Toward the Science of Industrial Control Systems Security and Resiliency

Mohammad Ashiqur Rahman and Ehab Al-Shaer

Abstract The supervisory control and data acquisition (SCADA) system is the major industrial control system (ICS), which is responsible for collecting data from end devices, analyzing data, and managing the system efficiently by sending necessary control commands to the corresponding end devices. Unlike traditional cyber networks, a SCADA system consists of heterogeneous devices that communicate with one another under various communication protocols, physical media, and security properties. Failures or attacks on such networks have the potential of data unavailability and false data injection causing incorrect system estimations and control decisions leading to non-optimal management or critical damages of the system. This chapter provides a theoretical baseline for assessing the security and resiliency of ICS by presenting two formal frameworks, one for security analysis and one for resiliency analysis, considering smart grid SCADA systems. These frameworks take smart grid configurations and organizational security or resiliency requirements as inputs, formally model configurations and various security properties, and verify the dependability of the system under potential attacks or contingencies. The execution of each of these frameworks is demonstrated on an example case study.

Introduction

Many cyber-physical systems (CPSs) like smart power grids, transportation systems, and water treatment plants are identified as critical infrastructures (CIs) due to their national importance. Secure and dependable operations of these infrastructures are extremely important. One or more industrial control systems (ICSs) are often found in a CPS, which are responsible for optimally and efficiently managing the system in

M. A. Rahman (✉)
Florida International University, Miami, FL, USA
e-mail: marahman@fiu.edu

E. Al-Shaer (✉)
University of North Carolina at Charlotte, Charlotte, NC, USA
e-mail: ealshaer@uncc.edu

C. Rieger et al. (eds.), *Industrial Control Systems Security and Resiliency*, Advances in Information Security 75, https://doi.org/10.1007/978-3-030-18214-4_4

real-time. The supervisory control and data acquisition (SCADA) system is the most important kind of ICS, which is responsible for monitoring and controlling dispersed assets by gathering and analyzing real-time data from remote (field) devices. Typical ICS operations include automated control loops, human machine interfaces (HMIs), and remote diagnostics and maintenance utilities.

In order to promote connectivity and remote access capabilities among corporate business systems, information technology (IT) is now increasingly used in ICSs, which escalates the possibility of cyber security vulnerabilities and incidents. Although there are some similarities between the characteristics of ICS and that of traditional IT systems, they differ in many places, especially due to the simultaneous existence of physical components and network components, along with different industrial communication protocols. That is why the vulnerabilities and threats, as well as the security requirements, of an ICS are often different from that of traditional IT systems. As such, it is important to develop automated security and resiliency analytics specifically for ICSs.

In this chapter, two formal frameworks – one for security analysis and the other for resiliency analysis – are presented providing a theoretical base for assessing the security and resiliency of ICS. These frameworks automatically and provably analyze the security and resiliency of the SCADA system, particularly, in terms of the data acquisition for executing control operations in smart grids. The frameworks take necessary SCADA configurations and security/resiliency requirements, formally model the analytics, and solve the models to verify the system with respect to the given security or resiliency specifications. The formal models are solved using state-of-the-art logical solvers. Each framework provides threat vectors, specifying when the security (or resiliency) requirements fail under the attack model. The unsatisfiable outcome can certify that the system is secure (or resilient) against the attack model. These frameworks can allow a grid operator to understand a SCADA system's resiliency, as well as to fix the system, by analyzing the threat vectors.

State of the Art of Research, Challenges, and Solutions

With the rise of cyber-warfare, the secure and dependable operation of a smart grid carries utmost importance. However, due to its many cyber and physical couplings, realizing the extent of security and resiliency of the system is challenging.

Supervisory Control and Data Acquisition Systems

An example topology of a SCADA system is shown in Fig. 1. Typical SCADA operations include automatic and human control loops, remote diagnostics, and maintenance utilities. There are also various kinds of physical devices, such as SCADA control servers or master terminal units (MTUs), remote terminal units (RTUs), programmable logic controllers (PLCs), intelligent electronic devices

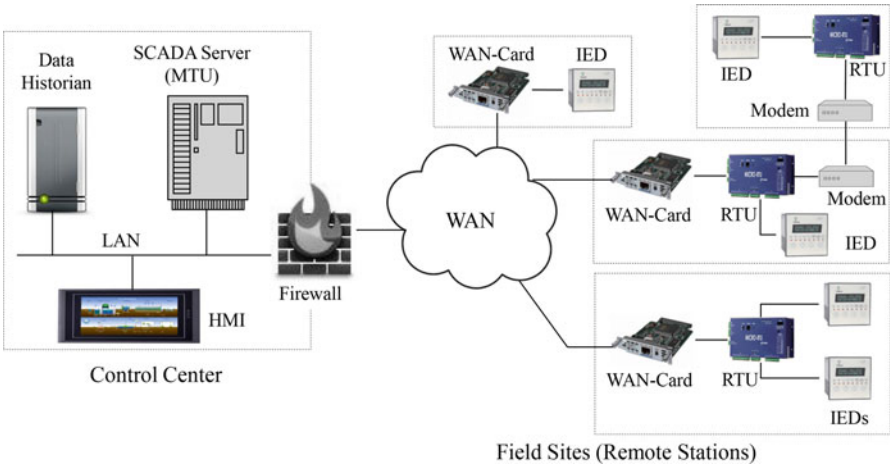

Fig. 1 An example of the SCADA network topology

(IEDs), human machine interfaces (HMIs), data historians, etc. IEDs, RTUs, and PLCs are considered as field or end devices, while the other devices reside in the control center. In addition to these control components, there are different network components, such as communications routers, modems, and remote access points. These components usually use ICS protocols like Modbus, DNP3, or IEC 61850 variants for communicating with one another.

