Runtime Semantic Security Analysis to Detect and Mitigate Control-Related Attacks in Power Grids

Hui Lin, Adam Slagell, Zbigniew Kalbarczyk, *Member*, IEEE, Peter W. Sauer, *Fellow*, IEEE, and Ravishankar K. Iyer, *Fellow*, IEEE

***Abstract*—In this paper, we analyze *control-related attacks* in SCADA (*S*upervisory *C*ontrol *A*nd *D*ata *A*cquisition) systems for power grids. This class of attacks introduces a serious threat to power systems because attackers can directly change the system's physical configuration using malicious control commands crafted in legitimate format. To detect such attacks, we propose a semantic analysis framework that integrates network intrusion detection systems (IDSs) with a power flow analysis capable of estimating the execution consequences of control commands. To balance detection accuracy and latency, the parameters of the power flow analysis algorithm are dynamically adapted according to real-time system dynamics. Our experiments on IEEE 24-bus, 30-bus, 39-bus systems, and a 2736-bus system demonstrate that (1) by opening 3 transmission lines, an attacker can put the tested system into an insecure state, and (2) the semantic analysis can complete detection in 200 milliseconds for the large-scale 2736-bus system with about 0.78% false positives and 0.01% false negatives, which allow for timely responses to intrusions.**

***Index Terms*—SCADA, intrusion detection system, semantic analysis, Bro, adaptive power flow analysis**

SUMMARY OF NOTATION

	Notations to Represent Power Systems
n	the number of generators in the power system
m	the number of buses in the power system
k, i	index of buses
j	imaginary unit
$\bar{V}_k = V_k \angle \theta_k$	voltage at bus k including magnitude V_k and angle θ_k
P_k^g, Q_k^g	real/reactive power generated at bus k
P_k^l, Q_k^l	demand of real/reactive power at bus k
Y, G, B	admittance matrix, which can be decomposed into two real-valued matrices: $Y = G + jB$; G_{ik} denotes the conductance and B_{ik} denotes the susceptance of the transmission line that connects bus i and bus k.
	Notations to Represent Control Systems
A, H	state and control matrix for a linear time-invariant control system
x	state variable for a control system
$u=F(x)$	input variable for a control system; function F represents state feedback control mechanisms
	Other Notations
p_a	the probability that the random attacks introduce insecure physical impacts
k_a	the number of physical components that are perturbed by the attackers
	Abbreviations
SCADA	Supervisory control and data acquisition
IDS	Intrusion detection system
DNP	Distributed network protocol
CA	Contingency analysis
CS	Contingency selection

Manuscript received July 13, 2015; This work was supported in part by the Department of Energy, the National Science Foundation, and the National Security Agency.

H. Lin, Z. Kalbarczyk, P. W. Sauer, and R. K. Iyer are with the Electrical and Computer Engineering Department, University of Illinois, Urbana, IL, 61801, USA (e-mail: hlin33@illinois.edu, kalbarcz@illinois.edu, psauer@illinois.edu, rkiyer@illinois.edu). A. Slagell is with the National Center for Supercomputing Applications at the University of Illinois, Urbana, IL, 61801, USA (e-mail: slagell@illinois.edu).

I. INTRODUCTION

In power grids, SCADA (***S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition*) systems are used to collect sensor measurements to monitor system state and deliver control commands for maintenance or economical purpose. The control commands can change the physical configuration of the power grid, e.g., the topology of the transmission network.

From the Northeast blackout of 2003, we learned that inappropriate changes of the grid's physical configuration can have catastrophic consequences, such as blackout. To make matters worse, the analysis presented in [1] claims that even small-scale attacks can have a wide range of impact. In today's power grid, SCADA systems are being deployed in the commercial network infrastructure. Even though the SCADA systems are not open to the public Internet, they can be remotely penetrated through corporate networks or personal devices used by employees. Current incidents and studies show that corporate networks can be penetrated by using stolen passwords [29], breaking poor encryption [2], installing and exploiting backdoors [3][5], and Trojans [4]. Also, attackers can compromise employees' personal devices, such as laptops or USB drives, which can be connected to the SCADA systems [6]. Because many communication protocols used by SCADA systems still lack security features, e.g., authentication and encryption, attackers can sniff the network and use the information obtained to inflict malicious changes to the grid configuration.

While both sensor measurements and control commands can be transmitted by communication network, the existing research, e.g., false data injection attacks [7][9][10][11], focuses on attacks that exploit corrupted sensor measurements of voltages, currents, and power usage; the impact of

compromising control fields of the network packets, e.g., the index of the breakers to disconnect, has not been well studied.

In this paper, we study a class of attacks referred to as *control-related attacks*, in which attackers modify control fields in network packets exchanged between SCADA and power substations. Instead of focusing on the extreme outage of power system components [45][46][47][48], the study is performed on the perturbation on the system that is within a normal range of legitimate operations or can be combined with normal operations. The control-related attack can become a serious threat to power grids for two reasons: (1) it can directly result in catastrophic losses or consequences, e.g., overloaded transmission lines or generators, and (2) it is undetectable by anomaly-based network *intrusion detection systems* (IDSs) because the modified control fields in a data packet can be encoded in the legitimate format. As smart grid technology introduces more control functionalities over network communication, this family of attacks will continue growing in the near future.

In this paper, we provide an in-depth study of control-related attacks targeting power systems. Our work includes introduction of real attack scenarios and quantitative assessment of the attack's physical impact in the context of simulated power system configurations (including a 2736-bus real-world power system). Some recent work [12] indicates threats of corrupting control commands in industrial control system environments, e.g., water plants [13][14], but this research cannot be used directly in power systems. The outages of substations and transmission lines due to accidents or attacks are evaluated [45][46][47][48], but it is not clear how practical the attacks can be performed through communication networks.

Because the control-related attack can introduce no anomaly in the syntax of network communication, detecting it requires understanding the semantics of payloads carried by network packets, e.g., the consequences (to the power grid) of delivering and executing the payload. For this purpose, we propose a semantic analysis framework to detect control-related attacks by using the knowledge of both the cyber and physical infrastructure of the power grid [15]. Specifically, in the semantic analysis framework, the IDS [1] identifies control commands on the SCADA network, extracts control fields in network packets, and at runtime, invokes power flow analysis software to perform look-ahead evaluation on the execution consequences of the control commands issued by SCADA. The proposed semantic analysis framework detects malicious commands at their *first appearance*, which makes it possible to deploy a timely response.

This paper makes the following contributions and we use Figure 1 to present the high-level relations among these contributions:

A theoretical base for demonstrating the impact of control-related attacks. We represent the power system in mathematical formulation, e.g., simplified control-theoretic formulation. By mapping the malicious changes of control fields in network traffic into the mathematical formulation, we demonstrate how systems' states can be maliciously modified.

A semantic analysis framework. The control network is monitored to identify control commands; semantics related to the control commands, e.g., which breakers to open, and the updated measurements are extracted from the network and delivered to power flow analysis tools to evaluate the physical consequences of executing the identified commands.

A rapid adaptive power flow analysis algorithm for runtime detection. To accurately detect the attack with short latency, the general AC power flow analysis algorithm dynamically adapts its parameters, e.g., the number of iterations, based on the semantics extracted from the control commands.

An intrusion response mechanism and its evaluation. We propose a response mechanism that exploits existing reclose logic in relays to prevent physical damage caused by an attempt to disconnect multiple transmission lines. Based on this response mechanism, we demonstrate the benefits of using the rapid adaptive power flow analysis algorithm.

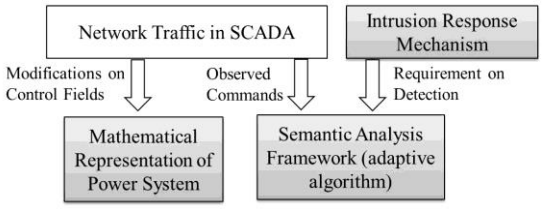


Figure 1. Illustration of relations among attacks, detections, and responses researched in the paper.

The remainder of this paper is organized as follows. In Section II, we present an overview of the control-related attack and compare it with attacks researched before. Section III presents the threat model. In Section IV, the semantic analysis framework to detect malicious commands is proposed. In Section V, we propose the adaptive power flow analysis and an intrusion response mechanism. Experimental implementation is described in Section VI. In Section VII and VIII, the evaluations on the control-related attacks and the proposed semantic analysis framework are presented. We conclude in Section IX.

II. Control-Related Attack Overview

In this section, we use Figure 2 to: (1) briefly demonstrate the mathematical representation of a power grid; (2) illustrate how attacks on the control fields in network packets (i.e., control-related attacks) can be mapped to the mathematical representation, in order to analyze their impacts; and (3) position the control-related attacks with respect to more broadly studied attacks, which are denoted by *A* and *B* in Figure 2.

Figure 2 shows a common communication structure used by SCADA systems [16]. In the current generation, IP-based network communication and intelligent devices are commonly being deployed to enable more accurate and efficient control at lower cost.

Generally, we can classify SCADA operations into two types: reactive and preemptive operations. In the reactive

[1] The used IDS was developed based on Bro [15] and included in Bro's standard distribution, which can be downloaded from Bro's website.

operation, state-estimation software collects from sensors the measurements of voltages, currents, and power usage to estimate the power system's state [30]. The result of the state estimation can lead to a feedback-control operation, denoted by u in Figure 2, which is performed via commands issued by the SCADA master. In addition to reactive operations, power system operators more commonly use the SCADA master to preemptively issue control commands for maintenance purposes, e.g., scheduled line outage.

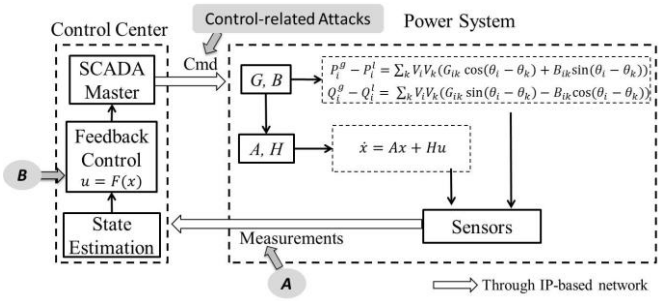


Figure 2. Control operations in SCADA systems.

A. Mathematical Representations of the Power System

A power system is composed of buses (representing substations) that are connected through transmission lines. In our discussion, the power system is represented by two mathematical models, as shown in Figure 2: (1) *power flow equations* with the parameters that capture the system's physical configuration, e.g., G and B which denote the conductance and susceptance of transmission lines, and (2) *control-theoretic formulation* with the parameters that capture the system's dynamics, e.g., A and H, which denote the state transition and control matrices.

