Formal Analysis for Dependable Supervisory Control and Data Acquisition in Smart Grids

Mohammad Ashiqur Rahman*, A H M Jakaria*, and Ehab Al-Shaer†
*Department of Computer Science, Tennessee Tech University, USA
†Department of Software and Information Systems, University of North Carolina at Charlotte, USA
Emails: marahman@tntech.edu, ajakaria42@students.tntech.edu, ealshaer@uncc.com

***Abstract*—Smart grids provide innovative and efficient energy management services that offer operational reliability. The Supervisory Control and Data Acquisition (SCADA) system is a core component of a smart grid. Unlike the traditional cyber networks, these components consist of heterogeneous devices, such as intelligent electronic devices, programmable logic controllers, remote terminal units, control servers, routing and security devices, etc. SCADA devices 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 critical damages including power outages and destruction of equipment. In this work, we develop an automated security and resiliency analysis framework for SCADA in smart grids. This framework takes smart grid configurations and organizational security and resiliency requirements as inputs, formally models configurations and various security constraints, and verifies the dependability of the system under potential contingencies. We demonstrate the execution of this framework on an example problem. We also evaluate the scalability of the framework on synthetic SCADA systems.**

***Index Terms*—Smart grids; SCADA; security; resiliency; formal verification.**

I. Introduction

In the energy transmission and distribution side of the smart grid, different communication networks exist for sensing measurements and transmitting control commands. These networks are associated with the SCADA system. SCADA is the major Industrial Control System (ICS) in smart grids, and connects the generating stations, substations, and control centers. SCADA is mainly responsible for monitoring and controlling the remote equipment by obtaining data from the remote devices, analyzing the received data at the control centers, and executing necessary control commands at the remote devices.

The control centers associated with the generation, transmission, and distribution systems are connected to the physical power system using cyber infrastructure. In order to promote connectivity and remote access capabilities among corporate business systems, information technology (IT) is now increasingly used in SCADA, which escalates the possibility of cyber security vulnerabilities and incidents, as ICS was not built taking security into consideration. 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 and different industrial communications protocols. Moreover, in order to operate the grid efficiently and reliably, various control routines are executing at the control centers which are actively dependent on the data acquisition from the field devices. Therefore, the vulnerabilities and threats as well as the security and resiliency requirements of SCADA are often different from that of the traditional IT systems. Therefore, it is important to develop an automated security and resiliency verification framework explicitly for SCADA in smart grids.

In this paper, we present a formal framework that automatically verifies the security and resiliency of the SCADA system, particularly the resilient data acquisition for reliable execution of control operations. This framework takes necessary SCADA configurations and resiliency requirements, formally models the analytics, and solves the model to verify the system with respect to the given resiliency specifications. This framework uses an SMT-based formal analysis engine [1] to solve the model. The framework provides threat vectors if the resiliency requirement fails. The unsatisfiable outcome certifies the specified resiliency of the system. Therefore, this framework allows a grid operator to understand the SCADA system's resiliency as well as to fix the system by analyzing the threat vectors.

The rest of the paper is organized as follows. We discuss necessary background for this work in the next section. We present formal models for the SCADA security analysis, which includes the formal modeling of SCADA configurations and that of SCADA resiliency constraints in Section III. We illustrate a case study with respect to a 5-bus system in Section IV. We briefly discuss the related work in Section VI and conclude the paper in Section VII.

II. State of the Art, Challenges, and Objectives

A. Supervisory Control and Data Acquisition Systems

Industrial Control Systems (ICS) are often found in industries, such as electric, water, oil, natural gas, chemical, transportation, etc. Supervisory Control and Data Acquisition (SCADA) systems are examples of ICS systems, which are generally used in controlling dispersed assets using centralized data acquisition and supervisory control. An example topology of SCADA is shown in Fig. 1. Typical SCADA operations includes automatic and human control loops, remote diagnostics, and maintenance utilities. There are various

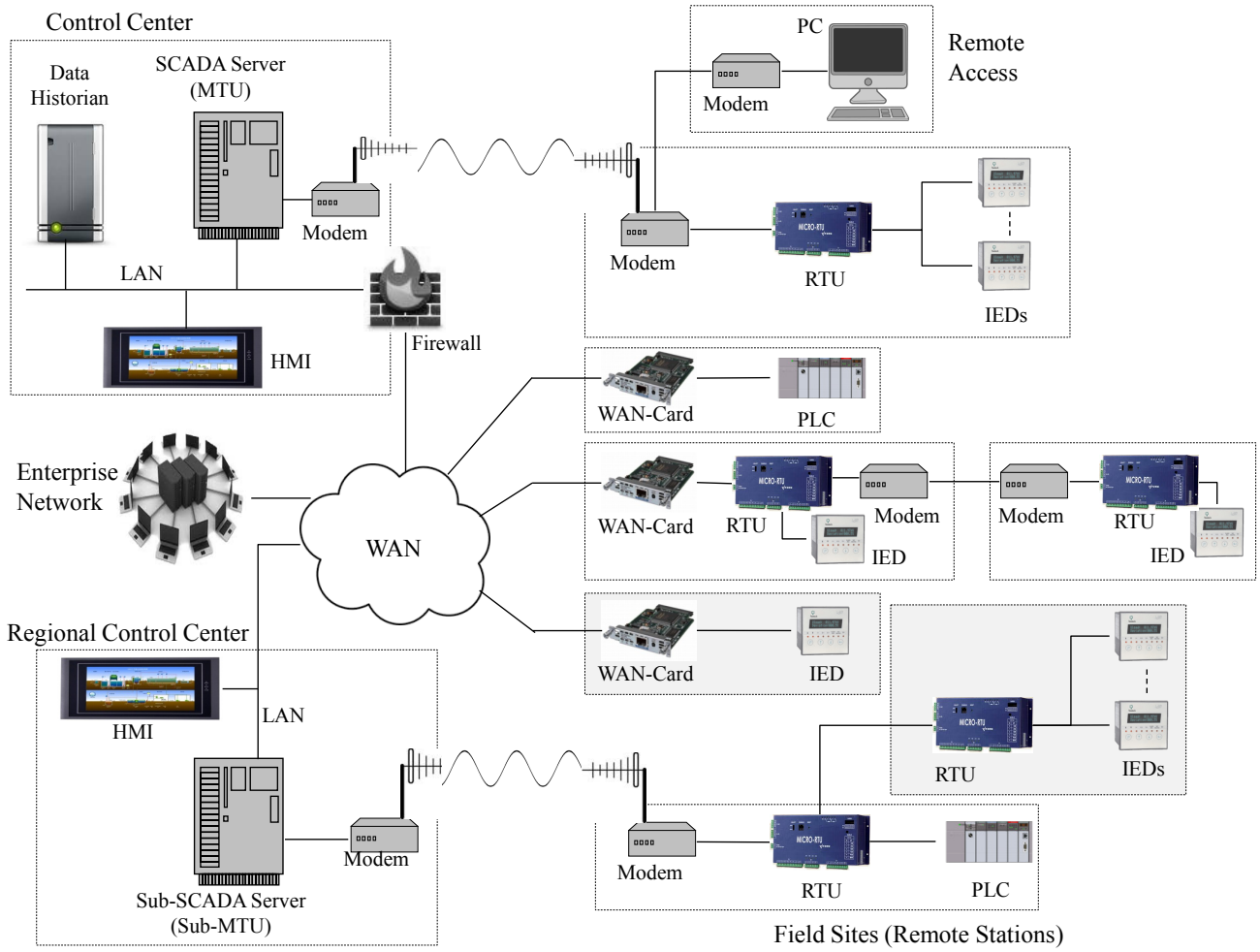

Fig. 1. An example of the SCADA network topology.

kinds of control components, such as SCADA control servers or Master Terminal Units (MTUs), Remote Terminal Units (RTUs), Programmable Logic Controllers (PLCs), Intelligent Electronic Devices (IED), Human Machine Interfaces (HMI), data historian, etc. 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 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. There are different control modules or routines to manage the grid efficiently and reliably [2], [3]. 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. Therefore, the dependable delivery of data is crucial.