The SCADA control server/MTU takes the sensor measurements from field devices through the power network and sends the control commands to them after analyzing the data using the same infrastructure. Control decisions are optimally made by the energy management system (EMS) by running several interdependent control modules or routines, namely, state estimation (SE), topology processor (TP), optimal power flow (OPF), contingency analysis (CA), and automatic generation control (AGC) [1, 2]. Executions of these EMS control routines are actively dependent on the data acquisition from the field devices. Among these modules, state estimation is the core component. Its function is to compute the unknown state variables of the power system from the sensor measurements received through the SCADA system. The output of state estimation is used in other control mechanisms to operate the grid optimally with respect to the generation cost and the physical safety of the grid. Figure 2 presents the core EMS modules and the interdependency among them according to the data flow.

Potential Cyber-Threats on SCADA

The increasing use of IT in smart grids escalates the possibility of cyber security vulnerabilities and incidents, as these systems have not been built taking security into consideration in the first place. The inherent complexity associated with

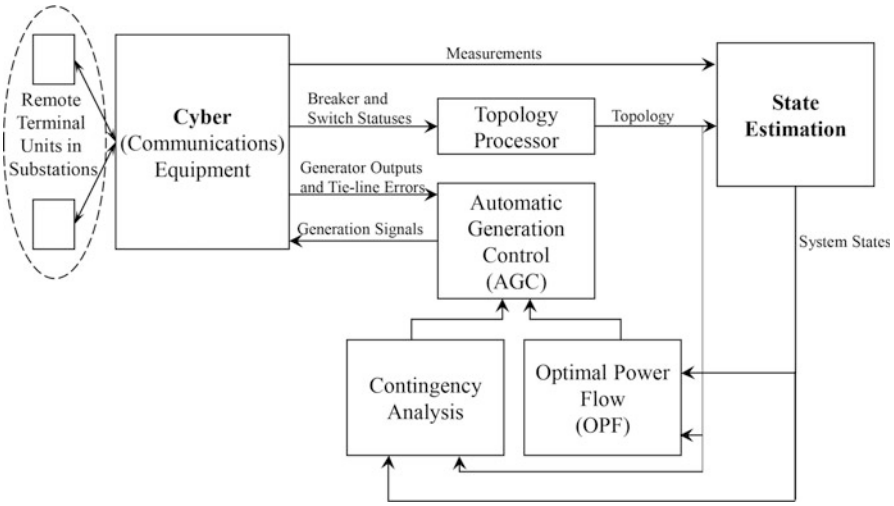

Fig. 2 A simplified EMS architecture. (Adapted from Ref. [1])

integrating different heterogeneous and legacy systems in SCADA systems significantly increases the potential of security threats, which can cause massive and devastating damage. There are two main causes of threats [3]. The first is the *misconfiguration or the lack of security controls* that can cause inconsistency, unreachability, broken security tunnels, and many other security breaches. The second is the *weakness or absence of resiliency controls* that can lead to cascaded failures in contingencies or cyber-attacks. As an example of cyber-attacks, denial of service (DoS) attacks can make one or more field devices unreachable or unavailable to or from the rest of the system.

The main purpose of a SCADA system is to deliver measurement data from the field or physical devices (meters/sensors) to the provider's side (control center or utility) while delivering control commands from the provider's side to the field/physical devices. To achieve successful data delivery, reachability must hold between the sender and the receiver. Inconsistencies in communication protocols or authentication/encryption parameters of the communicating devices may cause failed data transmission leading to service disruptions. In addition, data should be delivered such that it satisfies end-to-end integrity. The violation of this requirement not only can cause incorrect estimation of the system but may also launch malicious control commands toward physical devices. This scenario becomes worse in the case of contingencies, when some IEDs or RTUs fail due to technical errors or cyber-attacks, as there may not be enough (secured) measurements received by the control server to observe the whole system accurately.

Research Challenges and Formal Frameworks

The correct functioning of a SCADA system stands on consistent and secure execution of tasks in time. The safe security configuration depends not only on the local device parameters but also on the secure interactions and flows of these parameters across the network including SCADA control mechanisms. There are a significant number of logical constraints on configuration parameters of many SCADA devices, which need to be satisfied to ensure safe and secure communications among SCADA components while keeping the system stable during contingencies. The adversary must be modeled with respect to practical properties so that a realistic picture of the system's resiliency can be realized. The attack model needs to be flexible enough to consider a wide range of different attack scenarios. Implementing these security and resiliency controls in a scalable and provable manner is one of the major challenges in smart grid security modeling.

To address this grand challenge, formal frameworks are proposed that can allow energy providers to objectively assess and investigate SCADA security configurations to identify potential resiliency threats and to enforce smart grid operational and organizational security requirements [4–6]. These works primarily model secured communication, potential contingencies, and security/resiliency properties and provide an efficient solution to analyze the security and resiliency of the system by identifying the threat vectors that negate the security and resiliency requirements. The frameworks are designed as a constraint satisfaction problem. Although these frameworks include the formalizations of a limited set of constraints that are important for proper communication, an important feature of these frameworks is their easy extensibility. For further properties, one just needs to add formalizations for corresponding constraints.

