RAINCOAT: RAndomization of Network Communication in Power Grid Cyber INfrastructure to Mislead Attackers

Hui Lin, *Member*, IEEE, Zbigniew Kalbarczyk, *Member*, IEEE, and Ravishankar K. Iyer, *Fellow*, IEEE

Abstract— **Though attackers aim to introduce different physical perturbations on power grids, they need to rely on periodic data acquisitions performed by control centers to estimate the physical state of the grid and thus to prepare for destructive activities. In this paper, we present Raincoat, which randomizes data acquisitions to disrupt and mislead attackers' preparations. We transform one data acquisition into multiple rounds. In each round, we dynamically manipulate network flows in the control networks so that randomly selected "online" devices respond with real measurements. Meanwhile, we intelligently spoof measurements for other "offline" devices to mislead attackers into designing ineffective strategies. Based on experiments using large-scale power systems and six real wide area networks, Raincoat is effective against false data injection and control-related attacks with small overhead. The probability of successful attacks can be reduced from 70% to 1%; attacks introduce little damage even if they are executed. Network latency of data acquisition increases on average by less than 6%.**

Index Terms—**moving target defense, decoy attacks, SCADA, software-defined networking**

I. Introduction

CYBER-attacks on SCADA (supervisory control and data acquisition) systems used by industrial control systems (ICSes), e.g., power grids, can cause severe damage. In December 2015, remote intruders penetrated a Ukrainian power grid and caused a blackout that affected 225,000 residents [1]. To prepare and launch attacks that cause physical damage, attackers can rely on SCADA applications used in ICSes, in which two primary functions are *data acquisition* and *control*. Based on real attack incidents [1][2], we can classify adversaries' behavior into three stages, as shown in Figure 1. In the "penetration" stage, attackers establish footholds in SCADA communication networks, e.g., in human-machine interfaces (HMI) or remote terminal units (RTU) in a power system's control network, as shown in Figure 2. After they have obtained accesses to the SCADA networks, in the "preparation" stage, attackers can use information from communication networks, e.g., taken via periodic data acquisitions, to study the physical measurements of the power grids and determine effective attack strategies, e.g., malicious control operations that can cause physical damage. Finally, in the "execution" stage, attackers execute the strategies on ICS devices, e.g., intelligent electronic devices, sensors, or breakers (shown in Figure 2), by injecting or modifying the control operations used by system administrators.

To detect attacks in SCADA systems, previous work has focused on in-depth analysis of network and system activities when attackers execute their strategies. These efforts include using (i) anomalies in communication patterns, (ii) the physical impact predicted by state estimation algorithms (e.g., whether execution of a given command could cause an overload of a transmission line), and (iii) inconsistencies in compromised measurements [3][4][5]. These approaches are effective against specific malicious activities. However, once attacks evolve and use different execution channels based on new vulnerabilities, the detection methods can become less effective. Also, detecting attacks during their execution can leave system administrators little time to prevent damage from happening (by either delaying or reversing malicious activities) [3]. As shown in [1], it took attackers a few hours to perform the malicious operations, but the big concern is that "the strongest capability of the attackers is their capability to perform long-term reconnaissance operations required to learn the environment," which last around six months.

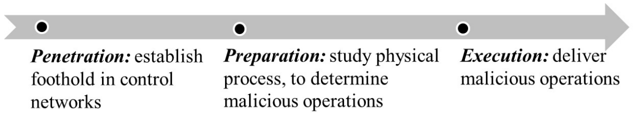

Penetration: establish foothold in control networks
Preparation: study physical process, to determine malicious operations
Execution: deliver malicious operations

Figure 1. Three stages of attacks that introduce physical damage.

Instead of focusing on the execution stage, we detect attacks in their "preparation" stage. Towards this goal, we obfuscate SCADA data acquisitions based on which attackers develop their strategies to cause physical damage. Detecting attacks during their preparations brings two major benefits that are difficult to achieve at attack execution stage (e.g., when attackers execute malicious commands). *First*, we can cover a wide spectrum of attacks including unknown ones. Though attackers may adopt diverse execution activities to cause different types of physical damage, they need to rely on measurement data to estimate the current physical state of the target system, based on which they plan effective attack strategies. *Second*, we can detect and mislead attacks before adversaries execute their strategies and inflict damage. Detecting attacks at this early stage makes it possible to remove potential threats and prevent damage.

To obfuscate attackers' knowledge, we propose Raincoat, a technique that randomizes data acquisitions in power systems. Raincoat aims to expose and mislead attackers while they are preparing attack strategies. At runtime, Raincoat transforms a single data acquisition operation issued by the control center to all substation devices into multiple rounds of data delivery. In each round, we enable network connectivity to a subset of randomly selected devices, i.e., online devices. We allow the

Manuscript received Jan 9, 2018. This work was supported in part by the Department of Energy and the National Science Foundation.

H. Lin is with the Computer Science and Engineering Department at the University of Nevada at Reno, Reno, IL, 89557 (email: hlin2@unr.edu). Z. Kalbarczyk and R. K. Iyer are with the Electrical and Computer Engineering Department, University of Illinois, Urbana, IL, 61801 (e-mail: kalbarcz@illinois.edu and rkiyer@illinois.edu).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2018.2870362, IEEE Transactions on Smart Grid

network traffic to reach online devices, which send real measurements upon receiving data acquisition requests. Meanwhile, we disallow traffic to offline devices (i.e., a subset of devices for which we disable network connectivity) but intelligently spoof the measurements for these devices.

Current moving target defense (MTD) mechanisms for SCADA systems can introduce disruptions in power systems that are visible to both system operators and attackers [27][28]. These MTD methods can affect grid's control operations, e.g., change the susceptance of transmission lines [17][24][36][37] or reduce the accuracy of state estimation [26]. However, Raincoat only disrupts attackers' knowledge without affecting power system's physical operations applying to field sites. The control center continuously collects the complete set of real measurements with the same rate and the accuracy of normal operations, such as state estimation and contingency analysis, remains unchanged. Only attackers' observations are limited because they are unable to distinguish the real and spoofed measurements.

So that Raincoat can further deter attackers' ability to compromise systems, we design an algorithm that includes decoy values in the spoofed measurements of offline devices. Consequently, when mixed with real measurements, the spoofed measurements present attackers with a valid power system state that is different from the real state. Using this crafted state information, attackers will always end up with ineffective attack strategies that expose the malicious activities but cause little or no harm to the real power system.

Even though we can implement Raincoat by any network manipulation techniques, we take advantage of network management and programming paradigm by implementing and deploying software-defined networking (SDN) in switches at the edge of power systems' communication infrastructure [12][29][30]. SDN can manipulate network flows related to data acquisition in SCADA systems while making little change to data acquisition procedure performed at control centers or the configurations of substation devices.

Specifically, Raincoat makes the following contributions:

1) ***Disrupts attacks at the preparation stage.*** To the best of our knowledge, Raincoat is the first technique to disrupt and mislead attackers as they prepare malicious activities in power systems. The randomized data acquisition exposes the attackers when they attempt to access offline devices, even before they carry out any destructive activities.
2) ***Mitigates damage by misleading attackers.*** We include in Raincoat an algorithm to generate decoy measurements for offline devices. The decoy measurements mislead attackers into designing ineffective attack strategies of both false data injection attacks and control related attacks. These strategies fail to introduce physical damage to power systems even if attackers execute them.
3) ***Has little overhead on control networks.*** We construct a cyber-physical testbed to evaluate performance overhead of Raincoat. In the testbed, we used MATPOWER to simulate power systems of more than 1000 buses, which provide measurements to communication networks. To mimic the communication, we used the GENI testbed to construct six real wide area networks (WAN), including one consisting of more than 100 nodes distributed at different geographical locations [7]. We implemented Raincoat as SDN controllers in ONOS, an open-source network operating system [6]. Compared to the default SDN controller, Raincoat introduces on average less than 6% additional latency in data acquisition.

II. Background & Threat Model

A. *Threat model*

In Figure 1, we show a hierarchical communication architecture used by SCADA systems in the context of an electric power grid, where IP-based networks provide connectivity between a control center and substation devices.

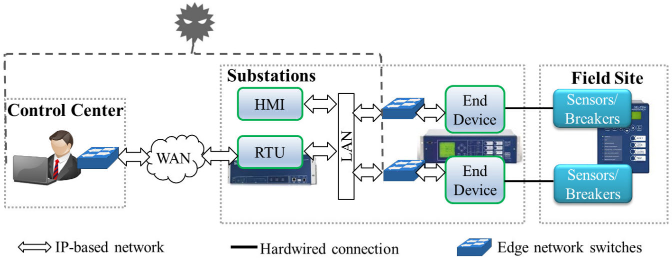


Figure 2. Control network setup in power systems.

Definition 1 (end device). Intelligent electronic devices (IED) in substations located at the edge of the communication path based on IP-based networks connecting the control center and substations.

By definition 1, end devices connect to sensors or circuit breakers through hardwired connections in their downstream communication. In their upstream communication, multiple end devices connect to an up-level IED, e.g., RTU (remote terminal unit), which forwards information (e.g., aggregate measurements or commands) to/from the control center. ■

Definition 2 (edge switch). Network switches located as the first or the last hop of communication paths that connect the control center and end devices. ■

In this work, we consider the remote insider threat model. We assume that attackers can bypass the barrier (or perimeter protection) between corporate networks and control networks and establish footholds in the control network connecting control center and end devices.