B. 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 integrating different heterogeneous and legacy systems in SCADA significantly increases the potential of security threats, which can cause massive and devastating damage. There are two main causes of threats [4]. 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 SCADA 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

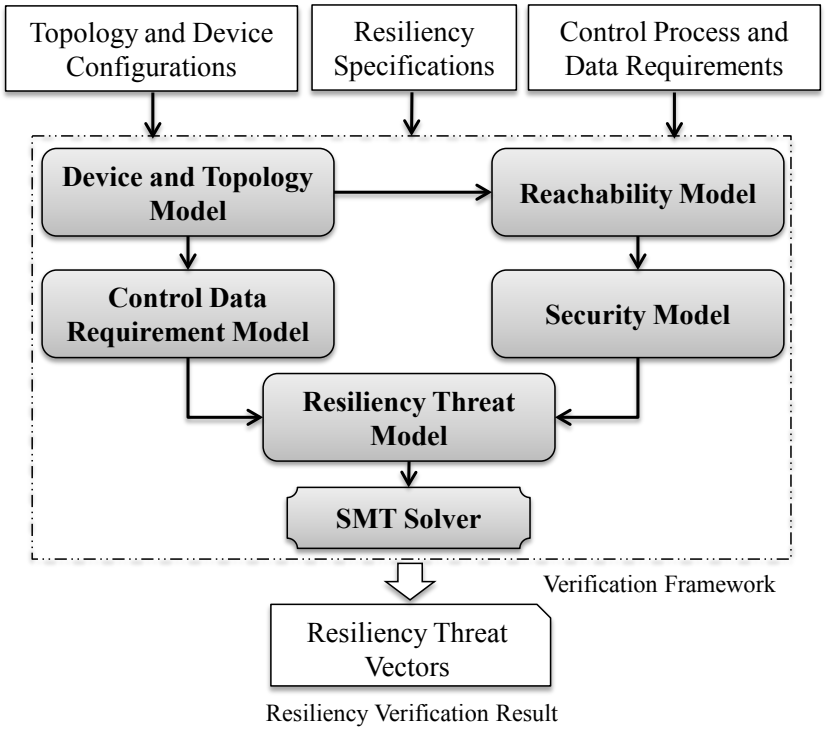

Fig. 2. The framework of the SCADA analyzer.

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 fails 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.

C. Objectives

The correct functioning of SCADA 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 is 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. Implementing these security and resiliency controls in a scalable and provable manner is one of the major challenges in smart grid security modeling.

The goal of this research is to develop a framework 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. This research aims at modeling secured communication, potential contingencies, and resiliency specifications, and at creating an efficient solution to analyze the resiliency of the system by identifying the threat vectors that negate the security and resiliency requirements. The research approach targets scalable and extensible design of the resiliency verification as a constraint satisfaction problem. In this work, we particularly focus on modeling the trusted and secure data communication from the field devices to the control server such that SCADA control routines can operate with valid data even in contingencies. Although this paper presents formalizations for a limited set of constraints that are important for proper communication, an important feature of the proposed formal framework is its easy extensibility. For further properties, one just needs to add necessary constraints.

III. Formal Model for SCADA Resiliency Verification

In this section, we present the formal model corresponding to SCADA security and resiliency requirements.

A. Preliminary

Fig. 2 presents the SCADA resiliency verification framework, SCADA Analyzer. It takes necessary inputs, particularly with regards to the physical components (*i.e.*, the communication and security properties of SCADA devices), the topology (*i.e.*, connectivity between the devices), the SCADA control

TABLE I
VARIOUS NOTATIONS USED IN FORMALIZATIONS

Notation	Definition
Ied_i	If device i is an IED.
Rtu_i	If device i is an RTU.
$Node_i$	Whether device i is available or not.
$NodePair_l$	The pair of nodes connected by link l.
$h_{Z,X}$	The element in the Jacobian matrix at row Z and column X representing if state X has an impact on measurement Z.
$P_{I,z}$	The z'th path from IED I to the MTU.
$StateSet_Z$	The set of states that constitute measurement Z.
$UMsrSet_E$	The set of measurements that represent the same electrical component (E).
$MsrSet_I$	The set of measurements transmitted by IED I.
D_Z	Whether measurement Z is successfully delivered.
$DelUMsr_E$	Whether one or more measurements.

operation (*i.e.*, the data requirements for the process), and the resiliency specifications. The analyzer formally models SCADA configurations, reachability among communicating parties, operational requirements, secured delivery, and the violation of resiliency specifications as constraints, and encodes these constraints into SMT logics. Then, these constraints are solved using an efficient SMT solver [5]. The solution result provides resiliency threat vectors (*i.e.*, how the resiliency requirements can be violated). If no threat is found, then it specifies that the system satisfies the given resiliency specifications. The formal modeling for the SCADA security and resiliency verification is discussed in the following in three parts. First, we model SCADA configurations. Then, we model the verification of operational resiliency of a control process. Lastly, we present the modeling of the resiliency of secured execution of the control operation.

The security and resiliency requirements that we consider in this work 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. We choose the observability analysis, which is a prior and crucial requirement for performing the power system state estimation control routine [2], [3]. Moreover, we consider the secured (particularly, authenticated and integrity protected) communication of the data that can provide correct results while adversaries can inject false data. Thus, we address mainly three resiliency specifications: (i) $k-$resilient observability, (ii) $k-$resilient secured observability, and (iii) $k, r-$resilient bad data detectability.

B. Formal Modeling of SCADA Configurations

A SCADA network consists of different types of devices, heterogeneous communication links, and various security policies. However, we present the formalizations of some selective configurations that are crucial to model security constraints. Table I includes a list of notations used in these formalizations.

Modeling of SCADA Device Configurations:

SCADA consists of different physical device components, among which IEDs, PLCs, RTUs, and MTUs are important. Usually IEDs, PLCS, and RTUs are associated with substations, while an MTU is associated with a control center. IEDs, PLCs, and RTUs are referred to as field devices. We model the SCADA physical devices, particularly IEDs, RTUs, and an MTU, based on their communication and security configurations. These properties are essential to model the reachability and secured communication.

Each IED, or RTU is identified by an ID. Ied_i and Rtu_i define if device i is an IED or an RTU, respectively. A device profile is represented as a conjunction of different parameters. A device can be unavailable if it is suffered with some technical failures, cyber attacks (DoS), or a link failure toward the device. $Node_i$ is a Boolean variable that denotes whether device i is available or not.

