Security Threat Analytics and Countermeasure Synthesis for Power System State Estimation

Mohammad Ashiqur Rahman and Ehab Al-Shaer
Department of Software and Information Systems
University of North Carolina at Charlotte, USA
{mrahman4, ealshaer}@uncc.edu

Rajesh G. Kavasseri
Department of Electrical and Computer Engineering
North Dakota State University, USA
Rajesh.Kavasseri@ndsu.edu

Abstract—**State estimation plays a critically important role in ensuring the secure and reliable operation of the power grid. However, recent works have shown that the widely used weighted least squares (WLS) estimator, which uses several system wide measurements, is vulnerable to cyber attacks wherein an adversary can alter certain measurements to corrupt the estimator's solution, but evade the estimator's existing bad data detection algorithms and thus remain invisible to the system operator. Realistically, such a stealthy attack in its most general form has several constraints, particularly in terms of an adversary's knowledge and resources for achieving a desired attack outcome. In this light, we present a formal framework to systematically investigate the feasibility of stealthy attacks considering constraints of the adversary. In addition, unlike prior works, our approach allows the modeling of attacks on topology mappings, where an adversary can drastically strengthen stealthy attacks by intentionally introducing topology errors. Moreover, we show that this framework allows an operator to synthesize cost-effective countermeasures based on given resource constraints and security requirements in order to resist stealthy attacks. The proposed approach is illustrated on standard IEEE test cases.**

***Keywords*-Power Grid, State Estimation, False Data Injection Attack, Formal Method.**

I. Introduction

In the power grid, state estimation (SE) is the process of finding the best estimate for the system state in a weighted least square sense, given a measurement model and a set of measurements acquired through a Supervisory Control and Data Acquisition (SCADA) system. The state corresponds to the vector of bus (or node) voltages, from which line (or branch) currents and power flows can be computed. The results from state estimation aid system operators in assessing security, initiating corrective control measures, and enabling pricing calculations for real-time electricity markets. Hence, state estimation is a critical and inherent part of energy management system (EMS) applications for the power grid. However, critical infrastructures relying on SCADA based measurements are vulnerable to cyber-attacks [1]. It is important to note that while Phasor Measurement Units (PMUs) are gradually being deployed, still the current grid largely relies on extensive SCADA measurements for several EMS applications, including state estimation.

Recent work, particularly by [2], has revealed that state estimation is vulnerable to a special type of cyber-attacks, where an adversary can alter certain measurements by injecting false data to corrupt the estimator's result, while remain invisible to the system operator by evading the existing bad data detection algorithms. These attacks are known as Undetected False Data Injection (UFDI) attacks. The idea behind these attacks is interesting. The state estimation process widely uses weighted least squares (WLS) to estimate states. The process uses high measurement redundancy to detect and filter bad data (*i.e.*, noisy meter measurements) by checking whether the measurement residual, which is the l_2-norm of the difference between observed and estimated measurements, is below a certain threshold [3], [4]. An adversary who knows the complete measurement model can then inject or manipulate meter measurements consistent with the measurement model to bypass the bad data detection (BDD) process [2], [5]. It is shown in [6], [7] that such UFDI attacks can be defended if a strategically chosen set of measurements are secured. The algorithms to identify such a measurement set was also shown to be equivalent to the NP-complete hitting set problem.

In contrast, we propose a security threat analytical framework, which considers a UFDI attack against state estimation in its most generic and broadest form by casting the entire problem into a formal verification, particularly a constraint satisfaction model. The framework is built using SMT (Satisfiability Modulo Theories), which is a powerful tool for solving constraint satisfaction problems with thousands of variables and millions of clauses [8].

Contributions: In this work, we define the UFDI attack model comprehensively in terms of different *attack attributes*, which model an adversary's knowledge, resources, and attack goals. Moreover, our approach allows one to model attacks on the *topology processor*, which is responsible to map the grid topology based on statuses of switches and circuit breakers across the system. This topology is used in state estimation. An attack on this unit introduces topology errors by excluding lines actually in service and including lines not in service. An attack on the topology is often known as topology poisoning. Since there are topology error detection algorithms [4], it is important to examine if an adversary can strengthen the potency of UFDI attacks by introducing topology errors. Our framework captures all possible interrelations between attack variables, along with topology poisoning, to determine the feasibility and outcomes of an attack, *i.e.*, the states under attack and the corresponding attack vector. More importantly, with this framework, we propose a mechanism for automatic synthesis of a security architecture (*i.e.*, the set of measurements or buses) that need to be secured, with respect to a list of security requirements (*i.e.*, expected attack model) and the grid operator's constraints. In summary, our contribution is twofold: (i) developing a formal framework for verifying potential UFDI attack threats, which includes the modeling of a comprehensive set of attack attributes as well as the modeling of topology poisoning attacks; and (ii) developing a mechanism for automatic synthesis of countermeasures to resist UFDI attacks under specified requirements and constraints.

The rest of this paper is organized as follows: In Section II, we provide the necessary background and our motivation. We present our formal model in Section III. The security architecture synthesis mechanism is described in Section IV followed by evaluations on test cases. We briefly discuss the related work in Section VI and conclude in Section VII.

II. Background

The stealthy attacks on state estimation (*e.g.*, [2], [5]) are based on the DC power flow model. This DC model is simplistic, but popular and useful for preliminary analytical power systems studies.

A. 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 [9]. The voltage magnitudes at all buses are taken 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: $[\mathbf{B}][\theta] = [\mathbf{P}]$.

B. State Estimation and UFDI Attack

The state estimation problem is to estimate n number of power system state variables $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$ based on m ($m > n$) number of meter measurements $\mathbf{z} = (z_1, z_2, \cdots, z_m)^T$, according to the relationship: $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e}$, where $\mathbf{h}(\mathbf{x}) = (h_1(x_1, \cdots, x_n), \cdots, h_m(x_1, \cdots, x_n))^T$ and $\mathbf{e}$ is the vector of measurement errors [3], [4]. In the case of the linearized estimation model, *i.e.*, according to the DC power flow model, we have:

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

$\mathbf{H}$ is known as the Jacobian matrix. When the measurement errors are normally distributed with zero mean, the state estimate $\hat{\mathbf{x}}$ is calculated as:

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

Here, $\mathbf{W}$ is a diagonal matrix whose elements are reciprocals of variances of the meter errors. Thus, estimated measurements are calculated as $\mathbf{H}\hat{\mathbf{x}}$ and the residual $||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}||$ is used to identify bad data. Under these assumptions, it can be shown that the residual follows a χ^2 distribution with $m - n$ degrees of freedom. A threshold τ is set using a hypothesis test at a significance level such that the condition $||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}|| > \tau$ implies the presence of bad data [4]. UFDI attacks [2] are based on the idea that if the attack vector $\mathbf{a}$ follows from $\mathbf{H}$, such that $\mathbf{a} = \mathbf{H}\mathbf{c}$, where $\mathbf{c}$ is the vector of changes in states due to $\mathbf{a}$, then the residual remains unchanged. 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 $||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}||$. Thus, the implicit assumption here is that the adversary has full knowledge of the measurement model $\mathbf{H}$.

