2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC) | 978-1-6654-2463-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/COMPSAC51774.2021.00104

DDAF: Deceptive Data Acquisition Framework against Stealthy Attacks in Cyber-Physical Systems

Md Hasan Shahriar*, Mohammad Ashiqur Rahman*, Nur Imtiazul Haque*, and Badrul Chowdhury†
*Analytics for Cyber Defense (ACyD) Lab, Florida International University, USA
†Department of Electrical and Computer Engineering, The University of North Carolina at Charlotte, USA
*{mshah068, marahman, nhaqu004}@fiu.edu, †b.chowdhury@uncc.edu

***Abstract*—Cyber-physical systems (CPSs) and the Internet of Things (IoT) are converging towards a hybrid platform that is becoming ubiquitous in all modern infrastructures. The massive deployment of CPS requires comprehensive, secure, and reliable communication. The integration of complex and heterogeneous systems makes enormous space for the adversaries to get into the network and inject malicious data. To obfuscate and mislead the attackers, we propose DDAF, a deception defense-based data acquisition framework for a hierarchical communication network of CPSs. Each switch in the hierarchical network generates a random pattern of addresses/IDs by shuffling the original sensor IDs reported through it. Later, while sending the measurement data from remotely located sensors to the central controller, the switches craft the network packets by replacing the sensors' original IDs with pre-generated deceptive IDs. Due to the deception, any stealthy attack turns into a random data injection and ends up as an outlier during the bad data detection process. By analyzing the outlier data, DDAF detects and localizes the attack points and the targeted sensors. DDAF is generic and highly scalable to be implemented in any size of networked control systems. Experimental results on the standard IEEE 14, 57, and 300 bus power systems show that DDAF can detect, mitigate, and localize almost 100% of the stealthy cyber-attacks. To the best of our knowledge, this is the first framework that implements complete randomization in the data acquisition of a networked control system.**

***Index Terms*—Deception defense, cyber-physical systems, stealthy attacks**

I. Introduction

Cyber-physical systems (CPSs) are omnipresent in critical infrastructure, which consolidate sensing, communication, processing, and control of both cyber and physical domains [1]. The Internet of things (IoT) has opened up a new dimension to the CPSs through phenomenal unification enabling real-time monitoring and data exchange. As CPSs are getting larger with heterogeneous sensor interaction due to widespread acceptance, they are also creating a massive attack space for the adversaries [2]–[4]. The contemporary cyber-attacks are sophisticated enough to overcome legacy defense tactics like intrusion or anomaly detection systems. Utilizing the targeted system's knowledge and states, adversaries can launch stealthy false data injection (FDI) attacks, which are specially designed for exploiting CPSs control loops [5]–[8].

Cyber-attacks in CPSs are performed in several ways. By eavesdropping on the communication channels, a powerful intruder gain access to the sensor data. Depending on the attack goal, he/she injects some malicious stealthy data into the sensor reading, misleading the state estimation (SE) of CPSs to his/her aspired direction. Cyber-attacks not only hamper the services' reliability but also introduce substantial financial losses. Stuxnet, a 500-kilobyte computer worm attacked over fifteen Iranian facilities in 2010, destroying over a thousand uranium enriching centrifuges [9]. A recent report states that the US government could lose $1 trillion if a Stuxnet-like attack would be carried out in the US smart grid [10]. Again, BlackEnergy Malware, a denial-of-service (DoS) attack on supervisory control and data acquisition (SCADA) system, is responsible for a massive power outage in Ukraine [11]. More recently, some utility companies in Utah, Wyoming, and California power grids also suffered from cyber-attacks that caused power outage in those areas [12]. The attacks can be classified into two categories- active attacks, and passive attacks [13]. Passive attacks are the process of reconnaissance of the system states. The goal of such attack is to study the parameters of the physical states and determine the optimal attack tactics without creating any attention of the defender. The active attacks are the injections of the malicious data into the sensor measurements to achieve the attacker's goal. Active attack exploits the integrity as well as the availability of sensor data.

The effectiveness of the active attacks depends on the success of the passive attacks. In most of the CPSs, the devices in the SCADA networks have limited computational power to perform high-end encryption on the sensor data streams [14]. Besides, some critical infrastructures (e.g., power systems) have time constraint on the data acquisition process. For example, IEC 61850 standard enforces a maximum end-to-end time delay of 4 ms for generic object-oriented substation event (GOOSE) messages within a substation [15]. Thus, in some cases, despite having sufficient computational capability, high-end security may not be applied due to the time-restriction. Several study showed, once a sophisticated attacker gains sufficient knowledge about the targeted system's topology and states, he/she can easily evade the existing defense mechanisms by utilizing the resources in generation of the attack data [16]–[18].

"All warfare is based on deception. Hence, when we are able to attack, we must seem unable; when using our forces, we must appear inactive; when we are near, we must make the enemy believe we are far away; when far away, we must make him believe we are near.— Sun tzu, The Art of War [19]".

Like the author said, in cyber war, deception also plays a vital role in misleading the attackers, failing them in accomplishment of their intent. For securing the CPSs from both of the active and passive attacks, we develop a deceptive defense-based data acquisition framework (DDAF) for any hierarchical SCADA network. In the framework, we secure the data acquisition steps by shuffling the sensor IP addresses/IDs at each node in the hierarchy.

Our contribution to the paper can be summarized in the following points:

- We model a deceptive data acquisition framework to mislead an attacker while reporting the sensor data through a hierarchical communication network.
- We develop a tree-based randomization algorithm to maximize the randomness in deceptive ID generation, that is utilized to obfuscate the passive attacks and mitigate the stealthy active attacks. Due to the deception, any attempt of stealthy FDI attack turns into a random data injection, which is easily removed by bad data detection techniques, keeping the attack impact within 2%.
- While most of the related works focus only on attack mitigation technique, we develop algorithms also to detect and localize stealthy FDI attacks. As deception makes compromised sensors outliers. Analyzing their position and the residuals reveals the injected attack vector, targeted sensors, and also the position of the attacker into the network.
- We analyze the performance and scalability of the framework on different sizes of standard IEEE bus systems that indicates the frameworks effectiveness. The codes and data are publicly available at [20].

Organization: The rest of the paper is organized as follows: We add sufficient background information in Section II. We introduce our proposed DDAF in Section III. Section IV discusses the technical details of the framework. In Section V, we explain an example case study. The evaluation setup and result analysis are formulated in Section VI. The related works are discussed in Section VII. At last, we conclude the paper in Section VIII.

II. Preliminaries

In this section we discuss the terminologies that are used throughout the paper to facilitate readers' comprehension.

A. State Estimation and Bad Data Detector

State estimation (SE) is defined as the process of knowing the states of the network from the redundant telemetry measurements. Let us assume that a CPS has n number of states variables, $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$, and m sensor measurements, $\mathbf{z} = (z_1, z_2, \cdots, z_m)^T$, with sufficient redundancy ($m >> n$). Thus, the relationship between the states vector $\mathbf{x}$ and the measurement set $\mathbf{z}$ can be represented as $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e}$, where $\mathbf{h}(\mathbf{x})$ is the measurement function mapping measurement set to the state vector, and $\mathbf{e}$ is the set of normally distribution measurement errors [21]. For example, in a power system, the state variables are the bus voltages (magnitude and phase angle) and the sensor measures are the line power flows, bus power injections, phasor measurement units (PMU) data, etc.

Due to the simplicity, robustness, and high speed of computation, DC power flow is vastly used for the real-time analysis of power systems in the industry. In the DC approximation of the power system, voltage magnitude is considered unity. Thus, only the phase angles of the bus voltages are regarded as the state variables. Besides, $\mathbf{h}(\mathbf{x})$ is the linear transformation using the Jacobian matrix, $\mathbf{H}$, calculated using line admittance.

