Securing Substations through Command Authentication Using On-the-fly Simulation of Power System Dynamics

Daisuke Mashima, Binbin Chen, Toby Zhou
Advanced Digital Sciences Center
Singapore
{daisuke.m, binbin.chen, zhou.bin}@adsc-create.edu.sg

Ramkumar Rajendran, Biplab Sikdar
National University of Singapore
Singapore
e0154173@u.nus.edu.sg, bsikdar@nus.edu.sg

***Abstract*—There are increasing concerns that cyber attackers may inject malicious remote control commands into smart grid systems, as witnessed in the Ukraine incidents in 2015 and 2016 and the recent *CrashOverride* malware campaign. To counter such risks, command authentication mechanisms, which evaluate the legitimacy and validity of each remote control command based on the up-to-date power grid status and context, can be deployed near the edge of smart grid infrastructure (e.g., in substations) as an additional line of defense. However, many of the state-of-the-art command authentication schemes only utilize steady-state power flow information, which does not capture all details of power grid behaviors in the transient state as well as cascading effects. Therefore, they may overlook indication of significant grid instability triggered by malicious commands. In this paper, we propose the use of on-the-fly power system dynamics simulation for command authentication to overcome such limitations. We also discuss system architecture and design considerations on longer simulation latency towards the practical deployment of the enhanced command authentication system.**

I. Introduction

In the recent years, we have witnessed a number of real cyber attacks targeting power grid systems. For instance, in the Ukraine incident in 2015 [1], a computer at the control center was hacked and remotely manipulated by an attacker to issue a large number of circuit breaker open commands. More recently, in 2017, CrashOverride malware [2], which has the capability to impersonate IEC 60870-5-104 server and cause damages to power grid operations by injecting malicious control commands, was reported.

Under such threats, the traditional security model of smart grid systems, where a control center and communication infrastructure between the control center and field systems are secure, is no longer valid, and additional layer of defense has been highly demanded. In this direction, command authentication schemes, which enable each field system near the edge of the infrastructure, e.g., a substation, to verify legitimacy of incoming remote control commands have been proposed [3], [4], [5]. Such a scheme is considered effective not only to counter malicious control commands injected by internal or external attackers but also to prevent execution of inappropriate control commands triggered by system malfunction or human errors, such as an incident reported in [6]. To perform such validation, state-of-the-art schemes rely on power flow simulation. However, to our knowledge, the command authentication schemes currently proposed only utilize steady-state information. In general, steady-state simulation is quick and suitable for real-time operation. However, it has limitations in capturing some key indications of power grid instability caused by execution of malicious or erroneous control commands. For instance, steady-state simulation is unable to capture violation, such as over/under-frequency violation, occurring during transient state (i.e., behaviors between steady states). In addition, steady-state simulation does not take cascading failures into consideration [7]. An example demonstrating such a potential limitation will be presented later in Section III.

In this paper, we discuss the use of on-the-fly power system dynamics (or system dynamics for short) simulation, which evaluates transient-state behaviors as well as cascading effects, to enhance the capability of command authentication mechanisms. An heuristic algorithm using simulation results is proposed, and its accuracy (false positive and negavie) is evaluated. We further elaborate on practical considerations for introducing the advanced security scheme into the smart grid systems. One of the key technical issues we need to address is the latency to run system dynamics simulation, which is longer than attack detection or command authentication using steady-state simulation [4], [5]. To accommodate required latency, we propose to utilize an artificial command-delaying scheme [3], [8], which delays command execution for a certain duration to buy time to detect malicious/suspicious commands without negatively affecting normal operations. We also discuss the overall system architecture for the integration.

The rest of this paper is organized as follows. In Section II, we discuss related work. We then show potential limitation of command authentication scheme relying on steady-state power flow simulation, which is the current state-of-the-art, in Section III, and propose an alternative mechanism integrating power flow dynamics simulation in Section IV. Section V discusses system architecture and design considerations pertaining to artificial command-delaying. Evaluation using a power flow simulator will be conducted in Section VI. Finally we conclude the paper with future work in Section VII.

II. Related Work

In recent years, given a number of real-world incidents, various cybersecurity technologies for protecting distributed substation systems have been proposed. Some are focused only on cyber-side information and others incorporate both cyber and physical system information.

