Automated Vulnerability Analysis of AC State Estimation under Constrained False Data Injection in Electric Power Systems

Sicun Gao[1], Le Xie[2], Armando Solar-Lezama[1], Dimitrios Serpanos[3], and Howard Shrobe[1]

***Abstract*— We introduce new methods for the automatic vulnerability analysis of power grids under false data injection attacks against nonlinear (AC) state estimation. We encode the analysis problems as logical decision problems that can be solved automatically by SMT solvers. To do so, we propose an analysis technique named "symbolic propagation," which is inspired by symbolic execution methods for finding bugs and exploits in software programs. We show that the proposed methods can successfully analyze vulnerability of AC state estimation in realistic power grid models. Our approach is generalizable towards many other applications such as power flow analysis and state estimation.**

I. Introduction

Vulnerability of power grid operations under cyber attacks has become a major concern for public safety, as illustrated by realistic incidents such as the Stuxnet attack [8]. In particular, false data injection (FDI) attacks can modify the measurement data produced by the SCADA (Supervisory Control and Data Acquisition) systems and mislead power grid operators to malfunction [9]. Such attacks against linear (DC) state estimation models has been extensively studied ([9], [14], [2], [12], [13], [6]). The more realistic nonlinear (AC) state estimation, however, is considered much harder to analyze ([5], [12], [11]). In most cases, the analysis for the nonlinear models has to bypass the difficulty of solving the full nonlinear power flow equations that are involved in AC state estimation. Consequently, existing work has mostly focused on network topological and statistical analysis, or completely unconstrained attack models, which assumes that the attackers can arbitrarily modify the values of any measurement of choice, so that the complexity of analysis can be reduced.

In this paper, we study strong attack models that do require reasoning about the nonlinear constraints imposed by AC state estimation. In particular, we aim to analyze the vulnerability of buses or branches whose local environment involves measurements that are *constrained*. Namely, their values can not be changed arbitrarily, either because of physical limits (as studied in [5]) or pre-existing defense and monitoring mechanisms. When such constraints are coupled with AC state estimation mechanisms, the search for FDI attack vectors generally becomes a highly nonlinear and non-convex constraint solving problem. Our goal is to propose an approach that utilizes intensive computation power to cope with this inherent complexity.

The approach we propose builds upon several methods that originate from the seemingly unrelated field of software *program analysis*, where bug-finding and exploit-generation are long-standing problems that have seen decades of fruitful research. In fact, these problems and FDI attack analysis bear strong mathematical similarity. A software program implicitly defines a *transition system* over its *program states*. Such transition systems can be represented as graphs labeled with the program state values. Finding a bug or exploit corresponds to finding inputs to the program such that certain assertions become true, such as "the program will enter an error state" or "the program will fail to terminate." Similarly, the FDI vulnerability analysis problem is the search for input vectors to the measurement variables such that the bad data monitoring mechanisms fail to raise alerts.

State-of-the-art program analysis techniques can scale on complex programs that encode transition graphs with up to millions of nodes. The mainstream approaches all involve encoding the analysis problems as *logical decision problems*, which are then solved with highly efficient SAT/SMT solvers [7]. Indeed, recent work [10] has exploited this connection for FDI analysis on DC state estimation, and demonstrated scalability of the techniques. A potential difficulty of applying program analysis techniques to AC state estimation comes from the need of reasoning about the nonlinear relations between the state and measurement variables in power grids. Typically, SMT solvers can not handle problems that are numerically-intensive and nonlinear. This difficulty has been recently resolved in the framework of delta-decision procedures [3], which are algorithms that can reason about highly nonlinear logic formulas over real numbers. Note that the logical decision problems correspond to problems that are non-convex and non-smooth and thus NP-hard, and SMT solvers are indeed designed to cope with such complexity, with much success in practice. In this work, we demonstrate that the SMT encoding of realistic analysis problems from the power systems domain can be effectively handled by the SMT solver dReal [4].

However, the use of powerful solvers can only bring us half-way towards the goal. While efficient solvers can often handle challenging problems with many (e.g., hundreds of) nonlinear constraints, it is unrealistic to expect fully automatic solutions through direct encoding of realistic power grid with thousands of buses. A main contribution of our work is the new analysis technique "symbolic propagation," which performs localized analysis and uses intermediate

*This work was supported by: the QCRI and MIT CSAIL partnership, NSF CCF-1161775, NSF ECCS-1150944, ONR N000141310090, and DGE-1303378. [1] S. Gao, A. Solar-Lezama, and H. Shrobe are with MIT Computer Science and Artificial Intelligence Lab. [2] L. Xie is with the Department of Electrical and Computer Engineering, Texas A&M University. [3] D. Serpanos is with Qatar Computing Research Institute.

results to guide the enlargement of the sub-grids under analysis. The idea comes from the similar technique of *symbolic execution* in program analysis, which performs analysis through incremental symbolic representations of possible program states. We demonstrate that the technique is effective in the power grid setting, and avoids the reasoning about global configuration of the grids. On the other hand, the applicability of localized analysis also highlights the vulnerability of existing monitoring mechanisms.

The paper is organized as follows. We first demonstrate how to encode vulnerability analysis as logic formulas in Section III. We then develop localized analysis and symbolic propagation techniques in Section IV. We show experimental results in Section V and conclude with discussion for future directions in Section VI.

II. Preliminaries

A. Electric Power Grids

An electric power grid is a network of buses and transmission lines between them, which can be mathematically viewed as a directed graph whose nodes and edges are labeled by real-valued variables. In steady state, the admissible values of these variables satisfy the Kirchhoff laws, and the control mechanism operates on the measurements of these values. The state estimation algorithm aims to compute an optimal estimate of the system state based on the available measurements. The raw measurements are processed to filter out measurement noise and detect gross errors.