Topology Processor: Instead of using a fixed *a priori* model of the system to generate $\mathbf{H}$, the EMS uses a *topology processor* to map the grid topology [3]. This processor analyzes the statuses of various switches and circuit-breakers in the system and determines the connectivity among different electrical nodes (*i.e.*, buses). These topology statuses from the switches and circuit-breakers are periodically telemetered to the control center. Once the grid connectivity matrix $\mathbf{A}$ and the branch admittance matrix $\mathbf{D}$ are known, the measurement matrix $\mathbf{H}$ is computed as follows [10]:

$$\mathbf{H} = \begin{bmatrix} \mathbf{DA} \\ -\mathbf{DA} \\ \mathbf{A}^T\mathbf{DA} \end{bmatrix} \tag{2}$$

Matrices $\mathbf{DA}$ (*i.e.*, multiplication of $\mathbf{D}$ and $\mathbf{A}$) and $-\mathbf{DA}$ represent the line power flows in forward and backward directions, respectively. Matrix $\mathbf{A}^T\mathbf{DA}$ (*i.e.*, multiplication of $\mathbf{A}^T$ and $\mathbf{DA}$) represents power consumption at the buses.

The state estimated solution (from Equation (1)) provides the estimate of bus voltages from which the system power-flows can be computed. Summing up the net power flows incident on a bus then yields the estimated power (or load) at that bus.

C. Attack Model

Our approach is to model a UFDI attack in its most generic form to allow the evaluation of the feasibility of an attack under various scenarios. The attack attributes that represent the attack model are discussed in the following:

Accessibility: An attacker may not have access to all of the measurements, when physical or remote access to substations is restricted or when certain measurements are already secured. For example, in order to inject false data to the measurements taken at a substation (*i.e.*, bus), an attacker needs to have the access to that substation (or to the corresponding Remote Terminal Unit) [10].

Resource Constraint: An adversary may be constrained in 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. It is useful to represent this resource limitation with respect to buses. 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 compared to the set of measurements distributed in a small number of substations.

Grid Topology and Knowledge: State estimation of a power system is done based on the given topology (*i.e.*, connectivity among the buses) of the grid. This topology is mapped by the topology processor. For a successful UFDI attack, an attacker needs to know the grid topology and the electrical parameters of the transmission lines, which is not trivial [2]. In the case of partial knowledge, the attacker's capability becomes restricted. On the other hand, an attacker can inflict novel UFDI attacks against SE by conveying false status information at the transmitting devices or media, such that the topology generated by the processor includes one or more open lines (*i.e.*, non-existing in the true topology), or excludes one or more closed lines (*i.e.*, existing in the true topology).

Attack Goal: An attacker may choose to inject false data on certain chosen measurements with a specific aim of corrupting a certain set of state estimated solutions, or target a specific portion of the system.

As the prior works (*e.g.*, [2], [5], [11]) address UFDI attacks considering these attack attributes in isolation, we take the challenge to assess the attack feasibility when these attributes, particularly topology poisoning attacks, are all considered simultaneously, in which case the interrelation between these attack variables has an integral impact. We model the UFDI attack on state estimation as a constraint satisfaction problem, the solution to which answers whether a UFDI attack can be launched in a particular attack scenario with respect to a given set of attack constraints. Our formal model framework allows a grid operator to analyze and explore potential threats under different attack scenarios and initiate appropriate security measures. The proposed framework is described in the following section. In addition, we also propose an automated mechanism to synthesize a security architecture (*i.e.*, measurements that need to be secured) satisfying given security requirements, which actually specify the protection of state estimation from UFDI attacks with respect to a given attack model.

III. Formal Model of Undetected False Data Injection Attack

In this section, we present our modeling of verifying potential undetected false data injection attacks. In order to model UFDI attack, we need a number of parameters to denote different system properties and attack attributes. These parameters are shown in Table I. In this paper, we use two-letter notations to denote many parameters. We expect that these two-letter notations will help the readers to recall them. Also note that, in this paper, no multiplication of two parameters is represented without the multiplication sign.

A. Preliminaries

According to the DC power flow model, the admittance of a line or branch is computed from its reactance. The direction of the line is taken based on the current flow direction, *i.e.*, from a end-bus to another 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 the line 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 direction of the power flows are opposite. We use P_i^L to denote the power flow through line i, while P_j^B 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):

$$\forall_{1 \leq i \leq l} \;\; P_i^L = ld_i(\theta_{lf_i} - \theta_{lt_i}) \tag{3}$$

Equation (3) 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 a bus j is simply 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:

$$\forall_{1 \leq j \leq b} \;\; P_j^B = \sum_{i \in \mathbb{L}_{j,in}} P_i^L - \sum_{i \in \mathbb{L}_{j,out}} P_i^L \tag{4}$$

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 II.

B. Parameters for Modeling UFDI Attack

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 is required to be changed, cb_j becomes true.

Here, 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. Note that if the end-buses of a line are unknown, the corresponding row in $\mathbf{A}$ is fully unknown to the attacker. In this case, there is no way for an adversary to launch UFDI attacks on the system. 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

TABLE I
MODELING PARAMETERS

Notation	Data Type	Definition
b	Integer	The number of buses in the grid.
l	Integer	The number of lines in the grid topology.
lf_i	Integer	The *from-bus* of line i.
lt_i	Integer	The *to-bus* of line i.
ld_i	Real	The admittance of line i.
bd_i	Boolean	Whether the admittance of line i is known.
P_i^L	Real	The power flow through line i.
P_j^B	Real	The power consumption at bus j.
θ_j	Real	The state value, *i.e.*, the voltage phase angle, at bus j.
n	Integer	The number of states.
m	Integer	The number of potential measurements.
cz_i	Boolean	Whether measurement z_i is required to be altered for the attack.
cx_j	Boolean	Whether state x_j is affected due to false data injection.
cb_j	Boolean	Whether any measurement residing at bus j is required to be changed.
mz_i	Boolean	Whether potential measurement z_i is taken.
az_i	Boolean	Whether measurement z_i is accessible to the attacker.
sz_i	Boolean	Whether the measurement is secured.
tl_i	Boolean	Whether line i exists in the true (real) topology.
fl_i	Boolean	Whether line i is fixed in the topology, *i.e.*, the line belongs to the core topology.
sl_i	Boolean	Whether the status information regarding line i is secured.
el_i	Boolean	Whether line i is excluded from the topology by an exclusion attack.
il_i	Boolean	Whether line i is included in the topology by an inclusion attack.
ml_i	Boolean	Whether line i is considered (though it may not exist) in the topology.

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.

C. Parameters for Modeling Topology Poisoning

The topology of a power grid represents the connectivity among the grid buses. An attacker can inject false data in the topology information sent by various circuit breakers and switches in order to change the topology. Changes in the topology that we assume in this work include: (i) exclusion of a (closed) line from the topology (*exclusion attack*), and (ii) inclusion of a open line in the topology (*inclusion attack*). Here, we also assume that the adversary can coordinate a topology error with other measurements to render the attack undetected. Therefore, a UFDI attack can be performed by leveraging the modified topology.

We assume that some of the lines in the topology are fixed (*i.e.*, they are never opened), which form the core part of the topology. We also allow the declaration of secure line statuses, *i.e.*, their topology is always faithfully represented in SE. In order to model all these properties plus the topology change, we use a list notations as shown in Table I. We use tl_i to denote whether line i is the true or real topology, while fl_i and sl_i denote whether the line is fixed and the line status is secure, respectively. In order to denote exclusion and inclusion attack, we use el_i and il_i, respectively. Finally, ml_i represents whether line i is considered (*i.e.*, mapped) in the topology.