To achieve end-to-end security, the communicating devices must agree in their cryptographic (authentication and encryption) properties. A device can support none, one, or multiple cryptographic properties. We model the cryptographic properties of a device (*e.g.*, i) using $Crypt_i$ as a conjunction of one or more cryptosecurity profiles ($CryptType_{i,k}$s). Each crypto profile (K, *e.g.*, $CryptType_{i,k} = K$) specifies an algorithm ($CAlgo_K$) and a key length ($CKey_K$).

Similarly, the communication protocols supported by a device are specified using $CommProto_i$. Typically, there are ICS-specific protocols (*e.g.*, modbus, DNP3, etc.) for communication. The IP addresses of an RTU is specified using $IpAddr_i$. It is worth mentioning that the ID is enough to identify a device for reachability modeling, as the communication among field devices in SCADA can be abstracted as point to point (*e.g.*, an IED to an RTU or an RTU to an RTU, without considering routers, if they exist in some cases).

Modeling of SCADA Topology:

Typically, multiple IEDs are connected with an RTU, while all or some RTUs are connected to an MTU directly or through some intermediate RTUs and/or WAN. 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 connected to the main one. The measurements and control commands flow through this communication topology between the devices. In this work, we consider single MTU-based SCADA systems (Fig. 1).

A link in the topology is identified by an ID (*e.g.*, l). $\mathcal{L}$ is the set of links in the topology. 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 analysis in the model. $NodePair_l$ represents the nodes connected by link l and $LinkStatus_l$ specifies if the link is up or down. There can be other properties, if necessary, such as the link type, including the medium type (*i.e.*, wireless, ethernet, modem, etc.), and the link bandwidth.

C. Formal Modeling of $k-$Resilient Observability

The modeling of this resiliency constraint inherently often needs to consider another constraint, namely assured data delivery ($AssuredDelivery$). $AssuredDelivery$ ensures that the data is transferred from the data measuring field device (*i.e.*, an IED) to the ultimate receiver (*i.e.*, the MTU) successfully.

Assured Data Delivery:

$AssuredDelivery$ is developed based on three properties: (i) reachability, (ii) communication protocol pairing, and (iii) crypto properties pairing. The reachability property states that there is a data routing or forwarding path from the sender to the receiver. We define the forwarding paths among the communicating devices based on the given topology that includes communication links among the devices. Without the loss of generality, we assume that two devices (IEDs, RTUs, or the MTU) are reachable if there is a communication path (can be represented as a link, for the simplicity, although there can be multiple links on the path) between them. $Reachable_{i,j}$ states that there is a communication path between device i and device j. In order to make successful communication, the communication protocol supported by these two communicating devices must be the same. If the sender, the receiver, or the both need authenticated or encrypted data transmission, the both parties must support the same crypto properties. $CommProtoPairing_{i,j}$ and $CryptoPropPairing_{i,j}$ does these matching functions, respectively.

We consider all possible forwarding paths from an IED (I) to the MTU, through RTUs. $\mathcal{P}_I$ represents the set of these paths and $\mathcal{P}_{I,z}$ ($\mathcal{P}_{I,z} \in \mathcal{P}_I$) is the z'th path from IED I to the MTU. $\mathcal{P}_{I,z}$ is a set of communication links $\{l_{I,z,1}, l_{I,z,2}, ...\} \subseteq \mathcal{L}$ that form data transmission or forwarding path from the IED to the MTU. Then, the following equation formalizes the conditions when the assured data delivery is ensured:

$$
\begin{aligned}
&Ied_I \wedge \\
&\exists_z \forall_{l \in |\mathcal{P}_{I,z}|} \ \ \{i,j\} \in NodePair_l \wedge \\
&\quad Node_i \wedge Node_j \wedge Reachable_{i,j} \wedge \\
&\quad CommPropPairing_{i,j} \wedge CryptoPropPairing_{i,j} \\
&\quad\quad \rightarrow \ AssuredDelivery_I
\end{aligned}
$$

Observability:

The power system is observable when the measurements can solve a list of unknown variables. Each of these variables stands for a state. Typically, each measurement represents a power equation. Therefore, we need to know each equation regarding a particular measurement, where the equation specifies the variables that produce this measurement. In state estimation, there is a Jacobian Matrix that represents these relationships between the measurements and the unknown variables [3]. The observability constraint ensures two conditions: (i) the received (*i.e.*, recorded by the IEDs and successfully delivered to the MTU) distinct or unique measurements can cover all the variables (*i.e.*, unknown states), and (ii) the number of these measurements is greater than or equal to the number of variables. These two conditions are minimal requirements to ensure that there is a single estimation of the system variables, and then the system is specified as observable. The uniqueness of a measurement needs to be considered, as there are often more than one measurement that actually represents the same electrical component. For example, the power flow through a line can be measured at both ends of the line [3]. Therefore, these two measurements (forward line power flow and backward line power flow) represent the same electrical component. In the following, we describe the formalizations of the observability constraint.

Each row of the Jacobian matrix has a set of entries (column values), where each entry is associated with a state/variable:

$$
\begin{bmatrix}
h_{1,1} & h_{1,2} & \cdots & h_{1,n} \\
h_{2,1} & h_{2,2} & \cdots & h_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
h_{m,1} & h_{m,2} & \cdots & h_{m,n}
\end{bmatrix}
$$

Here, $h_{Z,X}$ is an entry where row Z is associated with measurement Z ($1 \leq Z \leq m$) and column X is associated with state variable X ($1 \leq X \leq n$). The variables corresponding to the nonzero entries only have impact on the measurement. Let $StateSet_Z$ be the set of states that constitute measurement Z. Then, $StateSet_Z$ is formalized as follows:

$$\forall_X \forall_Z \ (h_{Z,X} \neq 0) \rightarrow \ X \in StateSet_Z$$

When two measurements represent the same electrical component, their corresponding rows must have non-zero entries on the same columns, and these values must be the same, although the direction (sign) can be the opposite (*e.g.*, forward and backward line power flows). We define $UMsrSet_E$ as the set of measurements that represent the same electrical component (E). Each pair of sets, $UMsrSet_E$ and $UMsrSet_{E'}$, must satisfy the following property:

$$
\begin{aligned}
&\forall_Z \ \exists_X \ (h_{Z,X} \neq h_{Z',X}) \wedge (h_{Z,X} \neq -h_{Z',X}) \ \rightarrow \\
&\qquad (Z \in UMsrSet_E) \wedge (Z' \in UMsrSet_{E'}) \wedge (Z \neq Z')
\end{aligned}
$$

The power consumption at a bus is the summation of all the power flows incident to that bus. Thus, if all of these power flows are received as measurements, then the bus consumption measurement is redundant (*i.e.*, not unique).

