On Optimization of Command-Delaying for Advanced Command Authentication in Smart Grid Systems

Daisuke Mashima[1], Ramkumar Rajendran[2], Toby Zhou[1], Binbin Chen[1], and Biplab Sikdar[2]

***Abstract*— Attacks from the control center or intermediate devices in the SCADA communication infrastructure was outside of the scope of traditional power grid operation. To counter such emerging threats, which were actually witnessed in the Ukraine incidents in 2015 and 2016, command authentication mechanisms are considered as an effective security measure. Command authentication performs context-aware checking of validity or legitimacy of incoming remote control commands near the edge of the cyber infrastructure of smart grid systems (e.g., at substations). While effective, in order to flag potentially malicious commands before they impact physical power grid systems, one of the major design issues for such schemes is the extended latency required to run an authentication algorithm. While some efforts have been made to utilize artificial delay to put commands on hold to wait for the completion of the command authentication algorithm, they delay command execution in a probabilistic manner. However, such a solution does not result in optimal security gain. In this paper, we develop a systematic way to find optimal delaying configurations. Our formulation of the optimization problem takes both potential damage caused under attack and negative impact on normal operation into consideration. We then discuss and evaluate a heuristic, efficient algorithm to find near-optimal solution for practical operation.**

I. Introduction

Securing communication infrastructure of smart grid systems is an impending necessity we need to address. While modernization of power grid systems has been in progress in recent years, because of the assumption of the "gap" that isolates smart grid systems from the rest of the world, security has not been the highest priority for power grid operators. As a result, unfortunately, we have witnessed a number of real cyber attacks targeting power grid systems, for example the Ukraine incident in 2015 [1] and another in 2016. In both incidents, by compromising the control center system or by using malware such as CrashOverride [2], remote control interfaces were exploited to cause massive outages. Even more recently, in 2017 it was reported that hackers succeeded in penetrating into the control rooms of US utility companies [3].

To counter such threats, an additional line of defense near the boundary of cyber and physical systems of smart grid infrastructure is needed, and *command authentication* systems are regarded as a promising solution that effectively complements other cybersecurity solutions like industrial firewall and intrusion detection systems. Such an additional security layer can be deployed near the cyber-physical boundary of smart grid infrastructure, for example on intelligent electronic devices (IEDs), programmable logic controllers (PLCs) [4], or, more practically, on a substation gateway as proposed in [5], to assess the legitimacy of remote control commands based on the power system context. This way, even when a control center or any other upstream entity in the SCADA command communication path becomes malicious or manipulated by an adversary, malicious commands can be blocked before they actually impact the physical power grid. State-of-the-art command authentication mechanisms rely on information derived out of power flow simulation [6], [7]. Specifically, on-the-fly power-flow simulation based on the up-to-date power grid status can be run to evaluate the consequence of each control command. If the command causes any stability issues, the corresponding command can be flagged as malicious (or anomalous) and therefore be blocked before impacting the physical power grid. While being effective, such advanced command authentication schemes take time to complete their job. In particular, power system dynamics simulation employed in [7] requires approximately 1 second even with relatively small scale power grid models. In order to complete the command authentication before execution of commands, it is necessary to put the command on hold for a sufficient duration. Such an artificial command-delaying mechanism was explored in [5], [8], but as discussed by the authors, applying delay to all commands would violate the latency requirement on power grid operations [9], [10] and also may cause power grid stability issues.

One straightforward, practical solution to balance the security gain and latency and stability requirements is to introduce delay in a probabilistic manner, aiming at blocking at least part of the attacks for mitigation while avoiding negative impact on normal operations [5]. However, the probabilistic approach does not necessarily result in optimal results. Therefore, in this paper we formulate a problem for finding an optimal command-delaying strategy, and further propose an efficient algorithm to solve it, which is desired because in practice delaying configurations often need to be modified frequently according to the changes in smart grid infrastructure. Our contributions include:

1) Formulation of a framework to configure the optimal command-delaying strategy
2) Design of a heuristic algorithm to solve the problem in an efficient manner
3) Simulation study for evaluating the optimality of the heuristic solution

The rest of this paper is organized as follows. In Sec-

[1]Advanced Digital Sciences Center, {daisuke.m, binbin.chen, zhou.bin}@adsc-create.edu.sg
[2]National University of Singapore, e0154173@u.nus.edu, bsikdar@nus.edu.sg

tion II, we discuss related work, including an overview of SCADA command authentication systems and an active command mediation defense framework with artificial command-delaying. Section III discusses the requirements for and options of practical command-delaying strategies. Then Section IV describes the framework for identifying a security-optimal command-delaying strategy, followed by an efficient, heuristic algorithm for finding the solution in Section V. Simulation study on command authentication accuracy and search of command-delaying configuration is presented in Section VI. Finally, we conclude the paper in Section VII.