D. Formalization of Change in State Estimation

The attack on state x_j specifies that the phase angle at bus j is changed. This condition is formalized as follows:

$$\forall_{1 \leq j \leq n} \quad cx_j \rightarrow (\Delta\theta_j \neq 0) \tag{5}$$

From Equation (3), 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 (3) turns into the following:

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

The subtraction of Equation (3) 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:

$$\forall_{1 \leq i \leq l} \quad ml_i \rightarrow (\Delta P_i^L = ld_i(\Delta\theta_{lf_i} - \Delta\theta_{lt_i})) \tag{6}$$

If a line is not considered in the topology, then there should be no requirement of false data injection to corresponding measurements for launching UFDI attacks:

$$\forall_{1 \leq i \leq l} \quad \neg ml_i \rightarrow (\Delta P_i^L = 0) \tag{7}$$

E. Formalization of Topology Change

In the case of an inclusion attack, a line is considered in the topology though the line is open in reality. Conversely, a closed line in service is omitted in an exclusion attack. These are formalized as follows:

$$\forall_{1 \leq i \leq l} \quad ml_i \rightarrow (tl_i \wedge \neg el_i) \vee (\neg tl_i \wedge il_i) \tag{8}$$

A line can be excluded from the topology if and only if the line exists in the real or true topology and it is not a securely fixed line. This is formalized as follows:

$$\forall_{1 \leq i \leq l} \quad el_i \rightarrow tl_i \wedge \neg fl_i \wedge \neg sl_i \tag{9}$$

Similarly, a line can be included in the topology if the following condition holds:

$$\forall_{1\leq i\leq l}\ \ il_i \rightarrow \neg tl_i \wedge \neg sl_i \tag{10}$$

Note that for a topology error to remain undetected, it is necessary to alter certain measurements in necessary amounts. If a closed line is excluded from the topology, the corresponding line power flow measurement must be zero. As the states remain the same after the topology change, the corresponding connected buses' power consumption measurements are adjusted accordingly. On the other hand, when a open line is included in the topology, there should be a non-zero line power flow according to the phase difference between the connected buses. Let $\Delta\bar{P}_i^L$ be the change amount in the power flow measurement of line i in the case of a topology change. Then, the following constraints hold:

$$\forall_{1\leq i\leq l}\ \ el_i \rightarrow (\Delta\bar{P}_i^L = -P_i^L) \tag{11}$$

$$\forall_{1\leq i\leq l}\ \ il_i \rightarrow (\Delta\bar{P}_i^L = P_i^L) \tag{12}$$

If no exclusion or inclusion attack is done on line i, then $\Delta\bar{P}_i^L = 0$. Now, if line power flow measurement i (or $l+i$) needs to change, according to Equations (11) and (12), we need to know P_i^L. In the case of exclusion attack, P_i^L already exists (*i.e.*, the actual measurement) and the attacker must have the access to it. In the case of an exclusion attack, P_i^L needs to be estimated based on the difference between the states (θ_js) of the connecting buses.

F. Formalization of False Data Injection to Measurements

Here, we compute and formalize necessary changes required on the measurement to coordinate the attack. The change for a power flow measurement is the summation of individual changes that are required for topology changes and state changes. If $\Delta P_{i,total}^L$ be the total change required on the line i's power flow, then:

$$\forall_{1\leq i\leq l}\ \ \Delta P_{i,total}^L = \Delta P_i^L + \Delta\bar{P}_i^L \tag{13}$$

According to Equation (4), the change in the measurement of the power consumption ($\Delta P_{j,total}^B$) at a bus depends on the total changes done in the power flow measurements of the lines incident to this bus. Therefore,

$$\forall_{1\leq j\leq b}\ \Delta P_{j,total}^B = \sum_{i\in\mathbb{L}_{j,in}} \Delta P_{i,total}^L - \sum_{i\in\mathbb{L}_{j,out}} \Delta P_{i,total}^L \tag{14}$$

When $\Delta P_{i,total}^L \neq 0$, then taken measurements corresponding to line i (*i.e.*, mz_i and mz_{l+i}) are required to be altered. Similarly, when $\Delta P_{j,total}^B \neq 0$, the power consumption measurement at bus j needs to be changed:

$$\begin{aligned}&\forall_{1\leq i\leq l}(\Delta P_{i,total}^L \neq 0) \rightarrow (mz_i \rightarrow cz_i) \wedge (mz_{l+i} \rightarrow cz_{l+i})\\ &\forall_{1\leq j\leq b}\ (\Delta P_{j,total}^B \neq 0) \rightarrow (mz_{2l+j} \rightarrow cz_{2l+j})\end{aligned} \tag{15}$$

Conversely, measurement z_i is altered only if it is taken and corresponding power measurement is changed:

$$\begin{aligned}&\forall_{1\leq i\leq l}\ \ cz_i \rightarrow mz_i \wedge (\Delta P_{i,total}^L \neq 0)\\ &\forall_{1\leq i\leq l}\ \ cz_{l+i} \rightarrow mz_{l+i} \wedge (\Delta P_{i,total}^L \neq 0)\\ &\forall_{1\leq j\leq b}\ \ cz_{2l+j} \rightarrow mz_{2l+j} \wedge (\Delta P_{j,total}^B \neq 0)\end{aligned} \tag{16}$$

G. Formalization of Attack Attributes

Attacker's Knowledge If the admittance of a line is unknown, then an adversary cannot determine the necessary changes that needs to be applied on the measurements associated to the line. We formalize this condition as follows:

$$\forall_{1\leq i\leq l}\ \ (\Delta P_i^L \neq 0) \rightarrow ((t_i \vee t_{l+i} \vee t_{f_i} \vee t_{e_i}) \rightarrow g_i) \tag{17}$$

The following equation shows an example of specifying the attacker's knowledge about the admittances of the lines:

$$bd_1 \wedge bd_2 \wedge bd_3 \wedge \neg bd_4 \wedge \cdots \wedge bd_l \tag{18}$$

Attacker's Accessibility. The attacker usually does not have necessary physical or remote access to inject false data to all the measurements. If a measurement is secured, then though the attacker may have the ability to inject false data to the measurement, the false data injection will not be successful. Hence, the attacker will only be able to change measurement z_i in order to attack, if the following condition holds:

$$\forall_{1\leq i\leq m}\ \ cz_i \rightarrow az_i \wedge \neg sz_i \tag{19}$$

Whether a measurement is secured or not as well as whether a measurement is accessible to the attacker or not are specified, for example, as follows:

$$\neg sz_1 \wedge sz_2 \wedge \neg sz_3 \wedge \neg sz_4 \wedge \cdots \wedge sz_m \tag{20}$$

$$az_1 \wedge \neg az_2 \wedge az_3 \wedge \neg az_4 \wedge \cdots \wedge az_m \tag{21}$$

Attacker's Capability for Simultaneous Attacks. The resource limitation specifies that, at a particular time, the attacker can inject false data to T_{CZ} number of measurements, at the maximum:

$$\sum_{1\leq i\leq l} cz_i \ \leq T_{CZ} \tag{22}$$

Due to limited resources, an attacker can only access or compromise a limited number of buses at a particular time. A bus is required to be accessed or compromised if a measurement residing at this bus is required to be altered. Therefore:

$$\begin{aligned}&\forall_{1\leq i\leq l}\ \ cz_i \rightarrow cb_{lf_i}\\ &\forall_{1\leq i\leq l}\ \ cz_{l+i} \rightarrow cb_{lt_i}\\ &\forall_{1\leq j\leq b}\ \ cz_{2l+j} \rightarrow cb_j\end{aligned} \tag{23}$$

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

$$\sum_{1\leq j\leq b} cb_j \leq T_{CB} \tag{24}$$

Attacker's Target. The attacker most often has a selected set of states for launching attack. However, the attacker usually

TABLE II
LINE INFORMATION OF THE EXAMPLE IN SECTION III-I

Line #	From Bus	To Bus	Line Admittance	Knowledge Status	In True Topology	In Core Topology	Topology Information Secured	Can Alter?
1	1	2	16.90	1	1	1	0	0
2	1	5	4.48	1	1	1	0	0
3	2	3	5.05	0 [a]	1	1	0	0
4	2	4	5.67	1	1	1	0	0
5	2	5	5.75	1	1	0 [b]	0	0
6	3	4	5.85	1	1	1	0	0
7	4	5	23.75	0	1	1	0	0
8	4	7	4.78	1	1	1	0	0
9	4	9	1.80	1	1	1	0	0
10	5	6	3.97	1	1	1	0	0
11	6	11	5.03	1	1	1	0	0
12	6	12	3.91	1	1	1	0	0
13	6	13	7.68	1	1	0	0	0
14	7	8	5.68	1	1	1	0	0
15	7	9	9.09	1	1	1	0	0
16	9	10	11.83	1	1	1	0	0
17	9	14	3.70	0	1	1	0	0
18	10	11	5.21	1	1	1	0	0
19	12	13	5.00	1	1	1	0	0
20	13	14	2.87	1	1	1	0	0

[a] The attacker does not know the impedance of this line.

