electronics

MDPI

Article

Introducing Security Mechanisms in OpenFog-Compliant Smart Buildings

Imanol Martín Toral [1], Isidro Calvo [1,*], Eneko Villar [2], Jose Miguel Gil-García [3] and Oscar Barambones [1]

[1] Department of Systems Engineering and Automatic Control, Faculty of Engineering of Vitoria-Gasteiz, University of the Basque Country (UPV/EHU), 01006 Vitoria-Gasteiz, Spain; imanol.martint@ehu.eus (I.M.T.); oscar.barambones@ehu.eus (O.B.)
[2] Aeronautical Technologies Centre (CTA), 01510 Miñano Mayor, Spain; eneko.villar@cta.aero
[3] Department of Electronic Technology, Faculty of Engineering of Vitoria-Gasteiz, University of the Basque Country (UPV/EHU), 01006 Vitoria-Gasteiz, Spain; jm.gil-garcia@ehu.eus
* Correspondence: isidro.calvo@ehu.eus

Abstract: Designing smart building IoT applications is a complex task. It requires efficiently integrating a broad number of heterogeneous, low-resource devices that adopt lightweight strategies. IoT frameworks, especially if they are standard-based, may help designers to scaffold the applications. OpenFog, established as IEEE 1934 standard, promotes the use of free open source (FOS) technologies and has been identified for use in smart buildings. However, smart building systems may present vulnerabilities, which can put their integrity at risk. Adopting state-of-the-art security mechanisms in this domain is critical but not trivial. It complicates the design and operation of the applications, increasing the cost of the deployed systems. In addition, difficulties may arise in finding qualified cybersecurity personnel. OpenFog identifies the security requirements of the applications, although it does not describe clearly how to implement them. This article presents a scalable architecture, based on the OpenFog reference architecture, to provide security by design in buildings of different sizes. It adopts FOS technologies over low-cost IoT devices. Moreover, it presents guidelines to help developers create secure applications, even if they are not security experts. It also proposes a selection of technologies in different layers to achieve the security dimensions defined in the X.805 ITU-T recommendation. A proof-of-concept Indoor Environment Quality (IEQ) system, based on low-cost smart nodes, was deployed in the Faculty of Engineering of Vitoria-Gasteiz to illustrate the implementation of the presented approach. The operation of the IEQ system was analyzed using software tools frequently used to find vulnerabilities in IoT applications. The use of state-of-the-art security mechanisms such as encryption, certificates, protocol selection and network partitioning/configuration in the OpenFog-based architecture improves smart building security.

Keywords: smart buildings; cybersecurity; OpenFog (IEEE1934); artificial intelligence of things (AIoT)

Citation: Martín Toral, I.; Calvo, I.; Villar, E.; Gil-García, J.M.; Barambones, O. Introducing Security Mechanisms in OpenFog-Compliant Smart Buildings. *Electronics* **2024**, *13*, 2900. https://doi.org/10.3390/electronics13152900

Academic Editors: Roberto Barchino, José Amelio Medina-Merodio, Salvador Otón Tortosa and José Javier Martínez-Herráiz

Received: 11 June 2024
Revised: 18 July 2024
Accepted: 19 July 2024
Published: 23 July 2024

1. Introduction

Smart building applications are expected to make a significant impact in buildings by providing advanced services that may leverage sustainable development goals [1]. As a matter of example, smart building applications carry out diverse tasks such as optimizing energy consumption [2,3], controlling indoor air quality (IAQ) [4–6] or delving into the information acquired [7,8]. Frequently, smart building applications combine the use of Artificial Intelligence (AI) and Internet of Things (IoT) technologies [9] and require the integration of a considerable amount of monitoring and actuation devices for automatic processing and control purposes [10].

However, the design and development of IoT-enabled smart building applications is a complex task [11], since they must satisfy substantial requirements in terms of comfort, energy consumption, usability and security [12]. Often, these applications require the analysis of big volumes of data, acquired from a large number of heterogeneous devices

located in a broad area [13]. The adoption of Edge/Fog/Cloud concepts promotes the organization of IoT devices in multi-tier computing infrastructures [14,15]. In particular, fog computing has been proposed in smart buildings to administer the massive, security-crucial, and delay-sensitive data, which are produced by IoT devices [16]. Thus, the use of fog computing allows the processing and storing of data locally, delivering high-quality services with short response times [15].

Designing smart building applications poses several challenges to integrating a broad number of heterogeneous, low-resource devices efficiently using lightweight strategies. Although vendor-specific solutions may be used in smart building applications, the adoption of Free and Open Source Software and Hardware technologies may increase flexibility and reduce deployment costs. In particular, the adoption of IoT distributed architectures may enhance service provisioning along the Cloud-to-Things continuum [17].

Unfortunately, increased connectivity and accessibility come with increased cyber-security threats, so smart buildings may become vulnerable to cyber-attacks that may cause discomfort, excessive energy usage, or unexpected equipment downtime [18]. In [19], the authors identify four key points in fog applications that can be highly vulnerable to cyberattacks, namely: (1) data storage, (2) data exchange, (3) data querying, and (4) data computation. In [20] a taxonomy of cyber-physical threats and their impact on smart homes is presented focusing not only on the attack vectors but also on the potential impact on the systems and the domestic life of the occupants. According to [21], the vulnerabilities of IoT devices may be grouped into four categories: confidentiality, integrity and authentication, access control and availability, and capability to reflect DDoS attacks. However, these vulnerabilities must be first identified. A pioneering study uses the Shodan search engine for detecting security issues in smart buildings [22].

Security issues in IoT applications require a myriad of security protocols, hardware and other components to work in harmony. This can only be achieved by outlining a holistic view of IoT systems [23]. In addition, IoT devices used in smart buildings typically have low resources, hence lightweight strategies that balance privacy and security must be adopted. Security-by-design is an approach that involves implementing security measures to protect IoT applications in the software and hardware development life cycle and not after detecting a security breach [24]. However, too often, fog applications are driven by the desire for functionality and end-user requirements, while the security aspects are ignored or considered as an afterthought [25]. As a result, the design issues for data security and privacy in IoT applications remain partially unsolved, making them easy targets for security attackers.

Introducing security mechanisms in smart building applications is critical, but it is not trivial. In fact, it complicates the design and operation of smart buildings, especially in the case of large buildings, which may handle huge amounts of data traffic. The introduction of security mechanisms increases the complexity of the applications, increasing the final cost of the deployed systems. It may even be the case that difficulties arise in finding qualified personnel to implement security mechanisms in this type of application. For this reason, the definition of a scalable architecture that may adapt to buildings of different sizes as well as the definition of several guidelines may assist application designers and implementers in the process of creating secure smart building applications.

In this scenario, the use of reference architectures and standards eases the integration and interoperability among devices in complex IoT applications. Some reference architectures available are RAMI 4.0, IMSA, IVRA, IIRA, and OpenFog [26]. In this work, OpenFog has been selected for several reasons: (1) OpenFog, established as IEEE 1934 standard [27], presents a simple and adaptable model that has been identified to be implemented in smart homes and buildings [16]; (2) this standard promotes the use and integration of open hardware and software technologies; (3) the OpenFog standard considers security as a key pillar and identifies the security requirements of the applications [28]. However, OpenFog defines a reference architecture that may combine a broad number of security mechanisms,

not being always obvious to select the best alternatives. In addition, it does not describe clearly how to implement these mechanisms.

The presented approach provides an implementation view for applications deployed in smart buildings. Thus, it eases the implementation of solutions that include key security aspects for small or medium-sized enterprises (SMEs), public properties, or domestic installations, which may find difficulties in implementing secure related technologies [29]. This approach integrates a broad number of state-of-the-art security mechanisms, such as the use of root-of-trust hardware, network authentication, certificates, encryption mechanisms, secure communication protocols, personal and corporate firewalls as well as adequate network configurations. Also, the use of a multilayer architecture, following the OpenFog reference architecture, allows adapting the presented approach to scale to buildings of different sizes, since it allows to define separate areas of operation, which do not interfere among them.

In summary, the contributions of this article are as follows:

1. It identifies the state-of-the-art security mechanisms that must be used to guarantee a certain degree of cybersecurity in smart building applications.
2. It presents a scalable architecture, based on the OpenFog reference architecture (IEEE 1934 standard), to provide security by design in smart building applications. The presented approach promotes the use of free and open-source software and hardware technologies and may be used to integrate low-cost IoT devices.
3. It presents guidelines to help developers and implementers create secure smart building applications, even if they are not experts in cybersecurity.
4. It illustrates how to apply the presented approach to develop smart building secure applications by means of a proof of concept implementation at the Faculty of Engineering of Vitoria-Gasteiz.

This article is structured as follows. Section 2 presents some background information, which analyzes the vulnerabilities and security requirements in smart buildings, presents the OpenFog security view and discusses some related works. Section 3 proposes an OpenFog-based secure architecture for smart buildings and presents some implementation guidelines. Section 4 illustrates the implementation of the proposed architecture, by means of a low-cost proof of concept deployed in the Faculty of Engineering of Vitoria-Gasteiz, following the guidelines presented in this work. Finally, Section 5 draws some conclusions.

2. Background

2.1. Vulnerabilities in Smart Buildings

The National Institute of Standards and Technology (NIST) defines vulnerabilities as weaknesses in the information systems that could be exploited by potential threats [30,31]. Typical smart building applications encompass a variety of components such as application servers, communication infrastructures, human–computer interfaces and even sensors and actuators. Each component may be affected by multiple vulnerabilities. Mostly, these vulnerabilities stem from applying security measures in an inadequate way or from using a poor network configuration with insufficient isolation. As a result, both sensitive information and the correct operation of systems can be left unprotected. This fact may impact the safety and privacy of the individuals. Particularly, processed and stored data should be protected in order to guarantee the privacy of the occupants. To prevent cyber-attacks, it is necessary to introduce countermeasures encompassing procedures and techniques employed to prevent, mitigate and eliminate potential attacks, thereby minimizing the harm they may cause [31]. These vulnerabilities need to be addressed by means of a correct methodology [30,31] that minimizes human error in the design of the security models [32].

In this scenario, standards such as OpenFog [27] and MUD [30] capture the nature of risks and threats according to the errors made and the points at which they can occur. In this way, they help to identify all possible factors to be considered. As identified in [23,33], these threats may address user identification, user tracking, profiling, utility monitoring

and network control. Table 1, provided by the IEEE 1934 standard [27], presents some common threats and attacks in complex IoT applications.

Table 1. Examples of threats and attacks [27].

Threat Categories		*Confidentiality Violation*	*Integrity Violation*	*Authentication Violation*	*Availability Violation*	*Privacy Violation*
Intents		Leaking information through overt/covert channels	Modifying data/code without proper authorization	Masquerading one entity as another entity	Rendering resources unreachable /unavailable	Leaking sensitive information of an entity (incl. identity)
Attack Venues	*Insider Attacks*	Data Leaks	Data Alteration	Identity/Password /Key Leaks	Equipment Sabotage	Data/Identity Leaks
	Hardware Attacks	Hardware Trojans, Side Channel Attacks	Hardware Trojans	Hardware Trojans	Radio Jamming, Bandwidth Exhaustion	Hardware Trojans, Side Channel Attacks
	Software Attacks	Malware	Malware	Malware	DoS/DDoS, Resource Depletion	Malware, Social Network Analyses
	Network Based Attacks	Eavesdropping	Message/ Transaction Replay	Spoofing, Man-in-Middle Attacks	DoS/DDoS, Subnet Flooding	Traffic Pattern Analyses

2.2. Security Requirements in Smart Buildings

There are several key aspects that need to be considered for ensuring security in IoT applications. For example, standards like IEEE 1934 [27], designed by the Industrial Communications Consortium (ICC), or MUD [30], designed by NIST, identify the requirements that need to be considered for developing smart buildings and home automation systems. These standards review the points in the software that might be vulnerable, which could pose a risk.

In addition to software security, it is crucial to highlight hardware security. In this standard [27], fog nodes are intended as the first barrier for encryption and access control to the devices in the edge layer. It is imperative to ensure that only authorized entities can access these devices. Although edge devices are quite heterogeneous, they should have a certain level of security, which includes securing their components and protecting their communications.

Smart buildings require several types and levels of security. In addition, each type of asset needs to be protected against different levels of threats specific to their intended use and location. Among these assets, the information technology infrastructures are especially critical. It is necessary to ensure the availability of the services and the information contained in these assets, focusing on sensitive and personal information.

In a smart building infrastructure, end-to-end security demands encompass everything in the Cloud-to-things continuum model. A security plan shall include measures to prioritize aspects where cybersecurity techniques should be implemented in the following categories

1. **Data Storage:** In smart buildings, data are stored on the nodes of the fog layer temporarily or indefinitely. These nodes must be especially protected against cyberattacks. Within this category, key protection should be taken into account. According to [23], it is recommended to include anti-tampering protections, as well as detection and recovery mechanisms.

2. **Device Interconnection:** The communication among fog and edge devices must be properly encrypted to protect communications against cyberattacks. This approach protects the information used in smart building applications from unauthorized access.
3. **Communication with the Cloud:** Information exchange with the cloud must be secure and reliable, avoiding information leakage. So, secure protocols that encrypt the information must be used. Also, institutional firewalls are necessary to filter, identify, and categorize each element of the data flow preventing access by unauthorized parties.
4. **Data Protection:** Confidential data of both users and entities connecting to the network must also be considered. Each element has its particular identification, and it is necessary to ensure that unauthorized entities do not access them or make a copy to try to access the architecture.

2.3. Openfog Security View

This subsection summarizes some security recommendations for Edge/Fog/Cloud applications according to the reference architecture proposed in the OpenFog standard [27]. OpenFog identifies certain security aspects that must be solved, which may be classified into four categories: cryptographic functions, node security aspects, network security aspects, and data security aspects.

