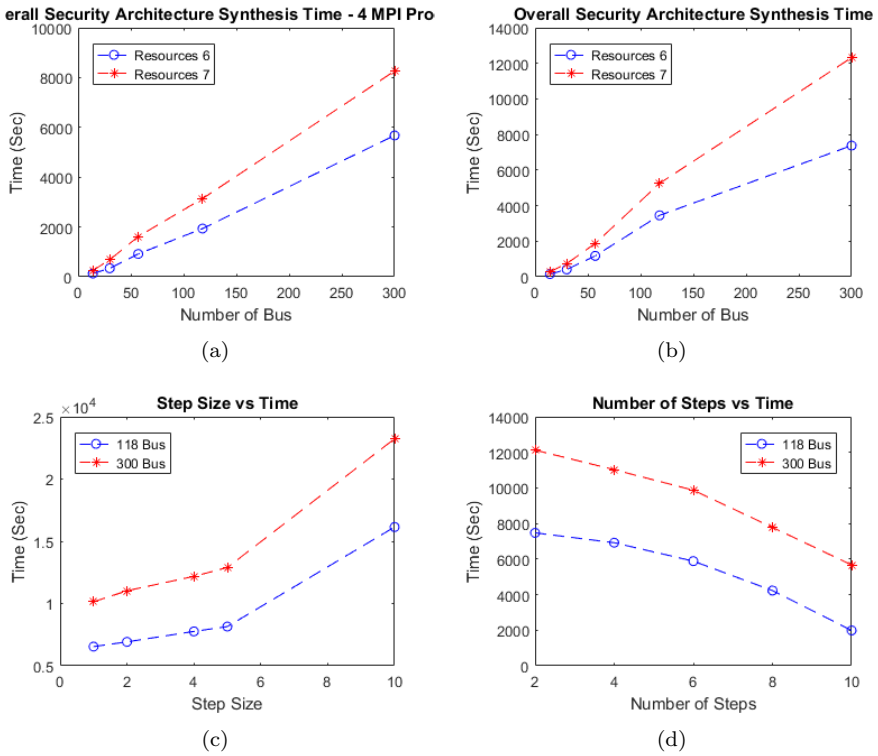

Figure 8: These graphs show the security configuration synthesis time in different experiments: (a) the execution time when parallelism is applied, (b) the execution time without the application of parallelism, (c) the impact of the step size on the execution time, and (d) the impact of the number of steps (initial resource size to start with) on the execution time.

In the last iteration, we consider an arbitrary subset of the security architecture, measurements {16, 40, 42, 44, 45, 46, 48, and 51}, as secured. We also consider that the attacker can alter a maximum of 30 measurements simultaneously, while the grid operator can secure a maximum of 13 measurements (including the assumed secured measurements). In this scenario, we find 1480 attack vectors (in 42 seconds). According to these attack vectors, the assumed secured measurements, and the operator's resource limitation, we get the following security architecture: measurements {1, 4, 16, 33, 40, 41, 42, 44, 45, 46, 48, 51, and 54} (in 1.5 seconds). We can see that the design mechanism takes less than 180 seconds (3 minutes) in total.

5.4. Performance Analysis of Stepped Security Design Mechanism

We further evaluate the scalability of the proposed stepped security design mechanism by analyzing the time required for executing the model in different problem sizes. The model is evaluated based on the same IEEE test systems as before [19].

Figure 8(a) presents the execution time of the multi-step security design mechanism varying the problem size (*i.e.*, the number of buses). We see that the time follows a linear order. In order to generate the attack vectors, parallelism is applied here utilizing 4 MPI processes. We also evaluate the performance in the case when no parallelism is used. The result is presented in Figure 8(b). We can see that the execution time linearly grows with the increase in the number of locations and the required time is significantly shorter than that we have seen previously in Figure 4(b). However, as it is expected, the required time is higher than that in the parallel case (Figure 8(a)).