[b] This line is not fixed in the topology (*i.e.*, it is not a part of the core topology

has no specification on the rest of the states. That is, an unspecified state might be attacked or not. For example, if the attacker targets to attack states 1, 4, and 6, then:

$$cx_1 \wedge cx_4 \wedge cx_6 \tag{25}$$

It is possible to launch a UFDI attack on a number of measurements if the attacker can form a cut that divides the grid into two disjoint islands [11]. The attacker can attack all of the buses of one side of the cut with respect to the other side by altering the power flow and consumption measurements of the lines and the buses on the cut. However, in this case, all of the attacked buses have the same change of their states (*i.e.*, phase angles). If the state change of a bus is the same as that of the neighboring buses, then there is no state change relative to each other. In this case, the impact due to the attack might not be significant. Therefore, we also consider the constraints specifying whether state changes are required to be different. For example, if the attacker requires that state 1 and state 4 must have a different amount of change, then:

$$(\theta_1 \neq \theta_4) \wedge \cdots \tag{26}$$

H. Implementation

We encode the system configuration and the constraints into SMT [8]. We write a program leveraging the *Z3 .Net API* [12] for encoding the formalization of our proposed false data injection model. We encode our formalizations mainly using Boolean (*i.e.*, for logical constraints) and real (*e.g.*, for the relation between power flows or consumptions with states) terms. The system configurations and the constraints are given in a text file (*input* file). By executing the model (in Z3), we obtain the verification result as either satisfiable (*sat*) or unsatisfiable (*unsat*). If the result is *unsat*, it means that there is no attack vector that satisfies the constraints. In the case of *sat*, we get the attack vector from the assignments of the variables, cz_is (and cb_is), which represent the measurements required to alter for the attack.

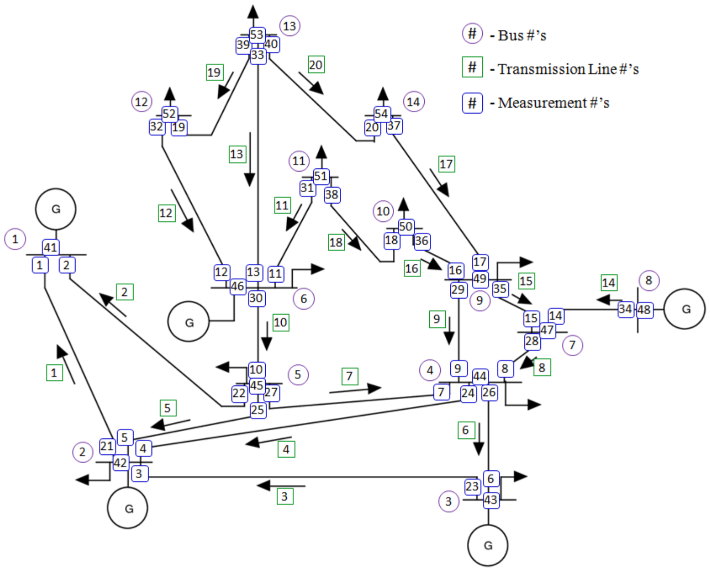


Fig. 1. The diagram of IEEE 14-bus test system. Red circles are used for bus numbers, green squares are for transmission line numbers, and round cornered blue squares are for measurement numbers.

I. An Example Case Study

We present our results on the *IEEE 14-bus test system* (see Fig. 1) [13]. The input about the line information is shown (partially) in Table II. The line information includes a set of data for each line: line number, end buses of the line, a value indicating the line admittance, the knowledge status (*i.e.*, whether the line admittance is known to the attacker), and three data about this line regarding the grid topology (*i.e.*, whether this line is included in the actual topology, whether its existence is fixed in the topology, and whether associated topology information is secured). In this example, the admittances of lines 3, 7 and 17 are unknown. All of the 20 lines (as shown in Fig. 1) are included in the true topology, though lines 5 and 13 are not a part of the core topology (*i.e.*, these lines can be kept open if necessary).

The input about the measurements is partially shown in

TABLE III
MEASUREMENT INFO OF THE EXAMPLE IN SECTION III-I

Measurement #	Is Recorded?	Secured	Can Alter?
1	1 [a]	1 [b]	0
2	1	1	0
3	1	0	1 [c]
4	1	0	1
5	0	0	0
…	…	…	…
11	1	0	1
12	1	0	1
13	1	0	1
14	0	0	0
15	1	1	1
…	…	…	…
21	1	0	1
22	0	0	0
23	1	0	1
24	1	0	1
25	1	1	1
…	…	…	…
41	1	1	0
42	1	0	1
43	1	0	1
44	1	0	1
45	1	1	0
…	…	…	…

[a] The measurement is taken or recorded for state estimation.
[b] The measurement is secured, especially in terms of integrity.
[c] The attacker has the accessibility to alter the measurement.

Table III. Since this system has 14 buses and 20 lines, the maximum number of potential measurements is (14 + 2×20) or 54. Each row of Table III includes (i) whether the measurement is taken for state estimation (all the potential measurements are taken except measurements 5, 10, 14, 19, 22, 27, 30, 35, 43, and 52), (ii) whether the measurement is secured (measurements 1, 2, 6, 15, 25, 32, and 41 are secured) and (iii) whether the attacker has the accessibility to alter the measurement. Let us now consider two different objectives of the attacker.

Attack Objective 1. Let the attacker's objective be to attack states 9 and 10 but in different amounts. Due to the resource limitation, he cannot alter more than 16 measurements at a time, and these measurements cannot be distributed in more than 7 substations (*i.e.*, buses). The execution of the model corresponding to this example returns *sat* along with the assignments to different variables of the model. From the assignments, we find that the measurements selected for attacking states 9 and 10 are 8, 9, 16, 18, 20, 28, 29, 36, 38, 40, 44, 47, 50, 51, 53, and 54. These measurements are distributed in buses 4, 7, 9, 10, 11, 13, and 14. If the attacker's resources are more limited (*e.g.*, 15 measurements and/or 6 buses only), then *unsat* is returned. However, if the attacks on states 9 and 10 can be the same, then there is a solution. In this case, the measurements for false data injection are 8, 9, 11, 13, 28, 29, 31, 33, 39, 44, 46, 47, 49, 51, and 53, while the corresponding buses are 4, 6, 7, 9, 11, and 13. In both of these cases, along with 9 and 10, some other states are also required to be corrupted; only states 9 and 10 cannot be attacked alone.

Attack Objective 2. Here the attacker's objective is to attack state 12 only, *i.e.*, no other states will be affected. The execution of the corresponding model shows that measurements 12, 32, 39, 46, and 53 are required to alter in this case. If measurement 46 is considered as secured, then no attack vector is possible. Let us now consider that the attacker has the ability to alter the topology information. In this scenario, we have a solution, where line 13 is excluded from the topology by injecting false data into the topology information. In this case, the measurements for false data injection are 12, 13, 32, 33, 39, and 53, which include necessary changes required for the state change along with the topology change.