II. Related Work

A number of cybersecurity solutions for smart grid systems have been proposed, including intrusion detection systems [11], [12], [13], [14] and mechanisms for securing SCADA messages [15], [16], [17], [18], and security standards for SCADA communication protocols, such as IEC 62351 [19]. While such schemes are effective for ensuring security at the network layer as well as message authenticity and integrity, they do not take physical system information and context into consideration. Therefore they are not fully effective once a trusted SCADA master system in the control center [1] or any component on the trusted communication infrastructure [2] becomes malicious.

Command authentication schemes utilize physical power grid status and context as well as a variety of safety conditions to evaluate the legitimacy of SCADA control commands. Therefore, they are complementary to the security solutions discussed earlier and can remain effective even under Ukraine-like situations [1]. Reference [6] utilizes distributed state estimation and simulation for detecting malicious commands. However, the proposed scheme requires intensive communication among peer substations in proximity, which may expand the attack surface against the proposed security solution. Reference [20] utilizes distributed sensors as well as a centralized attack detection system running power flow simulation. Their scheme exhibits high accuracy but its mitigation strategy is reactive. Besides, these schemes rely only on steady-state power flow information, whose limitation has been pointed out by [7], [21]. Related to command authentication systems, reference [4] proposes a proactive solution that mediates all control logic uploaded to PLCs for security assessment. However, it is designed for securing a specific PLC model and therefore requires customization efforts in multi-vendor environment. To address such limitations and enhance attack detection accuracy, use of power system dynamics simulation, which additionally evaluate transient-state behaviours and cascading failures, for command authentication has been recently proposed [7]. While being promising, one technical challenge for practical operation is its extended latency required for simulation.

In order to accommodate the latency for advanced power grid security, the idea of making the most of tolerable time delay for command authentication is discussed in an active command mediation defense (A*CMD) system [8], and detailed design consideration on the delay is elaborated in [5]. A*CMD system [8] can be deployed on the gateway of each substation, which is responsible for protocol translation (e.g., from IEC 60870-5-104 to IEC 61850 and vice versa) and therefore can reliably mediate all incoming remote control commands. When receiving any control command from the control center, each instance of A*CMD probabilistically adds a certain amount of artificial time delay (typically in the order of 100ms [5]) with a certain, pre-configured, probability. Such a command-delaying strategy is called *discrete-random-delay* in the same paper. Artificial delay allows the attack detection system that is deployed either locally on the A*CMD system or centrally at the control center, as done in [20], to complete its job and neutralize suspected commands before execution.

In order to configure the parameters for the discrete-random-delaying strategy, crucial consideration is needed to find delay tolerance, which corresponds to the amount of artificial delay that can be added without negative impact on power grid stability. The procedure for finding it for a power grid of interest are discussed in [5]. However, there is still a challenge in finding the optimal delaying strategy that maximizes the security gain under power system stability constraints. As demonstrated later in Section VI, a probabilistic approach will not result in the best result, which is the main motivation for this paper.

III. Options for Practical Command-delaying

Command authentication systems are intended to be an additional line of defense for mitigating the negative impact of malicious commands even when the SCADA master system in the control center system is compromised. Therefore, it is often implemented on the distributed field system in the smart grid, and command-delaying configurations should be configured on distributed security modules so that each instance can introduce artificial delay in an autonomous way. If the command-delaying functionality would require coordination with peer field devices and/or communication with the control center, it would result in not only additional delay but also broadened attack surface.

Because the command authentication module cannot tell whether the incoming remote control commands are legitimate or not when they are received, command-delaying strategy needs to be applied to all commands, regardless of whether they are actually malicious or not. One viable way to autonomously introduce an artificial delay is to let each security module add the delay in a probabilistic way. In this case, the system operator can pre-configure the probability of command-delaying so that, after deployment, each security module can make the autonomous command-delaying decision according to the probability. The operator could configure a system-wide probability that is to be enforced on all types of control commands. Instead, it is also possible to configure different delaying probability per command type. For instance, control commands for load shedding may have different delaying probability from those

for generator control. Delaying probability can be decided based on contingency simulation at the design phase [5].

