DecIED: Scalable k-Anonymous Deception for IEC61850-Compliant Smart Grid Systems

Dianshi Yang
dianshi_yang@mymail.sutd.edu.sg
Singapore University of Technology and Design

Daisuke Mashima, Wei Lin
{daisuke.m,lin.wei}@adsc-create.edu.sg
Illinois at Singapore Pte Ltd

Jianying Zhou
jianying_zhou@sutd.edu.sg
Singapore University of Technology and Design

ABSTRACT

As demonstrated by the past real-world incidents, sophisticated attackers targeting our critical infrastructure may be hiding in the system, perhaps at this moment, in order to collect information and prepare for massive attacks. If an attacker is mostly passive and monitoring SCADA communication traffic or is clever enough to act under the radar of intrusion/anomaly detection systems, it is challenging to counter them. In this direction, deception technology is an effective cybersecurity tool, by deploying a large number of dummy and decoy devices throughout the system infrastructure to be protected, for capturing probing attempts and lateral movement of persistent attackers and malware. In this paper, we discuss the practical design and implementation of high-fidelity deception devices for smart power grid systems, named DecIED. DecIED imitates the device characteristics and communication models of IEC 61850-compliant IEDs (intelligent electronic devices) and thus realize k-anonymous smokescreen, which virtually shows $k-1$ indistinguishable decoy devices, to protect our critical infrastructure. Based on our prototype implementation, a single industry PC can host over 200 deception devices, which demonstrates DecIED's scalability and feasibility of integration into the existing systems.

CCS CONCEPTS

• **Security and privacy** → **Intrusion/anomaly detection and malware mitigation**; • **Computer systems organization** → **Embedded and cyber-physical systems**; • **Networks** → **Cyber-physical networks**; • **Hardware** → *Smart grid*.

KEYWORDS

smart grid; deception technologies; IEC 61850; cyber security

ACM Reference Format:
Dianshi Yang, Daisuke Mashima, Wei Lin, and Jianying Zhou. 2020. DecIED: Scalable k-Anonymous Deception for IEC61850-Compliant Smart Grid Systems. In *Proceedings of the 6th ACM Cyber-Physical System Security Workshop(CPSS '20), October 6, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3384941.3409592

CPSS '20, October 6, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7608-2/20/10…$15.00
https://doi.org/10.1145/3384941.3409592

1 INTRODUCTION

In order to counter emerging cyber threats targeting our critical infrastructure, the existing cybersecurity solution, such as firewall, data diode, centralized intrusion/anomaly detection systems would not be fully effective. For instance, as demonstrated by the number of real-world incidents in the past years, such as Ukraine power plant attacks in 2015 and 2016 [66], persistent attackers were hiding in the smart grid infrastructure to collect information about systems for over 6 months. During such attack preparation (or reconnaissance) stage, attackers are likely sending query/control commands to collect intelligence about the infrastructure, flying under the radar. Furthermore, during the reconnaissance stage, an attacker could be completely passive and just overhearing communication in the control system infrastructure. Such activities may not be effectively detected by intrusion detection systems [42, 50, 62], or other security solutions such as industrial firewall.

When prevention and detection are bypassed, the potential next line of defence is to prevent attackers from learning the system. In order to make the reconnaissance difficult as well as to capture such activities, deception technologies are considered effective [26, 45]. At the high-level, deception technologies blend fake, but apparently real, devices (often called decoy devices or deception devices) in the system infrastructure. This way, an attacker, which may be passively sniffing network traffic, could not tell which are the real devices connected to the physical power grid system or what the real system topology is like. The fake device also works as a sensor (or tripwire), which raises an alarm when it is touched, detecting attackers' probing activity during the attack preparation phase. While there are some commercial services or solutions of deception technologies for enterprise IT systems, e.g., [9, 17], deception technologies for industrial control systems, including smart grid systems, are still immature and poses unique challenges. For example, devices used in industrial control systems would have unique device characteristics (e.g., OS fingerprints and services run on devices). Even more challenging is the imitation of communication patterns. The deception devices should use the same communication protocols as the real one, and the communication models and timings should be indistinguishable from the real devices. The other challenge comes from its cyber-physical nature, where messages transmitted in the cyber system should provide system-wide consistency about the physical system. For instance, voltage and current measurements should be consistent with the physical laws of power systems. Otherwise, sophisticated attackers with domain knowledge may be able to identify real devices from decoying.

In this paper, we discuss the practical design of high-fidelity deception devices for smart power grid systems, named *DecIED* or a *dec*eption *IED* (intelligent electronic device). DecIED imitates an

IED, which is an intelligent device that works as a communication end-point in the cyber side while monitoring or controlling physical power system devices in the modernized substation system. DecIED is compliant with IEC 61850 standards [4, 7], the established international standard for substation automation communication, and thus can be easily integrated into the standard-compliant substation network without requiring a major change or upgrade on existing devices or interfering with the functionality of existing (real) IEDs. Among the variants of deception technologies, in this paper we focus on the most basic, but fundamental, one, namely "smokescreen" with k-anonymity concept [21], which offers $k-1$ fake devices that appear and behave like the real IEDs. DecIED takes advantage of process-level communication using IEC 61850 to imitate the behaviour of a real IED in a real-time manner. When persistent attackers scan the system, they would find multiple devices that identically look and behave. Our particular focus in this paper is to prevent attackers from pinpointing real IEDs to compromise and/or learning the real system topology.

In order to demonstrate the practicality of the idea, we further present a proof-of-concept DecIED implementation that can imitate device/OS characteristics of real IEDs that are deployed in a state-of-the-art smart grid testbed [6] as well as communication models of IEDs that implement control logic that is often found in real-world substations. The comprehensive metric or methodology to evaluate deception technologies is still an open problem to our knowledge. However, because our goal is to implement k-anonymity, we evaluate the implementation in terms of resemblance and scalability. We show that, on commodity industrial PCs, we can run as many as 200 deception device instances. We believe DecIED can serve as a crucial building block for advanced deception solutions such as moving target defence (e.g., [40]) as well as other deception strategies discussed in [21].

The rest of the paper is organized as follows. In Section 2 we discuss state-of-the-art technologies in the relevant areas. In Section 3, we provide an overview of devices that enable smart grid communication and control and also their communication models based on IEC 61850 standards. Section 4 defines our design goals and technical challenges to tackle with. We then discuss our approach as well as DecIED proof-of-concept implementation in Section 5 and Section 6. We present the evaluation in Section 7. Finally, we conclude the paper with future research directions in Section 8.

2 RELATED WORK

There are a wide range of security solutions developed for security smart grid systems such as intrusion/attack detection systems [28, 57, 62], remote attestation [23, 29, 32], cryptographic solutions, such as IEC 62351 standard [5] and the reference [30], and cyber and physical zoning, as summarized in [61]. The technology explored in this paper is categorized under in-network deception, and is considered orthogonal and complementary to the other categories.

Honeypot is also considered as a deception technology, and there exist a number of open-source honeypot implementations for industrial control systems [10, 12, 53, 54]. Among them, Conpot [10] is an open-source, low-interaction honeypot designed for industrial control systems (ICS) and is actively maintained. Conpot supports several Internet and ICS-specific protocols such as Modbus. One major limitation is that it does not make sufficient consideration to maintain cyber-physical system consistency. Another limitation is that it is relatively easy to get fingerprinted [16, 51].

Many of the prior honeypot efforts in the smart grid area, which is also our focus, including CryPLH [27] and SHaPe [37], only imitate the cyber side, and thus, not enough to deceive power-system-aware attackers. Some other honeypot systems, such as [46, 56], utilize power-flow simulation. However, the use of simulators is impractical to imitate power grid system behaviour in real-time, high-fidelity manner, thus not suitable for implementing in-network deception technologies. We should note that, while honeypot is typically isolated from real systems, in-network deception devices are blended in the real system infrastructure and thus the cyber-physical view it presents must be more-tightly consistent with real systems. This is one of the reasons why we did not utilize simulators.

In a general IT context, [21] introduces a strategy composition for resilient cyber deception called CONCEAL with an optimal composition of various concealment techniques to maximize the deception utility. This cyber deception framework is a composition of mutation, anonymity, and diversity to maximize key deception objectives. The paper also illustrates desired properties for cyber deception techniques, including scalability for large numbers of hosts or services. CONCEAL framework composes m-mutation for address anonymization, k-anonymity for fingerprint anonymization, and l-diversity for configuration diversification. m-mutation means the address of the host alters per $1/m$ seconds. k-anonymity means that for a single host there are k-1 shadow hosts along with the real host with similar or identical fingerprints. l-diversity means that for each kind of service there are l-1 fake services in the same kind with different versions or vendors along with the real one. In the smart grid context, address mutation is not often practical to avoid major changes in the existing infrastructure. Moreover, l-diversity may not make enough sense given the limited number of services implemented. Thus, in this paper, we focus on k-anonymity to implement smokescreen.