By using Figure 1, we present the assumptions of the threat model used in this paper.

- In *control networks*, we assume that attackers can penetrate computing devices on any communication path that connects the control center and end devices. Those computing devices, such as human-machine interfaces (HMI) or RTUs, are often installed at substations located over a large geographical area. Because it is challenging to maintain computing devices in a wide area, we often find unpatched vulnerabilities in those devices, e.g., an old TCP vulnerability found in substation devices [8]. In most well-known attacks targeting industrial control systems, e.g., Stuxnet and attacks against Ukrainian power system [1][2], attackers targeted vulnerable computing devices and used them as footholds to prepare and execute attacks.
- In *substations*, we assume that attackers do not have physical

access to the devices on the downstream of end devices. Under this assumption, attackers need to use computing devices on the upstream of end devices, including control networks and the control center, to monitor system state and launch malicious operations. We argue that this is a reasonable assumption as substations are located in large geographical areas, and simultaneous physical breaking into those substations is not practical.

- In the *control center*, we trust the integrity of the state estimation software or data historians used in the control center. Under this assumption, attackers are not able to observe measurements or physical configurations of the target power systems by compromising the state estimation software or data historians; they need to penetrate into the control networks or other devices in the control center to learn the system state. We argue that this is a reasonable assumption, as it is practical to protect those critical software components by running them in a dedicated or a separate machine or by using attestation mechanisms, e.g., TPM (trusted platform module), to verify their integrity periodically. Also, we can use much existing research work that protects power systems against false data injection attacks to ensure the integrity of state estimation software [4].

  We assume that attackers can remotely penetrate the local area network environment of the control center. For example, by exploiting the vulnerabilities in workstations and employee's devices, e.g., laptops, smartphones, or USB drives, that are connected to control centers [2], attackers can further obtain the privileges necessary to install malware, sniff, inject, and even modify measurements delivered to state estimation software and commands issued to substations. Consequently, even if we can trust the state estimation software, attackers can still monitor system state and launch malicious commands in the control center.

- We trust the functionality of *Raincoat*. Raincoat uses edge switches and SDN controllers to randomize data acquisitions in SCADA systems; we trust SDN controllers and the edge switches. This is a reasonable assumption for two reasons. First, maintaining the integrity of SDN controllers and edge switches (e.g., being patched or upgraded) can be more practical to achieve, compared to computing devices in substations and control networks. Second, trusting edge switches can reduce the range of trusted computing base, as compared to trusting all computing devices in the control networks. Also, SDN controllers connect to switches through ports different from the data ports that are used to exchange information. Consequently, it is not practical for attackers to penetrate SDN controllers and thus to monitor network traffic between the controllers and the edge switches. Even if attackers compromised edge switches and SDN controllers, they would obtain the same privilege as if they penetrated control networks devices (e.g., RTUs or HMIs).

Under those assumptions, attackers can perform both *active* and *passive* monitoring from the compromised devices to collect system information, based on which they develop attack strategies. The active monitoring relies on attackers' ability to issue a valid and even authenticated request to end devices to retrieve measurements. To be stealthy (i.e., hide their presence), attackers can passively monitor the measurements when they go through the compromised devices, even if the measurements are transmitted between devices in encrypted traffic. Under our threat model, with information gained via passive and active monitoring, attackers can compromise measurement data, i.e., in false data injection attacks, and compromise commands, i.e., in control-related attacks, when measurements or commands go through the penetrated devices.

B. Learning the physical state

In this section, we discuss the physical measurements and system state information that attackers need in order to launch two types of attacks against power systems: (i) *false data injection attacks* (FDIA), whereby attackers compromise measurements sent to the control center, and (ii) *control-related attacks* (CRA), whereby attackers compromise commands sent to end devices at substations to change the system state and cause physical damage. Recent high-profile attacks against power grid infrastructures fall into these two broad categories [1][3][9][10].

A power system is composed of buses (representing substations) that are connected by transmission lines. The state of the system is specified by the voltage magnitude and the angle for each bus, i.e., (V_j, θ_j) in equations (1)–(4). For each bus j, two power flow equations, i.e., equations (1) and (2), are formulated based on the fact that the generated power (P_j^G and Q_j^G), the consumed power (P_j^L and Q_j^L), and the power delivered to other buses (indexed by k) are balanced at each timestamp [11]. In addition, we can formulate equations (3) and (4) to describe the power flow corresponding to each transmission line (P_{jk} and Q_{jk}).

$$P_j^G - P_j^L = \sum_k V_j V_k \big(G_{jk}\cos(\theta_j - \theta_k) + B_{jk}\sin(\theta_j - \theta_k)\big) \tag{1}$$

$$Q_j^G - Q_j^L = \sum_k V_j V_k (G_{jk}\sin(\theta_j - \theta_k) - B_{jk}\cos(\theta_j - \theta_k)) \tag{2}$$

$$P_{jk} = -V_j^2 G_{jk} + V_j V_k G_{jk}\, cos(\theta_j - \theta_k) + V_j V_k B_{jk}\sin(\theta_j - \theta_k) \tag{3}$$

$$Q_{jk} = V_j^2 B_{jk} + V_j V_k G_{jk}\, sin(\theta_j - \theta_k) - V_j V_k B_{jk}\, cos(\theta_j - \theta_k) \tag{4}$$

In Table 1, we list the *targets* of the two attack types (FDIA and CRA) along with the measurements that attackers need in order to design effective attack strategies. Attackers use existing state estimation algorithms to analyze the collected measurements (e.g., the *measurements for preparations* in Table 1) and determine the attack strategy, e.g., to change the measurements seen by the control center or to issue a command to open a transmission line [11]. Note that there are two classes of state estimation approaches for solving equations (1)–(4): *AC state estimation*, which employs iterative algorithms; and *DC state estimation*, which solves the linear approximations of the equations.

TABLE 1: TARGETS AND PREPARATIONS OF FDIA AND CRA.

Type	***Measurements for Preparations***	***Target***
FDIA	B_{jk}, susceptance of all transmission lines	P_j^G and P_j^L of all substations; P_{jk} of all transmission lines
CRA	P_j^G, Q_j^G, P_j^L, Q_j^L of all substations; P_{jk}, Q_{jk} of all transmission lines	Control commands that can disconnect transmission lines or substations in a power grid

III. Raincoat Approach

Raincoat objective. Our design is to obfuscate the cyber-physical infrastructure to prevent attackers from obtaining correct measurements of the system state, i.e., the measurements used for preparations as listed in Table 1, by (i) randomizing the data acquisition procedure and (ii) spoofing measurements.

High-level procedure. Traditional SCADA systems perform data acquisitions by issuing requests from the control center to all end devices in substations periodically. According to IEEE standard 1646 [13], the period, represented by T, ranges from 1 to 10 seconds.

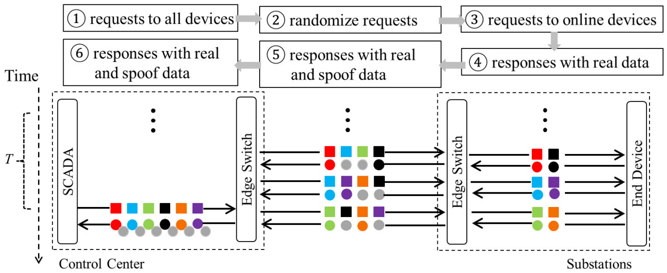

① requests to all devices
② randomize requests
③ requests to online devices
⑥ responses with real and spoof data
⑤ responses with real and spoof data
④ responses with real data
Time
T
SCADA
Edge Switch
Edge Switch
End Device
Control Center
Substations

Figure 3. Raincoat approach.

We illustrate the Raincoat approach in Figure 3, which consists of the following steps:

- Step ①: Raincoat does not change the procedure that the control center uses to issue data acquisition requests. The control center still issues the requests to all devices as in the case without Raincoat. In the example in Figure 3, there are 6 end devices as the destinations of the requests. We use square boxes of different colors to represent requests destined to different devices.
- Step ②: when data acquisition requests reach the edge switch at the control center, we divide them into multiple rounds of data acquisition requests. In each round, we randomly select a subset of end devices and only issue the requests to these devices. In the example shown in Figure 3, we divide data acquisition requests into three rounds. In each round, we send requests to 4 randomly selected devices, out of the total of 6 end devices.
- Step ③: When each round of data acquisition requests reach the edge switch at substations, we again randomly specify two sets of devices. The first set consists of online devices, whose network connectivity is enabled and which are allowed to respond with real measurements. All other devices not included in this set are offline devices, whose network connectivity is disabled. In the example shown in Figure 3, we randomly select 2 online devices in each round of data acquisition.
- Step ④ : Raincoat does not change the physical configurations of end devices; each online device responds with the collected measurement as in the case that Raincoat is not used (represented by colored circles).
- Step ⑤: We use the edge switch at substations to forward the responses of real measurement data from online devices to the control center. Meanwhile, we spoof the responses on behalf of offline devices and use the edge switch to send them to the control center (represented by grey circles in the figure). Under our threat model, attackers, who can compromise any device in the control networks, are not able to distinguish between online and offline devices.
- Step ⑥: We use the edge switch at the control center to combine responses from multiple rounds of data acquisitions. Consequently, the state estimation software in the SCADA systems receives both real and spoofed measurements. With the knowledge of randomized device connectivity, the state estimation software can distinguish real and spoofed measurements, and thus still collect real measurements from all devices with the same period as in the case without Raincoat.