While easy-to-deploy, the probabilistic approach would not always result in an optimal security gain. Another option is to apply artificial delay only to control commands targeting the selected resources or power grid devices. In other words, the system operator can design and pre-configure the strategy (i.e., which control commands are subject to artificial delay in terms of, for example, target devices). Such a design could be made by operators based on their experience and preference, but again it would not result in optimal outcome, because operators would be often too conservative. Thus, we formulate the optimization problem for finding an optimal command-delaying strategy that maximizes security gain without causing power grid instability.

IV. Problem Formulation for Finding Optimal Command-Delaying Strategy

In smart grid systems, remote control is introduced and utilized for a number of purposes, such as power shedding to curtail over-generation of renewables, load shedding for addressing generation-loss contingency, voltage regulation though shunt reactor control, topology control for economic optimization, and so forth. Out of these controls, load shedding is considered not only the most time-sensitive but also influential for customer experience as well as revenue of utility companies. Therefore, discussion in this section is focused on optimal command-delaying strategy for load-shedding controls. However, the applicability of the discussed concept is not limited to such scenarios.

For contingencies like generator fault, one of the typical recovery control strategies is load shedding to balance supply and demand. Typically, utility companies carefully simulate and prepare for all expected contingencies (e.g., $N-1$ contingencies or, more generally, $N-x$ contingencies). The loads to be shed are planned in a number of ways. In some cases, a set of loads is selected based on proximity to the source of contingency (e.g., a lost generator). Alternatively, a more advanced scheme, such as [22], which optimizes the number of loads to be shed based on the type of the contingency, the cost of the load changes and the curtailment charges, is employed in the contingency planning phase. Therefore, we assume that all contingencies to be taken into account, C_j $(j \in \{1,2,3...m\})$, and the set of loads to be shed for each contingency (L_j) are given. We further denote the superset of L_js as N, i.e., $N = L_1 \cup L_2 \cup L_3 \cup \ldots \cup L_m$, and its cardinality as $|N|$. We also denote each load included in N as l_i where $i \in \{1,2,3...,|N|\}$.

The control variable k_i $(i \in \{1,2,3...,|N|\})$ represents the amount of artificial delay to be added to load shedding control commands targeting each load in N. In the case of discrete-random-delay strategy, k_i has a discrete value $k_i \in \{0, D_{ub}\}$ where D_{ub} is the duration of artificial delay, which is given. $k_i = 0$ means control for the corresponding load is not delayed, while $k_i = D_{ub}$ means that the corresponding load-shedding control are delayed by D_{ub}. D_{ub} can be found by various methods, such as rigorous simulation, procedure formulated in [5], or determined based on the specification (e.g., processing time) of attack detection systems employed. For the sake of distributed and autonomous deployment throughout the smart grid infrastructure, the k_is are supposed to be pre-programmed on security devices in the field (e.g., substation gateways or protocol translators). Then, our goal is to find the optimal set of k_is, denoted by K, such that the following conditions are met. Intuitively, the objective is to maximize the security gain by adding delay to the maximal number of influential loads. Influence of loads (called *impact factor* s_i $(i \in \{1,2,3...,|N|\})$) can be determined based on the size and/or criticality of the load, which are assumed to be decided in advance and given as input. For the evaluation in Section VI, we utilize the size of the load as impact factor, but non-trivial definition on impact factor is part of our future work. In sum, the objective is formulated as maximization of the *security score* below:

$$Security\ Score = \sum_{j=1}^{m} \sum_{i \in L_j} s_i k_i$$

Note that, when we delay control commands for more loads with high impact factor (and thereby protect them from malicious commands), the score becomes larger.