The comprehensive evaluation of deception technologies, including honeypot systems, is still an open question. As discussed in [22], the quantitative evaluation of realism of ICS honeypot system is not feasible. Furthermore, we argue that realism alone does not capture the effectiveness of deception technologies because deception against human attackers would be a mind game, and therefore involvement human subjects would be a must. Such an approach for evaluating attack detection systems has been recently attempted [20], but it is not in general feasible. In addition, the outcome would be still biased depending on the quality of participants. In this paper, we narrow down the scope only to k-anonymity and thus focus on the evaluation of indistinguishability in device characteristics and communication patterns, as well as scalability.

Moving target defence (MTD) technologies for perturbation of cyber and/or physical topology to confuse attackers are explored, for instance in [40, 41, 60]. These are effective in preventing attackers from obtaining accurate system information to launch successful attacks, but it is not designed for capturing probing/reconnaissance activities attempted by persistent attackers that are hiding the infrastructure to prepare for a large-scale attack. Having that said,

implementing MTD on top of the deception devices is a promising future research direction.

Recently, deception technology for modernized substation systems is proposed by Lin et al. [43]. Their proposed technology utilizes software-defined networking and "seed devices", which are real power grid devices but, behind SDN (software-defined networking), interact with attackers on behalf of virtual, decoy devices to present realistic device characteristics, etc. Their system requires deep integration of software-defined networking, which requires major upgrades in smart grid network architecture. Besides, their design does not elaborate imitation of process-level communication and interaction among IEDs, for instance by means of protocols like IEC 61850 GOOSE. In these aspects, our solution has advantages over their scheme.

3 MODERNIZED SUBSTATION AND IEC 61850

In this section, we provide background about devices deployed in the smart power grid system, in particular modernized substations, as well as standard communication protocols used in the system.

3.1 Intelligent Electronic Device (IED)

Smart grid is, at the high level, a power grid system enhanced with ICT (information and communication technologies). While smart grid is a large-scale, inter-connected, complicated system, substation is one of the most crucial components that is responsible for reliably delivering electricity from generator to consumers through control of power grid topology as well as transformation of voltage at stages. By deploying intelligent devices, such PLCs (programmable logic controllers) and IEDs (intelligent electronic devices) in modernized substations and allowing real-time communication among them, smart grid aims at improving efficiency and effectiveness of power grid operation by means of telecontrol and automation [34, 47].

An IED works as a communication end-point in the cyber side of the smart grid system and also interacts with physical power grid components in the substation, such as circuit breakers, transformers, and so forth, according to the received commands and messages. IEC 61850 [7] is becoming popular for such communication within a modernized substation. For the sake of timely control and monitoring, communication among IEDs typically has stringent latency requirements [55]. Communication models of IEC 61850-compliant IEDs will be elaborated later in this section.

PLCs are also often found in substation systems, and they implement logic for automated control based on power grid measurements and status. Thus they can be seen as an advanced version of IEDs. In this paper, we mainly focus on the design and implementation of deception IED devices, while a similar design is applicable for implementing deception PLC devices.

3.2 IEC 61850 Communication Models

International Electrotechnical Commission (IEC) defines the standardized communication methods and patterns among devices in the substation network in the standard IEC61850 [7], for the design of the substation automation and control communication system [64].

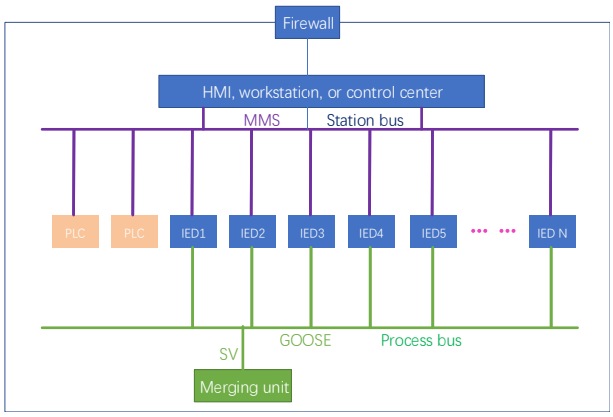

Figure 1: Topology of Substation System in Smart Grid

Figure 1 shows high-level view of the typical substation topology. The topology consists of multiple levels. SCADA/HMI (supervisory control and data acquisition/human-machine interface), substation gateway (or protocol translator) general-purpose workstations, servers, routers, firewalls, etc. are connected to "station bus" while merging units (e.g., meters and sensors) and physical power grid devices are connected to "process bus". IEDs are usually connected to both buses while PLCs may or may not have a connection to the process bus.

Roughly speaking, communication among ICS devices in the station bus uses IEC 61850 MMS (Manufacturing Messaging Specification) protocol while the communication in the process bus utilizes IEC 61850 GOOSE (Generic Object Oriented Substation Event) and SV (Sampled Values) protocols. As a reference, a communication model implemented in a state-of-the-art, IEC 61850-compliant substation testbed is discussed in [59]. The SCADA/HMI on the station bus sends the remote control commands and interrogation commands using IEC 61850 MMS. IEC 61850 standards basically define the information model for the interoperable communication among multi-vendor ICS devices, and, to transport messages, the model is mapped onto multiple protocols at different layers. The models, as well as data, attributes that are processed on each IED is defined in an SCL (substation configuration language) files [44]. SCL files may further describe the functional structure and the relation/connectivity among IEDs [4].

The MMS protocol is unicast, client-server type communication over TCP/IP. On the other hand, in the process bus, the GOOSE protocol is utilized for announcing status updates among IEDs. SV protocol is also utilized, typically by MUs (merging units), on the process bus for sharing power grid measurements. Owing to the stringent latency requirements on these communications [35], the publisher-subscriber model using link-layer multicast is employed. GOOSE is a type of multicast messaging with user-defined values in the dataset, such as the changes of state of substation parameters, based on a publisher/subscriber model [52]. For multicast, the GOOSE publisher will send its messages to all the devices in the same LAN in a fixed interval (e.g., in the order of seconds), and when any status update occurs, GOOSE messages are triggered immediately and re-transmitted with shorter interval, which then gradually goes back to the default interval.

According to the specification [7], a GOOSE frame can be divided into the following parts: the source and destination MAC addresses, the application protocol data unit (APDU), etc. The APDU involves the GOOSE Control Block reference (gocbRef), the time allowed to live, the GOOSE ID (goID), the status number (stNum), the sequence number (sqNum), and so forth. The user-defined data in the last portion can be defined by the users with different types of data attributes in Boolean, integer, bitstring, etc. [39]. GOOSE Control Block, along with destination MAC address, is utilized by a recipient IED to decide whether a received GOOSE frame is supposed to be processed by it or not.

SV (Sampled Values) is a protocol for the transfer of digitized samples of analog measurements [38]. While SV also uses the same communication model as GOOSE, the difference is that SV is sent at the constant rate (e.g., 80 messages per cycle, which translates into 4,000 messages per second in a 50Hz power grid system). The merging unit (MU) can distribute up-to-date power grid measurements to IEDs by SV protocol.

4 DECEPTION TECHNOLOGIES FOR SMART GRID SECURITY

During the past years, we have witnessed a number of cyber-originated attacks targeting smart grid systems. One of the examples is the cyberattack on Ukrainian power grid [58]. In the incident in 2015, attackers successfully executed so-called ICS cyber kill chain to penetrate into the smart grid control systems. Attackers were hiding and collecting system information for 6+ months before the actual attack. Moreover, in the incident in 2016, the malware named CrashOverride [11] was utilized. This new malware is the second-ever known case that can disrupt physical systems after Stuxnet [18]. The notable advancement seen in this malware was that the infected device can emit standard-compliant smart grid control/monitoring commands, impersonating a legitimate control center, and thus it can lead to mass power outages automatically. In addition, after the attack in 2015, the Ukrainian attack in 2016 was fully automated. The malware was programmed to be able to link to the devices and control them with commands by obscure protocols directly, which means that it can attack the grid system more rapidly without the management of humans, feedback from operators, or even without the connection to the Internet as a logic bomb. Even more recently, there were a number of incidents where hackers successfully penetrated into the power grid systems [24, 33]. Such incidents demonstrated that attackers hiding in our critical infrastructure is a realistic security threat.

After an attacker successfully penetrates the smart grid system the plausible next step is reconnaissance and probing to collect information about the system in both cyber and physical sides. During this phase, an attacker or malware may attempt to send ICS commands (e.g., interrogation and/or innocuous-looking control commands) to actively interact with the devices in the infrastructure. Malware may attempt lateral movement from an infected IED to another for propagation. Another type of attacker may stay completely passive to sniff ICS network traffic to derive system topology etc.

Deception technology, which deploys a number of decoy (often also called deception) devices that are indistinguishable from real devices, is considered effective to counter attackers of both types. Specifically, decoy devices would work as a "tripwire" or minefield that raises an alarm when an attacker or malware steps on any of them. At the same time, decoy devices could generate dummy network traffic to confuse passive attackers. For example, if we configure decoy devices so that they send out the same or similar information as a real device, it will prevent attackers from telling which one is the real device connected to the physical system. While blended in the real system infrastructure, deception/decoy devices should be isolated from real devices and their operation to avoid negative impact on security as well as the availability of them.