When using Raincoat, we acquire the measurements in k rounds so that the original data acquisition interval T is divided into k rounds, each of which has a duration p, where $k \cdot p \leq T$. In each round, Raincoat collects real measurements from different devices, but not from all devices. Consequently, the interval that it takes each individual device to respond with real measurements is still on the order of T, not p. According to [13], T ranges from 1 to 10 seconds, and the communication latency observed in wide area networks should be less than 100 milliseconds. Consequently, for data acquisitions from SCADA systems, which are the focus of this paper, the control center will have sufficient time to collect measurements from each device.

When using Raincoat, we can collect all measurements with the same accuracy and within a predefined time T. Even though each device responds with measurements at slightly different times within the data acquisition interval, Raincoat introduces little impact on operations that rely on SCADA measurements, e.g., state estimation, contingency analysis, and optimal power flow analysis (see Section V.B for evaluations).

By using Raincoat, we can disrupt both *active* and *passive* monitoring performed by attackers.

- ***Disrupting active monitoring.*** Under our threat model, any attempt by attackers to access offline devices while they monitor the target system, e.g., by scanning end devices or directly sending data acquisition requests to specific end devices, exposes the attackers' presence and results in raising an alert to the system operator.
- ***Disrupting passive monitoring.*** Raincoat includes two designs to disrupt passive monitoring. *First*, Raincoat has two rounds of randomization of data acquisitions: Step ② randomly selects a subset of end devices to which the data acquisition requests are sent, and Step ③ randomly selects online devices, which receive requests and respond with measurements. In other words, Raincoat forwards data acquisition requests (via the edge switch at the control center) not only to online devices but also to randomly selected offline devices, to prevent attackers from learning device connectivity by monitoring data acquisition requests. *Second*, Raincoat constructs spoofed measurements for offline devices to disrupt attackers' passive monitoring, as attackers cannot distinguish between online and offline devices.

Furthermore, under our threat model, attackers cannot compromise edge switches. As a result, they are unable to distinguish between real and spoofed measurements by passively monitoring (i) network packets between the edge switches in substations and end devices, or (ii) network packets between the edge switches and the corresponding SDN controllers. In addition, we present in Section IV a method to include decoy measurements in the spoofed responses to further prevent attackers from designing strategies that can cause physical damage.

Implementation. Raincoat achieves randomized data acquisition by manipulating the network flow on edge switches. To manipulate the network flows in edge switches, we can use SDN technology. Note that SDN-enabled switches are being designed and deployed for power system substations, making it practical to deploy Raincoat [12].

When the control center issues data acquisition requests, the edge switches in the substations are at the last hop of the communication; they filter in requests destined for online devices and redirect the requests destined for offline devices to SDN controllers (Step ③ in Figure 3). In responding to data acquisitions, the edge switches in the substations are at the first hop of the communications. We use SDN controllers to forward responses from online devices and craft spoofed responses and send them through these edge switches (Step ④ in Figure 3).

In using SDN controllers to randomize data acquisition, Raincoat does not make any changes to: (i) data acquisition procedure in the control center, (ii) physical configurations of end devices in substations, and (iii) existing network routing/forwarding configurations in the control networks.

IV. Craft Decoy Measurements to Mislead Attackers

For Raincoat, we can have multiple options on what to include in spoofed measurements. One option is to include random values in the spoofed measurements. Attackers, without the knowledge of device connectivity, cannot distinguish between real and spoofed random measurements. However, while the random measurements can hide the real measurements, they do not directly follow the physical model of power systems; they can easily attract attackers' suspicion.

In addition to hiding real measurements, we propose a method to include decoy values in the spoofed measurements, which further mislead attackers into designing ineffective strategies. To craft decoy measurements, we simulate a power system that has the same topology as the transmission network of the real power system under protection. We can implement that power system simulation in any simulation tool, e.g., MATPOWER which is what we use [14], and run it at the SDN controller, which is responsible for sending spoofed measurements on behalf of offline devices.

We craft decoy measurements in two steps to achieve two corresponding objectives:

- *Step 1*. We can set different initial values based on the attacks that we want to mislead. In this paper, we mislead FDIAs and CRAs, by crafting the measurements used for preparations listed in Table 1 such that the compromised "target" measurements fail to introduce physical damage. Specifically, we determine the susceptance of transmission lines in the simulated power systems, based on which the designed FDIAs become detectable in real power systems (in Step 1.a). We determine the power flows of the transmission lines in simulated power systems to mislead CRAs (in Step 1.b) into targeting non-critical devices in real power systems.
- *Step 2*. We further refine the values such that the decoy measurements follow physical models. Specifically, we iteratively put decoy measurements in the state estimation algorithm and use the obtained calculation results to adjust the measurements determined in Step 1, until the decoy measurements can observe the bad-data detection criteria used in the state estimation.

Note that the values of decoy measurements vary according to the types of attacks that we are trying to mislead. In this paper, we use the FDIAs and CRAs as two examples, because they are commonly found in the research literature and real incidents [1][9][10]. However, the proposed algorithm to craft decoy measurements is not restricted by the types of attacks, as long as their preparations rely on measurements observed in communication networks. For example, to mislead FDIAs based on the AC power flow model, we can craft decoy measurements based on the attack procedures discussed in [10]. We will leave this for future work.

A. Step 1.a: mislead FDIAs

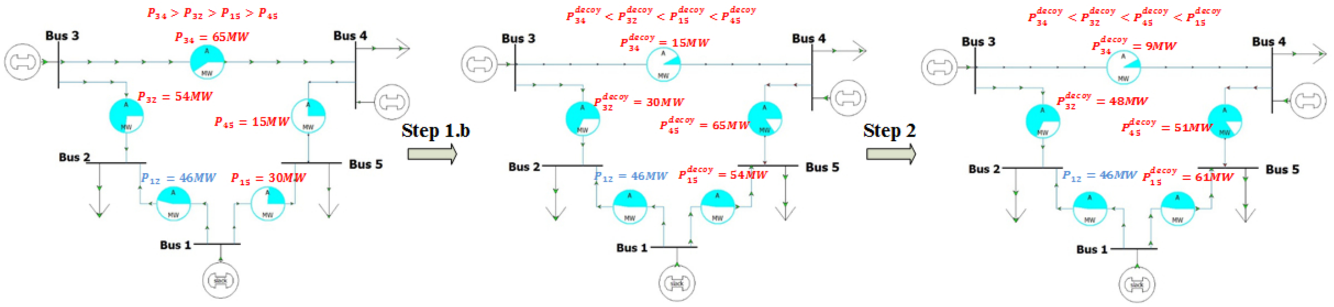
Background. We use the DC power flow model to discuss misleading FDIAs. In this model, we relate state variables, i.e., θ_j in equations (1)–(4), and active power measurements at substations and transmission lines, i.e., P_j^G, P_j^L, and P_{jk}, by using linear approximations of equations (1) and (3):

$$z = Hx + e, \tag{6}$$

where $z = (z_1, z_2, \dots, z_q)$ represents q measurements. For measurements of power flow along transmission lines, we have $z_l = -P_{jk}$; for measurements on substation j, we have $z_l = P_j^G - P_j^L$; $x = (\theta_1, \theta_2, \dots, \theta_p)$, which represents p physical state or p phasor angles at all buses; and $e = (e_1, e_2, \dots e_q)$ is the collection of q measurement errors. $H = (h_{ik})_{q\times p}$ is a q-by-p Jacobian matrix; all entries in H are determined based on entries B_{jk}, the susceptance of transmission lines connecting bus j and bus k (the detailed derivation of H can be found in [9]).

When measurement errors follow a normal distribution with zero mean, the estimation of state variable $\hat{x}$ can be obtained through statistical criteria, e.g., the weighted least-square criterion. When estimating $\hat{x}$, the state estimation further detects and removes bad data or measurements to ensure that the estimated state variable comes "closer" to that of the actual state. The state estimation uses an L2-norm of the measurement residual, i.e., $\|z - H\hat{x}\|$, to detect the presence of bad measurements. If the residual is larger than a threshold, i.e., $\|z - H\hat{x}\| > \tau$, we declare the presence of bad measurements.

In false data injection attacks, attackers maliciously compromise the measurements so that the estimated state variable $\hat{x}$ differs from the actual state without triggering alerts from bad-data detectors [9][10]. If attackers intend to make the state estimation describe a system state that is different from the



(a) (b) (c)

Figure 4. Procedure to craft decoy measurements in a 5-bus power system.

real one, i.e., $\hat{x}_a = \hat{x} + c$, they can inject an attack vector a into the original measurement z. With the full knowledge of H, attackers can construct a such that $a = Hc$ and make the corresponding compromised measurements, i.e., $z_a = z + a$. In that case, the L2-norm of the measurement residual becomes:

$$\|z_a - H\hat{x}_a\| = \|z + a - H(\hat{x} + c)\| = \|z - H\hat{x}\| \leq \tau.$$

Consequently, based on the L2-norm of the measurement residual of z_a, the state estimation cannot detect that the measurements are compromised.