In other words, when Z is defined as a $m \times |N|$ matrix where an element in jth row, corresponding to jth contingency C_j, z_{ij} is set to be k_i when the ith load in N is included in L_j and $z_{ij} = 0$ otherwise. This is equivalent to the maximization of product ZS where S is a $|N|$-dimensional vector consisting of s_i. This maximization must be done under the power grid stability constraints. For $\forall j$,

$$g(C_j, L_j, K, SC, PG) = 0$$

where $g()$ is the function to evaluate grid stability based on the given power grid models and configurations (PG) and required stability criteria (SC), such as threshold for voltage and frequency. $g()$ returns 0 when the grid results in stable state under contingency C_j when loads in L_j are controlled with the delaying configuration K and returns 1 otherwise. In practice, $g()$ can be implemented based on extensive power flow simulation, such as the power system

TABLE I
Description of Symbols Used in Formulation

Symbol	Description
m	The total number of contingencies
C_j	jth contingency
L_j	Set of resources (e.g., loads) to be controlled as recovery measures for jth contingency
$\|N\|$	Total number of resources for all the contingencies
k_i	The delay value for ith resource
K	The set of delay values for all resources
D_{ub}	Upper bound of artificial time delay
s_i	The impact factor of ith resource
g	The system stability function
PG	Power grid model and topology
SC	Power grid stability conditions

TABLE II
POWER SYSTEM DYNAMICS SIMULATION LATENCY WITH VARYING COMPLEXITY OF MODELS

Model	Equivalent Model Size	Duration [sec]	Latency [ms]
37-bus [24]	37 buses	30	458
	23 buses	30	298
	11 buses	30	151
2000-bus [23]	2,007 buses	30	9,134
	1,132 buses	30	5,041
	447 buses	30	1,684
2000-bus [23]	2,007 buses	10	3,083
	1,132 buses	10	1,645
	447 buses	10	578

dynamics simulation used in [7]. In other words, we aim at maximizing the total security gain over m contingencies without violating any grid stability conditions. K identified through this optimization can be statically configured or pre-programmed on each security device so that the delay settings are autonomously enforced without requiring any run-time coordination.

In the formulation based on the discrete-random-delay strategy in [8], each k_i is binary (i.e., 0 or D_{ub}), and thus the problem is regarded as an integer programming problem, which is known to be NP-hard. In other words, we need to try all combinations of k_is and evaluate $g()$ for each combination. Therefore, the required complexity is $O(2^{|N|})$. Given that the power system dynamics simulation is time-consuming as shown in Table II, the time complexity when $|N|$ is large can be significant. For instance, if we run a simulation for the 2000-bus power grid model with full fidelity, it would take over 9 seconds. When $|N|$ is the order of 100 (out of 1,417 loads in the 2000-bus model [23], for example), the time to find the truly optimal solution would become intractable. Thus, we may need to consider to lower the model complexity (using Thevenin equivalence, for example) and/or to shorten the simulation duration, both of which would affect the correctness of the result.

V. EFFICIENT ALGORITHM FOR SCALABILITY

When the grid size is large, it is not feasible to evaluate all possible combinations of delay settings. Moreover, a similar procedure should be executed for other types of contingencies in practice. If finding K would be only a one-time task before system roll-out, a brute-forcing approach to find the optimal setting may be possible. However, in practice, power grid configuration or topology can be often modified either temporarily or permanently, which is likely the case with increasing integration of renewables and distributed energy resources. Thereby, contingency simulations may need to be redone whenever any change occurs. In order to facilitate the update of command-delaying strategies under such a circumstance, in this section, we will discuss an efficient, heuristic approach to find the near-optimal solution.

Our proposed practical algorithm is shown in Algorithm 1. In the algorithm, $countOccurrence()$ is a function to count how many contingencies require shedding of each load in N. The count for each load is used for determining priority (or weight), w_i, when deciding which load to shed without delay. f_i is a factor provided as part of input that is decided based on the influence of load l_i on the power grid service, and it can be decided, for instance, based on MW of l_i. The high-level idea for the rest of the algorithm is as follows:

1) Start K that indicates all loads are delayed by D_{ub}.
2) Evaluate $g()$ for each contingency one by one, until finding a contingency that results in $g() = 1$.
3) For that contingency, change K so that the load with the lowest impact factor is shed immediately. Repeat this step until $g()$ returns 0.
4) Using the updated K, continue evaluation for the remaining contingencies. When encountering a contingency with $g() = 1$, do the same as the previous step to update K.