We assume that attackers/malware are in the control system infrastructure but not yet equipped with sufficient system knowledge. Attackers in our scope may have footprints in the substation network and compromise the HMI or engineering workstation at the station level, e.g., via compromised VPN interface. While usually the station bus and process bus are implemented as a separate network, we don't exclude the possibility that attackers have access to the process-bus communication (i.e., IEC 61850 GOOSE and SV).

Under such an attacker model, we focus on design of IEC 61850 compliant deception IEDs, named *DecIED*, that imitates characteristics and behavior of a real IED (also called a *base IED* hereafter) to realize "k-anonymity" concept. In other words, $k-1$ deception IED instances are run for each base IED in such a way that it is infeasible to identify which one is the real device. Reference [21] also discussed further concepts such as "l-diversity" and "m-mutation", and our design can be extended to support them in a straightforward manner. Furthermore, by wisely coordinating status and measurements messages reported by DecIED, we can mislead/lure attackers to mount non-optimal attack or attacks that can be detected or prevented by other security schemes, by incorporating the moving target defence [40]. Such extension is left for our future work.

While deception technology has been explored for general IT systems, ones for industrial control systems or more specifically for smart grid systems is still in the early stage. Below, we summarize requirements for deception devices for smart grid systems that we pursue in this paper.

(1) Imitation of device characteristics
(2) Imitation of communication model and patterns
(3) Scalability and deployability

For (1), all the DecIED instances in the system should have the same features as the base IED, so that the attackers cannot distinguish them. Features include network services (i.e., open ports), version or specification of network services running, and MAC addresses that belong to the same vendor as the base IEDs'. In addition, OS fingerprint that can be remotely obtained via attacker tools such as Nmap [15], should also be similar enough to avoid hinting attackers.

Requirement (2) includes support of smart grid communication protocols (e.g., IEC 61850) and implement appropriate communication patterns, such as ones discussed in Section 3. The communication patterns will be further elaborated in Section 5.2. Another challenge here is that DecIED should provide sufficient imitation in terms of the payload of IEC 61850 messages it emits. Besides, it is as important to be accessed by SCADA/HMI or other real devices as well as to behave in a similar way when handling received messages

in terms of both response content and timing. While it is crucial to make DecIED behaves in the same way as the base IED, the DecIED's activities, including messages sent out, should not affect the functionality of real IEDs, PLCs, and SCADA/HMI or overall system availability. This property is particularly important when DedIED inject fake/crafted information in our future extension.

Lastly, in order to make deception effective, we should be able to run a sufficiently large number of DecIED instances with a reasonable introductory and management cost. Moreover, deception technology can be integrated into existing infrastructure without requiring major changes or updates, which leads to (3). Regarding scalability, we aim at running over 200 DecIED instances on a single, commodity industrial PC. Based on "k-anonymity" concept, if there are 20 base IEDs in a substation, which is equivalent to a reference substation model discussed in [25], we can implement 10-anonymity or higher. In other words, an attacker would see 10 IEDs that look and behave in the same manner. Moreover, in the case of EPIC testbed [6], there are roughly 10 IEDs, and thus deploying 200 DecIED instances allows us to offer 20-anonymity.

5 APPROACH AND DESIGN OVERVIEW

In this section, we elaborate on our approach to addressing the design goals discussed in the previous section to lead the system design.

5.1 Imitation of Device Characteristics

We started with studying network and OS characteristics of real IEDs. In particular, we investigated Siemens IEDs deployed in publicly-accessible smart power grid testbed hosted by Singapore University of Technology and Design, called EPIC [6, 59], to use them as the base IEDs. By using Nmap [15] and Wireshark [19], we conducted a comprehensive scan to identify the following information:

- MAC address
- Network services (Open TCP/UDP ports and version/header information of each service)
- OS fingerprinting results

Faking MAC address is relatively straightforward and can be done by changing network interface configuration. Thus, we assign MAC addresses that belong to the same vendor. Regarding network services provided by the Siemens IED, we found that IEDs in EPIC testbed opens only port 80 and 102, which corresponds to HTTP (for offering web-based administrator interface) and IEC 61850 MMS respectively. We use Ngnix web server, which is widely used for an embedded platform, with customized HTTP headers to implement the former and virtual IED module based on open-source libIEC61850 [8], which will be elaborated later in this paper, for the latter.

Countering OS fingerprinting is not trivial. Because many of the fingerprinting techniques/tools investigate the distinction of protocol stack implementation of each device or OS. Thus, the ideal solution would require kernel-level modification. Since it is not feasible in practice, in this paper we employ an open-source tool called Honeyd [54] for our proof-of-concept. Honeyd has a feature to spoof OS fingerprints, by configuring the instance with fingerprint collected from the device of interest. Both network services discussed earlier are run behind Honeyd, and Honeyd can "proxy" incoming requests to these servers. We admit that Honeyd is not a complete solution to counter fingerprinting, but, given that IEDs only works as a server (i.e., passive), it is still effective to counter popular network scanners, like Nmap [48]. We should note that some smart grid devices, such as PLCs, sometimes work as a client (i.e, actively initiating TCP connection). Since outgoing traffic is not mediated by Honeyd, more radical solution will be needed for spoofing device fingerprints, which is part of our future work. We don't claim novelty regarding the use of Honeyd, and it is utilized for the sake of practical proof of concept.

5.2 Imitation of Communication Models and Patterns

Our goal in this paper is to design DecIED that behaves in the same way as a base IED. Taking the communication models discussed in Section 3, this can be divided into the following requirements in terms of communication models.

- Responds to IEC 61850 MMS interrogation commands from SCADA/HMI and substation gateway in the same way as the base IED
- Responds to IEC 61850 MMS control commands from SCADA/HMI and substation gateway in the same way as the base IED
- Sends IEC 61850 GOOSE message with the same (or similar) status/measurements and the same periodicity as the base IED, when there is not a status update.
- Sends IEC 61850 GOOSE message triggered by status changes on the base IED.
- Acts on the received GOOSE message in the same way as the base IED.
- Messages sent by DecIED should not cause any influence on real devices in the system

In order to meet these requirements, it is imperative for DecIED and the base IED to share a synchronized, consistent physical system view. One possible solution is to run a back-end power flow simulator and connect all DecIED to it, as attempted in [46]. However, it is in practice not feasible to keep the simulation model completely up-to-date to maintain synchronization and consistency with base IEDs. Another drawback is latency and resource consumption to run power flow simulation. As discussed in [49], running power system dynamics simulation would require non-negligible latency.

Another solution is to rely on multicast, process-bus communication using IEC 61850 GOOSE and SV. As discussed in Section 3, IEDs rely on IEC 61850 SV communication on the process bus for acquiring real-time power grid measurements. GOOSE messages are utilized for announcing any types of status updates on IEDs. As long as DecIED are deployed in the same network (more specifically, in the same broadcast domain), DecIED can hear the same messages at the same time as the base IED, and thus can have the information and system visibility synchronized with the base IED. By hearing and processing the process-bus communication, DecIED can acquire the same information as the base IED nearly at the same time, which allows DecIED to reply the same information as the base IED when it gets interrogation commands. The same applies to the GOOSE message reporting measurements and status.

IEDs may execute some control logic or status update based on the received GOOSE and SV messages. For instance, based on the current measurements conveyed in SV messages, IEDs would trigger some circuit breaker control to protect the power grid and then announce status updates using GOOSE. In order to imitate behavior of the base IED when it acts on received GOOSE and SV messages, the simplest solution would be to just sniff messages outgoing from the base IED and replicate them while updating DecIED's status accordingly. Although it is easy to implement, there is a major drawback. Namely, because in such a design the base IED always acts first, attackers could easily learn which one is the real device by overhearing the traffic for a certain duration. Another approach is to implement a processing logic implemented on the base IED. Given that process-bus communication is heard by the base IED and DecIED nearly at the same time, both can work on it simultaneously and thus timing would not give attackers any meaningful clue to identify real devices. Fortunately, we have access to a power grid testbed, where there are overviews in [6, 59], and studied the real IEDs deployed there. Based on our study, the logic implemented on IEDs is relatively simple and described in block logic diagrams, so replication of the logic is still doable as long as the design documentation is available. In this work, we implement some representative logic that is often found in real-world substations, which will be discussed in our proof-of-concept implementation shown in Section 6. This way, without having extra coordination among the base IED and DecIED, it is possible to imitate the communication models and patterns.

The remaining challenge is to prevent messages sent by DecIED from affecting real devices. This can be addressed by utilizing IEC 61850 GOOSE specification. Specifically, an IEC 61850 compliant IED is supposed to check GOOSE Control Block, which consists of multicast MAC address, AppID, etc., to see whether the corresponding message should be processed by it or not. Using this protocol specification, by assigning AppIDs that are *NOT* used in real systems or devices to the GOOSE messages sent by DecIED, we can make the DecIED's messages ignored (and silently discarded) by real devices. To avoid AppID from being a hint for attackers to identify real IEDs, we recommend to assign unique AppIDs for all IEDs (regardless of whether real or deception devices). Although we focused on AppID, other parameters used for filtering (e.g., multicast MAC address) can be used instead, when configuration of AppID is not feasible owing to the system configuration.

