Architecture and Its Vulnerabilities in Smart-Lighting Systems

Florian Hofer
Free University of Bolzano-Bozen
Bolzano
florian.hofer@stud-inf.unibz.it

Barbara Russo
Free University of Bolzano-Bozen
Bolzano
barbara.russo@unibz.it

ABSTRACT

Industry 4.0 embodies one of the significant technological changes of this decade. Cyber-physical systems and the Internet Of Things are central technologies in this change that embed or connect with sensors and actuators, supporting the creation of systems-of-systems interacting with the physical environment. When it comes to applying them to the definition of new *Smart-** systems architectures, such modern technologies may impose additional requirements. These limitations mainly arise when building and interconnecting components while maintaining reliability and security of a system with heterogeneous, multi-domain nature. This paper presents an approach also applied to a case study for application-specific, layer-based security analyses, that merges results and experiences from the different involved domains. We further create a unified taxonomy and analyze an event-based distributed Smart-* system through multiple layer-based models. By applying our approach to a Smart-lighting use case, we were able to identify the specific model's architecture layers in an iterative and incremental manner and derive potential attacks, threats, and vulnerabilities from the system specifications. The result shows the ability of the technique to evaluate the presence of potential multiple-domain security concerns.

KEYWORDS

Industry 4.0, CPS, Security, Smart-City, Smart-Lighting

ACM Reference Format:
Florian Hofer and Barbara Russo. 2021. Architecture and Its Vulnerabilities in Smart-Lighting Systems. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Progressive computerization brings technology into every corner and improves the automation and performance of environmental and manufacturing processes. The German Government envisions the fourth industrial revolution as an inevitable prospect for future development. This revolution endows modern systems with "Smart" attributes to increase operational efficiency, share information, and improve their services' quality [18]. These endowments allow the creation of fully flexible production systems. They bring in new business models, services, and products by *Smart-** systems such as *Smart-Home* or *Smart-City* [21].

Smart technologies rely on Cyber-physical systems (CPS) and the Internet of Things (IoT) to achieve such goals. *Smart-** systems operate via an autonomous, decentralized decision-making process that allows for local and faster reaction and thus enables higher efficiency and production quality [19]. They run on a mesh network of intelligent devices of different make and function, requiring standardized interfacing and communication. This heterogeneity could lead to inconsistencies making a system vulnerable. System attacks can exploit vulnerabilities to eavesdrop or harm an asset's value, causing virtual and physical loss. This distress is particularly the case of *Smart-Lighting* systems, where publicly installed devices may be subject to physical and cyber-attacks [6].

Security underwent often disregards when discussing architectural proposals. A systematic mapping study identifies a lack in research on security for Industry 4.0 architectures, in particular, confirming in-field studies [17]. Existing architectural models often suffer from simplifications and assumptions from the "offline" (secure) world. Modern architectural models' needs must include security (*security by design*). However, this is a hard to achieve task in a multi-domain environment where definitions and analysis models defer between the application-relevant functional models. Thus, there is a need for guidelines to build architectures and their models that incorporate security concerns. Such guidelines would also help assess systems' vulnerabilities and propose strategic and preventive countermeasures [19] or determine corrective mitigation measures that could reduce or eliminate a vulnerability [23].

This investigation presents a layer-based analysis and classification technique of architectural vulnerabilities for multi-domain systems. The technique relies on results and experiences from the diverse involved domains. While there are specific new weaknesses that will appear when interconnecting such heterogeneous systems, in this we paper we focus on a technique that extends what we know about a system. We explore security concerns through several reference models designed for CPSs or IoT ecosystems and systematically integrating vulnerability knowledge from connected domains. As a practical example of such a process, we analyze the decentralized *Smart-Lighting* architecture of an in-field case study. Overall, the contributions of this work are:

- A unified review and classification of architectural layers of existing reference models for CPSs and IoT environments;
- An architectural analysis of a Smart-Lighting system, part of a Smart-City pilot project;
- A unified multi-domain taxonomy of vulnerabilities and attacks for the proposed Smart-lighting architecture.

We organized the rest of the paper as follows. Section 2 and 3 present related work and our methodology and evaluation strategy. In Section 4, we analyze the case study and evaluation context.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Next, we review the layer descriptions, link the layers to our case study, and create a unified taxonomy table. In Section 6, we apply our iterative classification technique using this table and discuss results and conclude in Section 7.

2 RELATED WORK

We identified three major security topics: assessment through architecture layers, (traditional) offline analysis tools, and architecture design and patterns. In addition, we select cornerstone studies and describe their relevance in Industry 4.0 in the following.

Security and layers. Lezzi et al. [19] analyze how research deals with the current cybersecurity issues in Industry 4.0 contexts, laying down the state of development regarding Smart-* architectures. The authors argue that an ideal design and development strategy considers cybersecurity from the start. The study identifies norms and guidelines for architecture security and proposes structured solution approaches along with the taxonomy of standard cybersecurity terms. Within their list of threat identification methods, they mention a three-layer-based attack assessment technique. While they do not discuss nor compare the method's efficiency further, their concluding remarks highlight the lack of an all-layer cybersecurity analysis.

Although little research exists on vulnerability classifications in these new Smart contexts, we can adopt some published results on CPS architectures. Ashibani and Mahmoud [6] redacted a generic security analysis comparing CPS technologies to traditional IT security. The article is among the first to discuss the analysis and detection of multi-layer security requirements. It identifies security requirements, possible attacks, and issues for information security on three architectural layers. However, their theoretical considerations appear limited to their feasibility, and many discussed terms had non-traceable sources.

Varga *et al.* [29] created an analog, IoT focused overview. With a fourth architecture layer for data processing, the study targets the automation domain and enlists security threats and threat mitigation. The paper displays how similar analyses can impact results from their biased viewpoint. While IoT and CPS security present similarities, the article disregards CPS typical distributed control and treats issues as binary problems making the analysis incomplete. However, the strong data-centric viewpoint helps in the assessment of data processing systems.

Han *et al.* [15] submit in their layer analysis a different aspect to vulnerabilities by classifying them as internal or external. They propose a four-plus-one layer architecture and a framework for an intrusion detection system (IDS). Due to the lack of a unique definition of CPS, the authors suggest an iterative application of appropriate mitigation strategies. Unfortunately, this iterative notion applies to IDS design only. Furthermore, even though they deliver a control-centered selection of attacks for each layer, the article also admits definition issues.

(Traditional) offline analysis tools for security and safety. Safety and security relied on design time offline analysis tools for many years, a tradition that did not change much for cybersecurity. Bolbot et al. [9] describe the relationship between the two as a conditional dependence. Their article focuses on design-time safety assurance methods, their modifications, and their integration. They identify sources of CPSs' complexity and test offline assessment techniques against them. Within the remarks of this investigation, we find the need for a systematic method for issue identification. They highlight the importance of mixing and adapting existing techniques to deal with CPS's complexities to tackle cybersecurity issues.

Subramanian and Zalewski propose in [26], and [27] an alternative assessment approach for non-functional requirements to connect security and safety in the CPS domain. The non-functional domain's well-defined ontology allows for an inter-dependency graph, which then propagates information as needed. The method shows how the dependencies of a single requirement can change an issue's weight. Majed et al. [22] suggests a framework for evaluating security exposure by weight on a connected graph. Via the shortest path, we can then identify the most accessible vulnerability. Although an interesting approach, the distribution of weight and path for each node remains unclear.

Architecture design and patterns. Alguliyev *et al.* [2] analyze and classify in a recent literature review existing research on CPS security using the CIARR model, a variant of the CIAA security requirements. This variant separates availability into resilience and reliability, suggesting that CPS's non-functional requirements vary from traditional IT. The analysis discusses approaches of architectural design to improve system security. It draws up the context and risks, offers a generalized attack tree, proposes mitigation strategies, and informs about found countermeasures and dominant future research areas.

Ryoo *et al. 2015* [24] try to break assessment conventions by proposing a generic new three-stage approach. The three phases collect information based on tactics, patterns, and vulnerabilities. The process guides an analyst through three security analysis phases with an improved weakness (CWE-1000) and entirely new architecture pattern databases. However, the method is still subject to refinement and tuning.

3 METHOD

For our layer-based technique, we processed our use-cases' *Smart-Lighting* architecture (SLA) in all its components, as illustrated in Figure 1 and described in the following. Our method consists of two significant steps: identification and classification.