As the measurement functions $\mathbf{h}(\mathbf{x})$ are considered linear and the most probable system states vector $\hat{\mathbf{x}}$ can be estimated directly by solving: $\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\mathbf{z}$. Here, $\mathbf{W}$ is a diagonal weighting matrix. Thus, the measurement vector $\mathbf{z_{est}}$ can be estimated by $\mathbf{z_{est}} = \mathbf{h}(\hat{\mathbf{x}})$ and the measurement residuals set $\mathbf{r} = ||\mathbf{z} - \mathbf{z_{est}}||$. Bad data detection (BDD) technique is mostly used along with the SE procedure, that considers a threshold on the residual data to find the outliers. Considering a maximum threshold as τ, any measurement z_i is regarded as a bad data and removed from the SE procedure if $|r_i| > \tau$. The estimation process is repeated until there is no bad data into the considered measurement vector. At the end, an $\mathbf{SE} - \mathbf{BDD}$ procedure returns the estimated states $\hat{\mathbf{x}}$, estimated measurements $\mathbf{z_{est}}$, residual vector $\mathbf{r}$, and the list of outlier sensors.

B. Stealthy False Data Injection Attack

Attack vector is defined as a set of malicious data to be injected into a set of sensor measurements. Whereas random attack vectors create anomaly, intelligently calculated ones generated considering the targeted system's topology can bypass the BDD [22]. Let us assume that, the attacker's goal is to change state variables set $\hat{\mathbf{x}}$ by the malicious amount $\mathbf{c}$. Thus, to remain stealthy, he/she needs to inject false data $\mathbf{a}$ to the measurement set $\mathbf{z}$, where $\mathbf{a} = \mathbf{h}(\mathbf{c})$. Since $\mathbf{z} + \mathbf{a} = \mathbf{h}(\hat{\mathbf{x}} + \mathbf{c})$, the residual $\mathrm{r} = ||(\mathbf{z}+\mathbf{a}) - \mathbf{h}(\hat{\mathbf{x}}+\mathbf{c})|| = ||\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}) + \mathbf{a} - \mathbf{h}(\mathbf{c})|| = ||\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})||$. Thus, the data injection disappears from the residual vector, and the attack remains stealthy. This attack requires full knowledge of the targeted system's topology, parameters, and measurement configuration. For the rest of the paper, we use FDI to refer to the stealthy FDI attacks.

C. System Model

Future CPS networks are supposed to maintain a hierarchical communication structure as it lessens the communication overhead and assures the system's stability, reliability, and efficiency [23]. As smart grid is a perfect example of modern CPSs, we consider a smart grid hierarchical communication network as our testbed. Fig. 1 shows our considered hierarchical communication network in a smart grid. The sensors in the substations can be remote terminal units (RTUs), intelligent electronic devices (IEDs), phasor measurement units (PMUs), etc. The sensors are located within the substations, and directly report the measurement data to their substation switches. A level 1 (L-1) substation only receives measurement data from its own sensors, whereas a level 2 (L-2) substation receives data both from its sensors and the underneath L-1 substations,

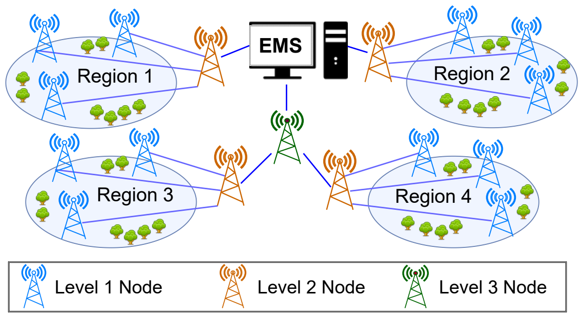


Fig. 1: Hierarchical networks of smart grid.

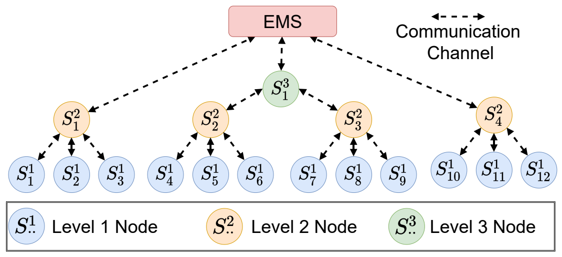

EMS
Communication Channel
S_1^3
S_1^2
S_2^2
S_3^2
S_4^2
S_1^1 S_2^1 S_3^1 S_4^1 S_5^1 S_6^1 S_7^1 S_8^1 S_9^1 S_{10}^1 S_{11}^1 S_{12}^1
Level 1 Node
Level 2 Node
Level 3 Node

Fig. 2: Tree diagram of the hierarchical network (Fig. 1).

and so on. After collecting remote sensor data from the nearest substations, the EMS runs the $\mathbf{SE-BDD}$ algorithm to estimate the states take appropriate control decisions. As Fig. 2 shows, such hierarchy forms a tree structure, where $\mathcal{S}_i^l$ indicates the i-th substation at level l and EMS is the root. We model all the substation switches in the network as node objects. Based on the connectivity, each node receives the data packets from its child nodes and forwards to its parent node. Moreover, we define the communication channels between L-1 and L-2 switches as L-1 channels and the ones between L-2 and L-3 switches as L-2 channel, and so on. For the rest of the paper, layer and level are used interchangeably.

D. Attack Model

In this subsection, we define the attack model according to the attacker's capabilities and goals. During the data acquisition process, there remains multiple vulnerable points which an adversary can compromise. Firstly, the attacker can compromise the individual targeted sensors but they are located within the substations, which are highly secure to break in. However, as the network communication channels span the overall SCADA network, physical security is fragile at some points and we consider them as the attack surface to inject the false data. Thus, when the attacker compromises the L-1 communication channels, we consider them as L-1 attacks, in case of attacking L-2 channels, we define them as L-2 attacks, and so on. We consider the man in the middle FDI attacks as the threats, where the attacker injects malicious data into the network packets to deviate the state estimation.

III. DDAF

DDAF applies to any hierarchical networked control system. The primary features of DDAF are shown in Fig. 3. As

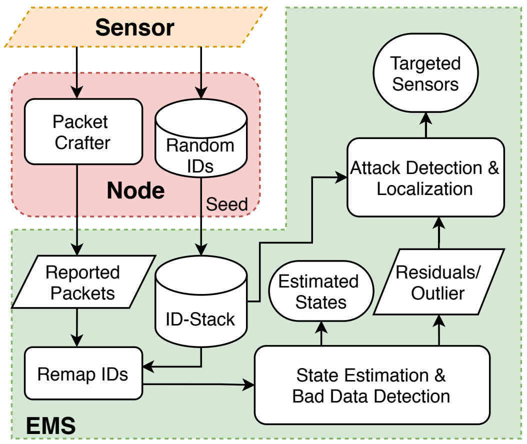


Fig. 3: Basic flow chart of DDAF.

the nodes need to analyze and modify the network packets, the "deceiving" nodes are equipped with software-defined networking (SDN) controllers [24]. An SDN controller can read, edit, and assemble packets in run-time. In modern communication networks, SDN is widely implemented. In case of the unavailability of SDN switches, similar capabilities can also be implemented on the conventional switches by VMware and Nicira [25].

The deception mechanism is implemented at the SDN controller of the nodes. EMS assigns each sensors in any of the two categories: *randomized* and *fixed*, and share the information with the nodes. Nodes keep track of the assigned sensor types and utilize them during the deception process. As a part of the deception, each node does two routine tasks: i) $recID$-$randID$ pair generation, and ii) packet crafting. A received ID - random ID or $recID$-$randID$ pair contains two sets of IDs. The received IDs are the set of sensor IDs whose data packets are reported to the node by its child nodes. At a specific interval, the received IDs are shuffled using a randomization algorithm to generate a set random IDs. Such a pair of IDs are used to craft the packets during the data acquisition process.