We use the following formal representation for power systems that are operating in steady state. The main state variables are the voltages magnitudes and phasor angles at each bus. Observations of the system are obtained through measuring the active and reactive power flows on each transmission line, as well as the active and reactive power injections at each bus.

Definition 1 (Power Grid System): A power grid system is defined as a tuple

$$\langle G, X^v, X^\theta, Z^p, Z^q \rangle,$$

where

- $G = \langle B, E \rangle$ is a directed graph. $B = \mathbb{N}^{<n}$ is a finite set of buses indexed by natural numbers, and $E \subseteq B^2$ is the set of transmission lines between buses.
- $X^v = \{v_i : i \in B\}$ is a set of n variables that denote the voltage magnitude of bus i.
- $X^\theta = \{\theta_i : i \in B\}$ is a set of n variables that denote the phase angle of bus i at time t.
- $Z^p = \{z^p_{ij} : (i,j) \in B \times B\}$ are variables that denote active power injections/flows on the buses/lines. z^p_{ii} is a measurement on a bus i, and z^p_{ij} is a measurement on a line (i,j) if $(i,j) \in E$.
- $Z^q = \{z^q_{ij} : (i,j) \in B \times B\}$ are variables that denote reactive power injections/flows on the buses/lines. z^p_{ii} is a measurement on a bus i, and, when $(i,j) \in E$, z^q_{ij} is a measurement on a line (i,j).

We can write $X = X^v \cup X^\theta$ and $Z = Z^P \cup Z^Q$. X is called the set of *state variables*, and Z is the set of *measurements*. We also write $B_G, E_G, X^v_G, X^\theta_G, Z^p_G, Z^q_G$ to denote the corresponding components of G.

B. First-Order Logic Formulas over Real Numbers

We will make extensive use of logic formulas in our analysis techniques. We use first-order logic formulas, in which we can mix logic and a wide range of real functions.

Definition 2 ($\mathcal{L}_{\mathbb{R}_\mathcal{F}}$-Formulas): Let $\mathcal{F}$ be a set of real functions. The $\mathcal{L}_{\mathbb{R}_\mathcal{F}}$ formulas are defined as

$$\begin{array}{rrl} \text{(terms)} & t := & x \mid f(t(\vec{x})), \text{ where } f \in \mathcal{F}; \\ \text{(formulas)} & \varphi := & t(\vec{x}) > 0 \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x \varphi. \end{array}$$

The other logical symbols are defined in the standard way.

Example 1: Consider a system of two buses connected by a transmission line. We use the variables v_1 and v_2 to encode the voltages at the two buses, and θ_1, θ_2 for the phasors. We use the variable z to represent the active power flow from Bus 1 to 2. Let the conductance and susceptance parameters of the line be given by g_{12} and b_{12}. Then, the equation $\varphi(v_1, \theta_1)$ defined as

$$\varphi: \ z = v_1^2 - v_1 v_2 g_{12} \cos(\theta_1 - \theta_2) - v_1 v_2 b_{12} \sin(\theta_1 - \theta_2)$$

is a simplest first-order formula that encodes all the possible values of the state variables x_1 and v_1 that are consistent with the power flow equations. Namely, an assignment $v_1 = a, \theta_1 = b$ is consistent with the equations if and only if $\varphi(a,b)$ is a true logical statement over real numbers, under standard interpretation of the function symbols.

The SMT problem asks solutions of logic formulas, in the sense that if some assignments to the free variables in the logic formula make the formula a true statement. Formally,

Definition 3 (SMT Problems): Let $\varphi(\vec{x})$ be an arbitrary $\mathcal{L}_{\mathbb{R}_\mathcal{F}}$-formula, where $\vec{x}$ are its free variables. The SMT problem for $\varphi(\vec{x})$ asks for one of the following answers:

- For some $\vec{a} \in \mathbb{R}$, $\varphi(\vec{a})$ is true;
- No assignment $\vec{a} \in \mathbb{R}$ makes $\varphi(\vec{a})$ true.

Naturally, SMT solvers implement algorithms for solving the SMT problems of logical formulas. When the $\mathcal{L}_{\mathbb{R}_\mathcal{F}}$-formula involve nonlinear real functions, the SMT problem become hard or undecidable. We can only aim to solve the δ-SMT problem for such formulas, which allows one-sided numerical errors bounded by δ. For small enough δ, the difference does not have practical effects for our application here, so we omit the discussion. More details of the δ-SMT problems for $\mathcal{L}_{\mathbb{R}_\mathcal{F}}$-formulas can be found in [3].

III. Logic-Based Vulnerability Analysis

A. Power Flow Equations

In steady state, the values of the state variables and measurements must satisfy the following equations, under AC state estimation.

Definition 4 (Power Flow Equations): In steady states, the active and reactive power flows on transmission lines between Bus i and j ($i \neq j$) satisfy the following equations:

$$\begin{array}{rcl} z^p_{ij} & = & h^p_{ij}(v_i, v_j, \theta_i, \theta_j) \\ z^q_{ij} & = & h^q_{ij}(v_i, v_j, \theta_i, \theta_j) \end{array}$$

where

$$h^p_{ij}(v_i, v_j, \theta_i, \theta_j) = v_i^2 g_{ij} - v_i v_j g_{ij} \cos(\theta_i - \theta_j) - v_i v_j b_{ij} \sin(\theta_i - \theta_j)$$

$$h^q_{ij}(v_i, v_j, \theta_i, \theta_j) = -v_i^2 b_{ij} - v_i v_j b_{ij} \cos(\theta_i - \theta_j) - v_i v_j g_{ij} \sin(\theta_i - \theta_j)$$

where g_{ij} and b_{ij} are constants for the conductance and susceptance of the transmission line.