Threat Analysis Architecture

The basic architecture for the formal threat analytics is shown in Fig. 3. The threat analyzer takes different inputs, including the bus/SCADA topology information (including connectivity between buses/SCADA devices), device configuration (including encryption and authentication properties, recorded measurements, etc.), and the control functions and corresponding data requirements. The analyzer also takes an attack model, along with a set of adversary attributes, as input. While the attack model specifies different kinds of attacks (e.g., false data injection attacks on measurements as well as topology statuses), the adversary attributes include, but are not limited to, (i) the attacker's knowledge of the system (e.g., the measurements location, topology, etc.); (ii) the attacker's capabilities for accessing and manipulating/compromising specific cyber/physical entities for launching attacks; and (iii) the attacker's resources, such as the potential for corrupting different physical entities at time-to-launch coordinated attacks. Based on this input, the security analyzer derives

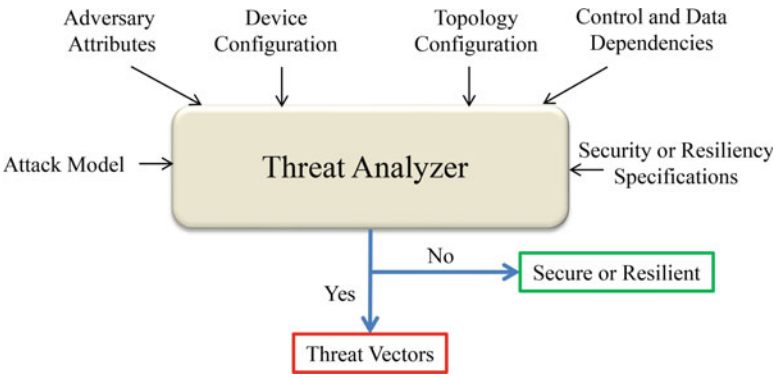

Fig. 3 The architecture of the threat analytics

the invariant properties of the physical system model, the attack model and potential evasion, and the inter-module dependencies and verifies feasible attacks/threats and provides corresponding threat vector(s) (i.e., states/measurements to be compromised or devices to be failed). The threat analysis modeling considers the interaction among the different EMS modules (i.e., SE, OPF, TP, CA, and AGC) in a way that can verify where and how an attack can be launched (e.g., SE and TP) and how far the attack (i.e., its impact) can percolate.

Formal Approach Characteristics

The formal security and resiliency frameworks [4–6] embrace the following key ideas, in general:

Security-Centric Modeling A security-centric model of the power system is one that explicitly integrates security properties into the core physical model. The model allows measurements/statuses received/sent from/to field devices through cyber-communication to be intentionally corrupted by adversaries. The model incorporates security properties into the entire measurement set by defining a set of variables to denote the cyber-physical and adversarial properties of the system.

Comprehensive Modeling of Adversary Attributes The adversary or attack attributes are expressed in terms of the knowledge about the target system, accessibility to the system, attack objectives, the resources to launch an attack, etc. The modeling needs to construct a formal description to map the changes that occurred on physical properties with respect to attacks. Modeling interdependencies among EMS modules will allow mapping of how the effect of an attack on a module can percolate to another module.

Unified Framework to Identify Coordinated Attack This step unifies the physical properties, attack model, and adversarial components into a comprehensive

model for the entire complex system. The resulting unified formal model, expressed as a set of constraint satisfaction verification problems, is solved for the potential attacks. Interdependency modeling helps identify coordinated attacks (e.g., coordinating false data injections on measurements, as well as topology statuses).

Efficient Solutions by Satisfiability Modulo Theories (SMT) Proposed models are formalized using SMT. Over the past 10 years, SMT has become the core engine behind many practical tools for software and hardware analytics for static software analysis, dynamic symbolic execution, model-based testing, and automated synthesis and planning [7]. SMT is a constraint satisfaction problem solver for logical formulas with respect to combinations of background theories (e.g., uninterpreted functions, linear and non-linear arithmetic, difference logic, etc.). The SMT formula can be considered as an instance of the Boolean satisfiability problem (SAT) in which some of the binary variables are replaced by binary-valued predicates over a set of non-binary variables. SMT solvers can determine the satisfiability of formulas that contain thousands of variables and constraints [7].

Formal Framework for SCADA Security Analysis

The security analysis framework focuses on formally modeling the EMS control routines, their interdependency, and how false data injection can alter the outcome of the control routines – in particular SE and TP – without being detected by the traditional bad data detection algorithms. The mathematical formulation of stealthy attacks against SE is introduced by Liu et al. in 2009 [8], which has received the attention of many researchers since then.

Methodology

The steady-state physical properties of the grid are governed by power flow equations, which express the conservation relations between generation and load at every bus or node in the system, at every instant. In this project, EMS modules are formally modeled, particularly to determine the means (attack vectors) of stealthy attacks. In the following, a simplified model of the security analyzer, which is based on the linear power flow equations (the DC model), is presented. The overall formal model capturing the physical system, the cyber-physical attack properties, and adversary attributes are summarized in Table 1. A brief explanation of the salient steps leading to the formalization is provided below.