In the former category, intrusion detection systems that can work with widely-used SCADA protocols, such as DNP3 and IEC 60870-5-104, are proposed [9], [10]. Firewall solutions that protect at the perimeter of substation systems, for instance [11], are also developed as commercial solutions. However, these cyber-oriented solutions are not fully effective when a control center system becomes malicious or compromised. To overcome this limitation, our threat model covers attacks mounted either at control centers or intermediate devices in traditionally trusted smart grid communication infrastructure.

Besides the cyber-side information, other efforts, including [3], [4], [5], [8] utilize physical power grid status as well as safety conditions for securing remote control infrastructure in smart grid systems. Reference [5] utilizes distributed state estimation and simulation for detecting malicious commands. Unfortunately, the proposed scheme requires intensive communication among peer substations in proximity, which may expand attack surface to confuse or mislead the proposed security solution with fake data. Reference [4] employs distributed sensors as well as a centralized command authentication system running power flow simulation. Their scheme exhibits high accuracy but its mitigation strategy is reactive. Moreover, these schemes rely only on steady-state information, whose limitation has been pointed out by Ten [7]. To address that limitation and enhance attack detection accuracy, in this paper we discuss utilization of power system dynamics, including transient-state behaviors and cascading effects, for the sake of enhanced command authentication.

Active command mediation defense (A*CMD) proposed in [3], [8] is a framework for securing substation remote control interface. At the high-level, it intercepts all incoming remote control commands for mitigation of physical impact of cyber attacks. The framework can be used with attack detection (and command authentication) mechanisms of both categories discussed above, and, because it supports artificial command-delaying, also is considered effective to practically integrate the command authentication mechanism proposed in this paper, as will be discussed in Section V.

III. Limitation of Steady-state Simulation

We here discuss one example where steady-state simulation does not indicate grid instability while serious problem is observed in system dynamics simulation. We use GSO 37-bus [12] model (Fig. 1) on PowerWorld [13].

As seen in Fig. 1, when the generator BLT69#1 is opened, we do not observe any stability issues in steady state. Even after the generator is opened, no transmission line or transformer is overloaded and bus voltage at all buses is between 0.983 p.u. and 1.030 p.u. On the other hand, if we run a system dynamics simulation for the same contingency, we observed violation of lower frequency limit during the first swing (Fig. 2). As can be seen, after the generator was opened at 1 second, the frequency started dropping rapidly and reached as low as 58.62Hz.

Note that, in this model the nominal frequency is 60Hz and according to [14], Western Electricity Coordinating Council (WECC) Category B and C minimum transient frequency standards are 59.6Hz for 6 cycles (i.e. 0.1 second) and 59.0Hz for 6 cycles respectively. Based on these criteria, although the frequency eventually goes back to around 59.8Hz, we should treat this as a non-negligible stability issue.

IV. Command Authentication Using On-the-fly Power System Dynamics Simulation

Examples in the previous section highlight that there exist potentially malicious commands that cannot be detected by command authentication schemes relying solely on steady-state simulation but can be flagged as suspicious if we use system dynamics simulation. Power system dynamics simulation allows us to simulate detailed behavior of a power system, e.g., frequency swing during the transient state after events that would affect power grid stability, such as outage of some components and (malicious) controls performed, as well as cascading events (e.g., automated protection for line overload and/or out-of-sync generators). However, a way for using it for smart grid security has not been explored. For the rest of this paper, we discuss command authentication using power system dynamics simulation and design considerations.

A. Heuristic Command Authentication Algorithm

One straightforward way of using system dynamics simulation for detecting malicious control commands injected by an attacker (or commands issued by system malfunction or human error) is to run simulation for each control command received by field systems. Namely, following the centralized command authentication framework discussed in [3], [4], each field system, e.g., a digitized substation, reports received remote control commands to the central command authentication system, which then runs simulation for evaluating the consequence of the commands.