The state of the system is specified by the voltage magnitude and the angle for each bus, i.e., (V_i, θ_i) in equations (1) and (2). For each bus i, two power flow equations are formulated based on the fact that the generated power (P_i^g and Q_i^g), the consumed power (P_i^l and Q_i^l), and the power delivered to other buses (indexed by k) are balanced at each timestamp [32].

$$P_i^g - P_i^l = \sum_k V_i V_k (G_{ik} \cos(\theta_i - \theta_k) + B_{ik} \sin(\theta_i - \theta_k)) \quad (1)$$

$$Q_i^g - Q_i^l = \sum_k V_i V_k (G_{ik} \sin(\theta_i - \theta_k) - B_{ik} \cos(\theta_i - \theta_k)) \quad (2)$$

The power system's steady state is obtained by solving equations (1) and (2). Because equations (1) and (2) are nonlinear equations, there are two groups of approaches to solve them: AC power flow analysis uses iterative algorithms, e.g., Newton-Raphson algorithm, to calculate solutions that are within a predefined error threshold; DC power flow analysis solves the linear approximation of equations (1) and (2) in order to get the solution more quickly.

To further evaluate the dynamic behavior of the power system, swing equations are formulated for each generator to establish mathematical relationships between the frequency/angle of the mechanical rotors and the generated electrical power [32]. The swing equations can be linearized under the same assumptions that are used as the foundation for the DC power flow analysis: (1) the system states are close to nominal values, i.e., $V_i \approx 1$, $|\theta_i - \theta_k| \approx 0$; and (2) the power network is lossless, i.e., $G_{ik} \approx 0$. The linearized swing equations for all generators can be grouped together in a control-theoretic formulation:

$$\dot{x} = Ax + Hu \quad (3)$$

In equation (3), x is a vector of size $2n$, where n is the number of generators in the power system. The first n entries in x are the rotor angle for all generators, the last n entries are the generators' rotor frequency, $\dot{x}$ is a vector that includes the first derivative of each entry in x, and u is a vector of control inputs expressed as a feedback control operation, i.e., $u = F(x)$. The control inputs are used at runtime to enforce that the rotor angle and frequency deviate in a certain range, e.g., in the United States, the rotor frequency is usually constrained in the range of 60 ± 0.5 Hz.

The structure of matrix A varies with different mechanical models of generators. To simplify the explanation, we use the classical model, in which each generator is simulated as a voltage source of constant magnitude connected in series with a constant reactance. In this case, A is a $2n$-by-$2n$ matrix with each of four n-by-n submatrices having different structures. For example, Figure 3 presents A's structure for a power system with two generators. The top two submatrices are zero and identity matrices. The left bottom submatrix is an "aggregated" representation of the power system's network topology related to the generators' mechanical rotors, and it is derived from G and B [36]. Although this submatrix hides the detailed topological aspects, e.g., the line's conductance and susceptance, it establishes the mathematical relations between the generators' rotors and simplifies the analysis of the power system's dynamics. The bottom-right submatrix is a diagonal matrix with the diagonal entries related to model parameters of each generator, e.g., the normalized inertia constant and damping constant [32].

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & 0 & a_{44} \end{bmatrix}$$

Figure 3. The structure of matrix A for a power system with two generators.

The structure of matrix H varies with different feedback-control algorithms. For example, if a turbine-governor control is used for each generator to restrict the rotor frequency in a certain range, H becomes a diagonal matrix with the diagonal entries related to the regulation constant [32].

Because all generators are connected through the network of transmission lines, if the topology of the transmission network is changed, entries in G and B, are changed accordingly, which further lead to changes in A and H.

B. Impacts of Control-Related Attacks

We use the introduced mathematical representation of the power system and the SCADA system communication structure (see Figure 2 and equations 1–3) to describe the impact of the control-related attacks. The control-related attacks exploit legitimate commands, which include both reactive and preemptive ones issued by the SCADA master, e.g., scheduled line outage, generation or load demand adjustment. Direct

malicious modification of the control fields in the commands can significantly change the power system's physical configuration, e.g., the topology of the transmission network, alter the power flow, and put the system in an insecure state. The changes in the grid's physical configuration can be mapped in the corresponding modifications of:

(1) Entries in matrices *G* and *B*; as a result, the power system transits into different steady states as dictated by the solution of equations (1) and (2); and

(2) Entries in matrices *A* and *H*; as a result, equation (3) is transformed into a different one, $\dot{x} = A'x + H'u'$, and the dynamic behavior of rotor frequency and angles become unpredictable to system operators.

The change in the power system's steady state can make power flow on the transmission lines exceed their physical constraints and cause them to overload. Similarly, a large deviation of the rotor frequency can violate the generator's physical constraints and cause the generator to overload. Because overloaded transmission lines and generators can be automatically disconnected by circuit breakers, these cascading changes can quickly degrade the grid operation and lead to catastrophic consequences, e.g., blackout.

In this paper, we focus on the impacts of control-related attacks on a system's steady state. Specifically, a power system is in an *insecure state* if at least one transmission line violates its physical constraints determined by the power flow limit.

C. Related Work

In this section, we position the control-related attacks to the attack scenarios researched before and compare the proposed semantic analysis framework (described in Section IV) with the existing detection mechanisms.

1) Attacks on Feedback-Control Loops

We classify the malicious attacks that target the *feedback-control loop* into two types (see Figure 2). In *Type A*, which is often referred to as false or bad data injection attacks, attackers introduce malicious measurements that affect the outcome of state estimation [7][8]. [9][33][53] study the range of measurements that need to be compromised in order to make the injected measurement undetectable. Qin et al. propose a different attack strategy that further reduces the number of compromised measurements [34]. In this strategy, even though the attack can be detected, it is challenging to identify the compromised measurements and, thus, to make the corresponding remedy. Under a *Type A* attack, incorrect system states are estimated and can have negative impact on power grids. For example, Xie et al. studied the economic impacts of the compromised measurements [35]. But how the incorrect system states can lead to damage of the physical infrastructure lacks sufficient research.

The *Type B* attack shown in Figure 2 is mainly discussed in [38]. DeMarco et al. exploit a control-theoretic approach to study the impact of malicious feedback control algorithms on a power system, i.e., the expression of *u=F(*x) in Figure 2. The modified control algorithms can mislead system operators into issuing wrong commands. Their paper assumes that the attackers have full control over a generator, which can be challenging to achieve in practice through the control network.

Both *Type A* and *Type B* attacks perturb the feedback-control loop of the power system, which can indirectly impact the issued control commands, i.e., reactive commands. However, in today's power grid, commands, including both reactive and preemptive ones, are more frequently transmitted over the IP-based control network. Consequently, after gaining access to the control network, the attackers have more incentives to compromise control commands, which can directly change the state of the power system. This is not to say that the attacks on sensor measurements are not important. Quite the opposite, compromised measurements can be used to hide the real (potentially anomalous) state of the power grid in order to delay the detection of the attacks before the actual damage to the system (as seen in the example of Stuxnet [6] and in the recent study [17]).

2) Compromise of Physical Infrastructure of Power Grids

The risks that the physical infrastructure of a power grid is compromised by attackers has drawn the attention of many researchers. The purpose is to identify and rank the vulnerable physical components in the power grid, e.g., substations or transmission lines. To achieve this goal, the metrics of a power system's electrical characteristics, e.g., the load of substation or transmission lines, can be used. For example, high-order contingencies are selected and ranked based on different DC power flow analysis algorithms [39][40]. Additionally, research uses the characteristics of the transmission network, e.g., connectivity or the length of the shortest path between substations, to identify the vulnerable components [41][42][69]. Recently, the computer system vulnerabilities identified by network IDSs are also included as a selection and ranking metric [43].

The similar risk analysis can also be applied to the cascaded attacks in which an adversary perturbs a power system by a sequence of events. A brief discussion on the risk of the cascaded outage caused by accidents or attacks is presented in [44]. Zhang et al. experimentally demonstrate that the cascaded attack can introduce more significant damage than the attacks that perturb multiple physical components simultaneously [46]. In addition to selecting and ranking physical components independently, the interdependence among these components, such as the outage order, are used to identify the power system's vulnerable components [45][46][47][48]. In [18], the authors further compare the analysis of cascading failure based on the steady state and the transient state.

In practice, due to the limit of computation capabilities, only top-ranked incidents are usually considered for further analysis. However, the motivation to perform an attack is not only decided by the damage the attack may cause, but also the practicality/cost to implement the attack, and chances of being detected [49]. For example, to cause the outage of a single substation is not a trivial task, because it may require compromising multiple network packets, which can introduce a detectable network-level anomaly. In order to avoid detection, the attacker may intentionally avoid the strategy causing the most severe damage. In the control-related attack, we take the

possibility of being detected into account and study the perturbation on the power system that is within a normal range of legitimate operations or can be combined with normal operations. We believe that a more comprehensive study of attacks that considers the characteristics of both cyber and physical infrastructure of the power grid can be especially beneficial and we leave this for future work.

3) Network IDS for SCADA Systems

To detect malicious activities in SCADA systems, previously proposed network IDSs usually rely on deviations from predefined or profiled normal communication patterns in the control network. For example, work in [15][50] defines the normal patterns based on SCADA protocols, and [51] adopts machine-learning techniques to cluster normal and abnormal communication patterns. However, control-related attacks can rely on legitimate commands with malicious contents, which can easily circumvent such detection mechanisms. Work in [19] correlates the local information in smart grids to detect attacks. In [19], the distributed IDS is based on anomaly-based methods. In our work, the IDS instances use the specification-based approach, which relies on the knowledge of a grid's physical infrastructure to detect cyber-attacks [25][26].

To detect intrusion targeting on a power grid, an IDS should take its cyber-physical characteristics into consideration [20][21][22]. In [12][36][37][68], the statistic characteristics of the sensor measurements or historical data are used to detect penetration into power systems. Based on these methods, detection takes place after the physical damage is done to the system, and it can require modification of existing SCADA systems. In [23], the authors combine the network activities, e.g., the information of suspicious log in of SCADA systems, and their possible physical impacts to detect penetrations. The anomaly-based methods can usually suffer from high false-positive rate and the detection results can be difficult to interpret [24]. In [13][52][54], a blacklist for malicious system states can be built through simulation. At runtime, the observed system change is compared with the blacklist to detect malicious changes. Because the power system state is continuously changing, it is challenging to cover all possible attack cases using the blacklist. Furthermore, building a blacklist may not scale for large-scale systems, such as the 2736-bus system considered in our experiments.