On the other hand, another routine task that a node executes is forwarding the packets. Whereas the conventional nodes forward the packets to the next devices, in DDAF, the SDN controller crafts the packets before forwarding. The packet crafting is done by replacing the received IDs of the *randomized* sensors with the random IDs using the node's $recID$-$randID$ pair. The process continues through all the levels until the packets reach EMS. Whereas packet crafting is almost a continuous process, $recID$-$randID$ generation is carried out whenever the deception patterns need to be updated. To maintain the defense's dynamicity, EMS asks the nodes to update their $recID$-$randID$ pairs at a regular interval by sending the updated list of *fixed* and *randomized* sensors.

The randomization algorithm uses a seed value, which is generated using the hardware/software token-based RSA SecurID authentication mechanism. This technique generates a secure authentication code (used as seed) for each node

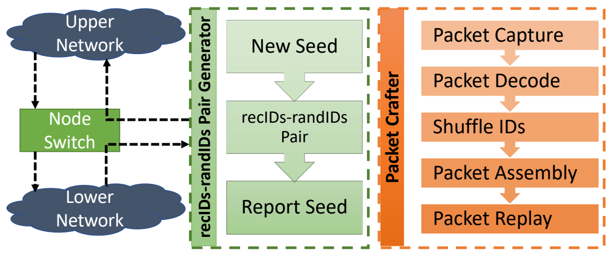


Fig. 4: Deception mechanism executed at node switches.

using the built-in clock, and the node's factory encoded secret key [26]. Thus, each node generates different seed at different time and uses it while running the randomization algorithm. As all the nodes are interconnected through physical lines, they can maintain the clock synchronization with EMS. Thus, whenever the nodes generate the seeds, EMS follows the same steps to generate the same seeds using the nodes' secret keys, which are already shared with the EMS during the installation stage. Using the seed values, EMS runs the same randomization algorithm for all the nodes to generate the $recIDs$-$randIDs$ pairs and build a stack, called ID-Stack. Using the ID-Stack containing all the randomization information, EMS recovers the original IDs from the collected crafted packets by running a remapping algorithm. Finally, EMS executes the necessary routine tasks (i.e., **SE-BDD**) on the remapped original data and takes control decisions for optimal operation of the system. As the sensor data packets are sent with random IDs, if an attacker injects stealthy false data into some targeted sensors, the injections happen to the wrong measurements. Later, **SE-BDD** removes them as bad data and makes the system robust from stealthy FDI attacks.

DDAF further analyzes the residual data and the list of the outlier sensors to detect and localize the existence of stealthy FDI attacks. If there is an attack, the framework provides the attacker's location in the communication network and his/her targeted sensors. In detecting and localizing the attack, the residual vector and the outliers are reshuffled again using the ID-Stack's $recIDs$-$randIDs$ pair of each level to observe them from an attacker's perspective. As the FDI attack vectors follow the system topology ($a = h(c)$), if there is an attack at any network level, the shuffling finds a topological pattern into the residual data and the outlier senors' locations. The following section provides a detailed overview of different parts of the framework.

IV. Technical Details

In this section, we discuss the details of each part of DDAF. We model the nodes with some common attributes, as explained in Table I. We divide the overall process into four steps: i) deception mechanism ii) remapping mechanism, iii) attack mitigation, and iv) attack detection and localization.

A. Deception Mechanism

Firstly, EMS sends the deception instructions to the sensors, and the nodes keep track of the instructions. The instruction contains a type flag where 0 and 1 to indicate the *fixed* and

TABLE I: Attributes of a node in the hierarchical network

Attributes	Type	Description	Example
nodeID	object	ID of the node, $\mathcal{S}_i^l$	$\mathcal{S}_1^2$
childs	list	Address of child nodes	[$\mathcal{S}_1^1$, $\mathcal{S}_2^1$, ..]
sensors	set	Original IDs of the sensors	{1, 2, ..., m}
randomized	set	Randomized IDs of the sensors	{3, 4, ..., m-1}

Algorithm 1: randIDGen($nodeX$)

```
initialization $recIDs, randIDs = []$ ;
$recIDs \leftarrow nodeX.sensors - nodeX.randomized$;
$randIDs \leftarrow nodeX.sensors - nodeX.randomized$;
$recIDs$.append($nodeX.randomized$);
for for each child $nodeC \in nodeX.childs$ do
    $randIDs$.append(Rand-Child($nodeX, nodeC.randomized$));
end
Return $recIDs, randIDs$
```

randomized sensors, respectively. Thus, each node constrains an m dimensional array $\mathcal{T}$ to store the sensor-wise deception instructions.As shown in Fig. 4, the deceptive mechanism is divided into two steps:

1) $recIDs$-$randIDs$ Pair Generation: $recIDs$-$randIDs$ pair generation is the first step of the deception process. The received IDs of *randomized* sensors $recIDs$ are shuffled among themselves to generate the deceptive random IDs $randIDs$. The set $randIDs$ is generated following procedure **randIDGen**, as shown in Algorithm 1 using $recIDs$, and the seed. We propose a tree-based approach for the randomization. Each node generates a seed value and utilizes it to generate the random IDs. The $recIDs$-$randIDs$ pairs of the nodes are updated at a regular interval with EMS's request.

The sub-task of random ID generation for a particular node is done by a recursive algorithm, **Rand-Child**, which ensures the uniform distribution of the deceptive sensors. The same process is performed for all the nodes in the network other than EMS to generate and store their $recIDs$-$randIDs$ pair.

The primary goal of random ID generation is to shuffle the sensors of one child node with the sensors coming from all the child nodes. Hence, if an adversary attacks the sensors coming from that child node or a critical part of the system, the injected malicious data will be distributed to the sensors coming from different child nodes, which makes the stealthy attacks invalid and removing them as bad data still keeps the system observable.

2) Packet Crafting: Packet crafting is the core part of the deception process that utilizes the $recIDs$-$randIDs$ pair. Once the data packets are received, SDN controller at the node decodes the packets and replaces the *randomized* sensors' received IDs $\mathcal{I}_{rec}$ with deceptive random IDs, $\mathcal{I}_{rand}$, following Algorithm 2. The probability that $\mathcal{I}_{rec}$ and $\mathcal{I}_{rand}$ are the same is $\frac{1}{n_{sen}!}$, where n_{sen} is the number of sensors in that node and the deception is a random process. For $n_{sen} = 5$ the probability is 0.008 and for 10 it is 2.75×10^{-7}. Thus, for

Algorithm 2: RanID($\mathcal{I}_{rec}, recIDs, randIDs, \mathcal{T}$)

```
1 initialize $\mathcal{I}_{rand}$ = $\mathcal{I}_{rec}$;
2 for  $i = 1$ to $len(\mathcal{I}_{rec})$ do
3     for  $j = 1$ to $len(recIDs)$ do
4         if $\mathcal{I}_{rec}[i] = recIDs[j]$ and $\mathcal{T}[i] == 1$ then
5             $\mathcal{I}_{rand}[i] = randIDs[j]$;
6             $break$;
7     end
8 end
```

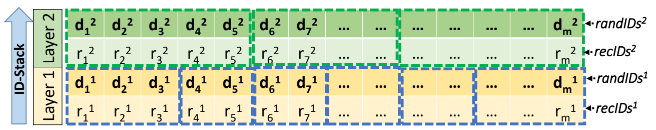


Fig. 5: An example ID-Stack for two level network.

a higher randomization level with more sensors, the probability converges towards zero. Furthermore, our randomization algorithm is based on a tree, which ensures the randomness.

Thus, each level of shuffling makes the IDs more randomized, and therefore, unpredictable for the attacker to remap to the actual ones. Once shuffling is done, the packets are updated with randomized IDs and assembled with a proper header. Finally, they are forwarded into the network, directed towards the next node in the hierarchy for further randomization until they reach EMS.