$findMinimum()$ returns the index of a load with minimal weight among loads that are currently set to be delayed (i.e., $k_i \neq 0$). If $k_i = 0$ for all loads already, this function returns 0, which implies there is no solution that satisfies stability constraints. Performing recovery load shedding immediately is expected not to have a favorable outcome in terms of grid stability compared to the situation where the same control is delayed. Thus, we don't need to re-evaluate the already-passed contingencies. Therefore, for each contingency, at most $|L_j|$ iterations are needed. The limitation of this algorithm is that this is adding a load to be shed incrementally and therefore may not be reaching the global optimum.

Algorithm 1 Heuristic Approach

Require: $PG \leftarrow$ Power grid model and topology
Require: $SC \leftarrow$ Power grid stability conditions
Require: $CTG \leftarrow \{C_1, C_2, \ldots, C_m\}$
Require: L_j $(j \in \{1, \ldots, m\}) \leftarrow$ Loads to be shed for C_j
Require: $d \leftarrow$ Duration of artificial delay
  $N \leftarrow L_1 \cup L_2 \cup \ldots \cup L_m$ (i.e., $N \leftarrow \{l_i\}$ $i \in \{1, \ldots, |N|\}$)
  $K \leftarrow |N|$-dimentional vector initialized with d $(\{k_i\})$
  $F \leftarrow |N|$-dimentional vector initialized with 0, $(\{f_i\})$
  $W \leftarrow |N|$-dimentional vector initialized with 0, $(\{w_i\})$
  for each i in $\{1, \ldots, |N|\}$ **do**
    $f_i \leftarrow countOccurrence(l_i, \{L_1, \ldots, L_m\})$
    $w_i \leftarrow s_i \times f_i$
  end for
  for each j in $\{1, \ldots, m\}$ **do**
    if $g(C_j, L_j, K, SC, PG) = 0$ **then**
      proceed to next contingency
    else
      $stability = FALSE$
      while $stability \neq TRUE$ **do**
        $idx \leftarrow findMinimum(L_j, W, K)$
        if idx = 0 **then**
          return NULL
        end if
        $k_{idx} \leftarrow 0$
        $stability \leftarrow g(C_j, L_j, K, SC, PG)$
      end while
    end if
  end for
  return K

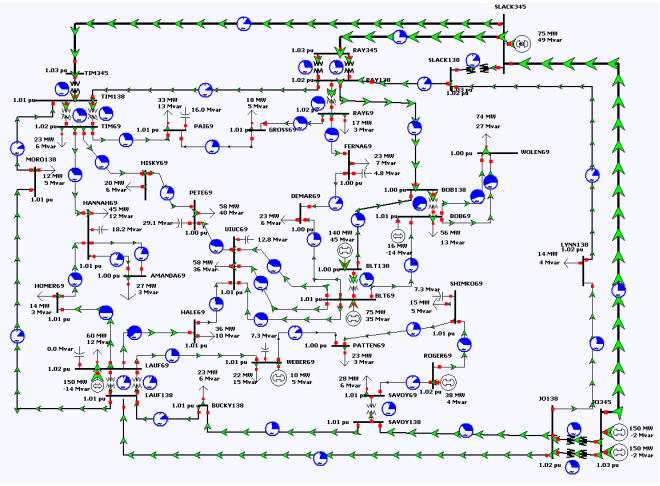

Fig. 1. GSO 37-bus System Overview

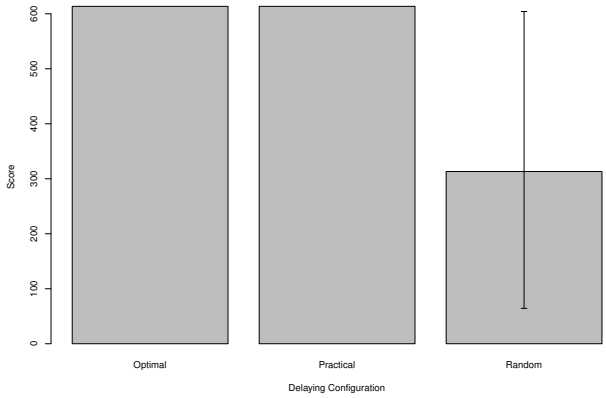

Fig. 2. Security Score over Different Delaying Configurations (37-bus system with 1-second delay).

