SDN-Based Reconfiguration of Distributed and Cooperative Microgrid Control Systems for Mitigating Synchronization Attacks

Aurélic Kpoze, Abdelkader Lahmadi, Isabelle Chrisment, Jules Degila

SDN-Based Reconfiguration of Distributed and Cooperative Microgrid Control Systems for Mitigating Synchronization Attacks

Aurélie Kpoze* and Abdelkader Lahmadi† and Isabelle Chrisment† and Jules Degila*
*Université d'Abomey-Calavi, IMSP, Dangbo, Bénin
†Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France
Email: {satou.kpoze, jules.degila}@imsp-uac.org; abdelkader.lahmadi@loria.fr; isabelle.chrisment@inria.fr

***Abstract*—Industrial Control Systems (ICSs) are widely used in various industries, enabling the control and monitoring of critical infrastructures such as microgrids. In these infrastructures, distributed and cooperative control systems are commonly employed to synchronize set points through information exchange over communication networks. However, these systems are increasingly vulnerable to various security threats, particularly those targeting synchronization data. This paper proposes a network reconfiguration mechanism leveraging Software-Defined Networking (SDN) to mitigate synchronization attacks in distributed and cooperative microgrid control systems. We implement and evaluate our proposed mitigation algorithm using MiniCPS to show its efficacy in avoiding synchronization attacks within such distributed energy systems.**

***Index Terms*—Industrial control systems, microgrid, synchronization attacks, mitigation**

I. Introduction

Industrial Control Systems (ICSs) are combinations of networking, computing, and control components, achieving various industrial objectives such as manufacturing and energy transportation [1]. They encompass SCADA (Supervisory Control and Data Acquisition) systems and DCSs (Distributed Control Systems) and are increasingly prevalent in modern energy infrastructures such as microgrids. A microgrid is a localized electricity network supplying power to a specific geographical area, leveraging Distributed Generators (DGs) such as solar panels or wind turbines for electricity generation. Distributed and cooperative control systems are employed by featuring primary and secondary controllers to maintain synchronization among diverse DGs, as elaborated in [2]. However, ICSs are increasingly vulnerable to security attacks aimed at disrupting their associated cyber-physical processes. This increasing vulnerability with new emerging attack surfaces is due to the integration of Information and Communication Technologies (ICTs). This integration often involves interconnections with untrusted IT systems, expanding the potential threat landscape to include risks originating from the broader Internet. Moreover, using IP protocols for communication within ICSs emphasizes their vulnerability, as these protocols introduce inherent weaknesses into ICSs.

Synchronization attacks are particularly common in microgrids, significantly affecting operational efficiency and overall reliability. One notable attack is False Data Injection (FDI), and in particular, the "Measurement as Reference" attack, demonstrated in [3] that specifically targets microgrids to disrupt their synchronization mechanisms. In this attack, the attacker mainly replaces reference values with measurement values to introduce a high instability in the synchronization process and a loss of control. Mitigating the effects of synchronization attacks on microgrids is essential for maintaining uninterrupted operations, considering that microgrids often power critical infrastructure and essential services.

Software-defined Networking (SDN) offers the flexibility to develop robust reconfiguration mechanisms to counter synchronization attacks by providing centralized control over network resources and enabling dynamic adjustments to network configurations in response to security threats. Using SDN, administrators can programmatically define and enforce security policies, rapidly reconfigure network paths, and implement traffic isolation measures to mitigate the impact of synchronization attacks on the microgrid. In this paper, we propose a network path reconfiguration mechanism leveraging SDN to counter MaR attacks within distributed and cooperative controls in microgrids. This mechanism involves dynamically adjusting the communication paths between distributed components of the microgrid control system by installing flow rules on SDN switches when an attack occurs. By rerouting network traffic through alternative paths, our approach aims to mitigate the impact of MaR attacks and ensure the continued stability and reliability of microgrid operations.

The contributions of our paper are summarized as follows:

1) We propose a novel SDN-based network path reconfiguration mechanism specifically designed to counter synchronization attacks in distributed and cooperative control systems within microgrids.
2) We demonstrate the feasibility of our approach and evaluate its performance by using an emulated environment of the microgrid system and its associated SDN network.