More specifically, we can evaluate if the simulation with the command execution causes any (additional) violation of power grid stability conditions or whether the command execution would make the situation worse, compared to the case without the command execution. In this section, we discuss a heuristic scheme that utilizes on-the-fly system dynamics simulation for authenticating remote control commands. Algorithm 1 (Alg. 1 for short hereafter) describes this command authentication approach in more formal way. The command authentication module, which is assumed to be centrally deployed (e.g., in the control center), is invoked whenever remote control commands (cmd_{new}) are reported by field systems. PG represents power grid topology along with up-to-date snapshot of power grid status required to run power system dynamics simulation ($DynSim()$). If there are preceding command(s) or event(s) to be jointly simulated under PG, $event_{pre}$ should be set accordingly. In practice, PG can be updated once in every few

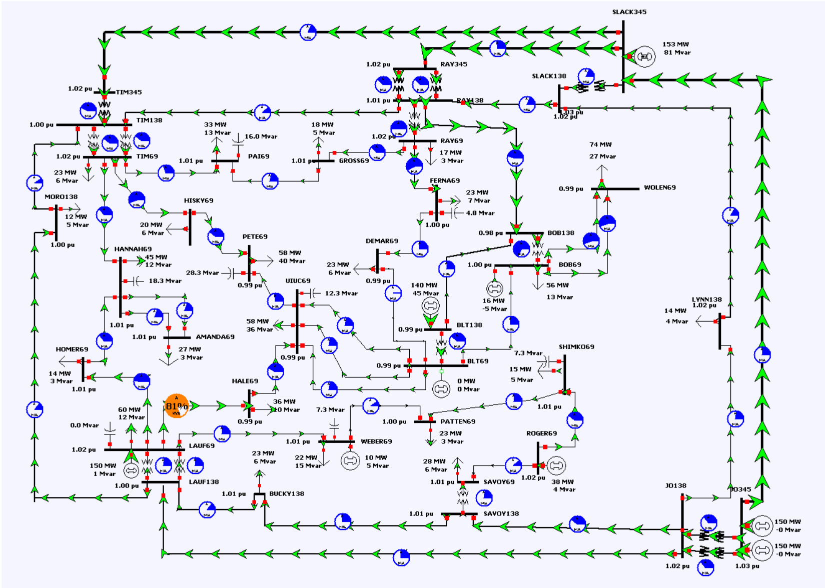

Fig. 1. GSO 37-bus System Overview [12] (Screenshot on PowerWorld [13]). This also shows the result of steady-state power flow calculation after a generator is opened. The highest line load is still 81%, which is shown with yellow color (See Section III).

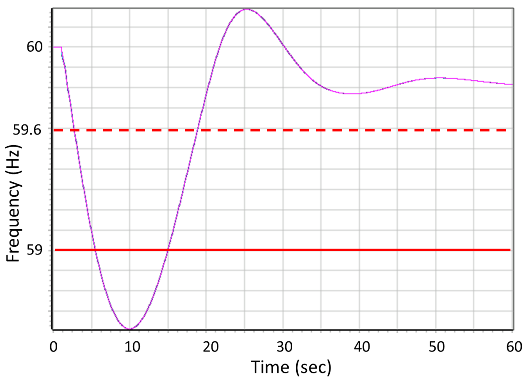

Fig. 2. Frequency plot from power system dynamics simulation when the generator BLT69#1 is opened. WECC Category B limit (59.6Hz) is violated for over 13 seconds while Category C limit (59.0Hz) is violated for over 7 seconds.

seconds or even in the order of minutes. Thus, for example, when multiple commands are reported within a short duration, commands reported between the last update of PG and receipt of cmd_{new} are supposed to be included in $event_{pre}$. Such a usage of $event_{pre}$ will be discussed later in Section IV-B. Both $event_{pre}$ and cmd_{new} contain timestamp relative to the latest system snapshot to be used for simulation (i.e., PG) so that they can be simulated at appropriate timing. If the power grid is under some contingency or stability issue, it should be also reflected in PG and/or $event_{pre}$. Then we compare ($isWorse()$) two simulation results, one with (Res_{cmd}) and the other without (Res_0) the command execution, to decide whether the command should be executed or not. Note that Res_0 can be calculated in advance or in parallel to Res_{cmd}. The decision criteria based on power system dynamics simulation include:

- Whether any (additional) stability violation (i.e., in voltage, frequency, and/or line capacity) is observed
- Whether a system dynamics simulation is aborted in the middle (owing to blackout or islanding)
- Duration till blackout/islanding situation occurs
- Magnitude of frequency/voltage deviation from the nominal value and/or line overload

While detailed investigation of the optimal priority and weighting among these criteria is part of our future work, in this paper we focus on the above ordering.

