Artificial Command Delaying for Secure Substation Remote Control: Design and Implementation

Daisuke Mashima, Prageeth Gunathilaka, and Binbin Chen
Advanced Digital Sciences Center, Singapore
{daisuke.m, prageeth.g, binbin.chen}@adsc.com.sg

***Abstract*—Electrical substations play a crucial role in power grids. A number of international standards, such as IEC 60870 and 61850, have emerged to modernize substations for efficient and timely control. However, owing to insufficient security consideration and implementation, the digitization of a large number of connected substations could dramatically increase the scale of damage on power grids caused by cyber attacks. In this paper, we discuss the practical design, implementation, and deployment of *active command mediation defense (A*CMD)*, a distributed cybersecurity solution to counter attacks injecting malicious remote control commands. A*CMD takes advantage of artificial command-delaying to realize an additional layer of security for each substation in an autonomous, decentralized manner. In particular, for grid operators to make appropriate delay configuration, the procedure to find tolerable delay for a power grid model of interest is formulated and demonstrated with multiple power grid models of different sizes. We further show practical deployment options of A*CMD along with proof-of-concept implementations, whose performance and stability are evaluated with a software-based smart-grid testbed.**

***Index Terms*—Cyber-physical systems security, substation remote control, artificial delay, IEC 60870-5-104, IEC 61850.**

I. Introduction

Reliable delivery of electricity is imperative to keep our lights on. Electricity from generators needs to be transformed through different voltage levels before reaching end consumers. Such operations are usually carried out in substations. Besides, substations enable intelligent switching operations to facilitate automatic fault response and optimize power grid operations. Depending on the size of service, there can be hundreds or thousands of substations. For example, in Singapore, there are over 10,000 transmission/distribution substations under a single utility company. Furthermore, within each substation, there are many kinds of physical components, such as transformers, circuit breakers, shunt reactors, etc. Power industry has put significant efforts to effectively manage substations. In particular, standard technologies, including IEC 60870 and 61850, have been established for substation remote control and automation, and the number of substations that employ these technologies has increased significantly over the years. However, coming with such modernization is the increased risk of cyber attacks that could subvert power grid services. A major incident in this category recently occurred in Ukraine [1], [2], where the control center system was hacked and manipulated to emit a number of circuit breaker open commands, making 30 substations off-line for hours. Very recently, *CrashOverride* malware, which abuses smart grid communication infrastructure using IEC 60870-5-104 and IEC 61850 to manipulate physical power grid systems was reported in 2017, and additional layers of defense are highly demanded to counter such threats [3].

To address the security risk associated with substation remote control, efforts in two directions are required. One is to minimize the possibility of successful attack. The other is to mitigate the impact of attack. In the former direction, a number of cybersecurity measures, such as firewalls and intrusion detection systems that support industrial control protocols ([4], [5], [6], [7]), as well as general cybersecurity technologies for securing enterprise systems, including user authentication and access control systems, are developed. Nevertheless, effective deployment and operation of them remain a major challenge in the real world. The Ukraine incident demonstrated not only the difficulty to eliminate the possibility of such attacks but also inability of mitigation once those security measures are circumvented, which led us to the research in the latter aspect. Specifically, our approach (with the initial idea first proposed in our recent work [8]) is to introduce an additional layer of security in each substation that takes advantage of artificial time delay, upon its handling of incoming remote control commands, for mitigating physical impacts caused by cyber attacks. We call it *active command mediation defense (A*CMD)*. As its name suggested, this layer actively mediates the execution of commands sent to the substations to manage the risk associated with substations' remote control interface.

In this paper, we further develop the approach in [8] and investigate several key issues regarding the A*CMD design to enable practical, real-world implementation and deployment of A*CMD. Our main contributions include:

- We study the impact of introducing artificial delay on typical substation remote control use cases, by means of simulation using a variety of grid configurations, to learn how much delay can be taken advantage of without affecting legitimate remote control operations. We further formulate the procedure for grid operators to systematically find such tolerable delay of a power grid model of interest.
- We present two practical deployment options of the proposed solution and a proof-of-concept implementation on Raspberry Pi platform, which has similar spec and hardware to actual smart grid devices. We also evaluate the performance of our prototype using SoftGrid testbed [9] and compare it with a commercial product.

Through these contributions, this paper demonstrates the procedure for grid operators to find acceptable amount of delay by using simulation with publicly-available power grid models, and shows that, for these power grids, practically-deployable A*CMD solution introducing well-bounded delay can significantly reduce potential attack impact on the grid stability when used with state-of-the-art attack detection systems.

The rest of the paper is organized as follows. In Section II, we discuss the related work. Section III discusses security threats in remote control of modernized substations. Section IV provides the recap of A*CMD solution for mitigating impact of attacks against substations. The procedure for finding delay tolerance of a power grid, along with consideration on the use of artificial delay for security, is elaborated in Section V. Discussion about the practical deployment and prototype implementation is made in Section VI. Simulation results regarding tolerable delay and attack mitigation as well as performance evaluation of our prototype are presented in Section VII. Finally, we conclude the paper in Section VIII.

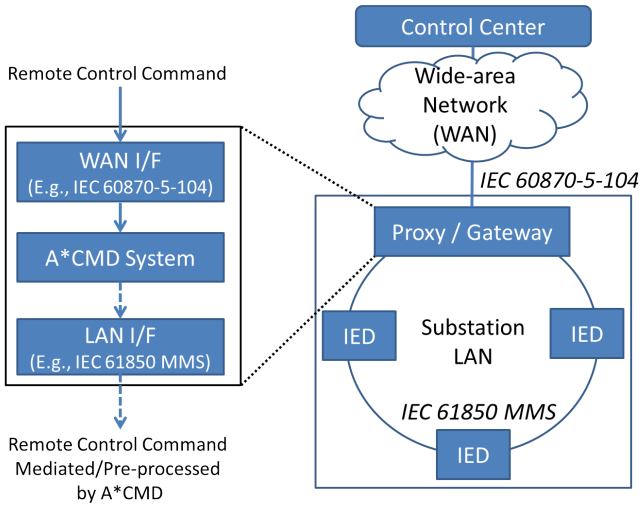


Figure 1. Overview of a Modernized Substation with Active Command Mediation Defense (A*CMD) System. A*CMD is implemented on Proxy/Gateway in a substation to reliably mediate all remote control commands [8].

II. Related Work

Regarding the tolerable amount of delay or communication latency, some guidelines are published by IEEE Power and Energy Society (PES) [10] and US Department of Energy [11]. Required network bandwidth and latency in wide-area network (WAN), neighborhood-area network (NAN), and home-area network (HAN) communication in various smart grid applications are also studied in academia, e.g., [12]. Although these provide useful information when studying latency requirements in the smart grid communication infrastructure, no quantitative evaluation is presented. Therefore, it is not clear if these are actual conditions on real systems. Moreover, it is likely that each power grid configuration has different delay tolerance, so in this paper, we demonstrate a way to quantitatively evaluate impact of delay in smart grid systems, particularly in substation remote control scenarios.

Use of artificial time delay for enhancing security and resiliency of smart grid systems itself is not entirely a new idea. For instance, in a smart metering context, Temple et al. [13] proposed a mechanism to enhance resilience against massive remote connect/disconnect attacks by inserting artificial delay on each smart meters. Although our solution is conceptually similar to theirs, securing substations requires different considerations for timing requirements and overall system design. The use of artificial delay in critical systems, including smart grids, to slow down and/or to detect and mitigate cyber attacks, is proposed by Hershey et al. [14]. However, they do not focus on securing remote control interface of substations or provide intensive evaluation regarding the impacts of added delay in a power grid environment.

The idea of inserting a mediation layer for cyber-physical system security is explored by Etigowni et al. [15]. Their system, sitting in front of programmable logic controllers (PLCs), mediates and inspects control logic binary sent to Siemens PLCs to evaluate if the outcome violates safety states. Theirs works at the different level of the smart grid systems (i.e., within substations) and aims at protecting devices of a specific model, while our solution secures standard-based remote control interface of substations.