IV. SYNTHESIS OF SECURITY ARCHITECTURE FOR PROTECTING STATE ESTIMATION

In the last section, we have described the model for figuring out potential UFDI attacks under given constraints. The proposed verification model allows a grid operator to understand potential threats on state estimation with respect to an expected scale of attack (expressed in terms of different attack attributes) and to take necessary security measures accordingly. However, we need an automated solution to find out such a security architecture. In this section, we present such an automated mechanism for synthesizing security architecture.

A. Background

Though the authors in [6], [7] show that UFDI attacks can be defended if a strategically chosen set of measurements are secured, they only consider a specific attack model, where adversaries have perfect knowledge and they are not limited in capability. Based on this worst case attack model, the set of measurements to be secured can exceed the grid operator's resource (budget). Therefore, a security design is required that can give security within the limited capability of the grid operator, while keep the power system state estimation secure with respect to an attack model (security requirements).

Our solution utilizes the verification model to find out a security architecture. A security architecture typically includes a list of measurements that are required to be secured. Since securing a number of measurements distributed in many substations are very costly compared to a set of measurements distributed in a small number substations, we mainly focus on substation, *i.e.*, bus specific security architecture. Moreover, securing a bus usually means securing all of the measurements taken in that bus. A bus can be secured by deploying a PMU (can be multiple for a large bus) at the bus with necessary security measures [14]. By the security measures, we mainly consider the *data integrity protection* of the measurements. Since the PMU can provide voltage phasor of the bus and current phasors of all the branches incident to the bus, if the PMU is secured then all of these measurements become secured. At the unit level, security is being provisioned by existing PMU vendors [15]. Here, though we propose a mechanism to find the security architecture as a set of buses to be secured, similar mechanism can be used for synthesizing security architecture with respect to measurements only.

B. Synthesis Design

Fig. 2 shows the flow diagram of the security architecture synthesis mechanism for resisting state estimation attacks. It is an iterative approach with the combination of two formal models. One of these models is the *candidate security*

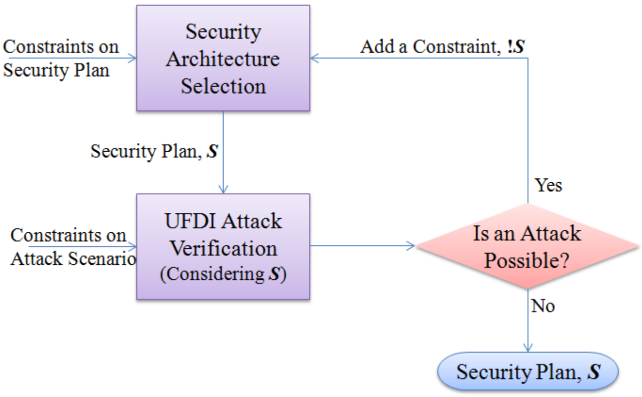

Constraints on Security Plan
Security Architecture Selection
Add a Constraint, !S
Security Plan, S
Yes
Constraints on Attack Scenario
UFDI Attack Verification (Considering S)
Is an Attack Possible?
No
Security Plan, S

Fig. 2. The flow diagram of the security architecture synthesis mechanism for protecting state estimation attack.

architecture selection model. That is, it selects the set of buses as a candidate of the security architecture considering some invariant and user-driven constraints on the security architecture. We discuss this candidate security architecture selection model in the following subsection. The second model is our UFDI attack verification model, which verifies whether the selected candidate architecture can protect state estimation from UFDI attacks with respect to the security requirements (*i.e.*, an expected attack model). Security requirements are ensured when the verification model returns *unsat* (*i.e.*, no attack vector can be found). If a candidate architecture fails to ensure the required security, a constraint is added to the candidate security architecture selection model so that this architecture is removed from the potential candidate set. The updated model is solved for another candidate architecture and the verification model is used to ensure the security requirements. This process continues till a security architecture is found, *i.e.*, as long as the verification model returns *unsat*. However, when the candidate architecture selection model fails to return a candidate set, then no security architecture is possible according to the given security requirements.

C. Formalization of Candidate Architecture Selection

The main constraint for selecting the buses in the architecture is the resource limitations of the grid operator. That is, the number of selected buses cannot exceed a limit (T_{SB}). If sb_j denotes whether bus j is secured, then:

$$\sum_{1 \leq j \leq b} sb_j \leq T_{SB} \tag{27}$$

Securing a bus implies that all of the measurements that are recorded at this bus are secured. If L_j denotes the lines connected to bus j, we formalize this as follows:

$$\begin{aligned} &\forall_{1 \leq j \leq b} \quad sb_j \rightarrow (mz_{2l+j} \rightarrow sz_{2l+j}) \\ &\forall_{1 \leq j \leq b} \quad sb_j \rightarrow \bigwedge_{i \in L_j} (mz_i \rightarrow sz_i) \wedge (mz_{l+i} \rightarrow sz_{l+i}) \end{aligned} \tag{28}$$

The grid operator may have a limitation that she is not capable to secure a particular set of buses. Those buses should be excluded from the candidate set, as shown in the following arbitrary example:

$$\neg sb_2 \wedge \neg sb_6 \wedge \cdots \tag{29}$$

Algorithm 1 Security Architecture Synthesis

```
1:  $F_{Attack}$ formalizes the UFDI attack verification model.
2:  $F_{Secure}$ formalizes the security architecture selection model.
3:  loop
4:      Save (Push) current $F_{Attack}$ into $\bar{F}_{Attack}$.
5:      if Solver returns a model $M$ (i.e., SAT) for $F_{Secure}$ then
6:          Get the security architecture $S$ from $M$.
7:      else
8:          Exit program.
9:      end if
10:     Add security constraints to $F_{Attack}$ based on $S$.
11:     if Solver returns UNSAT for $F_{Attack}$ then
12:         Return $S$.
13:     else
14:         Add the constraint $!S$ to $F_{Secure}$.
15:     end if
16:     Retrieve (Pop) the saved formalization $\bar{F}_{Attack}$ into $F_{Attack}$.
17: end loop
```

Different analytical constraints can be used to limit the search space in the security architecture selection model. From Equation (6), we know that if no change is possible in the line power flow, the phase difference between the two buses connected by the line cannot be changed. Hence, if a bus is secured (*i.e.*, all the measurements at the bus are secured), a connected bus' state cannot be changed with respect to the secured bus' state. UFDI attacks on the states of these two buses are possible through a third bus which is not connected to the secured bus but connected to the other bus. Therefore, securing the connected bus is not required to protect state estimation of the grid. Equation (30) formalizes this constraint.

$$\begin{aligned} \forall_{1 \leq j \leq b} \quad sb_j \rightarrow \bigwedge_{i \in L_j} &((lf_i = j) \wedge mz_i) \rightarrow \neg sb_{lt_i}) \wedge \\ &((lt_i = j) \wedge mz_{l+i}) \rightarrow \neg sb_{lf_i}) \end{aligned} \tag{30}$$