Table 1 DC power flow equations

#1: Physical power flow properties:	
Power flows and topology:	
$\forall_{1\le i\le l} \quad P_i^L = d_i\left(\theta_{f_i} - \theta_{e_i}\right)$	(1)
$\forall_{1\le i\le l} \quad k_i \rightarrow \left(P_i^L = d_i\left(\theta_{f_i} - \theta_{e_i}\right)\right)$	(2)
Power consumptions:	
$\forall_{1\le j\le b} \quad P_j^B = \sum_{i\in \mathbb{L}_{j,in}} P_i^L - \sum_{i\in \mathbb{L}_{j,out}} P_i^L$	(3)
Power generation, loads, and consumption relationship:	
$\forall_{1\le j\le b} \quad P_j^B = P_j^D - P_j^G$	(4)
$\sum_{1\le j\le b} P_j^G = \sum_{1\le j\le b} P_j^D$	(5)
#2: False data injection attack properties:	
Attack definitions:	
$\forall_{1\le j\le n} \quad c_j \rightarrow (\Delta\theta_j \neq 0)$	(6)
$\forall_{1\le i\le l} \quad k_i \rightarrow (u_i \wedge \neg p_i) \vee (\neg u_i \wedge q_i)$	(7)
$\forall_{1\le i\le l} \quad (p_i \rightarrow u_i \wedge \neg v_i \wedge \neg w_i) \wedge (q_i \rightarrow \neg u_i \wedge \neg w_i)$	(8)
Attack evasion (UFDI) properties:	
$\forall_{1\le i\le l} \quad \neg k_i \rightarrow \left(\Delta P_i^L = 0\right)$	(9)
$\forall_{1\le i\le l} \quad \neg(p_i \vee q_i) \rightarrow \left(\Delta \bar{P}_i^L = 0\right)$	(10)
$\forall_{1\le i\le l} k_i \rightarrow \left(\Delta P_i^L = d_i\left(\Delta\theta_{f_i} - \Delta\theta_{e_i}\right)\right)$	(11)
$\forall_{1\le i\le l} \quad \left(p_i \rightarrow \left(\Delta \bar{P}_i^L = -P_i^L\right)\right) \wedge \left(q_i \rightarrow \left(\Delta \bar{P}_i^L = P_i^L\right)\right)$	(12)
Attack plan properties:	
$\forall_{1\le i\le l} \quad \Delta P_{i,\text{total}}^L = \Delta P_i^L + \Delta \bar{P}_i^L$	(13)
$\forall_{1\le j\le b} \quad \Delta P_{j,\text{total}}^B = \sum_{i\in \mathbb{L}_{j,in}} \Delta P_{i,\text{total}}^L - \sum_{i\in \mathbb{L}_{j,\text{out}}} \Delta P_{i,\text{total}}^L$	(14)
$\forall_{1\le i\le l} \quad \left(\Delta P_{i,\text{total}}^L \neq 0\right) \rightarrow (t_i \rightarrow a_i) \wedge (t_{l+i} \rightarrow a_{l+i})$ $\forall_{1\le j\le b} \quad \left(\Delta P_{j,\text{total}}^B \neq 0\right) \rightarrow \left(t_{2l+j} \rightarrow a_{2l+j}\right)$	(15)
#3: Adversary attribute properties:	
Attacker's knowledge:	
$\forall_{1\le i\le l} \quad \left(\Delta P_{i,\text{total}}^L \neq 0\right) \rightarrow ((t_i \vee t_{l+i}) \rightarrow g_i)$	(16)
Attacker's access capability:	
$\forall_{1\le i\le m} \quad a_i \rightarrow r_i \wedge \neg s_i$	(17)
Attacker's resource:	
$\sum_{1\le i\le m} a_i \le T_M$	(18)

Physical Model

Power Flow Model The DC power flow model makes several simplifying assumptions [1] that yield a linear relation between a system of equations of the form: [**B**] [θ] = [**P**]. Here, P denotes a vector of new power injections at a bus (node), while θ denotes the phase angles of unit magnitude bus voltages. The later variables are treated as *states*. Equations 1, 2, 3, 4, and 5 represent the modeling of the power flow

properties, where b and l are the number of buses and lines, respectively. Denoting the admittance (reciprocal of the impedance) of line i between buses f_i and e_i by d_i the real power flow $\left(P_i^L\right)$ across the line is represented by Eq. 1. The topology status (processed by TP based on the information of the circuit breakers and switches) is considered using k_i to denote if the line is open or closed (Eq. 2). The power consumption at bus j $\left(P_j^B\right)$ is the summation of the power injections at the bus (Eq. 3, where $\mathbb{L}_{j,\text{in}}$ and $\mathbb{L}_{j,\text{out}}$ are the sets of incoming lines and outgoing lines of bus j, respectively).

This consumption is also the difference of the load $\left(P_j^D\right)$ and the generation $\left(P_j^G\right)$ at the bus (Eq. 4). As the DC flow model assumes a lossless system, the power balance constraint (total generation = total load) is given by Eq. 5.

State Estimation The state estimation is the process of estimating n unknown variables (states) from m ($m > n$) known measurements $\mathbf{z}$ assuming a system of the form [2, 9]:

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

With the DC power flow model, this reduces to a linear model of the form: $\mathbf{z} = \mathbf{H}(\mathbf{x}) + \mathbf{e}$ where $\mathbf{H}$ is a $(m \times n)$ constant matrix. In simple words, this estimation is the process of solving line power flow and consumption equations (i.e., Eqs. 1 and 3) based on the received measurements for P_i^Ls and P_j^Bs for θ_js, based on the estimated topology (Eq. 2).

Cyber-Physical Attack Model

Idea of Stealthy Attacks Measurements can be corrupted due to device errors or communication noise, while there are bad data detection (BDD) algorithms to filter them. The widely used weighted least squares (WLS)-based BDD algorithm identifies a measurement as bad if the difference between a received measurement and its corresponding estimation (calculated from the estimated states) is greater than a threshold value. However, Liu et al. have shown that it is possible to compromise the state estimation by injecting false data to measurements while evading the BDD algorithm [8]. This type of stealthy attack is based on the idea of altering measurements following the physical properties (i.e., Eqs. 1 and 3). Table 1 presents the constraints that define attacks on measurements and topology (Eqs. 6, 7, and 8), the evasion or stealthy properties (Eqs. 9, 10, 11, and 12), and the attack plan (Eqs. 13, 14, and 15). This attack modeling is based on the difference between the original (attack free) and the corrupted (under attack) measurement values, which makes it possible to verify the attacks without knowing the actual measurements. Since state estimation is done based on the estimated topology, an adversary can poison the

topology, along with measurements, by injecting a false status, leading to exclusion/inclusion of one or more open/closed lines from/to the topology.

Attack Modeling An attack on state j (c_j) specifies that θ at bus j is changed ($\Delta\theta_j$) (Eq. 6). In an "exclusion" attack (p_i), a line actually in service (u_i) is omitted, if the line is not a fixed one (v_i) and the status information corresponding to the line is not secured (w_i) (Eq. 8). An "inclusion" attack includes a line that is actually not in service, and its associated status information is not secured (Eq. 8). A line is considered (k_i) in the EMS routines if (i) the line is in service and no exclusion attack is launched against this line or (ii) an inclusion attack is performed on an open line (Eq. 7).