Attack detection mechanisms designed for securing substation remote control are also studied. To the best of our knowledge, the most closely related to ours is [16]. Their scheme relies on centralized semantic analysis of control commands, based on power flow simulation, for detecting attack attempts. To mitigate impact of attacks, they use command-reversing, which sends reversing control commands shortly after the execution of malicious commands. However, it may not be always ideal because some of the grid control would take non-negligible amount of time to reverse. On the other hand, our mitigation scheme aims at canceling at least part of the malicious control commands before execution [8], which can reduce the cost associated with such issues. Another type of work in this category is command authentication using real-time power flow simulation on each substation. For instance, [17] utilizes distributed state estimation and simulation to estimate the consequence of commands. Although the proposed scheme can work with short latency, theirs requires intensive communication among neighboring substations, which may be exploited by false data injection. On the other hand A*CMD [8] aims at defense with minimal reliance on external systems to reduce such an attack surface.

III. Threats in Substation Remote Control

End-to-end remote control of substations is enabled by using two international standards established by the International Electrotechnical Commission (IEC) [18]: IEC 60870 and IEC 61850 [19], [20]. The former defines specification for telecontrol by the control center, and in particular, IEC 60870-5-104 over TCP is increasingly used, while the latter focuses on communication for automation within each substation (Figure 1).

While remote control of substations has been explored in a number of use cases, e.g., power shedding, load shedding, voltage regulation, and topology control for cost optimization,

as enumerated in [8], [21], and automation has improved timeliness and efficiency of operation, serious consequence would occur when these are misused, as seen in [21] and [1], [2]. In this paper, we focus on remote control interfaces of modernized substations and present the practical design and deployment of an additional layer of security for each substation, which is complementary to existing cybersecurity measures. Below, we elaborate attacks exploiting remote control interface of substations, which we aim at countering.

Attacks from Control Center: Various security solutions, such as industrial firewall, anti-virus software, authentication and access control systems, are proposed to protect smart grid systems. However, it is still possible that the control center system becomes malicious. For instance, disgruntled insiders, which are granted access to the control center system, may commit attacks by abusing their privilege. Even without malicious insiders, the attack is still possible. In the incident in Ukraine, adversaries mounted well-planned, longitudinal attacks, starting from spear phishing for sending malicious files to targeted employees and computers, steal access credentials, explore the entire system to locate SCADA (supervisory control and data acquisition) dispatch workstations and servers, and so forth [2], ironically demonstrating the validity of ICS (industrial control system) Cyber Kill Chain [22]. CrashOverride could also be used to override the IEC 60870-5-104 master or impersonate it to emit malicious commands [23].

Attacks on SCADA Communication Channel: Owing to insufficient security design and consideration in many SCADA communication protocols, including IEC 60870-5-104, the communication channel between the control center and substations may be compromised. Such vulnerabilities largely stem from lack of message authentication in the protocol as well as dependency on the widely-used TCP/IP protocol. Specifically, the feasibility of replay attacks and man-in-the-middle attacks against IEC 60870-5-104 is demonstrated in [24]. Yang et al. discuss the use of ARP (address resolution protocol) spoofing [25], which allows an attacker to direct packets to himself, to secretly capture messages between the control center and the destination. Moreover, if a cellular channel is utilized for communication between the control center and substations, similar attack could be mounted by compromising intermediate base stations [26]. Although IEC 62351 standards are defined for enhancing security of IEC 60870 and 61850 protocols, the real-world deployment of it is not yet common unfortunately.

Attacks via VPN Interface: In some cases, substations may have VPN (virtual private network) interface for grid operators to perform remote maintenance and monitoring. VPN interfaces are sometimes made available even to third-party device vendors for technical support etc. Once such an interface is compromised, e.g., by stealing access credentials through spear phishing etc., an attacker could inject arbitrary control commands into substations directly (e.g., Tofino has a product for tunneling IEC 60870 protocol over VPN [24]).

IV. Active Command Mediation Defense (A*CMD)

Although there are differences in the ways attacks are mounted, from the perspective of substations, they can be collectively modeled as attacks coming from outside (i.e., external to substation gateways). In this section, we provide the overview of active command mediation defense (A*CMD) solution proposed in [8], which is specifically designed to counter such attacks. Then, we discuss attack detection mechanisms that are available for use with the A*CMD solution.

A. System Overview

Let us start with the high-level architecture of modernized electrical substations (see also Figure 1). According to [19], [20], a substation consists of three levels: station, bay, and process level. Communication among devices in the station level, at which Proxy/Gateway is located, and ones in the bay level, where IEDs (intelligent electronic devices) are typically deployed, is done via IEC 61850 MMS (manufacturing message specification). Proxy/Gateway may be implemented on an RTU (remote terminal unit). Because, in typical deployments, protocols used towards and within substations are different [20], one of its key responsibilities is protocol translation, for instance between IEC 60870-5-104 and IEC 61850 MMS. Namely, all control commands and configuration changes sent by the control center using IEC 60870-5-104 must be mediated by Proxy/Gateway before reaching IEC 61850-compliant IEDs, which are communication end points responsible for operating on physical devices [20].

The A*CMD's approach for enhancing security and resiliency of smart grid systems is to introduce an additional "thin but unbypassable" layer between the control center and physical components in substations to minimize changes required on existing systems while offering reliable mitigation [8]. As discussed earlier, all remote control commands go through Proxy/Gateway (called *gateway* hereafter), making a gateway an ideal place to deploy the A*CMD system (Figure 1). The A*CMD system is responsible for actively inspecting and pre-processing them before execution on physical power system devices. Next, we discuss one concrete example of a command mediation scheme, called *autonomous command-delaying*, that can be implemented on A*CMD.

As the name implies, autonomous command-delaying scheme makes each A*CMD system implemented on or as a substation gateway independently add artificial time delays before forwarding the control commands to IEDs. The purpose of the artificial delay is to provide an attack detection system with time buffer to complete its job and then to cancel any suspected commands. If these are done before delayed malicious commands are passed to IEDs, those commands will never be seen by the IEDs and hence will cause no impact. The overview is illustrated in Figure 2. In the figure, for the sake of simplicity, only one IED under each substation gateway is drawn though in practice there can be multiple IEDs per substation. We should note that the description below is applicable to the multiple-IED case in a straightforward way.

When receiving an incoming remote control command, the A*CMD system at each substation independently inserts an artificial delay before command execution. Such a delay is illustrated as $d_1, d_2, \ldots, d_n$. The upper bound of the delay (D_{ub}) can be configured based on system and security needs.

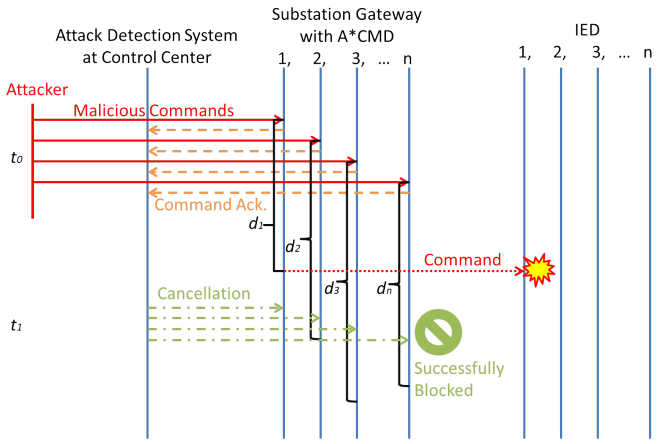


Figure 2. Overview of Autonomous Command-delaying [8]. In the example, although an attack command sent to IED 1 in Substation 1 is executed, the attack detection system initiates cancellation at t_1 to cancel the others during artificial delay. In this way, A*CMD scheme can reduce the number of attack commands executed and therefore can mitigate the impact on the power grid.

The A*CMD system sends confirmation of the command receipt according to the IEC 60870-5-104 standard. In addition, the A*CMD system sends an additional acknowledgment message, conveying copy of the received command payload, to a trusted attack detection system, ideally via a separate, secure communication channel. An attack detection mechanism deployed outside of substations (e.g., at the control center), works on evaluation based on the received command acknowledgement messages as well as other supplementary information. Once any command is suspected as attack, then it sends cancellation of the pending attack commands (i.e., malicious commands that are received by the A*CMD system but not yet executed owing to the artificial delay). Some concrete attack detection schemes will be discussed in Section IV-B. Again, the cancellation message should also be sent via a trusted channel to prevent, for example, denial-of-service attacks attempting to maliciously cancel legitimate commands.