Identification. As with any modern system-of-systems, a Smart-Lighting system contains multiple heterogeneous systems, each with its domain-specific constraints. Hence, we first need to analyze the system's composition by gathering components' specifications from technical data sheets and reconstructing its architecture diagram. In particular, with such analysis, each component gets assigned one or more architectural roles. For example, we will see that the LoRaWan end-node controllers (B) in Figure 2 take up two roles. They act as a communication gateway (networking) and perform some minor decentralized supervision of the connected lighting-bus devices (control).

Domain-specific security aspects further characterize it (e.g., physical tampering characterizes a light device). Based on previous work [17], we select representative research papers that propose a domain-specific layered architectural model for CPS for the investigated target to use as a reference (RM_i). Each model will hold a

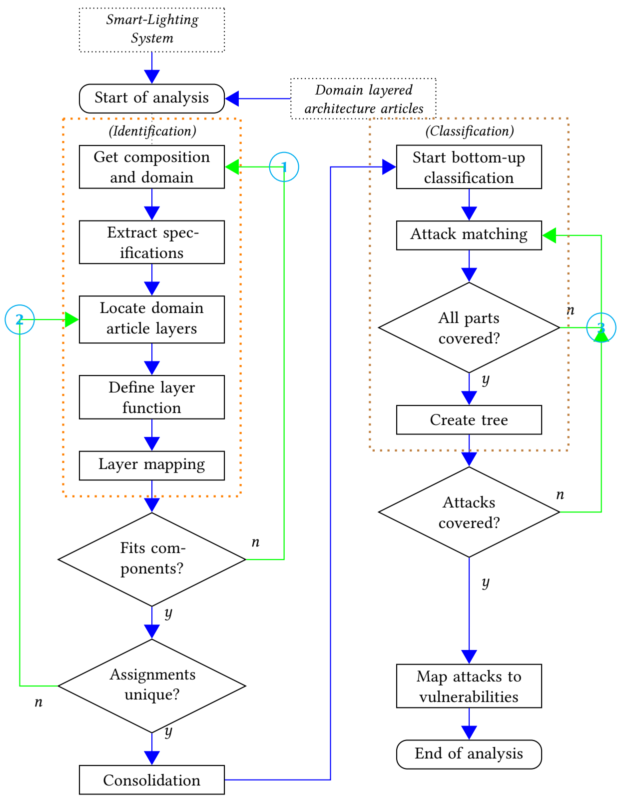

Figure 1: The flowchart shows the different steps carried out in this analysis for vulnerability identification in a Smart-Lighting system.

different architectural focus (e.g., control flow) or domain (e.g., IoT) and carries specific information on possible attacks and vulnerabilities at the layer level. Next, we build a map M_i between every component and one or more layers of each model $\Lambda(RM_i)$ via a unidirectional function, e.g., the sensor and actuator layer contains a light device.

$$M_i : SLA \longrightarrow \Lambda(RM_i) \quad (1)$$

We test each mapping and ensure that: 1) every component fits into at least one layer of a reference model (i.e., the mapping is a function applied to the layers) 2) for each layer of a reference model, there exists a component that maps into it (i.e., the mapping is a surjective function). The former claim ensures that each component can be described in each reference model and can get enriched with the information of a layer's attacks and vulnerabilities. The latter statement ensures that all attacks described in each reference model find a target in our system.

Through such layer mapping, each component now equips a role, attacks, and proposed vulnerabilities for each mapped reference model (RM_i) layer. Consequently, we can link layers from the different reference models through their common mapping to a component. Thus, in the consolidation phase, we can construct cross-mappings CM_{ij} among the reference model layers $\Lambda(RM_i)$ and $\Lambda(RM_j)$.

$$CM_{ij} : \Lambda(RM_i) \longrightarrow \Lambda(RM_j) \quad (2)$$

However, as a layer definition of one model may encircle only a subset of the definition of another model's layer, the cross-mappings are unidirectional functions that map layers from model RM_i to RM_j and may not hold in reverse. It is typically the case for models that cross-map to others with more architectural layers, a fact to consider when performing cross-mapping.

Returning to the LoRaWan controller example, mapping the selected reference models will produce different layer assignments for each component role. RM_A's generic CPS and IoT-oriented and the RM_L service-oriented model map the "Control" role to their Application layer. In contrast, the more control-oriented RM_H model maps this component to the Supervisory Control sub-layer due to its supervising function. Table 1 shows the roles assigned to a LoRaWan node and how these determine the layer mapping among the reference models, highlighting the influence of a paper's focus. While layer descriptions are similar, the focus diverges slightly between models, also reflected in attack definitions for the mapped layers. For Example, the definitions for *Malicious Code* (RM_H) and *Malicious virus/worm* (RM_L) refer to the same type of attack. However, they diverge due to focus, i.e., performance vs. data-centric, emphasizing the importance of creating a unified taxonomy.

As a result, the SLA gets enriched with information derived from its layer allocations. From the resulting cross-mapping, CM_{ij}, we create a table showing layer relationships and enrich each layer in the table with its attack taxonomy. For a clearer understanding, we further research the origin and original meaning of each attack. Such a table lets us compare model taxonomies and points up any eventual lacks and ambiguities in the corresponding definitions.

Once the table is completed with information, we iterate through the taxonomy and clear duplicates or integrate definitions. Starting from the least detailed model, we check other layers mapped to the same role and remove duplicate attack definitions or highlight differences in their definition. If an undefined term appears, we define it with the help of other domain-related and reference sources. Once we completed all layers, we created a differential attack and threat table that we can use to verify for attacks, threats, and vulnerabilities of a Smart-Lighting system. Consequently, the consolidation phase produces two outputs: a layer mapping to the architecture and between models and a taxonomy table that includes the analyzed multi-domain perspective.

Classification. With the above table, we perform a differential weakness discovery for the SLA through the component's vulnerabilities, threats, and attack options mapped to each reference model's layer. We evaluate each attack's definition and assess if, in the Smart-Lighting domain, the proposed attacks remain possible or sensible. Starting bottom-up in the architecture, we pick a network or its next component and verify each of the attacks in the differential taxonomy table for the assigned layers in the model. The process repeats until it analyzed all reference model layers and SLA components. We summarize and discuss the results of the attack analysis in a differential description that presents newly found attacks with respect to the previous model or layer.

With resulting data and based on the generic CPS attack tree created by Alguliyev *et al.* [2], we create a domain-specific attack tree for Smart-Lighting systems to highlight differences and commonality. Using their attacks-threats functional CPS model, we reuse or define further attacks and threats in the taxonomy derived from

Table 1: Example LoRaWan end-node layer role and definition differences for reference models

Role	RM_A [6]	RM_H [15]	RM_L [20]
Control	Application: [..] process the received information from the data transmission level and issue commands to be executed by the physical units, sensors and actuators.	Supervisory Control: By aggregating the measurement data from multiple points in the network, the supervisory subcontrol level creates system-level feedback control loops, which make system-level control decisions.	Application: [..] receives the data transmitted from network layer and uses the data to provide required services or operations. For instance, the application layer can provide the storage service [..] or provide the analysis service to [..] predicting the future state of physical devices.
Communication	Transmission: is responsible for interchanging and processing data between the perception and the application. [..] are achieved using local area networks, communication networks, the Internet or other existing networks [..].	Network: [..] takes charge of networking sensors and actuators as well as bridging the sensor/actuator layer and the higher control layer with a variety of communication devices and protocols.	Networking: [..] used to receive the processed information provided by perception layer and determine the routes to transmit the data and information to the IoT hub, devices, and applications via integrated networks.

our literature study. The reviewing of vulnerability definitions of the reference articles and the resulting attack tree will then serve as input for a final assessment of the possible vulnerabilities in a Smart-Lighting system.

4 THE SMART-LIGHTING ARCHITECTURE UNDER STUDY

To explain our method, we use an architecture based on a case study of a Smart-Lighting installation, part of a *Smart-City* pilot project running in the city of Merano, Italy. The project covers an area of $26km^2$ and more than 6.700 distributed lighting posts. Figure 2 illustrates the result of the identification step. The figure shows a simplified version for the demonstrative purpose of the installed system's architecture containing all the elements needed to create smart, remotely controlled lighting infrastructure. Three different networking technologies convey the control and status information between the end-nodes and the computing cloud: light devices (1-2), wireless network (3), and a traditional IP-based network (4-7).