Definition 5 (Power Injection Equations): The active and reactive power injections at each bus i are determined by its set of neighbors N_i,

$$N_i = \{i1, ..., ik\}$$

satisfy the following equations:

$$\begin{aligned} z^p_{ii} &= h^p_{ii}(v_i, \theta_i, v_{i1}, ..., v_{ik}, \theta_{i1}, ..., \theta_{ik}) \\ z^q_{ii} &= h^q_{ii}(v_i, \theta_i, v_{i1}, ..., v_{ik}, \theta_{i1}, ..., \theta_{ik}) \end{aligned}$$

where

$$h^p_{ii}(\vec{v}, \vec{\theta}) = v_i \sum_{j \in N_i} v_j(-g_{ij} \cos(\theta_i - \theta_j) - b_{ij} \sin(\theta_i - \theta_j))$$

$$h^q_{ii}(\vec{v}, \vec{\theta}) = v_i \sum_{j \in N_i} v_j(-g_{ij} \sin(\theta_i - \theta_j) + b_{ij} \cos(\theta_i - \theta_j)).$$

where g_{ij} and b_{ij} are constants for the conductance and susceptance of the transmission lines.

When we use logic formulas, it is important to distinguish between logic encodings and their *models*. An admissible model of a grid is an assignment to all its state and measurement variables that satisfy the power flow and injection equations.

Definition 6 (Admissible Model): Let G be a grid. An admissible model of G is a function

$$\sigma : X \cup Z \to \mathbb{R}$$

such that for every bus $i, j \in B(G)$, the power equations are true under assigned values $\sigma(v_{ij})$, $\sigma(\theta_{ij})$, $\sigma(z^p_{ij})$, and $\sigma(z^q_{ij})$.

Vulnerability analysis corresponds to finding admissible models that violate safety constraints imposed by the monitoring mechanisms in the grid, which we describe next.

B. State Estimation and Bad Data Monitoring

Power grids operators rely on state estimation algorithms to obtain estimated values of the state variables, through the values of measurements that come from meter readings.

State estimation is typically performed by weighted least square algorithms that find optimized estimation of state variables, assuming that the measurement errors are Gaussian. Although it is possible to encode the actual state estimation algorithms as logic formulas (a standard practice in program analysis), doing so is too costly and not necessary. Instead, it is sufficient to specify the relation between the measurement values and the estimated values of the state variables. They satisfy the power flow and injection equations up to bounded measurement errors, as is defined as follows:

Definition 7 (State Estimation with Bounded Errors): Let G be a power grid. The state estimation formula with ε-bounded errors is defined as

$$\begin{aligned} &\mathsf{est}_\varepsilon(\vec{z}, \vec{x}) := \\ &\quad \exists \vec{x}^o \Big(\bigwedge_{i,j \in B(G)} \Big(z^p_{ij} = h^p_{ij}(\vec{x}^o) \wedge z^q_{ij} = h^q_{ij}(\vec{x}^o) \Big) \\ &\qquad \wedge ||\vec{x} - \vec{x}^o|| < \varepsilon \Big). \end{aligned}$$

In words, the assignment to the state variables $\vec{x}$ and measurement variables $\vec{z}$ satisfy the state estimation formula iff that, with in some ε-bounded difference, they satisfy all the power flow and injection equations.

Note that the $\vec{z}$ and $\vec{x}$ variables are free, and thus the formula encodes the set of *all* state variable values that may be obtained by applying the state estimation algorithm on the measurement values $\vec{z}$. The $\vec{x}^o$ variables are bounded by the existential quantifiers, which denote the precise values that satisfy the actual equations, but not necessarily what the state estimation algorithm can obtain, because of the influence of numerical errors.

An important part of the state estimation process is the detection of sensor failure and measurement errors. Bad data monitoring mechanisms aim to identify such problems after state estimation. A typical method is to test the residue error in state estimation:

Definition 8 (Bad Data Monitoring): The residue is the difference between the measurements and the expected measurements computed from the estimated state variables:

$$\vec{r} = \vec{z} - \vec{h}(\vec{x})$$

Using different methods, one can choose a suitable threshold τ (for instance, through χ^2 test), such that if the residual is larger than the threshold,

$$||\vec{r}|| \geq \tau,$$

then the monitor flags bad data. Thus, we write

$$\mathsf{mon}_\tau(\vec{x}, \vec{z}) := \ ||\vec{z} - \vec{h}(\vec{x})|| < \tau$$

to encode the pairs $(\vec{z}, \vec{x})$ that are regarded as good data by the monitor, parameterized by τ.

C. Encoding False Data Injection

False data injection attacks replace the measurements $\vec{z}$ by artificial values $\vec{z}' = \vec{z} + \vec{a}$, such that $\vec{z}'$ can bypass the monitor and mislead the central controller. We now show how to encode false data injection vectors that can be successful, i.e., bypass the monitoring mechanisms.

Definition 9 (False Data Injection): Let G be a power grid with m measurement variables. Let est_ε and mon_τ be as specified above. Let $\vec{c}_z \in \mathbb{R}^m$ be a *constant vector* that encodes the correct reading of all measurements in G. A admissible measurement vector satisfies the following formula:

$$\mathsf{fdi}_{\varepsilon,\tau}(\vec{z}, \vec{c}_z) := \exists \vec{x} \Big(\mathsf{est}_\varepsilon(\vec{z}, \vec{x}) \wedge \mathsf{mon}_\tau(\vec{z}, \vec{x}) \wedge \vec{z} \neq \vec{c}_z \Big).$$

In words, although $\vec{z}$ is different from the actual measurement $\vec{c}_z$, $\vec{x}$ and $\vec{z}$ pass the state estimation and monitoring mechanisms. Note that $\vec{c}_z$ can be an arbitrary vector and does not include variables of the formula.