*B. Attack Detection Mechanisms for A*CMD Framework*

The overall performance and effectiveness of A*CMD schemes depend on the capability of the companion attack detection systems. Specifically, in case a legitimate command is mistakenly flagged as an attack (false positive), it would not be executed (i.e., canceled before execution) and therefore some backup action (e.g., re-sending) may be needed. On the other hand, if attack commands are not detected successfully (false negative), those are executed after the artificial delay and therefore their impacts on the grid may not be mitigated.

In this paper, we focus on a case where an external, centralized attack detection system is deployed at the resource-rich control center because this option is considered practical in the near term. When an attack detection system is deployed at the control center, to avoid it being compromised by an attacker, it should be sufficiently isolated (e.g., deployed on a physically different machine) from the SCADA master system used for substation remote control and monitoring. A*CMD scheme is intended to be used with a variety of attack detection mechanisms based on operators' needs, and therefore specific design of detection system is left outside of our primary scope. Having that said, we discuss a couple of practical options suitable for the proposed A*CMD framework. Each mechanism has advantages and limitations, so the choice should be made depending on system needs, security preference, and budget. We should also note that the list here is non-exhaustive and many other options are, and will be, available.

First example is a simple detection scheme based on log (or history) of commands that are sent out by the control center in the recent past. Such an attack detection scheme has read access to the command history and, when receiving command acknowledgement from substations, the system looks up the history. If no match is found, then the attack detection system flags it as an attack and sends cancellation. The mechanism is very simple and therefore can be completed quickly, say in the order of milliseconds. Another advantage is that this scheme causes no false positives (i.e., commands that are actually sent by the real control center are never flagged as attacks). On the other hand, one major limitation is that, if the control center system was misused either by malicious insiders or by means of malware, malicious commands sent out are included in the history and therefore are never flagged as attack.

Another simple option may be a centralized rate-limiting. An attack detection system can count the command acknowledgements for the entire grid, per substation, or per geographic area, and then raise an alarm when the count exceeds a predefined threshold. Such a scheme is also easy to implement and the processing time can be even shorter than the previous option. The limitation is that attacks involving a small number of commands, which might be well crafted to target only significant components, may not be detected.

Advanced options are also available in the literature. For example, as briefly discussed in Section II, our command mediation defense framework can be used with a centralized attack detection scheme using power flow simulation [16]. Owing to the complexity, its processing time (less than 600ms) is longer than the simpler options. The processing time could be further shortened by employing simplified simulation based on a Thevenin-equivalent circuit or the technique proposed in [17]. Regarding accuracy, authors demonstrated, by means of simulation using IEEE bus systems, that it can offer nearly 100% detection rate and very low false positive rate far below 1% (or even 0% in some power grid settings) [16]. This result allows us to assume that there exists an attack detection system that offers very high accuracy with acceptable latency, and therefore in the later experiments, we demonstrate how well the A*CMD framework works when used with the state-of-the-art attack detection system.

C. Security Considerations on Communication Channel

As discussed in Section IV-A, ideally an A*CMD system should utilize secure, dedicated communication channel for exchanging command acknowledgment and cancellation. To avoid the risk of message tampering and impersonation, integrity-protected, authenticated channel (e.g., TLS with client authentication) should be used. Use of secondary

channel, other than one for normal SCADA traffic, can further improve reliability and availability of such communication.

Some may wonder why not simply direct all SCADA communications through the trusted channel, if it is implemented for A*CMD. Although it may address many security problems, the main reason not to do so is to minimize the required changes on existing infrastructure. In reality, power grid operators have already established their system based on standards such as IEC 60870-5-104. If one mandates the use of a non-standard secure channel for all communication, it would require a major upgrade or replacement of the current system. On the other hand, use of trusted communication only between newly-added system components, namely an attack detection system and an A*CMD system in each substation, requires minimal change on the existing SCADA system.

A centralized detection design may suffer from potential attacks on its communication channel with the distributed A*CMD system. In other words, communication between the attack detection system and substations may not be functional or be blocked or delayed by adversaries. Such a situation could be set up by attackers when they are launching malicious command injection to substations. Once this happens, command acknowledgment and cancellation messages cannot be delivered, and therefore most of the benefit from A*CMD (except for slowing down attacks to some extent) would be lost. However, these are not part of the critical path for SCADA control and monitoring, so normal operations are not interrupted. To minimize the downtime, failure of the channel can be detected by using periodic heart-beat messaging etc.

Another direction for addressing such a problem is to employ a distributed detection design that collocates with the A*CMD system, instead of or in addition to the centralized detection system, where the distributed detection system utilizes locally collected, thereby trusted, measurements, e.g., bus voltage, frequency, etc. for its decision making. There are some ongoing work for such distributed attack detection, e.g., [17], and we leave discussion of the detailed mechanism and how it may be combined with the centralized one for future work.

V. Design Consideration on Artificial Delay

Perhaps the most controversial, but critical, issue in real-world deployment of command-delaying approach is the negative impact of artificial delay. Generally speaking, no (or short) delay is considered good in smart grid controls. For instance, fault isolation, relay protection, etc., require very short latency (e.g., 10ms or lower). However, it may not be a necessary condition in the substation remote control context while we admit they are definitely sufficient conditions for grid stability. We here argue that, if there is a gap between the necessary condition (i.e., maximal delay allowed) and technical constraints (e.g., network latency), we can take advantage of the difference for enhancing smart grid security. After that, we discuss practical command-delaying strategies.

A. Artificial Delay for Securing Substation Remote Control

Let us start with studying publicly available guidelines. According to IEEE PES guideline [10], maximum delivery time for transfer switching, line sectionalizing, and load control and shedding are 1, 5, and 10 seconds respectively. Thus, seconds-level delay is considered tolerable in these use cases. Note that these are closely related to the remote control use cases mentioned in Section III, namely power/load shedding, topology control, and voltage regulation. Besides, according to [12], communication latency requirements for the distribution automation (e.g., voltage/VAR control) are said to be less than 4-5 seconds, and even longer for many of other scenarios. We also found criteria regarding the protection against transmission line overload. According to [27], circuit breakers is supposed to trip within around 3 seconds in case current on a transmission line is twice as much as its nominal limit. Time-current curves (also called damage curves) for higher-voltage systems are also available from IEEE [28]. Based on these curves, in the case of 200% overload, transformers tolerate for around 30 seconds while around 8-second latency is acceptable for cables. Even with consideration of safety margin, delaying circuit breaker control by 1-2 seconds is considered tolerable.

Moreover, many of the remote control use cases do not have stringent timing constraints. For instance, since the purpose of topology control is economical optimization, circuit breaker control for this purpose may be delayed by seconds or even minutes without any negative consequences in grid stability. Regarding voltage regulation by controlling shunt reactors, according to our industry partner, controls are manually done by human operators usually in the morning and evening, which again is not considered sensitive to minor delay. Among the use cases, power shedding may be considered time-sensitive. Thus, in [8], we showed that delaying the recovery control by 2 seconds or so in an artificial over-generation scenario did not cause any voltage/frequency violation. In sum, for the typical substation remote control use cases in our scope, delay around 1-2 seconds is considered acceptable.