- **Cryptographic functions:** Most security services involving confidentiality, integrity, authentication and non-repudiation require cryptographic functions to provide secure execution environments for trusted software or data integrity. Cryptography may be implemented in different ways, either in software or by means of hardware security processors, such as Platform Security Processor (PSP) or Trusted Platform Module (TPM). Symmetric or asymmetric ciphers may be used to implement cryptographic algorithms with different levels of security and computational cost.
- **Node security aspect:** Node integrity is a fundamental part of security as it makes up the bulk of the architecture's structure. The Trusted Computing Base (TCB) refers to the platform hardware, software and networking components that if violated would compromise the ability of the system to enforce its security policies. The root of trust is at the heart of the TCB and the security of the fog node. Security needs to be anchored in the hardware so that it cannot be circumvented. The Hardware Root of Trust (HW-RoT) is the key to a fog node's TCB. Fog nodes should support a HW-RoT to ensure that non-verified code is executed in the boot process. This is a key stage to confirm the node security. It is critical that fog nodes have a method to securely establish a root of trust during the boot process, ensuring that the firmware and the system software have not been tampered with.
- **Network security aspect:** Fog computing applications are deployed between the frontend devices and the cloud computing data centers. The ITU-X.805 Recommendation [34], identifies two operational planes in order to provide end-to-end security communications: Security Provisioning and Security Monitoring and Management, with three functional layers: Communications Security, Services Security and Applications Security. Security Monitoring and Management typically involves the configuration of several network security appliances such as Firewalls, Deep Packet Inspection (DPI), Intrusion Detection and Protection Systems (IPS/IDS), System/Network Events and State Monitoring. The communications occurring in the Device-Fog-Cloud computing continuum can be categorized into three kinds of communication pathways:
  - *Node-to-Device:* It covers the communication between edge devices and fog nodes. Among many Internet resource-constrained IoT devices, limited cryptographic capability is available. Techniques such as symmetric ciphers using manually installed keys are commonly used in these kinds of devices. These devices must be installed in physically protected environments and connected via hardware connections to one or more fog nodes that can provide most of the X.800 communication security services. The use of wireless technologies is common in this

communication pathway, in particular, MQTT over WiFi and TLS for publish-subscribe messaging and 802.1X for authentication purposes.
  - *Node-to-Node:* It covers the communication among several fog nodes. Fog nodes coordinate between them in order to accomplish specific objectives. This layer combines the client-server computing model, typically over HTTPS, and the publish-subscribe messaging patterns, with MQTT being one of the most commonly used protocols. Strong authentication and non-repudiation services shall be implemented using security credentials derived from the hardware root-of-trust installed in the fog node.
  - *Node-to-Cloud:* It covers the communication between fog nodes and cloud services. Strong authentication and non-repudiation services shall be implemented using security credentials derived from the hardware root-of-trust installed in the fog node. Almost all these communications are currently conducted as web service transactions over TCP/IP protocols.
- **Data security aspect:** Encryption algorithms are used to enforce most data security aspects. OpenFog classifies data in three categories: data in use, during data processing; data at rest, when resides on non-volatile storage devices, such as hard disks or SSDs; data in motion when they are sent or received on network interfaces as packets. Memory encryption technologies could be implemented for data in use. The encryption of data at rest limits the access to the data to those with the correct keys. Three encryption methods are generally used for data at rest: full disk encryption; file system encryption and file system access control mechanisms. The use of technologies, such as TLS or VPNs, allows encrypting data in motion.

2.4. Related Work

Providing privacy, trust and security is a critical need in smart buildings since a security breach could have consequences in the services that may affect occupant safety, comfort, privacy and economic aspects [33]. Unfortunately, smart building applications are not always adequately protected, so some of the threats may pose a significant danger to building operations. Actually, ensuring privacy and security is one major research challenge, since IoT applications generate a huge amount of data that must be treated in a private and secure way [35]. Threat attacks exploit weaknesses in individual devices to access their data.

Security concerns have grown over the last decade due to the increase in cyber-attacks on building automation systems (BAS). The work in [18] presents a critical review of cyber-physical security for building automation systems, focusing on the four BAS dominant vendor-specific protocols, i.e., BACnet, KNX, LonWorks and Modbus. This review reports exhaustively several cases of the most common cyber-attacks identified. This work also mentions some typical threats that affect smart buildings more often. Different attack types are used to exploit known vulnerabilities in the systems. For example, Denial-of-Service (DoS) describes a type of attack that causes the services or resources from a system to be inaccessible to their legitimate users [36]. A fuzzing attack is a software technique consisting of sending many invalid and randomly generated inputs to a system [37]. In Man-In-The-Middle (MITM) attacks [38] a perpetrator positions himself in a conversation between a user and an application. Password Brute-Force algorithms generate all possible combinations of characters in a specified range and length, while the dictionary attack checks against a predefined word list [39]. A replay attack is one typical threat that does not require any system information, where the attacker, e.g., records a sequence of sensor messages and replays the sequence afterward [40]. Sniffing is a type of attack in which its process involves capturing, decoding, inspecting, and interpreting the data from the packets transmitted over the transmission channel, e.g., the TCP/IP network [41]. Spoofing is also a common attack on smart home systems where the attacker manipulates fake identities by forging a large number of identities to act as legitimate nodes, thereby compromising the effectiveness of IoT devices [42].

Table 2 summarizes the diverse approaches and methodologies available in the literature to cope with common threats. In some of them, authors prove the effectiveness of these practices by testing or simulating them.

Table 2. Diverse attacks that affect the security of smart buildings and approaches to cope with them.

Type of Attack	Proposed Solutions	Description
DoS/DDoS	[42–48]	Anomaly detection when the server is saturated by excessive data traffic preventing the use of online services and sites.
Tampering	[49–52]	Secure hardware systems against physical tampering and unauthorised access.
Fuzzing attack	[53,54]	System alerts when the data are being exploited by the attack or fuzz models for testing systems to seek vulnerabilities.
Replay attack	[42,45,46,55–57]	Protect the authenticity and timeliness of communications to secure the network between devices.
MITM	[42,57–61]	Ensure the confidentiality of information transferred between two entities.
Password Brute-Force attack	[57,62–64]	Introduce good user and password policies.
Sniffing	[65–68]	Encryption from malicious activities where an attacker intercepts and captures network traffic to gain unauthorized access to sensitive information.
IP Spoofing	[42,52,57]	Protect against false identities of entities impersonating users or devices to avoid compromising the IoT architecture.

In this scenario, it is possible to find several IoT architectures for smart buildings. Some of them are based on reference architectures. For example, ref. [6] presents an architecture for smart buildings based on the OpenFog reference architecture which uses fuzzy logic for controlling the IEQ conditions of a building. However, this architecture does not implement security mechanisms. Other works propose solutions that include security issues in smart buildings, but which are not based on standards or reference architectures. For example, ref. [69] presents a proprietary architecture aimed at improving energy-efficiency in smart buildings, which considers security aspects. In [70] a component-based architecture is proposed, aiming at managing residential environments, which implements fog computing concepts. Ref. [71] presents a distributed fog computing Home Automation System (HAS) that claims to provide seamless communications among ZigBee and WiFi devices. In [72] it is presented a framework to protect smart homes by using VLAN-based network isolation. Ref. [73] surveys research works conducted on the integration of Software Defined Networking (SDN) in smart buildings. Outside the smart buildings domain there are also some interesting works. For example, ref. [74] presents a network security model involving firewalls, routers, Virtual Local Area Networks and one AAA (Authentication, Authorization and Accounting) server. In [75], a novel cybersecurity technology, so-called HoneyNet, is implemented. Their approach uses a decoy for hackers,

creating a deliberate sense of vulnerability. Once an attack occurs, the system analyzes how the infiltration happened and learns from it to strengthen itself against future cyberattacks.

Previous examples highlight the necessity for introducing robust security measures. Although it is not possible to guarantee total protection against attacks it is necessary to maintain a certain level of security in order to mitigate risks as much as possible. However, this is a difficult task given the constantly evolving nature of the field and the challenge of introducing security measures in complex IoT applications. In this work, the authors present a scalable architecture, based on the OpenFog reference architecture, IEEE 1934 standard [27], aimed at providing security by design in smart building applications. The OpenFog standard promotes the use of open-source software and hardware technologies and may be used to integrate low-cost/low-resource IoT devices. Also, the authors try to help designers and implementers create secure applications, even if they are not experts in cybersecurity matters. As a result, the presented approach eases the design of secure complex smart building applications.

3. Secure Smart Buildings Based on the OpenFog Standard

This section particularizes the OpenFog security view to propose an OpenFog-compliant secure architecture for smart buildings. The presented architecture, which evolved from [6], integrates low-cost devices by means of free and open-source software technologies. It tries to avoid the vulnerabilities shown in Table 1 by means of a security-by-design approach. This architecture aims at helping non-cybersecurity experts to design and implement relatively secure smart buildings. More specifically, the current section (1) presents an OpenFog-compliant secure architecture for smart buildings; (2) discusses basic security mechanisms used at smart buildings and (3) provides implementation guidelines and recommendations.

3.1. Openfog-Based Architecture for Secure Smart Buildings

This subsection presents an OpenFog-compliant secure architecture aimed at smart buildings. This multilayer architecture is based on the following design principles:

- *Anti-tampering mechanisms*. All nodes must provide mechanisms to ensure that the firmware and system software have not been tampered with. Also, physical protection of the nodes must be guaranteed.
- *Common and low-cost technologies*. The architecture employs widely used technologies. Particularly, Free and Open Source Software (FOSS) technologies and open hardware devices are preferred.
- *Zero-trust security*. In order to reduce the risk of threats in the system, connectivity among nodes is restricted to the essential services. In addition, a hierarchical structure among nodes is proposed.
- *Security Monitoring and Management*. Several security appliances are used to detect attacks, such as Firewalls, Deep Packet Inspection (DPI), Intrusion Detection and Protection Systems (IPS/IDS) as well as System/Network Event and State Monitoring Systems.
- *Critical data protection*. This applies to both data at rest, i.e., stored in all nodes of the hierarchy, and data in motion, i.e., data transmitted by means of networking technologies.

3.1.1. Architecture Layers

Figure 1, overviews the multilayer architecture proposed with the main node types found in each layer:

- **Edge layer:** This layer includes sensor and actuator devices involved in the operation of the buildings. Typically, these devices (1) evaluate safety and comfort-related parameters, such as building occupancy, indoor air quality, or fire detection systems, (2) are responsible for security operations, such as security cameras or motion sensors, and (3) execute local operations by means of actuators, such as intelligent climate control systems or automated lighting systems. These resource-constrained IoT devices are the most exposed ones in the architecture, so in order to be included in smart

building applications several security mechanisms must be introduced. They need anti-tampering hardware systems to ensure software protection.

- **Fog layer:** This layer plays a key role in ensuring scalability and reliability. It is divided into two sub-layers of nodes: Zone nodes and supervisory nodes. Zone nodes, at the bottom of this layer, are responsible for specific areas in the building. These nodes process and store data locally and execute control operations in a restricted area. For security reasons, they are not visible from the Internet, their communication is restricted to connecting with a set of edge devices and the supervisory nodes. Typically, zone nodes are implemented over low-cost SBC platforms such as Raspberry Pi, or virtualized as containers in order to be centralized. Supervisory fog nodes are at the upper fog layer. These nodes are responsible for coordinating the operation of the whole building by supervising the operation of the zone nodes. Also, they act as proxies with cloud services for different operations.
- **Cloud layer:** This layer allows the use of certain advanced cloud services in smart building applications. These services may be related to different operations, such as weather forecasts, massive data storage, or data analysis. In order to avoid security flaws, only supervisory nodes are allowed to use cloud services. Since these nodes are exposed to the external Internet, they must be carefully configured to avoid or minimize the effect of external attacks.

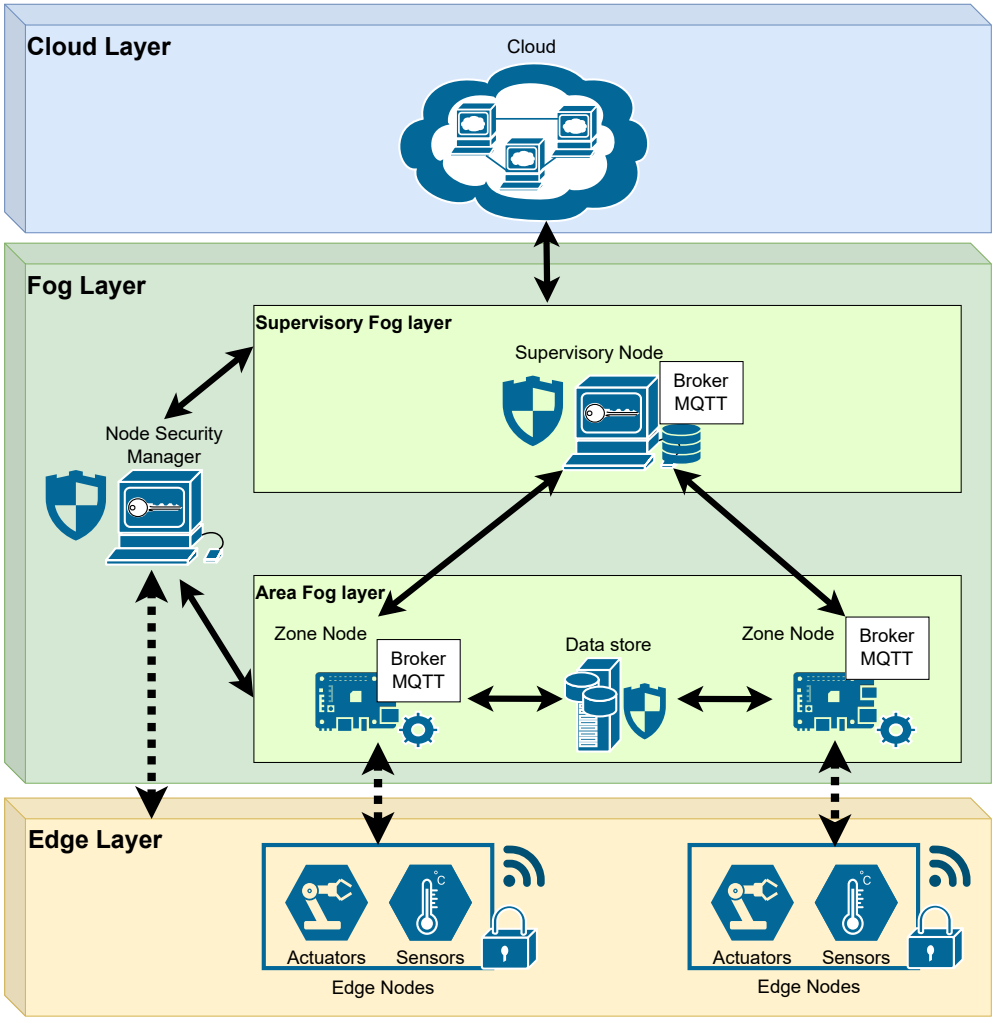

Figure 1. OpenFog-based architecture for smart buildings.