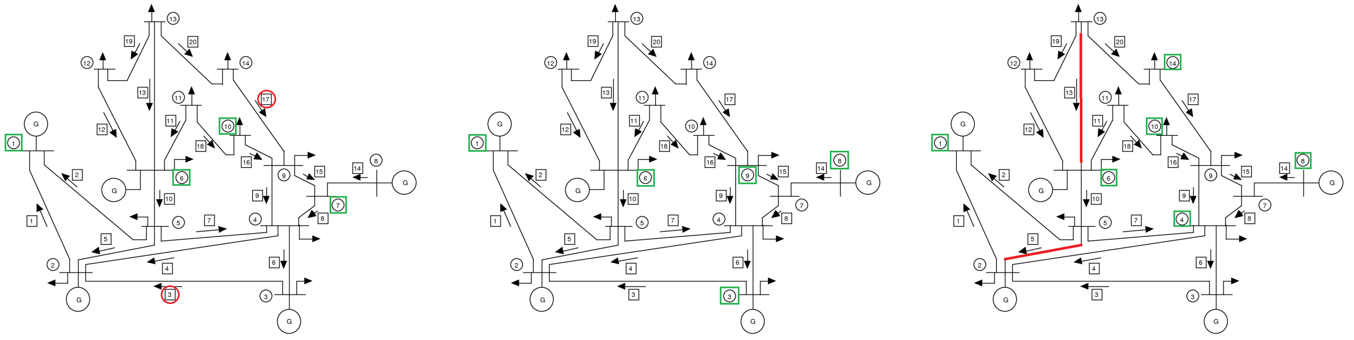
D. Implementation

Similar to our verification model, we encode the candidate security architecture selection model using SMT [8]. Then, we implement the synthesis mechanism by combining the verification model and candidate selection model as shown in Algorithm 1. The algorithm is an iterative process, which stops when a security architecture is found (line 12) or there is no more candidate set to verify (line 8).

E. Case Study

Here we present a case study based on the *IEEE 14-Bus Test System* illustrating how our proposed security architecture synthesis mechanism produces different security architectures in different scenarios, as shown in the below:

Scenario 1. The attack model of the first scenario is similar to the first part of the example (attacker's objective 1) as shown in Section III. In this scenario, the attacker has limited information, *i.e.*, admittances of lines 3 and 17 are unknown. The grid operator can consider such a constraint on the attacker's knowledge, if she is certain that the admittance information regarding this set of lines is neither disclosed nor predictable. The attacker is also have limited resources, such that he



(a) (b) (c)

Fig. 3. The security architectures (the green squared buses needs to be secured) in different scenarios: (a) incomplete information (the red circled line's admittance is unknown), (b) complete knowledge, and (c) incorporating with topology poisoning attack (the read lines are potential to inclusion or exclusion topology attacks). In all scenarios, bus 1 is the reference bus.

cannot attack more than 12 measurements simultaneously. The grid operator, due to resource constraints, can secure 4 buses maximally. Bus 1 is considered as the reference bus. In this scenario, the security architecture produced by our mechanism suggests that buses 1, 6, 7, and 10 are required to be secured (as shown in Fig. 3(a)), *i.e.*, all the measurements in these buses are data integrity protected. However, there can be different sets of buses, which also can secure the system. Our synthesis mechanism can synthesize all of these sets.

Scenario 2. In the second scenario, the attacker knows the complete information (*i.e.*, all line admittances) for launching UFDI attacks and he has the ability to inject false data to any number of measurements. In this case, there is no solution with 4 buses that can secure state estimation of the grid against UFDI attacks. If the grid operator can secure 5 buses, there is a solution. In this solution, we need to secure buses 1, 3, 6, 8, and 9 (see Fig. 3(b)).

Scenario 3. This scenario is the worse case situation compared to the last two scenarios. Here, the attacker has complete knowledge of the grid and he has the ability to inject false data to any number of measurements. In addition, the attacker can change the topology by injecting false data to the topology information. In this scenario, only lines 5 and 13 are considered as vulnerable to line exclusion or inclusion attacks. However, in this case, no solution is possible by securing 5 buses only. If 6 buses are possible to be secured, then we have a satisfiable security architecture (*i.e.*, buses 1, 4, 6, 8, 10, and 14), which is shown in Fig. 3(c).

V. Evaluation

In this section, we present the evaluation results showing the scalability of the proposed verification framework as well as that of the security architecture synthesis mechanism.

A. Methodology

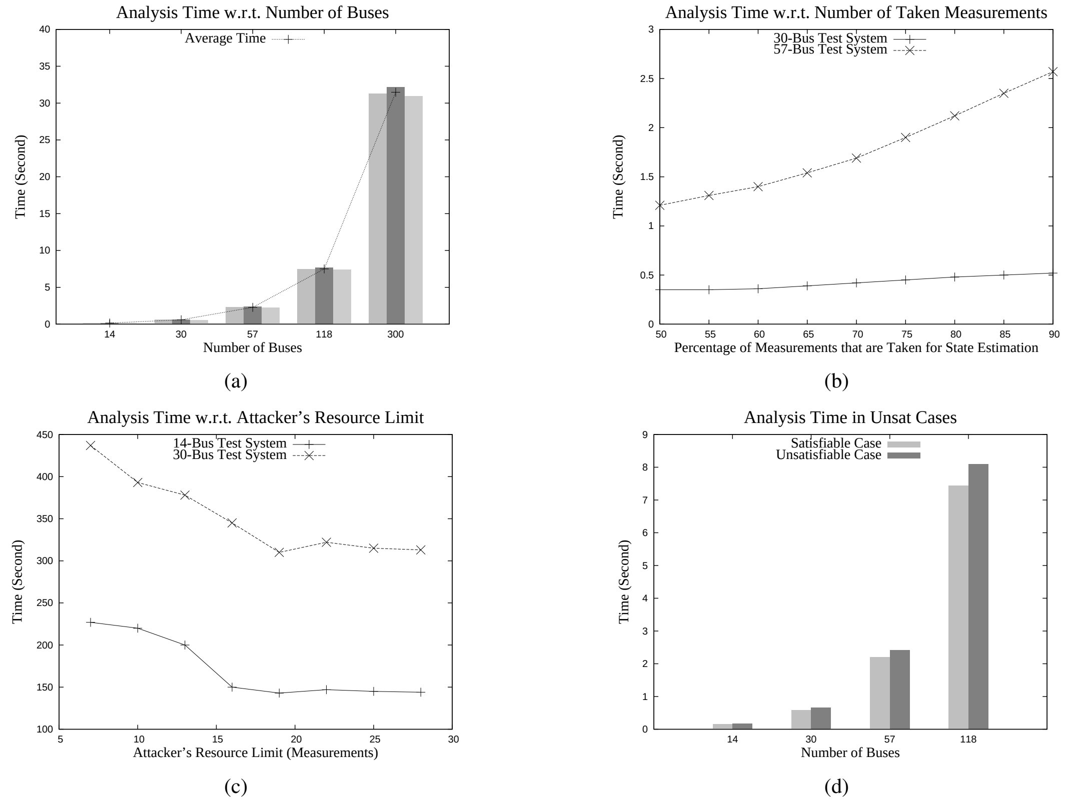
We evaluated the scalability of our proposed verification model by analyzing the time and memory requirements for executing the model in different problem sizes. Problem size depends mainly on the number of buses. We evaluated the scalability of our model based on different sizes of IEEE test systems, *i.e.*, 14-bus, 30-bus, 57-bus, 118-bus, and 300-bus [13]. We also evaluated the impact of constraints on the scalability. Similarly, we evaluated the scalability of our security architecture synthesis mechanism. We ran our experiments on an Intel Core i5 Processor with 8 GB memory. In this evaluation, we did not compare the time complexity of our proposed model with that of the related work, especially with respect to [6] and [7], as neither of them provide results showing the complexity of their respective mechanisms.

B. Time Complexity of Verification Model

Impact of the problem size: Fig. 4(a) shows the execution time of our proposed UFDI attack verification model with respect to the problem size. We varied the problem size by considering different IEEE bus test systems. We did three experiments taking different states to be attacked for each test case. The execution time of each case is shown in Fig. 4(a) using a bar chart. A graph is also drawn using the average execution time for each bus system. We observed that with respect to the bus size the increase in the execution time lies between linear and quadratic orders. For a specific bus size, we also observed that the execution time differs with a different choice of states to be attacked. It is worth mentioning that, although the general problem seems to have a quadratic growth considering the number of buses and the connectivity between them, we observed smaller execution time. Because, the complexity depends not only on the number of buses, but also on the number of lines, measurements, and attack attributes. An important feature of power grid networks is that the average degree of a node (or bus) is roughly 3, regardless of the number of buses in the system [16]. This feature can explain why the complexity is not strictly quadratic.