Preparations. As illustrated by [9][10], to conduct successful FDIAs, attackers must obtain the measurement matrix H of a power grid, which includes power systems' topological aspects, e.g., the susceptance of transmission lines. In this paper, we consider FDIAs based on the DC power flow model presented in [9], which relies on complete system topological information to prepare attacks. Based on [9][10], variations of FDIAs have emerged, e.g., [38][39], that rely on partial topological information to prepare attacks. We believe that Raincoat can mislead those FDIAs by adjusting the design presented in this section based on the varied attack procedures.

In our threat model, we assume that attackers rely on measurements exchanged over communication networks to prepare effective FDIAs. However, today's SCADA systems do not use communication networks to deliver power systems' topological information, e.g., the susceptance of transmission lines, periodically. Many research efforts have shown that in order to prepare attacks, attackers can indirectly estimate the topological information based on measurements that can be observed in networks, e.g., voltage magnitudes, power injections at the two ends of the transmission lines [15], and the historical profile of those measurements [39][40].

Protection. Raincoat can use device connectivity to limit the number of measurements that attackers can compromise and mislead attackers into using a decoy Jacobian matrix H'. Specifically, we determine the decoy values of the susceptance of transmission lines in H' such that it becomes challenging for the attack strategy based on H' to bypass the state estimation based on H.

We represent the decoy Jacobian matrix H' as $H' = H + \Delta H$. Based on their knowledge of H', attackers choose injected measurements $a = H'c = (H + \Delta H)c$, where c is decided by the attackers and is unknown to us. For attackers to bypass the state estimation of the protected power system, the following condition needs to be satisfied: $a = H'c = (H + \Delta H)c = Hc'$, where c' is a nonzero p-by-1 vector. Here, we analyze two cases, i.e., $c = c'$ or $c \neq c'$. In each case, we determine the condition of H' that make FDIAs challenging to be successful and to bypass state estimation.

Case 1: $c = c'$. To make $(H + \Delta H)c \neq Hc$ is equivalent to make $\Delta Hc \neq \mathbf{0}$ for all nonzero vectors c. Consequently, we have the following lemma.

Lemma 1. To disrupt the attack strategy for FDIAs (reflected in c), we determine ΔH such that equation $\Delta Hy = 0$ has a unique solution $y = 0$.

Proof. Because $\Delta Hy = 0$ has a unique solution, $y = 0$, then when $y \neq 0$, $\Delta Hy \neq 0$.

When $p \leq q$, $\Delta Hy = 0$ has a unique solution if and only if $rank(\Delta H) = n - 1$, where n is the number of buses. When attackers target a different state, i.e., $\hat{x}_a = \hat{x} + c$ with $c \neq 0$, they will try to determine the corresponding attack vector a to satisfy the condition $a = H'c$, to bypass the bad data detection. As we represent the decoy Jacobian matrix as $H' = H + \Delta H$, the condition $a = H'c$ becomes $a = (H + \Delta H)c = Hc + \Delta Hc$. Because $\Delta Hc \neq 0$, we always have $a \neq Hc$.

When $p > q$, ΔHy always has more than one nontrivial solution. This case should not happen in real power systems, as there aren't enough measurements available in real power systems to solve state estimation, even without attacks. ∎

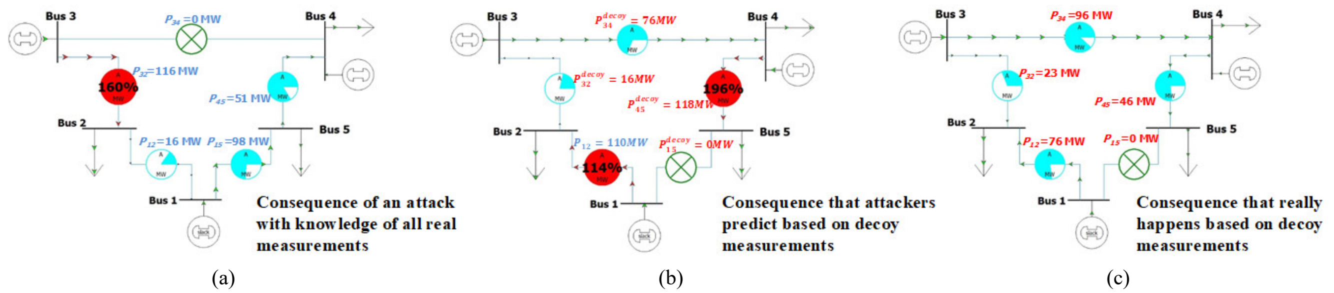
Case 2: $c \neq c'$. By satisfying Lemma 1, we guarantee that attackers' original strategy does not succeed. However, this does not ensure that attackers' activities are always detected. If $a = Hc'$ with $c \neq c'$, the corrupted measurements can still bypass the bad data detector in the state estimation, even though attackers fail to make state estimation estimates a malicious state that they intended to make (as $c \neq c'$).

In Lemma 2, we present the condition when the compromised measurements based on the decoy Jacobian matrix can bypass the bad-data detectors used in the real power system.

Lemma 2. An attack strategy based on the decoy Jacobian matrix, i.e., $a = H'c$, can bypass the bad-data detection if it satisfies the condition $rank(H) = rank([H\ H'c])$.

Proof. If $rank(H) = rank([H\ H'c])$ and $a = H'c$, the linear equation $Hy = a$ is consistent (i.e., the equation has at least one solution). The solution of this equation is the change of system state c' that can bypass the state estimation. ∎

If attackers use the decoy Jacobian matrix, they need to ensure that compromised measurements fall into the column



(a) (b) (c)

Figure 5. Decoy measurements misleading attackers.

space of H, to avoid detection. Even though we cannot find a sound proof to ensure that the condition $rank(H) = rank([H\ H'c])$ always fails, this condition can be challenging to satisfy in practice. In our experiments, we have not found any FDIAs that can bypass the bad-data detectors (see Section V.A for details).

Combining the analyses from those two cases, we formulate the following procedure for Step 1.a, which aims to satisfy Lemma 1.

The procedure of Step 1.a. We determine the decoy values of the susceptance of the transmission lines by randomly changing the corresponding line susceptance of the protected power systems, such that $rank(\Delta H) = n - 1$, where n is the number of buses.

To prepare FDIAs, attackers will need to use SCADA measurements, e.g., the power flow of transmission lines, to indirectly estimate the topological information. To make attackers use the decoy Jacobian matrix, we use the decoy values for line susceptance, as determined in this section, to calculate other SCADA measurements that can be observed over communication networks. In other words, when attackers observe those measurements, they will prepare compromised measurements based on the decoy Jacobian matrix. Furthermore, in Step 1.b, we also use the decoy Jacobian matrix to generate the misleading measurements of power flow on transmission lines, which allows the decoy measurements to mislead both FDIAs and CRAs.

B. *Step 1.b: mislead CRAs*

Background. In CRAs, attackers compromise commands delivered to end devices to change the physical states of power systems. In this paper, we focus on control-related attacks that seek to disconnect transmission lines. These operations are also used in real attacks [1]. Note that attackers can also use the commands to disconnect a substation, which is equivalent to disconnecting all the transmission lines that connect to that substation.

In this paper, we focus on CRAs that target power systems' steady states. Specifically, we consider an attack scenario similar to the one discussed in [3]: a power system is in an *insecure state* if at least one transmission line violates its physical constraints as determined by the power flow limit. In [3], we qualitatively presented the impact of CRAs on systems' dynamic states, which is affected by the parameters of the generators' physical model and feedback control. To mislead CRAs targeting a power system's dynamic states, we need to craft decoy measurements related to generators and feedback control approaches in addition to power flows on substations and transmission lines. In future work, we will quantitatively study the impact of CRAs on system dynamic states and how to mislead them correspondingly.

Preparations. Based on the study in [3], to cause physical disturbance of power systems (i.e., cause overload on transmission lines), attackers can target transmission lines that carry heavy power flows. Disconnecting those lines can cause overloading of other remaining transmission lines. To identify those critical transmission lines, attackers need to collect measurements of power flows of transmission lines and the power generations and consumptions of substations (the "measurements for preparations" of CRAs in Table 1).

In today's SCADA systems, measurements of the power usage of transmission lines and substations are periodically delivered to the control center. Consequently, under the threat model discussed in Section II.A, attackers can use the compromised devices in SCADA control networks to observe network packets and extract the measurements from the application-layer payloads.

Protection. To protect a power grid from CRAs, we craft decoy measurements such that the transmission lines with heavy loads correspond to the lines that carry light power flow in real power systems. Consequently, attack strategies that rely on decoy measurements ultimate target lightly loaded transmission lines and have little impact on the real power grid, even if the attack strategies are successfully executed.

The procedure of Step 1.b. We first list, in descending order, the active power of transmission lines that are controlled by offline devices in the real power system: $P_{I_1} \geq P_{I_2} \geq \cdots P_{I_t}$, where $I_1, \ldots, I_t$ represents the indices of transmission lines. We assign the initial decoy values of the active power of line $I_1, \ldots, I_t$ as the values (in reverse order) $P_{I_1}^{decoy} = P_{I_t}$, $P_{I_2}^{decoy} = P_{I_{t-1}}, \ldots, P_{I_t}^{decoy} = P_{I_1}$.