From the mappings between communicating field devices and measurements, we can logically identify the successfully delivered measurements (*i.e.*, $\forall_I \ AssuredDelivery_I$), while from the mappings between the measurements and the states, we can find out whether the delivered measurements can observe the system. Let $IedSet$ be the set of IEDs that are responsible to take necessary measurements (meters/sensor data) and send them to the MTU through one or more RTU. If $MsrSet_I$ is the set of measurements transmitted by IED I and D_Z is a Boolean variable denoting whether measurement Z is successfully delivered, the following two conditions must hold if measurement Z is secured:

$$\forall_{I \in IedSet} \forall_Z \ (Z \in MsrSet_I \wedge AssuredDelivery_I) \rightarrow D_Z$$

If a measurement is successfully delivered, the variables corresponding to this measurement can be uniquely estimated. Let DE_X denote whether state X be estimated. Then:

$$\forall_Z \ \ \forall_{X \in StateSet_Z} \ \ D_Z \rightarrow DE_X$$

We define $DelUMsr_E$ to denote whether one or more measurements within $UMsrSet_E$ are successfully delivered:

$$\forall_E \ \ \exists_{Z \in UMsrSet_E} \ D_Z \rightarrow \ DelUMsr_E$$

A system is unobservable ($\neg Observability$) when either the delivered measurements does not cover all of the state variables or the number of delivered measurements, which are unique, are less than the number of (unknown) variables.

$$\neg Observability \rightarrow (\exists_X \ \neg DE_X) \vee (\sum_E \ DelUMsr_E < m)$$

$k-$Resilient Observability:

This constraint verifies whether observability is ensured even if k field devices (*i.e.*, IEDs and RTUs) are unavailable. If IED failures and RTU failures are considered differently, then this constraint can be specified as k_1, k_2-resilient observability, where k_1 is the number of IED failures and k_2 is that of RTU failures. A device is unavailable when it fails to communicate with the MTU. This is possible because of its technical failures or remote attacks (*e.g.*, DoS) on it or the communication route. As we have stated before, we model this constraint as a threat verification. That is, the modeling will verify if there is a set of devices, no more than k in number (or k_1 IEDs and k_2 RTUs), which can make the system unobservable when they are unavailable. This set is a threat vector that states that the system is not $k-$resilient. If there is not such a threat vector, then it is $k-$resilient observable.

Remember that $Node_i$ denotes if node i (an IED or an RTU) is available. The number of unavailable devices is computed by considering the available nodes. Let N be the number of devices, while N_1 and N_2 be that of IEDs and RTUs, respectively. Now, we formalize the threat against the $k-$resilient observability constraint ($\neg ResilientObservability$) as follows:

$$((N - \sum_{1 \leq i \leq N} Node_i) \leq k) \wedge \neg Observability \rightarrow \neg ResilientObservability$$

For the threat verification with respect to the k_1, k_2-resilient observability constraint, the above formalization turns to be:

$$\begin{aligned} &((N_1 - \sum_{1 \leq i \leq N_1} (Node_i \times Ied_i)) \leq k_1) \wedge \\ &((N_2 - \sum_{1 \leq i \leq N_2} (Node_i \times Rtu_i)) \leq k_1) \wedge \\ &\neg Observability \\ &\quad \rightarrow \ \neg ResilientObservability \end{aligned}$$

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

D. Formal Modeling of $k-$Resilient Secured Observability

The modeling of this resiliency constraint verification utilizes secured (assured) data delivery ($SecuredDelivery$). Unlike $AssuredDelivery$, $SecuredDelivery$ ensures that the data is delivered from the sender (*e.g.*, an IED) to the receiver (*e.g.*, an RTU) with necessary security measures (authenticated and encrypted) successfully.

Secured Data Delivery:

The assured data delivery constraint verifies whether data can reach from the source to the destination, *e.g.*, from a field device to the MTU, through zero, one, or more intermediate devices, but does not ensure if the transmission has occurred under necessary security measures. Although this constraint checks security pairing between the communicating parties, it is only to ensure necessary handshaking for communication.

In the secured data delivery constraint ($SecuredDelivery$), we verify whether data is sent under proper security measures, particularly authentication and integrity protection, including the assured data delivery. That is, the communicating nodes, *e.g.*, an RTU and the MTU, may have correct security pairing, as they are using the same security protocol Challenge-Handshake Authentication Protocol (CHAP). However, this security pairing on CHAP only ensures authentication. In this case, the transmission will not be data integrity protected. Moreover, we need 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.

The formalization of the secured data delivery includes two more constraints, $Authenticated$ and $IntegrityProtected$, that ensure the authentication of the communicating parties and the integrity of the transmitted data, respectively. In the following, we present the formalization of secured data delivery.

$$\begin{aligned} &\exists_K (\exists_k CryptType_{i,k} = K) \wedge \\ &\quad (\exists'_k CryptType_{i,k'} = K) \wedge \\ &\quad ((CAlgo_K = hmac \wedge CKey_K \geq 128) \vee \ldots)) \\ &\qquad \rightarrow Authenticated_{i,j} \end{aligned}$$

$$\begin{aligned} &\exists_K (\exists_k CryptType_{i,k} = K) \wedge \\ &\quad (\exists'_k CryptType_{i,k'} = K) \wedge \\ &\quad ((CAlgo_K = sha2 \wedge CKey_K \geq 128) \vee \ldots)) \\ &\qquad \rightarrow IntegrityProtected_{i,j} \end{aligned}$$

$$\begin{aligned} &Ied_I \wedge \\ &\exists_z \forall_{l \in |\mathcal{P}_{I,j,z}|} \ \{i', j'\} \in NodePair_l \wedge \\ &\quad Node_{i'} \wedge Node_{j'} \wedge Reachable_{i',j'} \wedge \\ &\quad CommPropPairing_{i',j'} \wedge CryptoPropPairing_{i',j'} \\ &\quad Authenticated_{i',j'} \wedge IntegrityProtected_{i',j'} \\ &\qquad \rightarrow \ SecuredDelivery_I \end{aligned}$$

Secured Observability:

We logically identify the secured measurements from the mappings between communicating field devices and measurements. Next, using the mappings between the secured measurements and the states, we find out whether the system is observable securely. Let S_Z be a Boolean variable denoting whether measurement Z is secured. Then, the following two conditions ensure if measurement Z is secured:

$$\forall_{I \in IedSet} \forall_Z \ (Z \in MsrSet_I \wedge SecuredDelivery_I) \rightarrow S_Z$$

If a measurement is secured, the variables corresponding to this measurement can be securely estimated. If SE_X denotes whether state X is securely estimated, then:

$$\forall_Z \ \ \forall_{X \in StateSet_Z} \ \ S_Z \rightarrow SE_X$$

We identify the set of securely delivered unique measurements (with respect to $UMsrSet_E$). If this set is denoted by $SecUMsr_E$, it is formed as follows:

$$\forall_E \ \ \exists_{Z \in UMsrSet_E} \ S_Z \rightarrow \ \ SecUMsr_E$$

The secured observability ($SecuredObservability$) ensures that the minimum number (*i.e.*, at least n) of secured measurements are received and all states are covered by these secured measurements. Thus, the system is securely unobservable ($\neg SecuredObservability$) when either or both of these two conditions fail:

$$\neg SecuredObservability \rightarrow (\exists_X \ \neg SE_X) \vee (\sum_E \ SecUMsr_E < n)$$

$k-$Resilient Secured Observability:

This constraint verifies whether secured observability is ensured even if k field devices (or k_1 IEDs and k_2 RTUs) are unavailable due to technical failures or cyber attacks. Similar to the $k-$resilient observability, we verify these properties by searching for threat vectors under the specification of maximum k failures.

When the number of unavailable devices is no larger than k devices (or k_1 IEDs and k_2 RTUs), we formalize the threat against the $k-$resilient secured observability constraint ($\neg ResilientSecuredObservability$) as follows:

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

$$((N_1 - \sum_{1 \leq i \leq N_1} (Node_i \times Ied_i)) \leq k_1) \wedge ((N_2 - \sum_{1 \leq i \leq N_2} (Node_i \times Rtu_i)) \leq k_1) \wedge \neg SecuredObservability \rightarrow \ \neg ResilientSecuredObservability$$

The threat vector ($\mathcal{V}$) includes a list of devices such that if they fail, the secured observability is impossible. In this way, this proposed modeling synthesizes attack vectors and, thus, it helps us learn the dependability breach points.