We analyze the impact of the step size (*i.e.*, the increase in the attacker's resource at each step) on the security architecture synthesis time. The evaluation result is shown in Figure 8(c). We observe that the synthesis time increases as we increase the adversary's resources (*i.e.*, the number of buses the adversary can compromise at a time). This is because the more the resource size increases, the larger the attack space extends (with respect to the attack space at the previous resource size). As we can see in Figure 8(d), if we take a larger number of steps (*i.e.*, a lower initial resource size results in a larger number of steps), the execution time decreases. In this case, although the multi-step mechanism takes more rounds to get the security architecture, it saves a significant time to execute when the starting resource size is smaller. It is worth mentioning that if the starting resource size is too low, then the time to executing many rounds will increase the executing time. Therefore, there is an optimal choice for the starting resource size.

6. Related Work

We first present the literature review on UFDI attacks. Then, we discuss existing works that address the problem of securing the power system against these attacks.

6.1. UFDI Attack Analysis

Although cyber vulnerabilities of power grids have been discussed in literature for over a decade [24, 25], cyber security analysis of power systems has revolved largely around the concept of stealthy attacks, mostly known as UFDI attacks. The concept follows from the DC power flow model and was first presented in [11], and extended later in [26]. The authors discussed the UFDI attacks considering different scenarios, such as limited access to meters and limited resources to compromise meters, under random and specific targets, assuming that the adversary has complete information about the grid. Generally, the attack vector computation problem is NP-complete. Therefore, the authors presented few heuristic approaches that can find attack vectors.

Vukovic et al. proposed a number of security metrics to quantify the importance of individual buses and the cost of attacking individual measurements considering the vulnerability of the communication infrastructure [27]. Kin Sou et al. claimed in [28] that an l_1 relaxation-based technique provides an exact optimal solution of the data attack construction problem. The notion of unidentified attacks is presented in [29], where the grid operator can detect the existence of bad data, but cannot identify the bad measurements specifically. A different kind of cyber attack in power grids, namely the load redistribution attack, is introduced in [30].

UFDI attacks with incomplete or partial information (*i.e.*, partial knowledge of the bus system with respect to electrical properties of the transmission lines) are discussed in [12, 13]. It was also shown in [31] that an adversary can launch UFDI attacks even with no prior knowledge of the topology. The idea of the authors is to estimate the linear structure of the topology from the measurements and then launch UFDI attacks based on the estimated topology. The

load redistribution attack with respect to the attacker's incomplete knowledge is discussed in [32]. Algebraic conditions of undetected topology attacks in power grids are identified in [33] although these conditions do not coordinate the typical UFDI attacks. A modeling of the game between an attacker and a defender with respect to the impact of UFDI attacks on energy markets is presented in [8]. However, none of the works discussed above provide a comprehensive model of UFDI attacks that considers different attack attributes combined together.

Therefore, we proposed a new technique of logically verifying UFDI attacks. First, in [16, 6], we presented SMT-based formal verification models for the attacks with respect to a list of attack constraints/attributes and considered their impact on the Optimal Power Flow (OPF) solution. Then, in [17, 7], we introduced stealthy attacks through the novel idea of strengthening UFDI attacks by incorporating topology poisoning. Our models are comprehensive and extensible. Later, this idea of logically modeling the UFDI attacks was adopted in [34] and was used to analyze nonlinear (AC) state estimation-based false data injection attacks. Some other existing works also consider nonlinear state estimation. Zhang et al. analyzed the impact of the topology attacks on the nonlinear power system operations [35]. False data attacks against the nonlinear state estimation are discussed in [36, 37] with respect to the incomplete knowledge of the system. Chakhchoukh and Ishii discussed stealthy attacks on state estimation considering the possible presence of Phasor Measurement Units (PMUs) [38].