D. Constrained Measurements

We say a measurement is constrained, if its value is fixed within some range that is much smaller than the standard domain of the measurement variables. Such constraints may be imposed for different reasons. For instance, if a bus does not have generators or loads, then its power injection has to be 0. Also, a branch may be under external monitoring such that the power flow on it can not change by more than 5% of its standard value. It is straightforward to express these constraints as logic formulas on the measurements, which we use to define the formula

$$\mathsf{con}(\vec{z}).$$

For instance, a bus with no power injection needs to satisfy

$$\mathsf{con}(\vec{z}) := (z^p = 0 \wedge z_q = 0).$$

When measurements are constrained, the attacker can not modify them to arbitrary values. To launch an FDI attack, the constraints have to be taken into account, which typically requiring solving the power flow/injection equations involved. Consequently, constrained measurements are a main source of complexity for our analysis, as we will see in Section IV.

E. Walking through a 3-Bus Example

Putting everything together, we use a small 3-Bus example to demonstrate how the encoding works.

$$\begin{aligned} B &= \{0,1,2\} \\ E &= \{(0,1),(1,2),(2,0)\} \\ \vec{x} &= [\theta_0\ \theta_1\ \theta_2\ v_0\ v_1\ v_2]^T \\ \vec{z} &= [z^p_{01}\ z^p_{12}\ z^p_{20}\ z^p_{00}\ z^p_{11}\ z^p_{22}\ z^q_{01}\ z^q_{12}\ z^q_{20}\ z^q_{00}\ z^q_{11}\ z^q_{22}]^T \end{aligned}$$

The z^p and z^q variables are calculated through the power flow and injection equations. Here is an example. Note from E that the neighbors of bus 1 is bus 0 and bus 2, i.e., $N_1 = \{0,2\}$. We have:

$$\begin{aligned} z^p_{11} &= h^p_{11}(\vec{x}) \\ &= v_1 v_0(-g_{10}\cos(\theta_1-\theta_0) - b_{10}\sin(\theta_1-\theta_0)) \\ &\quad -v_1 v_2(g_{12}\cos(\theta_1-\theta_2) + b_{12}\sin(\theta_1-\theta_2)) \end{aligned}$$

Now, to define the formulas for the state estimators and monitors, we let h_i be the functions on the righthand side for each z_i. The state estimation formula is then defined as:

$$\mathsf{est}_\varepsilon(\vec{z},\vec{x}) :=$$
$$\exists \vec{x}^o \left(||\vec{x}-\vec{x}^o|| < \varepsilon \wedge \bigwedge_{(i,j)\in E(G)} \left(z^p_{ij} = h^p_{ij}(\vec{x}) \wedge z^q_{ij} = h^q_{ij}(\vec{x}) \right) \right.$$
$$\left. \wedge \bigwedge_{i\in B(G)} \left(z^p_{ii} = h^p_{ii}(\vec{x}) \wedge z^q_{ii} = h^q_{ii}(\vec{x}) \right) \right)$$

The monitoring mechanism simply tests

$$\mathsf{mon}_\tau(\vec{x},\vec{z}) := \ ||\vec{z} - \vec{h}(\vec{x})|| < \tau.$$

The measurements may be subject to additional constraints, such as

$$\mathsf{con}(\vec{z}) := |z^p_{12}| < 0.5 \wedge |z^q_{12}| < 0.5 \wedge z_{11} = 0.$$

In all, we have that the formula

$$\mathsf{fdi}_{\varepsilon,\tau}(\vec{z},\vec{c}_z) := \exists \vec{x} \left(\mathsf{est}(\vec{z},\vec{x}) \wedge \mathsf{mon}_\tau(\vec{x},\vec{z}) \wedge \mathsf{con}(\vec{z}) \wedge \vec{z} \neq \vec{c}_z \right)$$

defines the set of all possible measurement $\vec{z}$ that can pass the τ-monitoring mechanism, while different from the correct state estimation $\vec{c}_z$. This formula is an SMT formula. Thus, if an SMT solver can obtain solutions to this formula, then we have found attack vectors that can bypass the monitoring mechanisms and mislead the grid operations.

IV. Symbolic Propagation

A key fact, or weakness, of the standard bad data monitoring mechanism is that the monitoring on each measurement relies only on the values of state variables in the power flow equation. Different measurements are only indirectly linked through their shared state variables. As a result, attackers can *localize* attacks within sub-grids, by making sure that the values of the state variables *on the boundaries* of the sub-grids remain unchanged. In other words, the boundary of a sub-grids decouples measurements that are inside and outside the sub-grid. This observation is also the basis of existing analysis methods such as in [5].

If all measurements are unconstrained, then to hide an FDI attack on a particular measurement z, one only needs to modify a small set of additional measurements – those that share state variables with z in the power flow/injection equations. Topologically, the relevant measurements are all contained in the immediate neighborhood of z. However, if any of these measurements are subject to additional constraints, such that their values can not be changed arbitrarily, then hiding the attack typically requires a larger "sub-grid mask." The work in [5] has studied a special case: when a bus is not connected to loads or generators, the measurements in its neighborhood become coupled. Thus there is a *rippling effect* on the changes that need to be made to hide the attacks.

In this section we formalize this line of analysis. We define and prove properties about the notion of localized attacks under constraints. We propose algorithms that incrementally enlarge a sub-grid under analysis, based on whether the measurements are constrained on its boundary. Using this algorithm, we can find FDI attack vectors (targeting specific buses or branches) that can be hidden by changing measurements only in a sub-grid of minimal size.