E. Formal Modeling of $k, r-$Resilient Bad Data Detectability

The obtained measurements for observability must be able to detect bad data. It is worth mentioning that a measurement can be delivered in a secured way, but the data itself can be incorrect or noisy due to random variations and other inaccuracies at the sensor/IED corresponding to this measurement, or if the measuring device is compromised. Such noisy measurements are considered as outliers with respect to the rest, since usually a few measurements can have such alterations. There are bad data detection algorithms to detect such bad measurements and eliminate them from the estimation process. We verify the resiliency of the bad data detectability in a formal way as follows.

r-Bad Data Detectability:

If there is a single measurement associated with a state, the measurement is a critical one. When such a measurement is bad, it is not possible to detect that. Therefore, in order to detect bad data, it is required to have at least two measurements corresponding to each state, if we assume no more than one measurement among them can be bad at a time. If we assume r measurements can be corrupted at a time, then it is r-bad data detectability. It is worth mentioning that we only rely on secured measurements for detecting the bad data, since non-secured measurements cannot be trusted [6]. Similar to the resilient observability, we verify the bad data detectability from the threat verification point of view.

If a measurement is secured, the state corresponding to this measurement can be securely estimated. We define $SE_{X,Z}$ as a Boolean variable that denotes whether state X is securely estimated by measurement Z. The following two equations valuate $SE_{X,Z}$ with respect to S_Z:

$$\forall_Z \ \forall_{X \in StateSet_Z} \ \ S_Z \rightarrow SE_{X,Z}$$

$$\forall_Z \ \forall_{X \in StateSet_Z} \ \ \neg S_Z \rightarrow \neg SE_{X,Z}$$

Now, a bad measurement is not detectable (*i.e.*, $\neg BadDataDetectability$) if the following condition holds:

$$\neg BadDataDetectability \rightarrow \exists_X \ (\sum_Z \ SE_{X,Z} < r + 1)$$

$k, r-$Resilient Bad Data Detectability Constraint:

We define $k, r-$resilient bad data detectability as if k devices (RTUs or IEDs) are unavailable, the bad data is still detectable even if r measurements are corrupted. Now, we verify the threats with respect to this requirement by extending the previous equation:

$$((N - \sum_{1 \leq i \leq N} Node_i) \leq k) \wedge \neg BadDataDetectability \rightarrow \ \neg ResilientBadDataDetectability$$

We can extend this k resiliency threat verification to k_1, k_2-resiliency verification.

IV. A Case Study

In this section, we briefly discuss the implementation of the model and illustrate the model's execution with an example.

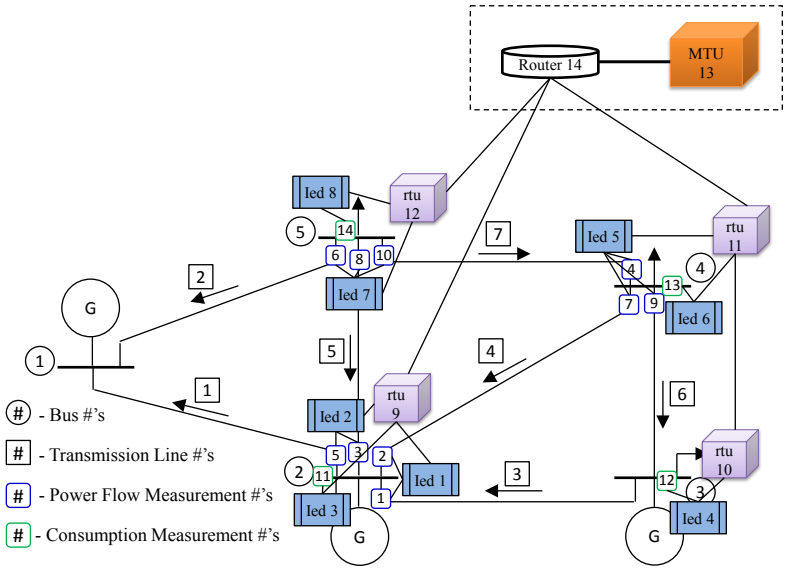

Fig. 3. An example SCADA topology of a 5-bus power grid.

A. Implementation

We use SMT logics [1] to encode the formalizations presented in the previous section. We use Boolean and integer terms in encoding. It is solved using Z3, an efficient SMT solver [5], [7]. The solution to the model gives a result as *sat* or *unsat*. In the case of *sat*, the solver provides elaborate result, specifically the values of the terms. From these valuations, we can find out the detailed scenario that makes the threat possible. For example, the result (particularly, $Node_i$ terms) shows us the devices (IEDs and/or RTUs) that are unavailable, and as a result, the (secured) observability is impossible. In the case of *unsat*, we can conclude that there is no threat scenario, *i.e.*, the failures of devices no more than given thresholds, that can make system unobservable.

B. Example: Scenario 1

This example considers a 5-bus SCADA system as shown in Fig. 3. This is an subsystem taken from the IEEE 14-Bus Test System [8]. The input is partially shown in Table II. The input includes primarily the Jacobian matrix corresponding to the bus system, the connectivity between the communicating devices, the association of the measurements with the IEDs, and security profiles of each communicating host pair. Each row of the Jacobian matrix corresponds to a measurement. The first row corresponds to measurement 1, and subsequent rows correspond to following measurements. Each row has 5 entries (columns) which correspond to 5 states/buses. We assume 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 RTUs. The server needs these measurements to estimate the current states of the system. The resiliency requirement specify that the secured observability must be satisfied even if one IED and one RTU are unavailable (due to having suffered from technical failures or cyber attacks). In this example, we demonstrate the k_1, k_2-resilient observability constraint. Thus, the security properties will not be used in this case.

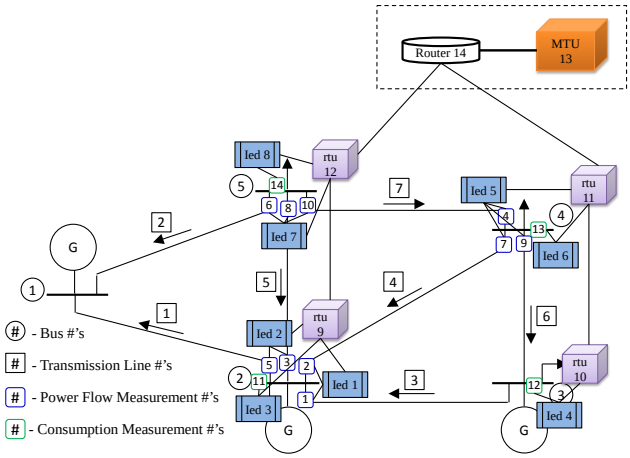

Fig. 4. The modified SCADA topology of the 5-bus power grid.

The solution to the formal model corresponding to this example returns *unsat*. That is, there is no resiliency threat vector that can make the system unobservable. The system is $(1, 1)-$resilient observable. However, if we increase the resiliency specification to $(2, 1)$, the model now provides a resiliency threat vector. The result shows that if IED 2, IED 7, and RTU 11 are unavailable, then the observability fails. It is worth mentioning that there are another 8 different threat vectors in this scenario that can make the system unobservable. In the case of IED failures only, the system can tolerate up to the failures of 3 IEDs.