B. Remapping Mechanism

As the reported sensor data contain randomized IDs, EMS needs to remap the shuffled measurement data to the original sequence before utilizing them control decisions. Remapping is exactly the opposite of the deception process. Like the deception process, it has two parts. Firstly, it generates the ID-Stack and then remaps the crafted data.

1) Building ID-Stack: During the $recIDs$-$randIDs$ pair generation process as discussed in Section IV-A1, EMS also generates the seeds using the nodes' secret keys. Using the seed values, EMS regenerates the same $recIDs$-$randIDs$ pairs that each node has generated and builds an ID-Stack with all the randomization patterns for different levels and nodes. EMS can use the ID-Stack to trace back to the original sensor IDs from the reported deceptive IDs. During the ID-Stack formation, the $recIDs$-$randIDs$ pairs of same level nodes are concatenated vertically, which gives the total view of the deception at that level. Each level adds additional randomization into the system. $recIDs$-$randIDs$ pairs of different levels (i.e., levels 1, 2, 3, etc.) are stacked sequentially so that the height of the ID-Stack is same as the height of the network hierarchy. Fig. 5 shows a sample ID-Stack for a two-layer hierarchical network. Here, the yellow and green boxes represent the **randIDGen** pairs for different nodes at level 1 and 2, respectively.

2) Remapping IDs: The second and core part of the remapping process is ID remapping that recovers original IDs from the deceived IDs. First, the crafted packets are collected and decoded at the EMS switch. The data packets contain the shuffled sensor IDs after multiple randomization. As the exact $recIDs$-$randIDs$ are generated in ID-Stack, multiple remaps of the reported IDs in reverse order of the levels provide the original IDs associated with the measurement values. We use the same **randID** algorithm (Algorithm. 2), but now the shuffling direction is changed. Thus, $randIDs$ are swapped with the $recIDs$. In this case, EMS calls the algorithm all the layers in downwards direction, and the last call for level 1 returns the original ID sequence of the measurement data.

C. Attack Mitigation

This subsection explains how $\mathbf{SE-BDD}$ algorithm mitigate FDI attacks from a system executing DDAF. EMS runs the $\mathbf{SE-BDD}$ procedure on the remapped data, where the topology matrix H and measurement vector z_{msr} are generated using the remapped sensors IDs and measurement values. Now, let us assume, there is an attacker trying to inject stealthy data $\mathbf{a_{org}}$ into the set of sensors $\mathcal{I}^c_{org}$ of that node, where $\mathbf{a_{org}} = [a_{o_1}, a_{o_2}, ..., a_{o_{m-1}}, a_{o_m}]$. The attack will be successful (stealthy) if $\mathbf{a_{org}} = \mathbf{Hc}$, where $\mathbf{c}$ is the targeted malicious state. Due to the randomization, the injection ends up as a randomized attack data, $\mathbf{a_{rand}}$ into the sensors $\mathcal{I}^c_{rand}$ where, $\mathbf{a_{rand}} = [a_{r_1}, a_{r_2}, ..., a_{r_{m-1}}, a_{r_m}]$. We already show that with sufficient randomization, $\mathcal{I}^c_{org} \neq \mathcal{I}^c_{rand}$, thus, $\mathbf{a_{org}} \neq \mathbf{a_{rand}}$ and $\mathbf{a_{rand}} \neq \mathbf{Hc}$. Therefore, the attack loses it stealthiness and with proper randomization, the BDD process at EMS will find all the compromised sensors as outliers. The process is explained in the next subsection.

Let us assume, and the original measurement vector is $\mathbf{z_{org}}$, which is randomized as $\mathbf{z_{rand}}$ at the compromised node. Thus, the attacker injects $\mathbf{a_{org}}$ to the randomized measurement vector $\mathbf{z_{rand}}$. The compromised measurement vector that EMS receives is $\mathbf{z^a_{rand}}$, where $\mathbf{z^a_{rand}}$ = $\mathbf{z_{rand}}$ + $\mathbf{a_{org}}$. However, EMS remaps the deceptive IDs $\mathcal{I}_{rand}$ to original IDs $\mathcal{I}_{org}$ (also $\mathcal{I}^c_{rand}$ to $\mathcal{I}^c_{org}$) using the ID-Stack. Thus, the compromised measurement data $\mathbf{z^a_{rand}}$ also changes to the original sequence as $\mathbf{z^a_{org}}$. As remapping and shuffling are two exact opposite tasks, during this remapping process, the original injected attack vector gets shuffled from $\mathbf{a_{org}}$ to $\mathbf{a_{rand}}$, where, $\mathbf{z^a_{org}} = \mathbf{z_{org}} + \mathbf{a_{rand}}$. Finally, EMS runs the $\mathbf{SE-BDD}$ algorithm on the $\mathbf{z^a_{org}}$ data that makes the remapped compromised sensors in the set $\mathcal{I}^c_{org}$ outliers, as the attack vector $\mathbf{a_{rand}} \neq Hc$.

Moreover, in the case of a ideal noiseless system with sufficient redundancy, as the compromised sensors in $\mathcal{I}^c_{org}$ become outliers and get excluded from the estimation process, the rest of the clean measurement data give an estimated measurement vector, $\mathbf{z^a_{est}}$ which is the same as $\mathbf{z_{org}}$. The residual vector, $\mathbf{r_{org}} = \mathbf{z^a_{org}} - \mathbf{z^a_{est}} = \mathbf{z^a_{org}} - \mathbf{z_{org}} = \mathbf{a_{rand}}$. Thus, the residual vector is actually the randomized version of the original attack vector $\mathbf{a_{org}}$. The set of outlier sensors, $\mathbf{out_{org}}$ = $\mathcal{I}^c_{org}$, is the randomized version of the compromised sensors $\mathcal{I}^c_{rand}$.

D. Attack Detection and Localization

Here we explain how the residual/outlier data are used to detect and localize the attacks. The residual and outlier data are

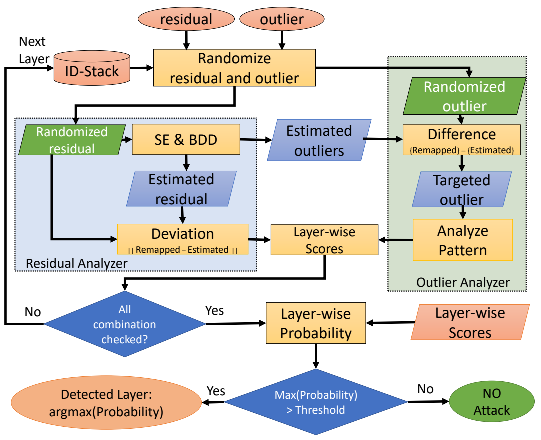


Fig. 6: Attack detection and localization.

leveraged to detect attacks and reveal the attacker's target and location. As discussed in the attack mitigation process, proper randomization makes all the compromised sensors outlier. The process is shown in Fig. 6. If we again randomize the residual vector and the outliers with the nodes' $recIDs$-$randIDs$ pairs, the randomized residual vector, $\mathbf{r_{rand}}$ goes back to the actual attack vector $\mathbf{a_{org}}$ and the randomized outliers $\mathbf{out_{rand}}$ becomes $\mathcal{I}^c_{rand}$. Now, as $\mathbf{r_{rand}} = \mathbf{a_{org}} = Hc$, if we run the $\mathbf{SE-BDD}$ algorithm on $\mathbf{r_{rand}}$ data, we get the same estimated attack vector and no outlier. This condition proves that the deceived residual is not just random system noises but a precisely calculated FDI attack vector.