TABLE III
CONTINGENCIES AND RECOVERY CONTROLS FOR 37-BUS SYSTEM [24]

j	Lost Generator	Loads to Be Shed (Size in MW)
1	JO345 #1	LYNN138(14), RAY69(16.8), BUCKY138(23.4), SAVOY69(28), LAUF69(59.8)
2	ROGER69	Nil
3	BLT138 #1	UIUC69(58.2), WOLEN69(74.4)
4	LAUF69	HOMER69(14), WEBER69(22.2), BUCKY138(23.4), AMANDA69(27), LAUF69(59.8)
5	JO345 #2	LYNN135 (14), RAY69(16.8), BUCKY138(23.4), SAVOY69(28), LAUF69(59.8)
6	BOB69	Nil
7	BLT69 #1	DEMAR69(22.65), BOB69(55.8)
8	WEBER69 #1	Nil

VI. SIMULATION STUDY

In this section, we demonstrate and evaluate, by using a power grid model of a manageable size that allows us to perform manual brute-forcing to find the optimal K, how the approach to find optimal command-delaying configuration works. In particular, we use GSO 37-bus system [24] (see Fig 1), which has 9 generators and 25 loads.

We use the PowerWorld simulator [25] to simulate all $N-1$ contingencies focusing on generator-loss. (We excluded a generator connected to a slack bus, so there are 8 in total). Among them, three contingencies did not cause any violation even without recovery controls, which have "Nil" in the last column of Table III, and are excluded from the following experiment. For each generator-loss contingency, we designed a set of load shedding controls based on the proximity to the contingency. The list of contingencies and the set of loads to be shed for each contingency are summarized in Table III. This list of load shedding controls is commonly used for both the brute-forcing approach and practical solution for the sake of comparison. As the priority of each load, we used the size of each load in MW, also shown in Table III. In addition, for the experiments in this section, we considered a situation where we need to delay loads by 1 second. (Note that, by executing the algorithm to find delay tolerance defined in [5], delaying up to 0.9-second was found to be acceptable, and therefore delaying 1 second requires to solve the optimization problem).

Based on our experiments, the best security score found in the brute-forcing approach, which can be considered as a truly optimal score, is 613.52 when only load LYNN138 is configured to be shed immediately while the others are delayed by 1 second. Through the simulation following Algorithm 1, we have reached the same configuration. Regarding the computational cost, in terms of the number of evaluations of $g()$, our heuristic approach needed 6 executions in total.

For the sake of comparison, we also computed the approach deciding whether to add delay or not purely probabilistically (i.e., equivalent to the approach taken in [5]). Through our preliminary experiment on the first contingency of Table III, we know that at least one load needs to be shed immediately to avoid violation. Thus, we set the delaying probability for the simulation to be 50%, which ensures with 97% probability that at least 1 out of 5 control commands are executed without delay. In addition, owing to the randomness, we repeated the same experiment 10 times. According to the simulation results, highest security score observed was 604.12, and the lowest one was 64.40 (see the error bar), while the average score was 313.20. The results are summarized in Fig. 2. Thus, we can see that the solution found by Algorithm 1 provides better security gain overall, compared to the probabilistic approach.

Using the same 37-bus system, we did the same experiment but assuming the case where longer delay, namely 1.2 second, is required. Because of the longer delay, it is expected that more loads need to be shed immediately in order not to cause any violation. Regarding loads to be shed for each contingency, we again considered the ones shown in Table III. Algorithm 1 resulted in deciding 4 loads to be shed immediately. The resulting score was 576.78. On the other hand, the optimal solution found by brute-forcing chose 3 loads to be shed immediately, which resulted in score

582.06. These two are still close enough, so we consider that Algorithm 1 works well with significantly lower computational cost. Note that, we don't use the score for comparison among different experiments, e.g., with the experiment with 1-second delay discussed above, since impact of duration of delay is in general not proportional or monotonic.

Since finding ground-truth optimal configuration requires brute-forcing of all possible configurations, we limit our evaluation in this paper a to relatively small power grid model. Evaluation with larger power grid models, such as one with thousands of buses [23], as well as evaluation with real-world power grid models will be part of the future work.