Dali end-nodes. Digital Access Light Interface (DALI), a master-slave two-wire message-based bus for lighting and illumination systems, interconnects the light devices [8]. Its self-clocked differential encoding runs the data on a low data rate of 1200baud in half-duplex, when externally powered, for multiple hundred meters and resilient to interference [8, 16]. The DALI end device controllers (A), also called ballast controllers, execute simple application-specific programs and require only small micro-controllers [10]. In 2017, the Digital Illumination Interface Alliance (DiiA) released a revised version of the standard. DALI 2 standardizes timing requirements and signal slopes, increasing interoperability [12]. It also adds multi-master operation or multiple logical units per bus device while maintaining backward compatibility with DALI 1.

The LoRaWan network. The wireless star-of-stars network is designed on a LoRaWan (Long Range Wide Area Network) master-slave protocol that runs on top of a Semtech LoRa wireless transmitter [4]. The transmitter operates in Industrial-Scientific-Medical (ISM) band with either 250 half-duplex channels of 5.5 kbps and one at 11 kbps in chirp spread spectrum (CSS), or one channel of 50 kbps in frequency shift key (FSK) modulation (Europe channels). Its transmission robustness outperforms traditional systems, enabling servicing thousands of devices and reducing the need for

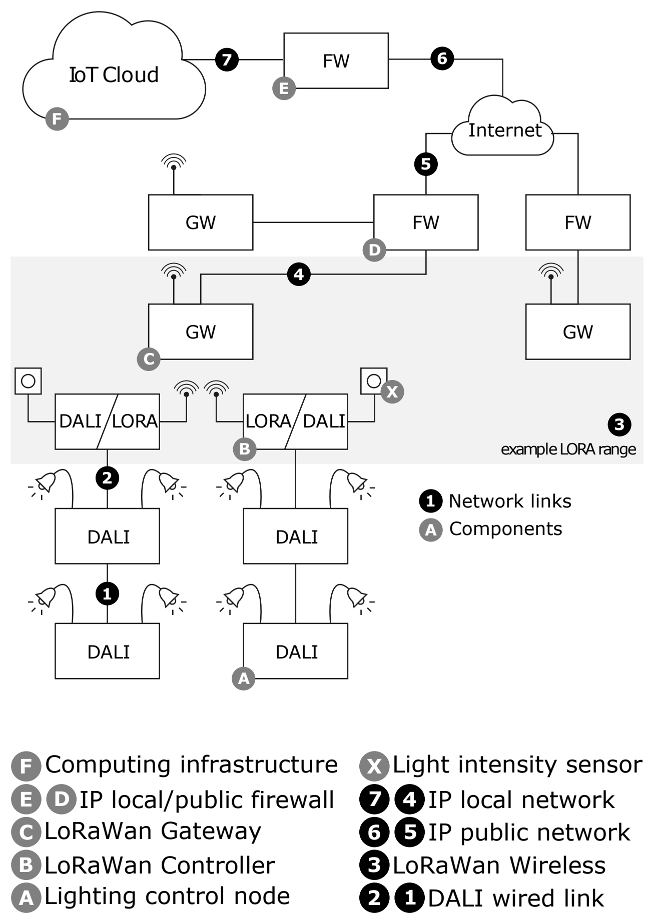

Figure 2: The architectural layout of Smart-Lighting case study

a mesh network [7]. The LoRaWan network associates nodes (B) through gateways (C) to network and application servers (F). A LoRaWan end-node (B) can have different modes: event-driven sensors, beacon scheduled actuators (usually both battery-powered), or always online. It stores two AES128 keys, securing the communication to the *network* and *application* server. The installed gateways

(C) serve as bi-directional relays and mount multichannel-multi-modem units for simultaneous reception on different frequencies and data rates without any end-node association or handover. A network server manages the distribution of data flow between an application and nodes. It reconfigures a gateway's multi-modem channels and data rate according to needs and environmental conditions. Such an adaptation targets the shortest air time (Adaptive data rate) and the best channel diversity (Channel maps) while increasing overall transmission efficiency and total throughput [4]. As the entry and exit-point of data flow are not binding, LoRaWan supports redundancy by default, though the network setup makes direct end-node communication impossible [7]. The used LoRaWan end-device mounts a LoRaWan/DALI master controller for routing and the timed control of connected DALI devices, and Bluetooth LE hardware for the initial configuration setup [28]. It offers over-the-air (OTA) firmware update and OTA device activation and features digital and analog inputs and outputs to attach optional sensor-actuator hardware. The hardware of the used LoRaWan gateway mounts an ARM Cortex-A™processor running a Linux kernel. It allows user program deployment and features a backup up-link over 4G/LTE.

The IP infrastructure. The IP-based infrastructure is configured as in a traditional IT system. Local networks use IPv4 or IPv6 connectivity through Internet (5-6) and unite firewalls with gateways (4) and computation cloud (7). Within and between networks, standard protocols (IPSec/HTTPS) secure connections. The firewalls (D-E) perform routing and protection tasks, providing traditional intrusion detection algorithms. The cloud environment (F) stores and analyzes data. The data coming from the on-site gateways enters the cloud through a software firewall, which forwards it to the "Loriot IoT" network server running as an IaaS instance. The latter forwards the message payload to a PaaS application server running an "Azure IoT" service running a set of custom-developed "Azure Functions" and micro-services. These gather and store the acquired information in a "Kosmos DB" No-SQL database and take control measures accordingly. The virtual LAN and firewall (E) configuration allow setting up internal data flow governance and additional fine-grained protection mechanisms.

5 ARCHITECTURAL MAPPINGS

We use different reference models, RM_i, originating from varying domains: one model to cover generic aspects of CPS's information security, one to highlight and stress the importance of information and control flow in CPS, and two to extend aspects peculiar to IoT, Big Data, and service orientation. The SLA maps then to the reference models, Equation 1. Figure **??** presents the result of the cross-comparison of reference models, Equation 2, showing an approximate horizontal alignment of layer roles. Reference models RM_i, mappings M_i, and cross-mappings CM_i are further detailed in the rest of this section.

RM_H, (Han *et al.* [15]): the architecture of systems arranges in a 4 plus 1 layer model, Figure **??** center. Its layer stack contains the Physical, Sensor and Actuator, Network, and Control layer. The latter divides further into three control-oriented sub-layers: Local-distributed control action layer, Supervisory sub-control level, and

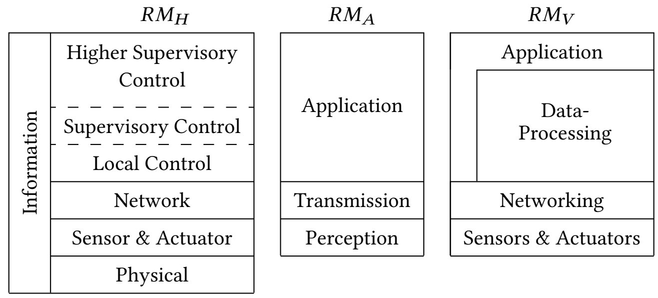

Figure 3: Comparison of layers, approximate competence alignment in respect to the SLA of Figure 2

Supervisory higher control level. This division highlights hierarchical separation and enables distributed independent control. The plus one (Information) layer interfaces transversely, acting on all four stacked layers. It represents the information flow sinking to the top and sourcing from among all layers in the architecture or vice versa, supporting the notion of shared information for distributed control.

RM_A, (Ashibani and Mahmoud [6]): uses a three-layer approach defined as the Perception, Transmission, and Application layer. Its architectural distribution is similar to RM_H in that both propose a centrally layered stack with similar features. While the authors acknowledge that three layers are not enough to abstract all CPS functionality, the model suffices to capture the functional core. Without a Physical and an Information layer, RM_A proposes a more generalized view that allocates all control and computation on the top layer.

RM_V (Varga et al. [29]): focus on IoT and distributed data acquisition. It draws on the previous three-layer model but adds a Data processing layer to take care of the vast data mole entering the IoT hub. This addition suggests a strong focus on data processing and process automation analytics.

RM_L, (Lin et al. [20]): details the aspect of service orientation in an IoT-based layered architecture. Similar to RM_V, they extend the three-layer model with an additional Service-oriented layer between Network and Application. The layer orchestrates and manages the available services to translate, process, and store incoming and outgoing data. As this role is passive, it suffers from adjacent layers' vulnerabilities, making it transparent. As we will see during mapping, RM_L ends up virtually equivalent to RM_A.