However, along with the attacked sensors, if there are some other outliers $\mathbf{out_{noise}}$ due to random system noise vector $\mathbf{n}$ ($\mathbf{n} \neq Hc$), $\mathbf{SE-BDD}$ on the residual data considers $\mathbf{out_{noise}}$ as outliers due to their noncompliance values. Hence, in that case, the set $\mathbf{out_{sus}}$ = $\mathbf{out_{rand}}$ - $\mathbf{out_{noise}}$ contains the list of suspected/targeted sensors in the attack. Similarly, there is a perfect pattern in the location of the sensors in $\mathbf{out_{sus}}$ as an FDI attack vector selects the sensors following the topology. DDAF considers all the layers one by one and checks if any layer provides a pattern in the randomized residual/outlier data to detect the attacker's position in the network. If random system noises cause the residuals/outliers, none of the layers will have a high likelihood. We cannot present the detection and localization algorithm due to page limitation.

V. Example Case Study

In this section, we present an example case study explaining the overall process of DDAF for the IEEE 5 bus system. Fig. 7 shows the electrical and communication network of the considered system. Generally, each transmission line contains two sensors reporting forward and backward line power flows, and each bus has one sensor reporting the power consumption. Thus, IEEE 5 bus system consists of 5 substations (buses), 7 lines and $(2 \times lines + buses) = 19$ measurement sensors.

1) $\mathbf{recIDs-randIDs}$ *pairs and ID-Stack generation:* Fig. 8 demonstrates the $recIDs$-$randIDs$ pair at each node generated by the **randIDGen**. Here, substation 1 has three sensors with IDs $\{1, 2, 15\}$. Thus, switch $\mathcal{S}^1_1$ generates a seed

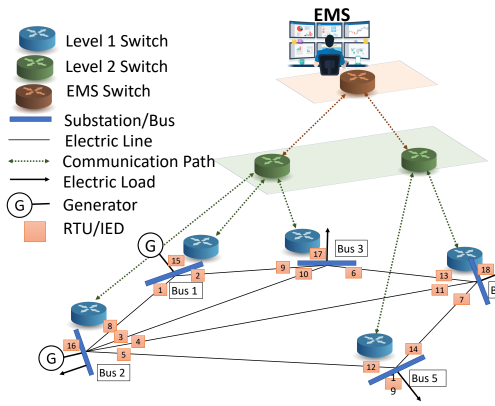


Fig. 7: SCADA network of IEEE 5 bus system.

and runs Algorithm 1 to generate the set of random IDs $\{15, 2, 1\}$, as shown with the vivid color boxes. Similarly, the rest of the nodes follow the same procedure to generate the deceptive IDs. Thus, the set of all the received and randomized IDs at level 1 are presented by $recIDs^1$ and $randIDs^1$, respectively.

As shown in the figure, the sequence of received IDs at level 2 switch $\mathcal{S}^1_2$ is {1, 2, 15, 3, 4, 5, 8, 16, 6, 9, 10, 17}. The randomized pattern of this received IDs is {1, 3, 17, 2, 4, 8, 9, 10, 15, 5, 16, 6}. According to Fig. 8, the sensors 1, 2, and 15 are reported by $\mathcal{S}^1_1$. However, $\mathcal{S}^2_1$ randomizes the pattern as 1, 3, and 17. The colors imply that these three IDs come from three different substations. Similarly, the rest of the sensors received from a single child node are distributed to multiple child nodes. Thus, the randomization at level 2 is more effective than level 1, and so on. Simultaneously, EMS generates the same seed and runs the randomization algorithm for each of the nodes to build the complete ID-Stack, as shown in Fig. 8.

2) Packet crafting and remapping: As the system executes DDAF, the deceptive sensors' packets are crafted at the nodes that belong to its path to the EMS. With the generated $recIDs$-$randIDs$ pairs, the received packets are crafted with the deceptive IDs. That means, all the level 1 switches send the original measurement data of sensor {1, 2, 15, 3,...., 18, 12, 14, 19} with deceptive IDs {15, 2, 1, 16, ..., 11, 19, 12, 14}. Similarly, the level 2 switches send the received data with sensor IDs {1, 2, 15, 3, ..., 18, 12, 14, 19} with the deceptive IDs {1, 3, 17, 2,...., 14, 7, 13, 19} and report to EMS. After collecting the crafted packets, EMS performs the remapping for all the layers in a backward direction to get

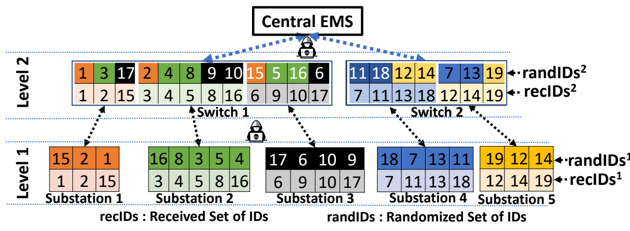


Fig. 8: ID-Stack of IEEE 5 bus system.

TABLE II: Attack Scenario for IEEE 5 Bus System

Original IDs, $\mathcal{I}_{org}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Original Sensor Data, $\mathbf{z_{org}}$	83	41	19	29	54	15	5	-83	-41	-19	-29	-54	-15	-5	125	20	-45	-40	-60
Randomized IDs, $\mathcal{I}_{rand}$	17	3	10	9	2	6	14	8	15	16	11	19	12	7	1	4	5	18	13
Actual Attack Vector, $\mathbf{a_{org}}$	**17**	**4**	0	0	0	0	0	**-17**	**-4**	0	0	0	0	0	**21**	**-17**	**-4**	0	0
Randomized Attack Vector, $\mathbf{a_{rand}}$	**-4**	0	0	**-4**	**4**	0	0	**-17**	**21**	**-17**	0	0	0	0	**17**	0	0	0	0
Remapped Sensor Data, $\mathbf{z^c_{org}}$	**79**	41	19	**25**	**58**	15	5	**-100**	**-20**	**-36**	-29	-54	-15	-5	**142**	20	-45	-40	-60
Estimated Sensor Data, $\mathbf{z_{est}}$	**83**	41	**19**	29	**54**	**15**	5	**-83**	-41	**-19**	-29	-54	-15	-5	**124**	20	-44	-39	-59
Residual Vector, $\mathbf{r_{org}}$	**-4**	0	0	**-4**	**4**	0	0	**-17**	**21**	**-17**	0	0	0	0	**17**	0	0	0	0

back to the original IDs. The remapping starts with the highest layer, which is layer 2 in this case, propagates to the first layer, and recovers the original IDs.

3) Impact of an FDI attack: Let us assume that an intruder launches an FDI attack on the communication path between layer 2 and EMS of the IEEE 5 bus system. Table II demonstrates an attack, where the original and deceived IDs are shown as $\mathcal{I}_{org}$ and $\mathcal{I}_{rand}$. The attacker plans to launch the attack by injecting 17, 4, -17, -4, 21, -17, and -4 MW to the measurement data of sensors in set $\mathcal{I}^c_{rand}$ which are 1, 2, 8, 9, 15, 16, and 17, respectively. However, due to deception at node $\mathcal{S}^2_1$, IDs in $\mathcal{I}^c_{rand}$ are used to send the original measurement data of sensors in $\mathcal{I}^c_{org} = \{15, 5, 8, 4, 9, 10, 1\}$. Thus, whenever the attacker injects into the sensors in $\mathcal{I}^c_{rand}$, the injections take place at the measurement data of sensors in $\mathcal{I}^c_{org}$.

As shown in Table II, $\mathbf{z^c_{org}}$ represents the remapped version of the compromised measurement vector. The estimation process on $\mathbf{z^c_{org}}$ makes the compromised sensors outliers, and only the good measurements prevail. Thus, the estimated measurement vector $\mathbf{z_{est}}$ remains almost the same as the original measurement vector $\mathbf{z_{org}}$, leaving no deviation in the state estimation due to the attack. Similarly, the residual vector $\mathbf{r_{org}}$ follows the pattern of the attack vector $\mathbf{a_{rand}}$, revealing the intent of the attack.