C. *Step 2: refine measurements*

In Step 2, we decide on the remaining decoy measurements, e.g., active and reactive power generations and consumptions in substations, and adjust the existing decoy measurements such that the decoy measurements become "legitimate." We regard the measurements as *legitimate* if they can pass bad-data detection in AC state estimations. In other words, when attackers use state estimation, they will not obtain results that indicate the existence of bad data.

The procedure of Step 2. We iteratively use AC state estimation on decoy measurements. In each iteration, we update the decoy measurement from offline devices with the results from the AC state estimation while keeping the measurements from online devices unchanged. Using the results of state estimation can bring the decoy measurements "closer" to being legitimate.

To reduce the duration of Step 2, we use a "cut-short" version of AC state estimation, similar to the method in [3]. Specifically, we reduce the number of iterations spent in each state estimation. Consequently, we make decoy measurements move "faster" towards being legitimate.

D. Case study

We illustrate the procedure for crafting the decoy measurements in a case study of a 5-bus system that contains 5 transmission lines, 3 generators, and 3 load units.

Figure 4 illustrates the procedure, including Step 1.b and Step 2. Figure 4(a) depicts the state of the real/original system. To simplify the discussion, we attach to each transmission line an end device (denoted by a pie chart in the figure) that measures the active power flow of the transmission line. We assume that the measurement of P_{12} is collected from an online device (shown in blue), while all other measurements are collected from offline ones (shown in red). In Step 1.b, we assign initial values of the decoy measurements in the reverse of the order of the real measurements: $\left(P_{34}^{decoy} = P_{45}\right) < \left(P_{32}^{decoy} = P_{15}\right) < \left(P_{15}^{decoy} = P_{32}\right) < \left(P_{45}^{decoy} = P_{34}\right)$, as shown in Figure 4(b).

In Step 2, we perform state estimation iteratively. The ultimate result is shown in Figure 4(c). After Step 2, measurement P_{12} remains unchanged. The decoy measurements, i.e., P_{32}^{decoy}, P_{34}^{decoy}, P_{15}^{decoy}, and P_{45}^{decoy}, change slightly compared to their initial values shown in Figure 4(b). However, the final decoy measurements can still map transmission lines with heavy power flows to the lines that carry light power flow in the real grid.

To assess the impact of attack strategies based on decoy measurements, we show an example attack on the 5-bus system. Figure 5(a) shows the consequences of an attack done with the knowledge of all the real measurements, which are shown in Figure 4(a). Attackers determine that the line connecting buses 3 and 4 carries the most power flows. After that line is disconnected, a transmission line connecting buses 2 and 3 is overloaded (as indicated by the red pie chart) and can be disconnected automatically. This can have a cascading effect on the whole power grid. Figure 5(b) shows that if attackers designed an attack strategy based on decoy measurements, as shown in Figure 4(c), they would target the transmission line connecting buses 1 and 5, which appears to carry the most power flow (as determined using the decoy measurements). Attackers would disconnect that line with the goal of overloading two other transmission lines. However, Figure 5(c) shows that the attackers would actually be disconnecting a transmission line that carries light power flow in the real power grid. Even if the attackers successfully disconnected this transmission line, they would not cause the overload of any transmission lines.

V. Evaluation

To evaluate Raincoat, we develop a testbed to simulate both the physical and cyber infrastructures of power systems.

- ***Power Grid Simulation***. We use MATPOWER to simulate power systems' physical infrastructures [19]. When a command is issued from the control center to end devices (simulated in the corresponding communication networks), we estimate the impact of the command and provide measurements to build network traffic.

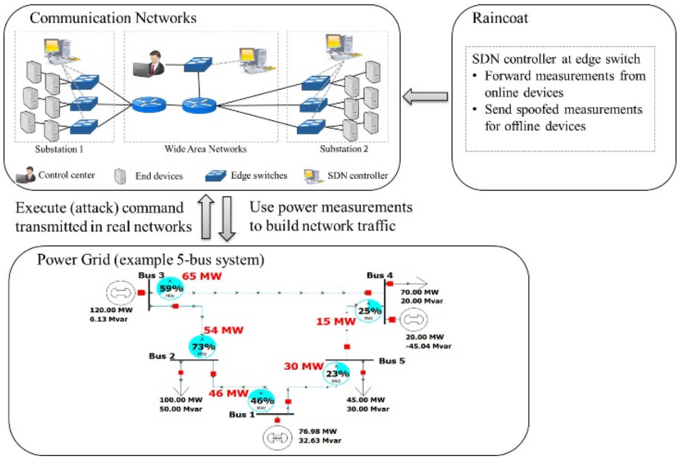


Figure 6. Cyber-physical testbed to evaluate Raincoat.

In our experiment, we simulated IEEE 24-bus, IEEE 30-bus, and IEEE RTS-96 (which includes 73 buses) systems, and three power systems representing three areas of Polish 400-, 220-, and 110-kV networks, which include a 286-bus, a 406-bus, and an 1153-bus system. The baseline configurations of the latter three systems are included in MATPOWER as examples of large-scale power systems.

To simulate the normal variability of operations in the simulated power grids, we created a benchmark profile based on one month of real data on power generation at our campus. In Figure 7, we show the power generation on the month's 12th day, which was the day power generation experienced the biggest variations. On the *y*-axis, we show the normalized power generation, where each data point corresponds to a ratio between the actual power generated at the time specified on the *x*-axis and the peak value for this month. For each simulated system, we randomly selected power generators and load units and adjusted baseline measurements for each unit by scaling them down according to the ratio selected from the benchmark data.

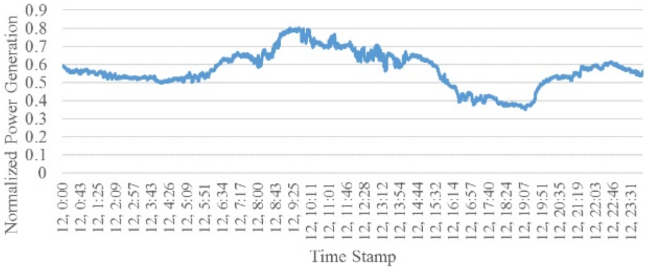
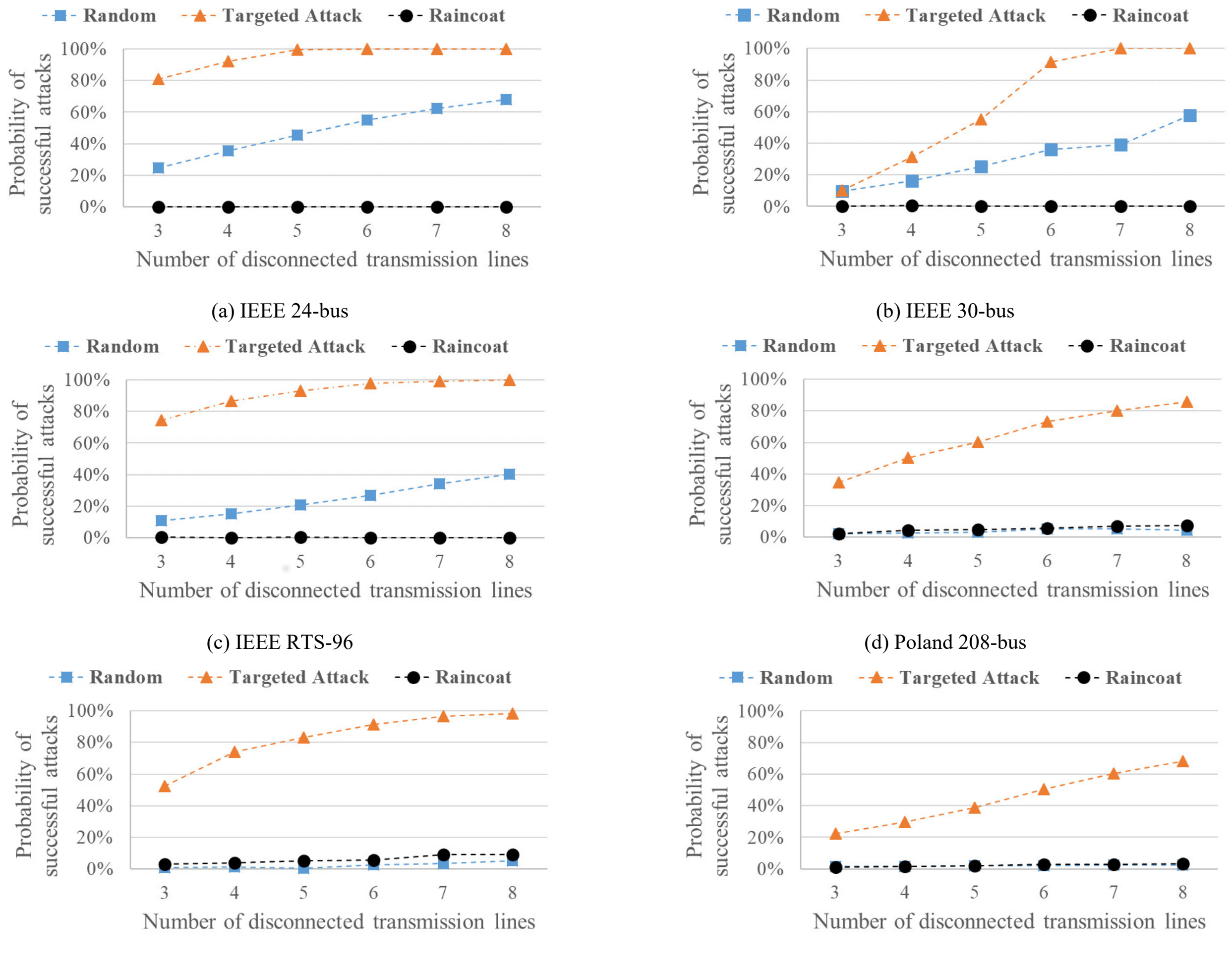


Figure 7: Recording of power generation on local campus.