6.2. Defense against UFDI Attacks

Different mechanisms are proposed to defend the power system against UFDI attacks. Bobba et al. showed that for detecting UFDI attacks, it is necessary and sufficient to protect a set of basic measurements, which correspond to the minimum set of measurements needed to ensure observability [14]. Kim and Poor proposed a greedy suboptimal algorithm, which selects a subset of measurements that can be made immune from false data injection for protection against UFDI attacks in [15]. An efficient algorithm is designed in [39] to find

all sparse UFDI attacks and determine countermeasures. Yang et al. developed a greedy algorithm, based on the least-effort attack model, for determining optimal PMU placement to defend against data integrity attacks [40]. Kosut et al. proposed mechanisms based on the generalized likelihood ratio test to detect UFDI attacks by comparing measurements with expected ones [41, 42] . Hunag et al. proposed a similar approach for defending state estimation in [43] with the help of an adaptive cumulative sum control chart test. The proposed mechanism is extended in [44] for real-time detection of the attacks. Zhu and Wei also proposed a defense technique by checking the consistency between the real-time running states obtained from the state estimation and the most likely running states derived from a historical running-state database [45]. Deng et al. presented an approach to designing a low cost defense strategy that would protect power systems against false data injection attacks by considering the cost variability in protecting the meters [46].

However, the above-mentioned defense mechanisms do not consider the attacker's attributes and hence, the derived security design cannot provide a cost-effective security design that considers attackers' practical capabilities. In [17], we have provided a candidate-based two-step mechanism, leveraging our formal UFDI attack verification model, to find a security plan that will protect the grid against stealthy attacks. However, this ad-hoc mechanism is not suitable for security plan synthesis for larger bus systems as it needs to run the attack verification model many times for arbitrary security architecture candidates. In this work, we provide an automated security architecture synthesis mechanism that uses the potential UFDI attack vectors. We extend our attack verification model to generate the attack vectors. Unlike prior works, our framework is flexible with respect to an attack model and also comprehensively considers the grid operator's resource constraints. It is worth mentioning that although we particularly focus on measurement-based UFDI attacks [6, 16] in this work, the proposed security architecture synthesis framework is generic enough for application to other stealthy attacks like topology poisoning-based UFDI attacks [7, 17].

7. Discussion

In this section, we discuss some limitations, intriguing aspects, and future directions of the study.

7.1. Nonlinear Power Flow Model

In this study, we consider the DC power flow model that is linearized by decoupling the voltage dynamics and neglecting resistive losses. In our previous works [16, 17], we demonstrated how formal methods can efficiently identify potential UFDI attack vectors within an extent of adversarial capacity. Later, a similar formal approach is applied for the stealthy attack analysis in the AC power flow model-based power system [36, 37]. In this paper, we focus on providing an efficient approach for synthesizing necessary security architectures, which does not depend on the power flow model, but rather on the identified attack vectors. In other words, although we consider the DC power flow model-based attacks to design the defense architecture, the approach is generic enough to consider the AC power flow-based attacks to find a mitigation plan.

7.2. Critical Attack Vectors

In this study, we exclusively look for potential UFDI attack vectors, where each attack vector represents a set of measurements that can be corrupted to change the state estimation result without being detected by the existing weighted least squares-based BDD algorithm. A reader may be interested about the impact of a stealthy attack, *e.g.*, given a number of attack vectors, can the attacker actually do anything interesting with those? We addressed such impact analysis in a different study [6, 7], where we proposed formal frameworks, along with systematic approaches, to identify critical attack vectors that can impose a desired increase in the electric power generation cost. We showed that while there are many stealthy attack vectors, there can be a few that can increase the generation cost to a desired extent (*e.g.*, a minimum of 5% increase in the generation cost).