Attack Evasion Properties In order to evade the BDD algorithm, the injection of false data in the measurements $\left(\Delta P_i^L\right)$ should follow the change in states along with the rest of the measurements and states (Eq. 11). In the topology poisoning attack alone, the BDD algorithm is evaded by keeping the states unchanged, while necessary measurements are changed accordingly $\left(\Delta \bar{P}_i^L\right)$. If a closed line is excluded from the topology, the corresponding line power flow measurement must be zero and the corresponding connected buses' power consumption measurements are adjusted accordingly (Eq. 12). On the other hand, when an 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 (Eq. 12).

Attack Vectors The attack vector (plan) is a set of measurements that need to be altered to coordinate the attack actions for stealthiness. The total change to a measurement is denoted by $\Delta P_{i,\text{total}}^L$ and $\Delta P_{j,\text{total}}^B$ (Eqs. 13 and 14). When $\Delta P_{i,\text{total}}^L \neq 0$, taken measurements corresponding to line i (i.e., t_i and t_{l+i}) are required to be altered (a_i and $a_{\text{l+i}}$) (Eq. 15) according to $\Delta P_{i,\text{total}}^L$. Similarly, when $\Delta P_{j,\text{total}}^B \neq 0$, the power consumption measurement at bus j needs to be changed (Eq. 15). Conversely, a measurement is altered only if it is required. A violation of these constraints will make the attack detectable.

Modeling Adversary Attributes

An adversary cannot have an unlimited capability, or a system cannot practically be secured from all possible attacks. Therefore, it is prudent to analyze an adversary with limited but practical capabilities (e.g., expressed in terms of knowledge, access, and resources). These capabilities can be expressed as formal constraints. With regard to knowledge, since the electrical characteristics of the grid and other system properties are usually well-guarded, they are not easily accessible to adversaries. If the admittance of a line is unknown (g_i), then an adversary cannot determine appropriate changes to power flow measurements (Eq. 16). The attacker usually does not have necessary physical or remote access (r_i) to inject false data into all of the measurements. If a measurement is secured (i.e., integrity-protected) (s_i), then even if the attacker has the ability to inject false data into the measurement, the false

data will be detected (Eq. 17). Due to resource constraints, an adversary can corrupt a limited number of measurements/buses at a time (Eq. 18).

Interdependency Models

An example is presented here that formally models interdependency toward assessing the impact on an EMS module, namely, OPF, through attacking SE. OPF is responsible for determining individual generator outputs that minimize the overall cost of generation while meeting physical properties (i.e., transmission, generation, and system-level operating constraints) [2, 9]. Since an attack on state estimation can result in a redistribution of loads, an OPF solution may be no longer optimal, leading to an economically disadvantageous solution. The OPF considers the entire set of power flow equations as one big constraint, which includes the constraints regarding load-generation balance, generation limits, and transmission line capacities, along with cost minimization. Thus, OPF can be expressed as a conjunction of a set of individual constraints, while it can be merged with the attack models, with respect to the load changes.

Change in Loads Due to Stealthy Attacks If $\Delta P_j^B \neq 0$ (Eq. 4), this specifies that there is a load and/or generation power change at the bus. Since the generation (metered at the plant) is typically altered at the request of the system operator, it can be assumed that the power consumption change specifies a change exclusively in the load:

$$\forall_{1 \leq j \leq b} \ \Delta P_{j,\text{total}}^D = \Delta P_{j,\text{total}}^B$$

If a load change is observed, the OPF process must be rerun to find the optimal generation dispatches.

Impact on OPF The minimization function of OPF can be abstracted as a cost constraint:

$$\sum_{1 \leq j \leq b} c_j\left(\hat{P}_j^G\right) \leq T_{\text{OPF}}$$

Here, C_j is the cost function for the generation at bus j and T_{OPF} is the OPF cost threshold. Assuming that the cost (T_{OPF}) is increased with respect to the original (no-attack scenario) OPF cost T_{OPF} by $I\%$, the minimum impact on OPF can be expressed as:

$$\left(T_{\text{OPF}} = T_{\text{OPF}} \ I/100\right) \rightarrow \neg\left(\exists_{\hat{P}_1^G, \hat{P}_2^G, \ldots, \hat{P}_b^G} \text{OPF}\right)$$

This constraint specifies that no OPF solution is possible that does not increase the generation cost by $I\%$.

Example Case Study

A case study of security threat analysis is briefly presented here [4, 5]. The system configuration and the constraints corresponding to the prior model are encoded into SMT [7] using the *Z3* [10] solver. Boolean variables are used for logical constraints, while real variables are used for values of measurements and states. By executing the model, if the result is unsatisfied (*unsat*), then no attack vector can satisfy the constraints. However, if result is satisfied (*sat*), the attack vector is received from the assignments of the corresponding variables, which represent the measurements required to be altered to achieve the attack.

Attack Verification [4] This case study is to demonstrate stealthy attack verification on the IEEE 14-bus test system [11], as shown in Fig. 4. In this example, the admittances of lines 3, 7, and 17 are unknown. All of the 20 lines (as shown in Fig. 4) are included in the true topology, though lines 5 and 13 are not part of the core topology (i.e., these lines can be kept open if necessary). Since this system has 14 buses and 20 lines, the maximum number of potential measurements is $(14 + 2 \times 20)$ or 54. Here, all the potential measurements are used except measurements 5, 10, 14, 19, 22, 27, 30, 35, 43, and 52, and among these measurements, 1, 2, 6, 15, 25, 32, and 41 are crypto-secured for integrity protection. In this example, let's assume that the attacker has access to all measurements and the target is to attack state 12 only (i.e., no other states will be affected). However, due to the resource