- ***Communication Networks.*** We used the GENI testbed, a nationwide network experiment platform, to construct communication networks of the kinds used by SCADA



(a) IEEE 24-bus

(b) IEEE 30-bus

(c) IEEE RTS-96

(d) Poland 208-bus

(e) Poland 406-bus

(f) Poland 1153-bus

Figure 8. Comparing the probabilities of successful attacks in different evaluation scenarios.

systems to deliver commands and measurements. We used real SDN-enabled hardware switches and virtual machines in different physical locations to build control networks, which support communications between the control center and substations.

To build a control network, we follow two steps. The first step is to construct a backbone network. To do this, we used one of three topologies of communication networks: a dumbbell topology and two topologies from the TopologyZoo dataset, namely ARPANET and NSF, which are the names of two WANs used in the U.S. [18]. The second step is to connect different numbers of edge switches to the switches in the backbone network. Following these two steps, we built six different control networks in our experiment. When indicating a network in the following paragraphs, we include the name of the backbone network and the number of nodes (including switches and end devices) in parentheses. For example, Figure 6 includes the (Dumbbell, 21) network, which is a 21-node network whose backbone network uses the dumbbell topology.

In all constructed communication networks, a control center communicated to end devices by DNP3 protocol, which is the protocol widely used in U.S. power grids. Specifically, we used the open DNP3 library to implement a DNP3 master in the simulated control center and DNP3 slaves in all simulated end devices [19].

- ***Raincoat Implementation***. We implemented Raincoat as an SDN controller in ONOS, an open source network operating system [6]. To generate decoy measurement, we implemented the procedure presented in Section IV as a MATLAB module and connected the module to ONOS. Because we used DNP3 as the protocol to deliver measurements, we included in ONOS an encoder to encapsulate decoy measurements in DNP3 packets.

When attackers collect decoy measurements, they can use the state estimation to determine the system state, based on which they prepare attacks. We regard the decoy measurements as valid if they can pass the bad-data detections in the state estimation. In our experiments, we collected over 13,000 sets of decoy measurements for all simulated power systems. Over 88% of the decoy measurements are valid. For invalid decoy measurements, attackers can find bad measurements in them based on the results of the state estimation, which can raise suspicions. However, because attackers cannot obtain the real system state and design effective strategies, they may abandon their attacks. Alternatively, attackers can randomly select target devices to compromise, which can raise alerts if they access any offline devices.

A. Security evaluation

In this section, we evaluate how Raincoat disrupts FDIAs and CRAs by using randomly selected online/offline devices and decoy measurements.

Mislead FDIAs. For each simulated power system, we implemented the false data injection attacks by changing the target measurements listed in Table 1 based on the procedure in [9]. We regard the FDIA as successful if the compromised measurements based on the decoy Jacobian matrix pass the bad-data detections based on the real Jacobian matrix. In other words, the L2-norm of the measurement residual satisfies the condition $\|z_a - H\hat{x}_a\| \leq \tau$.

In our experiment, the state estimation of all simulated power systems was able to detect all compromised measurements that had been determined based on decoy Jacobian matrix H'. The experiment results show that the L2-norm of the measurement residual can be at least 50 times larger than the bad-data detection threshold, i.e., $\|z_a - H\hat{x}_a\| > 50 \cdot \tau$. In this paper, we discuss how we crafted decoy measurements of susceptance of transmission lines to mislead FDIAs based on the DC power flow model; however, the experimental results also show that the L2-norm of the measurement residual calculated based on the AC power flow model can be at least 1000 times larger than the bad-data detection threshold.

Mislead CRAs. We simulated control-related attacks as disconnections of multiple transmission lines in the power system. For each attack, we analyzed its physical consequence. If the attack caused an overload on at least one transmission line, we regarded the power system to be in an *insecure state*, and thus the attack was deemed *successful*.

To demonstrate how Raincoat disrupts and misleads CRAs, we considered three scenarios:

Scenario 1: *Random attacks* (baseline), in which attackers randomly disconnect transmission lines to cause physical perturbations. In this scenario, we simulated attackers who had little or no knowledge of the power flows on the transmission lines. Note that we use the result of Random attacks to demonstrate the effectiveness of Raincoat. Random attacks can also be detected and mitigated by the randomized device connectivity.

Scenario 2: Targeted attacks, in which attackers identify the top 15 heavily loaded transmission lines that carry most power flows and randomly disconnect some of them.

Note that in this paper, we use power flows as an example metric to determine the criticality of transmission lines. In practice, system operators can select different metrics to determine the criticality of other physical devices. The proposed algorithm for crafting decoy measurements will not be restricted by the selection of those metrics. The system operator can follow the same concept to craft decoy measurements such that critical devices identified based on decoy measurements correspond to the noncritical devices in real power systems.

Scenario 3: *Raincoat*, in which attackers use decoy measurements to identify the top 15 transmission lines that carry heavy power flows and randomly disconnect some of them.

In each *scenario*, we made 2000 attack attempts and calculated the probability of successful attacks, denoted by p_a. We show in Figure 8 how p_a changes with the number of disconnected transmission lines. We present the result for each simulated power system in a separate plot. The three scenarios are indicated by different line patterns.

One can observe that attackers with full knowledge of the target power systems (for Scenario 2, Targeted attacks) can easily put the system into an insecure state. For example, in the mid-scale IEEE RTS-96 system, disconnecting seven strategically selected transmission lines (out of a total of 120) can be sufficient to put the system into an insecure state (i.e., overloading at least one transmission line). If the attackers have little knowledge, the probability of a successful attack (in Scenario 1, Random attacks) is significantly smaller.

Comparison of the p_a of Scenario 3 (Raincoat) with that of Scenario 2 (Targeted attacks) indicates that the proposed algorithm for crafting decoy measurements can mislead attackers into targeting lightly loaded transmission lines and significantly reduces the probability of successful attacks. For the Polish 1153-bus system (the largest system simulated), the value of p_a dropped from 70% (for Scenario 2, the Targeted attack) to less than 1% (for Scenario 3, Raincoat) when 8 lines were disconnected.

More importantly, the p_a observed in Scenario 3 (Raincoat) was of the same magnitude as, or less than, the probability observed in Scenario 1 (Random attacks). Consequently, Raincoat can successfully hide the real system state and obfuscate device connectivity to mislead attackers into designing ineffective strategies.

B. Performance evaluation

In this section, we evaluate the impact of Raincoat on (i) the accuracy of the state estimation performed at the control center, and (ii) the performance of the control networks by which measurement data are collected.

1) Impact on the state estimation

Using Raincoat, we collect measurements from all end devices within the same predefined data acquisition period of *T* time units. However, the time at which each device responds with measurements is slightly affected by Raincoat. In the original data acquisition procedure (when Raincoat is not used), measurements from all end devices are collected at the same time. When Raincoat is used, the measurements are collected at different times within the window of *T* time units.

TABLE 2: IMPACT ON ACCURACY OF STATE ESTIMATION.

(*a*) Under normal variations.

Case	*Accuracy*	*Case*	*Accuracy*
24-bus	0	Poland 208-bus	0
30-bus	0	Poland 406-bus	0
RTS-96	0	Poland 1153-bus	0

(b) Under 100 times speed-up variations (with 99% confidence interval)

Case	*Accuracy*	*Case*	*Accuracy*
24-bus	1.1% (0.03%)	Poland 208-bus	1.4% (0.07%)
30-bus	1.3% (0.04%)	Poland 406-bus	1.4% (0.06%)
RTS-96	1.4% (0.02%)	Poland 1153-bus	1.5% (0.06%)

In Table 2, we show the impact of Raincoat on the accuracy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2018.2870362, IEEE Transactions on Smart Grid

of state estimation if measurements are collected at slightly different times within the window of T time units (where T is on the range of 1 to 10 seconds). The accuracies were calculated as the differences between measurements estimated by the state estimation when Raincoat was and was not used. Because of mechanical inertia, the generation of a power system changes slowly, as shown in Figure 7. Under that normal variation in power generation, we find no impact of Raincoat on state estimation, as shown in Table 2(a). Even if we experimentally speed up the variations in power generation in the simulation by 100 times, which we would consider a worst-case scenario, we observe a less than 1.5% difference in the accuracy of the state estimation.

2) *Impact on the network performance*

In Figure 9, we show the average round-trip time (RTT) (with 99% confidence interval) between the control center and end devices when we use Raincoat to manipulate data acquisition. We compare its performance with the default ONOS *Forwarding* controller, which forwards packets to output ports chosen at runtime and static routing/forwarding rules preloaded into the edge switches without interacting with any SDN controller (*Base*). We normalized the results with respect to the RTT of the *Base* flow-manipulation method.