Carcano et al. propose a concept of a state-based network IDS that includes physical information to detect attacks in power systems [55]. This concept is consistent with the principle based on which we design the semantic analysis framework. However, the proposed semantic analysis framework further includes the practical constraints encountered in the power grid. First, [55] proposes using alerts or static patterns from the network IDS to trigger the analysis of network contents; however, in practice, the malicious commands may be encoded in a legitimate format without introducing any anomaly at the network level. The semantic analysis framework relies on the runtime network analyzer that we specifically develop for SCADA systems and can extract and analyze all SCADA-specific semantics [20]. Consequently, we can provide better accuracy and flexibility in deciding when and how to use the knowledge of the physical infrastructure. Second, [55] relies on a centralized image to detect attacks, while we deploy IDS instances in a master/slave architecture to detect the compromise of measurements or control commands during the communication. Third, the semantic analysis framework integrates the proposed adaptive power flow analysis algorithm, which can balance the detection latency and accuracy; the proposed algorithm plays an important role if response mechanisms are deployed.

III. Threat Model

We make the following assumption about the threat model considered in this paper:

- In the *control center*, we assume that attackers can remotely penetrate the local area network environment and, thus, are able to sniff and inject network packets that are received and delivered to remote substations. Remote penetration does not grant the attackers the same capability as operators in the control center, but it is more practical to achieve in today's SCADA systems. Even though it is not open to the public Internet, the network used in the control center can be penetrated indirectly through corporate networks or personal devices. Current incidents and studies show that corporate networks can be penetrated many ways [3][4]. For example, [28] demonstrates that attackers can use social engineering techniques, e.g., spamming emails or phishing, to obtain credentials that allow them to remotely login to computers used in SCADA systems. By exploiting the vulnerabilities in workstations or switches in the control centers [5], the attackers can further obtain the privilege necessary to install malware, sniff, inject, and even modify network traffic. Another common way is to penetrate employees' personal devices, e.g., laptops, smart phones, or USB drives, which usually do not have sufficient protection. When these devices are brought to work, i.e., “bring your own device” (BYOD), they can be connected to the network of the control center and start distributing malware if a given device is compromised by the attackers [6].

  Even though it is challenging for attackers to penetrate the control center, the consequence is severe once they successfully do that. The attackers can stay undetected from network monitoring for a long period of time until they collect sufficient information to launch attacks. Recent reports show that even after a vulnerability is identified, it can still take a long time to patch it [29]. Currently, the SCADA systems collect information on substations (such as the measurements of power usage and the status of circuit breakers) using communication protocols that usually lack security features such as authentication or encryption. Consequently, the attackers can choose to passively sniff network communications to learn the system configuration, such as the topology of the transmission network without affecting any normal operation. By using the collected information, the attackers can estimate the system states [30][31], design an attack strategy, and then at the right time, inject the malicious traffic to cause physical damage [6].

- In the *substations*, we do not trust "intelligent" devices, e.g., data aggregators, that can run full-featured operating systems (OS). As most proprietary SCADA protocols lack security features such as authentication or encryption, we assume that an attacker can install malicious software on those devices to modify the control commands when they are received and delivered by these devices. Similarly, the untrusted intelligent devices can also compromise measurements and combine the false data injection attacks with the control-related attacks (see the scenario in Figure 4).

  We trust devices equipped with proprietary industrial control functionality, e.g., sensors and actuators. Consequently, we can collect trusted measurements from sensors, which are treated as the root of trust in the design of the semantic analysis framework (discussed in Section IV). As discussed in [7], false data injection attacks can be achieved through two major channels: (1) manipulation of measurements before they are used for state estimation, or (2) physical tampering with sensing devices. Although we do not make any assumptions about the trustworthiness of data from intelligent devices upstream of sensors, we trust the information, e.g., voltage, current, and power usage, at the sensors. Concurrent physical accesses to and tampering with a large number of distributed sensors (across multiple substations) are hard to achieve in practice. Also, as indicated in [53], it is sufficient to protect "a strategically selected set of sensor measurements" to detect false data injection attacks.

  We classify control commands into two types: *manual commands* are issued by the control center through an IP-based network; *automatic commands* are issued through hardwired connections to locally protect physical components against short-circuit faults. We assume that only *manual commands* can be maliciously exploited.
- We do trust the functionality of the semantic analysis framework. We can dedicate a separate machine to run the required applications. Because the semantic analysis framework passively detects malicious commands without injecting or modifying any network packets, it does not introduce additional vulnerability to power grids.

Under this threat model, we present an example attack scenario as a sequence of steps to demonstrate a possible penetration procedure (shown in Figure 4).

Attack Entry Points. An attacker may penetrate a control center or field devices in substations as an insider or by remote access, e.g., by exploiting vulnerable software.

Attack Preparation Stage. An attacker can obtain data on power usage and breakers' status, and based on this information, estimate system state and determine network topology [30][31]. Then, the attacker can decide on the attack strategy, e.g., which transmission lines to open to cause maximum damage with minimum effort.

Alternatively, an attacker can open transmission lines at random when a power system operates under high generation and load demand. Our study (see Section VII for details) demonstrates that the random attack strategy can put the system into an insecure state. To avoid possible detection, the attack preparation stage can be performed offline.

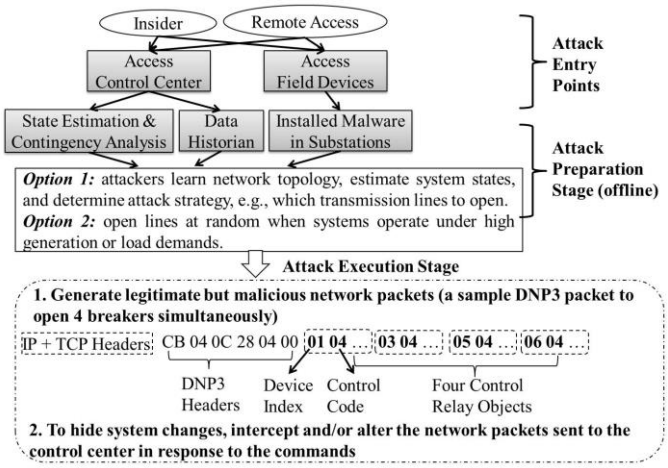


Figure 4. Attack steps to impact the physical infrastructure of a power grid.

Attack Execution Stage. The attacker can generate legitimate but malicious commands by replaying or modifying proprietary network packets. In this paper, we use the DNP3 protocol, a proprietary protocol widely used in power grids, as an example [27]. In *step 1* shown in Figure 4, a single DNP3 network packet includes four *control relay objects* to operate four breakers located in the same substation. Each control relay object uses a one-byte *device index* to indicate which breaker to operate and a one-byte *control code* to indicate the command to be performed. By modifying the *device indices* and the *control codes*, an attacker can change the selected breakers and the operations performed on them. In *step 2*, to hide the system changes, the attacker can intercept network packets and/or use the technique of false data injection attacks to alter the packets' payloads sent to the control center in response to the commands. If successful, the attacker can open four transmission lines simultaneously and put the system into an insecure state. Meanwhile, the false data injection attacks can be used to provide the control center with measurement data indicating error-free operation of the substation.

IV. Semantic Analysis Framework

In this section, we present the overall architecture of the semantic analysis framework to enable detection and mitigation of control-related attacks [15].

Why do we need the semantic analysis framework? Control-related attacks are hard to detect based solely on:

- *Monitoring the power systems' electrical state* because (1) traditional contingency analysis considers low-order incidents, i.e., the *N-1* contingency[2]; (2) traditional state estimation is performed periodically, detecting attacks after physical damage; (3) an attacker can hide changes in the

[2] Frequently, DC power flow analysis is used to rank high-order contingencies based on different criteria, e.g., the loss of real power. Because of limited computational time, only a small number of contingencies are accurately evaluated and, hence, the system changes introduced by malicious attackers can be easily missed.

physical system by replaying (or modifying) measurements that mislead the operator and indicate an error-free system state [7][12]; and (4) building a black list or white list of control commands in advance [54] is insufficient; evaluating the consequences of all control commands requires accurate high-order contingency analysis, which is impractical because of limited computation power; or

- *Using a network IDS* because the maliciously crafted control commands are encoded in the correct syntax and, hence, are not detectable by traditional network IDS, which validates the command syntax or monitors statistical anomalies in command control fields [13][15].

In order to detect the control-related attack, the proposed semantic analysis framework estimates at runtime the execution consequence of network packets by: (1) combining system knowledge on both the cyber and physical infrastructures in the power grid and (2) integrating network monitoring with look-ahead power flow analysis. By monitoring actual messages in the network and obtaining the ground truth regarding the state of the power system, we can look ahead to the actual physical state transition caused by commands (delivered as part of the packet payload) and thereby detect and mitigate attacks on the first appearance of the maliciously crafted command, rather than identifying the physical damage after the fact.

A. Overall Architecture

In Figure 5, we present the architecture of the semantic analysis framework, which is integrated with the communication structure presented in Figure 2. In Figure 5, we distinguish trusted and untrusted components with different colors based on our threat model. Note that, because we use DNP3 as an example of the network protocol, the "DNP3 slave" in Figure 5 represents the intelligent devices used in substations, e.g., data aggregators that receive network traffic and deliver the commands to sensors or actuators. Because the DNP3 slave can run full-featured OS, as demonstrated in a video demo,[3] it is possible to install malware on such devices in order to perform man-in-the-middle attacks on measurements and commands.

In the semantic analysis framework, a network IDS monitors the communication of the control network. In this paper, we focus on the detection of malicious *manual commands*. Even if the attacker gains physical access to relays or actuators and issues *automatic commands*, e.g., by connecting to the relay through a serial port, the commands' executions should be reported through broadcast messages to other neighbor substations or the control center, as recommended by IEEE standard 1646 [56]. Consequently, the semantic analysis framework can rely on the broadcast messages to detect malicious automatic commands.

The control network usually adopts proprietary network protocols to transmit commands. To support proprietary network protocols, we have implemented a DNP3 analyzer on top of Bro, a specification-based IDS [57][58] (the DNP3 analyzer is now included in Bro's standard distribution). With the help of the DNP3 analyzer, Bro can efficiently validate the syntax of the network packets and detect cyber-attacks that result in observable anomaly at the protocol level. Moreover, the DNP3 analyzer allows Bro extracting semantics related to control commands, which are further evaluated by power flow analysis.