Impact of the number of taken measurements: We also analyzed the impact of the number of taken measurements (represented as the percentage of the total potential measurements) on the model execution time. Fig. 4(b) presents the evaluation results for the 30 and 57-bus test systems. The results shows that the execution time increases linearly with the increase in the number of taken measurements. We also observed similar results for the other test systems. When the number of recorded measurements increases, the number of measurements to be considered for false data injection also increases, which results in a longer verification time.

Impact of the Constraints: The verification of potential



(a)

(b)

(c)

(d)

Fig. 4. The verification model execution time in different experiments: (a) the execution time with respect to the number of buses, (b) the execution time with respect to the number of recorded measurements, (c) the execution time with respect to the attacker's resource limit, and (d) the execution time in unsatisfiable cases with respect to the number of buses.

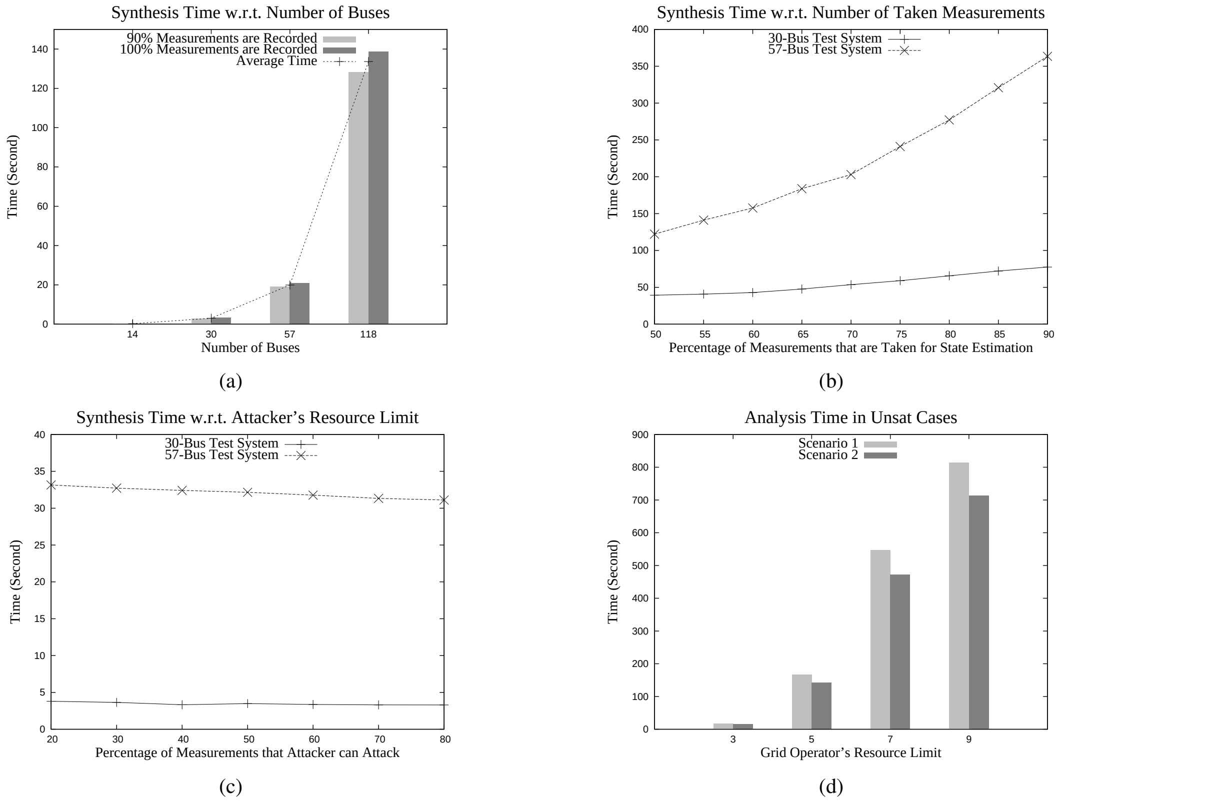
UFDI attacks depends on the given constraints, especially the attacker's access capability and resource limit. We evaluated the impact of the attacker's resource limit (*i.e.*, the number of measurements to which the attacker can inject false data at a time) on the analysis time. We consider IEEE 14- and 30-bus systems. The analysis result is shown in Fig. 4(c). We observed that the analysis time decreases with the increase in the attacker's resources (*i.e.*, resource constraints relaxes). This is due to the reason that by increasing the attacker's resources, the potential of UFDI attacks increases. However, increasing the attacker's resources does not help in UFDI attacks after some point (*e.g.*, when the attacker's resource limitation is almost 20 measurements, as shown in Fig. 4(c)). The reason is that to launch a UFDI attack to one or more states, the resource is sufficiently large.

Performance in Unsatisfiable Cases: When constraints are very tight (*e.g.*, when the attacker can attack a very limited number of measurements), there can be no satisfiable solution. In such cases, the SMT solver takes longer time to give the unsatisfiable (*unsat*) results compared to the execution time in satisfiable cases. In unsatisfiable cases, the SMT solver needs to explore the entire solution space to conclude that there is no solution based on the given constraints. Fig. 4(d) shows a comparison between the execution times for satisfiable and unsatisfiable cases, with respect to different bus systems. Since we considers different constraints and specific attack goals (corresponding to the attack attributes) for an attacker, the potentiality of an attack vector is already limited. Therefore, in our experiments we observed smaller execution time differences between satisfiable and unsatisfiable cases.

C. Time Complexity of Synthesis Mechanism

Impact of the number of buses: The execution time of our proposed security architecture synthesis mechanism with respect to different test bus systems is shown in Fig. 5(a). We considered two scenarios in our experiments: (i) 90% of the measurements are recorded for state estimation and (ii) all of the measurements are recorded for state estimation. We can see in the figure that the increase in the execution time is quadratic in order. However, this execution time is significantly longer than that of the UFDI attack verification model that we see in Fig. 4(a). Because, in order to synthesize the security architecture, the verification model may be required to be executed for many times till a security architecture is found.

Impact of the number of taken measurements: We again analyzed the impact of the number of taken measurements (the percentage of the total potential measurements) on the time of security architecture synthesis. Fig. 5(b) shows the evaluation results corresponding to the 30 and 57-bus test systems. We observed that with the increase in the number of taken measurements, the execution time increases linearly. Since the selection of security architecture is based on the buses, any increase in taken measurements does not increase the selection time. However, we know that verification time increases with the increase in taken measurements (recall Fig. 4(b)). As a result, the time for the security architecture synthesis increases.



(a)

(b)

(c)

(d)

Fig. 5. The execution time of the security architecture synthesis mechanism in different experiments: (a) the execution time with respect to the number of buses, (b) the execution time with respect to the number of taken measurements, (c) the execution time with respect to the attacker's resource limit, and (d) the execution time in unsatisfiable cases with respect to the number of buses.