Based on these reference models, RM_i, we can now construct layer mappings for each model, M_i. We iterate through components, survey each RM for matching role descriptions for an assignment starting from the physical world. An integrated streetlight (A) senses the lamp current and actuates lamp illumination levels. A generic sensor, such as a light intensity sensor (X), captures light intensity. We map both thus to RM_H's Sensor and Actuator layer. RM_V's Sensors and Actuators layer matches the same description, while for RM_A and RM_L, the Perception layer offers the best match. The lighting device further uses lamp data and control to monitor and govern light intensity and system health. In RM_H, the Local(-distributed) Control sub-layer manages the given sensory information locally, acting as a local control entity. All other RM

refer to a single Application layer for this purpose. Figure 2 shows these devices connected to peer nodes and a master controller (B) through wired couplings (1)-(2). The latter firstly functions as a network bridge between DALI and LoRaWan. It forwards the information over wireless connections (3) and IP networks (4-5-6-7) through firewalls (D-E) and gateways (C) to the IoT Cloud. The Network layer of RM_H and RM_L best describes these devices' and links' connectivity role. It is responsible for distributing and interconnecting devices, sensors, actuators, and services the Control layer. Similar descriptions fall into place for the Transmission layer in RM_A and the Networking layer in RM_V. Secondly, LoRaWan end-node controllers (B) perform minor decentralized supervision, switching, and timing operations of the connected lighting devices. This description maps to the Supervisory Control sub-layer of RM_H to which all local controllers subside. The nodes report back to a higher instance, a business process located at the IoT cloud. This process is in charge of management and control of the system's overall operation, i.e., the city, and relates to the Supervisory Higher Control of RM_H. All other RM refer to both mentioned control sublayers to the single Application layer. RM_V and RM_L, however, have different role associations for the IoT cloud. M_V includes an assignment of the Data processing layer in charge of information pre-processing. At the same time, M_L foresees a Service-oriented architecture to manage service interaction and processes. All the above components handle or contain information of some sort. RM_H's Information layer applies thus on all components and roles. Further, the physical world is specified only in RM_H mapped, thus to the physical layer.

CM_{AH} : The perception layer maps to the Sensor and Actuator layer from Han *et al.*, the Transmission to the Network layer, and the Control to the Application layer. However, RM_A has no reference neither for the physical nor for the information layer. While the latter might blend to the existing three layers of RM_A, no notion of physical components other than sensors or actuators is present in RM_A, making this a partial extending cross-mapping.

CM_{AV}, CM_{AL} : The two IoT models show only minor mapping differences to the three-layer model of RM_A. Both map almost directly with minor differences in naming for Transmission/Networking and Perception/Sensor and Actuator. The fourth layer in both proposals shares some functionality with the lower layer. However, it communicates to the upper layer, partially parallel to RM_A's application layer. Their function distribution on the example architecture results thus almost identical and transparent. These can extend RM_As notions of attack for application layers with detail based on data processing.

We finally build a taxonomy for vulnerabilities and attacks by unifying the reference models' existing taxonomies as in the following. RM_A proposes the simplest model used as a reference. We compare its definitions with the more detailing classification of RM_H and the definition extensions for service and data-centric architectures of RM_V and RM_L. Then, in a separate spreadsheet, we align definitions, mark inconsistencies, additions, duplicates in color, finally filtering and merging them. Attacks, threats, and role descriptions of the different stages of this work are available as download[1].

[1] Industry 4.0 - Smart-Lighting Taxonomy table https://bit.ly/3nHaxjN

6 ATTACKS AND VULNERABILITIES BY NETWORK AND COMPONENTS

In this section, we iterate through the SLA in Figure 2 and verify if the attacks remain possible or sensible. Next, we analyze all three networks by components and their layers and verify the feasibility of attacks using the unified differential taxonomy. Finally, we determine threats and connected vulnerabilities, concluding with suggestions for countermeasures.

6.1 Attacks

6.1.1 DALI Network. The DALI network consists of streetlights (A) connected through a two-wire bus (1-2) to a LoRaWan end-node (B) that acts as network master, Figure 2.

RM_A [6]: The lighting control node (A) maps to the Perception and Application layers of RM_A, while the LoRaWan end-node (B) maps to the Application and Transmission layers. At the Perception layer, we mainly see two types of attacks with the node: attacks that *physically* act on the node and attacks that *virtually* interact with the node. The former requires some form of physical activity on the node where an attacker can get, alter, or make information inaccessible through node capture, tampering, or destruction. The latter type aims to interfere with the node's function by intervening in sensor measurement or corrupting data and its integrity. Physical attacks may cause information disclosure by, e.g., replacing the node with a duplicate, stealing its data, replicating its functions, and attacking information integrity through false information. Such attacks may cause system malfunction, e.g., darken specific city areas, as lampposts are publicly accessible. For virtual attacks on systems using the new DALI 2 standard, a captured or inserted false node could act as master and take over other nodes (i.e., spoofing). It may actively poll or even change another node's data and configuration, directly controlling, corrupting, and desynchronizing a network. DoS attacks, such as flooding, can take out a node and make its services unavailable. Finally, electromagnetic interference attacks influence sensory measurements and actuation control, e.g., through action on the system's resonance frequency, corrupting measured values, or feedback loops.

At the Application layer, misleading attacks and buffer overflow occur in (A) and (B). The former attacks attempt to make status or value readings unreachable (Denial of Service – DoS), forge commands, or intercept and manipulate loops through altered information (Man in the Middle – MitM). The latter inject malicious code. Such attacks to be successful require specialized knowledge of the attacked micro-controllers [10, 28]. On this layer, all attacks interact virtually with the node. Thus, an attacker with network access can systematically trial all reachable nodes.

Virtual attacks gain even more visibility on the Transmission layer. Namely, physical access to the two-wire DALI bus allows an attacker to perform DoS or selective collision attacks (including the mere cutting of wires), muting targeted nodes, and disrupting or desynchronizing control loops. Flooding, or attacker-initiated off-the-schedule polls, can quickly exhaust the network's limited relay capacity. While the simple bus intrinsically avoids routing-based vulnerabilities such as MitM or selective forwarding, its standard lacks authentication and encryption. That enables eavesdropping or, on the new DALI 2 standard, data tampering and forging control messages. Finally, it is worth mentioning that an attacker can orchestrate most of the listed attacks remotely through, e.g., a captured gateway node.

RM_H [15]: In this model, we similarly map the DALI node (A) to the Local Control and the Sensor and Actuator layers, while LoRaWan end-node (B) maps to the Network and the Supervisory-Control layers. Both components further locate in the Information layer and, while the DALI node (A) extends to the Physical layer, unique for this model, Figure **??**. RM_H does not specify further attacks on the Sensor and Actuator layer. The model redefines the desynchronization attack (called Control Forgery in RM_A) for the Control layer. The new definition calls it specifically designed to damage a system, e.g., delayed instrument readings that dis-align physical and cyber worlds. In RM_A, it only causes generic system misbehavior. For the Network layer, RM_H suggests the spoofing attack may also aid in transmitting false error messages. These messages suggest fictitious lamp failures to the supervisory control, disabling the lamp. On the Physical layer, attacks to external system components are considered. An attacker may intervene on DALI infrastructure and hinder its operation, e.g., cover a lamp. Finally, the Information layer highlights privacy issues that might arise through the information extracted from the transmitted data. For example, the presence/passage of persons in motion-activated areas hints at vacancy. It may cause burglary of adjacent housing units.

RM_V [29]: Similarly to RM_A, the lighting control node (A) maps to the Sensors and Actuators, and Application layers of RM_V, while the LoRaWan end-node (B) maps to the Application and the Networking layers. While Sensors and Actuators or Networking layer identifies no new threats, the Application layer of this model adds configuration tampering attacks for both nodes. Due to resource constraints or programming errors, the embedded code on ballasts and LoRaWan end-nodes might not verify control parameters for limits and constraints. Such attacks set invalid operating values, e.g., default illumination values to zero, disabling illumination, and threatening safety.