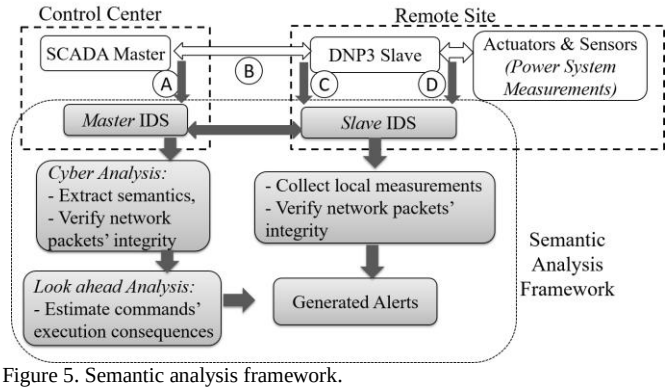


Figure 5. Semantic analysis framework.

To integrate power flow analysis into network IDS, we introduce two IDS instances with different functions and make them work collaboratively in a master/slave configuration (as shown in Figure 5). In practice, a master IDS can be deployed in the local area network of the control center. The centralized master IDS can be connected to multiple slave IDSs deployed in the local area network of remote substations.

B. Master IDS

To accurately look ahead to the state changes caused by a command's execution, the master IDS performs analyses on both the cyber and physical infrastructures of the power system. The *cyber analysis* monitors the control network, extracts specific parameters related to the control command, e.g., the *device indices* and the *control codes* in the DNP3 network packets that control relays, as shown in Figure 4, and verifies the integrity of the network packets (see Section IV.C for details).

TABLE 1. COMMAND CLASSIFICATION BASED ON DNP3.

Command Type	Description
Read	Retrieve measurements from remote substations, e.g., read binary outputs
Write (Critical)	Configure intelligent field devices, e.g., open, edit, and close a configuration file
Execute (Critical)	Operate actuators or sensors, e.g., open or close a breaker of a relay

The DNP3 analyzer allows Bro IDS to distinguish critical SCADA commands. Table 1 presents a classification of commands in the context of the DNP3 protocol. The *read* commands are *passive*, meaning that they do not make any changes to substations. The *write* and *execute* commands are invasive, meaning that they can reconfigure or change a substation. Consequently, we consider *write* and *execute* to be more critical commands than *read*. Based on this classification, IDS can select critical commands to analyze. Those control functionalities are common in power grids, so similar classifications can be applied to other protocols as well.

[3] https://www.youtube.com/watch?v=unb7b8myNvA

Based on the parameters obtained from the cyber analysis, the master IDS performs look-ahead power flow analysis on critical commands to evaluate their execution consequence, which is specified as *look-ahead analysis* in Figure 5. In our threat model, the trusted measurements are collected by sensors in substations, which are specified by *Activity D* in Figure 5. These measurements include the value of active/reactive power in substations with only load units and/or the value of voltage magnitude and active power in substations with generators. Delivering the measurements over the control network (*Activity B*) may introduce additional latency to semantic analysis. Such latency can be critical if an intrusion response mechanism is to be triggered. Therefore, the master IDS performs the semantic analysis based on the measurements collected at the control center (denoted by *Activity A*) and validates the integrity of the measurements concurrently.

Some SCADA protocols can further reduce the effort to collect measurements. For example, to lessen the network traffic, the DNP3 protocol allows the control center to retrieve measurements whose values are changed since the last sampling time. Also, it was recently proposed that power system state can be locally estimated for a subset of connected substations, and the local results are further used to calculate the global state [59].

C. Slave IDS

Under our threat model, sensor measurements can be corrupted during the transmission from the substation to the control center, e.g., by the DNP3 slave in Figure 5. As a result, the semantic analysis may use inaccurate measurements. Similarly, control commands can also be modified after they pass the semantic analysis.

To address this problem, the *slave IDS* is deployed locally in the remote substation to collect trusted measurements directly from sensors and to monitor the commands that are executed on actuators (denoted by *Activity D*). The slave IDS communicates with the master IDS, so that the network packets observed at different locations, e.g., the ones marked by *A*, *C*, and *D* in Figure 5, can be compared to reveal possible compromises. Although the slave IDS is assigned a simple task, its deployment allows the semantic analysis to be performed by the master IDS at a command's first appearance.

The communication between master and slave IDS can be established over the existing network with the protection provided by standard security protocols, such as SSL/TLS. Furthermore, today's sensing devices are deploying Ethernet interfaces [60], which make it possible for a slave IDS to collect the trusted measurements from sensors and monitor the commands on actuators.

V. Adapting Semantic Analysis Framework for Intrusion Response

In this section, we first *adapt* the classical AC power flow analysis algorithm to support low-latency detection of malicious control commands. Then, we introduce an intrusion-response mechanism that neither affects the system's normal operations nor adds new vulnerabilities that could be exploited by attackers. The purpose of proposing this response mechanism is to demonstrate how the adaptive power flow analysis algorithm makes the trade-off between detection accuracy and latency.

A. Adapting AC Power Flow Analysis Algorithm

The classical AC power flow analysis algorithm uses iterative algorithms, e.g., the Newton-Raphson algorithm, to accurately calculate the power system's state. In the iterative algorithms, a long time is usually spent getting the solution to converge within a predefined threshold.

To avoid iterative computations, different types of DC power flow analyses are used to solve the linear approximation of the nonlinear power flow equations [39]. Because computation time is significantly reduced, these methods are used to select and rank high-order contingencies. However, solutions obtained from DC power flow analyses can be inaccurate. For example, the study in [61] demonstrates that the calculation error introduced by DC power flow analysis can be as high as 35%. Consequently, if the semantic analysis framework uses the DC power flow analysis algorithm, the resulting calculation errors can lead to a large number of false detections on both malicious and normal commands.

Instead of directly using AC or DC power flow analyses, we adapt the Newton-Raphson iterative algorithm to find the trade-off between the calculation accuracy and latency. *First*, we use the most recent known system state at the time when a malicious command is issued as the initial guess of the solution to the power flow equations. *Second*, the convergence threshold is set such that the estimated system state is accurate enough to correctly decide whether the system is in an insecure state, i.e., whether one or more transmission lines are overloaded. These two approaches are commonly used in today's power flow analysis algorithm.

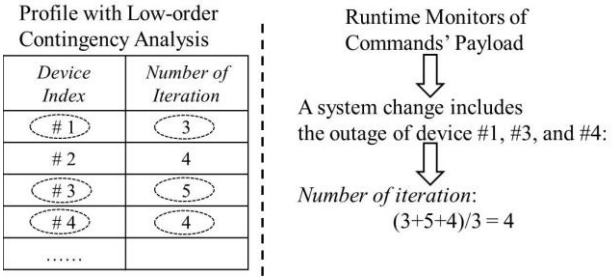


Figure 6. Dynamically adapting the number of iterations.

Third, we adapt the number of iterations that the iterative algorithm uses to estimate the power system state. Instead of statically fixing this parameter, e.g., being fixed by one loop of iteration in [40], we dynamically adapt the number of iterations based on the control fields of SCADA packets observed at runtime. Specifically, when a disturbance of multiple devices is observed, the number of iterations to analyze it is assigned as the *average number of iterations* that the classical AC power flow analysis takes to analyze the disturbance of each involved device, i.e., the *N-1* contingency analysis. Based on our experiments, we find that the number of iterations used to analyze *N-1* contingency analysis varies in a small range. Consequently, we can limit the number of iterations used by the adaptive algorithm, which can reduce the detection latency.

Additionally, the dynamic adaption allows using more iterations if one of the involved devices need more iterations to analyze, which can balance the detection accuracy and the latency. As shown in Figure 6, for each *N-1* contingency, we record in a profile the number of iterations after which the classical AC power flow analysis converges. When the disturbance of devices 1, 3, and 4 is observed, the adaptive algorithm uses 4 iterations to calculate the system states, which are further used as detection.

TABLE 2. COMPARISONS OF POWER FLOW ANALYSIS ALGORITHMS USED IN LOWER-ORDER CONTINGENCY ANALYSIS (CA), HIGH-ORDER CONTINGENCY SELECTION (CS), AND THE SEMANTIC ANALYSIS.

	Low-Order CA	**High-Order CS**	**Semantic Analysis**
Pros	AC analysis calculates *highly accurate* state	*Low latency* to rank contingencies for further analysis	*Medium latency* and *medium accuracy* by setting number of iteration dynamically
Cons	*Long latency* to wait all solutions to converge	*Low accuracy* with linear approximation to estimate system states	Need to periodically profile the number of iterations for each low-order contingency

In Table 2, we summarize positives and negatives of the power flow analysis algorithms used for low-order contingency analysis, high-order contingency selection, and the proposed semantic analysis.

B. Intrusion Response

Before proposing specific intrusion response mechanisms, we study the timeline of steps that occur when a control command is executed (shown in Figure 7) and use this timeline to analyze two categories of response mechanisms: *delay command execution* and *reverse command execution.*

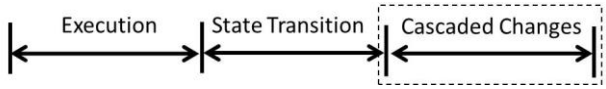


Figure 7. Timeline of steps of executing a command.

The *Execution* stage encompasses the delivery and execution of a control command. Specifically, substation devices make physical changes after receiving the command, e.g., opening multiple circuit breakers. After the command's execution, e.g., circuit breakers are opened and the corresponding transmission lines are disconnected, the power system experiences transient changes and may ultimately reach a new steady state. This period is represented by the *State Transition* stage. In the new steady state, if one or more transmission lines are overloaded, the steady state is regarded as insecure.[4] When in the insecure steady state, the power system further experiences the *Cascaded Changes* stage, in which the changes initiated by the malicious command propagate through the whole system. During this stage, the protection relays automatically operate in an attempt to contain the spread of potential damage. If not successful, the power system may collapse, e.g., causing a blackout.

Although the *command delaying* mechanism prevents malicious physical changes from being initiated, it requires the semantic analysis to complete before the command begins its execution. This puts strict time constraints on the semantic analysis. In addition, delaying a command may require the interception of every normal power system operation. Consequently, it is very challenging to design and implement a response mechanism based on this concept.

Instead, we use the *command-reversing* mechanism, which remedies the impact of malicious commands. We take advantage of the fact that this mechanism is widely deployed in practice to handle small system disturbances. For example, relays are usually equipped with reclosing logic that can immediately auto-reclose an unwanted breaker trip command to restore the system state. In this case, existing logic can be reused to design and implement a command-reversing intrusion response mechanism. Because it is challenging to remedy the damage spread to the whole power system, reverse commands need to be sent before the malicious commands transit into the cascaded changes stage.

C. Intrusion Response: Case Study

In this section, we present an example intrusion response mechanism designed for the attack scenario in which an intruder issues malicious manual commands to open one or more circuit breakers to disconnect multiple transmission lines simultaneously. We assume that under normal conditions those circuit breakers are closed.