In sum, if the grid is in a stable state, the command authentication system can focus on whether the control command of interest causes any (additional) violation (e.g., in terms of frequency etc.) and/or blackout or not. This way, we can allow legitimate and harmless commands to be executed while commands that bring the power grid to an unstable condition to be blocked. On the other hand, when the grid is already under unstable condition, we need different handling. If the grid is facing some faults or disasters, instead of just focusing on occurrence of violation, our decision should be based on whether the command of interest will make the situation worse or not, in order NOT to block legitimate recovery commands. As will be demonstrated in Section VI, although we do not claim it optimal, this heuristic scheme works reasonably well.

Algorithm 1 Command Authentication

```
Require: $PG$ ← Latest power grid model and status snapshot
Require: $event_{pre}$ ← Preceeding events to be jointly simulated
Require: $cmd_{new}$ ← Reported control command to be authenticated
  $Res_0$ ← DynSim(PG, $event_{pre}$, $null$)
  $Res_{cmd}$ ← DynSim(PG, $event_{pre}$, $cmd_{new}$)
  if $isWorse(Res_0, Res_{cmd})$ then
    Block execution of $cmd_{new}$
  else
    Allow execution of $cmd_{new}$
  end if
```

B. Consideration on Command Authentication Latency

To perform authentication, we need to run simulation and then extract and evaluate simulation results. The latency measurements using PowerWorld with *Transient Stability* and *SimAuto* add-ons [13] on a PC with Intel Core i7-6700 CPU and 32GB RAM are summarized in Table I. We used GSO 37-bus model to run 30-second system dynamics simulation. (Based on our experiments, 30-second is usually enough to capture initial, largest swing.) As can be seen, the expected latency with this power grid model is around 1.0 second including time to evaluate results, time for communication (typically less than 10ms [8]) and decent safety margin. Time needed for simulation varies depending on size and complexity of the model and duration of simulation. If necessary, the

TABLE I
LATENCY OF 30-SECOND SIMULATION WITH GSO 37-BUS MODEL

Task	Latency [ms]
Opening and initializing case file	38
Transient Stability simulation	458
Extracting results	363
Total	859

latency can be lowered, for example, by shortening the simulation duration or by using simplified equivalent circuit. For instance, concerning the 37-bus model (Table I), if we simulate on the equivalent model with 23 buses and 11 buses, the simulation time goes down to 298ms and 151ms respectively.

Next, let us discuss a case where multiple commands are reported within the mandatory simulation duration (e.g., one summarized in Table I for GSO 37-bus system). As discussed in Section IV-A, those commands must be simulated together to evaluate the joint effect on the power grid stability. On the other hand, decision on each command should be made with the shortest possible latency.

Given that PowerWorld [13] and other popular simulators, such as MatDyn [15], do not allow us to add or modify commands or events in the middle of simulation, our solution is to run parallel simulations. To demonstrate the idea, let us see an example with 3 commands ($C1$, $C2$, and $C3$) reported within a short duration (e.g., within 1 second) below.

1) For the first command ($C1$), we run simulation only by itself. (i.e., compare $DynSim(PG, null, null)$ and $DynSim(PG, null, C1)$), following the logic illustrated in Alg. 1.
2) The decision logic for the second command ($C2$) depends on the decision on $C1$. If $C1$ is not flagged as attack, it is assumed to be executed and therefore we need to run the joint simulation with $C1$ (i.e., comparing $DynSim(PG, C1, null)$ and $DynSim(PG, C1, C2)$). Otherwise, the command authentication system assumes that $C1$ is not executed and simulate the outcome of $C2$ alone, just as done in 1). These 2 ($= 2^1$) pairs of simulations (totally 2^2 simulations) are run in parallel, and based on the decision on $C1$ in 1), which is made before the simulations in this step end, one of the pairs is selected for decision making on $C2$.
3) Likewise, the decision for the third command ($C3$) should take the decisions on $C1$ and $C2$ into consideration. Namely, the system runs 4 ($= 2^2$) pairs of parallel simulations and picks one of them to authenticate $C3$.