The following part shows the detection of the stealthy attack, analyzing the residual vector and outlier sensors.

- **Analyzing the Residual Vector:** EMS inspects $\mathbf{r_{org}}$ and $\mathbf{out_{org}}$ from an attacker's point of view at each level, starting from level 1. The shuffled residual vector $\mathbf{r_{rand}}$ and its estimated version for each of the layers are shown in Fig. 9. For level 1, the shuffled residual vector does not follow any topological pattern. Thus the estimated residual becomes almost zero, indicating a high deviation. On the other hand, for level 2, the estimation perfectly follows the provided residual vector, and the deviation is almost zero. A low deviation proves that the shuffled residual vector is not random noises rather a perfectly synthesized FDI attack vector injected at the communication medium between level 2 and EMS switches.
- **Analyzing the Outliers:** Similar to analyzing the residual vector task shown in the previous subsection, the actual outliers in $\mathbf{out_{org}}$ are shuffled back to different levels. The shuffled outliers, $\mathbf{out_{rand}}$ for level 1 is $\{1, 3, 5, 6, 8, 10, 15\}$ and the noisy outlier during the residual estimation,$\mathbf{out_{noise}}$ is also $\{1, 3, 5, 6, 8, 10, 15\}$. Thus, the ultimate suspected outliers, $\mathbf{out_{sus}}$ is $\mathbf{out_{rand}}$ - $\mathbf{out_{noise}}$ = $\{\}$. All the shuffled outliers are random, and there is no suspected outliers. On the other hand, for level 2, $\mathbf{out_{rand}}$ is $\{1, 2, 8, 9, 15, 16, 17\}$. As SE-BDD finds a perfect pattern in the shuffled residual data, noisy outlier $\mathbf{out_{noise}}$ is $\{\}$. In this case, $\mathbf{out_{sus}}$ is $\{1, 2, 8, 9, 15, 16, 17\}$, which is the same as $\mathbf{out_{rand}}$. The location of these suspected outliers in the electrical network is shown in Fig. 10. Whereas for level 1, there is no suspected sensor, for level 2, we find all the initial outliers as suspects, and they follow a clear topological pattern in their location. Analyzing the suspected sensors' locations and connectedness, we assign a similarity score/probability for each level, which is not discussed in detail due to space limitation. Thus, the detection algorithm finds a high probability that there is an FDI attack at level 2 (after node $\mathcal{S}^2_1$), and the attacker intended to compromise the set of sensors $\{1, 2, 8, 9, 15, 16, 17\}$.

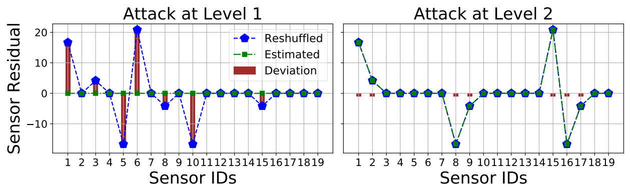


Fig. 9: Residual estimation under an FDI attack.

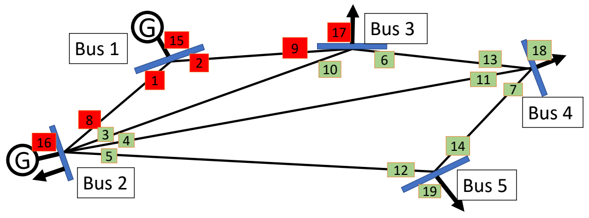
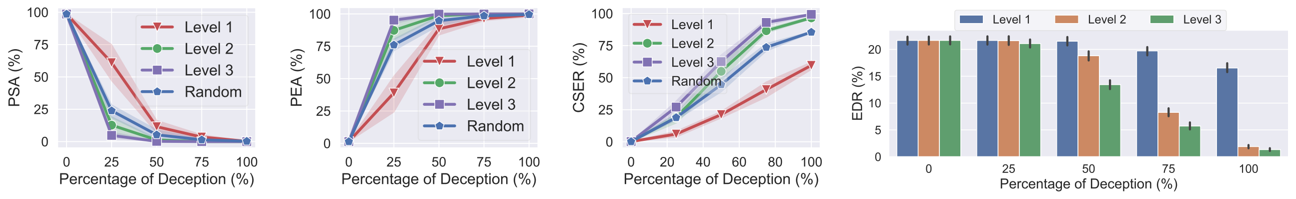


Fig. 10: Distribution of the suspected sensors for level 2.

VI. Evaluation

We evaluate the performance of DDAF on IEEE 14, 57, and 300 bus systems [27]. We also analyze the attack impacts for different amounts (0 to 100%) of sensors used in the deception process, which we define as a percentage of deception. For each test system, we implement 250 FDI attack vectors at every five seconds. We conduct our experiments on a Dell Precision 7920 Tower workstation with Intel Xeon Silver 4110 CPU @3.0GHz, 32 GB memory, 4 GB NVIDIA Quadro P1000 GPU.



(a) PSA (b) PEA (c) CSER (d) EDR

Fig. 11: Performance evaluation of DDAF for different deception and attack levels.

A. Evaluation Metrics

To assess effectiveness of deception, we consider six evaluation metrics.

Percentage of Stealthy Attacks (PSA): We consider an attack to be stealthy if all the compromised sensors remain undetected during the state estimation process. PSA is defined as:

$$PSA = \frac{\textit{\# of stealthy attacks}}{\textit{\# of total attacks}} \times 100$$

Percentage of Exposed Attacks (PEA): We consider an attack vector to be exposed if any of the compromised sensors becomes outlier and creates alarm at BDD. PEA is defined as:

$$PEA = \frac{\textit{\# of exposed attacks}}{\textit{\# of total attacks}} \times 100$$

Compromised Sensor Elimination Rate (CSER): Even though an attack is exposed, it's impact on state estimation depends on the number of eliminated compromised sensors. CSER of each attack is expressed as:

$$CSER = \frac{\textit{\# of eliminated compromised sensors}}{\textit{\# of compromised sensors}} \times 100$$

Estimation Deviation Rate (EDR): It represents the amount of the deviation in the measurement estimation due to an attack. EDR is defined as:

$$EDR = \frac{|\mathbf{z^o_{est}} - \mathbf{z^a_{est}}|}{|\mathbf{z^o_{est}}|} \times 100$$

where, $\mathbf{z^o_{est}}$ and $\mathbf{z^a_{est}}$ are the estimated measurement sets without and under the attack, respectively.

Precision is the percentage of correctly predicted positive event to the total predicted positive events.

Receiver Operating Characteristic (ROC) Curve is a way of measuring the performance of classification problems for different thresholds. The area under the ROC curve (AUC) determines separability measures for different classes.

B. Evaluation on Attack at Different Levels

In this part, we evaluate the performance of our proposed DDAF based on the attack mitigation capability.

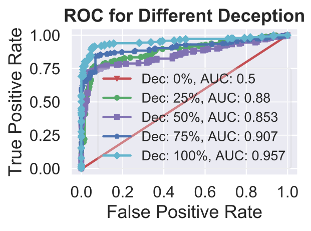
First, we evaluate DDAF's performance for the 57 bus system, considering attacks at different levels. Fig. 11a shows, irrespective of attack location, PSA decreases as the percentage of deception increases. Usually, PSA is higher for level 1 attacks as there are fewer sensors to randomize in each substation. Fig. 11b shows how the attacks get exposed, and thus, PEA increases with higher levels and more randomization. Fig. 11c shows the average CSER for each levels of injection. As figure illustrates, for attacks of level 2 and 3, almost all the compromised sensors are eliminated from the state estimation process with 80-100% deception. However, for level 1, a maximum of 83% of compromised sensors can be detected and eliminated from the system due to limited randomization. With 50% randomization, even though all the attacks get exposed (Fig. 11(a-b)), only half of the compromised sensors are eliminated (Fig. 11c), allowing the attacks to create some deviations in the state estimation. Fig. 11d shows the EDR of the attacks at different levels. Although the EDR for level 1 attacks decreases slowly with more randomized sensors, it decreases drastically for levels 2 and 3 once the randomization is more than 50%. Even though there remains a few attacks that create a little deviations in the estimation, in most cases, the EDR gets shifted toward zero and making the attacks ineffective with sufficient randomization. Moreover, although EDR for level 1 is still high, the later evaluation shows they can be detected and localized, allowing the system operator to take defensive steps.