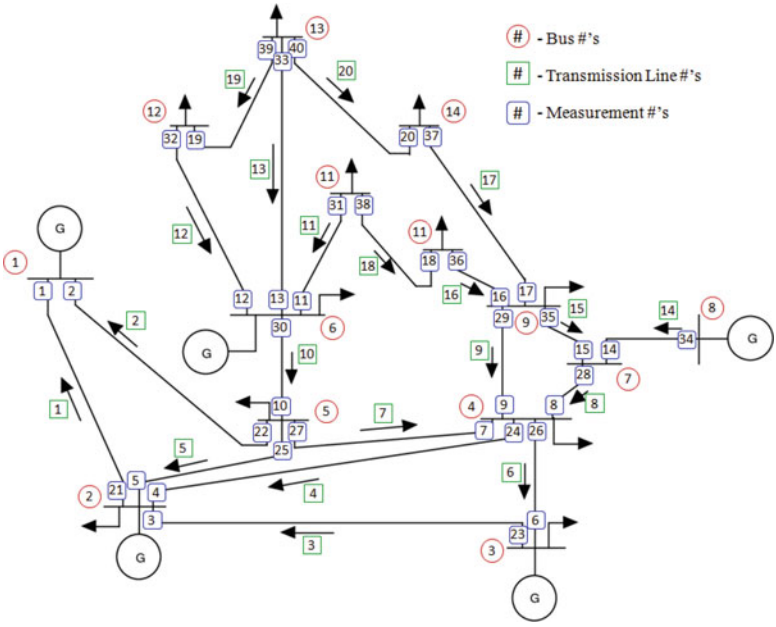

Fig. 4 IEEE 14-bus test system with measurement numbers

limitation, the attacker can alter up to 10 measurements distributed in three or less substations (i.e., buses). Is there a feasible attack vector in this case? The preliminarily implemented model (security analyzer) shows that attack for state 12 is feasible under this model when, for example, measurements 12, 32, 39, 46, and 53 are compromised (alerted). However, if only measurement 46 is considered as crypto-secured, then no feasible attack exists unless the attacker can alter (poison) the topology information as well. Specifically, the model can also tell if line 13 needs to be deceivably excluded from the topology by corrupting the topology information. This attack is feasible by compromising measurements 12, 13, 32, 33, 39, and 53. This is an example of novel attacks showing the capability of the proposed framework that allows the discovery of coordinated attacks (e.g., in this case, by integrating the topology poisoning with false data injection to measurements).

Impact-Based Attack Verification [5] The attack vector can be conditioned with the target impact. For example, if the adversary's objective is to launch a stealthy (undetected) attack to induce at least a 5% increase to the OPF base solution on the same 14-bus test system and the adversary can attack 25 measurements distributed on 10 buses, then the model identifies a feasible attack vector at which (1) states 10, 11, 12, and 14 are compromised through injecting false data to measurements 4, 11, 12, 16, 17, 18, 20, 24, 31, 32, 36, 37, 38, 39, 40, 42, 44, 46, 49, 50, 51, 53, and 54 and (2) the topology information is corrupted to falsely remove lines 4 and 17. The model can provide valuable insight on the required attacker profile and capabilities to accomplish an attack. For example, the model can confirm that, under any circumstance, an attacker with the same capability cannot cause 7% or more upward drift in OPF outcome even with coordinated false data injection (stealthy) attacks on the measurements and the topology.

Formal Model for SCADA Resiliency Analysis

The resiliency requirements that are considered in the resiliency threat analyzer [6] ensure whether or not a SCADA control process receives sufficient data (i.e., measurements from field devices) to perform its operation even in (limited) contingencies. The threat analyzer solves a formal model that formalizes necessary control function data requirements and security properties, along with the physical topology and devices.

Methodology

The observability analysis is a prior and crucial requirement for performing the power system state estimation control routine [1, 2]. The threat analysis is performed considering this observability analysis. Three resiliency specifications are modeled

in this analyzer: (i) k —resilient observability; (ii) k—resilient secured observability; and (iii) k, r—resilient bad data detectability. A brief description of the formal model for k—resilient secured observability analysis is presented below.

SCADA Cyber-Physical System Modeling

The SCADA system modeling primarily includes the configurations associated with various SCADA devices and the topology, as follows:

SCADA Device Configuration Modeling A SCADA system consists of different physical device components, among which IEDs, PLCs, RTUs, and MTUs are important. The SCADA physical devices are modeled mainly based on their communication and security configurations, especially those properties that are essential to model the security specifications and resiliency requirements. Each SCADA device (e.g., an IED) is identified by an ID (e.g., i). Whether a device with an ID is an IED is determined using a parameter (e.g., Ied_i). Similar parameters are there for other SCADA devices. A parameter (e.g., Ied_i) can define whether device i is an IED. A device profile is represented as a conjunction of different parameters. The data reporting often follows a schedule and it is modeled based on the reporting mode. The reporting mode for the field devices can be pull or push, although a SCADA device typically delivers data to the control server upon receiving a request from the server. To achieve end-to-end security, the communicating devices must be configured with necessary cryptographic (authentication and encryption) properties. However, a device can support none, one, or multiple crypto properties. The crypto property of a device (e.g., $Crypt_i$ for device i) can be modeled as a conjunction of one or more crypto profiles ($CryptType_{i,k}$ for one or more ks). For example, each crypto profile (K, e.g., $CryptType_{i,k} = K$) specifies an algorithm ($CAlgo_K$) and a key length ($CKey_K$). Similarly, the communication properties (e.g., supported communication protocols) for each device are modeled.