However, tolerable delay, i.e., an amount of delay that can still preserve the power grid stability, would vary depending on grid configuration and other factors. The procedure to find such a tolerable delay, D^*, can be formulated as in Algorithm 1. The algorithm requires three inputs, PG, SC, and CTG. The PG input includes the grid topology, the configuration of each power system component, and status snapshots that are typically used for contingency analysis and power-flow simulation. The SC input contains stability conditions in terms of frequency and voltage violation etc. The CTG input is a set of contingencies that require (typically automated) remote control for recovery and can be defined based on $N-1$ (or more generally $N-x$) contingencies. $findRecoveryControl()$ invokes contingency analysis to find an appropriate set of recovery control commands. $findTolerableDelay()$ invokes iterative power flow simulations (in particular transient-state simulation) with varying delays to find the maximum delay that can be introduced without causing violation of SC for the given contingency scenario and recovery control. D^* is selected as the minimum of all $Delay_c$s, each of which represents delay tolerance for a certain contingency. The search space for $Delay_c$ can be narrowed down based on the discussion in this section. The initial value for D^* can also be set accordingly, e.g., to around 2 seconds. Note that, instead

Algorithm 1 Finding D^* for Given Power Grid Model

Require: $PG \leftarrow$ Power grid model and topology
Require: $SC \leftarrow$ Power grid stability conditions
Require: $CTG \leftarrow$ List of contigencies in scope
  $D^* \leftarrow$ Initialize with maximum delay to be considered
  for each C in CTG **do**
    $Ctl \leftarrow findRecoveryControl(C, PG, SC)$
    $Delay_c \leftarrow findTolerableDelay(C, PG, SC, Ctl)$
    $D^* \leftarrow Min(Delay_c, D^*)$
  end for
  return D^*

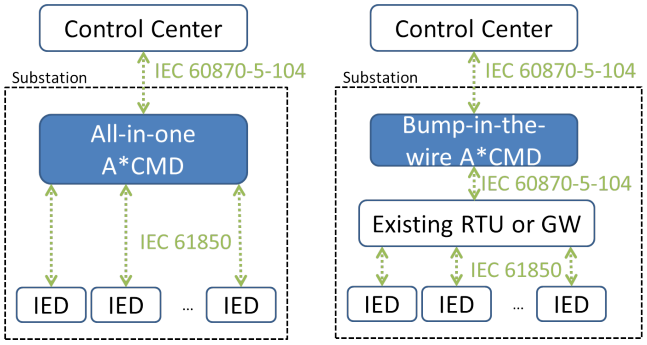
of defining a global D^*, we can also consider associating different delay tolerance with each individual device, and Algorithm 1 can be generalized accordingly. We leave the generalization of this algorithm to future work. Section VII-A will show an example following Algorithm 1, focusing on circuit breaker controls for load shedding.

In order to evaluate mandatory latency caused by network, we separately conducted experiments for measuring communication latency, using 4-hop local area network with industry-grade switches, and found that pure round-trip network latency was far below 10ms. As we saw earlier in this section, second-level delay is acceptable for typical remote control use cases. Even controls like power shedding that may be often automated can still be delayed by this degree. Thus, we are convinced that a certain fraction of the difference between the acceptable amount of delay and pure communication latency (in the order of 100 milliseconds to second) can be taken advantage of for enhancing power grid resiliency.

B. Practical Command-Delaying Strategy for A*CMD

This section discusses some practical ways to add artificial delay in an autonomous manner. One straightforward strategy is to delay all control commands by a constant value to buy sufficient time for an attack detection system. The delay should be minimal and bounded not to cause any negative impact. We denote the time needed for attack detection and response be T_d. If the detection mechanism at the control center is simple and deterministic (e.g., history-based attack detection described in Section IV-B) and the round-trip time through the trusted communication channel is stable, one may regard T_d as a known constant value. In this case, delaying all commands by T_d (or slightly longer to include some safety margin) allows us to expect the best-possible mitigation (i.e., all detected attack commands can be canceled in time). If delay tolerance of the grid D^*, which can be found in a way discussed in Algorithm 1, is greater than T_d, the operator can simply choose to delay all remote control commands by D^* (i.e., $D_{ub} = D^*$).

In some cases, certain legitimate command combinations cannot be delayed long enough (i.e., $D^* < T_d$), especially when T_d is relatively long. In such a case, a grid operator can identify (e.g., again by simulation) how many of the commands have to be executed immediately to avoid negative consequence (e.g., frequency/voltage violations), if the rest of commands are delayed by T_d. In other words, the operator executes some commands with no delay to earn sufficient time for the attack detection and response mechanism to complete its task to save the rest. (Observation that justifies this approach will be discussed in Section VII-A.) Such a delaying strategy can be realized without any communication or runtime coordination among substations, by using a *discrete-random-delay* approach, where each A*CMD instance, for each remote control command, independently adds delay of T_d with probability $1 - P_{nd}$ (i.e., $D_{ub} = T_d$) and adds no delay (i.e., immediate execution) with probability P_{nd}. The value of P_{nd} that ensures high probability of having the minimum number of commands immediately executed can be analytically calculated based on binomial distribution. For instance, if we set $P_{nd} = 0.5$, at least 2 commands out of 10 are executed with no delay with 99% probability. Besides the partial guarantee of timely command execution, probabilistic delaying makes it difficult for attackers to mount well-crafted, timing-sensitive attacks (i.e., moving target defense [29]).



(a) All-in-one (b) Bump-in-the-wire (BITW)

Figure 3. Practical A*CMD Deployment Strategies

Instead of determining delays in an probabilistic way, yet another option is to make decision based on real-time simulation. For instance, recently distributed state estimation and simulation technology has been proposed [17]. Using such a scheme, upon receiving control commands, a command mediation system could run simulation with immediate execution of the incoming commands. Then, if the simulation result does not show any violation, the command could be passed to an IED immediately, while otherwise delay is introduced for secondary, centralized investigation based on the comprehensive view of the grid. Moreover, it is also possible to utilize the similar simulation to dynamically determine the duration of the maximal delay. For instance, upon receiving a control command, A*CMD system runs multiple simulations with various delay for evaluating the negative impact and then decide the maximal delay for each command. Exploring this option further is also an interesting future work.

VI. A*CMD Deployment Options and Prototype

In this section, we discuss two deployment options for practical integration of A*CMD to meet various system needs and preferences, followed by prototype implementations of both.

A. Deployment Option 1: All-in-one Substation Gateway

In the real-world deployment, the role of a gateway in the reference model [20] is often implemented as a protocol

translator (or protocol gateway) [30], and there are a number of commercial products in the market, such as [31], [32] that do translation of protocols used in smart grid context, including IEC 60870-5-104, IEC 61850, and DNP3. Therefore, one straightforward deployment option is to implement all the components in "Proxy/Gateway" in Figure 1 on a single box and replace an existing substation gateway or protocol translator device with it. The substation system architecture with this deployment option is found in Figure 3(a).

An advantage of this approach is that no extra hardware is needed, and therefore degradation of reliability and availability of the entire infrastructure is minimal. On the other hand, drawback may be that this option requires implementation of equivalent features available on the existing gateway device or protocol translator to be replaced, which is often non-trivial without sufficient technical supports from device vendors.

It is also possible to implement A*CMD on the existing protocol translator (or RTUs) if the platform is extensible. In this case, A*CMD could be introduced simply by updating software of such devices, which is highly beneficial for large-scale roll out. However, implementation has to be entirely device-specific, which may require significant customization efforts especially if we need to support multiple models. In addition, protocol translators and RTUs are usually resource-constrained, and therefore we may face extra technical challenges when porting the A*CMD system to those devices.

B. Deployment Option 2: Bump-in-the-wire (BITW) Approach

There may be a situation where the all-in-one option is not desirable. For instance, as mentioned earlier, there may be a case where implementing equivalent features on the new gateway device is not feasible. Also, depending on system configuration, there may be a situation where no protocol translation gateway is needed but additional security for just a single RTU is desired. To meet such demands, another deployment option is to deploy A*CMD as an add-on system component without requiring modification on an existing gateway or RTU. Furthermore, this option allows us to take advantage of functionality of the existing gateway device or RTU for handling and interacting with IEDs or physical power system devices in a substation.

Specifically, we can deploy the A*CMD system as a separate box in "bump-in-the-wire" manner as shown in Figure 3(b). Compared to the all-in-one option, this approach might suffer extra latency caused by communication between the A*CMD box and the existing gateway/RTU. Another potential drawback is that adding extra component into the communication path may increase failure rate of the system, which however can be addressed by redundancy. On the other hand, this option has a significant advantage in terms of broad applicability and feasibility of integration. Namely, this option does not require hardware/firmware customization on existing devices and can support any standard-compliant devices.