Regarding the second and third commands, because simulation for them must be started before knowing the results of simulations for preceding commands, 2^n pairs of, potentially necessary, simulations are executed in parallel to save time, where n represents the number of preceding commands. This way, we can ensure that, for all commands, required latency can be bounded by the latency of 1 execution of the logic in Alg. 1. Because the command authentication scheme is supposed to be run at a resource-rich central server, we think it is a practically viable design decision for minimizing latency.

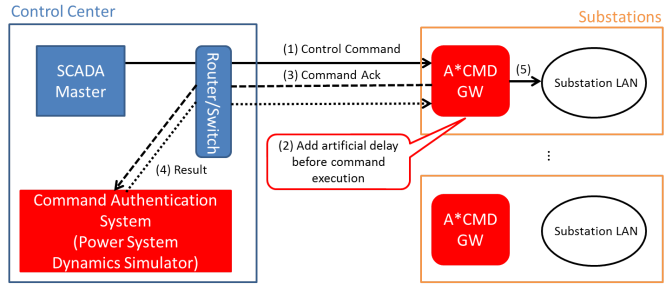

Fig. 3. System overview with A*CMD. When a remote control command is received by A*CMD gateway in each substation (1), artificial time delay is added so that the command is held pending for a certain duration (2). Simultaneously, the receipt of the command is reported back to the command authentication system in the control center, which then evaluates the consequence of the command by means of system dynamics simulation (3). The result is informed to the A*CMD gateway (4), and only if the command is not flagged as malicious, the command is sent to an IED in the substation LAN for execution (5). If the command is flagged as an attack, the pending command is canceled before execution and never is seen by the IED.

We admit that, if the number of simultaneous commands is very large, the number of parallel simulations could become intractable. However, the number of commands that are normally issued within a short duration is typically small [3], so we think it is not a critical issue in the real operation. (In other words, if the number of commands reported within a unit of time by far exceeds a certain threshold, it is immediately considered as an anomaly and operators should respond accordingly.)

Another type of concurrency is handling of multiple commands reported during the duration of system dynamics simulation. For instance, when we use 30-second simulation, a command received within 30 seconds from the receipt of the first command should be evaluated together in the single simulation. Such a situation can also be handled similarly. The only difference is that, if the simulation for the first command is finished before receiving the next command, we do not need to run multiple pairs of parallel simulations. Note that, also in this case, the expected latency for authenticating each command is equal to the latency for 1 execution of Alg. 1.

V. DEPLOYMENT STRATEGY

In this section, we discuss the overall system architecture incorporating the command authentication system using power system dynamics simulation. Configuration on artificial delay, which is the most essential parameter, is also elaborated.

A. Integration into A*CMD Framework

For accommodating the latency for running simulation, in this paper we propose to integrate such a command authentication mechanism into an active command mediation defense system (A*CMD for short hereafter) proposed in [3], [8]. A*CMD makes the most of tolerable time delay in execution of remote control commands to enhance security and resilience against malicious control command injection by proactively blocking it. A*CMD system [3], [8] is typically deployed on a gateway of each substation, which is responsible for protocol translation (e.g., from IEC 60870-5-104 to IEC 61850 MMS) and therefore can reliably mediate all incoming control commands. When receiving any remote control commands, each instance of A*CMD inserts a carefully-determined amount of artificial time delay, which is configured in advance based on a number of contingency simulations as discussed in [3]. Such a delay is utilized to allow a central command authentication system to complete its job before command execution.

The system architecture and procedure of the A*CMD framework using the proposed command authentication scheme is summarized in Fig. 3. The command authentication system can be implemented at the control center, but it should be securely isolated from the SCADA master to counter the cases where the SCADA master is compromised ([1], [2], [16]), or is manipulated by malicious insiders. Our recommendation is to deploy the command authentication system on a physically isolated device, which is connected via the router or switch to the same LAN as the SCADA master through a trunk port. This way, besides receiving command information from distributed A*CMD systems, the authentication system can passively overhear all SCADA communication including interrogation request and response to obtain real-time power grid status. Thereby the system can keep its power grid model for simulation (i.e., PG in Alg. 1) up-to-date. Note that, given the possibility of an attacker on the SCADA master, the command authentication module must collect information independently, instead of relying on the SCADA master. Another advantage of such a deployment is that the command authentication system is not on the critical SCADA communication path, and therefore it does not interfere with normal operation or lower overall system availability or throughput.