7.3. Knowledge Uncertainty

As we discussed in Section 2, an adversary needs to know the electrical parameters (*i.e.*, admittance) of the transmission lines to launch UFDI attacks. Although partial knowledge restricts the ability to plan for a stealthy attack, there are still attack opportunities for the adversary. We define the "partial" knowledge as knowing the admittance values of a subset of the transmission lines. There can be an issue of uncertainty about the admittance value of a transmission line. In our proposed modeling, if the adversary is uncertain about an admittance value, we assume that that value is unknown. While our model is extendable enough to consider uncertainty as a parameter, connected with the knowledge about the admittance, to identify attacks with certain probabilities, we focus on providing a framework to detect potential threats in certain but flexibly configurable attack scenarios.

7.4. Power Grid Architecture and Security Design

The power grid topology, which includes the connectivity among the buses and the placements of the measuring devices/meters, has an impact on the number/set of valid attacks, and the number/placements of secured measurements, thus on the time to generate a security architecture. Since the power grid transmission structure depend on the generation and distribution centers, there is a mere chance of modifying the topology to impact the attack vectors, so the security architecture. Moreover, the required budget for restructuring the topology will not easily allow such a modification. However, within a limited budget, the grid operator may be able to restructure the measurement points (and/or deploy some extra measurement meters/sensors) to optimize the security architecture. In this work, we design the security architecture considering that the given power grid architecture is fixed. Analyzing the impact of the architecture on the system's security will be a future direction of this research.

7.5. Real-World Power Grids and Scalability

A real-world power grid, as found in the literature, has buses from a range of several hundred to a few thousand [21, 47, 48]. As we apply parallelism to

generate the attack vectors, it can easily scale with the grid size if necessary computing power is available. Using our limited computing power (only Intel i7 1.8 GHz dual-core processor with 16GB memory, 4 MPI parallel processes), the overall security analysis takes just over an hour for a 300-bus system. With a higher computing capability, in terms of the number of cores, the processing power of the processor, and the size of the memory, allowing a higher number of MPI processes, the analysis time will reduce further. The security architecture synthesis is not a run-time task, rather it is a proactive, long-term decision-making process. Therefore, the proposed security architecture synthesis framework is designed to run offline. This tool allows the grid operator to analyze for an appropriate security plan. The grid operator can easily allow some time for exploring and mitigating the potential security breaks.

8. Conclusion

Securing state estimation against cyber-attacks is of paramount importance in maintaining the integrity of the power grid. In this paper, we have proposed a formal approach that can capture interdependency among attack attributes to find attack vectors and synthesize a security architecture that secures a set of measurements for immunity against identified UFDI attacks. We have also devised a stepped design mechanism to increase the scalability of the security synthesis approach. The scalability of the proposed approach is evaluated with experiments and case-studies on different IEEE test systems. Our results show that our approach can efficiently solve large problems. This work provides a basis for the development of cyber-security tools for modern power grids.

References

[1] A. Ipakchi, F. Albuyeh, Grid of the future, in: IEEE Power and Energy Magazine, 2009, pp. 52–62.

[2] A. Group, Security in the smart grid, `https://library.e.abb.`

com/public/93eaf4b5464f6002c1257a93002f7edb/3BUS094984_en_Security_in_the_Smart_Grid.pdf (2009).

[3] R. J. Campbell, Cybersecurity issues for the bulk power system, https://www.fas.org/sgp/crs/misc/R43989.pdf, Congressional Research Service (CRS) Report (2015).

[4] B. Gertz, FBI warns of cyber threat to electric grid, http://freebeacon.com/issues/fbi-warns-cyber-threat-electric-grid/, Washington Free Beacon (2016).

[5] D. Kundur, X. Feng, S. Liu, T. Zourntos, K. Butler-Purry, Towards a framework for cyber attack impact analysis of the electric smart grid, in: IEEE SmartGridComm, 2010, pp. 244 – 249.

[6] M. A. Rahman, E. Al-Shaer, R. Kavasseri, A formal model for verifying the impact of stealthy attacks on optimal power flow in power grids, in: ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), 2014.

[7] M. A. Rahman, E. Al-Shaer, R. Kavasseri, Impact analysis of topology poisoning attacks on economic operation of the smart power grid, in: International Conference on Distributed Computing Systems (ICDCS), 2014.