The architecture shown in Figure 1 is highly hierarchical, which increments scalability. Nodes in the Edge layer, i.e., sensors and actuators, are connected by means of a separate WiFi VLAN, implemented by a large number of WiFi access points that may cover buildings of very big sizes, such as University Faculties. In addition, the nature of smart building applications requires sampling times in the range of several minutes and small-size data. Thus, the number of nodes in the architecture could be easily scaled up without affecting the performance of the application. Edge nodes are not allowed to connect among them, but they are connected to a set of zone nodes depending on their location in the building. Zone nodes are connected by means of wired networks, which improves the performance and reduces the amount of WiFi traffic.

3.1.2. Security Features

The proposed approach combines a set of mechanisms and procedures aimed at providing confidentiality, integrity and availability, the so-called CIA triad. For instance, the combination of encryption, permission management and strong password usage helps to ensure data confidentiality. Access control, intrusion detection and prevention systems safeguard information integrity, alongside network and system event monitoring [76]. Also, keeping systems and security up to date, and using an adequate network security model, involving topology, network segmentation, using Virtual Local Area Networks (VLANs) and security appliances (firewalls and IPS/IDS) thwarts attacks and ensures information availability [74]. By integrating additional services, such as memory encryption or backups, the system gains supplementary layers of protection and operational resilience, contributing to a more secure and reliable infrastructure.

The OpenFog standard recommends introducing diverse security features to enhance system robustness [27]. These security features are summarized below, also there is a short discussion on how they are implemented in the proposed architecture:

- **Hardware Root of Trust:** The HW-RoT is a hardware component that establishes a secure and trusted point of initiation for the chain of trust in a system. An example of such a component would be a security chip, such as a Trusted Platform Module (TPM), which stores cryptographic keys and other security credentials. This chip ensures that only trusted software and hardware are loaded during boot-up, thus creating a secure environment for the node. In the OpenFog context, this security chip is typically developed as part of a SoC (System on Chip), integrating various security functions directly into the node hardware.
- **Secure Communications:** Secure communications among the nodes in the architecture may be achieved by means of encryption services and identity brokering mechanisms. The cryptographic functions must ensure confidentiality (with private keys), integrity, authentication, and key generation (with public keys), while also establishing long-term credentials (with certificates). Only communication technologies involving security credentials and message encryption are considered able to provide confidentiality, integrity and availability.

  The MQTT protocol over TLS is suitable for Node-to-Device and Node-to-Node communications in big smart buildings. This protocol allows client devices to receive updates without constantly requesting them, reducing resource consumption and making MQTT ideal for high-latency networks [77]. Despite the limitations of this protocol, it is becoming widely adopted for resource-constrained IoT devices, since it is fast, simple, reliable, highly scalable and with a low implementation cost. The defense mechanisms are discussed in detail in [78]. Several MQTT implementations regarding security issues are analyzed in [79].

  The node-to-cloud communication, i.e., between the supervisory fog nodes and the cloud, is provided by means of the HTTPS protocol. This protocol, widely accepted on the Internet, uses encryption for secure communication over TCP/IP networks. HTTPS encrypts the messages by means of the TLS protocol. Thus, HTTPS provides

authentication of the accessed website and protection of the privacy and integrity of the transmitted data.

- **Network configuration:** Figure 2 shows the network structure of the architecture. In order to reduce the risk of threats in the system, connectivity among nodes will be restricted to the essential services (zero-trust), organized in a hierarchical way. The architecture may benefit from the use of existing communication infrastructures already available in the buildings.
  Edge nodes use corporate WiFi networks to establish Node-to-Device communication with the zone nodes, in the fog layer. This proven technology is a flexible alternative to connect devices in most public and private buildings since it is typically deployed to reach every corner of the buildings by means of the existing infrastructure. The use of a specific VLAN helps isolate different network segments and provides an additional layer of security by restricting communication to authorized devices within this VLAN. For security reasons, the horizontal communication among edge devices is disabled, so that only zone nodes are allowed to connect to a group of edge devices. In addition, all nodes, especially the nodes in the fog layer, must be protected by firewalls.
  The Node-to-Cloud communication is enforced by the supervisory fog nodes. These are the unique nodes allowed to connect to cloud services, not accessible from the Internet neither the zone nodes nor the edge nodes. Configuration efforts must be focused on the supervisory fog nodes.

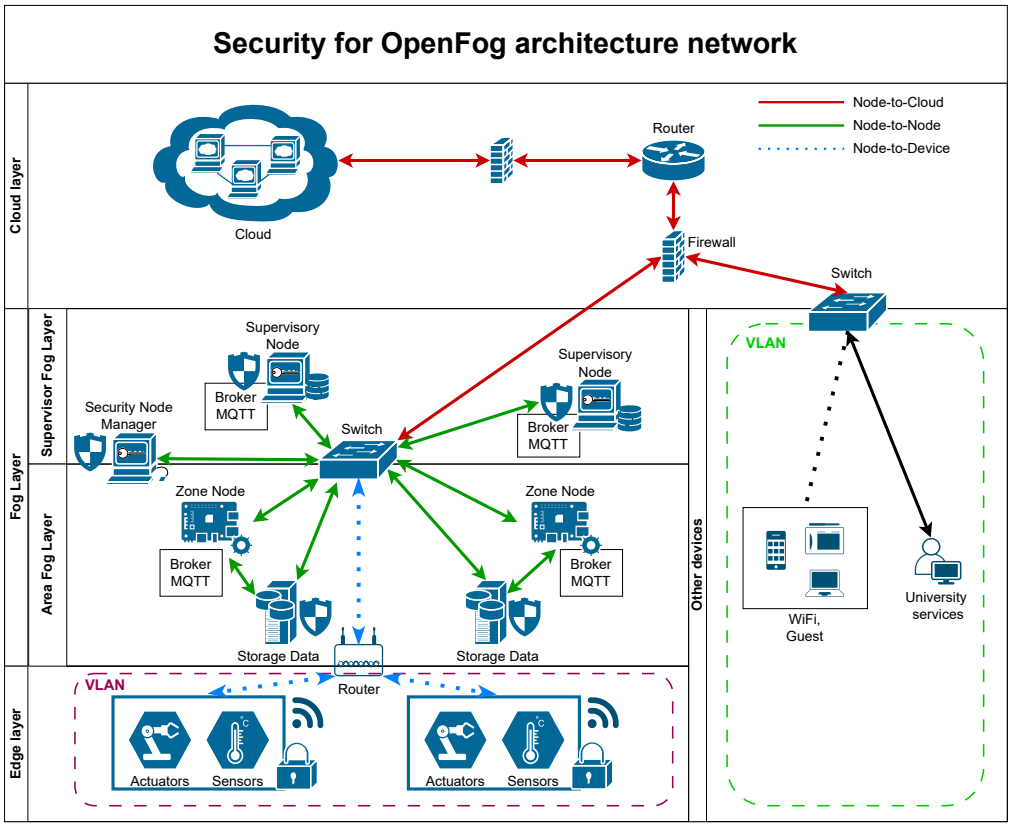

Figure 2. OpenFog network view for smart buildings.

- **Node Management and Network Management:** OpenFog introduces the concept of Out-of-Band (OOB) management, which refers to remote control of nodes from a centralized point, allowing the management of software, resources, security and health monitoring of the nodes. This approach provides an additional layer of control, as it enables administrators to manage multiple nodes from a single control node. The Node Security Manager centralizes the tasks related to the management of the

keys and certificates of the nodes. In some cases, existing corporate infrastructures may be used for network security monitoring and management tasks. For example, Directory Services Management, Firewalls, Deep Packet Inspection (DPI), Intrusion Detection and Protection Systems (IPS/IDS) as well as System/Network Event and State Monitoring Systems may be used to manage the connectivity among the nodes of the architecture.

- **Data storage:** Protecting sensitive data, including keys, credentials, passwords and user/device identifiers is another essential aspect of the architecture. This information, usually stored in files, can be encrypted using different mechanisms, such as the File Encryption Key (FEK) or the Encrypting File System (EFS). The configuration of file system permissions in OS-based platforms may ensure that these files are accessible only to authorized users. Finally, an adequate backup policy is essential to avoid accidental data losses, especially in the fog layer.
- **Containerization (Recommended):** The use of containerization may improve the security of the critical nodes in the fog layer since containers increase the system's resilience to attacks. OpenFog recommends to prioritize the use of containers over virtual machines (VMs). It uses a lightweight executable that is more versatile and secure than a VM due to its isolated Operating System (OS). Unlike VMs, which share resources, this approach has its own libraries, creating a self-contained environment. This isolation reduces cross-contamination risk and limits the impact of security breaches, offering additional protection. Thus, fog nodes could be executed in containers in order to provide higher security and flexibility.

3.1.3. Security Dimensions

This subsection overviews the security dimensions identified in [34] and discusses about how they are achieved in the scope of this work. Security dimensions are provided in different ways in the Cloud-to-Thing continuum model. Table 3 summarizes the security mechanisms used across the OpenFog-based proposed architecture for smart buildings. Figure 3 locates the use of some relevant mechanisms across all layers of the architecture.

1. **Access control:** Resource access is only granted to authorized users and devices. These resources include network elements, stored information, information flows, or services and applications.
   The recommended way of providing access control is by means of the IEEE 802.1X standard [80]. This approach, in combination with a centralized authentication, authorization and accounting (AAA) management service, like Radius, provides an authentication mechanism for connecting to a LAN or WLAN. Directory services, such as LDAP, also play an important role in IP applications to share information about users, systems, networks, services and applications. The use of the 802.1X standard may not be available in some edge platforms. Although it is not recommended, if no other means are available, the combined use of pre-shared keys and MAC addresses may provide a minimum level of access control security for node-to-device connectivity.
2. **Authentication:** It allows confirmation of the validity of the identities of the communication participants (i.e., users, devices and applications). It avoids masquerading or unauthorized attempts in communication.
   All devices must provide a HW-RoT in order to establish a secure and trusted point of initiation of the chain of trust in a system, ensuring a secure boot. The use of credentials, e.g., X509 certificates, is also needed for providing authentication at the nodes. Since edge nodes are not connected to the Internet, a local certificate authority (CA) may be established locally, in the institution, to create the certificates installed in the edge nodes.

3. **Non-repudiation:** It prevents participants from denying the execution of a particular action. It supplies evidence to prove that every event or action has been carried out. The use of network and device logs for networking operations and local node tasks, respectively, may assist in achieving non-repudiation features. Critical operations should be signed with the certificates installed in the nodes.
4. **Data confidentiality:** Confidentiality requires that data must only be understood by authorized participants. It involves both data at rest and data in motion.
   Diverse mechanisms are used, such as encryption, access control lists and file permissions. In the case of nodes with an OS, confidentiality may be enforced by encrypting disk information or by arranging file system configuration permissions. Confidentiality of data in motion is enforced by means of secure protocols over TLS, namely MQTT and HTTPS.
5. **Communication security:** The information flow must be guaranteed between authorized end points so that the information is not diverted or intercepted in transit.
   Communication security is achieved by means of a secure stack of protocols over TCP/IP. In the case of node-to-device communications, MQTT over TLS is used as an application protocol. These data are transmitted over a WiFi VLAN, with access credentials serving as the first level of security, Node-to-Node communication is also enforced by means of MQTT over TLS, transmitted over wired networks, mostly Ethernet. Finally, Node-to-Cloud communication is provided by the corporate TCP/IP infrastructure involving diverse security appliances, such as corporate firewalls as well as other devices like IDS/IPS.
6. **Data integrity:** Data must be correct and accurate. They must be protected against unauthorized modification, deletion, creation and replication.
   The use of encrypted protocols based on the use of certificates may guarantee the integrity of the data in motion. Additionally, critical data should be signed in order to ensure integrity. Access to the devices must be protected through hardware measures, such as flash encryption or physical access control. Antimalware packages and End Point Protection and Response (EDR) systems may contribute to protecting the integrity of the data. Finally, a backup policy should be established to avoid accidental data loss, especially in the fog layer.
7. **Availability:** There must not be a denial of authorized access to the application services and resources. Disaster recovery solutions must be also included.
   Several mechanisms may be used to ensure edge node availability. The combination of a secure network configuration, involving network partitioning and connectivity restrictions, increases availability. If possible, nodes should have personal firewalls in order to restrict the connectivity to a certain number of ports and addresses. Device firewalls may complement corporate firewalls to achieve higher availability. The use of containerization may improve system availability. Finally, network monitoring and management appliances, such as IDS/IPS systems, are necessary to monitor the network and detect attacks early.
8. **Privacy:** Protecting the information that could be derived from observing the network activity.
   The combination of previous measures guarantees privacy, In particular, secure end-to-end communications and data protection measures contribute to the privacy of the data in the applications. Also, a secure network configuration must be established.

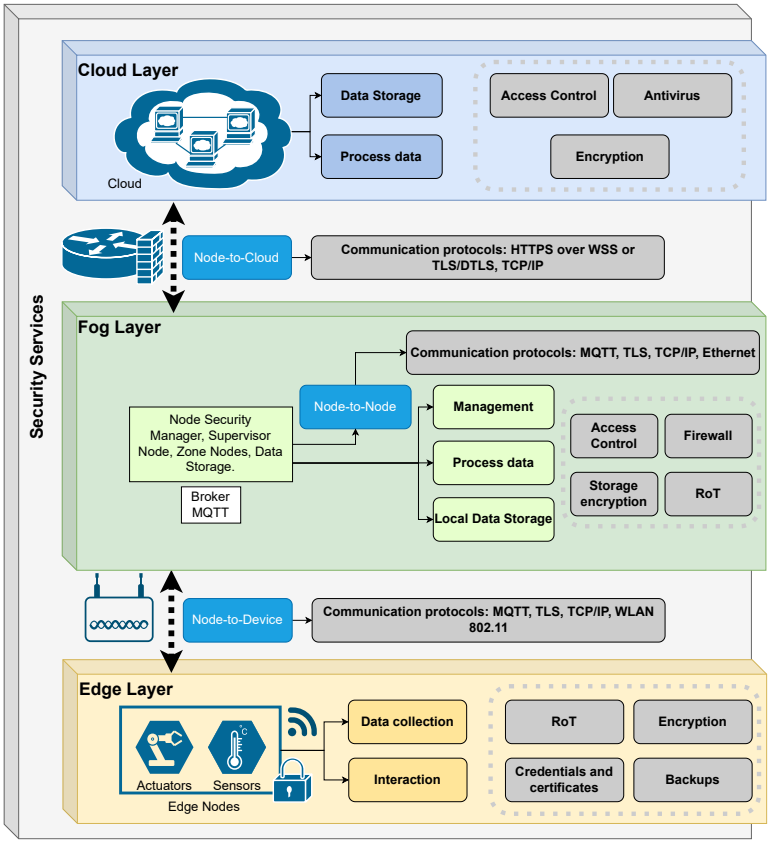

Figure 3. Technologies and services across the layers of the architecture.