5.3 Scalability and Deployability

To facilitate the integration of the deception technology into the existing infrastructure, we design the solution on a single security appliance box (e.g., an industrial PC) that is connected to switches at both station bus and process bus. Generating virtual network interfaces on the PC, each DecIED instance can act with a unique MAC address. While the use of virtual machines for each DecIED instance would be an alternative option, it would not scale to support a large number of DecIED instances. The deployment is summarized in Figure 2. In addition, according to our design approach discussed in earlier subsections, DecIED will not require computationally heavy processing such as power system dynamics simulation, which is also expected to contribute to enhanceing scalability.

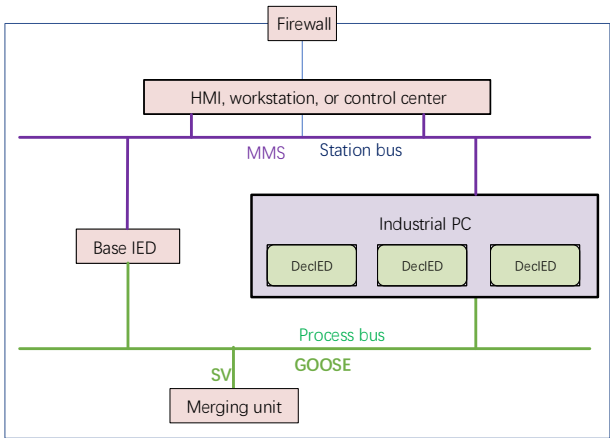

Figure 2: The topology of the network with DecIED

5.4 DecIED Prototype Module Architecture

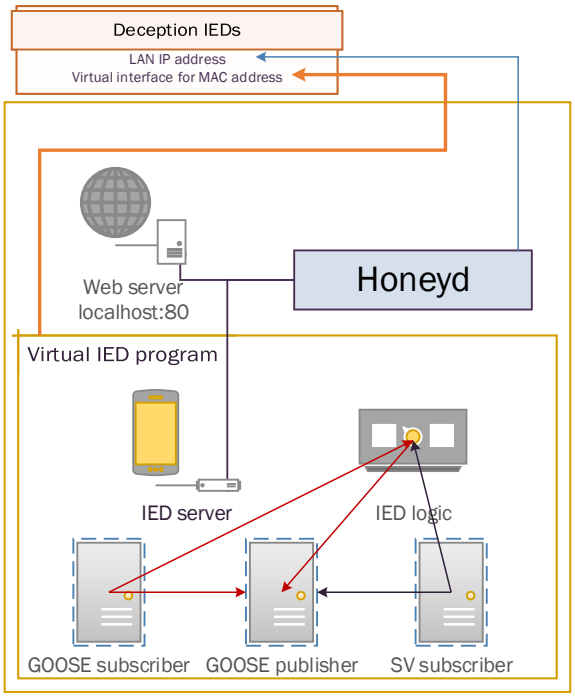

Figure 3: Module Architecture of DecIED

Figure 3 shows the architecture of the DecIED implementation. As seen in the figure, each DecIED consists of Honeyd with unique IP and MAC addresses and a virtual IED module. Honeyd proxies incoming TCP requests for port 80 and 102 to backend Nginx web server and virtual IED module, which implements IED 61850 MMS server, respectively.

The virtual IED module consists of the following components: the GOOSE publisher/subscriber, SV subscriber, IED server, and IED logic. The GOOSE publisher and subscriber are to multicast GOOSE messages about DecIED's states and receive GOOSE messages from the real IEDs for learning the latest IED status. The SV subscriber aims to receive the real-time measurement values in current and

voltage from merging units. The IED logic implements control logic based on power grid measurements as well as other IEDs' status change that is implemented on the base IED to imitate. This may include protection mechanisms, which will be elaborated in Section 6.1.

5.5 Discussion

Besides the AppID tweak discussed above, in order to make the DecIED's communication model indistinguishable from the base IED's, we also need to make SCADA/HMI or substation gateway to interact with DecIED instances just like the base IED. For instance, when a SCADA/HMI queries status from the base IED, the same interrogation request should be sent to DecIED instances too. One solution for this issue is that we register DecIED instances on the SCADA/HMI along with the base IED so that all are interrogated equally. In this case, we also need to prevent the SCADA/HMI system (or human operator) from getting confused. A trivial solution would be to configure the list of DecIED instances on the SCADA/HMI to be excluded. Unfortunately, this is not an ideal solution for a couple of reasons: first, we need to make changes on the SCADA/HMI, and secondly, once an attacker compromises the SCADA/HMI, deception is no longer effective (i.e., SCADA/HMI must be fully secure).

Although the generic solution that works for any kinds of deception technologies may not be trivial, our k-anonymous DecIED design inherently avoids this problem. Recall that all DecIED instances and the base IED share the power grid measurements (via SV messages) and status (via GOOSE messages). Thus, measurements and status reported by all are essentially the same (i.e., reporting the same values for the same power grid component). Thus, from the SCADA perspective, they are treated as redundant, repeated messages. Therefore, even when DecIED instances are equally registered on SCADA/HMI and historian database, the SCADA/HMI or human operators will not be confused. This way, we can hide any traits of deception devices in the communication patterns as well as configuration of the SCADA/HMI, and therefore, even if SCADA/HMI is compromised, an attacker cannot break the deception technology.

6 PROOF-OF-CONCEPT VIRTUAL IED

Among the components illustrated in Figure 3, the virtual IED module plays an essential role in attaining indistinguishability from real IEDs. Thus, in this section, we elaborate on the design details of it with hypothetical but representative control mechanisms implemented on real-world IEDs to demonstrate the feasibility of implementation.

6.1 Control Logic Implemented on IED

In this section, we discuss automated control mechanisms that are typically implemented on IEDs. In practice, IEDs listens to IEC 61850 GOOSE messages sent by peer IEDs and SV messages sent by MUs (merging units) and execute control logic for automated control. Among such mechanisms, we focus on protection mechanisms because of popularity and importance. While we demonstrate only a small number of logics, we note that, by using a tool like *Matiec* [13], we can convert logic written according to IEC 61131-3 standard into

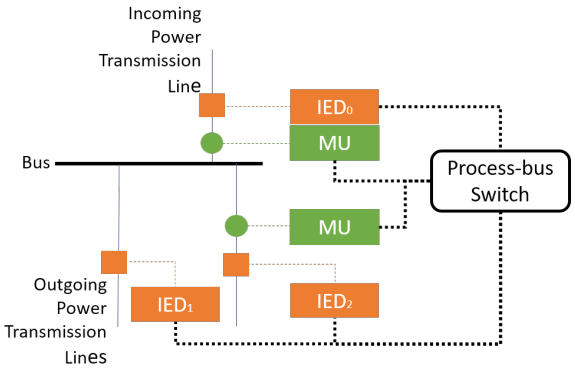

Figure 4: Cyber and Physical Topology in Substation

C language so that existing logic codes on a base IED can be feasibly ported into DecIED, and thus our design and implementation are extensible.

Figure 4 shows system topology around one of the buses in a substation, which is drawn based on a setup discussed in [25]. As can be seen, IEDs are connected to a circuit breaker, or CB, (a rectangle on each power transmission line) and MUs are also connected to the transmission lines to collect measurements. Both IEDs are MUs are connected to a switch on process bus of the substation. The breaker is closed when there is no fault and is opened to cut down the power flow on a transmission line when an overcurrent situation is detected to protect the line as well as connected equipment. Besides, IEDs also implement a backup protection mechanism that can be triggered when a certain circuit breaker associated with another IED in the substation fails to trip.

Some popular protection mechanisms, namely *busbar protection* and *breaker failure protection* are discussed in [25] along with hypothetical but realistic substation topology. Below, we summarize how these work according to the setting shown in Figure 4 so that we can implement them on our proof-of-concept virtual IED implementation.

Busbar protection: This protection aims at preventing fault situation on a bus in a substation. The measurement of current is based on the messages by SV protocol. If the SV messages indicate anomalous current values, IED_0 quickly realises the fault and triggers a trip of its own circuit breaker as well as sends notification for other relevant IEDs (i.e., IED_1 and IED_2 in Figure 4) to trip by using GOOSE communication.

Breaker failure protection: There may be a situation that the overcurrent occurs but the breaker does not trip because of mechanical failure or any other reasons. When one IED (e.g., IED_0) picks up occurrence of overcurrent and attempts to open its own circuit breaker, but then it detects a failure preventing the CB from opening. Once such an event occurs, the corresponding IED to send out a GOOSE message for nearby IEDs (i.e., IED_1 and IED_2) to let them know the situation, which will then trigger those IEDs to trip their CBs.