In this step, along with the attack vectors, we inject another 250 random noise vectors, and analyze the attack detection and localization performance of the framework. Fig. 12b shows the ROC curve for attack detection at different levels. With 100% sensor deception, the AUC for levels 1, 2, and 3 are 0.861, 1.00, and 0.996, respectively. Fig. 12a shows the ROC curve for different randomization and random level (average scenario) attacks. The figures show that, the framework can detect attacks accurately (AUC: 0.88) even only with 25% sensor randomization. Fig. 12c shows the confusion matrix of detected levels of the attacks. Among 250 attacks at level 1, 203 are localized correctly as level 1 attacks. Moreover, the detection accuracy is 99% and 100% for attacks at level 2 and 3, respectively. Fig. 12d shows a high precision (around 80-98%) in the localization of the compromised sensors.

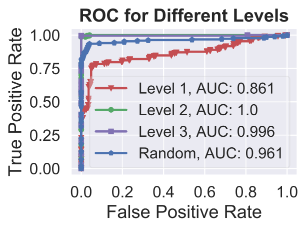
Lastly, we evaluate the framework's effectiveness against the number of nodes equipped with SDN controllers at different levels. Fig. 14 shows EDR decreases almost linearly with increasing SDN-enabled nodes into the system. 50-75% percent upgrades provide an effective defense, but a 100% percent upgrade is necessary for total resiliency.

C. Evaluation of Scalability

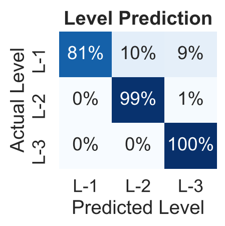
In this part, we analyze the scalability of the framework for different sizes of systems from two perspectives.



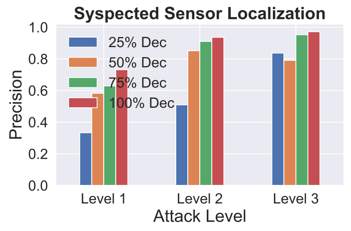
(a) Attack detection (deception)



(b) Attack detection (level)

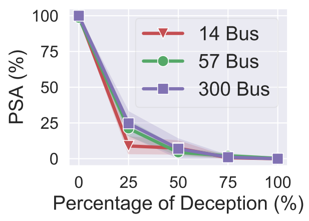


(c) Level detection

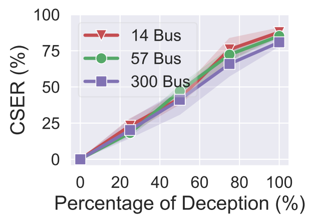


(d) Targeted sensor localization

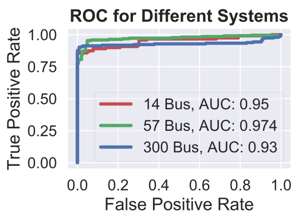
Fig. 12: Performance evaluation of DDAF in attack detection and localization for different deception and attack levels.



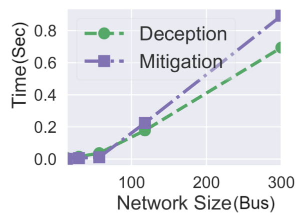
(a) PSA



(b) CSER



(c) Attack detection



(d) Running time

Fig. 13: Scalability analysis of DDAF on IEEE test systems with respect to (a-c) performance and (d) running time.

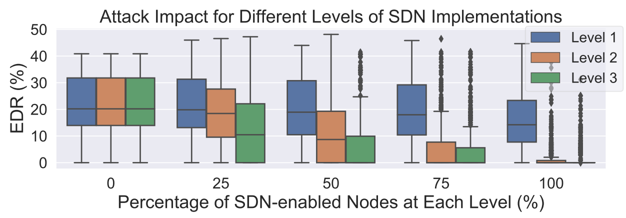


Fig. 14: Impact of SDN-enable nodes in defense.

1) Effectiveness of the defense: We analyze the framework's performance on IEEE 14, 57, and 300 bus systems. Fig. 13(a-c) show the PSA, CSER, and ROC curve for attack detection for the three cases systems under 100% deception and random level attack. Similarly, from figures, it is evident that irrespective of the network size, the framework shows similar performance in the attack detection/mitigation.

2) Time Complexity: We analyze the time needed for the deception process, as well as the reallocation process for each of the networks. Fig. 13d shows the framework's time complexity for those two tasks. The figure demonstrates almost a linear relationship among the time complexity and number of buses, proving that the framework is highly scalable and generic enough to be implemented for any size of the hierarchical network.

VII. Related Work

Moving target defence (MTD) techniques are extensively being proposed to defend different cyber-attacks for mitigating FDI attacks on the sensory channels.

There are several works introducing uncertainty/randomness in the CPS control loop. Griffioen et al. proposed an MTD introducing stochastic and time-varying parameters in the control loop of the CPS [28]. Giraldo presented MTD by randomly changing the availability of the telemetry sensor data for detecting and minimizing impacts of FDI attacks [29]. Rahman et al. reduced the window of the attack by adding uncertainty in the sensor being considered in the state estimation process [30]. Hu et al. proposed a stealthy attack detection strategy leveraging skewness coefficients can distinguish the forged residuals from the attack-free residuals [31]. In some works, authors proposed the application of D-FACTS to perturb the physical properties of the considered systems to make the stealthy attack vectors invalid [32], [33].

Some researches focused on deceiving the attacker in the MTD with crafty network packets. Li et al. used CPSMorph creating several fake network sessions along with the actual ones to hide them from attackers [34]. Pappa et al. proposed a seed-based end-to-end IP hopping among trusted peers of a SCADA system [35]. They used the seed to generate random IP and share them through a pre-installed public-private encryption channel. Groat et al. introduced MT6D as a secured IPv6 based smart grid communication system [36].

Some other research works are also performed for deceptive defense in CPS. Lin et al. proposed a randomized data acquisition into multiple rounds [37]. The software-defined network-based framework controls the network flows and collects real measurements from randomly selected online sensors, and spoof measurements for the rest. However, as only a few sensors send original data at each session, an intelligent attacker may inject false data into the online devices. Even analyzing the pattern of the sensor measurements may give the attacker the insight about the correct measurement. In another work, they proposed a physical function virtualization technique along with randomizing and crafting the decoy data to disrupt the reconnaissance attack in power grids [38]. However, an attacker can inject FDI attacks into a part of the system, ignoring the rest of the system information. Thus, their proposed approach only secures the system where virtual nodes are placed, leaving some part unsecured. However, these works consider the partial randomization of the data acquisition process and primarily focus on only the attack mitigation part. On one other hand, our proposed framework can be implemented on the overall system with complete randomization. Besides, DDAF allows the system to detect, mitigate, and localize FDI attacks in a single framework.

VIII. Conclusion

In this paper, we propose DDAF, a generic deceptive defense-based secure data acquisition framework for CPS hierarchical communication networks. Using SDN controllers at the network nodes, DDAF can deceive an attacker by replacing the original sensors IDs with the deceptive IDs. Such deception allows the system to detect, mitigate, and localize any stealthy cyber-attack. Experimental results on standard IEEE bus systems show that the framework can detect/locate most of the stealthy cyber-attacks with high accuracy. Besides, the framework is highly scalable, which can easily be implemented for large network sizes. In the future, we will explore for a data-driven intelligent technique to share the randomization pattern among the nodes. Besides, we plan to add decoy data to support the randomization.