The proposed intrusion response mechanism uses the command-reversing concept and reuses a reclosing logic that already presents in relays. Specifically, when a command issued to open circuit breakers is determined as malicious, the response mechanism recloses the breakers.

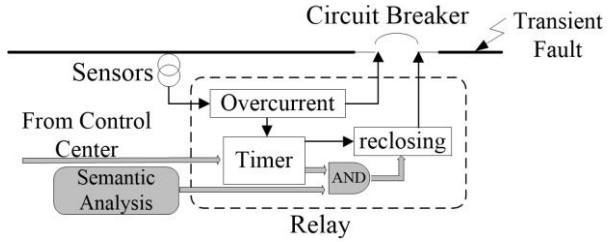


Figure 8. Reclosing logic for accidental events and its extension (shaded part) to support response to malicious commands.

We use an example protection scheme, shown in Figure 8, to explain how the original reclosing logic reacts to a transient fault on a transmission line, e.g., lightning strikes a transmission line. In this protection scheme, a relay operates a circuit breaker to connect or disconnect a transmission line through a hardwired connection.

To facilitate the explanation, we include in Figure 8 key components of the reclosing logic implemented in real relays based on the descriptions in [62]. An *overcurrent* component is connected through a hardwired connection to sensors that monitor line current. When a transient fault occurs, the line is

[4] After some system changes, the power system may lose synchronism and never reach a steady state [32]. To detect this phenomenon, methods other than power flow analysis should be used. We leave this for future work.

short-circuited to the ground and its current magnitude dramatically increases. When the current magnitude exceeds a predefined threshold, the overcurrent component initiates an *automatic* command to open the breaker. To better describe the reclosing procedure, we use $t_{execute}$ to denote the time that the relay needs to process and execute a command. After the breaker is opened, the overcurrent component starts a timer that expires after a period of time $t_{reclose}$ and then recloses the breaker. During this period, the transient fault can clear itself, so the original system state is restored after the breaker is reclosed. Based on the relay design documents [62], the execution time of automatic command is usually restricted to no more than 10 clock cycles, i.e., approximately 167 *ms* for 60HZ frequency; the reclose time is generally set as 500 *ms* [62].

To make the reclosing logic work for malicious commands, the relay initiates the timer when it receives a manual command from the control center (as shown in the shaded part of Figure 8). We use an AND gate (or similar logic) to combine the detection result of the semantic analysis framework and the output of the timer. Consequently, when the timer expires, the breakers are reclosed only if the control command is malicious.

As a result, the detection latency, which is defined as the time needed to complete the semantic analysis, must satisfy the following condition: $t_{semantic} \leq t_{execute} + t_{reclose} = 167 + 500 = 667ms$. The adaptive power flow analysis algorithm described in Section V.A allows us to meet this condition (see Section VIII.C for details).

By using this intrusion response mechanism, a false positive detection can involve manual intervention. When this happens, a manual command that opens a breaker is mistakenly detected as malicious, the proposed response mechanism recloses the breaker, and the intended command is not executed. In order to resolve the false positive detection, the system operator needs to disable the detection, reissue the commands, and enable the detection again. Consequently, it is critical to use the adaptive power flow analysis algorithm to: (1) reduce the rate of false positive detection and (2) finish the detection with short latency so that the detection (including the false positive ones) can be reported in seconds.

VI. Evaluation Test Bed Setup

To perform experiments, we set up a test bed, which consists of a physical machine with an Intel i3 (3.07 GHz) quad-core processor and 8GB memory running the Ubuntu 12.04 operating system. In this test bed, we perform the evaluation of the control-related attacks and the proposed semantic analysis framework, whose results are presented in Section VII and Section VIII.

Network Communication. The network communication is implemented based on the structure shown in Figure 5. To produce different types of synthetic DNP3 network traffic, e.g., the *read*, *write*, and *execute* commands in Table 1, a SCADA master and a DNP3 slave are implemented based on the open DNP3 library [63] and running on two virtual machines.

Specifically, the *read* command is issued every second and implemented by a DNP3 network packet that reads all sensor measurements. The *write* command, simulated as a Poisson process with average arrival interval of 50 seconds, is implemented by a DNP3 network packet that sets analog values, e.g., generation and load adjustment by automatic generation control device or load-shedding controller [64][65]. The *execute* command, simulated as another Poisson process with average arrival interval of 100 seconds, is implemented by a DNP3 network packet that sets binary values, e.g., the status of a relay's binary outputs, which usually control the status of electrical breakers.

In our attack scenario, the maliciously crafted commands are encoded in correct syntax. Consequently, the same SCADA master is used to issue both legitimate and malicious commands.

Power System Simulation. To simulate small-scale power systems, we use IEEE 24-bus, 30-bus, and 39-bus systems whose baseline configurations are included in Matpower, a MATLAB toolbox for power flow analysis [66]. We also use a 2736-bus power system in MATPOWER to simulate a large-scale power system. This power system represents the Polish 400-, 220-, and 110-kV networks during the summer of 2004. Because we focus on the impacts of control-related attacks on a system's steady state in this paper, implementing the feedback control for a power system is not necessary for our experiments. Furthermore, we calculate the system state by using the power flow analysis, which does not perform the bad data detection of the state estimation.

The parameters of simulated power systems are modified based on control commands in the simulated network communication. The power flow analysis module (provided by MATPOWER) is used to analyze power systems' steady state and identify the number of overloaded transmission lines. The results are stored locally as a ground truth to be used for validating the detection mechanism provided by the semantic analysis framework.

VII. Evaluation of Control-Related Attacks

The control-related attack is mimicked by perturbing physical components of the simulated power network as follows: (1) select a set of generators (we exclude the generator with the *slack bus*) and increase or decrease their outputs by at most 50%, (2) select a set of load units and increase or decrease their demands by at most 50%, (3) open a set of transmission lines, and (4) combine these three types of perturbations. The threshold of 50% is set based on the figures of load/generation changes from [67]. As shown in [67], such variation of load/generation can happen within one week. Directly observing such variation requires intelligent attack plans, such as the one in [6], where the attackers penetrate and stay in the power grids during that period. A smart attacker, however, can indirectly estimate the variations of load/generation based on the environmental factors, such as the local temperature. For each perturbation, we use the AC power flow analysis to calculate and decide whether the perturbation puts the system into an insecure state.

We further consider two scenarios of applying the perturbations. *Scenario 1* (random attacks): separately perturb

generators, alter load units, or open transmission lines at random. *Scenario 2* (targeted attacks): perturb vulnerable components, e.g., open transmission lines carrying power that is more than 70% of their power flow limit. In both scenarios, we regard *an attack as successful* if the system is put into an insecure state, i.e., at least one transmission line becomes overloaded, and the probability of successful attacks (denoted by p_a) is measured. The value of p_a reflects how easily attackers can perturb the power system even if they obtain only a little knowledge related to the grid.

A. Scenario 1: Random Attacks

Random attacks target generators, load units, and transmission lines separately. We perform 1000 random attempts for the small-scale test systems (24-bus, 30-bus, and 39-bus systems) and 100 attempts for the large-scale test system (2736-bus system). After each run, we measure the probability (p_a) of successful attacks, which is defined as the ratio between the number of successful attacks and the total number of attempts.

Plots in Figure 9 present the probability of successful attacks (p_a) as a function of the number of perturbed components (k_a). For the attacks on generators and load units whose results are shown in Figure 9(a) and Figure 9(b), the *x*-axis indicates the percentage of perturbed components. For attacks on transmission lines whose results are shown in Figure 9(c), the *x*-axis indicates the number of transmission lines that are disconnected.

Generally, the value of p_a increases as the value of k increases. An exception is the 30-bus system, where no successful attack is observed when we perturb the generation. As a result, there is no curve corresponding to the 30-bus system in Figure 9(a). Further analysis of the 30-bus system indicates that 29 out of 41 transmission lines carry power that is less than 30% of the line power flow limit. Consequently, after the generation is increased due to the simulated attack, there are sufficient margins for transmission lines to carry more power without violating the power flow limits. In Figure 9(c), the value of p_a for the 2736-bus system is small, compared with the other three small-scale test systems. For the 2736-bus system, 10 transmission lines correspond to less than 0.3% of all transmission lines in use. As a result, opening 10 transmission lines is perceived as a small disturbance and, hence, the likelihood of a successful attack is also small.

We also perform coordinated attacks by randomly opening k_a transmission lines while simultaneously increasing the power generation and load demand. For small-scale test systems, we randomly select 3 generators and 3 load units. For the large-scale test system, we randomly select 20 generators and 20 load units. Figure 9(d) shows how p_a changes as a function of k_a, i.e., the number of line outages. Compared with Figure 9(c), p_a for coordinated attack increases two- to three-fold. However, the coordinated attack requires more perturbed components and, hence, it is easier to detect malicious activities. To avoid the detection, an opportunistic (or smart) attacker may wait until the system becomes heavily loaded, i.e., the grid devices are working close to their physical limits, and then force additional outages of transmission lines to cause cascading failures or potential blackout. Note that a significant fluctuation in energy demand can happen naturally, e.g., in the Midwest on June 28, 2012, the energy demand was about 70% higher than it was on June 2, 2012 [67].

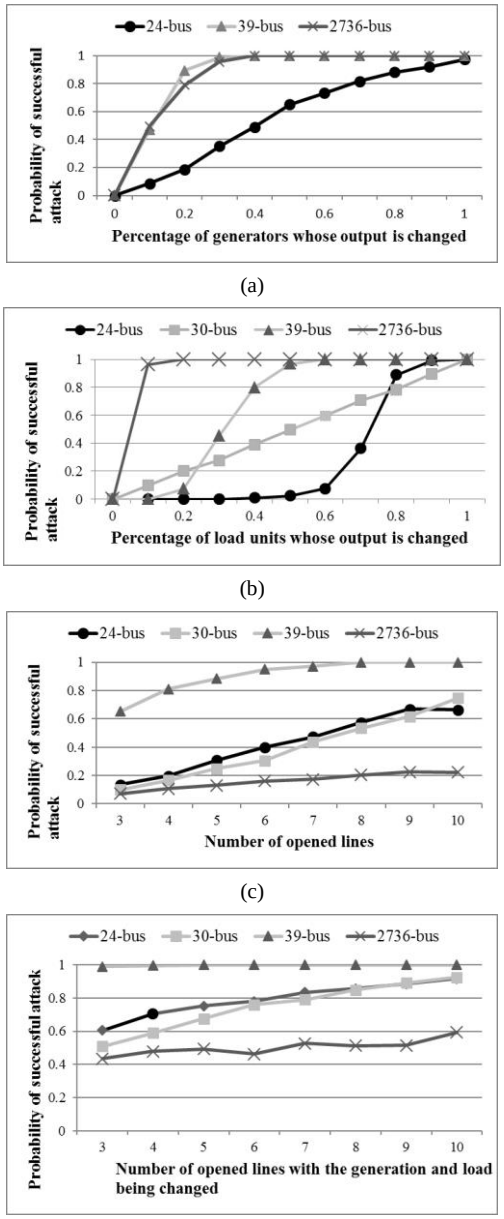


Figure 9. The probability of successful attacks as a function of the number of perturbed components.