TABLE IV
THE REQUIRED MEMORY SPACE (IN MB)

# of Buses	Verification Model	Candidate Selection Model
14	1.32	0.05
30	2.60	0.10
57	4.56	0.16
118	9.69	0.31

Impact of the Constraints: A security architecture depends on the given constraints, *e.g.*, the attacker's resource limit (*i.e.*, the number of measurements to which the attacker can inject false data). We analyzed the impact of this resource limit (represented as the percentage of the total measurements) on the security architecture synthesis time. The analysis result is shown in Fig. 5(c). We observed that the synthesis time decreases slowly with the increase in the attacker's resource limit value. This is due to the reason that by increasing the attacker's resources (*i.e.*, higher possibility of successful attack), the time to find that a candidate security architecture is unsuccessful (*i.e.*, satisfiability of the UFDI attack model) decreases. As a result, the synthesis time decreases.

Performance in the Unsatisfied Cases: When the grid operator's resource is very limited, then there may be no security solution. The execution time in such an unsatisfiable case is usually very high, because the synthesis mechanism requires verifying of all the potential security architectures to conclude that there is no security solution based on the given constraints. Fig. 5(d) shows the execution times of the synthesis mechanism in unsatisfiable cases. In this analysis, we took the IEEE 30-bus test system and varied the resource limit values in two different scenarios. In the first scenario a security plan needs a minimum number of 10 buses, while in the second the number is 12. No security plan is possible with less than this many buses. In the figure, we see that the closer the resource limit is to the minimum number of necessary buses, the higher the execution time is to find out that there is no solution. When the limit is too close to the minimum requirement, the unsatisfiability comes at the very end of the search, *i.e.*, the early rejection of a potential search mostly does not take place.

D. Memory Complexity

The memory required by the SMT solver [12] for executing our verification model and candidate security architecture selection model is evaluated in different IEEE bus test systems. The memory requirement for an execution of the SMT model depends mainly on the number of variables defined in the model and the number of intermediate variables generated by the solver to implement the satisfiability modulo theories used in the model. The memory analysis results are presented in Table IV, which shows that memory usage of our models increases almost linearly with the number of buses.

VI. RELATED WORK

The concept of undetected false data injection attack was presented in [2] for the first time, which was extended in [17]. The authors discussed UFDI attacks considering different scenarios, such as limited access to meters and limited resources to compromise meters, under arbitrary or specific targets, assuming that the adversary has complete information about the grid. In the general case, the attack vector computation problem is NP-complete. Therefore, the authors presented few heuristic approaches that can find attack vectors. Bobba et al. in [6] showed that for detecting UFDI attacks it is necessary and sufficient to protect a set of basic measurements, which is actually a minimum set of measurements ensuring observability. Kim and Poor in [7] proposed a greedy suboptimal algorithm, which selects a subset of measurements that can be made immune from false data injection for the protection against UFDI attacks. Kosut et al. in [18] proposed a mechanism based on the generalized likelihood ratio test to detect UFDI attacks. Similar approach is found in [19] with the help of adaptive cumulative sum control chart test.

Vukovic et al. in [10] 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. In [20], authors claimed that an l_1 relaxation-based technique provides an exact optimal solution of the data attack construction problem. UFDI attacks with incomplete or partial information are discussed in [5], [11]. These works mathematically showed the impact of incomplete knowledge on the potentiality of UFDI attacks.

However, none of the works discussed above provides a comprehensive model of UFDI attacks considering different attack attributes together. In our previous works [21], [22], we have presented verification models for the UFDI attacks with respect to a list of attack constraints and the impact on the optimal power flow (OPF) solution, which are limited to typical UFDI attacks. In this work, we give a comprehensive solution to this challenge with a broader attack scenario. We consider topology poisoning attacks, *i.e.*, false data injection attack to topology status information, in modeling UFDI attacks, and show that novel UFDI attacks are possible by intelligently introducing topology errors along with the false data injection to the measurements. Very recently, Kim and Tong have presented algebraic conditions of undetected topology attacks in power grids [23]. However, unlike to our work, the authors have not addressed the undetected attacks to state estimation leveraging the topology poisoning. In addition, utilizing this framework, we also provide an automated security architecture synthesis mechanism, which considers the grid operator's resource constraints with respect to an attack model.

VII. Conclusion

Securing state estimation against cyber-attacks is of paramount importance to maintain the integrity of the power grid. We propose an SMT based formal framework to systematically investigate potential security threats, particularly the feasibility of stealthy cyber-attacks, on state estimation. The framework allows an operator to capture interdependency among attack attributes to synthesize a security architecture, which secures a set of buses for immunity against UFDI attacks. The scalability of the model is evaluated with experiments and case-studies on different IEEE test systems. Our results show that our model can efficiently solve problems with hundreds of buses. In the case of the IEEE 118-bus test system, our verification model execution time is 7 seconds on average, while our synthesis mechanism takes around 2 minutes. The proposed method provides a basis for the development of cyber-security tools for modern power grids.

References

[1] D. Kundur, X. Feng, S. Liu, T. Zourntos, and K. Butler-Purry. Towards a framework for cyber attack impact analysis of the electric smart grid. In *IEEE International Conference on Smart Grid Communications*, pages 244 – 249, Oct 2010.

[2] Yao Liu, Peng Ning, and Michael K. Reiter. False data injection attacks against state estimation in electric power grids. In *ACM Conference on Computer and Communications Security*, pages 21–32, Nov 2009.

[3] A. Monticelli. *State estimation in electric power systems: a generalized approach*. Kluwer Academic Publishers, Norwell, MA, 1999.

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

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

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

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

[8] Leonardo de Moura and Nikolaj Bjørner. Satisfiability modulo theories: An appetizer. In *Brazilian Symposium on Formal Methods*, 2009.

[9] Allen J. Wood and Bruce F. Wollenberg. *Power Generation, Operation, and Control, 2nd Edition*. Wiley, 1996.

[10] O. Vukovic, Kin Cheong Sou, G. Dan, and H. Sandberg. Network-layer protection schemes against stealth attacks on state estimators in power systems. In *IEEE International Conference on Smart Grid Communications*, Oct 2011.

[11] M. Ashfaqur Rahman and H. Mohsenian-Rad. False data injection attacks with incomplete information against smart power grids. In *IEEE Conference on Global Communications (GLOBECOM)*, Dec 2012.

[12] Z3: An efficient smt solver. In *Microsoft Research*. http://research.microsoft.com/en-us/um/redmond/projects/z3/.

[13] Power systems test case archive. http://www.ee.washington.edu/research/pstca/.

[14] J. De La Ree, V. Centeno, J. S. Thorp, and A. G. Phadke. Synchronized phasor measurement applications in power systems. *IEEE Transactions on Smart Grid*, 1:20–27, Jun 2010.

[15] J. Stewart, T. Maufer, R. Smith, C. Anderson, and E. Ersonmez. Synchrophasor security practices, 2011. https://www.selinc.com/WorkArea/DownloadAsset.aspx?id=8502.

[16] Dennis J. Brueni and Lenwood S. Heath. The pmu placement problem. *SIAM Journal on Discrete Mathematics*, 19(3):744–761, 2005.

[17] Yao Liu, Peng Ning, and Michael K. Reiter. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security*, 14(1):13:1–13:33, Jun 2011.

[18] O. Kosut, L. Jia, R. J. Thomas, and L. Tong. On malicious data attacks on power system state estimation. In *International Universities Power Engineering Conference (UPEC)*, Aug 2010.

[19] Yi Huang, Husheng Li, K. A. Campbell, and Zhu Han. Defending false data injection attack on smart grid network using adaptive cusum test. In *Annual Conference on Information Sciences and Systems (CISS)*, pages 1 – 6, Mar 2011.

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

[21] M. Ashiqur Rahman, E. Al-Shaer, and Md. Rahman. A formal model for verifying stealthy attacks on state estimation in power grids. In *IEEE International Conference on Smart Grid Communications*, Oct 2013.

[22] M. Ashiqur Rahman, E. Al-Shaer, and 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*, Apr 2014.

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