The structure of this paper is outlined as follows. Section II presents some essential information about Microgrids and

Sofware-Defined Networking (SDNs). Section III discusses similar works in this area of research. Section IV gives an overview of the Microgrid control system model, the threat model, and the proposed mitigation technique. The evaluation results are described in Section V. Finally, Section VI concludes the paper and discusses future work.

II. Background

A. Microgrid

Microgrids are self-sufficient, small-scale electricity networks and are considered one of the greatest innovations in modern power grids. They utilize renewable energy resources such as solar and wind power to reduce reliance on fossil and preserve the environment [4]. Their unique capability to operate both independently and in conjunction with the main grid ensures a continuous supply even when the main grid is disrupted. Typically, the main components of microgrids include Distributed Generators (DGs), energy storage devices, control systems, and loads. DGs are low-power generation units placed close to the point of energy consumption to provide localized power generation. Microgrids can be controlled either through a centralized or hierarchical control structure. In a hierarchical control structure, the control mechanism is layered with primary and secondary levels as described in [2].

This type of microgrid relies on a distributed and cooperative control system to maintain stable voltage reference across its DGs units. Each DG unit comprises inverter-based Distributed Energy Resources (DERs) for electricity generation, a load, and a primary controller. The primary controller, a droop controller locally implemented at each DG unit, receives a reference signal from the secondary controller. This is achieved through a synchronization process with neighboring nodes and by obtaining voltage measurements from local meters. The secondary controller relies on a centralized control structure to generate the voltage reference signal required by the primary controller in each DG unit [2]. This reference signal is only sent to a leader DG unit, which then shares this value with the other DGs through the communication network. For each DG unit i, the required reference voltage $V_i^*(t)$ and the measured voltage $V_i^s(t)$ are exchanged through the communication network.

B. Software-defined Networking

Software-defined networking (SDN) is a new network paradigm that separates the control plane from the data plane, allowing for more centralized and programmable network management. In an SDN environment, a central controller manages the flow of network traffic, and network devices are reduced to simple forwarding devices incapable of making packet-routing decisions. By centralizing network intelligence the SDN paradigm offers greater flexibility and agility, with capabilities for dynamic reconfiguration, dynamic policy enforcement, and disaster recovery in demanding environments such as modern power grids [5]. The SDN architecture is three-layered, each serving distinct functions :

- Data Layer: This layer consists of the physical or virtual networking devices, such as switches and routers, that forward data packets between network nodes. These devices are programmable and controlled by the higher layers of the architecture.
- Control Layer: The SDN controller is located at this layer. It is responsible for managing and configuring the infrastructure layer by programming the networking devices with appropriate forwarding rules.
- Application Layer: The application plane contains software that runs on top of the SDN controller and provides network services to the end users. These services can include network monitoring, security, traffic engineering, etc.
- Southbound and Northbound APIs: The Southbound API is the interface between the controller and the network devices, allowing the controller to program the devices with forwarding rules. *OpenFlow* [6] is the most popular open standard protocol for this communication. The Northbound API is the interface between the SDN applications layer and the controller, which allows the controller to execute requests from applications.

III. Related work

Approaches for mitigating attacks in ICSs based on SDN are typically either reactive or proactive. Reactive methods respond to attacks after they occur, aiming to minimize damage and restore operations, while proactive approaches anticipate and prevent attacks through preemptive measures, enhancing overall resilience.

A. Proactive methods

Moving Target Defense (MTD) has emerged as a proactive solution, leveraging SDN as a powerful tool for its development. MTD dynamically transforms network infrastructure to introduce complexity and unpredictability, addressing the vulnerability of inherently static ICSs to attackers [5]. Techniques like path randomization or multipath routing alter flow routes, making it more difficult to predict attack surfaces.

In this vein, Da Silva et al. [7] proposed a multi-path routing strategy to disperse traffic and prevent interception. The mechanism uses Dijkstra's algorithm to calculate multiple routes and install static and dynamic rules on switches. The traffic is then dispersed across the calculated shortest routes, with each route transmitting only a portion of the packets exchanged in a communication. This basic multi-path routing strategy is found ineffective by Ndonda et al. [8], as it can cause delays due to rule expiration, which is unacceptable in ICSs. To address this, they enhanced the strategy by adding priorities to dynamic rules, executing the highest priority rule first. Upon expiration, the next rule is used. Additionally, they propose a path selection algorithm that considers path length and disjointness to limit eavesdropping risks and meet real-time constraints.