B. Configuring Artificial Command-Delaying

Next, we elaborate how the command delaying can be configured or tailored for each power grid model or operational requirement, using the 37-bus system as a concrete case study.

Identifying Required Delay: According to the discussion in Section IV, the required latency (D) for the command authentication using system dynamics simulation is largely constrained by the time to run the simulation and time for processing the results, and in the case of 37-bus system for 30-second simulation, $D \approx 1.0$ sec.

Finding Tolerable Delay: For this step, we can utilize the framework proposed in [3]. In short, given the power grid model of interest, we can define set of contingency scenarios. For example, in practice, power grid operators can consider traditional $N-1$ contingencies, which evaluate cases where one of the components of a certain type (e.g., generator, transmission line, etc.) is disabled. In some cases it may be necessary to take $N-x$ contingencies, which includes failure of multiple components, or $N-1-1$ contingencies, which considers a series of failures, into account. Then, for each contingency, a set of recovery control is defined. Using the given power grid mode and contingency scenario, we can utilize a power-flow simulation (e.g., *Transient Stability* simulation on PowerWorld) to find maximal latency allowed for the recovery control to avoid any violation of power grid stability conditions, including over-/under-frequency, over-/under-voltage, and/or transmission line overload. That maximal delay is considered as tolerable delay for the corresponding contingency (and targets of the recovery control). Based on these simulation results, the global delay tolerance over all the contingencies for the power grid of interest can be obtained by taking the minimum of them (D^*). We can find the simulation study using GSO 37-bus model considering $N-1$ generator-loss contingencies in [3]. According to the result, $D^* \approx 0.9$ sec. Note that this delay tolerance is not sufficient to safely accommodate latency for command authentication using power system dynamics simulation (see Section IV-B).

Command-delaying Configuration: The last step is comparison of the required delay (D) and the global delay tolerance (D^*). If $D^* > D$ holds, the optimal delaying configuration will be applying artificial delay of length D for all power system components so that decisions on all control commands can wait the results of command authentication using power system dynamics simulation. However, if $D^* < D$ (e.g., the attack detection algorithm is complicated or needs to run simulation of longer duration), which is the case with the GSO 37-bus model we have been referring to, we would not be able to save all. Therefore we need to apply another strategy, such as probabilistic delaying to ensure that a certain number of commands are executed with no delay to accommodate sufficient delay for the rest. The minimum number (or fraction) of commands to be executed immediately can be obtained also through preliminary contingency simulations as done in [3]. Then, the probability of inserting artificial delay is computed based on binomial distribution. Alternatively, it is also possible to configure priority of commands in advance so that a certain type of commands are delayed with higher probability.

VI. Evaluation on Detection Accuracy

In this section, we evaluate accuracy of the command authentication using system dynamics simulation. The evaluation should cover two aspects: false positive, which prevents execution of legitimate control commands, and false negative, which allows malicious/anomalous commands to be executed.

When the power grid is in a stable state (e.g., without any observed contingency), our scheme evaluates each reported remote control command to see if it causes any violation during the simulation. In other words, our approach guarantees that commands are executed as long as they don't cause violations. While our design allows legitimate commands, assuming that they don't cause violations, are correctly executed (i.e., no false positive), it may fail to detect malicious commands when they don't cause any violation. For instance, when an attacker tries to control a non-significant circuit breaker to test his attack capability in his probing or reconnaissance phase, it may not be detected. While the latter should be considered as limitation, it is acceptable in practice as long as the missed malicious commands don't cause any stability issue.

In order to evaluate cases where the power grid is facing some stability issue or violation, we consider $N-1$ generator-

TABLE II
CONTINGENCIES AND RECOVERY CONTROLS FOR 37-BUS SYSTEM

j	Lost Generator	Loads to Be Shed
1	JO345 #1	LYNN138, RAY69, BUCKY138, SAVOY69, LAUF69
2	ROGER69	*Nil. No violation is caused.*
3	BLT138 #1	UIUC69, WOLEN69
4	LAUF69	HOMER69, WEBER69, BUCKY138, AMANDA69, LAUF69
5	JO345 #2	LYNN138, RAY69, BUCKY138, SAVOY69, LAUF69
6	BOB69	*Nil. No violation is caused.*
7	BLT69 #1	DEMAR69, BOB69
8	WEBER69 #1	*Nil. No violation is caused.*