Table 3. Mechanisms used in the proposed OpenFog-based architecture for smart buildings to achieve the security dimensions of the X.805 recommendation [34].

X.805 Security Dimensions	Node-to-Cloud Security Functions	Node-to-Node Security Functions	Node-to-Device Security Functions
Access control	IEEE 802.1X, Centralized AAA, Directory services (LDAP)	IEEE 802.1X, Centralized AAA, Directory services (LDAP)	IEEE 802.1X, Centralized AAA, Directory services (LDAP), PSK+MAC
Authentication	HW-RoT (TPM), X509 certificates, Credentials	HW-RoT (TPM), X509 certificates, Credentials	HW-RoT, X509 certificates, Credentials
Non-repudiation	Network and device logs. Signed data and operations	Network and device logs. Signed data and operations	Network and device logs (if available). Signed data and operations
Data confidentiality	Encryption, Access control list and file permissions. HTTPS	Encryption, Access control list and file permissions. MQTT over TLS	Encryption, Access control list and file permissions. MQTT over TLS
Communication security	HTTPS, corporate and personal firewalls, IDS/IPS systems	MQTT over TLS, corporate and personal firewalls, IDS/IPS	MQTT over TLS, WiFi VLAN, corporate and personal (if aviable) firewalls
Data integrity	File encryption systems, Backups, Signed data, Antimalware, EDR systems.	File encryption systems, Backups, Signed data, Antimalware, EDR systems.	File encryption systems, Backups, Signed data, Antimalware, EDR systems. (If available)
Availability	Secure network configuration (zero-trust), network partitioning, corporate and personal firewalls, IDS/IPS systems. Containerization	Secure network configuration (zero-trust), network partitioning, corporate and personal firewalls, IDS/IPS systems, Containerization	Secure network configuration (zero-trust), network partitioning, VLANs, corporate and personal firewalls,
Privacy	Secure end-to-end communication technologies, secure network configuration	Secure end-to-end communication technologies, secure network configuration	Secure end-to-end communication technologies, secure network configuration

3.2. Ensuring Security: Implementation Guidelines

This subsection presents some guidelines to help designers and programmers deploy the OpenFog-based architecture shown in Figure 1. Some of these guidelines involve network configuration. The steps that should be followed are detailed below:

1. Ensure that all devices in the architecture satisfy the OpenFog hardware and software requirements. Besides HW-RoT, nodes should support symmetric key ciphers (AES with at least 128-bit keys or Triple-DES), asymmetric key ciphers (DH, RSA, DSA, ECDH, ECDSA or ECQV), cryptographic hash functions [81], true random number generators [82]) and message authentication codes (CCM, GCM, GMAC, CMAC or HMAC).
2. Create a structure of certificates. This involves creating the certificates through a trusted Certificate Authority (CA) and installing them on the respective devices. Since the architecture design assumes that edge nodes are not available from the internet, the certificates installed on these nodes could be created by a local CA. Also, the configuration of the MQTT broker must include a user key and password for all Edge and Fog nodes in order to establish Node-to-device and node-to-node communication.
3. Encrypt the firmware in Edge devices to secure the keys used for accessing networking services, e.g., MQTT or WiFi, as well as other sensitive data of the application.
4. Define a separate WiFi VLAN to connect all edge layer nodes. This approach provides WiFi connectivity to all edge nodes in a more restrictive way. This approach achieves two major benefits: reducing interference with non-smart building applications and limiting the possibility of attacks. In addition, horizontal device-to-device communications in the Edge layer should not be allowed. Node-to-device communication, between the Edge and Fog layers, is limited to a few zone nodes, preferably connected to a wired network.
5. Create a hierarchy of fog nodes that can be scaled to cover buildings of different sizes. Two types of fog nodes are used (see Figure 1): zone nodes and supervisory nodes. Zone nodes must run an MQTT broker to provide connectivity to a set of Edge nodes. Supervisory nodes must also run an MQTT broker to collect data from zone nodes. They need to be able to identify each other, which is achieved using the MQTT access configuration and associated access credentials.
6. Robust security requires running accredited anti-virus software to protect the nodes against potential malware threats. Corporate and personal firewalls should be configured to grant access to a limited number of trusted devices and ports. IPS/IDS should detect and protect against suspicious activities.
7. To secure communication between the cloud and fog/edge nodes, HTTPS is needed to encrypt data and apply strong authentication using certificates and multi-factor authentication.

3.3. Prevention of Cyber-Attacks with the Security Model

Section 3 presents an architecture that implements security-by-design in smart building IoT applications, as well as guidelines to help designers of the applications create new applications. This subsection discusses how the presented approach copes with common threats that affect the security of smart buildings (see Table 2).

The adoption of a zero-trust approach mitigates DoS/DDoS attacks since the network is partitioned by means of specific VLANs. Also, the implementation of restrictive firewall rules, at both personal firewall configuration and corporate level avoids this type of attack. The approach limits the connectivity from only a few authorized devices by a reduced number of TCP/IP ports. In addition, edge nodes are connected to a separate intranet not connected to the Internet.

Critical nodes, especially those located in the fog layer, i.e., zone and supervisory nodes, must be located in protected areas to reduce the risk of tampering attacks. The use of a HW-RoT also avoids tampering attacks. Sensor and actuator devices, at the edge layer, should be protected by means of security chips to avoid, or reduce, tampering attacks. Also, critical information, such as passwords, is protected by means of diverse mechanisms, including file encryption systems or signed data.

The use of a restrictive network configuration, which involves network partitioning, restricts allowed connections achieving a reduction in the number of fuzzing attacks. Thus, the number of connected devices and the number of available connection ports is limited to those strictly necessary, following a zero-trust approach. Also, the adoption of network access control mechanisms, such as IEEE 802.1X or centralized AAA directory services avoids access by non-authorized devices.

The use of both secure communication protocols, which involve encryption and certificates, and network access mechanisms protects the authenticity to secure the network between devices. This approach mitigates different replay attacks. Certificates are another measure to thwart these attacks, as they validate the authentication of clients connecting to the network.

The use of secure protocols, such as TLS, which involve the use of certificates and encryption mechanisms, guarantees an end-to-end confidential channel to transfer information between two entities. The use of these kinds of technologies reduces MITM attacks. Edge nodes employ a VLAN and a secure wireless access point to repel threats such as MITM. This configuration makes it challenging for an attacker to gain access to information. The university's corporate network security systems prevent the same attack at higher layers, such as Node-to-Cloud communications.

Preventing password brute force attacks requires the use of a robust user and password policy, along with restricting access to resources at different levels of the system. This approach effectively thwarts potential attackers from gaining unauthorized access to system data through password theft.

Sniffing attacks may be prevented by means of encrypted communication protocols, such as TLS, and the use of personal and corporate firewalls, which monitor and control inbound and outbound network traffic. However, it is recommended the use of a robust user and password policy.

End users cannot detect IP spoofing attacks. However, this kind of attacks may be detected by means of network monitoring, finding atypical activity patterns or inconsistences, e.g., outgoing packets with non-matching IP addresses The use of corporate firewalls and IDS/IPS systems is key to avoid or mitigate this kind of attack. Certificates are another measure to thwart IP spoofing attacks, as they validate the authentication of clients connecting to the network.

4. Case Study

This section presents a proof-of-concept application aimed at illustrating the implementation of the presented OpenFog-related security aspects. The case study focuses on a low-cost Indoor Environmental Quality (IEQ) system that extends the work presented in [6]. This IEQ system is aimed at balancing the consumption of energy resources and the comfort for the occupants in some areas of the Faculty of Engineering of Vitoria-Gasteiz. However, this previous work did not address security aspects in its deployment, since they were considered as an afterthought. Although security by design is a desired feature, this is a common situation in this kind of application.

As discussed in [6], the architecture of the system is scalable. However, this proof-of-concept involves a simplified IEQ system. Namely, it involves two nodes in the Fog layer, one supervisory node and one zone node, and six smart nodes, of two different types, in the Edge layer. It follows a short description of the nodes:

- **Fog Layer:** The IEQ fog layer comprises the supervisory and zone nodes. The supervisory node is responsible for centralizing the interaction with cloud services and

coordinating the behavior of several zone nodes at the building level. The zone node is responsible for registering and processing the data in specific areas of the building, but it lacks direct Internet connectivity. This node executes an algorithm, based on Fuzzy logic, which is responsible for collecting the values of the target parameters in local areas of the building and making a decision accordingly. This decision involves controlling the actuator, in this case, opening or closing the window for some time.

- **Edge Layer:** Edge nodes (smart sensors and actuators) are scattered in selected areas of the building. They are responsible for measuring the environmental parameters and enforcing the response of the actuators. Prototypes of the smart nodes have been built over an ad hoc Printed Circuit Board (PCB) that integrates the selected sensors. In the proof-of-concept application, there are two node categories:
  - **Smart nodes Group A:** They measure IEQ parameters, namely temperature, humidity, and light intensity. Nodes N2, N3, N4 and N5 belong to this group in the proof-of-concept application.
  - **Smart nodes Group B:** These nodes measure environmental parameters, such as the amount of CO_2, and control the behavior of the actuators. Nodes N0 and N1 belong to this group, but only N0 is responsible for controlling the actuator that opens/closes the window.

Briefly, the IEQ proof-of-concept system operates as follows: The zone node collects the environmental parameters from all sensor nodes, such as temperature, humidity, CO_2 concentration, and light intensity every five minutes. In addition, it receives the current status of the window (closed or open) from node N0. The zone node also receives the external environmental conditions, i.e., outside temperature and humidity, which are proxied by the supervisory node. This node requests the external weather data from OpenWeatherMap [83], by means of an API. With all this information, the zone node executes a fuzzy logic algorithm [6], which determines whether the window should be open or closed. Figure 4 shows the implementation diagram applied to the case of the study.

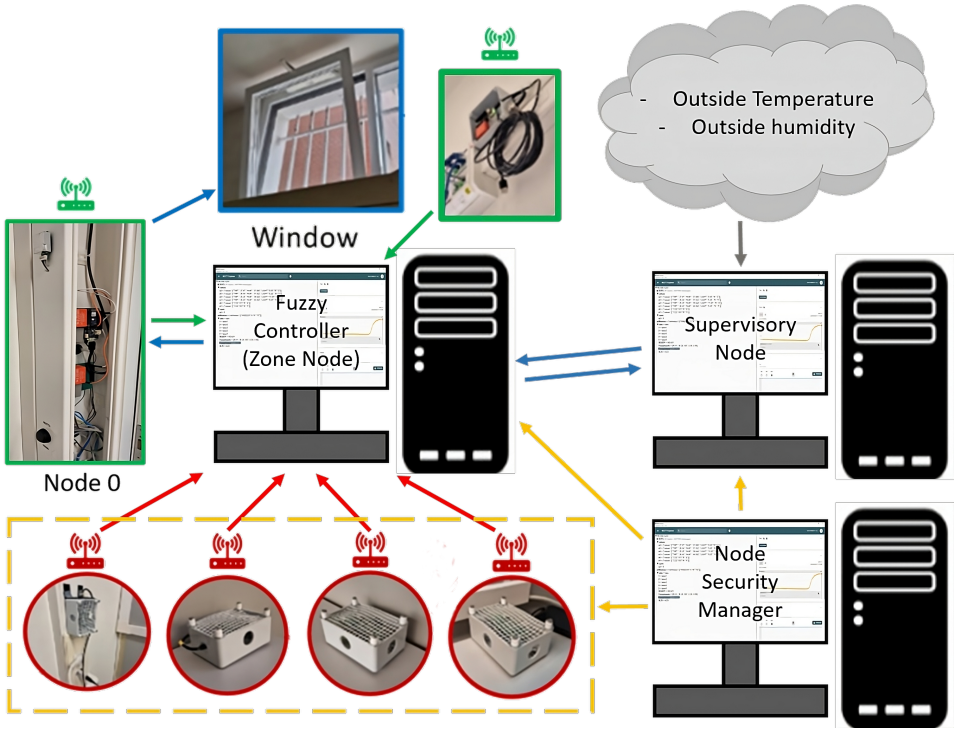

Figure 4. Overview of the proof-of-concept IEQ system. Nodes in red belong to Group A and nodes in green belong to Group B.

The IEQ application uses MQTT over TLS for implementing Node-to-Device communications. Namely, it is used to collect data from the sensor nodes and send the control values to the actuator nodes. In this case, the application uses the corporate networking infrastructure already deployed at the Faculty. WiFi communication covers the whole building. The definition of a WiFi VLAN segregates the traffic of the smart building application from other applications. In addition, connectivity is restricted to authorized devices only, incrementing the network security. Node-to-Node communications, i.e., among supervisory and zone nodes, is provided by means of MQTT over TLS over an already existing wired networking infrastructure. Node-to-Cloud communication, i.e., between the supervisory node and cloud services, is provided by means of the HTTPS protocol. The communication in all layers, but especially with the cloud layer, is restricted by means of corporate firewalls and IDS/IPS systems.

4.1. Components and Materials Used in the Architecture

It has been necessary to adapt the hardware devices of the experimental proof-of-concept presented in [6] in order to satisfy the security requirements of the OpenFog framework. It follows a short description of the employed nodes in each layer:

- **Edge Layer:** Edge IoT devices are the most exposed since they are scattered throughout a wide area. Following OpenFog's recommendations, physical protection has been designed to encapsulate each node's hardware. This protection safeguards the hardware from physical tampering. In the deployed proof-of-concept, ESP32-S2-Saola-1V1.2 boards [84] were used. They offer significant advantages in terms of cost and availability. This board meets the requirements of the OpenFog standard, allowing features like flash encryption and HW-RoT. Additionally, the board's libraries enable the implementation of security measures, such as secure network connections with usernames and passwords, as well as the use of certificates.
- **Fog Layer:** This layer includes three fog nodes: zone node, supervisory node and Node Security Manager (NMS). Desktop computers running Windows 10 were used to implement all nodes. The zone node hosts the MQTT broker and executes environmental control algorithms. The NSM provides additional security services, such as the creation of certificates and key creation, using OpenSSL as well as device and communication monitoring services. In this proof-of-concept application, the OOB network monitoring and diagnostic tasks were carried out directly by the networking staff, since they monitor all the traffic at the University.

4.2. Implementing Security Mechanisms

In accordance with the security dimensions outlined in Section 3, we have integrated diverse technologies to safeguard the systems and guarantee data integrity. As [74] indicates, there must be a balance between the level of security and the operability of the system. Table 4 summarizes how the security dimensions were addressed in the proof-of-concept application.