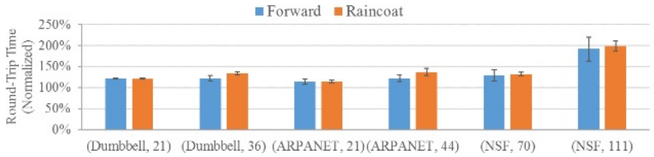


Figure 9. Comparing RTTs under three flow control mechanisms.

As shown in Figure 9, compared to the *Forwarding* controllers, the *Raincoat* controllers add a small latency, which is less than 6% on average, to encode decoy measurements in the form of DNP3 packets. Figure 9 also shows that use of *Forwarding controller* can add communications between an edge switch and an SDN controller. The resulting latency causes a 30% increase in the RTTs relative to the *Base* case for the first five networks, and an almost 100% increase for the last network (NSF, 111). The reason is that we were sharing computing resources in the GENI testbed with other projects. As the scale of the networks increased, we could not allocate sufficient computing resources for the SDN controllers to handle more traffic from the edge switches, and thus additional latency was introduced in the data acquisition. However, even with the limited computing resources, the RTTs of the data acquisitions in the (NSF, 111) network were around 120 to 150 ms, which is less than 200 ms (the required maximum time specified by IEEE standard to deliver measurements from substations to the control center [13]). To remedy the communication latency between controllers and switches, we can increase the number of controllers or allocate more computing power to run controllers when deploying Raincoat in real power systems. Based on the figures from studies on networks in data centers, the latency between the controller and switches are typically around 15 ms [31].

VI. Discussion

Integrating with power system applications. Because SDN controllers can manipulate network flows based on their application-layer payloads, we can create multiple views of online/offline devices for different power system applications. For example, we can manipulate the network flows such that a device is offline for data acquisitions but online for other power system applications, e.g., commands that operate end devices.

Integrating with real-time measurement collection. The ability to manipulate network flows for different applications makes Raincoat suitable for data acquisitions with short periods. For example, smart grids can collect PMU (phasor measurement unit) measurements 200 times per second. Dividing one such data acquisition into multiple rounds can become challenging. To overcome these challenges, we can adjust the randomization procedure in SDN controllers by prioritizing the measurements that have experienced significant changes since the last sampling timestamp and randomizing only the remaining measurements. Note that many network protocols used in power grids, such as DNP3, support unsolicited responses, which are used to deliver measurements that have big changes. Raincoat can use SDN controllers to forward the unsolicited responses directly while randomizing other responses.

The differences between real and spoofed measurements. Because spoofed measurements are changed when Raincoat changes the set of online/offline devices, there can be multiple spoofed measurements from each end device. Consequently, in the long term, attackers can observe different measurements from the same end device, i.e., the real measurement and different spoofed measurements generated at different times. The differences between those measurements may make attackers suspicious, but the attackers will not be able to distinguish real measurements from spoofed measurements. Consequently, attackers may abandon the attacks or randomly select devices as targets, which can still reduce the impact of the attacks.

VII. Related work

Moving target defense (MTD) on cyber-physical systems. In the last two decades, moving target defense mechanisms have been proposed to protect computing and network environments [20]. In [21][22], the authors assign random IP addresses and port numbers to end hosts to disrupt attackers' knowledge of target network infrastructure. As such MTDs randomize only network infrastructure and still deliver the true measurement over communication networks, attackers can still learn the physical state. Further, based on the measurements, it is possible for attackers to identify devices' identities [23]. In Raincoat, we obfuscate both network infrastructure and physical measurements; we can not only hide the cyber-physical characteristics of power systems but also use intelligently crafted measurements to mislead attackers into designing ineffective strategies.

Recent research has begun using MTDs to detect attackers in ICSes. Based on their impacts on existing physical operations, we categorize these MTD approaches as either *passive* or *intrusive*. Some "passive" MTD approaches disrupt measurements while adversaries execute their attack strategies. In [25], Miao et al. rely on a lightweight matrix to encode sensor measurements of linear time-invariant systems, such that the state estimation module can detect stealthy false data injection attacks. Compared to this work, Raincoat serves a different objective: it relies on SDN controllers to obfuscate measurements in power systems to disrupt and mislead attackers into designing ineffective strategies for both FDIAs and CRAs, before they launch malicious activities. Other "passive" MTD approach, as shown in [26], uses randomly selected measurements in state estimation to detect FDIAs. This approach can randomly remove some compromised measurements and reduce the effectiveness of the FDIAs against state estimation. However, because fewer measurements are used, the passive MTD approach reduces the measurement redundancy, which can downgrade the accuracy of existing power system applications, e.g., state estimation. In Raincoat, we randomize measurements only for potential attackers; legitimate users, e.g., the control center, can still collect the complete set of real measurements, which can maintain the accuracy of state estimation.

The "intrusive" MTD approach, as shown in [17][27][28], intentionally injects into ICSes some perturbations, e.g., by changing the communication paths or adjusting the admittance of transmission lines. System operators would use the deviations from the expected consequences of the perturbations to detect attackers. In [36], Liu et al. enhanced the MTD approach based on topology perturbations in [17] by optimizing reactance perturbations in order to identify maliciously compromised measurements in addition to detecting attacks. [37] mainly focused on improving the stealthiness of the approaches based on topology perturbation such that it becomes difficult for attackers to detect the activation of the approaches. Those approaches can expose attackers when they perform malicious activities. However, such approaches themselves heavily rely on the deployment of domain-specific devices, e.g., D-FACTS (distributed flexible AC transmission system) devices, to perform perturbations. Also, they require changes to physical operations and introduce some physical perturbations. Raincoat manipulates network flows to obfuscate the data acquisitions without changing the existing physical operations or the configuration of end devices, which make it easy to apply Raincoat to different attack scenarios. In addition, the decoy measurements and randomized device connectivity mislead attackers and prevent them from introducing any unauthorized activities, even if the activities introduce little physical damage.

Dynamic scheduling of ICS operations. In recent work, SDN has been used to adjust ICS operations, including both data acquisition and control commands, to meet different QoS requirements [29][30]. In [32][33], the authors proposed scheduling real-time measurements that have different QoS requirements to maintain the performance of control networks. In addition, SDN technology is used to increase the resilience of ICS networks in the case of accidental events, e.g., a link or node failure [30]. In Raincoat, we use SDN to manipulate network flows that deliver SCADA measurements. The difference is that our objective is to randomize data acquisitions and thus disrupt and mislead attackers and mitigate physical damage.

Honeypots for ICS. Several honeypot projects aim to build separate computing or network environments for ICSes, to attract and trace attackers' activities on ICS devices, e.g., PLCs (programmable logic controllers) [34][35]. Those ICS honeypots can mimic the cyberinfrastructure of an ICS (including the network protocols and response time). However, in their constructed network communications, the projects lack support for constructing meaningful application-layer payloads, e.g., measurements exchanged between ICS devices. Without careful design, randomly generated measurements included in communication networks can reveal the presence of a bogus environment to attackers.

Raincoat is not a honeypot for ICSes; it uses SDN to manipulate existing network flows of power systems to disrupt attackers' preparations. However, we include in Raincoat a method to craft spoofed measurements that follow the physical model of power systems. This method is based on general AC state estimation. It can be used independently in an ICS honeypot to mimic valid yet deceptive physical measurements, to increase the honeypot's authenticity.

VIII. Conclusions

This paper presents the design of Raincoat, which randomizes data acquisitions performed in SCADA systems to foil attackers in the attack-preparation stage. Raincoat manipulates network flows to transform a single deterministic data acquisition request into multiple rounds of data acquisitions of randomly selected online/offline devices. While online devices respond with real measurements, Raincoat spoofs measurements on behalf of offline devices. To spoof measurements that follow physical models of power systems, we include in Raincoat an algorithm that generates decoy measurements. Decoy measurements mislead attackers into designing (i) false data injection attacks that cannot pass the state estimation, and (ii) control-related attacks whose probability of generating physical damage is less than 1% in a real-world power system. Evaluations done in both cyber and physical domains in power systems show that Raincoat introduces a small overhead. The latency of the data acquisitions increases by less than 6%.

In future work, we plan to use Raincoat in other implementation scenarios, e.g., high-frequency data acquisition used in PMU networks, and to disrupt and mislead more complicated attacks, including control-related attacks that can affect power grid's dynamic states.

Acknowledgments

This material is based upon work supported in part by: (1) the National Science Foundation under award number CNS 1314891. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation; and (2) the Department of Energy under award number DE-OE0000780. Neither the United States Government not any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof.

References

[1] R. Lee, M. Assante, and T. Conway. Analysis of the cyber attack on the Ukrainian power grid. SANS and E-ISAC technical report, Mar. 18, 2016.

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

[3] H. Lin, A. Slagell, Z. Kalbarczyk, P. Sauer, and R. Iyer, “Runtime semantic security analysis to detect and mitigate control-related attacks in power grids,” *IEEE Trans. Smart Grid*, March 2016.

[4] R. Bobba, K. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. Overbye, “Detecting false data injection attacks on DC state estimation,” In *Proc. of Workshop on Secure Control Systems*, 2010.