We should emphasize that the logic implemented on our prototype can be implemented on multiple (real) IEDs. For instance, breaker failure protection logic is implemented on all IEDs that control circuit breakers. Thus, our prototype can be readily customized for providing deception for the real IEDs of this sort.

6.2 Virtual IED Implementation Details

The MMS server, GOOSE publisher, and GOOSE/SV subscriber modules utilize the libIEC61850 open-source C library [8]. These modules utilize an SCL file, which is prepared based on the configuration discussed in [25], to generate data models used for parsing as well as constructing IEC 61850 messages. The virtual IED is running with four threads: GOOSE Publisher, GOOSE Subscriber, SV Subscriber, and IED logic. In addition, there is a buffer to record real-time values from incoming GOOSE and SV messages. For the data attributes in measurement, due to the values of current and voltage are in sinusoidal function, the coming data values in SV messages are recorded in a certain interval (e.g., duration corresponding to 1 cycle), and the effective values are calculated by all the SV values in the period. The initial values are defined according to the normal status of the real device.

Table 1: The data attributes of GOOSE messages defined by the SCL file of IED. (FC: function constraint [65], i32: 32bits integer. The values shown are for the normal state.)

Dataset	NAME	FC	TYPE	VALUE
IED-CTRL/Status	PTRC.EEHealth.stVal	ST	i32	1
	XCBR.Loc.stVal	ST	bool	false
	XCBR.Pos.stVal	ST	i32	1
	XSWI.Pos.stVal	ST	i32	1
IED-PROT/Alarm	LPHD.PwrSupAlm.stVal	ST	bool	false
	PIOC.Op.general	ST	bool	false
	PSCH.ProRx.stVal	ST	bool	false
	PSCH.ProTx.stVal	ST	bool	false
	XCBR.EEHealth.stVal	ST	i32	1

Table 1 shows the GOOSE data attributes defined by the SCL file generated based on [25]. We have another data set to announce measurements (IED-MEAS/Meas), but it is omitted for the interest of space. In sum, there are three groups of data attributes, including status (control), alarm (protection), and measurement, which are sent in separate GOOSE messages. The detailed explanations of the data attributes are introduced in IEC61850-7-2 [1], 7-3 [3], and 7-4 [2].

For the protection mechanisms on IEDs introduced in Section 6.1, here let us elaborate on the implementation of busbar protection mechanism. The virtual IED program firstly gets the real-time SV values about the current via SV Subscriber module. If the real-time effective value is larger than the pre-configured threshold, it is treated as an overcurrent situation. Then, virtual IED "pretends" to open circuit breaker, and updates the data attribute about the circuit breaker status. The data attribute "XCBR.Pos.stVal" is associated with the position of the breaker, which is updated to 0 (open), and "PIOC.Op.general" is used for the status of instantaneous overcurrent, which is updated to true to alarm the status. These updates will be handled by GOOSE Publisher for the announcement to other IEDs. The implementation results of the protection scenarios will be introduced in Section 7.2.

Regarding the circuit breaker failure protection case, IED Logic module checks the position of the circuit breaker. If the circuit breaker remains closed, a breaker failure protection is triggered to give the alarm. The virtual IED sends out the breaker failure message to the other IEDs, and updates the related data attributes about the external equipment health of the breaker ("XCBR.EEHealth.stVal" and "PTRC.EEHealth.stVal") by changing the value from 1 (normal state) to 3 (alarm).

7 EVALUATION

This section evaluates the DecIED prototype based on the design goals in Section 4.

7.1 Similarity in Device Characteristics

In order to evaluate the similarity in device characteristics on the cyber side, we utilized a popular network scanning tool, Nmap [15]. The scanning method used was to probe all the TCP and UDP ports. Then the discrepancies of the OS fingerprints of the DecIED and a real IED in EPIC testbed [6, 59] (i.e., base IED) are compared.

Some earlier approaches for implementing deception technologies for industrial control systems, such as [22, 46], utilized Mininet [14] to implement virtual devices. Thus, we also obtained OS fingerprints of the Mininet-based implementation that are used in the literature [46]. DecIED as well as Mininet-based implementation (called Mininet IED hereafter) are set up on a separate virtual machine running Ubuntu Linux OS.

Regarding the open ports, since we configured DecIED based on a Siemens IED used in the EPIC testbed, no difference was seen, and both opens port 80 and 102. Besides, since we customized HTTP headers on the web server in DecIED according to the base IED, no difference was observed.

Table 2 shows Nmap's OS fingerprinting results of a DecIED, the base IED, and Mininet IED. As can be seen, compared to Mininet IED, DecIED presents significantly better similarity. Moreover, while Mininet IED was fingerprinted as "Linux 3.2-4.9", the DecIED and the base IED returned "No exact OS matches". Since virtual nodes on Mininet are essentially copy of the host OS, and thus OS fingerprint revealed the information of the host OS (i.e., Ubuntu Linux).

For both DecID and Mininet IED, differences from the base IED are highlighted in the table, and we immediately see the limitation of Mininet IED. There are some differences seen in the SEQ lines in DecIED fingerprint. Among these, SP and ISR fluctuate ±2 based on our repeated experiments. Therefore, from the results it can be concluded that the differences between these two values do not affect the fingerprint resistance of the system. Regarding the other differences, namely SS and IPL, according to Nmap documentation [15], the former indicates whether TCP and ICMP shares IP ID sequence or not and the latter is based on the total length of an IP packet used for a port unreachable ICMP message when UDP packet is sent to a closed port. Other smart grid devices, such as WAGO PLC used in EPIC testbed [59], returned IPL=164, which is the same as DecIED. Thus, without knowing the exact characteristics of the specific IED model used in the system, it is not feasible for an attacker to tell differences between real and fake devices. Moreover, collection of IPL requires active fingerprinting, repetition of which can be caught by other security measures, such as intrusion detection systems. Having that said, we can radically address these differences by modifying socket implementation, which will be left for our future work.

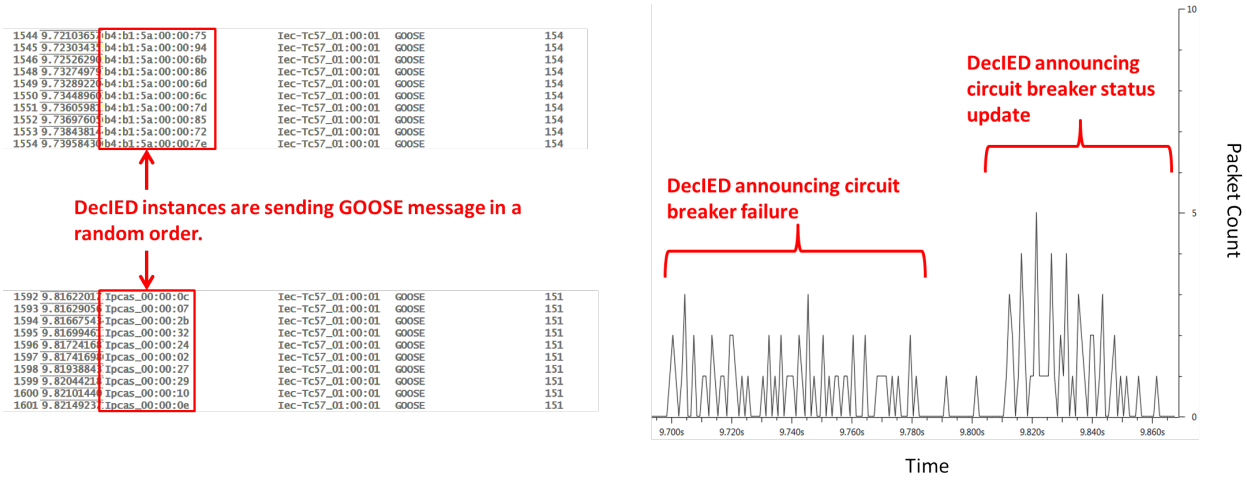

Figure 5: IEC 61850 GOOSE communication pattern of DecIED (50 instances) under circuit breaker failure situation