A. Monitoring Mechanisms Are Local

Consider the power flow/injection equations in Definition 4 and 5. They are all of the form

$$\vec{z} = \vec{h}(\vec{x})$$

where $\vec{z}$ are the measurement variables and $\vec{x}$ are the state variables. The monitoring mechanism checks for residue errors

$$||\vec{z} - \vec{h}(\vec{x})|| < \tau.$$

In both equations, the value of z does not rely on any other measurement, or any state beyond its immediate neighbors. The consequence is that the monitoring mechanism is entirely local. For instance, consider three buses i, j, and k, with branches (i, j) and (j, k). The measurements on both branches satisfy

$$\begin{aligned} \vec{z}_{ij} &= h_{ij}(\vec{x}_i, \vec{x}_j) \\ \vec{z}_{jk} &= h_{jk}(\vec{x}_j, \vec{x}_k) \end{aligned}$$

which only share the state variables $\vec{x}_j$. Suppose $\vec{x}_j$ is fixed to some value $\vec{x}_j = \vec{a}$. Then, assuming that there are no other constraints on the measurements (in particular, the power injection on bus j is not constrained), then we can change $\vec{z}_{ij}$ and $\vec{z}_{jk}$ to arbitrary values, as long as those values are consistent with *some* $\vec{x}_i$ and $\vec{x}_k$. In other words, the measurements $\vec{z}_{ij}$ and $\vec{z}_{jk}$ are decoupled, in the sense that the change on one of them does not require the change on the other one. In general, attacks on measurements can be *hidden* in their neighborhood, which form a sub-grid, as long as the estimated values of the state variables on the boundary of the sub-grid can remain unchanged.

B. Attack Localization

A sub-grid of a grid is simply a subgraph together with all the state and measurement variables on it. Namely,

Definition 10 (Sub-Grid): Let $G = \langle V, X^v, X^\theta, Z^p, Z^q \rangle$ be a power grid. A sub-grid of G is a power grid $\bar{G} = \langle \bar{V}, \bar{X^v}, \bar{X^\theta}, \bar{Z^p}, \bar{Z^q} \rangle$ that satisfies:

$$\begin{aligned} &\bar{B} \subseteq B, \bar{E} = \{(i,j) \in E : i, j \in \bar{B}\}, \\ &\bar{X^v} = \{v_i \in X^v : i \in \bar{B}\}, \bar{X^\theta} = \{v_i \in X^\theta : i \in \bar{B}\}, \\ &\bar{Z^p} = \{z^p_{ij} \in Z^p : i, j \in \bar{B}\}, \bar{Z^q} = \{z^q_{ij} \in Z^q : i, j \in \bar{B}\}. \end{aligned}$$

The boundary of $\bar{G}$ is defined as

$$\partial\bar{G} = \{i \in B(\bar{G}) : \text{ There exists } j \in B(G) \setminus B(\bar{G}) \text{ such that } (i,j) \in E(G).\}$$

Namely, $\partial\bar{G}$ is the set of all buses in the sub-grid that are connected to some bus outside of $\bar{G}$. We also define the set of boundary branches, which are branches that are connected to at least one boundary bus, i.e.:

$$\partial E(\bar{G}) = \{(i,j) \in E(G) : \ i \in \partial\bar{G} \text{ or } j \in \partial\bar{G}\}.$$

Also, we define the interior measurement of $\bar{G}$ to be the measurements in $\bar{G}$ that are not on the boundary buses or boundary branches of $\bar{G}$.

Attacks on sub-grids can be hidden in its neighborhood, or *localized*, in the following sense.

Proposition 1 (Attack Localization): Let G be a grid and let σ be an admissible assignment to all its variables. Let $\bar{G}$ be a subgraph of G. Suppose σ' is another assignment on G that satisfies the following three conditions:

1) σ and σ' agree on all the state and measurement variables outside of $\bar{G}$, i.e.:

$$\sigma|_{G \setminus \bar{G}} = \sigma'|_{G \setminus \bar{G}},$$

where $\sigma|_G$ means the restriction of σ on G.
2) σ and σ' agree on the state variables on the boundary of $\bar{G}$, i.e.:

$$\forall i \in \partial B(\bar{G}), \sigma(v_i) = \sigma'(v_i) \text{ and } \sigma(\theta_i) = \sigma'(\theta_i).$$

3) $\sigma'|_{\bar{G}}$ is an admissible model for the subgrid $\bar{G}$.

Then σ' is also an admissible model for G.

To verify this fact, one only needs to check all the power flow equations, as described in the previous section. As a result, if an attacker only changes the measurement variables within a sub-grid while keeping the states on its boundary unchanged, then the attack is hidden in the sub-grid.

Now the question is, how large is the sub-grid that can hide an attack? In the case of unconstrained measurements, this is simple to characterize.

Definition 11 (Measurement Closure): Let z be a measurement in a grid G. Write $X(z)$ to denote the state variables that appear in the power flow/injection equations for z. The measurement closure of z is defined as

$$MC(z) = \{z' \in Z(G) : X(z) \cap X(z') \neq \emptyset\}.$$

Namely, the measurement closure of z is the set of all measurements that share state variables with z in their power flow/injection equations.

Definition 12 (Sub-Grid Masks): Let G be a grid and Z^a be a set of measurements in G. A sub-grid mask on Z^a is a sub-grid $\bar{G}$ such that

$$\bigcup_{z \in Z^a} MC(z) \subseteq Z(\bar{G}).$$

Namely, the measurement closure of all measurements in Z^a has to be contained in the sub-grid mask. The minimal sub-grid mask is then defined as:

$$MS(Z^a) = \bigcap \{\bar{G} \subseteq G : \bigcup_{z \in Z^a} MC(z) \subseteq Z(\bar{G})\},$$

which is simply the intersection of all sub-grid masks.

In the case of a single bus or branch, the minimal sub-grids are as shown in Figure 1. For a bus, the minimal sub-grid includes its immediate neighbor buses. For a branch, it includes the terminal buses of the branch and the neighbors of these buses. Note that the minimal sub-grid mask of a branch is different from the minimal sub-grid that contains it, which is simply the branch itself with the buses that it connects.