In addition, MTD approaches can combine various network attribute randomization techniques, as demonstrated by Chavez

et al. [9]. They periodically configure flows by implementing randomization at three levels: port randomization techniques at the host level, IP randomization at the switch level, and random path selection at the controller level.

B. Reactive methods

Considering the reactive scenario, many research works employ Integer Linear Programming (ILP) to optimize the ICS network reconfiguration in response attacks. Genge et al. [10] propose an SDN-based network optimization solution for ICSs, which dynamically reconfigures the network topology in response to link failures. They calculate a new flow distribution based on optimization results and implement the updated network configuration using the SDN controller, installing static forwarding rules on switches.

However, solving formulated ILP problems can cause delays due to NP-hard nature, often necessitating heuristic algorithms for efficient solutions, as shown in [11] where the authors design an attack detection algorithm and an optimal intervention strategy. The detection algorithm starts with an anomaly detection process, continuously monitoring flow data for anomalies. Subsequently, a localization algorithm is employed to identify affected segments of the network. To isolate critical segments, a new network configuration is computed using a multi-objective optimization problem that selects routing paths, prioritizes flows, and minimizes resource usage. The authors use a heuristic approach to address the problem's NP-hard complexity. Lin et al. [12] propose an approach to isolate compromised Phasor Measurement Units (PMUs) or Phasor Data Concentrators (PDCs) in power systems, and reroute remaining trusted traffic. They formulate an ILP problem with constraints on PMU and PDC connections, aiming to minimize system observability restoration time and maximize observability across power system buses. Additionally, they propose a heuristic algorithm that prioritizes reconnecting PMUs based on bus connectivity and path latency.

Other SDN-based reactive approaches for attack mitigation involve using honeypots to simulate vulnerable services or resources, diverting malicious activity away from critical infrastructure as proposed in [13]. A similar approach is proposed by Salazar et al. [14], where malicious traffic is directed to a fake industrial network mirroring the original.

Although previous work provides valuable solutions for mitigating cyber attacks in ICSs, most approaches focus on reconfiguring data flows rather than control flows, failing to take into account distributed and cooperative control systems. A main constraint on flow reconfiguration in such systems is that achieving synchronization between nodes involves establishing a spanning tree over the communication network, as discussed in [15]. To the best of our knowledge, this paper represents one of the first attempts to implement attack mitigation strategies in distributed cooperative microgrid control systems.

IV. Mitigation Approach

In this section, we describe the microgrid system model and its associated SDN network. We detail the threat model, specifically how an attacker can substitute the reference voltage with the measurement voltage value during exchange between two neighboring Distributed Generators (DGs), aiming to destabilize the system. Finally, we present our mitigation approach, which involves reconfiguring the synchronization paths between the DGs to avoid the compromised link.

A. System model

We have extended the distributed and cooperative microgrid control system by incorporating an SDN controller and OpenFlow switches. OpenVSwitch (OVS) are deployed for building the communication network between the DG units. As shown in Fig. 1, our microgrid system comprises a physical layer containing the DG units, which are interconnected through a series of OpenFlow switches forming the data layer. An SDN controller, located in the control layer, manages the communication flows between these switches. We take an example of a microgrid with four DGs for readability and practicality. The OpenFlow network forwards synchronization data packets between DG units on the physical layer within the microgrid. Each DG unit is connected to a switch, and the switches are interconnected in a ring configuration, enabling traffic between DGs to be managed within the OpenFlow network. We chose a ring topology to link switches due to its scalability and simplicity compared to a full mesh configuration.

The network topology is modeled as an undirected graph $G = (V, E, A_G)$, where V is a finite set of N nodes $V = \{v1, v2, \ldots, vN\}$ and $E \subseteq V \times V$ represents the set of edges. Each DG unit and its connected OVS are modeled as nodes within this graph, while their communication links are represented as edges.

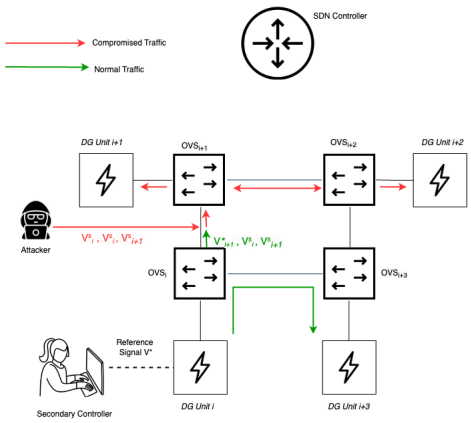

Fig. 1. The SDN-based microgrid system and its threat model.