Table 2: Fingerprinting results of DecIED, Base IED, and Mininet IED

IED system	Fingerprint
DecIED	SEQ(SP=C5 %GCD=1% ISR=D5 %TI=I%CI=I%II=I% SS=O %TS=U) ECN(R=N) T1(R=Y%DF=N%T=72%S=O%A=S+%F=AS%RD=0%Q=) T2(R=N) T3(R=N) T4(R=Y%DF=N%T=72%W=0%S=A%A=Z%F=R%O=%RD=0%Q=) T5(R=Y%DF=N%T=72%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=) T6(R=Y%DF=N%T=72%W=0%S=A%A=Z%F=R%O=%RD=0%Q=) T7(R=Y%DF=N%T=72%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=) U1(R=Y%DF=N%T=72% IPL=164 %UN=0%RIPL=G%RID=G%RIPCK=G% RUCK=6339%RUD=I) IE(R=Y%DFI=N%T=72%CD=Z)
Base IED	SEQ(SP=CD%GCD=1%ISR=D6%TI=I%CI=I%II=I%TS=U) ECN(R=N) T1(R=Y%DF=N%T=72%S=O%A=S+%F=AS%RD=0%Q=) T2(R=N) T3(R=N) T4(R=Y%DF=N%T=72%W=0%S=A%A=Z%F=R%O=%RD=0%Q=) T5(R=Y%DF=N%T=72%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=) T6(R=Y%DF=N%T=72%W=0%S=A%A=Z%F=R%O=%RD=0%Q=) T7(R=Y%DF=N%T=72%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=) U1(R=Y%DF=N%T=72%IPL=240%UN=0%RIPL=G%RID=G%RIPCK=G% RUCK=6339%RUD=I) IE(R=Y%DFI=N%T=72%CD=Z)
Mininet IED	SEQ(SP=106 %GCD=1% ISR=10E%TI=Z %CI=I%II=I% TS=8) OPS(O1=M5B4ST11NW9%O2=M5B4ST11NW9%O3=M5B4NNT11NW9% O4=M5B4ST11NW9%O5=M5B4ST11NW9%O6=M5B4ST11) WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120) ECN(R=Y%DF=Y%T=40%W=7210%O=M5B4NNSNW9%CC=Y%Q=) T1(R=Y% DF=Y%T=40 %S=O%A=S+%F=AS%RD=0%Q=) T2(R=N) T3(R=N) T4(R=Y% DF=Y%T=40 %W=0%S=A%A=Z%F=R%O=%RD=0%Q=) T5(R=Y% DF=Y%T=40 %W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=) T6(R=Y% DF=Y%T=40 %W=0%S=A%A=Z%F=R%O=%RD=0%Q=) T7(R=Y% DF=Y%T=40 %W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=) U1(R=Y% DF=N%T=40%IPL=164 %UN=0%RIPL=G%RID=G%RIPCK=G% RUCK=G%RUD=G) IE(R=Y%DFI=N% T=40%CD=S)

7.2 Indistinguishability in Communication Models

In this section we present the network traffic generated by our DecIED prototype as a preliminary evaluation. According to the specification (see also Section 3), DecIED generates periodic GOOSE traffic when there is no event or status change. For each period, DecIED sends out 3 GOOSE messages corresponding to the 3 datasets discussed in Table 1. Measurement values are calculated by SV traffic generated by an emulated MU.

Let us next see the behavior of DecIED under a disturbance scenario, namely circuit breaker failure case. In order to emulate an overcurrent situation, we configured MU to send out SV messages conveying overcurrent measurement, and also run two sets of DecIED instances, corresponding to IED_0 and IED_1. As seen in Figure 5, after overcurrent is picked up, DecIED instances imitating IED_0 start sending GOOSE messages to report circuit breaker failure (around time 9.7s). The figure is showing the network trace as well as the number of GOOSE messages for each 0.001s time slot. As can be seen, DecIED instances are sending GOOSE messages in a random order. The time duration from the first to the last DecIED on average was 0.067s in the case of 50 instances. When we run 200 instances, the duration becomes 0.51s on average. This may be caused by the limitation in the number of concurrent processes as well as the single NIC (network interface controller). However, the IED's operation typically involves interaction with physical systems, and actuation latency will be involved. For instance, to open/close circuit breaker, based on our observation in the testbed, it takes in the order of second, which outweighs the aforementioned timing difference.

Because we were not able to conduct experiments emulating the breaker failure scenario on EPIC testbed, we cannot compare DecIED with the real IED deployed there. However, one measurement in a similar setting was found in [63], which showed that the latency of between the receipt of alarm and announcement of circuit breaker status update measured on a real, commercial IED was 783.385ms on average with standard deviation 125.277ms. By configuring artificial delay on DecIED, we can overlap this duration with the real IED to make them indistinguishable. As future work, we plan to conduct rigorous evaluations in terms of similarity in communication patterns and behaviors, once we establish our own testbed for measurements.

7.3 Scalability

In order to evaluate the scalability of DecIED, we conducted experiments to run a different number of DecIED instances on an industrial with varying resource configuration. Our evaluation here is based on whether DecIED instances can generate the expected number of periodic GOOSE messages (e.g., 3 GOOSE messages for each interval per instance). If the number of observed GOOSE messages is significantly below the expected number, we conclude that the number of DecIED instances exceeds the capacity.

We have conducted a series of experiments with different numbers of DecIED instances for 6 minutes with various periodic GOOSE messaging interval (5, 3, and 1 second(s)). 5-second interval is the default interval used in libIEC61850 [8] while 3-second interval is found in EPIC testbed [6]. All the GOOSE messages are captured by Wireshark, and the numbers of messages observed in each second are recorded. We ran all experiments on 3 industrial PCs: "high"-end one with Intel Core i7-7700 and 32GB RAM, "middle"-end one with Intel Core i7-3610QE with 16GB RAM, and "low"-end one with Intel Celeron J1900 with 4GB RAM. The results are summarized in Figure 6. In the figure, solid and dashed lines represent ethe xpected number of GOOSE messages per second, and in order to judge the scalability, we evaluated whether DecIED instances are catching up with this expected rate. For instance, in the case of 1-second interval, high-end and middle-end ones can support up to 200 instances while with 5-second interval, they support up to 500 instances and even the low-end one can support 200 instances. The max number of instances supported for each setting as well as resource consumption are summarized in Table 3.

Combined with the observation in Section 7.2, we claim that at least 200 DecIED instances can be run on a single industrial PC without losing indistinguishability in the communication models, which meets our design goal.

Table 3: Summary of Scalability Evaluation

Time Interval(s)	PC Grade	Number of DecIEDs	CPU Usage	Memory Usage
1	"high"	200	61%	0.3226%
	"medium"	200	99.1%	1.3548%
	"low"	50	99.4%	2.7027%
3	"high"	450	66%	1.5974%
	"medium"	350	99.4%	2.3226%
	"low"	50	99.8%	3.2432%
5	"high"	500	70%	1.631%
	"medium"	500	98.9%	3.1613%
	"low"	100	99%	5.6757%

7.4 Security Discussion

Our goal in this paper is to practically implement deception IEDs that offer "k-anonymous smokescreen" to deceive such attackers. If we configure 20-anonymity (e.g., by deploying 200 DecIED instances for EPIC testbed [6]), attackers in the substation system (either at station bus or process bus) will see 20 devices that look and behave indistinguishably. Thus, passive attackers cannot distinguish which of the 20 devices is the real IED and cannot learn which one to target when he mounts an attack against the infrastructure. An attacker may send out innocuous control commands for probing. When an attacker sends an IEC 61850 MMS commend randomly (and sequentially to fly under the radar), the success probability for him is $\frac{1}{k}$. In other words, the probability that the first attempt is flagged by our deception appliance is $1 - \frac{1}{k}$, which is translated, in the case of 20-anonymity setup, to 95%.

It might be argued that a committed attacker would send the same malicious commands (in particular, control commands) to all of them, including a base IED and DecIED instances. If all commands are sent out at the same time, the malicious control command would be executed before being reported by DecIED. However, such a traffic pattern (i.e., simultaneously sending the same commands to multiple DecIED instances) can be detected or blocked by intrusion detection/prevention systems. If the commands are sent sequentially, DecIED still can report the incident before the base IED is affected by the attacker's command as discussed earlier, by working with, for instance, intrusion prevention systems. We note that DecIED is not intended to be a standalone, self-contained security solution. The primary purpose of DecIED is enhancing capability to capture activities by attackers in pre-attack phase, and, DecIED can increase the probability of detection.

As discussed in Section 5, by overhearing broadcast process-bus communication (IEC 61850 GOOSE and SV), DecIED shares power grid status and measurements with the real IED, and thus, DecIED can generate the traffic like the real IED in both station bus (IEC 61850 MMS) and process bus (IEC 61850 GOOSE). Given that our attacker model does not exclude attackers who can access to the process-bus network, let us discuss the case where attackers may inject maliciously-crafted messages there. Even when an attacker at process bus would inject fake GOOSE or SV messages, because DecIED is designed to process the message according to the same logic as the real IED, the impact of such messages is the same on both, which thus makes them indistinguishable. Note that detection or prevention of malicious message injection itself is not the goal of the deception technology. It is possible to implement some sort of bad data detection mechanism on the DecIED appliance, and it is part of our future work.

Regarding the countermeasures against device/OS fingerprinting, there are other types of techniques, such as ones based on latency between a certain request and response [31] and clock skew [36]. We admit that Honeyd is not the universal solution to counter them. For instance, to counter latency-based fingerprinting, we need to carefully measure and model the latency of the real device, and the countermeasure for the latter may require radical implementation change. Evaluation with such techniques and implementation of the evading mechanism is part of our future work.