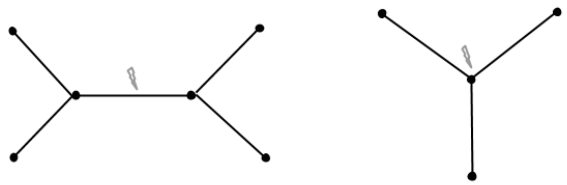

Fig. 1. Minimal sub-grids that can hide attacks. Left is the case of attack on a branch. Right is the case of attack on a bus.

C. Symbolic Propagation

Given that local sub-grids can hide attacks, we should start our analysis on the minimal sub-grid mask over the targeted measurements. In the unconstrained case, this is already sufficient – we only need to find the consistent values of all measurements in the sub-grid, and keep the state variables on the boundary of the sub-grid unchanged. However, if some measurements are constrained, we may need to incrementally expand the sub-grid.

We now describe the full algorithm. Suppose the goal is to attack some measurement z. We perform the analysis in the following way:

1) We start with the minimal sub-grid mask over z, $\bar{G} = MS(z)$.
2) We use an SMT solver to search for an FDI attack vector that manipulates the measurements with in $\bar{G}$, while keeping the value of the state variables on the boundary unchanged. Note that the attack vector needs to respect the constraints on the measurements in $\bar{G}$, if there is any. The full formula is given below in Definition 13. If we have successfully found an attack vector, then terminate the search.
3) If the formula in the previous step is not satisfiable, we choose a constrained measurement z on either the boundary buses or the boundary branches, and expand $\bar{G}$ to be $\bar{G} \cup MS(z)$, and repeat from the previous step. Note that this set of constrained measurements is always nonempty, because otherwise we would have found a solution in the previous step.

We now focus on two details in the algorithm: the formula we solve in each round, and the termination criteria.

We say an FDI attack vector is masked by a sub-grid, if it does not change the values of the state variables on the boundary of the sub-grid. Formally:

Definition 13 (Masked FDI Attacks): Let $\bar{G}$ be a sug-grid of G. Let σ be an admissible assignment of $\bar{G}$. For each bus $i \in \partial\bar{G}$, we write $\sigma(v_i) = a_i^v$ and $\sigma(\theta_i) = a_i^\theta$ to denote the values that are assigned to the state variables on Bus i under σ. The following formula defines the set of FDI attack vectors that are masked by $\bar{G}$ and σ:

$$\mathsf{fdi}_{\varepsilon,\tau}^{\bar{G},\sigma}(\vec{z}) := \exists\vec{x}\left(\mathsf{est}_\varepsilon(\vec{z},\vec{x}) \wedge \mathsf{mon}_\tau(\vec{z},\vec{x}) \wedge \mathsf{con}(\vec{z}) \wedge \bigwedge_{i\in\partial\bar{G}} (x_i^v = a_i^v \wedge x_i^\theta = a_i^\theta)\right)$$

where x_i^v and x_i^θ are components of $\vec{x}$. est_ε, mon_τ, and con follow the definitions in Section III. In words, the formula defines the set of measurements that can pass the monitoring mechanism, satisfy all the constraints on the measurements, and most importantly, can not be distinguished from σ by monitoring on measurements outside $\bar{G}$.

To describe the termination criteria, we need to define one more structure on the grid.

Definition 14 (Elastic Boundary Buses): Let i be a bus on $\partial\bar{G}$. We say i is an elastic boundary bus in $\bar{G}$ if

1) The measurements on i are not constrained.
2) For any branch $(i,j) \in E(\bar{G})$, the measurements on (i,j) are not constrained.

The importance of elastic boundary buses is that they define the boundary of a sub-grid mask.

Proposition 2: Suppose all boundary buses of $\bar{G}$ are elastic, then $\bar{G}$ is a sub-grid mask for its interior measurements.

Proof: Suppose z is an interior measurement on a bus i that is connected to a boundary bus j. Since j is elastic, there is no constraint on the branch (i,j) or the bus j. Consequently, for any value of the state variable on i, we can set the measurements on (i,j) to be a consistent value, without changing the state variable values on the boundary bus j. Following Proposition 1, any attack on i can be localized. ■

Consequently, when we have a sub-grid with elastic boundaries, we can terminate the propagation and obtain an attack vector. This serves as the termination criteria for the symbolic propagation algorithm. Algorithm 1 describes the full algorithm.

Algorithm 1: Symbolic Propagation

input : Power grid G, ε, τ, target measurement z_0.
output: An attack vector.

```
Ḡ ← the minimal sub-grid mask MS(z_0);
σ ← vector of state estimation on the relevant variables;
while not all boundary buses are elastic do
    if fdi^{Ḡ,σ}_{ε,τ}(z⃗) is satisfiable then
        return solution of the formula;
    else
        i ← a non-elastic boundary bus in B(Ḡ);
        Ḡ ← Ḡ ∪ MS(Z(i)) ;
    end
end
```

V. Experiments

Standard benchmarks for power grid analysis are the IEEE 14-Bus, 30-Bus, 157-Bus, and 300-Bus benchmarks [1]. To

best evaluate the proposed techniques, we summarized the parameter characteristics in the benchmarks and generated a large number of random grids, so that we can easily control different parameters such as degrees, measurements under constraints, etc. All experiments are conducted on a machine of 1.8Ghz Intel i5 core and 8GB RAM.[1]