B. Threat model

We adopt a threat model based on the "Measurement as Reference" (MaR) attack realized using a Man-in-the-Middle (MitM) technique as specified in [3]. The attacker uses the MitM technique to intercept communications between neighboring DGs, as the communications are usually not encrypted

in ICS environments. As shown in Fig. 1, the attacker can impersonate both nodes, gaining access to the reference and measurement values that the two nodes exchange and substituting the reference signal V_{i+1}^* with the voltage measurement V_i^*. The goal of MaR attack is to cause voltage fluctuations and irregular adjustment to harm the loads without violating the admissible range. This type of attack is inherently stealthy because the reference value and the voltage measurement are usually very close, making it difficult for traditional threshold-based detectors to identify the malicious data replacement without generating false alarms. The MaR attack focuses on the communication link between DG units i and $i+1$. In this scenario, the attacker intercepts and manipulates the exchanged data by substituting the reference signal $V_{i+1}^*(t)$ for DG unit $i+1$ with the voltage measurement $V_i^s(t)$ from DG unit i as follows:

$$V_{i+1}^*(t) = V_i^s(t) \tag{1}$$

In our scenario, we assume the MaR attack targets the communication links between OVSs, disrupting only one link simultaneously. This approach focuses on breaking OVSs connections rather than targeting individual nodes. Multiple communication links are not compromised simultaneously, ensuring that an attack-free path can always be found to reach each DG unit. Each OVS has a few links: one connecting it to its respective DG unit; and others connecting it to neighboring OVSs in a ring manner. The attacker intercepts traffic between two OVSs and replaces the reference value with the measurement, causing some DGs to receive the tampered command while others receive the correct command, leading to DG desynchronization.

We formulate the communication network between OVSs in the microgrid as follows:

Given the communication graph $G = (V, E)$:

- V is the set of nodes representing OVSs, denoted as $V = \{OVS_1, OVS_2, \ldots, OVS_N\}$.
- E is the set of edges representing communication links between OVSs, denoted as $E = \{(OVS_i, OVS_j) \mid OVS_i, OVS_j \in V\}$.

For each link (OVS_i, OVS_j), define a binary variable x_{ij} where:

$$x_{ij} = \begin{cases} 1 & \text{if link } (OVS_i, OVS_j) \text{ is compromised,} \\ 0 & \text{if link } (OVS_i, OVS_j) \text{ is operational.} \end{cases}$$

The constraints for the MaR attack targeting only one link at a time can be formulated as:

$$\sum_{(OVS_i, OVS_j) \in E} x_{ij} = 1 \tag{2}$$

where $x_{ij} \in \{0, 1\}$ for all $(OVS_i, OVS_j) \in E$.

This constraint ensures that exactly one link in the microgrid network is compromised at any given time.

C. Reconfiguration mechanism

We mitigate the MaR attack through a reconfiguration process by instructing the OVSs to divert traffic away from the compromised link. We assume that the MaR attack has already been detected in the microgrid based on the findings in [16] and the reconfiguration process is triggered after this detection. We use the SDN controller to perform port forwarding, ensuring traffic redirection based on the provided link information. The achievement of the synchronization consensus among nodes during reconfiguration in the context of the distributed and collaborative microgrid control system required the building of a spanning tree over the communication network [15]. In our system, the OVS connected to the leader DG unit acts as the root node of this spanning tree. We adapt Kruskal's algorithm to compute the reconfiguration path, with a modification to exclude the compromised edge. The Minimum Spanning Tree (MST) is a list that represents a subset of the graph's edges, connecting all nodes with minimal total edge weight. Below, we outline the steps of our reconfiguration process:

Step 1: Initialization: The controller initiates the reconfiguration process by clearing the flow tables on all connected OVS switches and initiates an empty MST.