1. **Access control:** At the edge layer, the Pre-Shared Key (PSK) is used due to its ability to provide simple but effective authentication of client devices. While the use of stronger measures is recommended, the combination of PSK and filtering by MAC addresses represents the minimum viable alternative for granting access to the WiFi VLAN. In the fog layer, the Lightweight Directory Access Protocol (LDAP), associated with corporate credentials, is implemented to ensure that only authorized personnel can access the corresponding resources. In the Fog layer, the use of the IEEE802.1X in combination with a centralized AAA (Authentication, Authorisation and Audit) provides access control capabilities to the network services.
2. **Authentication:** At the edge layer, all smart sensors have a HW-RoT that grants a secure boot by only running trusted software. These nodes also include flash memory encryption to protect stored data and private keys, as well as security measures against physical fault injection attacks. At the fog layer, all devices are equipped

with TPM NTC 7.2.1.0 version 2.0 TPM security hardware to verify their identity, complemented by two-factor authentication, requiring username and password for authorized personnel access. This access allows verification of the permissions assigned to the user, ensuring that only personnel with the appropriate authorization can interact with the information. All layers make use of x509 certificates for robust and consistent authentication across the communication infrastructure.

3. **Non-repudiation:** Within the network infrastructure, activity monitoring is carried out on Node-to-Device, Node-to-Node and Node-to-Cloud communications. In the presented proof of concept, network monitoring was carried out by the University staff. In parallel, at the fog level, the computer operating system, Windows 10, has an integrated logging mechanism that records all operations performed, thus providing an additional layer of audit and control of system activity.
4. **Data confidentiality:** In the edge layer, certificates in MQTT over TLS are used for Node-to-Device communication and credentials such as user names and passwords to prevent unauthorized access and queries. In the fog layer, in addition to the encryption of communications using certificates, access is controlled through a list of user permissions, restricting access to specific areas of the system. At the cloud layer, Node-to-Cloud communication is secured by means of the HTTPS protocol, which encrypts communications. File permission and access control lists ensure that only authorized personnel can access these communications.
5. **Communication security:** The combination of WiFi VLAN, TCP/IP, and MQTT over TLS provides secure Node-to-Device communications, by means of encryption and authentication. Wired networks are used for node-to-node communications in the fog layer. In addition, the use of corporate firewalls and IDS/IPS systems implements some security features. Node-to-Cloud communication is enforced by means of the HTTPS protocol. In this case, security is enhanced with the use of corporate firewalls and IDS/IPS systems, managed by the networking staff of the University.
6. **Data integrity:** The edge layer ensures data integrity by encrypting the flash memory and using the Digital Signature peripheral, on the ESP32-S2-Saola-1V1.2, to generate RSA digital signatures. This approach protects critical data and private keys from unauthorized access. At the fog layer, the zone and supervisory nodes safeguard the received data, using memory encryption and backups to prevent loss, while the anti-malware and corporate EDR systems may contribute to ensuring the data integrity in these nodes.
7. **Availability:** At the edge layer, only a list of authorized devices are allowed to connect to the WiFi VLAN, following the zero trust principle. These nodes are also restricted to prevent access to higher layers of the network, restricting the communication capabilities in the network. Moving up to the fog layer, the PCs are shielded with the personal firewall system provided by the Windows OS. This approach blocks unauthorized access and maintains the security of the network architecture. In Node-to-Cloud communications, the node is protected from malware attacks by a corporate anti-virus and EDR system, complemented by corporate firewalls and IDS/IPS systems.
8. **Privacy:** At the edge layer, fog layer and cloud layer, the use of the previous mechanisms may help to improve the privacy of the applications. It is important that network activities do not reveal private data, maintaining the confidentiality and integrity of information across the university network.

Table 4. Mechanisms integrated in the proof-of-concept application to achieve the security dimensions of the X.805 recommendation [34].

X.805 Security Dimensions	Node-to-Cloud Security Functions	Node-to-Node Security Functions	Node-to-Device Security Functions
Access control	IEEE 802.1X, Centralized AAA, Directory services (LDAP)	IEEE 802.1X, Centralized AAA, Directory services (LDAP)	PSK+MAC
Authentication	HW-RoT (TPM), X509 certificates, Credentials	HW-RoT (TPM), X509 certificates, Credentials	HW-RoT, X509 certificates, Credentials
Non-repudiation	Network and device logs. Signed data and operations	Network and device logs. Signed data and operations	Network and device logs (if available). Signed data and operations
Data confidentiality	Encryption, Access control list and file permissions. HTTPS	Encryption, Access control list and file permissions. MQTT over TLS	Encryption. MQTT over TLS
Communication security	HTTPS, corporate and personal firewalls, IDS/IPS systems	MQTT over TLS, corporate and personal firewalls, IDS/IPS	MQTT over TLS, WiFi VLAN, corporate(if available) firewalls
Data integrity	Encrypted memory, Backups, Signed data, Antimalware, EDR systems.	Encrypted memory, Backups, Signed data, Antimalware, EDR systems.	Encrypted memory, Signed data
Availability	Secure network configuration (zero-trust), network partitioning, corporate and personal firewalls, IDS/IPS systems.	Secure network configuration (zero-trust), network partitioning, corporate and personal firewalls, IDS/IPS systems,	Secure network configuration (zero-trust), network partitioning, VLANs, corporate and personal firewalls,
Privacy	Secure end-to-end communication technologies, secure network configuration	Secure end-to-end communication technologies, secure network configuration	Secure end-to-end communication technologies, secure network configuration

4.3. Following Implementation Guidelines

This subsection summarizes how the implementation guidelines have been applied in this proof-of-concept application:

1. Fog nodes have the requirements recommended by the OpenFog standard [27] (e.g., TPM module, HW-RoT, personal Firewall and credentials) [85]. Edge nodes have also been analyzed to check that ESP32-S2-Saola-1V1.2 board [84] satisfy the security aspects required by OpenFog (e.g., Secure Boot, Flash encryption, certificates and credentials).
2. Since this is a proof-of-concept system, self-signed certificates have been managed manually using the Node Security Manager. Self-signed certificates offer significant advantages [86]. Also, this is a versatile way to manage certificates for specific intended purposes [87]. All certificates and private keys for edge devices were created using OpenSSL [88]. The following commands are an example of the most relevant ones
   This command establishes the type of certificate, the duration, and the version and creates the public certificate and key:

```
openssl req -new -x509 -days 365 -extensions v3_ca -keyout ca.key
    -out ca.crt -subj /CN="ID_My_server"
```

   After generating the public certificate and key, the broker certificate must be defined:

```
openssl req -out mosquitto.csr -key mosquitto.key -new -subj /CN=
    "server_IP"
```

   This last line saves the public and server certificates:

```
openssl x509 -req -in mosquitto.csr -CA ca.crt -CAkey ca.key -
    CAcreateserial -out mosquitto.crt -days 365
```

   The certificates for the clients are made by this command:

```
openssl req -out esp.csr -key esp.key -new -subj /CN="IP_client"
```

3. The use of flash encryption of firmware and sensor node keys [89] and the use of TPM modules in fog nodes ensures that devices connecting to the architecture's network are authorized and that data are not vulnerable.
   The supervisory and zone nodes as well as the NSM node are located in a restricted access area. These computers are equipped with restricted access policies for only university staff. The encryption of the firmware such as the flash memory encryption, and the secure keys used for accessing the architecture network protect the information, communications and the system [90–92].
4. All MAC addresses of the edge nodes have been registered by the central networking services of the Basque Country University to grant them access to the WiFi VLAN that it is also created by them. Within the VLAN, the edge nodes are hidden from each other. This means that if one device is hacked, it does not put the other sensor nodes at risk, since direct connection among them is not allowed. Only the zone and NSM nodes can connect bidirectionally with edge devices. The security configuration of the system verifies that all devices attempting to connect are trusted by using security keys, credentials and certificates.
5. The configuration of the MQTT broker includes the required credentials and certificates. In the MQTT setup, certificates have been activated, including client certificates, user and password. The anonymous access has been disabled to prevent unauthorized access to the MQTT broker. The certificates and Inbound/Outbound rules created in the personal firewall limit the connectivity to only the devices shown in the image are able to connect.
6. All fog nodes are equipped with the corporate antivirus software of the University. This approach protects devices from malware and trojans, as recommended by the Threat model (see Table 1). The Windows OS has been configured to provide personal firewalls that restrict access from edge devices using a set of selected ports. Figure 5 shows a configuration example of the personal firewall rules in the zone node.
7. The use of LDAP directory services and the use of certificates protects against unauthorized access and it is used to encrypt communications. In Node-to-Cloud communication, the use of HTTPS provides a secure and reliable communication means with the cloud application. The corporate firewall has been configured so that only trusted devices can access the architecture's network.

4.4. Experimental Results

This section describes briefly the operation of the proof-of-concept IEQ system deployed at the facilities of the Faculty of Engineering of Vitoria-Gasteiz. Also, two situations have been recreated that could compromise the security of the system. The behavior of the IEQ system is analyzed when using the Wireshark and Nmap tools. Wireshark may be easily used for a sniffing attack. The second one the Nmap application is used to explore the devices in the WiFi VLAN trying to find out whether ports are available. By means of these examples, the authors intend to illustrate the operation of the security protocols in the proposed deployment.

4.4.1. Operation of the Proof-of-Concept IEQ System

Figure 6 shows the behavior of the system in operation. The flashing red lines indicate the precise moments when the actuator is activated, triggering the opening or closing of the windows, which is an important component for maintaining the desired environmental balance like in [6].

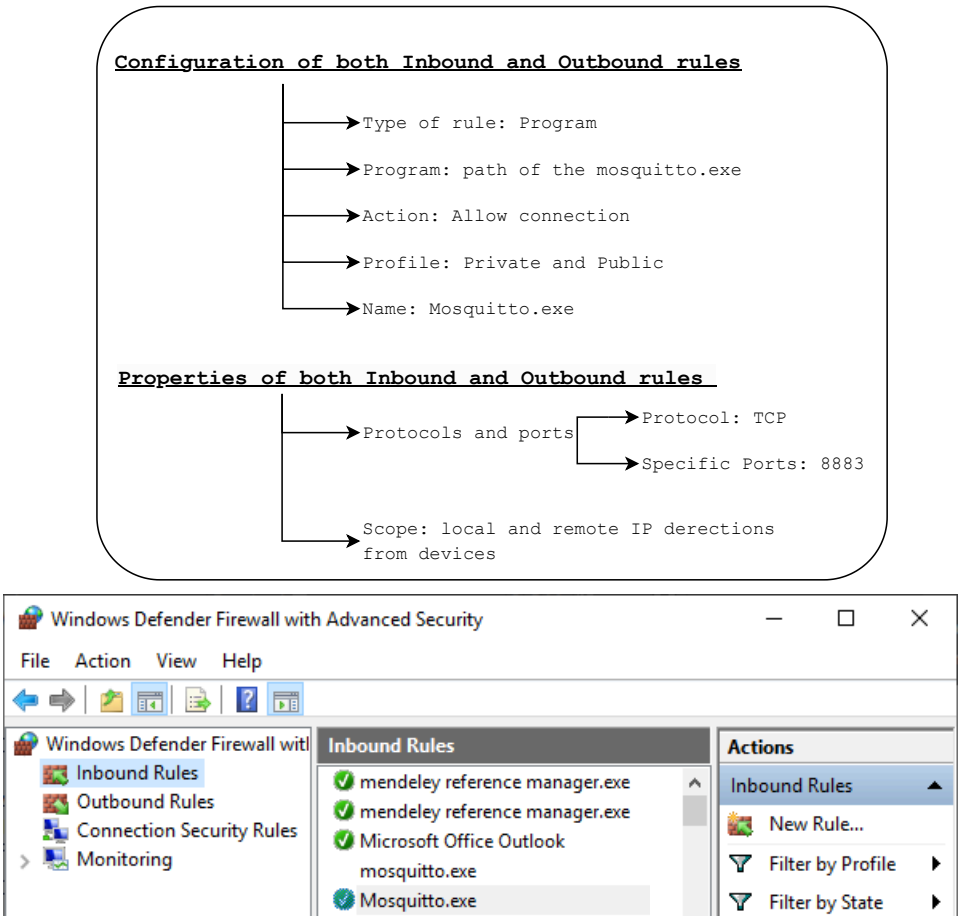

Figure 5. Configuration of the personal firewall rules in the zone node.

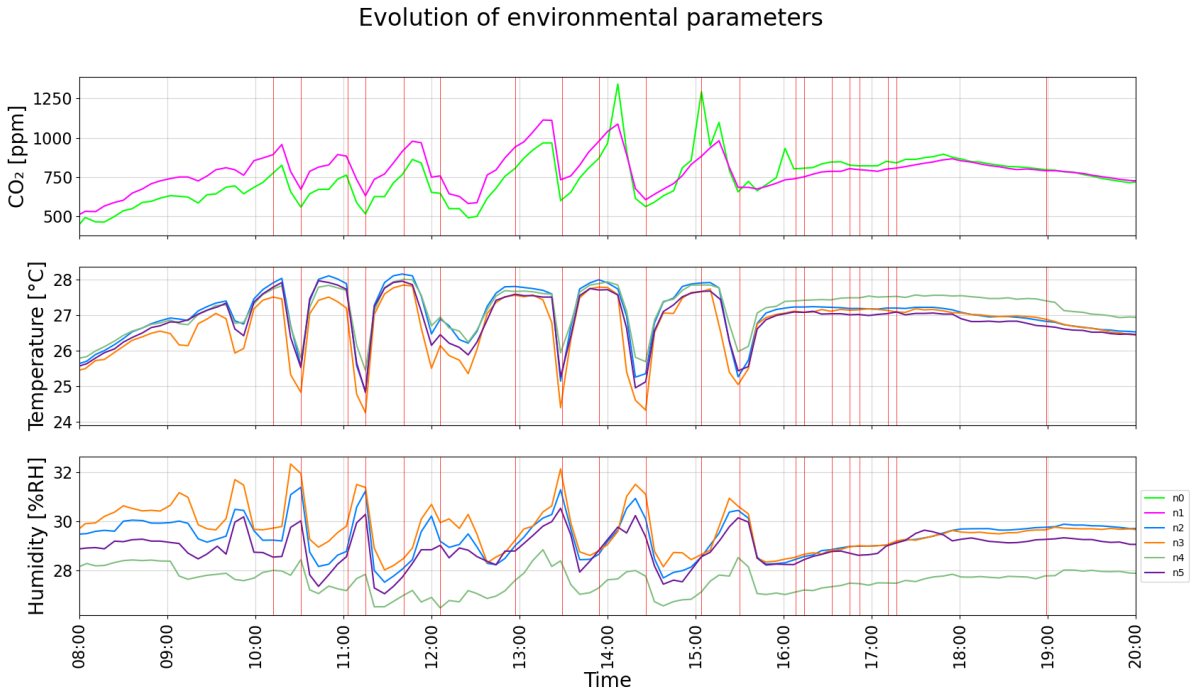

Figure 6. System response to temperature, humidity and CO_2 data.