B. Scenario 2: Targeted Attacks

To perform the targeted attacks, k_a vulnerable transmission lines are opened without changing the power generation or load demand.

Figure 10 plots p_a as a function of k_a. Compared with the results in Figure 9(d), the probability of successful attacks, p_a, increases for the same value of k. Most interestingly, for the large-scale test system where opening less than 10 randomly selected transmission lines is considered a small disturbance, one can see a dramatic increase of p_a, above 90%, when only three vulnerable lines are opened simultaneously. Also, for the

30-bus system whose transmission lines have sufficient margins to carry more power, p_a increases to above 90% when six vulnerable lines are opened simultaneously.

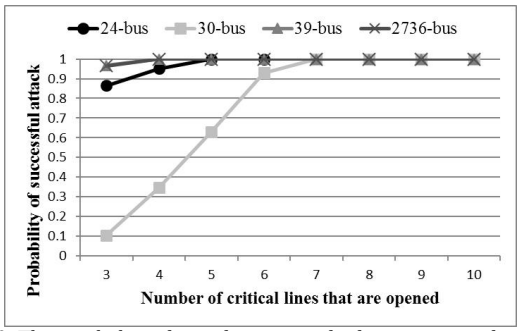


Figure 10. The attack that selectively opens multiple transmission lines.

VIII. Evaluation of Semantic Analysis Framework

In this section, we evaluate the detection latency and accuracy of the proposed rapid adaptive power flow analysis algorithm. The experimental results demonstrate that the adaptive algorithm allows the semantic analysis framework to achieve rapid detection and timely intrusion response. The semantic analysis also includes cyber analysis whose evaluation can be found in [15].

A. Detection Accuracy

We use the false positive and negative ratio to estimate the detection accuracy of the adaptive power flow analysis algorithm. Specifically, we use the classical AC power flow analysis algorithm, the adaptive power flow analysis algorithm, and the DC power flow analysis algorithm to calculate the system states. Then, the same set of *power flow limits* included in MATPOWER is used to decide whether the system is in an insecure state. We regard the power system state calculated from the classical AC power flow analysis as ground truths, and we regard the corresponding detection as accurate. Both adaptive power flow analysis and DC power flow analysis make the false positive detection if they find at least one overloaded transmission line when there are actually no overloaded lines. Similarly, a false negative detection is made if no overloaded lines are found but actually at least one line is overloaded based on the calculation of the classical AC power flow analysis.

We use the Newton-Raphson algorithm in the classical AC power flow analysis: the flat voltage profile is used as the initial solution (i.e., $V_i = 1$ and $\theta_i = 0°$ for each bus), 50 loops are set as the maximum number of iterations to calculate power system states, and the convergence threshold is set as the 10^{-6} in per unit for the resultant real power and reactive power of each transmission line.

To configure the adaptive power flow analysis algorithm, we first use the aforementioned classical AC power flow analysis to calculate an accurate system state on the base power generation and load demand. Then, we use this state as the initial solution to analyze the considered system changes. We set the convergence threshold as 10^{-3} in per unit, which is the same precision level as the power flow limit included in MATPOWER.

To decide the number of iterations to evaluate malicious perturbations on multiple devices, we first use the same AC power flow analysis to perform an *N-1* contingency analysis on the perturbations of each single device and record the number of iterations to calculate the accurate system state. In our experiments, we find that even when all generations and load demands are increased or decreased up to 50%, the number of iterations does not vary significantly. Consequently, in this paper, we only adjust the iterations when the outage of multiple transmission lines is observed. Specifically, when the outage of multiple lines is evaluated, the number of iterations is set as the *average number of iterations* that are used to analyze the outage of each involved transmission line.

Both the proposed adaptive algorithm and the DC power flow analysis can introduce calculation errors. To compare the calculation errors, we follow the evaluation procedure used in [48]. In each perturbation, we use the adaptive algorithm and the DC power flow analysis to calculate the real power of each transmission line. As suggested by [48], we filter out the perturbations for which the classical AC power flow analysis does not converge and focus on the real power of the transmission line that is loaded above 70% of its power flow limit.

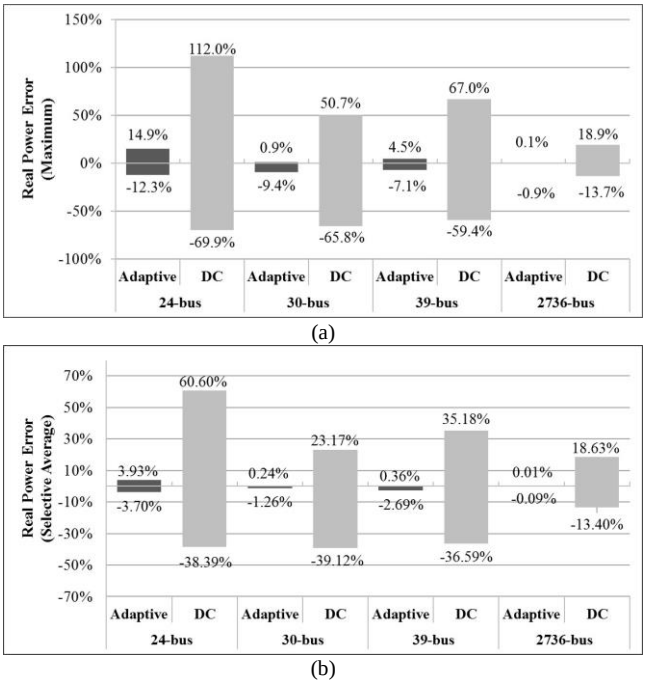


Figure 11. Real power error caused by the adaptive and DC power flow analysis.

Figure 11 compares the calculation errors, both positive and negative, caused by the proposed adaptive and the DC power flow analysis algorithms. The *x*-axis indicates the types of the simulated systems; for each system, we put the results from these two algorithms, which are specified by bars filled with different colors, side by side. The *y*-axis specifies the real power errors, which are normalized to the results obtained by using the classical AC power flow analysis. In Figure 11(a), we present the maximum positive and negative errors when two algorithms are used. Because of the large number of perturbations in our experiments, we select the top 1000 maximum positive and negative errors and present the average of these errors in Figure

11(b). As shown in Figure 11, the real power errors caused by the proposed adaptive algorithm are at least one order of magnitude smaller than those caused by the DC power flow analysis. Furthermore, the calculation errors caused by the adaptive algorithm are within 15% in the worst case, i.e., the 24-bus system; such calculation accuracy can help to reduce the false detections.

In this paper, we consider the system to be in an insecure state if there is at least one overloaded transmission line. As a result, making the correct decision as to whether a perturbation puts the system into an insecure state is decided by the calculated real power and also the power flow limit of each transmission line.

To demonstrate the accuracy of the detections of insecure perturbations, Table 3 shows the rate of false positives (*FP*) and negatives (*FN*) for the adaptive power flow analysis algorithm (*Adaptive*) and the DC algorithm included in MATPOWER (*DC*). If the DC power flow analysis is used in semantic analysis, large numbers of false positives and false negatives are expected. The false negatives do not affect the power system's normal operation. However, in the worst case, e.g., with 20% false negatives for the IEEE 30-bus system, the system is vulnerable to control-related attacks, i.e., the changes in the power system caused by the malicious command may not be detected. On the other hand, false positives usually require further action, e.g., manual inspection and a response to mitigate the detected problem. With a high false positive rate, an operator may intervene frequently without there being an actual need for action.

TABLE 3. DETECTION ACCURACY BASED ON THE ADAPTIVE POWER FLOW ANALYSIS AND DC POWER FLOW ANALYSIS.

		24-bus	30-bus	39-bus	2736-bus
Adaptive	***FP***	0.00049%	0.78%	0	0
	FN	0.012%	0.013%	0.012%	0.00048%
DC	***FP***	7.6%	2.6%	6.7%	5.3%
	FN	1.3%	20%	0.3%	1.9%

With the help of the adaptive power flow analysis algorithm, the false negative rate is reduced to 0.01%, on average. The false positive rate is reduced to 0.78% in the worst case (the 30-bus system). For the 2736-bus and 39-bus systems, we do not find any false positives in the conducted experiments.

B. Detection Latency

For each attack, we measure the detection latency when the semantic analysis uses the classical AC power flow analysis (*AC*), the adaptive power flow analysis (*Adaptive*), and the DC power flow analysis (*DC*). Figure 12 presents the average detection latency for all considered attacks with 95% confidence interval (CI). The *x*-axis indicates the four test systems; bars of different colors distinguish the three algorithms. The *y*-axis indicates detection latency. Because the 2736-bus system has a larger scale than the other three systems, its detection latency is at least one order of magnitude longer than those of the small-scale systems. Therefore, we show the detection latency for the 24-bus, 30-bus, and 39-bus systems in a separate figure.

With the help of the adaptive power flow analysis algorithm, the detection latency is reduced by approximately 50% for the 24-bus and 39-bus systems. This reduction rate increases to 66% for the 2736-bus system. The reduction of latency is directly contributed by the reduction of iterations performed in the adaptive algorithm. In our experiments, the classical AC power flow analysis can take up to ten iterations until the solution converges or fifty iterations if the solution does not converge. The proposed adaptive algorithm reduces the calculation to two to four iterations for different perturbations. The smallest reduction occurs for the 30-bus system at merely 16%. With further analysis on the 30-bus system, we find that the classical AC power flow analysis needs only three iterations, on average, to evaluate attack attempts. Hence, the detection latency is already short when using the classical AC algorithm.

The DC power flow analysis uses a single iteration to calculate system state. As a result, it requires much less time for the calculation. For the 2736-bus system, its latency can be one or two orders of magnitude shorter than the latency caused by the classical AC power flow analysis. However, the detection accuracy based on this algorithm can be unacceptable, as shown in Table 3.