Let us change the SCADA topology to Fig. 4. The difference with the previous topology is that RTU 9 is now connected to

TABLE II
THE INPUT TO THE CASE STUDY

```
# Number of states and measurements
5 14

# Jacobian matrix (mapping between the states and the measurements)
0 -5.05 5.05 0 0
0 -5.67 0 5.67 0
0 -5.75 0 0 5.75
0 0 0 -23.75 23.75
16.9 -16.9 0 0 0
4.48 0 0 0 -4.48
0 5.67 0 -5.67 0
0 5.75 0 0 -5.75
0 0 5.85 -5.85 0
0 0 0 23.75 -23.75
-16.9 33.37 -5.05 -5. 67 -5.75
0 -5.05 10.9 -5.85 0
0 -5.67 -5.85 41.85 -23.75
-4.48 -5.75 0 -23.75 37.95

# Number of each type of devices in the topology
# IEDs (Id 1-8), RTUs (Id 9-12), MTU (Id 13), Router (Id 14)
8 4 1 1

# Topology (Links)
13 #Number of communicating links
1 9
2 9
3 9
4 10
5 11
6 11
7 12
8 12
9 14
10 11
11 14
12 14

# Measurements corresponding to IEDs
1 1 2
2 3 5
3 11
4 12
5 4 7 9
6 13
7 6 8 10
8 14

# Security profile (if exists) between the communicating entities
11 # Number entries of security profiles
1 9 hmac 128
2 9 chap 64 sha2 128
3 9 chap 64 sha2 128
5 11 chap 64 sha2 256
6 11 chap 64 sha2 256
7 12 chap 64 sha2 128
8 12 chap 64 sha2 128
9 13 rsa 2048 aes 256
10 11 hmac 128
11 13 rsa 4096 aes 256
12 13 rsa 2048 aes 256

# k−resiliency requirements (IED, RTU)
1 1
```

RTU 12. In this case, $(1,1)-$resiliency verification fails. The model returns a satisfiable result, showing that if IED 4 and RTU 12 are unavailable, then the system is unobservable. We also find that this system (in this case) is not resilient to any RTU failure. If RTU 12 fails, there is no way to observe the system. This system is maximally $(3,0)$-resilient observable.

C. Example: Scenario 2

In this scenario, we demonstrate the k_1, k_2-resilient secured observability constraint. Let us consider the topology of Fig. 3 and the same inputs from Table II. In this case of $(1,1)-$resiliency verification, the model provides a *sat* result. That is, the system is not $(1,1)-$resilient in terms of secured observability, although it is $(1,1)-$resilient observable. According to the result, if IED 3 and RTU 11 are unavailable, it is not possible to observe the system securely. There are 4 more threat vectors that can make the system unobservable. This is because, as the result also shows, 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 anymore.

If we reduce the resiliency specification to $(1,0)$ or $(0,1)$, the model gives *unsat* result. That is, the system is securely observable even if any IED or RTU fails. If we consider the topology of Fig. 4, the system is not resilient any more for one RTU failure. However, there is only one threat vector (unavailability of RTU 12) to fail the secured observability.

V. EVALUATION

In this section, we present the evaluation results showing the scalability of the proposed resiliency verification framework with respect to the synthetic SCADA systems.

A. Methodology

We evaluate the scalability of the proposed verification model by analyzing the time requirements for executing the model in different problem sizes, particularly with respect to the number of buses. It is worth mentioning that the number of SCADA devices (IEDs and RTUs) and the number of links are not fixed for a specific bus size for a SCADA system. However, their number is usually proportional with the number of bus sizes. We generate the synthetic SCADA systems based on different sizes of IEEE test systems, *i.e.*, 14-bus, 30-bus, 57-bus, and 118-bus [8]. We arbitrarily create the SCADA network. On average, we choose one IED for two power flow measurements and one IED for each power consumption measurement. The communication path from an IED to the MTU is formed arbitrarily considering a parameter, hierarchy level. This hierarchy specifies the average number of intermediate RTUs on the path toward the MTU.

In addition, we also analyze the average maximum resiliency (or the number of resiliency threat vectors) of a SCADA system in different problem sizes, hierarchy levels, and resiliency specifications. We run our experiments on an Intel Core i5 Processor with 8 GB memory. We run a specific experiment several times and take the average of the results. In this evaluation, we did not compare the scalability or efficiency of our proposed model with other works, as no work we have found address a similar resiliency verification.

B. Scalability Evaluation

Impact of the Problem Size: Fig. 5(a) shows the execution time of the proposed resiliency framework for the $k-$resilient observability verification with respect to the problem size. We vary the problem size by considering different IEEE bus test systems. As the SCADA system is randomly generated, we take at least three random inputs for each type of experiment, while each specific experiment is run at least five times. The

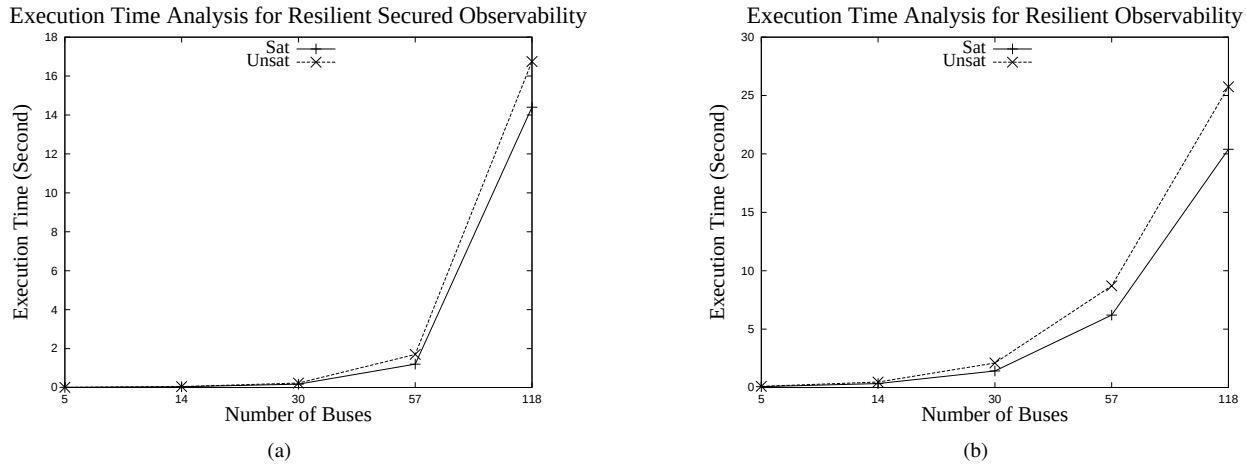

Fig. 5. The resilient observability model verification (execution) time with respect to the SCADA size (*i.e.*, the number of buses): (a) the k-resilient observability and (b) the k-resilient secured observability.