SCADA Topology Modeling This task models the SCADA communication network topology (i.e., the connectivity between the SCADA physical devices). Typically, multiple IEDs or PLCs are connected with an RTU, while all or some RTUs are connected to an MTU directly or through some intermediate RTUs using WAN. However, different topology patterns are possible. There can be more than a single MTU, in which case one of them works as the main MTU (corresponding to the main control center), while the rest of the MTUs are hierarchically connected to the main one. The measurements and control commands route through this communication topology between the devices. Although the communication among field devices in SCADA often can be point-to-point (e.g., an IED to an RTU or an RTU to an RTU), this modeling will consider intermediate network devices like routers and firewalls when they exist. A link in the topology is identified by an ID (e.g., l), while a parameter (e.g., $NodePair_l$) represents the nodes connected by the link, and a parameter (e.g., $LinkStatus_l$) specifies if the link is up or down. There can be other properties, including the medium type (i.e., wireless, Ethernet, modem, etc.) and the

link bandwidth. It is worth mentioning that a communication path (e.g., a routing path through routers and links from an RTU to another RTU) can be abstracted as a link as long as the internal routing path is not considered for a resiliency specification.

Modeling of Attacks and Security Controls

The attack model is designed with respect to SCADA networks. The design of security controls primarily deals with end-to-end data communication.

Attack Model Design Cyber-attacks corresponding to data unavailability are modeled here. A particular data can be unavailable due to some technical failures at the source node, intermediate forwarding nodes, or communication links, as well as due to distributed denial of service (DDoS) attacks. A Boolean parameter ($Node_i$) is used to denote whether device i is available or not.

Security Control Modeling A requirement analysis is performed on standard security recommendations for SCADA (e.g., NIST and NERC security guidelines [12, 13]) to identify logical structures associated with SCADA configurations and security properties. The major control modeled here is secured data delivery.

The secured data delivery checks for assured data delivery, as well as whether the data is sent under proper security measures, particularly authentication and integrity protection. The data delivery is ensured with the satisfaction of various constraints, which primarily include these three: (i) reachability, (ii) communication protocol pairing, and (iii) crypto property pairing. The communicating nodes (e.g., an RTU and the MTU) may have correct security pairing, as they are using the same security protocol (e.g., challenge-handshake authentication protocol (CHAP)). However, this security pairing on CHAP can only ensure authentication. In this case, the transmission will not be data integrity-protected. Moreover, it is needed to consider the vulnerabilities of the security measures in use. For example, if data encryption standard (DES) is used for data encryption, the transmitted data cannot be considered as protected, as a good number of vulnerabilities of DES have already been found. Hence, the formalization of the secured data delivery (*SecuredDelivery*) includes two constraints – *Authenticated* and *IntegrityProtected* – that ensure the authentication of the communicating parties and the integrity of the transmitted data, respectively:

$$\exists_K \left(\exists_K CryptType_{i,k} = K\right) \wedge \left(\exists'_k CryptType_{i,k'} = K\right) \wedge$$
$$\left((CAlgo_K = hmac \wedge CKey_K \geq 128) \vee \ldots\right) \rightarrow Authenticated_{i,j}$$
$$\exists_K \left(\exists_K CryptType_{i,k} = K\right) \wedge \left(\exists'_k CryptType_{i,k'} = K\right) \wedge$$

$$((CAlgo_K = sha2 \wedge CKey_K \geq 128) \vee \ldots)) \rightarrow IntegrityProtected_{i,j}$$

$$Ied_I \wedge \exists_z \forall_{l \in |P_{I,j,z}} \{i', j'\} \in NodePair_l \wedge Node_{i'} \wedge Node_{j'} \wedge$$

$$Reachable_{i',j'} \wedge CommPropPairing_{i',j'}, \wedge CryptoPropPairing_{i',j'} \wedge$$

$$Authenticated_{i',j'}, \wedge IntegrityProtected_{i',j'}$$

$$\rightarrow \quad SecuredDelivery_I$$

Modeling of Resiliency Threats Based on SCADA Operations

An essential resiliency requirement in general can ensure whether or not a SCADA control process receives sufficient (secured) data (i.e., measurements from field devices) to perform its operation, even if some contingency occurs within some limit (e.g., a threshold number of field devices are under failure due to data unavailability attacks). The resiliency threat verification is designed to answer the same query, in other words, by looking for a set of field devices (e.g., IEDs and RTUs) such that the set size is no more than a threshold value and the unavailability of these devices will make the SCADA control process fail because of insufficient data. The modeling of *k*—resilient secured observability property is performed as follows.

The property is verified by searching for threat vectors under the specification of maximum failures of *k* field devices. When the number of unavailable devices is no larger than *k* devices (IEDs and/or RTUs), the threat against the *k*—resilient secured observability requirement ($\neg ResilientSecuredObservability$) is formalized as follows:

$$\left(\left(N - \sum_{1 \leq i \leq N} Node_i\right) \leq k\right) \wedge \neg SecuredObservability$$

$$\rightarrow \quad \neg ResilientSecuredObservability$$

The resiliency requirement can specify the device type clearly (e.g., k_1 IEDs and k_2 RTUs), instead of *k* devices of any (or multiple) types. The threat against the (k_1, k_2)—resilient secured observability requirement is formalized as follows:

$$\left(\left(N_1 - \sum_{1 \leq i \leq N_1} (Node_i \times Ied_i)\right) \leq k_1\right) \wedge$$

$$\left(\left(N_2 - \sum_{1 \le i \le N_2} (Node_i \times Rtu_i)\right) \le k_1\right) \wedge \neg SecuredObservability$$
$$\rightarrow \quad \neg ResilientSecuredObservability$$

The threat vector ($\mathcal{V}$) represents those devices for which the following equation is true: $\forall_{i \in \mathcal{V}} \neg Node_i$.