VII. Conclusions

In order to counter cyber-originated remote control command injection attacks targeting power grid systems that we witnessed in recent years, command authentication schemes are considered an effective additional line of defense. While state-of-the-art command authentication schemes that utilize power-flow simulation can effectively detect and prevent attacks, such advanced authentication algorithms require longer latency to make the decision. In order to accommodate the extended latency and block malicious commands before they impact the physical system, remote control commands need to be put on hold to wait the authentication results. In this direction, we formulated a problem for finding a security-optimal command-delaying strategy under power-grid stability constraints. We also discussed a heuristic algorithm that can efficiently find a near-optimal solution. Based on our experiments, compared to the solution that probabilistically applies artificial delay according to a pre-configured probability, the proposed solution allows us to expect a higher security gain. In future work, we plan to evaluate the effectiveness of the proposed framework in the practical settings, ideally by partnering with real-world power grid operators.

Acknowledgment

This research is supported in part by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme and is also 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. NRF2017EWT-EP003-047).

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

[3] "Russian hackers reach u.s. utility control rooms, homeland security officials say," [Online]. Available: https://www.wsj.com/articles/russian-hackers-reach-u-s-utility-control-rooms-homeland-security-officials-say-1532388110, (Date last accessed on Jan. 31, 2019).

[4] 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.

[5] D. Mashima, P. Gunathilaka, and B. Chen, "Artificial command delaying for secure substation remote control: Design and implementation," *IEEE Transactions on Smart Grid*, 2017.

[6] 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.

[7] D. Mashima, B. Chen, T. Zhou, R. Rajendran, and B. Sikdar, "Securing substations through command authentication using on-the-fly simulation of power system dynamics," in *Smart Grid Communications (SmartGridComm), 2018 IEEE International Conference on*. IEEE, 2018.

[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] IEEE Power and Energy Society, "IEEE Standard Communication Delivery Time Performance Requirements for Electric Power Substation Automation."

[10] US Department of Energy, "Communications requirements of smart grid technologies," *US Department of Energy, Tech. Rep*, pp. 1–69, 2010.

[11] 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.

[12] 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).

[13] Y. Yang, H.-Q. Xu, L. Gao, Y.-B. Yuan, K. McLaughlin, and S. Sezer, "Multidimensional intrusion detection system for iec 61850-based scada networks," *IEEE Transactions on Power Delivery*, vol. 32, no. 2, pp. 1068–1078, 2017.

[14] W. Ren, T. Yardley, and K. Nahrstedt, "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, 2018, pp. 1–7.

[15] J. H. Castellanos, D. Antonioli, N. O. Tippenhauer, and M. Ochoa, "Legacy-compliant data authentication for industrial control system traffic," in *Proceedings of the Conference on Applied Cryptography and Network Security (ACNS)*, July 2017.

[16] L. Zhou, H. Guo, D. Li, J. Zhou, and J. Wong, "A scheme for lightweight scada packet authentication," 2017.

[17] P. P. Tsang and S. W. Smith, "Yasir: A low-latency, high-integrity security retrofit for legacy scada systems," in *IFIP International Information Security Conference*. Springer, 2008, pp. 445–459.

[18] A. K. Wright, J. A. Kinast, and J. McCarty, "Low-latency cryptographic protection for scada communications," in *International Conference on Applied Cryptography and Network Security*. Springer, 2004, pp. 263–277.

[19] F. Cleveland, "Iec tc57 wg15: Iec 62351 security standards for the power system information infrastructure," *White Paper*, 2012.

[20] H. Lin, A. Slagell, Z. Kalbarczyk, P. W. Sauer, and R. K. Iyer, "Semantic security analysis of scada networks to detect malicious control commands in power grids," in *Proceedings of the first ACM workshop on Smart energy grid security*. ACM, 2013, pp. 29–34.

[21] C.-W. Ten, K. Yamashita, Z. Yang, A. Vasilakos, and A. Ginter, "Impact assessment of hypothesized cyberattacks on interconnected bulk power systems," *IEEE Transactions on Smart Grid*, 2017.

[22] X. Lou, D. K. Yau, H. H. Nguyen, and B. Chen, "Profit-optimal and stability-aware load curtailment in smart grids," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1411–1420, 2013.

[23] "Texas 2000-june 2016," [Online]. Available: http://icseg.iti.illinois.edu/synthetic-power-cases/texas2000-june2016/, (Date last accessed on Jun. 7, 2017).

[24] J. D. Glover, M. S. Sarma, and T. Overbye, *Power system analysis and design*. China Machine Press, 2004.

[25] "PowerWorld," [Online]. Available: http://www.powerworld.com/, (Date last accessed on Jun. 7, 2017).