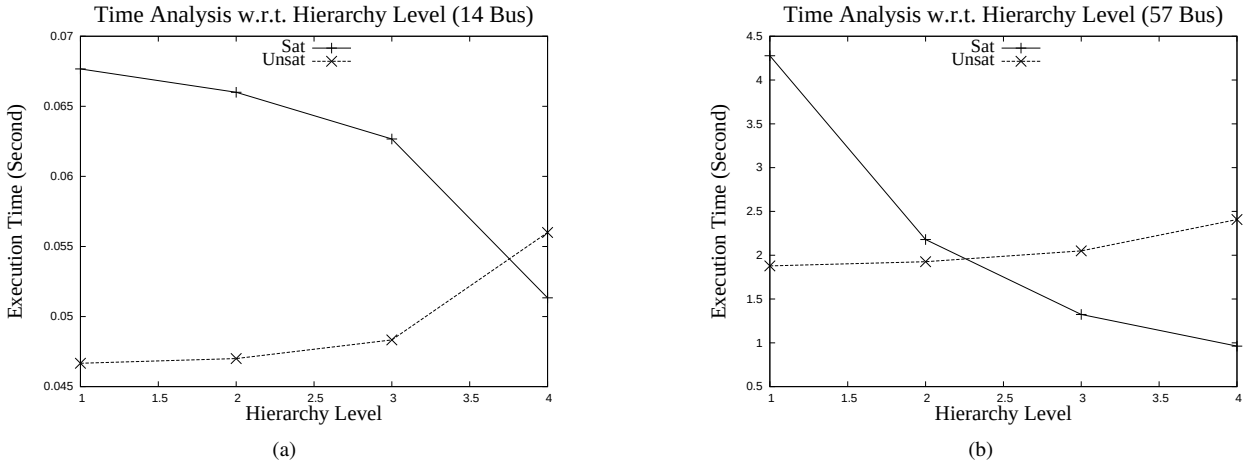

Fig. 6. (a) the execution time with respect to the communication hierarchy level (14-bus system), and (b) the execution time with respect to the communication hierarchy level (57-bus system).

execution time of each bus system is shown in Fig. 5(a). We observe that the increase in the execution time, with respect to the bus size, lies between linear and quadratic orders. For a specific bus size, we also observe that the execution time differs for *sat* and *unsat* results, while the latter takes a longer time than the former. We observe a similar growth in $k-$resilient secured observability verification (Fig. 5(b)), although the time requirement is a little high. This is because, in this case, the resiliency model needs to consider the constraints of secured communication, and as a result, the model size increases.

It is important to note that the problem size principally depends on two topologies: bus topology and SCADA topology. The bus topology mainly includes the buses and the power transmission lines, where the latter has a quadratic order with respect to the number of buses. The SCADA topology includes RTUs and IEDs whose numbers are proportional to the number of buses. The number of communication lines also follows a quadratic order in terms of the number of SCADA devices. Therefore, the complexity of the problem has a quadratic upper bound in general with respect to the number of buses. However, 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 [9]. The degree of a node in the SCADA topology is also very low. This feature can explain why the observed complexity is not strictly quadratic.

Impact of the Hierarchy Level: We also analyzed the impact of the hierarchy level on the model execution time. Fig. 6(a) and Fig. 6(b) present the evaluation results for the 14 and 57-bus test systems. The results shows that the execution time decreases for the satisfiable results while it mostly increases for the unsatisfiable results. This is because, with the increase of the hierarchy, some RTUs become important, and consequently the threat space increases. As a result, the search time for a threat vector decreases. With the unsatisfiable cases, to conclude that there is no threat vector, the model

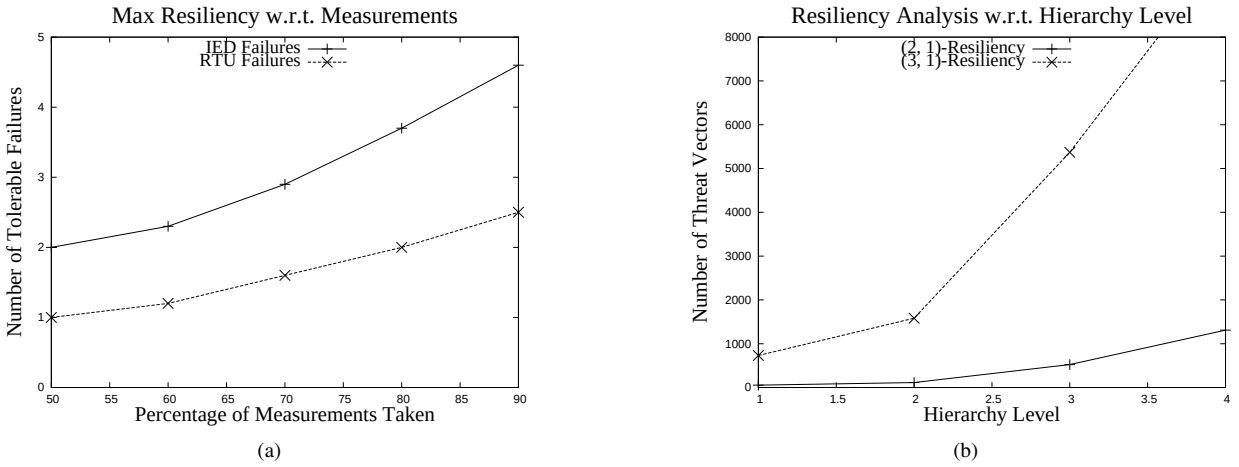

Fig. 7. (a) The average maximum resiliency with respect to the SCADA size (*i.e.*, the number of buses) and (b) the number of threat vectors with respect to the communication hierarchy level (14-bus system).

needs to search the whole problem space and thus execution time increases.

C. Resiliency Analysis

Maximum Resiliency Analysis: We run experiments in different SCADA networks with respect to the number of measurements for the 14-bus system. The number of measurements is represented as the percentage of the maximum possible measurements for a bus system [10]. We change the resiliency requirements with respect to the number of IED failures and RTU failures from smaller values to larger values. The result shows that the larger the number of measurements, the higher the maximum possible resiliency. Although this behavior depends on the SCADA network and the dependency among the measurements, IEDs, and RTUs, we consider arbitrary SCADA systems as we have described in Section V-A. The results are shown in Fig. 7(a). We find that an SCADA system can tolerate a larger number of IED failures than that of RTU failures. This is because the RTUs are often responsible for multiple IEDs for their communication with the MTU. As a result, one RTU failure often has larger impact compared to IED failures.

Resiliency Threat Space Analysis: In Fig. 7(b), we show the threat space varying the hierarchy level for the 14-bus system. We observe that the higher the hierarchy level, the larger the threat space (*i.e.*, the number of potential threat vectors). The increase in hierarchy creates more dependency among the RTUs and IEDs as there is more connectivity among the RTUs. As a result, there is more scope for contingencies that can violate the resiliency specifications, and so the threat space increases. Moreover, if we increase the resiliency specifications the threat space becomes larger.

VI. Related Work

The security policy misconfiguration and its verification have been studied extensively [11], [12], [13], [14] for more than a decade. In these approaches, the formal definition of configuration anomalies and safe deployment of single or multiple security devices are proposed and algorithms are presented to discover configuration inconsistency. There are also a number of works on risk-based security configuration analysis. Risk analysis and security hardening using attack graphs is proposed by several researchers [15], [16], [17]. However, all these above mentioned security analysis tools are proposed for analyzing misconfiguration problems in traditional networks. These tools cannot be applied for security or resiliency analysis in smart grids as it requires considering different cyber-physical operations and security controls.