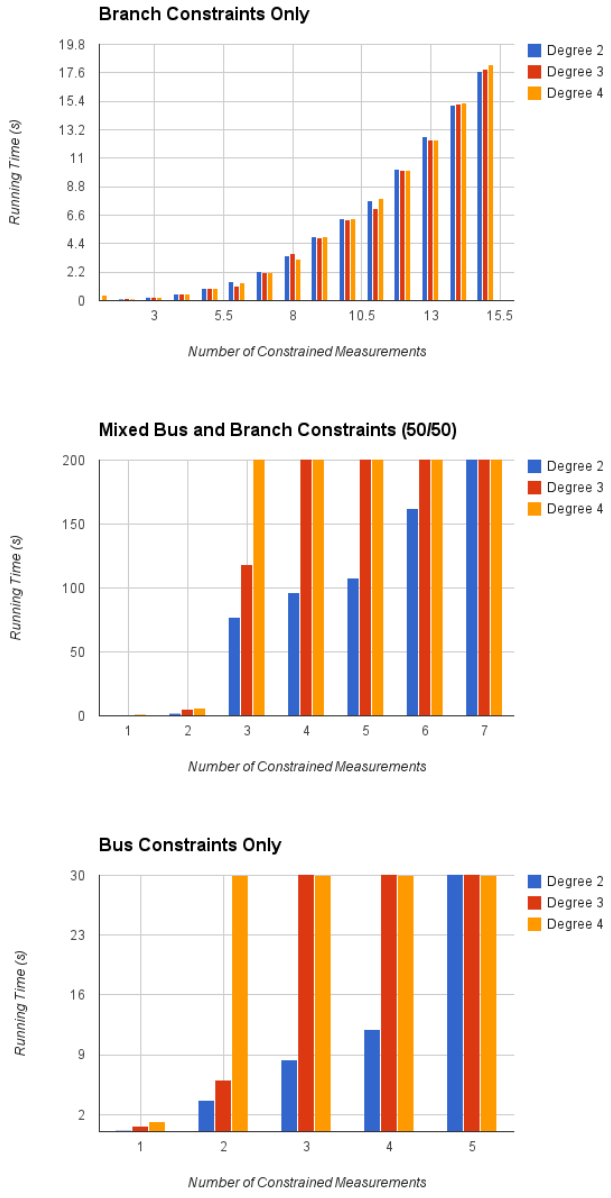

Fig. 2. Effect of Constraint Types on Attack Difficulty

First of all, because we analyze localized attacks, the raw size of the full grid does not affect scalability. In Table I we show that the running time on a 200-bus grid and a 500-bus grid is not distinguishable if all other parameters are the same. Thus we perform most of the experiments on random sub-grids within 500-bus grids.

[1]The test code and benchmarks are available at `http://scungao.github.com/powersystem`

We observe the following facts in the experiments, which can be reflected in the diagrams in Figure 2 to Figure 3.

1. Experiments show that there is a big difference between constraining measurements on buses and branches. In Figure 2, we show the running time of three cases: all constrained measurements are on branches, on buses, or a mixture of the two. The solver can scale well when all constrained measurements are on branches. The hardest problems come from when all constrained measurements are on buses. The reason for this result is that the constraints on buses involve all the state and measurement variables on its neighbor buses, while to handle a branch we only need to involve its terminal buses. It is especially difficult to find attack vectors when all the constrained buses are immediate neighbors, in which case all nonlinear constraints share many variables. This suggests that, from a defense perspective, it is more effective to apply additional monitoring on buses, especially on buses that directly connect to each other.

T	D	#K	#M	#CM	#V	#NL	TIME(s)	δ
U	2	1	8	4	8	30	0.014	0.1
U	2	1	8	4	8	30	4.851	0.01
U	2	3	24	12	18	100	0.028	0.1
U	2	3	24	12	18	100	5.012	0.01
U	3	1	28	4	10	40	1.291	0.1
U	3	2	36	8	18	90	25.875	0.1
U	3	2	36	8	18	90	60.172	0.01
U	3	3	48	12	26	140	310.212	0.1
U	3	3	48	12	26	140	912.318	0.01
U	4	1	8	4	12	50	0.362	0.1
U	4	2	54	8	22	110	timeout	0.1
M	2	2	8	2	5	9	2.093	0.1
M	2	4	20	8	14	36	96.517	0.1
M	2	5	28	10	32	90	108.075	0.1
M	2	6	32	14	43	135	162.204	0.1
M	3	2	34	4	18	60	5.331	0.1
M	3	3	42	8	24	110	118.814	0.1
M	4	4	48	12	34	160	timeout	0.1
R	2	1	8	2	5	9	0.047	0.01
R	3	4	32	8	14	36	0.541	0.01
R	3	10	60	20	32	90	6.251	0.01
R	3	15	90	30	43	135	7.069	0.01
R	4	4	32	8	14	36	0.821	0.01
R	5	4	32	8	14	36	1.108	0.01

TABLE I

EXPERIMENTAL DATA. "T" IS THE TYPE OF CONSTRAINTS ON MEASUREMENTS: "U" MEANS "BUS ONLY", "M" MEANS "MIXED BUSES AND BRANCHES"; "R" MEANS "BRANCH" ONLY. "D" IS THE DEGREE OF BUSES IN THE GRID. "#K" IS THE NUMBER OF PROPAGATION STEPS."#M" IS THE NUMBER OF MEASUREMENTS INVOLVED IN THE SUB-GRID. "#CM" IS THE NUMBER OF CONSTRAINED MEASUREMENTS; "#V" IS THE NUMBER OF VARIABLES IN THE SMT FORMULA; "#N" IS THE NUMBER OF NONLINEAR CONSTRAINTS IN THE FORMULA; "δ" IS THE NUMERICAL ERROR IN SOLVER.

2. The degree of the buses affects running time greatly, but only when measurements on buses are constrained. The reason is as follows. When the constrained measurements are on branches only, then the degree of each bus does not matter, because each power flow equation only involves two

buses. When buses are constrained, then higher degree results in much more complex formulas. We can see from Figure 2 that the running time grows roughly exponentially with respect to the degree when buses are constrained, while no such effects occur when only branches are constrained. Note that in practice, power grids typically contain buses of degree 2 to 3. Thus the running time here indicates applicability to realistic problems.