In our prototype implementation, we focused on popular logic related to circuit breaker control. There should be other logic associated with other types of power grid physical components, and studying feasibility to implement such logic is part of our future work. However, circuit breakers are one of the most critical power grid components that significantly influence power grid stability, and it is usually the first priority for attackers, as was the case in Ukraine power plant attack [66]. Thus, our proof-of-concept focused on circuit breaker control still demonstrates the feasibility to implement additional lines of defence of real-world importance. Validation of a real power grid system under various attack scenarios is planned in our future work.

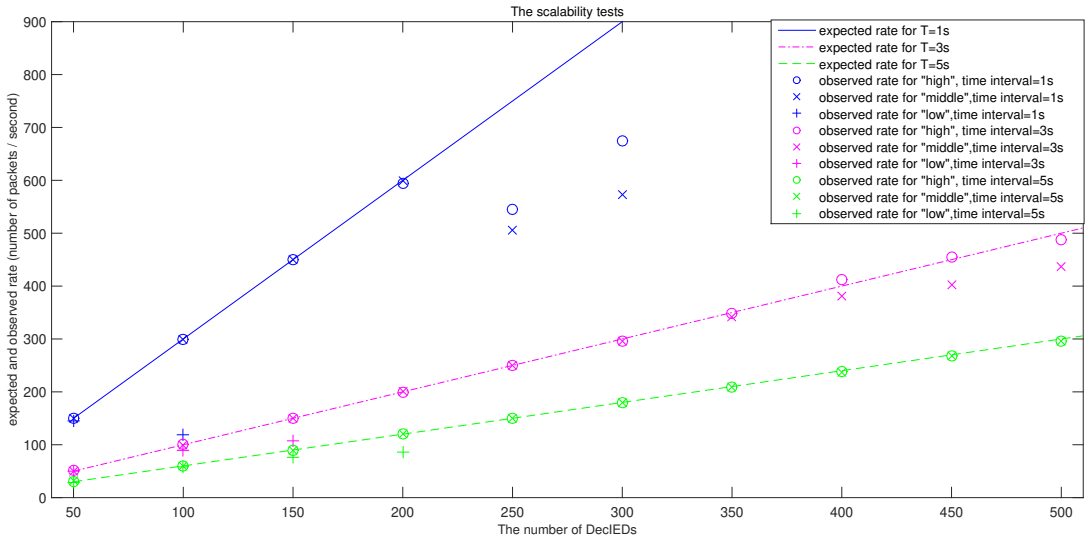

Figure 6: Scalability evaluation with different GOOSE messaging interval (T) in different industrial PCs

8 CONCLUSIONS

In this paper, we discussed the practical design and implementation of deception technologies for IEC61850-compliant smart grid systems, called DecIED. DecIED can imitate externally-visible characteristics of the real IED in a modernized electrical substation to implement k-anonymous smokescreen. By deploying sufficient numbers of DecIED instances, we can make it difficult for persistent attackers in the attack-preparation phase to learn the actual system configuration and topology correctly. Furthermore, once attackers attempt to touch any of DecIED instances, an alarm is triggered to alert system operators and/or to invoke intrusion prevention functionality. This way, DecIED helps to detect probing activities and lateral movement by persistent attackers or malware. Our proof-of-concept implementation, which we plan to publish as an open-source project, demonstrated feasibility, indistinguishability, and scalability when deployed on commodity industrial PCs.

As future work, we plan to conduct the evaluation in a practical environment, ideally in a real power grid system. For instance, we will implement the same control logic implemented on the IEDs in a real system and compare the behavior of DecIED in a quantitative manner. Another evaluation strategy is deploying DecIED in a hacking/capture-the-flag competition to evaluate difficulty to distinguish it from a real IED. Evaluating capability and effectiveness of DecIED under various attack models and exploring integration with other security measures, such as intrusion detection systems, would be also interesting directions.

ACKNOWLEDGEMENT

This work is supported in part by the National Research Foundation, Prime Minister's Office, Singapore under the Energy Programme and administrated by the Energy Market Authority (EP Award No. NRF2017EWT-EP003-047) and is partly supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. Jianying Zhou's work is supported by the SUTD start-up research grant SRG-ISTD-2017-124.

REFERENCES

[1] 2003. International Standard IEC 61850-7-2 Communication networks and systems in substations - Part 7-2: Basic communication structure for substation and feeder equipment - Abstract communication service interface (ACSI).
[2] 2003. International Standard IEC 61850-7-4 Communication networks and systems in substations - Part 7-4: Basic communication structure for substation and feeder equipment - Compatible logical node classes and data classes.
[3] 2010. International Standard IEC 61850-7-3 Communication networks and systems for power utility automation - Part 7-3: Basic communication structure - Common data classes.
[4] 2011. IEC 61850 Communication protocol manual. https://www.naic.edu/~phil/hardware/sitePower/evd4/1MRK511242-UEN_-_en_Communication_protocol_manual__IEC_61850__650_series__IEC.pdf.
[5] 2018. IEC 62351:2018 SER Series. https://webstore.iec.ch/publication/6912
[6] 2018. Electric Power and Intelligent Control (EPIC) Testbed. [Online]. Available: https://itrust.sutd.edu.sg/wp-content/uploads/sites/3/2019/02/EPIC_technical_details-231018-v1.2.pdf. (Date last accessed on Feb. 12, 2019).
[7] 2019. IEC 61850 - Communication Networks and Systems in Substations. https://webstore.iec.ch/
[8] 2019. libIEC61850: open source libraries for IEC 61850 . https://libiec61850.com/libiec61850/new-version-1-3-3-of-libiec61850/.
[9] 2019. Revolutionary Deception Technologies. https://cybertrap.com/.
[10] 2020. CONPOT ICS/SCADA Honeypot. http://conpot.org.
[11] 2020. 'Crash Override': The Malware That Took Down a Power Grid. [Online]. Available: https://www.wired.com/story/crash-override-malware/.
[12] 2020. Digital Bond. http://www.digitalbond.com/tools/scada-honeynet.
[13] 2020. Matied. https://directory.fsf.org/wiki/Matiec.
[14] 2020. Mininet. http://mininet.org/.
[15] 2020. Nmap: the Network Mapper. https://nmap.org/.
[16] 2020. Shodan. https://www.shodan.io/.
[17] 2020. ThreatDefend Platform. https://attivonetworks.com/product/deception-technology/.
[18] 2020. What is Stuxnet? https://www.mcafee.com/enterprise/en-sg/security-awareness/ransomware/what-is-stuxnet.html.
[19] 2020. Wireshark. https://www.wireshark.org/.
[20] Sridhar Adepu and Aditya Mathur. 2018. Assessing the effectiveness of attack detection at a hackfest on industrial control systems. IEEE Transactions on Sustainable Computing (2018).
[21] Ehab Al-Shaer, Jinpeng Wei, Kevin W. Hamlen, and Cliff Wang. 2019. CONCEAL: A Strategy Composition for Resilient Cyber Deception: Framework, Metrics, and Deployment. Springer International Publishing, Cham, 101–124. https://doi.org/10.1007/978-3-030-02110-8_6

[22] Daniele Antonioli, Anand Agrawal, and Nils Ole Tippenhauer. 2016. Towards high-interaction virtual ICS honeypots-in-a-box. In Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy. ACM, 13–22.

[23] Nadarajah Asokan, Ferdinand Brasser, Ahmad Ibrahim, Ahmad-Reza Sadeghi, Matthias Schunter, Gene Tsudik, and Christian Wachsmann. 2015. Seda: Scalable embedded device attestation. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 964–975.

[24] BBC News. 2018. Russian hackers penetrate US power stations. https://www.bbc.com/news/technology-44937787 (Date last accessed on Sep. 22, 2019).

[25] Partha P Biswas, Heng Chuan Tan, Qingbo Zhu, Yuan Li, Daisuke Mashima, and Binbin Chen. 2019. A Synthesized Dataset for Cybersecurity Study of IEC 61850 based Substation. In 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). IEEE, 1–7.

[26] Boyd Brown. 2020. Deception as a security strategy. https://trapx.com/whitepapers A whitepaper by TrapX Security, Inc.

[27] Dániel István Buza, Ferenc Juhász, György Miru, Márk Félegyházi, and Tamás Holczer. 2014. CryPLH: Protecting smart energy systems from targeted attacks with a PLC honeypot. In International Workshop on Smart Grid Security. Springer, 181–192.

[28] John Henry Castellanos and Jianying Zhou. 2019. A Modular Hybrid Learning Approach for Black-Box Security Testing of CPS. In Applied Cryptography and Network Security, Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung (Eds.). Springer International Publishing, Cham, 196–216.

[29] Binbin Chen, Xinshu Dong, Guangdong Bai, Sumeet Jauhar, and Yueqiang Cheng. 2017. Secure and efficient software-based attestation for industrial control devices with arm processors. In Proceedings of the 33rd Annual Computer Security Applications Conference. ACM, 425–436.

[30] Shaik Mullapathi Farooq, SM Suhail Hussain, and Taha Selim Ustun. 2019. Performance Evaluation and Analysis of IEC 62351-6 Probabilistic Signature Scheme for Securing GOOSE Messages. IEEE Access 7 (2019), 32343–32351.