The resulting architecture is shown in Figure 3(b). "Bump-in-the-wire A*CMD" and "Existing RTU or GW" together offer the equivalent functionality to "All-in-one A*CMD" in Figure 3(a). Also note that incoming and outgoing messages on "Bump-in-the-wire A*CMD" are in the same protocol.

C. A*CMD-Pi: An A*CMD Prototype on Raspberry Pi

To evaluate the feasibility and practicality of our design, we have implemented prototype systems that support IEC 60870-5-104 (for the control center) and IEC 61850 MMS (for IEDs). Although in [9] we have reported some preliminary results corresponding to all-in-one option, we implemented both deployment options according to the discussion made in this section to study their performance and compare the two. The high-level module architecture is shown in Figure 4.

*A*CMD Main Module* is the main component of the added security, which intercepts and interprets incoming commands and handles them according to its mediation logic. For instance, in the case of autonomous command-delaying, this module inspects command payload and, if the command is a control command, adds an artificial delay with a certain probability before passing it to the IEDs, while letting interrogation commands go with no delay. Supplementary logic to determine delay and/or detect attacks can be optionally implemented on *Decision Support Module*. This module can utilize voltage, power flow, frequency, device status etc. measured via local IEDs, historical data, and other contextual information for decision making, and is currently a place holder to accommodate future research. *Secure Communication Module* establishes secure, authenticated communication channel with external *Attack Detection & Response System*. Necessity of this module depends on the logic implemented on A*CMD Main Module. Autonomous command-delaying uses it for reliably sending command acknowledgment and cancellation messages. As discussed in Section IV, this channel is independent of the one for standard-based communication (e.g., IEC 60870-5-104).

A number of commercial products, e.g., [32], run on Linux OS and ARM processors. Thus, we decided to use Raspberry Pi 1 Model B with a single-core ARM processor (700MHz) and 512 MB RAM as the platform for our prototype, named *A*CMD-Pi*. We used 8GB SD card for the storage and loaded Linux Lite OS. We implemented the A*CMD system in Java, and employed OpenMUC [33] for supporting communication using IEC 60870-5-104 and IEC 61850 MMS. Regarding secure communication with an attack detection system, we employed TLS (Transport Layer Security) with client authentication, using 2048-bit RSA keys.

In the all-in-one approach (Section VI-A), protocol translation leverages the IEC 61850 Substation Configuration Language (SCL) for the entity mapping and supports typical commands for different substation components, including circuit breakers, transformers, shunt reactors, and generators. The BITW A*CMD-Pi does not implement protocol translation itself, and both incoming and outgoing messages are in IEC 60870-5-104 so that it can interact with any existing gateway or RTU compliant with the widely-used standard.

VII. Evaluation

A. Finding Tolerable Delay for Various Power Grid Models

Configuring appropriate command-delaying is crucial to avoid any negative consequences. In this section, we demonstrate execution of Algorithm 1, to find delay tolerance on three power grid models of different sizes. In this study, we

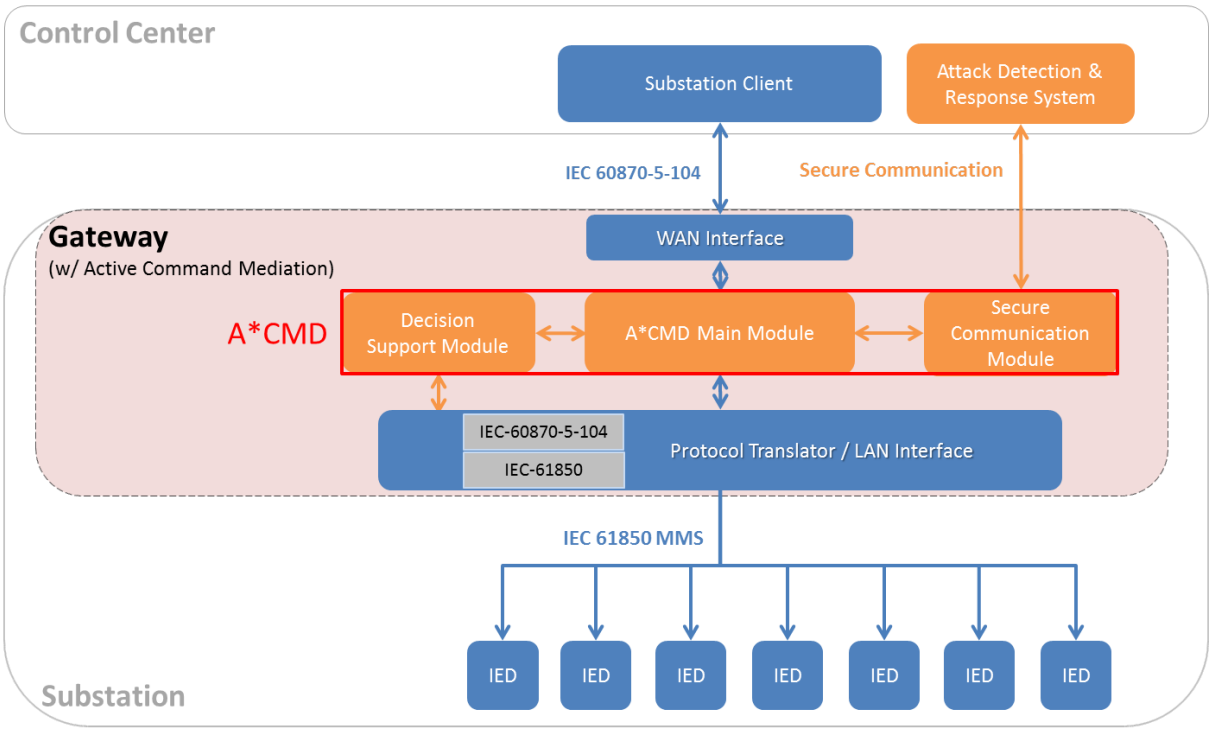


Figure 4. Overview of A*CMD Prototype Module Architecture (All-in-one). The received IEC 60870-5-104 messages are first intercepted by A*CMD Main Module, which determines artificial delay. At this step, it can optionally call Decision Support Module. At the same time, A*CMD Main Module sends command acknowledgement to Attack Detection & Response System, using Secure Communication Module. When an attack is suspected, Attack Detection & Response System sends cancellation, which is provided to A*CMD Main Module via Secure Communication Module. In the case of BITW A*CMD, Protocol Translator is off-loaded onto an existing gateway.

focused on load-shedding control to maintain grid stability under generator-loss contingency, though different type of contingency and recovery control can be evaluated likewise.

We simulated all $N-1$ contingencies focusing on generator loss, using the well-known **GSO 37-bus system** [34] with 9 generators. The procedure of our experiment is:

1) For each $N-1$ contingency case, run transient-state simulation to see if violation (in this study, in terms of frequency) occurs.
2) If the case faces violation, find a set of loads to be shed to avoid the violation ($findRecoveryControl()$ in Algorithm 1).
3) Apply the load shedding with different delay configurations and run transient-state simulations to find maximal delay that does not cause violation ($findTolerableDelay()$ in Algorithm 1).

As the threshold for frequency violation, we used $\pm 1\%$ (i.e., ± 0.6Hz in this case) [35]. Regarding Step (2), our strategy to find the appropriate load-shedding set is to add loads one by one from the one nearest to the lost generator. Although we do not claim it is the optimal set, we believe our approach is reasonable from grid operators' perspective. Development of mechanism for finding a cost-optimal load-shedding strategy (or other types of recovery strategy) in a systematic and/or distributed way is part of our future work. The results are summarized in Table I. As can be seen, response to the loss of bigger generators is less tolerable to delay. For the loss of the largest generator (150 MW), a maximum of 0.9 second delay can be introduced, which is in fact sufficient to be used with advanced attack detection scheme like [16].