3. Running time grows as a monotonic function of propagation depth. In each propagation depth, more constraints and variables are added to the formula. The size of the formula never decreases (so there is no backtracking in search). In Figure 3, we see that the number of nonlinear constraints increases more rapidly when the degree of the buses is high.

4. The SMT solver we use implements δ-complete decision procedures that allow δ-bounded numerical errors in its solutions. The running time of the solver is affected by the delta of choice. In Figure 3 we see how running time increases with respect to delta. In fact, this figure shows the running time on formulas with branch only constraints. In formulas with bus constraints, the run time increases exponentially with respect to δ.

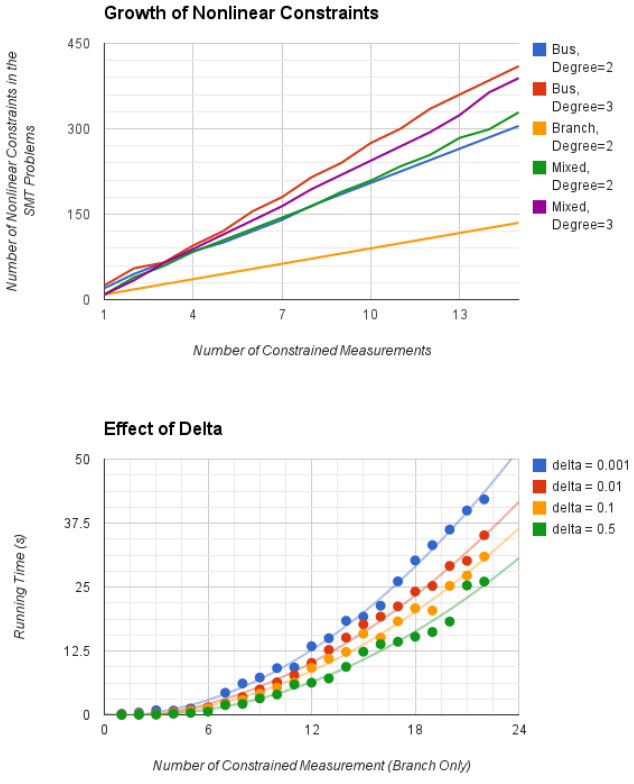

Fig. 3. Size of Nonlinear Constraints and Effects of Delta

Effects of Numerical Errors in the Solver. A main underlying principle for solving the nonlinear formulas is that one needs to aim for delta-decisions. While this is imprecision in the solver, it becomes a valuable feature in the vulnerability analysis context. When a formula is returned as delta-satisfiable, it means that the condition is solvable under some delta-bounded perturbations of the original formula. Since measurements contain errors, these answers take into account of such imprecision in measuring. From each solution, we can also calculate the amount of perturbation that an attacker imposes to compromise the system and take that into account when designing the defense mechanism. These are problems that would not be approachable by an exact solver, let along the limited scalability of exact solvers in this context.

VI. Conclusion

We demonstrated that it is possible to perform automated vulnerability analysis of AC state estimation under constrained false data injection attacks. We showed how to encode the problems as SMT problems and solve them with efficient nonlinear solvers. We proposed the new technique symbolic propagation to localize the analysis.

There are many possible extensions of this work. The underlying technique for vulnerability analysis could also be applied to solving distributed power flow and distributed state estimation. We also aim to develop analysis method that combine program analysis and the proposed methods, to defend against attacks similar to the Stuxnet attack.

References

[1] `https://www.ee.washington.edu/research/pstca/`.

[2] G. Dán and H. Sandberg. Stealth attacks and protection schemes for state estimators in power systems. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 214–219. IEEE, 2010.

[3] S. Gao, J. Avigad, and E. M. Clarke. Delta-complete decision procedures for satisfiability over the reals. In B. Gramlich, D. Miller, and U. Sattler, editors, *IJCAR*, volume 7364 of *Lecture Notes in Computer Science*, pages 286–300. Springer, 2012.

[4] S. Gao, S. Kong, and E. M. Clarke. dReal: An smt solver for nonlinear theories over the reals. In *Automated Deduction–CADE-24*, pages 208–214. Springer, 2013.

[5] G. Hug and J. A. Giampapa. Vulnerability assessment of ac state estimation with respect to false data injection cyber-attacks. *Smart Grid, IEEE Transactions on*, 3(3):1362–1370, 2012.

[6] O. Kosut, L. Jia, R. J. Thomas, and L. Tong. Malicious data attacks on the smart grid. *Smart Grid, IEEE Transactions on*, 2(4):645–658, 2011.

[7] D. Kroening, R. Bryant, and O. Strichman. *Decision procedures: an algorithmic point of view*. Springer Science & Business Media, 2008.

[8] R. Langner. Stuxnet: Dissecting a cyberwarfare weapon. *Security & Privacy, IEEE*, 9(3):49–51, 2011.

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

[10] M. A. Rahman and E. Al-Shaer. A formal model for verifying stealthy attacks on state estimation in power grids. In *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*, pages 414–419. IEEE, 2013.

[11] M. A. Rahman and H. Mohsenian-Rad. False data injection attacks against nonlinear state estimation in smart power grids. In *Power and Energy Society General Meeting (PES), 2013 IEEE*, pages 1–5. IEEE, 2013.

[12] A. Teixeira, S. Amin, H. Sandberg, K. H. Johansson, and S. S. Sastry. Cyber security analysis of state estimators in electric power systems. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5991–5998. IEEE, 2010.

[13] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson. Revealing stealthy attacks in control systems. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 1806–1813. IEEE, 2012.

[14] L. Xie, Y. Mo, and B. Sinopoli. Integrity data attacks in power market operations. *Smart Grid, IEEE Transactions on*, 2(4):659–666, 2011.