RM_L [20]: In this last model, we map the lighting control node (A) to the Perception and Application layers of RM_L, while the LoRaWan end-node (B) maps to the Application and Network layers. While there are no new threats on the Application layer, this model considers the implications unauthorized users may have on the network level. Like configuration attacks, unprotected DALI allows an attacker to alter device settings with comparable results on the Perception layer. The model identifies malicious code injection attacks as a source of access for multiple levels of the system. The node would act as a vehicle for diffusion on all levels. However, as for the attack in RM_A, resource constraints and specificity make it hard to predict their success.

6.1.2 LoRaWan Network. A typical SLA combines multiple gateways (C), LoRaWan end-nodes (B), and at least one Network server (F) through LoRa (3) to create a city-wide LoRaWan network, Figure 2. In addition, these LoRaWan end-nodes may feature additional sensors and actuators (X) for further illumination control or monitoring.

RM_A [6]: The LoRaWan Network, including Gateways (C) and Nodes (B), maps to the Transmission layer, while the control algorithms run on the end-node (B) and map to the Application layer. The external sensor (X) and the LoRaWan node further place on the Perception layer. On the Transmission layer, the LoRaWan network exposes to multiple availability-related attacks. Adversaries have direct access to LoRaWan running across the ether. For example, despite the robust multi-channel multi-modem gateway configuration, typical DoS attacks are feasible through multi-channel frequency jamming, intentional collision, or flooding. A random message flooding attack targeting those gateways might disrupt a network section as the latter entirely, by default, reacts to preambles and cannot handle more than ten packets at a time [25].

Suppose these messages are further "replays" of join requests (replay attack). In that case, forwards to Join or Network servers add computation burden and eventually exhaust available resources. Collision attacks have a similar overpowering effect. Unverified transmission practice on the medium and its protocol similarities to ALOHA impact severely on successful message reception, i.e., channel exhaustion at 60% load and only 18% of total capacity [7, 13]. A jamming attack is harder to perform and requires at least three parallel transmissions on the default LoRaWan frequencies close to the end device [4]. Namely, an adequately configured device will see jamming as radio interference and re-transmit on a different channel. Related attacks, such as a resonance attack, will also identify as interference and cause the same response [7]. Listen-in and analyzing this high number of re-transmitted packages enables side-channel and time analysis attacks to deduce session-key composition. However, the used two-layered encryption limits attack effectiveness. Other typical attacks for the Transmission layer, such as MitM, Sybil, and eavesdropping, remain ineffective until successful capture of the key. Despite missing keys, traffic analysis helps conclude origin, network configuration, and message function. Alternatively, an attacker can attempt node capture and tamper with its memory. The node hosts the necessary keys needed to send manipulated messages, opt to disrupt the network using valid credentials.

As for DALI, we classify attacks at the Perception layer again in two ways. First, through the same physical attacks of Section 6.1.1, node capture, tampering, or destruction, an attacker can extract secret keys and gain access to the network or a sensor [4]. Additionally, through differential power and resulting computation time analysis, an attacker can extract or estimate the keys (side-channel attack). On the other hand, most virtual attacks are not addressable in this network as the device-dedicated session key limits the joining of nodes or fake message transmission. However, a targeted DoS attack will cause collisions and force re-transmissions, finally exhausting a battery-powered node's energy. Finally, for sensors connected to the LoRaWan, electromagnetic interference attacks can influence sensory measurements and actuation control.

We have again misleading and buffer overflow attacks for a node (B) at the Application layer. However, while session-keys-protected channels harden manipulation through MitM, command forgery and interception, attempts to make status or value readings unreachable (DoS) stay valid. Thus, even though transmission requires master capabilities and session keys, the same risks for code injection as for the DALI network apply. Again, an attacker with network access can systematically trial all reachable nodes.

RM_H [15]: Again, most of the LoRaWan Network maps to the Network layer. The control algorithms are running on the end-node (B) map to the Supervisory Control Sub-layer. The external sensor (X) and the LoRaWan node place both on the Information and Sensor and Actuator layers. Furthermore, an external sensor interacts with the physical world, placing (X) on the physical layer. While the attack mapping of this model does not reveal any new threats for both Network and Sensor and Actuator Layer, we encounter privacy and policy-related issues at the Information layer, desynchronization problems at the Supervisory Control sublayer, and issues with direct intervention at the Physical layer. The Information layer is mainly protected by encryption; however, this does not stop attackers from traffic analysis; gathering event-based information such as pedestrian or vehicle passing results are helpful, e.g., to assess citizens' behavioral patterns in their neighborhood. Multiple join attempts may help an attacker de-crypt keys used for network and application sessions through excuse attacks. An adversary can tamper with sensory devices on the physical layer to manipulate measurements and influence lamp control, e.g., artificially boost sky illumination levels, tricking the system into believing that a shallow street illumination level suffices. Finally, a control issue that might emerge is a side effect of scalability. Similar to the situation described in Section 6.1.1, the size of the network influences the throughput capabilities. Even though the control loop involving LoRa is less tight, an extended period of reduced or interrupted communication with a gateway or network server could lead to unpredictable behavior.

RM_V [29]: In this model, Gateways (C), Network servers (F), and LoRaWan end-nodes (B) map to the Networking layer. The control algorithms are running on the end-node (B) map to the Application Layer. We map the external sensor (X) and the LoRaWan node on the Sensor and Actuator layer. At the Sensors and Actuators Layer, the model identifies tampering as a selected attack for node-identity theft and cloning. Similar to DALI, configuration tampering attacks at the Application layer may befall LoRaWan end-nodes with similar side effects. At the Network layer, the model adds fairness mechanism attacks and extends the definition of DoS flooding. The former attack tampers with the open-source WAN algorithm to elude medium sharing mechanisms and exhausting transmission resources. Flooding's extended definition reveals a similar purpose: malformed packets flood a targeted network or application to overload and corrupt resource availability.

RM_L [20]: Similar to RM_V, the main components of the LoRaWan Network (B, C, F) map to the Network layer, the control algorithms are running on the end-node (B) map to the Application layer. The external sensor (X) and the LoRaWan node place on the Perception layer. While there are no new threats on the Application and Network layer, the model identifies malicious code injection attacks on the Perception layer as a source of access to multiple system levels and similar constraints to the DALI network.

6.1.3 *IP-Based Infrastructure.*

The most traditional network in our SLA, the IP infrastructure, connects multiple IP-based devices. It transports data between the on-site LoRaWan gateways (C) and the IoT computing cloud (F) through dedicated firewalls (D-E), Figure 2. In addition, the network is in charge of a higher level of connectivity, servicing LoRaWan and DALI for the applications supervising the city's lighting.

RM_A [6]: This first model maps the Gateway (C), Firewalls (D-E), and IoT Cloud (F) to the Transmission Layer. The Application layer is further present on the IoT Cloud (F). On the Transmission layer, we find typical communication-related attacks that target resource availability or intercept or manipulate messages. Most parts of the network apply double-encryption, making attacks such as MitM and eavesdropping onerous. Routing-based attacks are most effective on routed LAN packets, available at the Cloud internal LAN (7). Here selective forwarding, routing, sinkhole, wormhole, replay, spoofing, or compromised key attacks could occur. They help an attacker to weaken and delay network traffic or reroute data for traffic and side-channel analysis. If integrated with traffic analysis, such attacks get more efficient and difficult to detect.

Besides, despite tunneling and encryption, most of the DoS attacks keep their effectiveness. A typical DDoS attack could target VPN end-points, e.g., FW (E), which makes up a single point of failure for the two sub-nets, and a bottleneck on high traffic. Similarly, all network components are susceptible to exhaustion attacks. Finally, tampering and node capture, e.g., the external firewall, could help acquire stored secrets and, e.g., reroute VPN tunnels for general data capture. The primary function of the Application layer is storing and elaboration of information. Primary attacks to this layer identify thus as Database attacks, including data alteration and User Privacy leakage through data mining on the sensed data. Via malicious code on shared instances or buffer overflow and consequent code injection, an attacker may gain access to a system.

Furthermore, along with continuously more service-oriented systems, service discovery spoofing helps integrate malicious services into the system, gathering data access. Replayed messages on this service plane may help an attacker to get the trust of the system. Message interception and alteration (MitM) and eavesdropping can cause data leaks or corruption. A malicious service can flood other services until exhaustion, making them unavailable. Such attacks' effectiveness depends on the architecture and implementation of the data processing cloud, not specified by any examined standard.