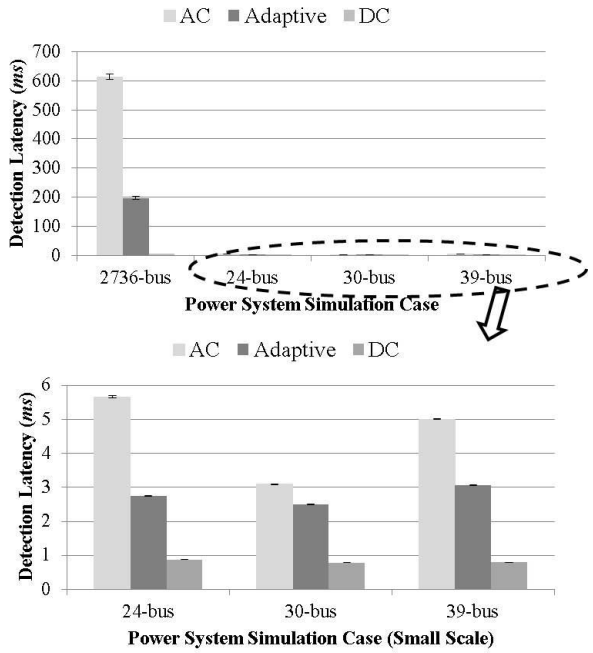


Figure 12. Execution time to calculate systems based on the classical AC, the adaptive, and the DC power flow analysis (with 95% CI, error margins are within 10 *ms* for the large-scale test system and are within 0.01 *ms* for the small-scale test systems, which are not obvious in the figure).

C. Integration of Intrusion Response Mechanism

Based on the analysis in Section V.C, we assume 667 *ms* as the allowed detection latency for the proposed semantic analysis framework to complete detection. For all small-scale test systems, the detection latency based on the classical AC power flow analysis is usually less than 10 *ms*. For the large-scale 2736-bus system, however, 600 *ms* (around 40 clock cycles) is usually needed for the semantic analysis to make detection based on the classical AC power flow analysis algorithm. The safety margin between the detection and the response is only about 67 *ms*, which may not be sufficient in

practice. However, with the help of the adaptive power flow analysis algorithm, we can reduce the calculation time to less than 200 *ms*, which increases the safety margin to around 400 *ms*.

IX. Conclusions

In this paper, we study the impact of control-related attacks and propose a semantic analysis framework to detect such attacks. Network IDSs that are developed based on Bro leverage the proposed adaptive power flow analysis algorithm to perform timely and accurate detection of malicious control commands observed from the vulnerable SCADA network. To demonstrate the usage of the semantic analysis framework, an example intrusion response mechanism that targets malicious commands attempting to open multiple transmission lines is studied.

We study the impact of control-related attacks and evaluate the proposed semantic analysis framework on IEEE 24-bus, 30-bus, and 39-bus systems, and a 2736-bus system. The proposed adaptive power flow analysis algorithm introduces at most a 0.8% false positive rate and a 0.01% false negative rate in our experiments. The semantic analysis can complete the detection in about 200 *ms*, even for the large-scale test system, which makes response to the intrusion practical.

In future work, we will focus on control-theoretic approaches and formal methods to study the control-related attacks in a hybrid system model. We plan to include in the analysis the factors extracted from both cyber and physical infrastructure of a power grid and study their impacts on the grid state.

Acknowledgment

This material is based upon work supported in part by the Department of Energy under award numbers DE-OE0000780 and DE-OE0000097, by the National Science Foundation under award number CNS 13-14891, by the National Security Agency under award number H98230-14-C-0141, and by the IBM Faculty Fellowship.

The authors thank Robin Sommer and the rest of the Bro Project team at International Computer Science Institute for their consult and support to include the DNP3 and Modbus analyzer in Bro's source code repository. The authors also thank Prosper Panumpabi from the University of Illinois at Urbana-Champaign and Donald Borries from Ameren for their insightful discussions.

References

[1] R. Smith. (2014, March). U.S. risks national blackout from small-scale attack. The Wall Street Journal, [Online] available: http://www.wsj.com/articles/SB10001424052702304020104579433670284061220.

[2] J. Kirk. (2014, August). Study finds firmware plagued by poor encryption and backdoors. [Online] available: http://www.itworld.com/security/431186/study-finds-firmware-plagued-poor-encryption-and-backdoors.

[3] K. Zetter. (2013, October). Researchers uncover holes that open power stations to hacking. [Online] available: http://www.wired.com/2013/10/ics/.

[4] D. Goodin. (2014, June). Attackers poison legitimate apps to infect sensitive industrial control systems. [Online] available: http://arstechnica.com/security/2014/06/attackers-poison-legitimate-apps-to-infect-sensitive-industrial-control-systems/.

[5] D. Goodin. (2015, September). Cisco routers in at least 4 countries infected by highly stealthy backdoor. [Online] available: http://arstechnica.com/security/2015/09/attackers-install-highly-stealthy-backdoors-in-cisco-routers/.

[6] N. Falliere, L. Murchu, and E. Chien, "W32.Stuxnet dossier," Symantec Security Response, 2011.

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

[8] S. Cui, Z. Han, S. Kar, T.T. Kim, H.V. Poor, A. Tajer, "Coordinated data-injection attack and detection in the smart grid: A detailed look at enriching detection solutions," in *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 106–115, Sep. 2012.

[9] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 645–658, Oct. 2011.

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

[11] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures," in *Proc. 2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 220–225, 2010.

[12] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *47th Annual Allerton Conference on Communication, Control, and Computing*, Allerton 2009, pp. 911–918, 2009.

[13] D. Hadžiosmanović, R. Sommer, E. Zambon, and P.H. Hartel, "Through the eye of the PLC: Semantic security monitoring for industrial processes," in *Proc. 2014 Annual Computer Security Applications Conference* (*ACSAC)*, pp. 126-135, 2014.

[14] W. Gao, T. Morris, B. Reaves, and D. Richey, "On SCADA control system command and response injection and intrusion detection," in *eCrime Researchers Summit (eCrime), 2010*, pp. 1–9, 2010.

[15] H. Lin, A. Slagell, Z. Kalbarczyk, P.W. Sauer, and R.K. Iyer, "Semantic security analysis of SCADA networks to detect malicious control commands in power grids," in *Proc. 2013 Smart Energy Grid Security workshop (SEGS)*, pp. 29–34, 2013.

[16] K. Stouffer, J. Falco, and K. Kent, "Guide to supervisory control and data acquisition (SCADA) and industrial control systems security," NIST Special Publication, May 2006.

[17] Z. Li, M. Shahidehpour, A. Alabdulwahab, A. Abusorrah, "Bilevel model for analyzing coordinated cyber-physical attacks on power systems," in *IEEE Trans. Smart Grid*, vol. PP, no. 99, pp. 1, Aug. 2015.

[18] J. Yan, Y. Tang, H. He, and Y. Sun, "Cascading failure analysis with DC power flow model and transient stability analysis," in *IEEE Trans. Power Systems*, vol. 30, no. 1, pp. 285–297, Jan. 2015.

[19] Y. Zhang, L. Wang, W. Sun, R.C. Green II, and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smart grids," in *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 796–808, Dec. 2011.

[20] S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber–physical system security for the electric power grid," in *Proc. of the IEEE*, vol. 100, no. 1, pp. 210–224, Jan. 2012.

[21] C.W. Ten, C.C. Liu, and G. Manimaran, "Vulnerability assessment of cybersecurity for SCADA systems," in *IEEE Trans. Power Systems*, vol. 23, no. 4, pp. 1836–1846, Oct. 2008.

[22] C.W. Ten, G. Manimaran, and C.C. Liu, "Cybersecurity for critical infrastructures: Attack and defense modeling," in *IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 40, no. 4, pp. 853–865, July 2010.

[23] C.W. Ten, J. Hong, and C.C. Liu, "Anomaly detection for cybersecurity of the substations," in *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 865–873, Dec. 2011.

[24] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. 2010 IEEE Symposium on Security and Privacy (SP)*, pp. 305–316, May 2010.

[25] H. Lin, A. Slagell, C. Di Martino, Z. Kalbarczyk, and R.K. Iyer, "Adapting Bro into scada: Building a specification-based intrusion detection system for the DNP3 protocol," in *Proc. Cyber Security and Information Intelligence Research Workshop (CSIIRW)*, 2013.

[26] R. Berthier, W.H. Sanders, and H. Khurana, "Intrusion detection for advanced metering infrastructures: Requirements and architectural directions," in *Proc. First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 350–355, 4–6 Oct. 2010.

[27] *IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3)*, IEEE Standard 1815-2010, pp. 1775, July 1 2010.

[28] A. Sharma, Z. Kalbarczyk, J. Barlow, and R.K. Iyer, "Analysis of security data from a large computing organization," in *Proc. 2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)*, pp. 506–517, June 2011.

[29] L. Constantin. (2015, November). Thousands of Java applications vulnerable to nine-month-old remote code execution exploit. [Online] available: http://www.itworld.com/article/3004632/thousands-of-java-applications-vulnerable-to-nine-month-old-remote-code-execution-exploit.html?phint=newt%3Ditworld_today&phint=idg_eid%3D2aeed3d6b75b04f4e1ce1f250495ad1c#tk.ITWNLE_nlt_today_2015-11-12.

[30] A. Monticelli, "Electric power system state estimation," in *Proc. of the IEEE*, vol. 88, no. 2, pp. 262–282, 2000.

[31] M. Prais and A. Bose, "A topology processor that tracks network modifications," *IEEE Trans. Power Systems*, vol. 3, no. 3, pp. 992–998, Aug. 1988.

[32] J.D. Glover, M.S. Sarma, and T.J. Overbye, *Power System Analysis and Design*, 5th ed., Cengage Learning, 2011.

[33] Q. Yang, J. Yang, W. Yu, D. An, N. Zhang, and W. Zhao, "On false data-injection attacks against power system state estimation: Modeling and countermeasures," *IEEE Trans. Parallel and Distributed Systems*, vol. 25, no. 3, pp. 717–729, Mar. 2013.

[34] Z. Qin, Q. Li, and M.-C. Chuah, "Defending against unidentifiable attacks in electric power grids," *IEEE Trans. Parallel and Distributed Systems*, vol. 24, no. 10, pp. 1961–1971, Sep. 2012.

[35] L. Xie, Y. Mo, and B. Sinopoli, "Integrity data attacks in power market operations," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 659–666, Dec. 2011.

[36] F. Pasqualetti, "Secure control systems: A control-theoretic approach to cyber-physical security," Ph.D. dissertation, Dept. Mechanical Engineering, Univ. of California, Santa Barbara, 2012.

[37] J. Valenzuela, J. Wang, and N. Bissinger, "Real-time intrusion detection in power system operations," *IEEE Trans. Power Systems*, vol. 28, no. 2, pp. 1052–1062, May 2013.