Figure 7 shows the smart sensor node model, powered by the ESP32-S2-Saola-1V1.2 board. Table 5 summarizes the smart sensor used for this test.

Table 5. List of sensors used in the architecture.

	Group A	Group B
Nodes	N2, N3, N4, and N5	N0 and N1
Paramaters to measure	Temperature, humidity, and light intensity	CO_2
Sensors	- BME680 - KPS-3227-SP1C	- MH-Z19B

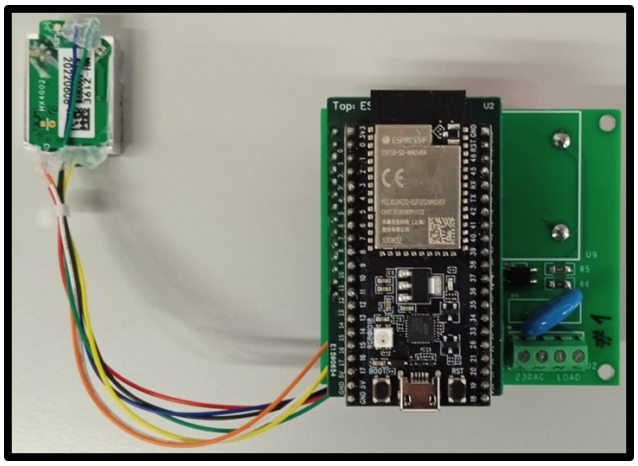

Figure 7. Prototype of CO_2 smart sensor based on ESP32-S2-Saola-1V1.2.

Finally, Figure 8 offers a detailed view of the facilities where the setup was tested. This environment has been selected to replicate the conditions under which the system will eventually be deployed, thus ensuring that the results obtained are as relevant and applicable as possible in real scenarios.

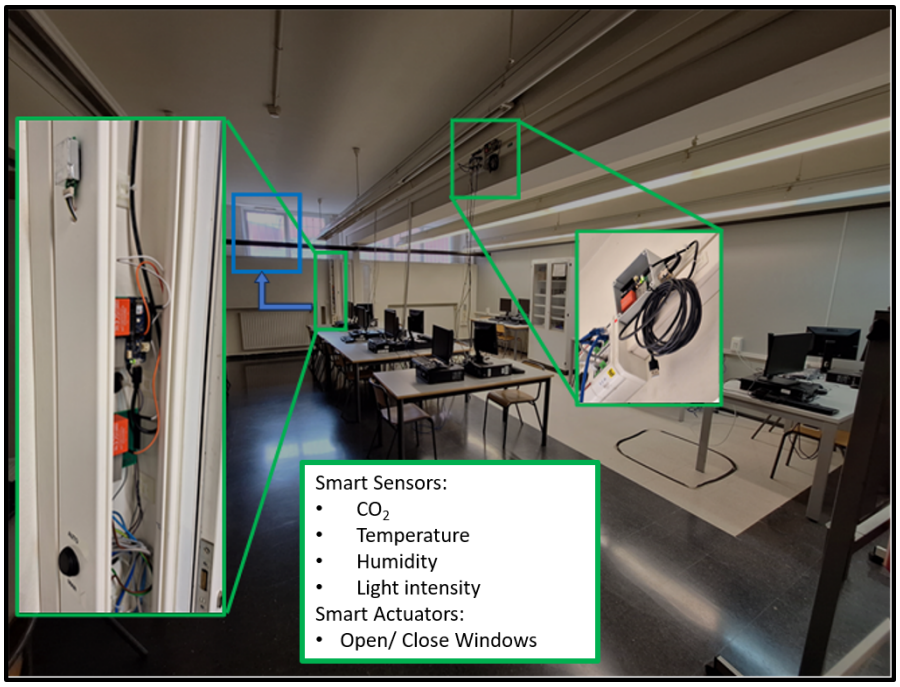

Figure 8. Picture of the teaching laboratory where the IEQ system was deployed.

4.4.2. Security Tests over the IEQ System

Certain actions have been simulated in order to find security weaknesses in the proof-of-concept IEQ system. In this way, it is possible to verify which parts of the system successfully prevent some of the common threats identified in Table 2. Two free and powerful software tools, namely Wireshark and Nmap, were used to simulate how an attacker might look for vulnerabilities in the proof-of-concept IEQ system.

1. **Capture of the traffic generated by the proof-of-concept IEQ system:** This test involved using the Wireshark network packet analyzer. Wireshark is a free and open-source packet analyzer used for network troubleshooting, analysis, software and communications protocol development among others.
   In this test, Wireshark was used to capture the messages exchanged between two nodes of the IEQ system application. These nodes were the Zone node, with the following IP addresses: 192.168.240.217 and one Smart node type A, with IP address 192.168.240.202).
   Figure 9 shows a capture of Wireshark in which it is possible to extract some information from the network traffic, such as the IP addresses as well as the identification of the MQTT protocol. It also detected that the reception TCP port in 192.168.240.217 was 8883, which is a common alternative when MQTT messages over TLS are sent. This information may allow attackers to guess that the computer with IP address 192.168.240.217 is executing the MQTT broker.
   The data within the TCP segment is encrypted, making it inaccessible and thereby preventing sniffing attacks. Additionally, the TLS protocol mandates the use of certificates that are linked to particular computers. For this reason, the presented approach reduces several threats such as MITM attacks among others.

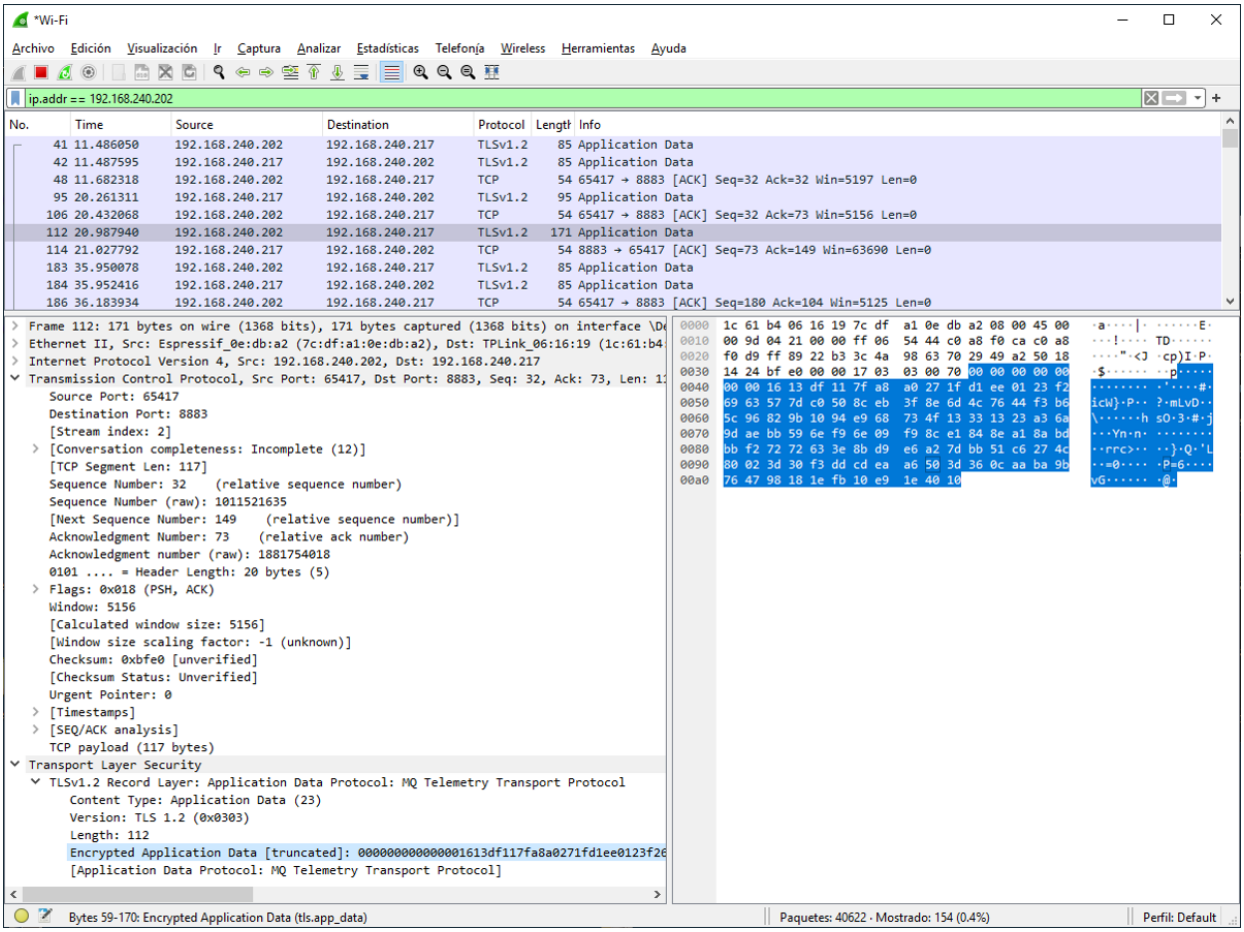

Figure 9. Wireshark capture of the MQTT traffic generated with the proof-of-concept IEQ system.

2. **Network exploration with Nmap:** The software used, Nmap (Network Mapper), is a free and open-source network discovery and security auditing tool. Nmap is frequently used to discover hosts and services on a computer network by sending packets and analyzing the responses. This application is frequently used for exploring IP networks.
In this test a zero-trust approach was been followed. So, the network and the Firewall were configured to restrict the number of devices. Attackers could use this technology to identify the IP addresses of devices connected to the network and the ports used. Figure 10 shows all devices connected to the WiFi VLAN. In the figure, it is possible to find the IP addresses of nine devices. Devices highlighted in blue are smart nodes powered by ESP32 boards. The IP address of the zone node is 192.168.240.217, the supervisory node is 192.168.240.1, and the IP address of the attacker is 192.168.240.228. Figure 10 shows that the zone node (192.168.240.217) has only one open port to accept connections, which is 8883. Although it may be changed, this port is the default MQTT security port for the MQTT broker used in the test. Firewalls were configured to refuse the connections to other ports. Also, the smart nodes (in blue) did not show any open TCP port. Finally, the supervisory node has some open ports for remote configuration and monitoring.
The use of network scanning tools that allow obtaining MAC and IP information poses potential risks. For this reason, the presented approach uses a configuration that reduces the exposition in the operation network to avoid attacks. This approach allows us to mitigate attacks such as MITM, DoS/DDoS, or IP Spoofing. The security measures in place prevent unauthorized access and manipulation, ensuring the integrity and resilience of our network.

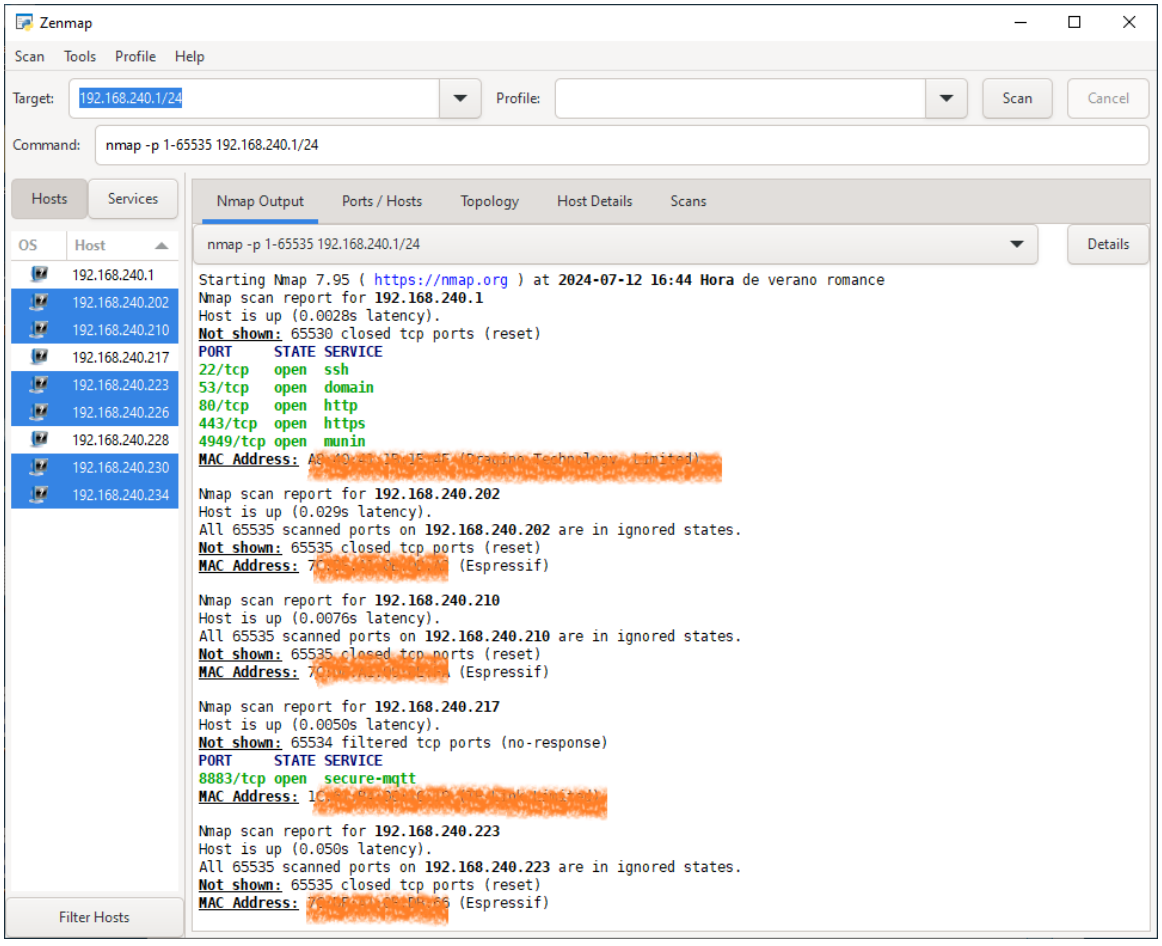

Figure 10. Network exploration with Nmap.