Table I
SIMULATION RESULTS WITH 37-BUS SYSTEM [34]

Name of Gen.	Gen. MW	# of Loads Shed	Max Delay [s]
JO345 #1	150	5	0.9
JO345 #2	150	5	0.9
LAUF69	150	5	1.0
BLT138	140	3	1.2
BLT69	75.23	2	2.5
ROGER69	38	1	3.0

We did the same $N-1$ contingency experiment with **Illini 42 Tornado** case including 42 buses, 65 transmission lines, 14 generators, and 55 loads, whose sum is 10,487MW. (Further details can be found in the case file available on [36].) For this experiment, we deleted all pre-configured contingencies and simply used it for $N-1$ generator-loss simulations. In this experiment, we also found that load-shedding operations can be delayed by 1 second without causing violation. We then conducted an experiment with the much bigger **Texas 2,000-bus system**. This case file includes 2,481 transmission lines among 2,007 buses, 282 generators, and 1,417 loads (in total 49,775MW). Other configurations and parameters can be found in the case file available on [37]. In this case, because no $N-1$ contingency caused violation, we focused on a case where top-4 largest generators are lost. Owing to the size of the grid, we considered a tighter violation threshold,

namely ± 0.4Hz. The experiment showed that the recovery load-shedding control can be safely delayed by 0.7 second.

Through the experiments, we also found that, in some cases, applying one load-shedding control with no or short latency allowed us to delay the other recovery controls longer. For example, in the case where generator "JO345 #1" is lost in 37-bus system (see Table I), if one of the load is shed with 500ms delay, the rest of the controls can be delayed by 1.2 seconds, instead of 0.9 second. Thus, when some of the controls are either configured to be triggered automatically or delay is probabilistically added as discussed in Section V-A, acceptable delay for the rest can be longer. This observation justifies discrete-random-delay strategy discussed in Section V-B.

Finally, in order to evaluate acceptable delay under a realistic contingency scenario, we again used the Illini 42 Tornado case. As described in [36], this case involves pre-configured line faults and generator loss simulating a tornado attack, and the goal is to find a way to prevent system-wide blackout from happening. For the interest of space, we focus on load shedding as recovery actions for the generator loss and line faults hard-coded in the case. To evaluate how long the recovery control commands can be delayed, we varied the time at which these load-shedding commands were executed (in regard to the time the contingency events happen). When we used 10-second delay for all, as shown in Figure 5 (1), we observed a frequency change shown in Figure 5 (2), which confirmed that 10-second delay did not cause a blackout. This observation also backs the discussion in Section V-A.

B. Attack Mitigation by Autonomous Command-delaying

When conducting experiments, the parameters to be defined to simulate the A*CMD with autonomous command-delaying are upper bound of artificial delay added by each A*CMD system (D_{ub}), probability that no delay is added (P_{nd}), and time required by the security system in the control center for detection of attacks and cancellation of commands (T_d). Among them, D_{ub} and P_{nd} can be configured by grid operators based on their operational needs, while T_d is determined by the attack detection system employed. In our experiments, T_d is set to be 620ms, including 20ms for network latency (for command acknowledgment from substation and cancellation from the control center) based on our preliminary measurements using our testbed consisting of industry-grade Ethernet switches and 600ms for attack detection based on the number provided in [16]. To analyze the attack impact on grid stability, we used the 37-bus system [34] on PowerWorld simulator [38].

We here evaluate the impact of large-scale, simultaneous attacks, which is similar to the Ukraine incident [1], and the effectiveness of the autonomous command-delaying scheme. As metrics of attack impact, owing to the space limitation, we focus on the discussion of the following: (1) the number of buses that experience voltage violation; (2) the occurrence of frequency violation; (3) the amount of unserved load (i.e., reduction in load). For stable power grid operation, it is crucial to maintain electricity frequency and voltage within a certain range from the nominal values. We used Western Electricity Coordinating Council (WECC) criteria [39] for these. Besides, after execution of malicious control commands, some loads are entirely disconnected or shed as a result of balancing process. Thus, we included the third one in the list, which is calculated as follows. For each load in the simulated power system, the difference of the load at the beginning of the simulation (i.e., before attacks) and that at the end (i.e., after attacks) is calculated. Unserved loads calculated in this way are summed up for each experiment, which is then averaged over all experiments.

Table II
PERFORMANCE MEASUREMENTS

Setup	Sustainable Throughput (Commands / sec)	CPU Usage (%)	Memory Usage (%)
All-in-one [9]	33	36.70	15.40
No A*CMD [9]	33	23.97	13.60
BITW w/ RPi	33	26.16	8.60
BITW w/ PC	65	37.50	8.80
BITW only	over 87	44.28	16.20
ZNX 202 [40]	less than 10	-	-

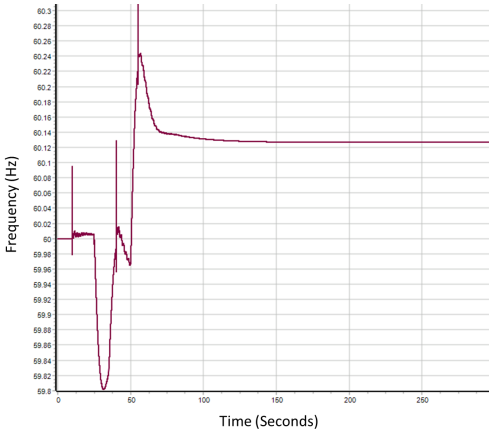
Similar to the Ukraine case, we simulated attackers sending "open" commands to randomly-selected circuit breakers. The simulation experiments were performed with different fraction of attacked circuit breakers, ranging from 10% to 50%. We set D_{ub} to 700ms, which is slightly longer than T_d=620ms. Note that this level of delay is considered acceptable based on the observation in Section V-A and Section VII-A and also that it is compatible with an advanced attack detection system discussed in Section IV-B. For each setting, we repeated experiments for 100 times with randomly selected (i.e., different set of) circuit breakers, based on which we calculated the average. Figures 6 to 8 show results with the discrete-random-delay approach with different probability for delaying [8]. We can see significant mitigation especially when delaying probability is high (i.e., P_{nd} is small). Specifically, the negative impact, in terms of both occurrence of violation and amount of unserved loads, is reduced by over 98% (with P_{nd}=10%) and 90% (with P_{nd}=25%). We also performed the similar experiments with T_d=30ms (and D_{ub}=50ms), which corresponds to the case with simple, history-based attack detection scheme, and confirmed the similar (see Figure 9). Although these experiments assumed an ideal case with 100% attack detection accuracy after T_d, given the high accuracy of state-of-the-art detection scheme, e.g., [16], these results can be considered good approximation of the practical deployment.

*C. Performance Evaluation of A*CMD-Pi Using SoftGrid*

Next, we present evaluation of A*CMD-Pi in terms of performance and resource usage. We used SoftGrid [9], which is an open-source toolkit designed for evaluating compatibility, performance, and effectiveness of cybersecurity products or solutions that are deployed between the control center and substation IEDs. Although SoftGrid allows us to perform various evaluation, e.g., the degree of mitigation of physical impact by simulating attacks etc. [9], we focus on the performance aspect since it is the crucial factor to see the practicality of A*CMD on resource-constrained smart grid devices.

No.	Time (sec)	Contingency / *Control*	Location
1	10.00	Line Fault	Prairie345 - Bear345
2	10.05	Line Opened	Prairie345 - Bear345
3	25.00	Generator Opened	Prairie345
4	*	*Load Shed*	Prairie345#1
5	*	*Load Shed*	Prairie345#2
6	40.00	Line Fault	Hawk345 - Prairie345
7	40.05	Line Opened	Hawk345 - Prairie345
8	*	*Load Shed*	Valley138#3
9	*	*Load Shed*	Bear138#1
10	*	*Load Shed*	Rose138#1
11	55.00	Line Fault	Tiger345 - Prairie345
12	55.05	Line Opened	Tiger345 - Prairie345

(1) Pre-defined contingencies [36] as well as added load-shedding controls



(2) Frequency change when load-shedding controls are applied with 10-second delay

Figure 5. Evaluation of Acceptable Delay Using Illini 42 Tornado Case [36]. This case has pre-configured contingencies, including line faults at 10 second, 40 second, and 55 second, and generator trip at 25 second, as found in (1). Five load-shedding controls, which are shown in *italic*, are introduced by us with different delay from the preceding events. As seen in (2), we confirmed that delaying recovery controls by 10 seconds did not cause blackout.