RM_H [15]: RM_H maps Gateways (C) and Firewalls (D-E), as well as the IoT Cloud (F), to the Network layer. All components further map to the Information layer. The IoT Cloud finally hosts the Higher Supervisory Control Sub-layer. Although no new attacks are present in the Information layer, the Network layer presents re-definitions of Sybil and spoofing. For example, at the inter-VM LAN connection (7), injected routing error messages make the grid seem partially offline. At the same time, Sybil attacks target fake network size. On the Control layer, the system keeps being subject to desynchronization attacks. An attacker can, e.g., tamper with time-servers to misalign lamp control from status.

RM_V [29]: Varga et al.'s interpretation of layers sees the Networking layer on Gateway (C), Firewalls (D-E), and IoT Cloud. Besides the Application layer, the IoT Cloud further hosts the Data Processing layer, separating human supervision from the computation. RM_V's application layer considers user interaction with system and data separately from its computation. Thus, if a user connects remotely to the system, a new path opens, allowing network-based threats like for RM_A, including eavesdropping, MitM, routing, or system

Figure 4: Attack tree for Smart-Lighting, modified (gray), blurred removed from [2].

exhaustion attacks. The new terminal may further be affected by configuration tampering attacks, attempting to remotely influence the lighting system's function. On the Data processing layer, we identify Malware attacks again to gain system-level access. RM_V further highlights the interactions and attacks that might occur inter-VM and based on shared resources' contention. The former include instant-on gap attacks, where due to performance concerns, immediate demand requirements allow initial unrestrained executions. The latter rely on the exhaustion of shared resources. As a result, the attacked service is depleted and unable to perform the requested services. Another mentioned attack, exhaustion flooding, achieves a similar result. The flooding with requests requires additional resources, slows down the system, and finally exhausts all resources. Side-channel attacks could extract information from non-sanitized shared memory or CPU caches among the VMs. The model does not include additional attacks for the Networking layer.

RM_L [20]: The software-oriented architecture locates the IoT Cloud at the SoA and Application layer. In contrast, Gateway, Firewalls, and IoT Cloud locate on the Network layer. This model sees user-focused attacks on the application layer during client interaction. They try to leak data and capture user access data through infected emails, phishing websites, and malicious scripts. The model then adds two more definitions on the Network layer: the sinkhole attack, as a maneuver to get more input data routed through for, e.g., traffic analysis and device tampering, to secure a device's configuration data and secrets, and consequently, gain unauthorized access to devices and networks.

6.2 Attack tree and Vulnerabilities

6.2.1 Attack tree for Smart-lighting architecture. Inspired by the attack and threat tree developed by Alguliyev *et al.* [2], we created a modified version dedicated to SLA attacks. The tree in Fig 4 illustrates the resulting attacks-threats CPS functional model for SLAs, where threats directly result from attacks. The gray highlighting in the figure marks alterations w.r.t. the original, i.e., renamed or relocated branches.

Attacks on actuation. Our SLA of Figure 2 contains two actuators: DALI ballasts that control the lamps and LoRaWan timed controllers to manage these ballasts. Both are installed mostly on or near a light pole. A threat of *Tampering with Hardware* results when physical interaction with the node can occlude actuation. An attacker can manipulate a LoRaWan end-node or DALI ballast to take control, disable or extract secrets with device tampering or node destruction attacks. *Tampering with Software* occurs when changes on it make actuation non-functional. For instance, Integrity attacks on a lamp-driving LoRaWan-node can cause incorrect configuration of lamp switching times, impeding proper lighting. Finally, *Interception of compromising interference signals* refers to actuation instability caused by external intervention on the actuator signals in closed-loop systems. For example, an attacker can destabilize lamp control through command-control forgery attacks on DALI ballast and manipulating switching behavior.

Attacks on Communication: The communication infrastructure of our SLA is represented by DALI, LoRaWan, and IP-network infrastructures. These include bridges between DALI, LoRaWan, and IP-based components, i.e., LoRaWan end-nodes and Gateways, the two firewalls, and the Internet. In addition, all connections, except DALI, are encrypted at least once; AES128-CBC for LoRaWan, IPSEC, and HTTPS for IP-based connectivity. *Information exposure* refers to the threat that allows data gathering on a non-protected communication channel. An attacker can listen in and obtain information on system and encryption passively via an eavesdropping attack on DALI networks or actively through polling via replay attacks on LoRaWan channels. *Behavior spying* results when an attacker can gather long-term information on the system's operation, people, and activity remotely. Via traffic analysis attacks, e.g., an adversary, can inspect the event-based transmissions of a LoRaWan end-node that reports on pedestrian movement. As stated in Section 6.1, such circumstantial information can help determine citizens' whereabouts for planned burglary. *Software malfunction* results from circumstances that cause incoherent, incomplete, or timely inadequate data transmission that inhibit the system's correct operation. Typical attacks that might cause such behavior are selective forwarding or flooding attacks, applicable on every link on the IP network, or collision attacks that delay the successful reception of event-based packets from the LoRaWan end-node until a successful re-transmission attempt. The threat of *Corruption of data* occurs when an attacker can manipulate information and thus void data integrity. On DALI networks, e.g., the adversary could easily tamper with the transit data as the protocol has no encryption or access control. *Interception of compromising interference signals*, again, refers to communication instability caused by external intervention on the data transmission. Such instability can be caused by jamming attacks on the LoRaWan network or spoofing attacks on the IP connectivity and flooding attack with consequent loss or alteration of packets or connectivity.

Attacks on feedback: Feedback refers to the control function that Cyber-physical systems perform when acting through actuators on sensory input or computational status changes. These include, thus, control algorithms and systems for their implementation. *Control disruption* occurs when the system cannot react to sensory input or status changes, thus destabilizing a system. Via a control-command forgery attack, an attacker could, e.g., manipulate the status of a DALI ballast, desynchronizing feedback control and influencing correct actuation.

Attacks on Computing: Computing refers to the equipment used for data storage and elaboration. Cloud services and infrastructure (F) serve data mining, user interaction, and process performance improvement. The threat of *Corruption of data* refers to manipulating information, stored and computed values, e.g., programmed light switching times, to secretly damage the system. A data tampering or integrity attack can alter stored control information. The *Equipment failure* occurs when the computing infrastructure is unable to fulfill the requested computation task. These failures can happen due to physical wear-out and resource exhaustion, an attack that depletes computing resources. *Software malfunction*, yet, results when the computation does execute as requested, but not correctly. These malfunctions are often caused by bugs but can also be due to malicious code installed in the cloud servers, e.g., viruses and trojans, that tamper with software functionality. Finally, *Illegal data processing* happens when an unauthorized agent or a user accesses more than the allowed amount of resources and data and discloses user privacy. For example, such exposure can be a consequence of installed malware (Worms) or an attacker that performs side-channel attacks. A malign virtual machine on the shared cloud tap shared memory and manipulate the computing instance.

Attacks on Sensing: Sensing in our SLA is performed on two locations: DALI ballasts that inform about the lamps' real-time data, and LoRaWan controllers, sometimes battery-powered, that sense the environment, e.g., luminosity or movement sensors. *Loss of Power Supply* is relevant for devices with reduced energy resources that may suffer from energy exhaustion and fail service. Battery-powered LoRaWan end-node may experience an outage due to forced repeated transmissions through LoRa jamming attacks that sleep-deprived the node. *Equipment failure*, yet, refers to the total inoperability of nodes and their inability to perform the required task. A node outage attack can put a LoRaWan or DALI node out of order via physical destruction. *Tampering with hardware* on sensing identifies issues that might arise when hardware modifications impede correct measurement. Direct physical intervention attacks can cover a lighting sensor, making it inoperable. *Unauthorized actions* recall the possibility of prohibited intervention on sensors that access or alter data, misuse the node, or impede its function. The sensing configuration data on the unprotected DALI nodes can be manipulated through data tampering attacks during writes on the bus link, altering measured results. The same attack can also be the source of other threats. *Equipment malfunction* is the result of incorrect sensing due to technical hindrance. Tampering with a sensor's configuration would cause sensing to fail its function. Finally, we subject to the *Disturbance due to radiation* when an attacker interferes with the normal sensory function by manipulating the measured physical unit. The LoRaWan node. e.g., can be fooled through a physical direct intervention attack, irradiating the luminosity sensor with a torch.