[8] M. Esmalifalak, G. Shi, Z. Han, L. Song, Bad data injection attack and defense in electricity market using game theory study, IEEE Transactions on Smart Grid 4 (1) (2013) 160–169.

[9] A. Monticelli, Network Topology Processing, Power Electronics and Power Systems, Springer US, 1999.

[10] A. Abur, A. G. Exposito, Power System State Estimation : Theory and Implementation, CRC Press, New York, NY, 2004.

[11] Y. Liu, P. Ning, M. Reiter, False data injection attacks against state estimation in electric power grids, in: ACM Conference on Computer and Communications Security (CCS), 2009, pp. 21–32.

[12] A. Teixeira, S. Amin, H. Sandberg, K. Johansson, S. Sastry, Cyber security analysis of state estimators in electric power systems, in: IEEE Conference on Decision and Control, 2010, pp. 5991–5998.

[13] M. Rahman, H. Mohsenian-Rad, False data injection attacks with incomplete information against smart power grids, in: IEEE Conference on Global Communications, 2012.

[14] R. Bobba, K. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, T. Overbye, Detecting false data injection attacks on dc state estimation, in: IEEE Workshop on Secure Control Systems, CPS Week, 2010.

[15] T. Kim, H. Poor, Strategic protection against data injection attacks on power grids, IEEE Transactions on Smart Grid 2 (2) (2011) 326 –333.

[16] M. A. Rahman, E. Al-Shaer, M. Rahman, A formal model for verifying stealthy attacks on state estimation in power grids, in: IEEE International Conference on Smart Grid Communications (SmartGridComm), 2013.

[17] M. A. Rahman, E. Al-Shaer, R. Kavasseri, Security threat analytics and countermeasure synthesis for state estimation in smart power grids, in: IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2014.

[18] M. Research, Z3 theorem prover, `https://github.com/Z3Prover/z3/wiki` (2012).

[19] U. of Washington, Power systems test case archive, `http://www.ee.washington.edu/research/pstca/` (2013).

[20] A. J. Wood, B. F. Wollenberg, Power Generation, Operation, and Control, 2nd Edition, Wiley, 1996.

[21] Power grid datasets, `https://github.com/ComplexNetTSP/ComplexNetWiki/wiki/PowerGrid-datasets`, last Accessed on December 24, 2018.

[22] S. Schrecker, Enhanced security for the power system edge, https://www.energy.gov/sites/prod/files/2017/07/f35/Intel_Security%20for%20Power%20System%20Edge_FactSheet.pdf, office of Electricity Delivery and Energy Reliability, Department of Energy.

[23] L. de Moura, N. Bjørner, Satisfiability modulo theories: An appetizer, in: Brazilian Symposium on Formal Methods, 2009.

[24] J. Salmeron, K. Wood, R. Baldick, Analysis of electric grid security under terrorist threat, IEEE Transactions on Power Systems 19 (2) (2004) 905–912.

[25] C. W. Ten, C. Liu, G. Manimaran, Vulnerability assessment of cybersecurity for scada systems, IEEE Transactions on Power Systems 23 (4) (2008) 1836–1846.

[26] Y. Liu, P. Ning, M. K. Reiter, False data injection attacks against state estimation in electric power grids, ACM Transactions on Information and System Security 14 (1) (2011) 13:1–13:33.

[27] O. Vukovic, K. C. Sou, G. Dan, H. Sandberg, Network-layer protection schemes against stealth attacks on state estimators in power systems, in: IEEE International Conference on Smart Grid Communications, 2011.

[28] K. C. Sou, H. Sandberg, K. Johansson, On the exact solution to a smart grid cyber-security analysis problem, IEEE Transactions on Smart Grid 4 (2) (2013) 856–865.