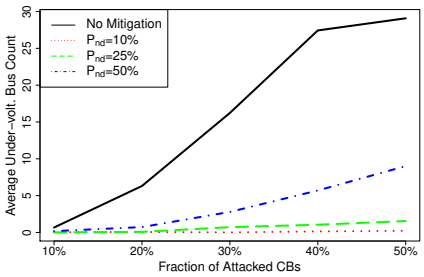


Figure 6. Attack Impact in Terms of # of Buses with Under-volt. Violation (T_d=620ms)

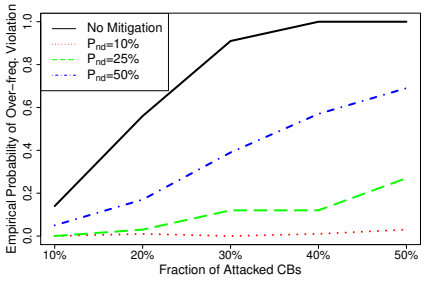


Figure 7. Attack Impact in Terms of Occurrence of Over-frequency Violation (T_d=620ms)

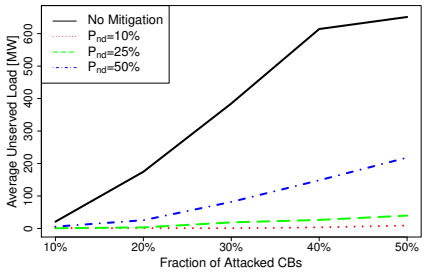


Figure 8. Attack Impact in Terms of Total Unserved Load (T_d=620ms)

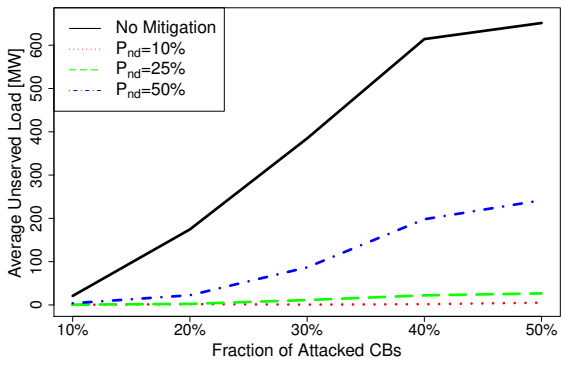


Figure 9. Attack Impact in Terms of Total Unserved Load (T_d=30ms)

Our goal is to identify the maximum number of commands that each A*CMD-Pi implementation can handle in a "sustainable" way. Our strategy is to first gradually increase the number of commands sent by the control center module and identify the maximum number of commands that A*CMD-Pi can stably handle (i.e., without showing increasing trends in its CPU and memory usage or response time). We ran SoftGrid on a Windows 7 PC with 32GB memory and Intel Core i7-6700 processor, which is connected to A*CMD-Pi via an industry-grade switch. On SoftGrid, we used the 37-bus system [34], which is accessed by 132 simulated IEDs. We customized SoftGrid and utilized multi-threaded control center. Then, we configured all the threads to send a control command in each time interval of a certain length. To avoid sending all the commands at the same time, the command dispatch time was randomly distributed within each interval. Then, we gradually shortened the interval to find the sustainable throughput.

Based on our measurements, all-in-one A*CMD-Pi can handle 33 commands per second. In other words, if the A*CMD-Pi receives more commands, excess commands will need to be queued up in its buffer or be dropped when the buffer is full. Regarding BITW A*CMD-Pi, we considered two cases: deployment with a protocol translation gateway on another Raspberry Pi and with a gateway on a resource-rich desktop PC. As seen in Table II, when the A*CMD-Pi is connected to the Raspberry Pi gateway ("BITW w/ RPi"), the system as a whole can handle 33 commands per second, whereas the deployment with the PC gateway ("BITW w/ PC") can handle 65 commands per second. It is clear that the performance of the PC is much higher than the Raspberry Pi's, and hence it can stably handle higher number of commands per second. This also indicates that the bottleneck was the protocol translation part and the security functionality of BITW A*CMD-Pi itself can support a significantly larger number of commands per

second. To further investigate how many commands BITW A*CMD-Pi alone can handle, we measured the number of commands that BITW A*CMD-Pi can accept and forward to an external gateway. The results are shown in "BITW only" row of the table, which shows an even larger number. Lastly, to discuss the overhead caused by the A*CMD module on a substation gateway, we present the result from [9], which corresponds to the all-in-one option without any command mediation features. As seen in "No A*CMD" row, it handles the same number of commands per second, and we see that enabling A*CMD does not lower the throughput.

We were not able to find specific information about the desired throughput for a substation remote control gateway, so we studied network traffic dump captured in a real substation system during a system testing phase. The duration of the collection was 21 hours, and contained 954 request commands, including clock synchronization, interrogation, and control commands. The number of control commands, which are of our particular interest, was 204, and per-minute count was at most 16, which is translated to 0.26 commands per second. In addition, we evaluated the performance of ZNX 202, an off-the-shelf, commercial protocol translator that is equipped with ARM9 400MHz processor and 128MB RAM and is capable of translation between IEC 60870-5-104 and IEC 61850 MMS [40] (Table II). Although ZNX 202 is designed for low-voltage, small-scale distribution substations, the comparison with it provides us with a meaningful baseline. We conducted measurements using the same SoftGrid testbed [9]. Since we were not able to directly evaluate the stability in resource consumption on ZNX 202, we relied on increase in response time to identify the sustainable throughput. During our 1-hour experiment sending 10 commands per second to the ZNX 202, the response time was no shorter than 103ms and showed a monotonous increasing trend, which implies that this rate is already beyond the limit of its sustainable command throughput. We also note that, when commands were sent at higher rate or concurrency, the increasing slope was even steeper. Based on these findings, the throughput that A*CMD-Pi can stably offer is considered practically sufficient.

Finally, we measured CPU and memory usage for both deployment options, and results are summarized in Table II. As expected, memory consumption in BITW implementation is less than the all-in-one case. This is because memory-intensive protocol translation is off-loaded. Furthermore, we note that even during over-night experiments with the same configurations, both types of A*CMD implementation offered consistent performance without causing any stability issue.

VIII. Conclusions

In this paper, we discussed practical issues to be addressed when designing and deploying an additional layer of security, namely active command mediation defense [8], that can make the most of tolerable delay to secure the remote control interface of modernized electrical substations. We formulated and demonstrated a procedure to find appropriate command-delaying strategy and also discussed deployment options and prototypes on an embedded platform. Based on our simulation experiments using multiple power grid configurations of different sizes, we showed that, in typical remote control use cases, delaying control commands by 0.7-1.0 seconds is acceptable in terms of negative impact on the power grid stability, which at the same time allows us to expect significant mitigation of attack consequences when the scheme is used with established attack detection systems. Although delay tolerance is different among power grid configurations, our procedure to find acceptable delay for a given power grid model is generally applicable, which guides grid operators to establish an effective command-delaying strategy suitable for their own systems and security preferences.

Our future work includes deployment of the proposed solution in operational smart grid systems to prove its practical effectiveness. We will explore an opportunity for such a pilot testing through collaboration with power grid operators.

Acknowledgment

This research is partly supported by the National Research Foundation, Prime Minister's Office, Singapore under the Energy Programme and administrated by the Energy Market Authority (EP Award No. NRF2014EWT-EIRP002-040) and in part by the research grant for the Human-Centered Cyber-physical Systems Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR). Besides, We greatly appreciate valuable comments from anonymous reviewers to improve the quality of this paper. We also thank Ramkumar Rajendran from National University of Singapore for his contribution to this paper by running simulation experiments.

References

[1] K. Zetter, "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).

[2] D. U. Case, "Analysis of the cyber attack on the ukrainian power grid," 2016.

[3] "Crashoverride malware," [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA17-163A, (Date last accessed on Aug. 18, 2017).

[4] "Tofino firewall lsm," [Online]. Available: https://www.tofinosecurity.com/products/Tofino-Firewall-LSM, (Date last accessed on Jun. 7, 2017).

[5] H. Lin, A. Slagell, C. Di Martino, Z. Kalbarczyk, and R. K. Iyer, "Adapting bro into scada: building a specification-based intrusion detection system for the dnp3 protocol," in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*. ACM, 2013, p. 5.