Example Case Study

The resiliency threat analyzer's execution is illustrated with an example. This example considers a 5-bus SCADA system as shown in Fig. 5 and demonstrates the k_1, k_2—resilient secured observability analysis. This is a subsystem taken from the IEEE 14-bus test system [11]. The input includes primarily the Jacobian matrix corresponding to the bus system [2], the connectivity between the communicating devices, the association of the measurements with the IEDs, and the security profiles of each communicating host pair. It is assumed that the measurements are recorded by different IEDs only, and these measurements are sent to the MTU (i.e., the SCADA server at the control center) through the RTUs. The server needs these measurements to estimate the current states of the system. The resiliency specification is 1, 1—resilient secured observability. The formal model corresponding to this

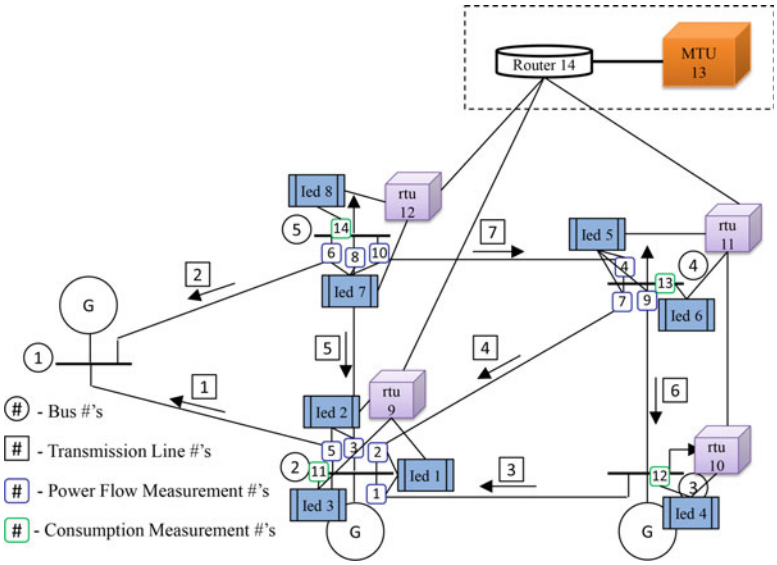

Fig. 5 An example SCADA topology of a 5-bus grid

example is solved using Z3 [10]. The solution to the model gives a result as *sat* (satisfiable) or *unsat* (unsatisfiable). In a *sat* case, the solver provides an elaborated result, specifically the values of the terms, from which the required output is assembled. In the case of (1, 1)—resilient secured observability verification, the model provides a *sat* result. That is, the system is not (1, 1)—resilient in terms of secured observability. According to the result, if IED 3 and RTU 11 are unavailable, it is not possible to observe the system securely. The result also justifies the answer, showing that measurements from IED 1 and RTU 9 are not data integrity-protected, and thus, when IED 3 and RTU 11 are unavailable, some states cannot be observed securely. By continuing to look for the next threat vector, another four threat vectors can be observed. However, if the resiliency specification is reduced to (1, 0) or (0, 1), the model returns *unsat* (i.e., the system is securely observable even if an IED or RTU fails). Now, if the SCADA topology of Fig. 5 is modified by connecting RTU 9 with RTU 10, while removing the direct path between RTU 9 and the MTU, the system is not resilient any more for one RTU failure. There is only one threat vector (unavailability of RTU 12) that fails the secured observability.

Conclusion

Unlike the existing approaches that focus on discovering specific attack vectors, the presented formal approach offers a comprehensive analysis for verifying SCADA security and resiliency properties systematically, provably, and efficiently. The corresponding frameworks take the bus data, SCADA device configurations, operational constraints, security properties, and resiliency requirements as inputs; formally model secure interactions among the devices and potential contingencies; and solve the model to verify the resiliency of the system. The key features of these frameworks are as follows: (i) a formal framework that utilizes advanced formal logics; (ii) provable verification of security and resiliency threats; (iii) a generic framework design capable of being applied in any SCADA architecture; and (iv) an extensible model to accommodate new security and resiliency properties. These formal frameworks can allow the grid operators to provably query and inspect the system's security and resiliency without relying on invasive, laborious, and expensive real-life or testbed-based experiments.

References

1. A.J. Wood, B.F. Wollenberg, *Power Generation, Operation, and Control*, 2nd edn. (Wiley, New York, 1996)
2. A. Abur, A.G. Exposito, *Power System State Estimation: Theory and Implementation* (CRC Press, New York, 2004)

3. Nistir 7628: Guidelines for smart grid cyber security. (Smart Grid Interoperability Panel- Cyber Security Working Group, Aug 2010), http://www.nist.gov/smartgrid/upload/nistir-7628_total.pdf
4. M.A. Rahman, E. Al-Shaer, R. Kavasseri. Security threat analytics and countermeasure synthesis for state estimation in smart power grids. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, June 2014
5. M.A. Rahman, E. Al-Shaer, R. Kavasseri. Impact analysis of topology poisoning attacks on economic operation of the smart power grid. In *International Conference on Distributed Computing Systems (ICDCS)*, July 2014
6. M.A. Rahman, A.H.M. Jakaria, E. Al-Shaer. Formal analysis for dependable supervisory control and data acquisition in smart grids. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, June 2016
7. L. de Moura, N. Bjørner. Satisfiability modulo theories: An appetizer. In *Brazilian Symposium on Formal Methods*, 2009
8. Y. Liu, P. Ning, M. Reiter. False data injection attacks against state estimation in electric power grids. In *ACM Conference on Computer and Communications Security (CCS)*, pp. 21–32, Nov 2009
9. A. Monticelli, *State Estimation in Electric Power Systems: A Generalized Approach* (Kluwer Academic Publishers, Norwell, 1999)
10. Z3: Theorem prover. (Microsoft Research, 2013), http://research.microsoft.com/en-us/um/redmond/projects/z3/
11. Power systems test case archive. http://www.ee.washington.edu/research/pstca/
12. National Institute of Standards and Technology. U.S. Department of Commerce. http://www.nist.gov/, http://www.nist.gov/publication-portal
13. North American Electric Reliability Corporation. http://www.nerc.com, http://www.nerc.com/pa/Stand/Pages/default.aspx