[29] Z. Qin, Q. Li, M. C. Chuah, Unidentifiable attacks in electric power systems, in: IEEE/ACM Third International Conference on Cyber-Physical Systems (ICCPS), 2012, pp. 193–202.

[30] Y. Yuan, Z. Li, K. Ren, Modeling load redistribution attacks in power systems, IEEE Transactions on Smart Grid 2 (2) (2011) 382–390.

[31] M. Esmalifalak, H. Nguyen, R. Zheng, Z. Han, Stealth false data injection using independent component analysis in smart grid, in: IEEE International Conference on Smart Grid Communications (SmartGridComm), 2011, pp. 244–248.

[32] X. Liu, Z. Li, Local load redistribution attacks in power systems with incomplete network information, IEEE Transactions on Smart Grid PP (99) (2014) 1–12.

[33] J. Kim, L. Tong, On topology attack of a smart grid: Undetectable attacks and countermeasures, IEEE Journal on Selected Areas in Communications 31 (7) (2013) 1294–1305.

[34] S. Gao, L. Xie, A. Solar-Lezama, D. Serpanos, H. Shrobe, Automated vulnerability analysis of AC state estimation under constrained false data injection in electric power systems, in: IEEE Conference on Decision and Control (CDC), 2015, pp. 2613–2620.

[35] J. Zhang, Topology attacks on power system operation and consequences analysis, arizona State University (Jun 2015).

[36] J. Zhang, Z. Chu, L. Sankar, O. Kosut, False data injection attacks on power system state estimation with limited information, in: IEEE Power and Energy Society General Meeting (PESGM), 2016, pp. 1–5.

[37] X. Liu, Z. Li, False data attacks against ac state estimation with incomplete network information, IEEE Transactions on Smart Grid 8 (5) (2017) 2239–2248.

[38] Y. Chakhchoukh, H. Ishii, Coordinated cyber-attacks on the measurement function in hybrid state estimation, IEEE Transactions on Power Systems 30 (5) (2015) 2487–2497.

[39] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, K. Poolla, Smart grid data integrity attacks: characterizations and countermeasures,

in: Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on, 2011, pp. 232–237.

[40] Q. Yang, R. Min, D. An, W. Yu, X. Yang, Towards optimal pmu placement against data integrity attacks in smart grid, in: Annual Conference on Information Science and Systems (CISS), 2016, pp. 54–58.

[41] O. Kosut, L. Jia, R. J. Thomas, L. Tong, Limiting false data attacks on power system state estimation, in: IEEE Conference on Information Sciences and Systems (CISS), Princeton, NJ.

[42] O. Kosut, L. Jia, R. J. Thomas, L. Tong, On malicious data attacks on power system state estimation, in: International Universities Power Engineering Conference (UPEC), 2010.

[43] Y. Huang, H. Li, K. Campbell, Z. Han, Defending false data injection attack on smart grid network using adaptive cusum test, in: Annual Conference on Information Sciences and Systems (CISS), 2011.

[44] Y. Huang, J. Tang, Y. Cheng, H. Li, K. A. Campbell, Z. Han, Real-time detection of false data injection in smart grid networks: An adaptive cusum method and analysis, IEEE Systems Journal 10 (2) (2016) 532–543.

[45] J. Zhu, X. Wei, Defending false data injection attacks against power system state estimation: A stealthiness corruption-oriented method, in: IEEE International Conference on Power System Technology (POWERCON), 2016, pp. 1–5.

[46] R. Deng, G. Xiao, R. Lu, Defending against false data injection attacks on power system state estimation, IEEE Transactions on Industrial Informatics 13 (1) (2017) 198–207.

[47] R. D. Zimmerman, C. E. Murillo-Sanchez, R. J. Thomas, Polish power grid data set, `http://www.pserc.cornell.edu/matpower/`, last Accessed on December 24, 2018.

[48] D. J. Watts, S. H. Strogatz, Western states power grid data set, https://toreopsahl.com/datasets/#uspowergrid, last Accessed on December 24, 2018.