5. Conclusions

Modern smart building IoT applications require the efficient integration of a broad number of heterogeneous, low-resource devices to achieve better monitoring and control of the environmental conditions of the occupants. However, the increased connectivity and accessibility of IoT devices, combined with the common lack of robust security in their deployment within extensive architectures such as smart buildings, result in increased cyber-security threats. Consequently, smart buildings may become vulnerable to cyber-attacks that can cause discomfort, excessive energy usage, or unexpected equipment downtime. Privacy, trust and security should be ensured in smart buildings since a security breach could affect the safety, comfort and privacy of the occupants, as well as cause economic inconveniences. However, security issues in IoT applications require a myriad of components (protocols, hardware and other components) to work in harmony. This can only be achieved by outlining a holistic view of IoT systems. Unfortunately, introducing security mechanisms in smart buildings is critical but it is not trivial. In fact, security issues complicate the creation of smart building applications, increasing the deployment cost of the applications and they are frequently considered as an afterthought. In addition, difficulties may come up to find qualified personnel to implement the required security mechanisms.

This work defines an architecture for smart buildings that tries to achieve security by design and presents some guidelines to assist designers and implementers in creating new applications. This architecture is based on the OpenFog reference architecture, which has been selected for several reasons: (1) OpenFog, established as IEEE 1934 standard, presents a simple and adaptable model that has been identified to be implemented in smart homes and buildings; (2) this standard promotes the use and integration of open hardware and software technologies; (3) the OpenFog standard considers security as a key pillar and identifies the security requirements of the applications. The OpenFog reference architecture allows the combination of a broad number of security mechanisms, not being always obvious to select the best alternatives. Moreover, it does not describe clearly how to implement these mechanisms.

The integration of all the state-of-the-art mechanisms necessary to achieve secure smart building applications poses a considerable challenge, especially when the designers and implementers are not experts in cybersecurity. Firstly, they must learn to use the terminology. In this point, the presentation of use cases may guide them to create new applications. One of the major limitations of this study is that security mechanisms are continuously evolving in order to solve the new vulnerabilities. This requires that the designers must adapt to the new scenarios. Also, the architecture must be able to evolve to include new mechanisms. In the future, the authors will update the security mechanisms used in order to adapt to evolving requirements by adopting state-of-the-art technologies. For example, more recent security technologies, e.g., blockchain, will be evaluated for implementation in the most critical nodes, especially in the Fog layer.

Author Contributions: Conceptualization, I.C.; methodology, I.C., I.M.T. and E.V.; investigation, I.M.T., E.V. and I.C.; writing—original draft preparation, I.M.T. and I.C.; writing—review and editing, E.V., I.M.T., J.M.G.-G. and I.C.; hardware: J.M.G.-G.; visualization, I.M.T. and E.V.; supervision, I.C.; project administration, I.C. and O.B.; funding acquisition, O.B. and I.C. All authors have read and agreed to the published version of the manuscript.

Funding: The authors wish to express their gratitude to the Basque Government through the project EKOHEGAZ II (ELKARTEK KK-2023/00051), to the Diputacion Foral de Alava (DFA) through the project CONAVANTER, to the UPV/EHU through the projects GIU23/002, and to the MobilityLab Foundation (CONV23/14, CONV23/12) for supporting this work.

Data Availability Statement: Data will be available on request.

Acknowledgments: The authors wish to express their gratitude to Juan Carlos Cuesta, network technician at the University of the Basque Country, for his support in the design of the guidelines.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AAA	Authentication, Authorization and Accounting
AES	Advanced Encryption Standard
AI	Artificial Intelligence
AIoT	Artificial Intelligence of Things
ANN	Artificial Neural Network
CA	Certificate Authority
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CIA	Confidentiality, Integrity and Availability
CMAC	Cipher-based Message Authentication Code
DDoS	Distributed Denial of Service
DH	Diffie-Hellman
DPI	Deep Packet Inspection
DSA	Digital Signature Algorithm
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ECQV	Elliptic Curve Qu-Vanstone
EFS	Encrypting File System
EDR	Endpoint Detection and Response
ESM	Event and State Monitoring
FEK	File Encryption Key
FIPS	Federal Information Processing Standards
FOSS	Free and Open Source Software
GCM	Galois/Counter Mode
GMAC	Galois Message Authentication Code
HAS	Home Automation System
HMAC	Hash-based Message Authentication Code
HW-RoT	Hardware Root of Trust
IAQ	Indoor Air Quality
ICC	Industrial Communications Consortium
IEQ	Indoor Environmental Quality
IIRA	Industrial Internet Reference Architecture
IMSA	Intelligent Manufacturing System Architecture
IoT	Internet of Things
IPS/IDS	Intrusion Detection and Protection System
IVRA	Industrial Value Chain Reference Architecture
LDAP	Lightweight Directory Access Protocol
MAC	Media Access Control
MUD	Manufacturer Usage Description
NIST	National Institute of Standards and Technology
NSM	Node Security Manager
OOB	Out-of-Band
OS	Operating System
PSK	Pre-Shared Key
PSP	Platform Security Processor
PVC	Polyvinyl Chloride
RAMI 4.0	Reference Architecture Model Industrie 4.0
RSA	Rivest-Shamir-Adleman
SBC	Single Board Computing
SDN	Software Defined Networking
SHA	Secure Hash Algorithm
SME	Small or Medium-sized Enterprise
SoC	System on Chip
TCB	Trusted Computing Base

TPM	Trusted Platform Module
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPN	Virtual Private Network

References

1. Jia, M.; Komeily, A.; Wang, Y.; Srinivasan, R.S. Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications. *Autom. Constr.* **2019**, *101*, 111–126. [CrossRef]
2. Jiang, T.; Li, Z.; Jin, X.; Chen, H.; Li, X.; Mu, Y. Flexible operation of active distribution network using integrated smart buildings with heating, ventilation and air-conditioning systems. *Appl. Energy* **2018**, *226*, 181–196. [CrossRef]
3. Huotari, M.; Malhi, A.; Främling, K. Machine Learning Applications for Smart Building Energy Utilization: A Survey. *Arch. Comput. Methods Eng.* **2024**, *31*. [CrossRef]
4. Starace, G.; Tiwari, A.; Colangelo, G.; Massaro, A. Advanced Data Systems for Energy Consumption Optimization and Air Quality Control in Smart Public Buildings Using a Versatile Open Source Approach. *Electronics* **2022**, *11*, 3904. [CrossRef]
5. Bushnag, A. An improved air quality and climate control monitoring system using fuzzy logic for enclosed areas. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 6339–6347. [CrossRef]
6. Toral, I.M.; Calvo, I.; Xenakis, J.; Artetxe, E.; Barambones, O. Architecture for Smart Buildings Based on Fuzzy Logic and the OpenFog Standard. *Electronics* **2023**, *12*, 4889. [CrossRef]
7. Vanus, J.; Gorjani, O.M.; Bilik, P. Novel proposal for prediction of CO_2 course and occupancy recognition in intelligent buildings within IoT. *Energies* **2019**, *12*, 4541. [CrossRef]
8. Khazaei, B.; Shiehbeigi, A.; Kani, A.R.H.M.A. Modeling indoor air carbon dioxide concentration using artificial neural network. *Int. J. Environ. Sci. Technol.* **2019**, *16*, 729–736. [CrossRef]
9. Wang, Y.; Zhang, B.; Ma, J.; Jin, Q. Artificial intelligence of things (AIoT) data acquisition based on graph neural networks: A systematical review. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7827. [CrossRef]
10. Verma, A.; Prakash, S.; Srivastava, V.; Kumar, A.; Mukhopadhyay, S.C. Sensing, Controlling, and IoT Infrastructure in Smart Building: A Review. *IEEE Sens. J.* **2019**, *19*, 9036–9046. [CrossRef]
11. Eini, R.; Linkous, L.; Zohrabi, N.; Abdelwahed, S. Smart building management system: Performance specifications and design requirements. *J. Build. Eng.* **2021**, *39*, 102222. [CrossRef]
12. Minoli, D.; Sohraby, K.; Occhiogrosso, B. IoT Considerations, Requirements, and Architectures for Smart Buildings-Energy Optimization and Next-Generation Building Management Systems. *IEEE Internet Things J.* **2017**, *4*, 269–283. [CrossRef]
13. Sabireen, H.; Neelanarayanan, V. A Review on Fog Computing: Architecture, Fog with IoT, Algorithms and Research Challenges. *ICT Express* **2021**, *7*, 162–176. [CrossRef]
14. Martin, B.A.; Michaud, F.; Banks, D.; Mosenia, A.; Zolfonoon, R.; Irwan, S.; Schrecker, S.; Zao, J.K. OpenFog security requirements and approaches. In Proceedings of the 2017 IEEE Fog World Congress, FWC 2017, Santa Clara, CA, USA, 30 October–1 November 2017; pp. 1–6. [CrossRef]
15. Yousefpour, A.; Fung, C.; Nguyen, T.; Kadiyala, K.; Jalali, F.; Niakanlahiji, A.; Kong, J.; Jue, J.P. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *J. Syst. Archit.* **2019**, *98*, 289–330. [CrossRef]
16. Rahimi, M.; Songhorabadi, M.; Kashani, M.H. Fog-based smart homes: A systematic review. *J. Netw. Comput. Appl.* **2020**, *153*, 102531. [CrossRef]
17. Omoniwa, B.; Hussain, R.; Javed, M.A.; Bouk, S.H.; Malik, S.A. Fog/edge computing-based IoT (FECIoT): Architecture, applications, and research issues. *IEEE Internet Things J.* **2019**, *6*, 4118–4149. [CrossRef]
18. Li, G.; Ren, L.; Fu, Y.; Yang, Z.; Adetola, V.; Wen, J.; Zhu, Q.; Wu, T.; Candan, K.S.; O'Neill, Z. A critical review of cyber-physical security for building automation systems. *Annu. Rev. Control* **2023**, *55*, 237–254. [CrossRef]
19. Guan, Y.; Shao, J.; Wei, G.; Xie, M. Data Security and Privacy in Fog Computing. *IEEE Netw.* **2018**, *32*, 106–111. [CrossRef]
20. Heartfield, R.; Loukas, G.; Budimir, S.; Bezemskij, A.; Fontaine, J.R.; Filippoupolitis, A.; Roesch, E. A taxonomy of cyber-physical threats and impact in the smart home. *Comput. Secur.* **2018**, *78*, 398–428. [CrossRef]
21. Sivanathan, A.; Loi, F.; Gharakheili, H.H.; Sivaraman, V. Experimental evaluation of cybersecurity threats to the smart-home. In Proceedings of the 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Bhubaneswar, India, 17–20 December 2017; pp. 1–6. [CrossRef]
22. Mulero-Palencia, S.; Baeza, V.M. Detection of Vulnerabilities in Smart Buildings Using the Shodan Tool. *Electronics* **2023**, *12*, 4815. [CrossRef]
23. Gebremichael, T.; Ledwaba, L.P.; Eldefrawy, M.H.; Hancke, G.P.; Pereira, N.; Gidlund, M.; Akerberg, J. Security and Privacy in the Industrial Internet of Things: Current Standards and Future Challenges. *IEEE Access* **2020**, *8*, 152351–152366. [CrossRef]
24. Atlam, H.F.; Wills, G.B. IoT Security, Privacy, Safety and Ethics. *Internet Things* **2020**, 123–149. [CrossRef]
25. Khan, S.; Parkinson, S.; Qin, Y. Fog computing security: A review of current applications and security solutions. *J. Cloud Comput.* **2017**, *6*, 19. [CrossRef]
26. Nakagawa, E.Y.; Antonino, P.O.; Schnicke, F.; Capilla, R.; Kuhn, T.; Liggesmeyer, P. Industry 4.0 reference architectures: State of the art and future trends. *Comput. Ind. Eng.* **2021**, *156*, 107241. [CrossRef]