[31] David Formby, Preethi Srinivasan, Andrew M. Leonard, Jonathan D. Rogers, and Raheem A. Beyah. 2016. Who's in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems. In 23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016. The Internet Society. https://pdfs.semanticscholar.org/d160/c46512ebc12c172d26f150797b42592a9095.pdf

[32] Hamid Reza Ghaeini, Matthew Chan, Raad Bahmani, Ferdinand Brasser, Luis Garcia, Jianying Zhou, Ahmad-Reza Sadeghi, Nils Ole Tippenhauer, and Saman Zonouz. 2019. PAtt: Physics-based Attestation of Control Systems. In 22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019). USENIX Association, Chaoyang District, Beijing, 165–180. https://www.usenix.org/conference/raid2019/presentation/ghaeini

[33] Andy Greenberg. 2019. The Highly Dangerous 'Triton' Hackers Have Probed the US Grid. https://www.wired.com/story/triton-hackers-scan-us-power-grid/ (Date last accessed on Sep. 22, 2019).

[34] IEC TC57. 2015. IEC 61850-90-2 TR: Communication networks and systems for power utility automation – Part 90-2: Using IEC 61850 for the communication between substations and control centres. International Electro technical Commission Std (2015).

[35] IEEE Power and Energy Society. 2005. IEEE Standard Communication Delivery Time Performance Requirements for Electric Power Substation Automation. (2005).

[36] Tadayoshi Kohno, Andre Broido, and Kimberly C Claffy. 2005. Remote physical device fingerprinting. IEEE Transactions on Dependable and Secure Computing 2, 2 (2005), 93–108.

[37] Kamil Kołtyś and Robert Gajewski. 2015. Shape: A honeypot for electric power substation. Journal of Telecommunications and Information Technology 4 (2015), 37–43.

[38] Jakub W Konka, Colin M Arthur, Francisco J Garcia, and Robert C Atkinson. 2011. Traffic generation of IEC 61850 sampled values. https://ieeexplore.ieee.org/abstract/document/6089025. In 2011 IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS). IEEE, 43–48.

[39] Carl Kriger, Shaheen Behardien, and John-Charly Retonda-Modiya. 2013. A detailed analysis of the GOOSE message structure in an IEC 61850 standard-based substation automation system. International Journal of Computers Communications & Control 8, 5 (2013), 708–721.

[40] Subhash Lakshminarayana, E Veronica Belmega, and H Vincent Poor. 2019. Moving-Target Defense for Detecting Coordinated Cyber-Physical Attacks in Power Grids. arXiv preprint arXiv:1908.02392 (2019).

[41] Hui Lin, Zbigniew Kalbarczyk, and Ravishankar K Iyer. 2018. RAINCOAT: RAndomization of Network Communication in Power Grid Cyber INfrastructure to Mislead Attackers. IEEE Transactions on Smart Grid (2018).

[42] Hui Lin, Adam Slagell, Catello Di Martino, Zbigniew Kalbarczyk, and Ravishankar K Iyer. 2013. Adapting bro into scada: building a specification-based intrusion detection system for the dnp3 protocol. In Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop. ACM, 5.

[43] Hui Lin, Jianing Zhuang, Yih-Chun Hu, and Huayu Zhou. 2020. DefRec: Establishing Physical FunctionVirtualization to Disrupt Reconnaissance of Power-Grids' Cyber-Physical Infrastructures. In The Proceedings of 2020 Network and Distributed System Security Symposium (NDSS).

[44] Ralph E Mackiewicz. 2006. Overview of IEC 61850 and Benefits. In 2006 IEEE Power Engineering Society General Meeting. IEEE, 8–pp.

[45] Yuval Malachi. 2020. Kaspersky Labs hacked - Deception technology could help - TrapX Security. https://trapx.com/kaspersky-labs-hacked-deception-technology-could-help Posted by Yuval Malachi, CTO of TrapX Security, Inc.

[46] Daisuke Mashima, Binbin Chen, Prageeth Gunathilaka, and Edwin Lesmana Tjiong. 2017. Towards a grid-wide, high-fidelity electrical substation honeynet. In 2017 IEEE International Conference on Smart Grid Communications (SmartGridComm). IEEE, 89–95.

[47] Daisuke Mashima, Prageeth Gunathilaka, and Binbin Chen. 2019. Artificial Command Delaying for Secure Substation Remote Control: Design and Implementation. https://doi.org/10.1109/TSG.2017.2744802. , 471-482 pages.

[48] Daisuke Mashima, Derek Kok, Wei Lin, Muhammad Hazwan, and Alvin Cheng. 2020. On Design and Enhancement of Smart Grid Honeypot System for Practical Collection of Threat Intelligence. In 13th USENIX Workshop on Cyber Security Experimentation and Test.

[49] Daisuke Mashima, Ramkumar Rajendran, Toby Zhou, Binbin Chen, and Biplab Sikdar. 2019. On Optimization of Command-Delaying for Advanced Command Authentication in Smart Grid Systems. In Proc. of IEEE PES ISGT Asia 2019. IEEE.

[50] Kieran McLaughlin. 2015. High-level design documentation and deployment architecture for Multi-Attribute SCADA Intrusion Detection System. https://project-sparks.eu/wp-content/uploads/2014/04/SPARKS_D4_1_Multi-Attribute_SCADA_Intrusion_Detection_System.pdf (Date last accessed on Jun. 7, 2017).

[51] Ariana Mirian, Zane Ma, David Adrian, Matthew Tischer, Thasphon Chuenchujit, Tim Yardley, Robin Berthier, Joshua Mason, Zakir Durumeric, J Alex Halderman, et al. 2016. An Internet-Wide View of ICS Devices. In 14th IEEE Privacy, Security, and Trust Conference (PST'16).

[52] Kapuge Kariyawasam Mudalige and Sachintha Kariyawasam. 2016. Implementation of an IEC 61850 Sampled Values Based Line Protection IED with a New Transients-Based Hybrid Protection Algorithm. http://hdl.handle.net/1993/31306. (2016).

[53] Venkat Pothamsetty and Matthew Franz. 2005. SCADA HoneyNet Project: Building Honeypots for Industrial Networks. http://scadahoneynet.sourceforge.net/.

[54] Niels Provos. 2003. Honeyd-a virtual honeypot daemon. In 10th DFN-CERT Workshop, Hamburg, Germany, Vol. 2. 4.

[55] Muhammad Talha Abdul Rashid, Salman Yussof, and Yunus Yusoff. 2016. Trust system architecture for securing GOOSE communication in IEC 61850 substation network. https://doi.org/10.14257/ijsia.2016.10.4.27. International Journal of Security and Its Applications 10, 4 (2016), 289–302.

[56] Owen Redwood, Joshua Lawrence, and Mike Burmester. 2015. A symbolic honeynet framework for scada system threat intelligence. In International Conference on Critical Infrastructure Protection. Springer, 103–118.

[57] Wenyu Ren, Timothy Yardley, and Klara Nahrstedt. 2018. EDMAND: Edge-Based Multi-Level Anomaly Detection for SCADA Networks. In 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). IEEE, 1–7.

[58] Electricity Information Sharing and Analysis Center (E-ISAC). 2016. Analysis of the cyber attack on the Ukrainian power grid. (2016).

[59] Ahnaf Siddiqi, Nils Ole Tippenhauer, Daisuke Mashima, and Binbin Chen. 2018. On Practical Threat Scenario Testing in an Electric Power ICS Testbed. In Proceedings of the Cyber-Physical System Security Workshop (CPSS), co-located with ASIACCS. https://doi.org/10.1145/3198458.3198461

[60] Jianhua Sun and Kun Sun. 2016. DESIR: Decoy-enhanced seamless IP randomization. In IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. IEEE, 1–9.

[61] Heng Chuan Tan, Carmen Cheh, Binbin Chen, and Daisuke Mashima. 2019. Tabulating Cybersecurity Solutions for Substations: Towards Pragmatic Design and Planning. In Proceedings of IEEE PES ISGT Asia 2019. IEEE.

[62] Robert Udd, Mikael Asplund, Simin Nadjm-Tehrani, Mehrdad Kazemtabrizi, and Mathias Ekstedt. 2016. Exploiting bro for intrusion detection in a SCADA system. In Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security. ACM, 44–51.

[63] Noriyuki Ueda. 2019. Prototyping of Substation Automation System Testbeds for Cyber Security Evaluation. In CIGRE 2019. 103–118.

[64] Craig Wester, Mark Adamiak, and J Vico. 2011. Practical Applications of IEC 61850 Protocol in Industrial Facilities. IAS, Orlando, FL (2011), 1–2.

[65] Yubo Yuan and Yi Yang. 2019. IEC 61850-Based Smart Substations: Principles, Testing, Operation and Maintenance. Elsevier Science. https://books.google.com.sg/books?id=ji6dDwAAQBAJ

[66] Kim Zetter. 2016. Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid. [Online]. Available: http://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/. (Date last accessed on Jun. 7, 2017).