loss contingencies using GSO 37-bus system [12]. For each contingency, we define a set of load shedding controls summarized in Table II, which are regarded as legitimate recovery commands. Among them, we focus on the contingencies that, without any recovery control, face violation and/or islanding (i.e., contingencies 1, 3, 4, 5, and 7), and evaluate if the legitimate recovery commands are executed as intended. For instance, regarding the first contingency in the table, we set timings of each recovery control within 1 second from the contingency and evaluated the results using *Transient Stability* analysis on PowerWorld, which are shown in Table III. In the table, the order of rows corresponds to the order of execution, and they are executed in a cumulative manner (i.e., to evaluate load shedding command for RAY69, load shedding commands for LYNN128 and RAY69 are included in $event_{pre}$). As can be seen in the table, for each legitimate command we observed improvements in occurrence of islanding, duration till islanding, and/or magnitude of frequency deviation. Therefore, following our logic, all commands are allowed to be executed. We also observed the same for the other 4 contingencies (3, 4, 5, and 7 in Table II), and no false alarm was observed.

TABLE III
POWER SYSTEM DYNAMICS SIMULATION RESULTS FOR CONTINGENCY 1

Recovery Control	Islanding? (Time [s])	Frequency Deviation [Hz]	Decision
$Nothing$	Yes (6.0)	-	-
LYNN128	Yes (6.3)	-	OK
RAY69	Yes (6.8)	-	OK
SAVOY69	Yes (8.6)	-	OK
LAUF69	No	-0.68	OK
BUCKY128	No	-0.49	OK

Using the same contingency scenarios, we did the experiments to evaluate false negatives. Assuming the load shedding controls listed in Table II are the legitimate controls, we randomly injected other control commands (commands to open irrelevant circuit breakers and transformers), which are considered as malicious/anomalous control commands, to see whether they are flagged or not. This experiment corresponds to a situation where an attacker without detailed knowledge about the power grid model and configurations is blindly injecting malicious commands. We experimented 100 cases with different malicious command injection settings over the contingencies 1, 3, 4, 5, and 7 (20 for each), and found that 83% of attack commands are correctly detected (Fig. 4). We consider that detection rate of this level is sufficient to provide grid operators with situational awareness and advance warning. Note that, even though the rest of the attack commands were not detected, they did not cause power grid instability.

We also evaluated the same setting and attack commands using steady-state simulation. The data that can be derived based on steady-state simulation is different (and also limited), so completely fair comparison is difficult. Thus we here consider the scheme proposed in [17], which is to our knowledge one of the state-of-the-art command authentication schemes using steady-state simulation. Regarding the detection criteria for the steady-state approach, we focused on occurrence of any transmission line overload (90% or higher). As seen in Fig 4, steady-state simulation did not flag many of the injected commands that are potentially malicious. Also, we did not see any command that was flagged by steady-state approach but was not by system-dynamics one. Even though we admit that these commands are not well-crafted attacks, we should seriously consider the possibility of such omission, which can be lowered by incorporating power system dynamics.

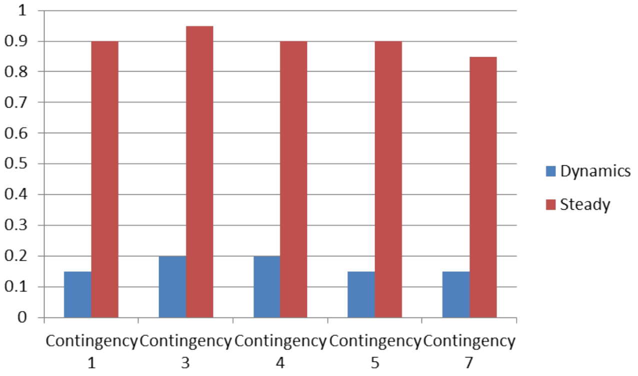

Fig. 4. False Negative Rates on 37-bus System

Instead of blindly injecting commands, a sophisticated attacker who knows the grid topology may craft attack commands to maximize the negative outcome. Such attacks are more likely to cause violation and therefore get caught by our command authentication system. Attackers who are aware of our defense mechanism may try to avoid detection. However, to do so they are limited to issue less harmful commands and therefore their attack capability will be restricted.