27. Adoption of OpenFog Reference Architecture for Fog Computing (IEEE Standard 1934–2018). IEEE Communications Society. 2018. Available online: https://ieeexplore.ieee.org/document/8423800 (accessed on 6 November 2023).
28. OpenFog Reference Architecture for Fog Computing 2017. Available online: https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf (accessed on 6 November 2023).
29. Barton, M.; Budjac, R.; Tanuska, P.; Gaspar, G.; Schreiber, P. Identification Overview of Industry 4.0 Essential Attributes and Resource-Limited Embedded Artificial-Intelligence-of-Things Devices for Small and Medium-Sized Enterprises. *Appl. Sci.* **2022**, *12*, 5672. [CrossRef]
30. Dodson, D.; Montgomery, D.; Polk, T.; Ranganathan, M.; Souppaya, M.; Johnson, S.; Kadam, A.; Pratt, C.; Thakore, D.; Walker, M.; et al. *Securing Small-Business and Home Internet of Things (IoT) Devices: Mitigating Network-Based Attacks Using Manufacturer Usage Description (MUD)*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2021. [CrossRef]
31. Jhanjhi, N.Z.; Humayun, M.; Almuayqil, S.N. Cyber Security and Privacy Issues in Industrial Internet of Things. *Comput. Syst. Sci. Eng.* **2021**, *37*, 361–380. [CrossRef]
32. Kuo, P.H.; Mourad, A.; Lu, C.; Berg, M.; Duquennoy, S.; Chen, Y.Y.; Hsu, Y.H.; Zabala, A.; Ferrari, R.; Gonzalez, S.; et al. An integrated edge and Fog system for future communication networks. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2018, Barcelona, Spain, 15–18 April 2018; pp. 338–343. [CrossRef]
33. Seliem, M.; Elgazzar, K.; Khalil, K. Towards Privacy Preserving IoT Environments: A Survey. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 1032761. [CrossRef]
34. X.805: Security Architecture for Systems Providing End-to-End Communications. Available online: https://www.itu.int/rec/T-REC-X.805-200310-I/en (accessed on 17 July 2024).
35. Chanal, P.M.; Kakkasageri, M.S. Security and Privacy in IoT: A Survey. *Wirel. Pers. Commun.* **2020**, *115*, 1667–1693. [CrossRef]
36. Chen, W.; Ding, D.; Dong, H.; Wei, G. Distributed Resilient Filtering for Power Systems Subject to Denial-of-Service Attacks. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1688–1697. [CrossRef]
37. Casteur, G.; Aubaret, A.; Blondeau, B.; Clouet, V.; Quemat, A.; Pical, V.; Zitouni, R. Fuzzing attacks for vulnerability discovery within MQTT protocol. In Proceedings of the 2020 International Wireless Communications and Mobile Computing, IWCMC 2020, Limassol, Cyprus, 15–19 June 2020; pp. 420–425. [CrossRef]
38. Ahmad, F.; Kurugollu, F.; Adnane, A.; Hussain, R.; Hussain, F. MARINE: Man-in-the-Middle Attack Resistant Trust Model in Connected Vehicles. *IEEE Internet Things J.* **2020**, *7*, 3310–3322. [CrossRef]
39. Alkhwaja, I.; Albugami, M.; Alkhwaja, A.; Alghamdi, M.; Abahussain, H.; Alfawaz, F.; Almurayh, A.; Min-Allah, N. Password Cracking with Brute Force Algorithm and Dictionary Attack Using Parallel Programming. *Appl. Sci.* **2023**, *13*, 5979. [CrossRef]
40. Chen, B.; Ho, D.W.; Hu, G.; Yu, L. Secure Fusion Estimation for Bandwidth Constrained Cyber-Physical Systems under Replay Attacks. *IEEE Trans. Cybern.* **2018**, *48*, 1862–1876. [CrossRef] [PubMed]
41. Prabadevi, B.; Jeyanthi, N. A Review on Various Sniffing Attacks and its Mitigation Techniques. *Indones. J. Electr. Eng. Comput. Sci.* **2018**, *12*, 1117–1125. [CrossRef]
42. Anthi, E.; Williams, L.; Slowinska, M.; Theodorakopoulos, G.; Burnap, P. A Supervised Intrusion Detection System for Smart Home IoT Devices. *IEEE Internet Things J.* **2019**, *6*, 9042–9053. [CrossRef]
43. Esquivel-Vargas, H.; Caselli, M.; Peter, A. Automatic deployment of specification-based intrusion detection in the BACnet Protocol. In Proceedings of the CPS-SPC 2017—Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy, co-Located with CCS 2017, Dallas, TX, USA, 3 November 2017; Volume 17, pp. 25–36. [CrossRef]
44. Zheng, Z.; Reddy, A.L. Safeguarding building automation networks: THE-driven anomaly detector based on traffic analysis. In Proceedings of the 2017 26th International Conference on Computer Communications and Networks, ICCCN 2017, Vancouver, BC, Canada, 31 July–3 August 2017. [CrossRef]
45. Yang, Y.S.; Lee, S.H.; Chen, W.C.; Yang, C.S.; Huang, Y.M.; Hou, T.W. Securing SCADA Energy Management System under DDos Attacks Using Token Verification Approach. *Appl. Sci.* **2022**, *12*, 530. [CrossRef]
46. Sheikh, A.; Kamuni, V.; Patil, A.; Wagh, S.; Singh, N. Cyber Attack and Fault Identification of HVAC System in Building Management Systems. In Proceedings of the 2019 9th International Conference on Power and Energy Systems, ICPES 2019, Perth, WA, Australia, 10–12 December 2019. [CrossRef]
47. Hachem, J.E.; Chiprianov, V.; Babar, M.A.; Khalil, T.A.; Aniorte, P. The Journal of Systems and Software Modeling, analyzing and predicting security cascading attacks in smart buildings systems-of-systems Systems-of-systems Security modeling and analysis Model driven engineering Software architecture Multi-agent systems simulation Smart buildings. *J. Syst. Softw.* **2020**, *162*, 110484. [CrossRef]
48. Peacock, M. Anomaly Detection in BACnet/IP Managed Building Automation Systems. Ph.D. Thesis, Edith Cowan University, Joondalup, Australia, 2019.
49. Zhang, F.; Kodituwakku, H.A.D.E.; Hines, J.W.; Coble, J. Multilayer Data-Driven Cyber-Attack Detection System for Industrial Control Systems Based on Network, System, and Process Data. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4362–4369. [CrossRef]
50. Fauri, D.; Kapsalakis, M.; dos Santos, D.R.; Costante, E.; den Hartog, J.; Etalle, S. Leveraging semantics for actionable intrusion detection in building automation systems. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) LNCS*; Springer Science+Business Media: Berlin, Germany, 2019; Volume 11260, pp. 113–125. [CrossRef]

51. Zhang, S.; Rong, J.; Wang, B. A privacy protection scheme of smart meter for decentralized smart home environment based on consortium blockchain. *Int. J. Electr. Power Energy Syst.* **2020**, *121*, 106140. [CrossRef]
52. Feng, T.; Zhang, B.; Liu, C.; Zheng, L. Security assessment and improvement of building ethernet KNXnet/IP protocol. *Discov. Appl. Sci.* **2024**, *6*, 162. [CrossRef]
53. Daneshgar, F.F.; Abbaspour, M. Extracting fuzzy attack patterns using an online fuzzy adaptive alert correlation framework. *Secur. Commun. Netw.* **2016**, *9*, 2245–2260. [CrossRef]
54. Ban, X.; Ding, M.; Liu, S.; Chen, C.; Zhang, J. IoTFuzz: Automated Discovery of Violations in Smart Homes with Real Environment. *IEEE Internet Things J.* **2024**, *11*, 10183–10196. [CrossRef]
55. Fovino, I.N.; Coletta, A.; Carcano, A.; Masera, M. Critical state-based filtering system for securing SCADA network protocols. *IEEE Trans. Ind. Electron.* **2012**, *59*, 3943–3950. [CrossRef]
56. Ding, D.; Han, Q.L.; Xiang, Y.; Ge, X.; Zhang, X.M. A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing* **2018**, *275*, 1674–1683. [CrossRef]
57. Lee, J.; Yu, S.; Park, K.; Park, Y.; Park, Y. Secure Three-Factor Authentication Protocol for Multi-Gateway IoT Environments. *Sensors* **2019**, *19*, 2358. [CrossRef] [PubMed]
58. Elnour, M.; Meskin, N.; Khan, K.; Jain, R. Application of data-driven attack detection framework for secure operation in smart buildings. *Sustain. Cities Soc.* **2021**, *69*, 102816. [CrossRef]
59. Paridari, K.; Mady, A.E.D.; Porta, S.L.; Chabukswar, R.; Blanco, J.; Teixeira, A.; Sandberg, H.; Boubekeur, M. Cyber-Physical-Security Framework for Building Energy Management System. In Proceedings of the 2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems, ICCPS 2016—Proceedings, Vienna, Austria, 11–14 April 2016. [CrossRef]
60. Ji, X.; Li, C.; Zhou, X.; Zhang, J.; Zhang, Y.; Xu, W. Authenticating Smart Home Devices via Home Limited Channels. *ACM Trans. Internet Things* **2020**, *1*, 24. [CrossRef]
61. Lahmadi, A.; Duque, A.; Heraief, N.; Francq, J. MitM Attack Detection in BLE Networks Using Reconstruction and Classification Machine Learning Techniques. *Commun. Comput. Inf. Sci.* **2020**, *1323*, 149–164. [CrossRef]
62. Aloseel, A.; Al-Rubaye, S.; Zolotas, A.; He, H.; Shaw, C. A Novel Approach for Detecting Cyberattacks in Embedded Systems Based on Anomalous Patterns of Resource Utilization-Part i. *IEEE Access* **2021**, *9*, 103204–103229. [CrossRef]
63. McBride, J.; Hernandez-Castro, J.; Arief, B. Earworms Make Bad Passwords: An Analysis of the Nokē Smart Lock Manual Override. In Proceedings of the 2017 International Workshop on Secure Internet of Things, SIoT 2017, Oslo, Norway, 15 September 2017; pp. 30–39. [CrossRef]
64. Helen, D. Exploring cyber attacks in blockchain technology enabled green smart city. In *Green Blockchain Technology for Sustainable Smart Cities*; Elsevier: Amsterdam, The Netherlands, 2023. [CrossRef]
65. Acar, A.; Fereidooni, H.; Abera, T.; Sikder, A.K.; Miettinen, M.; Aksu, H.; Conti, M.; Sadeghi, A.R.; Uluagac, S. Peek-a-boo: I see your smart home activities, even encrypted! In Proceedings of the WiSec 2020—Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Linz, Austria, 8–10 July 2020; pp. 207–218. [CrossRef]
66. Vaccari, I.; Cambiaso, E.; Aiello, M. Evaluating Security of Low-Power Internet of Things Networks. *Int. J. Comput. Digit. Syst.* **2019**, *8*, 101–114. [CrossRef]
67. Liu, X.; Zeng, Q.; Du, X.; Valluru, S.L.; Fu, C.; Fu, X.; Luo, B. SniffMislead: Non-intrusive privacy protection against wireless packet sniffers in smart homes. In Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses, San Sebastian, Spain, 6–8 October 2021; pp. 33–47. [CrossRef]
68. Ahlawat, B.; Sangwan, A.; Sindhu, V. IOT System Model,Challenges and Threats. *Artic. Int. J. Sci. Technol. Res.* **2020**, *9*, 6771–6776.
69. Kumar, A.; Sharma, S.; Goyal, N.; Singh, A.; Cheng, X.; Singh, P. Secure and energy-efficient smart building architecture with emerging technology IoT. *Comput. Commun.* **2021**, *176*, 207–217. [CrossRef]
70. Filho, G.P.; Meneguette, R.I.; Maia, G.; Pessin, G.; Gonçalves, V.P.; Weigang, L.; Ueyama, J.; Villas, L.A. A fog-enabled smart home solution for decision-making using smart objects. *Future Gener. Comput. Syst.* **2020**, *103*, 18–27. [CrossRef]
71. Froiz-Míguez, I.; Fernández-Caramés, T.M.; Fraga-Lamas, P.; Castedo, L. Design, Implementation and Practical Evaluation of an IoT Home Automation System for Fog Computing Applications Based on MQTT and ZigBee-WiFi Sensor Nodes. *Sensors* **2018**, *18*, 2660. [CrossRef]
72. Gordon, H.; Batula, C.; Tushir, B.; Dezfouli, B.; Liu, Y. Securing smart homes via software-defined networking and low-cost traffic classification. In Proceedings of the 2021 IEEE 45th Annual Computers, Software, and Applications Conference, COMPSAC 2021, Madrid, Spain, 12–16 July 2021; pp. 1049–1057. [CrossRef]
73. Younus, M.U.; ul Islam, S.; Ali, I.; Khan, S.; Khan, M.K. A survey on software defined networking enabled smart buildings: Architecture, challenges and use cases. *J. Netw. Comput. Appl.* **2019**, *137*, 62–77. [CrossRef]
74. Alabady, S.A.; Al-Turjman, F.; Din, S. A Novel Security Model for Cooperative Virtual Networks in the IoT Era. *Int. J. Parallel Program.* **2020**, *48*, 280–295. [CrossRef]
75. Tan, L.; Yu, K.; Ming, F.; Cheng, X.; Srivastava, G. Secure and Resilient Artificial Intelligence of Things: A HoneyNet Approach for Threat Detection and Situational Awareness. *IEEE Consum. Electron. Mag.* **2022**, *11*, 69–78. [CrossRef]
76. Cisco, C.M. AAA PROTOCOLS: Authentication, Authorization, and Accounting for the Internet. *IEEE Internet Comput.* **1999**, *3*, 75–79.

77. Katsikeas, S.; Fysarakis, K.; Miaoudakis, A.; Bemten, A.V.; Askoxylakis, I.; Papaefstathiou, I.; Plemenos, A. Lightweight & secure industrial IoT communications via the MQ telemetry transport protocol. In Proceedings of the IEEE Symposium on Computers and Communications, Heraklion, Greece, 3–6 July 2017; pp. 1193–1200. [CrossRef]
78. Lakshminarayana, S.; Praseed, A.; Thilagam, P.S. Securing the IoT Application Layer from an MQTT Protocol Perspective: Challenges and Research Prospects. *IEEE Commun. Surv. Tutor.* **2024**, 1. [CrossRef]
79. Mishra, B.; Kertesz, A. The use of MQTT in M2M and IoT systems: A survey. *IEEE Access* **2020**, *8*, 201071–201086. [CrossRef]
80. IEEE. *802.1X-2020-IEEE Standard for Local and Metropolitan Area Networks–Port-Based Network Access Control*; IEEE: Piscataway, NJ, USA, 2020; ISBN 978-1-5044-6440-6. Available online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9018454 (accessed on 6 June 2024).
81. Dobraunig, C.; Eichlseder, M.; Mendel, F. Analysis of SHA-512/224 and SHA-512/256. In Proceedings of the Advances in Cryptology–ASIACRYPT 2015: 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, 29 November–3 December 2015. [CrossRef]
82. May, W.E. *Approved Random Number Generators for FIPS PUB 140-2, Security Requirements for Cryptographic Modules*; FIPS PUB. 2021. Available online: www.nist.gov/cmvp (accessed on 6 June 2024).
83. Weather API—OpenWeatherMap. Available online: https://openweathermap.org/api (accessed on 6 June 2024).
84. ESP32S2WROVER ESP32S2WROVERI Datasheet. 2022. Available online: www.espressif.com (accessed on 17 January 2024).
85. Device protection in Windows Security—Microsoft Support. Available online: https://support.microsoft.com/en-us/windows/device-protection-in-windows-security-afa11526-de57-b1c5-599f-3a4c6a61c5e2 (accessed on 8 July 2024).
86. Garba, A.; Khoury, D.; Balian, P.; Haddad, S.; Sayah, J.; Chen, Z.; Guan, Z.; Hamdan, H.; Charafeddine, J.; Al-Mutib, K. LightCert4IoTs: Blockchain-Based Lightweight Certificates Authentication for IoT Applications. *IEEE Access* **2023**, *11*, 28370–28383. [CrossRef]
87. Li, B.; Lin, J.; Wang, Q.; Wang, Z.; Jing, J. Locally-Centralized Certificate Validation and its Application in Desktop Virtualization Systems. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 1380–1395. [CrossRef]
88. OpenSSL. Available online: https://www.openssl.org/ (accessed on 8 July 2024).
89. Flash Encryption ESP32 ESP-IDF Programming Guide Latest Documentation. Available online: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/security/flash-encryption.html (accessed on 8 July 2024).
90. LUKS on Raspberry Pi | LUKS-on-Raspberry-Pi. Available online: https://rr-developer.github.io/LUKS-on-Raspberry-Pi/ (accessed on 8 July 2024).
91. Raspberry Pi-Full Disk Encryption | Kali Linux Documentation. Available online: https://www.kali.org/docs/arm/raspberry-pi-with-luks-full-disk-encryption-2/ (accessed on 8 July 2024).
92. Increasing security | The Raspberry Pi Guide. Available online: https://raspberrypi-guide.github.io/other/Improve-raspberry-pi-security (accessed on 8 July 2024).