Step 2: Network Topology Configuration: The algorithm retrieves network links consisting of source and destination node IP addresses, ports, and a random weight. A high weight is assigned to the compromised edge, excluding it from the MST calculation. Random weights are allocated to the remaining edges to simulate real-world conditions, introducing variability. It is assumed that the compromised edge is known beforehand.

Step 3: MST calculation: For the MST calculation, the algorithm iteratively selects edges with the smallest weights from the network links retrieved in the previous step. It ensures that no cycles are formed during this process, which is crucial for creating a tree-like structure. A tree that spans all nodes in the microgrid network is formed, including the root node connected to the leader Distributed Generator (DG) unit. An alternative path for the traffic intended to traverse the compromised link is generated. This path consists of a list of all links that allow nodes to reach each other without relying on the compromised link.

Step 4: Flow Installation: For flow rules installation, the algorithm operates in reactive and proactive modes. In the reactive mode, flows are installed in response to *packet-in* events triggered on the switches. Conversely, in the proactive mode, flows are installed immediately upon computing the minimum spanning tree. Despite the mode used, the number of installed forwarding rules remains constant. Specifically, each OVS h is configured with n flow entries corresponding to its connections within the network. As shown in Fig. 2, the reconfiguration process reroutes traffic to avoid compromised links, thus maintaining synchronization among the DGs.

Algorithm 1 presents the functions corresponding to the aforementioned steps of the reconfiguration process in the microgrid control system.

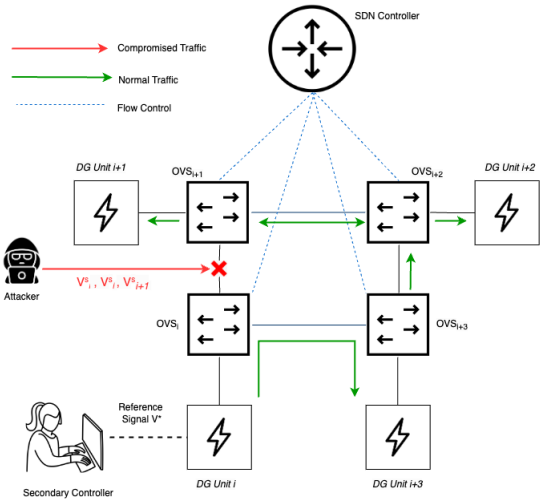

Fig. 2. The SDN-based microgrid control system with the reconfiguration process performed.

V. Evaluation

A. Testbed architecture

We evaluated the performance of our solution using the MiniCPS platform [17], which enhances Cyber-Physical Systems research by extending Mininet with real-time network emulation and tools for simulating industrial components. The testbed was deployed on a Linux Ubuntu 22.04 host with an Intel Core i7 processor and 16 GB of RAM, which served as the host machine for running the MiniCPS and the POX controller as our SDN controller. The MiniCPS network topology consists of DG units each connected to OVSs arranged in a ring topology, with the POX controller managing the network as depicted in Fig. 2.

B. Results

Reconfiguration time: We analyze the time required to calculate an alternate path to isolate the compromised link and the duration of the network interruption corresponding to the network reconfiguration time. Fig. 3 shows the network reconfiguration time, divided into the time to retrieve network links information and the time to perform the MST calculation in both proactive and reactive modes. It shows that the time required to retrieve the network topology using the function *retrieveLinks()* increases considerably as the number of nodes increases, as its time complexity increases with the number of edges in the network. The time required for most other calculations increases only slightly. Furthermore, there is about a 20 ms difference between the times for MST calculation in reactive and proactive modes due to the automatic preemptive installation of network flows in the proactive mode. In the reactive mode, the SDN controller installs flow rules upon a switch-triggered control packet arrival.

Microgrid Synchronization Time: We evaluate the microgrid synchronization time, which is the time between the moment the secondary controller issues the voltage synchronization instruction to the leader DG and when the last DG in the grid receives and acknowledges the instruction. As shown in Fig. 4, the microgrid synchronization time increases by around 40 ms for every ten DGs added to the topology. A variation of around 40 ms is considered acceptable in microgrid operations, as the synchronization remains timely.