6.2.2 Vulnerabilities for Smart-lighting architecture. After the evaluation of attacks and threats for this SLA, we now identify the causing vulnerabilities. Tracing vulnerability descriptions from the related papers [6, 15], we align threats and attacks to detect possible vulnerabilities of our system.

At the perception and transmission layers of RM_A, most of the attacks identified have two common causes: the low resource constraint the devices withhold and their physical size and exposure. Resource limitation is mostly the enabler of attacks that hinder proper communication, protection, and access control. Unprotected DALI allows an attacker to eavesdrop or inject any command or data. Targeted LoRa or DALI network attacks can deplete available communication or energy resources, disabling parts of the network and feedback control. Similarly, the limited ether availability constraints the transmission capacity of LoRaWan and eases the attacker's channel interference.

Furthermore, the large scale of an SLA contributes to resource scarcity. It increases channel contention and utilization and coexistence problems [14], finally forcing air-time management or transmission power throttling to reduce range and interference rate. The Wide distribution of a Lighting system conduces to the vulnerability of physical exposure. Unattended areas ease network integrity attacks through device tampering, targeted interference, and device destruction. It makes nodes accessible and allows for physical interaction, altering measurements and feedback. Similarly, on the transmission layer, the SLA's wide distribution and large scale cause LoRa's ether resources to incur bottlenecks if an incorrect device configuration neglects available channels. The same holds for gateway setup where incorrect settings can ease preamble-based resource availability attacks.

Software bugs and inconsistent protocols may enable unauthorized access to infrastructure and information on the transmission and application layer. Human-made error or incorrect device configuration may allow attackers to access systems due to incorrect or mixed permissions schemes or cause system failure. Further vulnerabilities present at the IP and Cloud infrastructure are mostly the typical issues encountered in modern systems. We find missing specification details for the software components running the Smart-* architecture's back-end in addition to service attacks and information leakage issues. Indeed, the two non-standard components, an IDS (D-E) for CPS and the network server (F), have not been defined thoroughly in their specification and architecture [3, 15]. While we can secure the rest of the IP system by applying traditional architectural patterns and techniques, these two components suffer from inconsistent or incomplete specifications.

6.3 Reflection on countermeasures

This section reflects on some countermeasures specific to the Smart-Lighting system's weaknesses under study that we leverage from the analysis in the previous sections, the existing literature, and specifications of the technologies. The list should not be seen as exhaustive.

At the lamp end-posts, we have to deal primarily with physical exposure where the DALI bus, the controllers, and sensors. Using cabinets and locks that require a specific tool or key and mounting controllers at height might impede immediate access to wires and devices, reducing the risk of physical destruction and tampering. Wires should further be carried through shielded conducts, diminishing the risk of interference. Unfortunately, the resource constraints and the limiting standards do not permit protection measures against eavesdropping or MitM attacks; a replacement with more powerful hardware could significantly impact unit installation cost and solution attractiveness.

One main point that helps mitigate the attacks on the limited ether availability is a balanced configuration of the LoRaWan network. The LoRaWan standard provides the network server with the ability to reconfigure channels and optimize ether usage for gateways and end-nodes. However, there is no binding requirement for such capability. To our knowledge, no network server product includes an automatic channel distribution on gateways and nodes.

Proper distribution of bandwidth and frequencies can drastically increase the resilience of infrastructure. The sixteen-plus available settings-slots provided by the LoRaWan standard allow a complementary configuration of adjacent gateways. For better resilience, each node should reach at least two gateways using the minimum spread factor on non-adjacent channels. This approach increases the communication robustness via the high channel selectivity of LoRa and the switchable, more robust, higher symbol rates. Such a setup makes it easy to increase SNR by the selection of a higher SF. Moreover, it reduces the risk that jamming or the interference on one or more frequencies impedes the reliable transmission through a secondary channel [4].

Likewise, adjacent nodes' down-link and up-link settings should be distributed equally among reachable gateways and channels. End-nodes typically communicate on two different channels: one for uploading and downloading the node's payload to the gateway and a second shared RX window from the gateway to all nodes. This second link adds resilience to the network. As long as one down-link is available, the network server can reconfigure a node to a new up-link frequency [4]. If possible, node channel configurations should contain a disabled configuration of all gateway channels in reach. Disabled channels are automatically enabled after several unsuccessful transmissions, empowering a node in distress to reach all available gateways.

An algorithm running on the network server may manage such additional channel configurations to exploit maximum robustness. It might use geo-information and empirical measurement results to compute channel distribution appropriately and send updates over the secondary RX window. An algorithm for this purpose has been developed by Demetri et al. [11]. It approximates signal coverage and considers the environment, locations, buildings, and city structure through satellite imaging and experimental measurements. To avoid the issue of limited throughput and co-existence interference, the number of nodes per channel and gateway should also be equally distributed [7]. A tool called LoRaSim by the university of Lancaster [2] helps this purpose. Although the tool does not consider the environmental situation, it can verify if a configuration is viable. It selects optimal frequencies, captures situations of hidden terminals and exposed nodes, and determines the best-case range and coverage for a given network configuration. Lower spread factor and less interference reduce required air-time and repetition. A service that incorporates such an algorithm could improve overall resilience, optimize hardware use and increase end-node battery lifetime.

Unfortunately, LoRa (physical layer) and specification-dependent vulnerabilities cannot directly be dealt with. The specification of protocols is an alliance product (DiiA and LoRa Alliance) and might be open to improvement proposals [4]. At the moment, different proposals exist for both vulnerabilities [1]. The alliance also recently proposed an intra-channel hopping technique (FHSS) to mitigate collisions and contention [5]. The higher robustness comes at a price of a very low throughput of only a few hundred bps. It promises to be an elegant solution for high-density and coexisting networks. However, such changes need time for validation and processing and can therefore be considered only in the long run.

Simple stateful packet inspection is not enough for a CPS's IP-based network. Han *et al.* [15] identifies security challenges not uniquely at the border to the external networks, but everywhere in this complex interconnected and heterogeneous system. Therefore, intrusion detection must be entwined in the whole CPS system according to each node's limits. As seen in Section 6.1.2, each node could be tampered with generating invalid data. Thus, the solution extends from brute physical force and consequent failures to uncertain information degraded and influencing a system's control. Finally, border firewalls are often the responsible routing point for point-to-point networks. Ideally, to avoid bottlenecks and targeting attacks, multiple connections between IP-based networks should be created, routing traffic as needed.

The final set of discussed vulnerabilities connects solely to the application layer. Most of the software modules of the control units and in the application cloud work with parameters. To avoid that those settings are invalid, ideally, the final device or application that uses the information must verify correctness. Han *et al.* see this also as a possible application for an IDS. An adversary could inject invalid values to cause a control deviation or misbehavior. However, the limited resources make a distributed IDS difficult on some devices. Therefore, based on resource availability, a parameter check, a distributed IDS, or both should be installed.

More problems arise if the specifications for these software modules have errors or are incomplete. Unfortunately, in this case, the specifications should also follow standards and might suffer from this dependency. Nevertheless, many details can be derived and adapted following best practices and generalizations, leaned from experience with similar installations and architectures. Multi-tenant microservice-based systems are popular in cloud-based computation, making them an excellent architectural template source. Therefore, implementing a computing cloud infrastructure could be derived from microservice-based architectures for data elaboration, integrated with the knowledge gained from running experiments and prototypes. These should finally help achieve the highest security standards without impacting the overall performance. Lastly, most software is following new technology trends, subject to multiple changes in a short time, and suffering from high defect probability. Therefore, agile practice and testing tool-chains are the only suggestions to be given from the development standpoint.

[2]https://www.lancaster.ac.uk/scc/sites/lora/lorasim.html

7 DISCUSSION AND CONCLUSIONS

This security analysis presented a technique for the offline analysis of a Smart-* multi-domain system-of-systems. We proposed an approach that relied on the connected domains' experiences and performed a layer-based cross-analysis on a Smart-Lighting use case. Using four-layered architecture modeling approaches, we identified architectural roles, assigned model layers. We created a unified taxonomy that reflects and extends attack definitions, threats, and vulnerabilities of each involved domain. Finally, we determined possible attacks, valid threats and discussed vulnerabilities and first possible countermeasures for the merged-domain Smart-Lighting architecture in an iterative process.