IX. Acknowledgement

This research was partially supported by National Science Foundation (NSF) under award #1929183.

References

[1] Volkan Gunes, Steffen Peter, Tony Givargis, and Frank Vahid. A survey on concepts, applications, and challenges in cyber-physical systems. *KSII Transactions on Internet & Information Systems*, 8(12), 2014.

[2] Yao Liu, Peng Ning, and Michael K Reiter. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security (TISSEC)*, 14(1):1–33, 2011.

[3] A. K. Sikder, G. Petracca, H. Aksu, T. Jaeger, and A S. Uluagac. A survey on sensor-based threats and attacks to smart devices and applications. *IEEE Communications Surveys & Tutorials*, 2021.

[4] AKM Newaz, Amit Kumar Sikder, Mohammad Ashiqur Rahman, and A Selcuk Uluagac. A survey on security and privacy issues in modern healthcare systems: Attacks and defenses. *arXiv:2005.07359*, 2020.

[5] Mark Yampolskiy, Peter Horvath, Xenofon D Koutsoukos, Yuan Xue, and Janos Sztipanovits. Systematic analysis of cyber-attacks on cps-evaluating applicability of dfd-based approach. In *2012 5th International Symposium on Resilient Control Systems*, pages 55–62. IEEE, 2012.

[6] Md Hasan Shahriar, Md Jawwad Sadiq, and Md Forkan Uddin. Stability analysis of grid connected pv array under maximum power point tracking. In *2016 9th International Conference on Electrical and Computer Engineering (ICECE)*, pages 499–502. IEEE, 2016.

[7] Nur Imtiazul Haque, Md Hasan Shahriar, Md Golam Dastgir, Anjan Debnath, Imtiaz Parvez, Arif Sarwat, and Mohammad Ashiqur Rahman. Machine learning in generation, detection, and mitigation of cyberattacks in smart grid: A survey. *arXiv preprint arXiv:2010.00661*, 2020.

[8] M Ashiqur Rahman, Md Hasan Shahriar, and Rahat Masum. False data injection attacks against contingency analysis in power grids: poster. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 343–344, 2019.

[9] Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011.

[10] D. Drinkwater. Stuxnet-style attack. http://www.innotap.com/2015/07/stuxnet-style-attack-on-us-smart-grid-could-cost-government-1-trillion/.

[11] E Kovacs. Blackenergy malware used in ukraine power grid attacks. securityweek. https://shorturl.at/pstHS, 2016.

[12] B. Sobczak. 'denial of service' attack caused grid cyber disruption: Doe. https://www.eenews.net/stories/1060254751, 2019.

[13] A. Simmonds, P. Sandilands, and L. Van Ekert. An ontology for network security attacks. In *Asian Applied Computing Conference*. Springer.

[14] Salman Ali, Saad Bin Qaisar, Husnain Saeed, Muhammad Farhan Khan, Muhammad Naeem, and Alagan Anpalagan. Network challenges for cyber physical systems with tiny wireless devices: A case study on reliable pipeline condition monitoring. *Sensors*, 15(4):7172–7205, 2015.

[15] Iec 61850. https://www.iec.ch/smartgrid/standards/.

[16] Mohammad Ashiqur Rahman, Md Hasan Shahriar, Mohamadsaleh Jafari, and Rahat Masum. Novel attacks against contingency analysis in power grids. *arXiv preprint arXiv:1911.00928*, 2019.

[17] Md Hasan Shahriar, Nur Imtiazul Haque, Mohammad Ashiqur Rahman, and Miguel Alonso. G-ids: Generative adversarial networks assisted intrusion detection system. In *IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2020.

[18] Mohamadsaleh Jafari, Md Hassan Shahriar, Mohammad Ashiqur Rahman, and Sumit Paudyal. False relay operation attacks in power systems with high renewables. *arXiv preprint arXiv:2102.12041*, 2021.

[19] Sun Tzu, Sun Tzu, Wu Sun, Sun Cu Vu, et al. *The art of war*, volume 361. Oxford University Press, USA, 1971.

[20] Ddaf. https://sites.google.com/view/ddaf4cps/home.

[21] Ali Abur and Antonio Gomez Exposito. *Power system state estimation: theory and implementation*. CRC press, 2004.

[22] Yao Liu, Peng Ning, and Michael K. Reiter. False data injection attacks against state estimation in electric power grids. In *ACM Conference on Computer and Communications Security (CCS)*, pages 21–32, 2009.

[23] Shizhong Wang, Yuanrui Zhang, Zhihua Yang, and Yixiang Chen. A graphical hierarchical cps architecture. In *2016 International Symposium on System and Software Reliability (ISSSR)*, pages 97–105. IEEE, 2016.

[24] Nils Dorsch, Fabian K, Hanno Georg, Christian Hägerling, and Christian W. Software-defined networking for smart grid communications: Applications, challenges and advantages. In *IEEE international conference on smart grid communications (SmartGridComm)*, 2014.

[25] Jim Doherty. *SDN and NFV simplified: a visual guide to understanding software defined networks and network function virtualization*. 2016.

[26] Rsa securid suite. https://www.rsa.com/en-us/products/rsa-securid-suite.

[27] Ieee bus systems. https://icseg.iti.illinois.edu/power-cases/.

[28] Paul Griffioen, Sean Weerakkody, and Bruno Sinopoli. A moving target defense for securing cyber-physical systems. *IEEE Transactions on Automatic Control*, 2020.

[29] Jairo Giraldo, Alvaro Cardenas, and Ricardo G Sanfelice. A moving target defense to detect stealthy attacks in cyber-physical systems. In *2019 American Control Conference (ACC)*, pages 391–396. IEEE, 2019.

[30] Mohammad Ashiqur Rahman, Ehab Al-Shaer, and Rakesh B Bobba. Moving target defense for hardening the security of the power system state estimation. In *Proceedings of the First ACM Workshop on Moving Target Defense*, pages 59–68, 2014.

[31] Yan Hu, Hong Li, Hong Yang, Yuyan Sun, Limin Sun, and Zhiliang Wang. Detecting stealthy attacks against industrial control systems based on residual skewness analysis. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):74, 2019.

[32] S. Lakshminarayana and D. KY Yau. Cost-benefit analysis of moving-target defense in power grids. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE.

[33] Jue T., Rui T., Xiaohong G., and Ting L. Enhanced hidden moving target defense in smart grids. *IEEE Transactions on Smart Grid*, 2018.

[34] Yu Li, Rui Dai, and Junjie Zhang. Morphing communications of cyber-physical systems towards moving-target defense. In *IEEE International Conference on Communications (ICC)*. IEEE, 2014.

[35] Aswin Chidambaram P., Aditya A., and Manimaran G. Moving target defense for securing smart grid communications: Architecture, implementation & evaluation. In *IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2017.

[36] S. Groat, M. Dunlop, W. Urbanksi, R. Marchany, and J. Tront. Using an ipv6 moving target defense to protect the smart grid. In *IEEE PES Innovative Smart Grid Technologies (ISGT)*. IEEE, 2012.

[37] H. Lin, Z. T Kalbarczyk, and R. K. Iyer. Raincoat: Randomization of network communication in power grid cyber infrastructure to mislead attackers. *IEEE Transactions on Smart Grid*, 10(5):4893–4906, 2018.

[38] Hui Lin, Jianing Zhuang, Yih-Chun Hu, and Huayu Zhou. Defrec: Establishing physical function virtualization to disrupt reconnaissance of power grids' cyber-physical infrastructures. In *The Proceedings of 2020 Network and Distributed System Security Symposium (NDSS)*, 2020.