Latency: We measured the latency introduced by the reconfiguration using the *ping* command to determine the round-trip time (RTT). As shown in Fig. 5, the latency is

Algorithm 1 Network Flow Reconfiguration Algorithm

function MST(G, Edge, CompromisedEdge)
  Initialize an empty set of edges MST
  Edges ← RetrieveNetworkLinks()
  AssignWeights(Edges)
  Sort all edges of Edges in non-decreasing order of weights
  for each edge (u, v) in Edges **do**
    if (u, v) is not equal to CompromisedEdge **then**
      if u and v are not in the same connected component in MST **then**
        Add (u, v) to MST
        Union(u, v)
      end if
    end if
  end for
  return MST
end function
function InstallFlows(EdgeSet MST)
  for each edge (u, v) in MST **do**
    InstallFlow(u, v)
  end for
end function
function Main
  G ← RetrieveNetworkLinks()
  AssignWeights(G)
  CompromisedEdge ← IdentifyCompromisedEdge(G)
  MST ← MST(G, CompromisedEdge)
  InstallFlows(MST)
end function

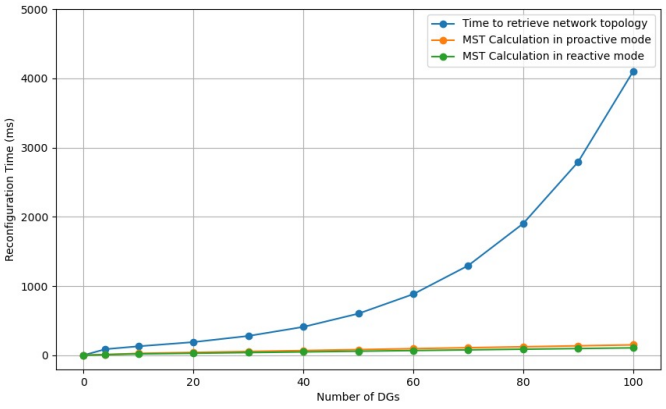

Fig. 3. Reconfiguration time for reactive and proactive mitigation scenarios.

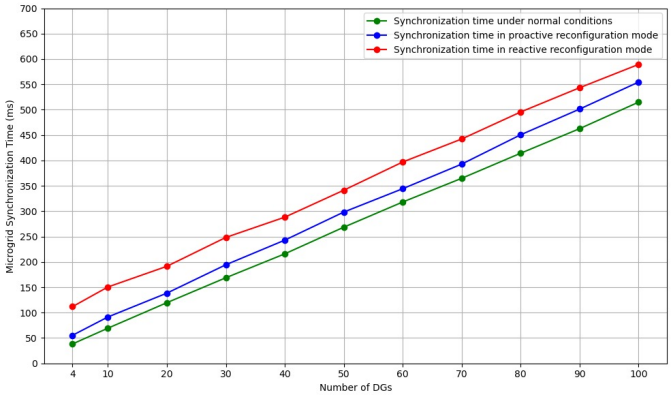

Fig. 4. Microgrid synchronization time for reactive and proactive mitigation scenarios

around 100 ms in a microgrid with a dozen of DGs, which is acceptable for power control operations [18]. The microgrid latency with 20 DGs is 120 ms, increasing by about 50 ms for each additional set of ten DGs. This latency remains acceptable despite the network's increasing complexity under distributed and cooperative control.

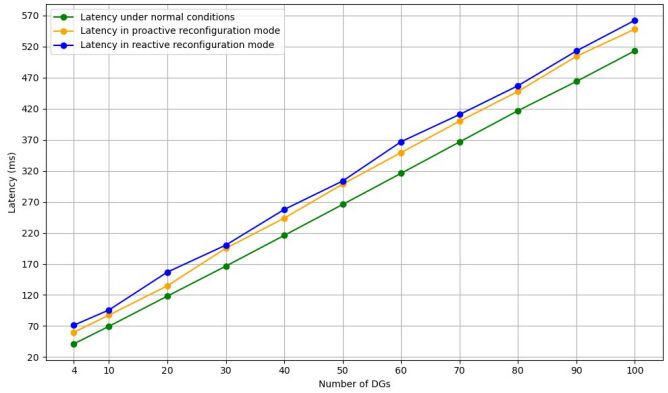

Fig. 5. Latency for reactive and proactive mitigation scenarios.

VI. Conclusion and Future Work

This paper proposes an SDN-based network reconfiguration mechanism tailored for distributed and cooperative microgrid control systems to mitigate synchronization attacks, in particular, the stealthy MaR attack that disrupts microgrid synchronization. Through experimentation within an emulated testbed using MiniCPS, our proposal has been rigorously evaluated across various scenarios. The results indicate the proposed approach is able to maintain the timing of the synchronization process between the DGs while avoiding the compromised link.