After the execution of our analysis, we can assess three significant discoveries for Industry 4.0. Firstly, the domain overlapping configuration of such a system-of-systems makes it infeasible to cover all threats and attacks based on a single domain's viewpoint. The integrative approach we presented detected more issues than a single model would. Interestingly, we find the central definition of diversity in the "cyber"-layers, where computation and decision occur, while most data exchange and physical interaction layers remain unchanged. This consistency is probably because gathering, actuation, and data transport are a joint function of all four analyzed papers. When integrating future analyses with other studies, we expect changes in the upper architecture layers only. Secondly, the changing focus of the discussed models highlights aspects of a heterogeneous system. It proves that the new multi-domain architecture inherits many, if not all, characteristics of the involved domains. For example, Cloud-security issues are not a typical concern for traditional control-oriented CPS. Thirdly, vulnerabilities, threats, and attacks may alter definition, range, and weight depending on the application domain. We have seen in the taxonomy table and Section 6.2 how similar threat or attack names can have different definitions and applications that the domain of origin might influence. It is thus reasonable to pre-define and clarify all taxonomy before reaching conclusions. However, as the resulting multi-domain taxonomy is a product of role, layer, and attack allocation of the involved reference models, each new system-of-systems analysis requires repeating or refining the present analysis.

Future work will test and extend the results of this analysis. Through a second study case, we will analyze the change and variability of detected issues. Simultaneously, on-site tests will help validate the extent and risks of the vulnerabilities involved.

ACKNOWLEDGMENT

We thank Systems S.r.l for the funding and support of this project under the "Industry 4.0 for the Smart-* (I4S)" project.

REFERENCES

[1] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. 2017. Understanding the Limits of LoRaWAN. *IEEE Communications Magazine* 55, 9 (2017), 34–40. https://doi.org/10.1109/mcom.2017.1600613

[2] Rasim Alguliyev, Yadigar Imamverdiyev, and Lyudmila Sukhostat. 2018. Cyber-physical systems and their security issues. *Computers in Industry* 100 (sep 2018), 212–223. https://doi.org/10.1016/j.compind.2018.04.017

[3] LoRa Alliance. 2017. *LoRaWAN™ Backend Interfaces 1.0 Specification.* Technical Report. LoRa Alliance.

[4] LoRa Alliance. 2017. *LoRaWAN™ Specification 1.1.* Technical Report. LoRa Allinance.

[5] LoRa Alliance. 2020. *LoRaWAN™ Regional Parameter Specification 1.0.2.* Technical Report. LoRa Allinance.

[6] Yosef Ashibani and Qusay H. Mahmoud. 2017. Cyber physical systems security: Analysis, challenges and solutions. *Computers & Security* 68 (jul 2017), 81–97. https://doi.org/10.1016/j.cose.2017.04.005

[7] Aloÿs Augustin, Jiazi Yi, Thomas Clausen, and William Townsley. 2016. A Study of LoRa: Long Range & Low Power Networks for the Internet of Things. *Sensors* 16, 9 (sep 2016), 1466. https://doi.org/10.3390/s16091466

[8] Francisco Jose Bellido-Outeirino, Jose Maria Flores-Arias, Francisco Domingo-Perez, Aurora Gil-de Castro, and Antonio Moreno-Munoz. 2012. Building Lighting and Automation through the Integration and of DALI and with Wireless Sensor Networks. IEEE.

[9] Victor Bolbot, Gerasimos Theotokatos, Luminita Manuela Bujorianu, Evangelos Boulougouris, and Dracos Vassalos. 2019. Vulnerabilities and safety assurance methods in Cyber-Physical Systems: A comprehensive review. *Reliability Engineering & System Safety* 182 (feb 2019), 179–193. https://doi.org/10.1016/j.ress.2018.09.004

[10] Cecilia Contenti. 2002. Digitally Addressable DALI Dimming Ballast. IEEE.

[11] Silvia Demetri, Marco Zúñiga, Gian Pietro Picco, Fernando Kuipers, Lorenzo Bruzzone, and Thomas Telkamp. 2019. Automated Estimation of Link Quality for LoRa:A Remote Sensing Approach. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks - IPSN '19*. ACM Press. https://doi.org/10.1145/3302506.3310396

[12] DiiA. 2018. *DALI-2: The differences and new version of the DALI standard.* Technical Report. Digital Illumumination Interface Alliance. https://www.digitalilluminationinterface.org/data/downloadables/5/4/1711_technical-note-dali-2-the-new-standard.pdf

[13] Chékra El Fehri, Mohamed Kassab, Slim Abdellatif, Pascal Berthou, and Abdelfettah Belghith. 2018. LoRa technology MAC layer operations and Research issues. *Procedia Computer Science* 130 (2018), 1096–1101. https://doi.org/10.1016/j.procs.2018.04.162

[14] Branden Ghena, Joshua Adkins, Longfei Shangguan, Kyle Jamieson, Philip Levis, and Prabal Dutta. 2019. Challenge: Unlicensed LPWANs Are Not Yet the Path to Ubiquitous Connectivity. In *The 25th Annual International Conference on Mobile Computing and Networking (MobiCom '19)*. Association for Computing Machinery, New York, NY, USA, Article 43, 12 pages. https://doi.org/10.1145/3300061.3345444

[15] Song Han, Miao Xie, Hsiao-Hwa Chen, and Yun Ling. 2014. Intrusion Detection in Cyber-Physical Systems: Techniques and Challenges. *IEEE Systems Journal* 8, 4 (dec 2014), 1052–1062. https://doi.org/10.1109/jsyst.2013.2257594

[16] P. F. Hein. 2001. DALI - A Digital and Addressable Lighting and Interface for Lighting and Electronics. In *Industry Applications Conference, 2001. Thirty-Sixth IAS Annual Meeting. Conference Record of the 2001 IEEE*, Vol. 2. IEEE, IEEE, 901–905.

[17] Florian Hofer. 2018. Architecture, technologies and challenges for cyber-physical systems in Industry 4.0 - A systematic mapping study. In *12th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. https://doi.org/10.1145/3239235.3239242

[18] Jay Lee, Behrad Bagheri, and Hung-An Kao. 2015. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters* 3 (jan 2015), 18 – 23. https://doi.org/10.1016/j.mfglet.2014.12.001

[19] Marianna Lezzi, Mariangela Lazoi, and Angelo Corallo. 2018. Cybersecurity for Industry 4.0 in the current literature: A reference framework. *Computers in Industry* 103 (dec 2018), 97–110. https://doi.org/10.1016/j.compind.2018.09.004

[20] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. 2017. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal* 4, 5 (oct 2017), 1125–1142. https://doi.org/10.1109/jiot.2017.2683200

[21] Yang Lu. 2017. Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration* 6 (June 2017), 1–10. https://doi.org/10.1016/j.jii.2017.04.005

[22] Shahir Majed, Suhaimi Ibrahim, and Mohamed Shaaban. 2014. Energy Smart Grid Cyber-Threat Exposure Analysis and Evaluation Framework. In *Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services - iiWAS '14*. ACM Press. https://doi.org/10.1145/2684200.2684308

[23] John Moteff. 2005. *Risk management and critical infrastructure protection: Assessing, integrating, and managing threats, vulnerabilities and consequences.* Technical Report. Congressional Research Service - The Library of Congress.

[24] Jungwoo Ryoo, Rick Kazman, and Priya Anand. 2015. Architectural Analysis for Security. *IEEE Security & Privacy* 13, 6 (nov 2015), 52–59. https://doi.org/10.1109/msp.2015.126

[25] Semtech. 2017. *SX 1301 Datasheet.* Technical Report. Semtech.

[26] Nary Subramanian and Janusz Zalewski. 2016. Quantitative Assessment of Safety and Security of System Architectures for Cyberphysical Systems Using the NFR Approach. *IEEE Systems Journal* 10, 2 (jun 2016), 397–409. https://doi.org/10.1109/jsyst.2013.2294628

[27] Nary Subramanian and Janusz Zalewski. 2018. Safety and Security Analysis of Control Chains in SCADA Using the NFR Approach. *IFAC-PapersOnLine* 51, 6

(2018), 214–219. https://doi.org/10.1016/j.ifacol.2018.07.156

[28] NAS Nordic Automation Systems. [n.d.]. UL20x0 - LoRaWan™ luminaire controller.

[29] Pal Varga, Sandor Plosz, Gabor Soos, and Csaba Hegedus. 2017. Security threats and issues in automation IoT. In *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, NA (Ed.). IEEE. https://doi.org/10.1109/wfcs.2017.7991968