[38] C.L. DeMarco, J.V. Sariashkar, and F. Alvarado, "The potential for malicious control in a competitive power systems environment," in *Proc. 1996 IEEE International Conference on Control Applications*, pp. 462–467, Sep. 15–18 1996.

[39] A.O. Ekwue, "A review of automatic contingency selection algorithms for online security analysis," in *1991 International Conference on Power System Monitoring and Control*, pp. 152–155, 1991.

[40] F. Albuyeh, A. Bose, and B. Heath, "Reactive power considerations in automatic contingency selection," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-101, no. 1, pp. 107–112, Jan. 1982.

[41] P. Hines, E. Cotilla-Sanchez, and S. Blumsack, "Do topological models provide good information about electricity infrastructure vulnerability?" *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 20, no. 3, 2010.

[42] R. Albert, I. Albert, and G.L. Nakarado, "Structural vulnerability of the North American power grid," *Physical Review E*, vol. 69, no. 2, Feb. 2004.

[43] S. Zonouz, C.M. Davis, K.R. Davis, R. Berthier, R.B. Bobba, and W.H. Sanders, "SOCCA: A security-oriented cyber-physical contingency analysis in power infrastructures," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 3–13, Jan. 2014.

[44] M. Vaiman, K. Bell, Y. Chen, B. Chowdhury, I. Dobson, P. Hines, M. Papic, S. Miller, and P. Zhang, "Risk assessment of cascading outages: Methodologies and challenges," *IEEE Trans. Power Systems*, vol. 27, no. 2, pp. 631–641, Dec. 2012.

[45] Y. Zhu, J. Yan, Y. Sun, and H. He, "Revealing cascading failure vulnerability in power grids using risk-graph," *IEEE Trans. Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3274–3284, Jan. 2014.

[46] Y. Zhu, J. Yan, Y. Tang, Y. Sun, and H. He, "Resilience analysis of power grids under the sequential attack," *IEEE Trans. Information Forensics and Security*, vol. 9, no. 12, pp. 2340–2354, Oct. 2014.

[47] J. Yan, Y. Zhu, H. He, and Y. Sun, "Multi-contingency cascading analysis of smart grid based on self-organizing map," *IEEE Trans. Information Forensics and Security*, vol. 8, no. 4, pp. 646–656, Feb. 2013.

[48] Y. Zhu, J. Yan, Y. Tang, Y. Sun and H. He, "Joint substation-transmission line vulnerability assessment against the smart grid," *IEEE Trans. Information Forensics and Security*, vol. 10, no. 5, pp. 1010–1024, Jan. 2015.

[49] E. LeMay, W. Unkenholz, D. Parks, C. Muehrcke, K. Keefe, and W.H. Sanders, "Adversary-driven state-based system security evaluation," in *Proc. 6th International Workshop on Security Measurements and Metrics*, 2010.

[50] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for SCADA networks," in *Proc. SCADA Security Scientific Symposium*, pp. 127–134, Jan. 2007.

[51] O. Linda, T. Vollmer, and M. Manic, "Neural network based intrusion detection system for critical infrastructures," in *Proc. International Joint Conference on Neural Networks (IJCNN)*, pp. 1827–1834, June 2009.

[52] J. Hazra and A.K. Sinha, "Identification of catastrophic failures in power system using pattern recognition and fuzzy estimation," *IEEE Trans. Power Systems*, vol. 24, no. 1, pp. 378–387, June 2009.

[53] R.B Bobba, K.M. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T.J. Overbye, "Detecting false data injection attacks on DC state estimation," in *Preprints of the First Workshop on Secure Control Systems, CPSWEEK*, 2010.

[54] R. Mitchell and I. Chen. "Behavior-rule based intrusion detection systems for safety critical smart grid applications," *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1254–1263, 2013.

[55] A. Carcano, I.N. Fovino, M. Masera, and A. Trombetta, "State-based network intrusion detection systems for SCADA protocols: A proof of concept," *Critical Information Infrastructures Security*, *Lecture Notes in Computer Science*, vol. 6027, pp. 138–150, 2010.

[56] IEEE Standard Communication Delivery Time Performance Requirements for Electric Power Substation Automation, IEEE Standard 1646-2004, 2005.

[57] V. Paxson, "Bro: A system for detecting network intruders in real-time," *Computer Networks*, vol. 31, no. 23, pp. 2435–2463, 1999.

[58] The Bro Project. (2014). The Bro Network Security Monitor. [Online]. available: http://bro.org, 2014.

[59] T. Yang, H. Sun, and A. Bose, "Transition to a two-level linear state estimator—Part 1: Architecture," *IEEE Trans. on Power Systems*, vol. 26, no. 1, pp. 46–53, Feb. 2011.

[60] T.S. Sidhu and P.K. Gangadharan, "Control and automation of power system substation using IEC61850 communication," in *Proc. 2005 IEEE Conference on Control Applications (CCA)*, pp. 1331–1336, 2005.

[61] B. Stott, J. Jardim, and O. Alsac, "DC power flow revisited," *IEEE Trans. Power Systems*, vol. 24, no. 3, pp. 1290–1300, 2009.

[62] Schweitzer Engineering Laboratories, Inc., SEL-421-4,-5 Relay Protection and Automation System, Instruction Manual, June 27th, 2013.

[63] Open DNP3 Group. (2012). DNP3—Distributed Network Protocol 3.0–Google project hosting. [Online]. available: http://code.google.com/p/dnp3/

[64] B. Qiu, Y. Liu, E.K. Chan, and L.L.J. Cao, "LAN-based control for load shedding," *Computer Applications in Power, IEEE*, vol. 14, no .3, pp. 38–43, July 2001.

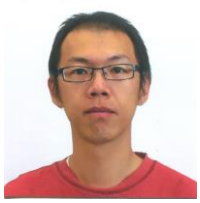
[65] ABB (2014). Grid Automation Controller COM600 Product Guide. [online] available: http://www05.abb.com/global/scot/scot229.nsf/veritydisplay/04a937b04ea99cb3c1257aad0031463f/$file/COM600_pg_756764_ENe.pdf

[66] R.D. Zimmerman, C.E. Murillo-Sánchez, and R.J. Thomas, "MATPOWER: Steady-state operations, planning and analysis tools for power systems research and education," *IEEE Trans. Power Systems*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[67] Midwest Independent Transmission System Operator, Inc. Monthly Market Assessment Report, June 2012.

[68] S. Zonouz, K.M. Rogers, R. Berthier, R.B. Bobba, W.H. Sanders, and T.J. Overbye, "SCPSE: Security-oriented cyber-physical state estimation for power grid critical infrastructures," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 1790–1799, Dec. 2012.

[69] B.C. Lesieutre, A. Pinar, and S. Roy, "Power system extreme event detection: The vulnerability frontier," in *Proc. 41st Annual Hawaii International Conference on System Sciences*, Jan. 2008.

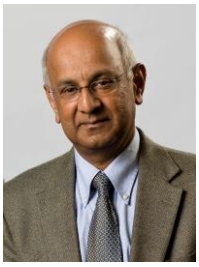


Hui Lin received his B.S. degree from Huazhong University of Science and Technology in 2006 and his M.S. degree from the University of Illinois at Chicago in 2010, both in electrical and computer engineering. He is currently working toward his Ph.D. degree at the University of Illinois at Urbana-Champaign. His research interests include cybersecurity and intrusion detection in power systems.

Adam Slagell received an M.S. degree in computer science from the University of Illinois at Urbana-Campaign in 2003, a master's degree in mathematics from Northern Illinois University (NIU) in 2000, and a B.S. degree in mathematics from NIU in 1999. Currently, he is a Director of the Cyber Security Division and the Chief Information Security Officer at the National Center for Supercomputing Applications and Co-PI for the NSF Bro Center, which brings its network security monitoring expertise and support to NSF cyber infrastructure and projects.

Zbigniew T. Kalbarczyk is a Research Professor at the Coordinated Science Laboratory of the University of Illinois at Urbana-Champaign. Dr. Kalbarczyk's research interests are in the area of design and validation of reliable and secure computing systems. His current work explores emerging technologies, such as resource virtualization to provide redundancy and assure system resiliency to accidental errors and malicious attacks. His research involves also analysis of data on failures and security attacks in large computing systems, and development of techniques for automated validation and benchmarking of dependable and secure computing systems using formal (e.g., model checking) and experimental methods (e.g., fault/attack injection). He served as a program Chair of International Conference on Dependable Systems and Networks (DSN) 2002 and 2007. He is an Associate Editor of IEEE Transactions on Dependable and Secure Computing. Dr. Kalbarczyk has published over 130 technical papers and is regularly invited to give tutorials and lectures on issues related to design and assessment of complex computing systems. He is a member of the IEEE, the IEEE Computer Society, and IFIP Working Group 10.4 on Dependable Computing and Fault Tolerance.

Peter W. Sauer (S 73, M 77, SM 82, F 93, LF 12) obtained his Bachelor of Science degree in Electrical Engineering from the University of Missouri at Rolla in 1969, the Master of Science and Ph.D. degrees in Electrical Engineering from Purdue University in 1974 and 1977, respectively. From 1969 to 1973, he was the electrical engineer on a design assistance team for the Tactical Air Command at Langley Air Force Base, Virginia. From August 1991 to August 1992, he served as the Program Director for Power Systems in the Electrical and Communication Systems Division of the National Science Foundation in Washington, D.C. He is a cofounder of the Power Systems Engineering Research Center (PSERC) and the PowerWorld Corporation. He is a registered Professional Engineer in Virginia and Illinois, a Fellow of the IEEE, and a member of the U.S. National Academy of Engineering. He is currently the Grainger Chair Professor of Electrical Engineering at Illinois.



Ravishankar K. Iyer is the George and Ann Fisher Distinguished Professor of Engineering at the University of Illinois at Urbana-Champaign. He holds appointments in the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory (CSL) and the Department of Computer Science, he serves as Chief Scientist of the Information Trust Institute and is affiliate faculty of the National Center for Supercomputing Applications (NCSA). He currently is co-leads the CompGen Center at Illinois. Professor Iyer is a Fellow of the American Association for the Advancement of Science, the IEEE, and the ACM. He has received several awards, including the AIAA (American Institute for Aeronautics and Astronautics) Information Systems Award, the IEEE Emanuel R. Piore Award and the 2011 Outstanding Contributions award by the Association of Computing Machinery - Special Interest Group on Security. Professor Iyer is also the recipient of the degree of Doctor Honaris Causa from Toulouse Sabatier University in France.