Future work will focus on exploring several directions for enhancing the proposed mechanism. Specifically, our research will focus on integrating programmable data planes. By leveraging P4 programmability, we aim to improve the adaptability and performance of our network reconfiguration mechanism by integrating the rerouting decision in the network switch data plane to enhance the response time of the reconfiguration process.

References

[1] D. Bhamare, M. Zolanvari, A. Erbad, R. Jain, K. Khan, and N. Meskin, "Cybersecurity for industrial control systems: A survey," *Computers Security*, vol. 89, p. 101677, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404819302172

[2] M. Ma and A. Lahmadi, "On the impact of synchronization attacks on distributed and cooperative control in microgrid systems," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2018, pp. 1–6.

[3] M. Ma, A. Lahmadi, and I. Chrisment, "Demonstration of synchronization attacks on distributed and cooperative control in microgrids," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 727–728.

[4] M. H. Saeed, W. Fangzong, B. A. Kalwar, and S. Iqbal, "A review on microgrids' challenges perspectives," *IEEE Access*, vol. 9, pp. 166 502–166 517, 2021.

[5] M. Abdelkhalek, B. Hyder, M. Govindarasu, and C. G. Rieger, "Moving target defense routing for sdn-enabled smart grid," in *2022 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2022, pp. 215–220.

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM computer communication review*, vol. 38, no. 2, pp. 69–74, 2008.

[7] E. Germano da Silva, L. A. Dias Knob, J. A. Wickboldt, L. P. Gaspary, L. Z. Granville, and A. Schaeffer-Filho, "Capitalizing on sdn-based scada systems: An anti-eavesdropping case-study," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 165–173.

[8] G. K. Ndonda and R. Sadre, "A low-delay sdn-based countermeasure to eavesdropping attacks in industrial control systems," in *2017 IEEE conference on network function virtualization and software defined networks (NFV-SDN)*. IEEE, 2017, pp. 1–7.

[9] A. R. Chavez, W. M. Stout, and S. Peisert, "Techniques for the dynamic randomization of network attributes," in *2015 international carnahan conference on security technology (ICCST)*. IEEE, 2015, pp. 1–6.

[10] B. Genge and P. Haller, "A hierarchical control plane for software-defined networks-based industrial control systems," in *2016 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE, 2016, pp. 73–81.

[11] H. Sándor, B. Genge, Z. Szántó, L. Márton, and P. Haller, "Cyber attack detection and mitigation: Software defined survivable industrial control systems," *International Journal of Critical Infrastructure Protection*, vol. 25, pp. 152–168, 2019.

[12] H. Lin, C. Chen, J. Wang, J. Qi, D. Jin, Z. T. Kalbarczyk, and R. K. Iyer, "Self-healing attack-resilient pmu network for power system operation," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1551–1565, 2018.

[13] F. Wang, W. Qi, and T. Qian, "A dynamic cybersecurity protection method based on software-defined networking for industrial control systems," in *2019 Chinese Automation Congress (CAC)*. IEEE, 2019, pp. 1831–1834.

[14] L. E. Salazar and A. A. Cardenas, "Enhancing the resiliency of cyber-physical systems with software-defined networks," in *Proceedings of the acm workshop on cyber-physical systems security & privacy*, 2019, pp. 15–26.

[15] D. He, D. Shi, and R. Sharma, "Consensus-based distributed cooperative control for microgrid voltage regulation and reactive power sharing," in *IEEE PES Innovative Smart Grid Technologies, Europe*. IEEE, 2014, pp. 1–6.

[16] M. Ma, A. Lahmadi, and I. Chrisment, "Detecting a stealthy attack in distributed control for microgrids using machine learning algorithms," in *2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*, vol. 1, 2020, pp. 143–148.

[17] D. Antonioli and N. O. Tippenhauer, "Minicps: A toolkit for security research on cps networks," 07 2015.

[18] M. Yadav, N. Pal, and D. K. Saini, "Microgrid control, storage, and communication strategies to enhance resiliency for survival of critical load," *IEEE Access*, vol. 8, pp. 169 047–169 069, 2020.