We further evaluated the detection scheme using Illini 42 Tornado case on PowerWorld [18]. This case has preconfigured contingencies, including line faults at 10 second, 40 second, and 55 second, and generator trip at 25 second (Table IV). For this experiment, we defined 2 load shedding controls (shown in italic in Table IV), which are considered as legitimate recovery commands, and then evaluated if they are flagged as attacks. Since we did not observe islanding, we focused on frequency deviation and line overload. The results are summarized in Table V. In this experiment, the first three events in Table IV are regarded as contingencies (i.e., as part of PG in Alg. 1), and our decision relies on

TABLE IV
PRE-DEFINED CONTINGENCIES [18] AND RECOVERY CONTROLS

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	*27.00*	*Load Shed*	*Bear345#1*
5	*29.00*	*Load Shed*	*Bear138#1*
6	40.00	Line Fault	Hawk345 - Prairie345
7	40.05	Line Opened	Hawk345 - Prairie345
8	55.00	Line Fault	Tiger345 - Prairie345
9	55.05	Line Opened	Tiger345 - Prairie345

TABLE V
RESULTS OF EXPERIMENTS WITH ILLINI 42 BUS MODEL [18]

No.	Frequency [Hz]	Line Load [%]	Decision
1 - 3	59.8	128	-
4	59.87	115	OK
5	59.87	107	OK

whether the recovery control commands improve the situation or not. As can be seen, the simulation result with the first load shedding improved both frequency deviation and the degree of line overload, and therefore the command is not flagged. In addition, the second load shedding command at time 29 further improved the line overload situation. We also evaluated false negatives by crafting 20 different scenarios. Each scenario contained a different circuit breaker open command, before, between, or after the recovery commands. All of the 20 commands were flagged correctly in terms of occurrence of islanding and the degree of frequency deviation.

Based on these results, command authentication based on power system dynamics is promising. However, the proposed heuristic scheme is discussed as a viable example that can be implemented on top of off-the-shelf power flow simulator, and exploration of advanced schemes is part of our future work.

VII. CONCLUSIONS

In this paper, we proposed use of power system dynamics simulation for command authentication, which can enhance the attack detection capability compared to the traditional schemes solely using steady-state information. To accommodate the latency needed to run the system dynamics simulation, we proposed integration of such a scheme into an active command mediation defense mechanism [3], [8]. Based on the simulation study with a simple, but practical, command authentication algorithm, the detection accuracy is promising.

Major component of our future work is extensive evaluation, including experiments with larger-scale systems and/or real-world data. In particular, based on our preliminary measurements, simulation with 2000-bus power grid model [19] takes in the order of 10 seconds for running simulation. Thus, finding a way to practically reconcile the simulation fidelity (i.e., model size and complexity) and latency for higher scalability is an important future research direction. We hope our work shed light on research for using power system dynamics simulation for securing smart grid systems and also that ours be used as a baseline of such studies.

ACKNOWLEDGMENT

This research is supported in part by the National Research Foundation, Prime Minister’s Office, Singapore under the Energy Programme and administrated by the Energy Market Authority (EP Award No. NRF2014EWT-EIRP002-040) and is also partly supported by the National Research Foundation, Prime Ministers Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

REFERENCES

[1] K. Zetter, “Inside the cunning, unprecedented hack of ukraines 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] D. Mashima, P. Gunathilaka, and B. Chen, “Artificial command delaying for secure substation remote control: Design and implementation,” to appear in IEEE Transactions on Smart Grid.

[4] 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.

[5] 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.

[6] J. M. Weiss, “Control systems cyber securitythe need for appropriate regulations to assure the cyber security of the electric grid,” in *US Congress Testimony, October*, 2007.

[7] 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.

[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] 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.

[10] 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.

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

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

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

[14] L. L. Grigsby, *Electric power generation, transmission, and distribution*. CRC press, 2016.

[15] S. Cole and R. Belmans, “Matdyn, a new matlab-based toolbox for power system dynamic simulation,” *IEEE Transactions on Power systems*, vol. 26, no. 3, pp. 1129–1136, 2011.

[16] Defence Use Case, “Analysis of the cyber attack on the ukrainian power grid,” 2016.

[17] 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)*.

[18] “Illini 42 tornado,” [Online]. Available: http://icseg.iti.illinois.edu/illini-42-tornado/, (Date last accessed on Jun. 7, 2017).

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