A distinctive number of studies [4], [18], [19], [20] have been initiated to describe the security and resiliency issues based on different attack scenarios. These works also describe the functional operations of smart grid components with guidelines for reliable and robust communication among them. They advise that the energy providers or utilities cannot be trusted without a proper verification that required security measures have been implemented. Later, McLaughlin et al. [21] analyze the security and privacy challenges in smart grid networks. The authors also present an approach for penetration testing on AMI systems [22]. They develop archetypal and concrete attack trees for energy fraud, denial of service, and targeted disconnect attacks. Rahman et al. [23] develop a formal model based tool for the end-to-end security verification of advanced metering infrastructures in smart grids. However, this group of works do not analyze various misconfiguration problems and security controls on smart electric grid networks.

Wang et al. [24] present an artificial intelligence-based approach for analyzing risks in smart grid networks. However, in their analysis they do not consider network node, link, or communication failures or how that can effect the control procedures. Anwar et al. [25], [26] propose frameworks for modeling power grids and their control elements using first order logic. These frameworks are only capable of evaluating power flows and overloading violations in smart grids. Several research works [6], [27] study false data injection attacks in

power grids. The authors discuss the undetectability properties of these attacks at which adversaries can bypass the existing bad data detection algorithm. They consider different scenarios, such as limited access to meters, limited resources to compromise meters, under arbitrary or specific targets, assuming that the adversary has complete or incomplete information about the grid. Rahman et al. in [10], [28] present verification models for various UFDI attacks with respect to attacker's constraints and impact objectives. Yuan et al. [29] propose a variant of this kind of attack, which are known as a load redistribution attack.

Although the above mentioned research shows how stealthy attacks can be launched against a control process, namely the state estimation mechanism, they neither model data deliveries or secured communications nor resiliency properties of the system while some physical devices or communication links are facing cyber attacks or accidental failures. Therefore, there is still the need for modeling SCADA configurations and analyzing the security and resiliency properties. In this work, we address this need by developing a formal model for provably analyzing operational consistency, security, and resiliency in SCADA systems.

VII. Conclusion

A smart grid contains a large number of cyber and physical devices as well as various critical control mechanisms that exhibit highly dependent configuration parameters leading to increased potential for security and resiliency vulnerabilities. In this paper, we present an automated formal framework for verifying the SCADA resiliency. We focus on security configurations, controls, and resiliency requirements that are important for protecting smart grids in various contingencies. We create a logic-based verification model and use SMT to solve this model as a constraint satisfaction problem. Our proposed framework performs static configuration analysis in order to determine potential threats as violations to the smart grid security and resiliency requirements. We demonstrate the framework using a test configuration and show its efficacy. The scalability evaluation of the framework shows that the execution time lies within 30 seconds for a SCADA system with 400 physical devices (IEDs and RTUs). In future, we would like to consider the automated synthesis of necessary configurations for resilient SCADA systems satisfying the security and resiliency requirements.

References

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

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

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

[4] Nistir 7628: Guidelines for smart grid cyber security. Smart Grid Interoperability Panel- Cyber Security Working Group, August 2010. http://www.nist.gov/smartgrid/upload/nistir-7628_total.pdf.

[5] L. de Moura and N. Bjrner. Z3: An efficient smt solver. In *Conf. on Tools and Algo. for the Construction and Analysis of Systems*, 2008.

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

[7] Z3: Theorem prover. Microsoft Research, 2013. http://research.microsoft.com/en-us/um/redmond/projects/z3/.

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

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

[10] M.A. Rahman, E. Al-Shaer, and 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.

[11] E. Al-Shaer and H. Hamed. Discovery of policy anomalies in distributed firewalls. In *23rd IEEE International Conference on Computer Communications (INFOCOM)*, pages 2605–2616, 2004.

[12] X. Ou, S. Govindavajhala, and A. Appel. Mulval: A logic-based network security analyzer. In *14th USENIX Security Symposium*, pages 113–128, 2005.

[13] E. Al-Shaer, W. Marrero, A. El-Atawy, and K. Elbadawi. Network configuration in a box: Towards end-to-end verification of network reachability and security. In *IEEE International Conference on Network Protocols (ICNP)*, pages 107–116, NY, USA, 2009.

[14] P. Bera, S. Ghosh, and P. Dasgupta. Policy based security analysis in enterprise networks: A formal approach. *IEEE Transactions on Network and Service Management*, 7(4):231–243, 2010.

[15] Steven Noel and Sushil Jajodia. Attack graphs for sensor placement, alert prioritization, and attack response. In *Cyberspace Research Workshop of Air Force Cyberspace Symposium*, Shreveport, Louisiana, 2007.

[16] R. Dewri, N. Poolsappsi, I. Ray, and D. Whitley. Optimal security hardening using multi-objective optimization on attack tree models of networks. In *14th ACM conference on Computer and Communications Security (CCS)*, pages 204–213, 2007.

[17] J. Homer and X. Ou. Sat-solving approaches to context-aware enterprise network security management. 27(3):315–322, 2009.

[18] Guide to industrial control systems (ics) security. NIST Special Publication 800-82 (Revision 1), May 2013. http://dx.doi.org/10.6028/NIST.SP.800-82r1.

[19] ABB Group. Security in the smart grid, 2009. http://www02.abb.com/db/db0003/db002698.nsf/0/832c29e54746dd0fc12576400024ef16/\$file/paper_Security+in+the+Smart+Grid+(Sept+09)_docnum.pdf.

[20] Honeywell. The communications requirements of electric utilities. http://energy.gov/gc/downloads/nbp-rfi\-communications\-requirements-honeywell\-responses-request-information-rfi.

[21] S. McLaughlin, D. Podkuiko, and P. McDaniel. Energy theft in the advanced metering infrastructure. In *4th International Workshop on Critical Information Infrastructure Security*, pages 176–187, 2009.

[22] S. McLaughlin, D. Podkuiko, S. Miadzvezhanka, A. Delozier, and P. McDaniel. Multi-vendor penetration testing in the advanced metering infrastructure. In *26th ACSAC*, pages 107–116, 2010.

[23] M.A. Rahman, E. Al-Shaer, and P. Bera. Smartanalyzer: A noninvasive security threat analyzer for ami smart grid. In *31st IEEE International Conference on Computer Communications*, pages 2255–2263, 2012.

[24] Y. Wang, D. Ruan, J. Xu, M. Wen, and L. Deng. Computational intelligence algorithms analysis for smart grid cyber security. *Advances in Swarm Intelligence*, 6146:77–84, 2010.

[25] Z. Anwar, R. Shankesi, and R. H. Campbell. Automatic security assessment of critical cyber-infrastructures. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 366–375, 2008.

[26] Z. Anwar and R. H. Campbell. Automated assessment of critical infrastructures for compliance to cip best practices. In *2nd IFIP WG 11.10 International Conference on Critical Infrastructure Protection*, Arlington, Virginia, 2008.

[27] 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.

[28] M.A. Rahman, E. Al-Shaer, and R. Kavasseri. Impact analysis of topology poisoning attacks on economic operation of the smart power grid. In *Distributed Computing Systems (ICDCS), 2014 IEEE 34th International Conference on*, pages 649–659, June 2014.

[29] Yanling Yuan, Zuyi Li, and Kui Ren. Quantitative analysis of load redistribution attacks in power systems. *IEEE Transactions on Parallel and Distributed Systems*, 23(9):1731–1738, Sept 2012.