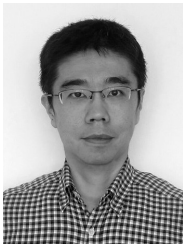
[6] R. Udd, M. Asplund, S. Nadjm-Tehrani, M. Kazemtabrizi, and M. Ekstedt, "Exploiting bro for intrusion detection in a scada system," in *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*. ACM, 2016, pp. 44–51.

[7] K. McLaughlin, "High-level design documentation and deployment architecture for multi-attribute scada intrusion detection system," [Online]. Available: 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).

[8] D. Mashima, P. Gunathilaka, and B. Chen, "An active command mediation approach for securing remote control interface of substations," in *Smart Grid Communications (SmartGridComm), 2016 IEEE International Conference on*. IEEE, 2016.

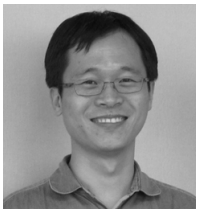
[9] P. Gunathilaka, D. Mashima, and B. Chen, "Softgrid: A software-based smart grid testbed for evaluating substation cybersecurity solutions," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*. ACM, 2016, pp. 113–124.

[10] IEEE Power and Energy Society, "IEEE Standard Communication Delivery Time Performance Requirements for Electric Power Substation Automation."
[11] U. DOE, "Communications requirements of smart grid technologies," *US Department of Energy, Tech. Rep*, pp. 1–69, 2010.
[12] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Communication network requirements for major smart grid applications in han, nan and wan," *Computer Networks*, vol. 67, pp. 74–88, 2014.
[13] W. G. Temple, B. Chen, and N. O. Tippenhauer, "Delay makes a difference: Smart grid resilience under remote meter disconnect attack," in *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*. IEEE, 2013, pp. 462–467.
[14] J. E. Hershey, B. G. Barnett, M. J. Dell'Anno, and D. Thanos, "Intelligent system and method for mitigating cyber attacks in critical systems through controlling latency of messages in a communications network," Sep. 2 2014, uS Patent 8,826,437.
[15] S. Etigowni, S. Hossain-McKenzie, M. Kazerooni, K. Davis, and S. Zonouz, "Just-ahead-of-time controller recovery," in *Smart Grid Communications (SmartGridComm), 2016 IEEE International Conference on*. IEEE, 2016.
[16] 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," *Smart Grid, IEEE Transactions on (to appear)*.
[17] S. Meliopoulos, G. Cokkinides, R. Fan, L. Sun, and B. Cui, "Command authentication via faster than real time simulation," in *Power and Energy Society General Meeting (PESGM), 2016*. IEEE, 2016, pp. 1–5.
[18] "International Electrotechnical Commission," [Online]. Available: http://www.iec.ch, (Date last accessed on Jun. 7, 2017).
[19] J. Hong, Y. Chen, C.-C. Liu, and M. Govindarasu, "Cyber-physical security testbed for substations in a power grid," in *Cyber Physical Systems Approach to Smart Electric Power Grid*. Springer, 2015, pp. 261–301.
[20] IEC TC57, "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.
[21] J. M. Weiss, "Control systems cyber security—the need for appropriate regulations to assure the cyber security of the electric grid," in *US Congress Testimony, October*, 2007.
[22] M. J. Assante and R. M. Lee, "The industrial control system cyber kill chain," *SANS Institute InfoSec Reading Room*, 2015.
[23] "CRASHOVERRIDE: Analysis of the threat to electric grid operations," [Online]. Available: https://dragos.com/blog/crashoverride/CrashOverride-01.pdf, (Date last accessed on Aug. 18, 2017).
[24] P. Maynard, K. McLaughlin, and B. Haberler, "Towards understanding man-in-the-middle attacks on iec 60870-5-104 scada networks," in *Proceedings of the 2nd International Symposium on ICS & SCADA Cyber Security Research 2014*. BCS, 2014, pp. 30–42.
[25] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, E. G. Im, Z. Yao, B. Pranggono, and H. Wang, "Man-in-the-middle attack test-bed investigating cyber-security vulnerabilities in smart grid scada systems," in *Sustainable Power Generation and Supply (SUPERGEN 2012), International Conference on*. IET, 2012, pp. 1–8.
[26] D. Perez and J. Pico, "A practical attack against gprs/edge/umts/hspa mobile data communications," *Black Hat DC*, 2011.
[27] "Characteristics of circuit breaker trip curves and coordination," [Online]. Available: http://testguy.net/content/197-Characteristics-of-Circuit-Breaker-Trip-Curves-and-Coordination?s=c79092656bc492c180b1b34c86af05fe, (Date last accessed on Jun. 7, 2017).
[28] "Time-current curves," [Online]. Available: https://ewh.ieee.org/r4/iail/Time-Current%20Curves.pdf, (Date last accessed on Jun. 7, 2017).
[29] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, *Moving target defense: creating asymmetric uncertainty for cyber threats*. Springer Science & Business Media, 2011, vol. 54.
[30] P. Gopalakrishnan, J. Thomas, and F. Ka, "Introducing protocol converter in a substation communication environment for iec 61850 compatibility," *Kalka Communication Technologies Private Limited, INDIA*, 2008.
[31] "Communication gateway," [Online]. Available: https://library.e.abb.com/public/9d37bb56b86d31afc125746d004b534f/COM610_tob_755425_ENf.pdf, (Date last accessed on Jun. 7, 2017).
[32] Bueno Electric, "Iec-61850 gateways," [Online]. Available: http://www.buenoptic.net/iec-61850-gateways, (Date last accessed on Jun. 7, 2017).
[33] "OpenMUC," [Online]. Available: https://www.openmuc.org, (Date last accessed on Jun. 7, 2017).
[34] J. D. Glover, M. S. Sarma, and T. Overbye, *Power system analysis and design*. China Machine Press, 2004.
[35] "Frequency response," [Online]. Available: http://www2.nationalgrid.com/uk/services/balancing-services/frequency-response/, (Date last accessed on Jun. 7, 2017).
[36] "Illini 42 tornado," [Online]. Available: http://icseg.iti.illinois.edu/illini-42-tornado/, (Date last accessed on Jun. 7, 2017).
[37] "Texas 2000-june 2016," [Online]. Available: http://icseg.iti.illinois.edu/synthetic-power-cases/texas2000-june2016/, (Date last accessed on Jun. 7, 2017).
[38] "PowerWorld," [Online]. Available: http://www.powerworld.com/, (Date last accessed on Jun. 7, 2017).
[39] "WECC-0100 proposed transient voltage criteria," [Online]. Available: https://www.wecc.biz/Reliability/WECC-0100%20Proposed%20Transient%20Voltage%20Criteria.docx, (Date last accessed on Jun. 7, 2017).
[40] "ZNX 202," [Online]. Available: http://www.mashima.us/daisuke/files/20161123072448.pdf, (Date last accessed on July. 11, 2017).



Daisuke Mashima received his Ph.D. degree in Computer Science from Georgia Institute of Technology in 2012. He is currently a research scientist at the Advanced Digital Sciences Center (ADSC) in Singapore, where he is working on smart grid security research. Before joining ADSC, he worked as a member of research staff in the smart energy group at Fujitsu Laboratories of America, Inc. His research interest covers cybersecurity and privacy in cyber-physical systems in general.

Prageeth Gunathilaka is a senior software engineer at Advanced Digital Sciences Center (ADSC) and has been leading development of an open-source, software-based smart grid testbed, a cybersecurity risk assessment tool, and so forth. He earned his M.Sc degree in Artificial Intelligence from University of Moratuwa, Sri Lanka in 2014. Before joining ADSC, he worked as a software engineer as well as a team leader for Paypal and CodeGen International.



Binbin Chen received his B.S. from Peking University, China, in 2003 and his Ph.D. from the National University of Singapore in 2010. He is currently a Senior Research Scientist at the Advanced Digital Sciences Center (ADSC), a research center of the University of Illinois located in Singapore. His current research interests include wireless networks, cyber-physical systems, applied algorithms in networking, and network security. More information about his research can be found at http://adsc.illinois.edu/people/binbin-chen.