[5] 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.

[6] P. Berde et al., “ONOS: Towards an open, distributed SDN OS,” In *Proc. 3rd Workshop on Hot Topics in SDN*, pp. 1–6, 2014.

[7] Raytheon BBN Technologies. GENI (Global Environment for Network Innovations) exploring networks of the future. [Online] available at: www.geni.net.

[8] D. Formby, S. Jung, J. Copeland, and R. Beyah, “An empirical study of TCIP vulnerabilities in critical power system devices,” In *Proc. 2nd Workshop on Smart Energy Grid Security*, pp. 39–44, 2014.

[9] Y. Liu, P. Ning, and M. Reiter, “False data injection attacks against state estimation in electric power grids,” In *Proc. 16th ACM Conf. Computer and Communications Security (CCS* ’09), pp. 21–32, 2009.

[10] K. Oliver, L. Jia, R. J. Thomas, and L. Tong, “Malicious data attacks on the smart grid,” *IEEE Trans. Smart Grid* (Dec. 2011), vol. 2, no. 4, pp. 645–658.

[11] A. Monticelli. “Electric power system state estimation,” In *Proc. of the IEEE* (2000) , vol. 88, no. 2.

[12] Schweitzer Engineering Laboratories, Inc. SEL-2740S Software-Defined Network Switch. [Online]. Available: https://www.selinc.com/SEL-2740S/

[13] IEEE standard communication delivery time performance requirements for electric power sub-station automation, IEEE Std. 1646-2004, 2005.

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

[15] Y. Liao and M. Kezunovic, “Online Optimal Transmission Line Parameter Estimation for Relaying Applications,” in *IEEE Transactions on Power Delivery*, vol. 24, no. 1, pp. 96-102, Jan. 2009.

[16] M. L. Crow, “State estimation” in *Computational Methods for Electric Power Systems*, 3rd. ed., Chapter 5, CRC Press, 2015.

[17] K. Morrow, E. Heine, K. Rogers, R. Bobba, and T. Overbye. “Topology perturbation for detecting malicious data injection,” In *Proc. 45th Hawaii Int. Conf. System Science (HICSS ‘12),* pp. 2104–2113, 2012.

[18] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, “The internet topology zoo,” *IEEE Journal on Selected Areas in Communications* (Oct. 2011), vol. 29, no. 9, pp. 1765–1775.

[19] Open DNP3 Group. (2012), “DNP3: Distributed Network Protocol 3.0 Google Code Archive,” [Online]. Available: http://code.google.com/p/dnp3/.

[20] D. Kewley, R. Fink, J. Lowry and M. Dean, “Dynamic approaches to thwart adversary intelligence gathering,” In *Proc. DARPA Information Survivability Conf. & Exposition II,* pp. 176–185, 2001.

[21] S. Antonatos, P. Akritidis, E. Markatos, and K. Anagnostakis, “Defending against hitlist worms using network address space randomization,” *Computer Networks* (Aug. 2007), vol. 51, no. 12, Aug. 2007.

[22] J. Haadi Jafarian, E. Al-Shaer, and Q. Duan, “OpenFlow random host mutation: Transparent moving target defense using software defined networking,” In *Proc. 1st Workshop Hot Topics in Software Defined Networks*(*HotSDN ‘12*), pp. 127–132, 2012.

[23] D. Formby, P. Srinivasan, A. Leonard, J. Rogers, and R. Beyah, “Who’s in control of your control system? Device fingerprinting for cyber-physical systems,” In *Proc. Network and Distributed System Security Symposium.* (*NDSS* ‘16*)*, Feb. 2016.

[24] K. Davis, K. Morrow, R. Bobba, and E. Heine, “Power flow cyber attacks and perturbation-based defense,” in *Proceedings of 2012 IEEE Third International Conference on Smart Grid Communications* (*SmartGridComm*), Tainan, 2012, pp. 342-347.

[25] F. Miao, Q. Zhu, M. Pajic and G. J. Pappas, “Coding Schemes for Securing Cyber-Physical Systems Against Stealthy Data Injection Attacks,” in *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 106-117, March 2017.

[26] M. Rahman, E. Al-Shaer, and R. Bobba, “Moving target defense for hardening the security of the power system state estimation,” In *Proc. 1st ACM Workshop on Moving Target Defense* (*MTD* ‘14), 2014.

[27] M. Ali and E. Al-Shaer, “Randomization-based intrusion detection system for advanced metering infrastructure,” *ACM Trans. Information System and Security (Dec. 2015)*, vol. 2, 7.

[28] S. Weerakkody, Y. Mo and B. Sinopoli, “Detecting integrity attacks on control systems using robust physical watermarking,” In *Proc. 53rd IEEE Conference on Decision and Control*, 2014, pp. 3757-3764.

[29] A. Cahn, J. Hoyos, M. Hulse and E. Keller, “Software-defined energy communication networks: From substation automation to future smart grids,” In *Proc. of IEEE Int. Conf. Smart Grid Communications*, pp. 558–563, 2013.

[30] X. Dong, H. Lin, R. Tan, R. Iyer, and Z. Kalbarczyk, “Software-defined networking for smart grid resilience: Opportunities and challenges,” in *Proc. 1st ACM Workshop on Cyber-Physical System Security*, pp. 61–68, 2015.

[31] R. Sherwood, et al., “Can the production network be the testbed,” In *Proc. 9th USENIX Conf. Operating Systems Design and Implementation* (*OSDI ‘10*), pp. 365–378, 2010.

[32] A. Goodney, S. Kumar, A. Ravi, and Y. Cho, “Efficient PMU networking with software defined networks,” in *Proc. IEEE Int. Conf. Smart Grid Communications*, pp. 378–383, 2013.

[33] K. Nagananda, S. Kishore and R. Blum, “A PMU scheduling scheme for transmission of synchrophasor data in electric power systems,” *IEEE Trans. Smart Grid (Sept. 2015)*, vol. 6, no. 5, pp. 2519–2528.

[34] K. Wilhoit and S. Hilt, “The GasPot experiment: Unexamined perils in using gas-tank-monitoring systems,” TrendLabs, Trend Micro Inc., August 2015. [Online]. Available: https://www.blackhat.com/docs/us-15/materials/us-15-Wilhoit-The-Little-Pump-Gauge-That-Could-Attacks-Against-Gas-Pump-Monitoring-Systems-wp.pdf.

[35] D. Buza, F. Juhász, G. Miru, M. Félegyházi, and T. Holczer, “CryPLH: protecting smart energy systems from targeted attacks with a PLC honeypot,” in *Proc. Int. Workshop Smart Grid Security*, pp. 181–192, 2014.

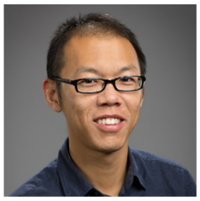
[36] C. Liu, J. Wu, C. Long and D. Kundur, “Reactance Perturbation for Detecting and Identifying FDI Attacks in Power System State Estimation,” in *IEEE Journal of Selected Topics in Signal Processing*.

[37] J. Tian, R. Tan, X. Guan and T. Liu, “Enhanced Hidden Moving Target Defense in Smart Grids,” in *IEEE Transactions on Smart Grid*.

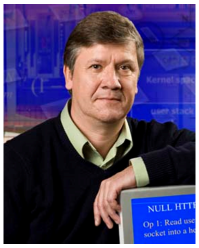
[38] M. A. Rahman and H. Mohsenian-Rad, “False data injection attacks with incomplete information against smart power grids,” *2012 IEEE Global Communications Conference (GLOBECOM)*, Anaheim, CA, 2012, pp. 3153-3158.

[39] J. Kim, L. Tong and R. J. Thomas, “Subspace Methods for Data Attack on State Estimation: A Data Driven Approach,” in *IEEE Transactions on Signal Processing*, vol. 63, no. 5, pp. 1102-1114, March1, 2015.

[40] M. Esmalifalak, H. Nguyen, R. Zheng and Zhu Han, "Stealth false data injection using independent component analysis in smart grid," *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Brussels, 2011, pp. 244-248.



Hui Lin is an Assistant Professor at the Computer Science and Engineering Department in the University of Nevada at Reno. He earned his Ph.D. degree from the University of Illinois at Urbana-Champaign in 2017 in electrical and computer engineering. His research interests include cyber security, intrusion detection systems, and software-defined networking (SDN) in the areas of cyber-physical systems, such as power systems. He has successfully adapted Bro, a runtime network traffic analyzer, to support network protocols (e.g., DNP3) commonly used in power grid infrastructure. The DNP3 analyzer that he developed has been included in Bro and can be downloaded freely by utility companies. His current work focuses on applying SDN in cyber-physical systems; he intends to use SDN's network programmability to design flexible cyber-physical systems which can quickly respond to cyber-attacks and accidents.



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 also involves 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 the Program Chair for the International Conference on Dependable Systems and Networks (DSN) in 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 the IFIP Working Group 10.4 on Dependable Computing and Fault Tolerance.

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, serves as Chief Scientist of the Information Trust Institute, and is affiliate faculty of the National Center for Supercomputing Applications (NCSA). He currently 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's Special Interest Group on Security. Professor Iyer is also the recipient of the degree of Doctor Honaris Causa from Toulouse Sabatier University in France.