IoTFC: A Secure and Privacy Preserving Architecture for Smart Buildings

Amna Qureshi(✉), M. Shahwaiz Afaqui, and Julián Salas

Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC), Barcelona, Spain
{aqureshi,mafaqui,jsalaspi}@uoc.edu

Abstract. In the pursuit of cities to be more efficient and responsive, various kind of Internet of Things (IoT) devices, such as actuators and sensors are used. This paper focuses on one specific IoT application - the smart building, and investigates the security and privacy issues in an integrated IoT-fog-cloud (IoTFC) smart building architecture. We consider the surveillance, maintenance, environment, and concierge use cases for smart building, in terms of their characteristics, compatible communication technology, and security and privacy requirements. IoTFC provides a comprehensive solution to the security and privacy challenges of authentication, access control, anomaly detection, data privacy and location privacy. To the best of our knowledge, IoTFC is a novel architecture, as it combines a complete set of light-weight security and privacy solutions suitable for smart buildings.

Keywords: Smart building · Fog computing · IoT solutions · Security · Privacy

1 Introduction

Despite the potential benefits of a smart building, the reliance on the collection of data by multiple entities contradicts the expectations of privacy. The raw information collected by different sensors such as temperature control, elevators, fire alarms, motion detectors or surveillance cameras, etc., can yield a tremendous amount of data that can be combined, analyzed and acted upon, all potentially without adequate accountability, security or meaningful consent e.g., the building manager can perform user profiling. Additionally, there are other privacy concerns [15], such as privacy-violating interactions, presentations, (e.g.,

This work was partly funded by the Spanish Government through grants INCIBEC-2015-02491 "Ayudas para la excelencia de los equipos de investigaciòn avanzada en ciberseguridad", RTI2018-095094-B-C22 "CONSENT". This research is partially funded by the SPOTS project (RTI2018-095438-A-I00) funded by the Spanish Ministry of Science, Innovation & Universities and 2017 SGR 60 Funded by the Generalitat de Catalunya.

D. Wang et al. (Eds.): SPNCE 2020, LNICST 344, pp. 102–119, 2021.
https://doi.org/10.1007/978-3-030-66922-5_7

answering a private query by displaying it on a screen that may be display of a private query accessible to others), transitions (when the IoT devices change of owner or location of devices), inventory attacks (the possibility to learn externally to a building what are its characteristics, equipment or personal devices) and the information linkage risk due to the collection of data from multiple sensors. Thus, privacy of a smart building requires the protection of occupant's private data, patterns, and interactions with other people or IoT devices. Unfortunately, such privacy challenges are only partially addressed in smart buildings.

Ubiquitous wireless communication technologies and protocols connecting the IoT devices, systems and application platform (building automation system) to each other and to the Internet are vulnerable to cyber attacks, e.g., if an automation system is compromised, a hacker can potentially take control of security functions such as locking doors to potentially gain access to a building, e.g., in Finland, a DDoS attack targeting the heating system left residents of two building apartments in the cold. In 2017, hackers reportedly took control of the electronic key system of a hotel in the Austrian Alps. Fully booked and with little alternative, the hotel paid the 1,600 euro ransom [5]. A recent Kaspersky report [6] on security of smart buildings revealed that out of the 26.5% protected smart building systems management computers, 12% were attacked with malware aimed at stealing account credentials and other valuable information. The sheer range of device types, conflicting and incompatible standards, and complex protocols makes securing these systems a daunting task for the security designers.

In the literature, there exists many solutions addressing security and privacy properties based on traditional cryptographic primitives, which are not compatible with the computational capacities of IoT devices. Thus, there is a need to design a reliable and light-weight security and privacy architecture for a smart building that can provide device authenticity, data confidentiality, location and data privacy, anomaly detection, secure data transmission and storage.

As an alternative, the paradigm of fog computing can be used as a viable solution to enhance the security within smart building systems. Fog computing extends the cloud computing platform with additional computational, storage and networking resources placed in the immediate vicinity of the end user devices. The proximity of the fog to IoT devices enables fog computing to deliver extremely low network latencies between the end user devices and the fog computing resources serving them, and to locally process transient data. In addition, decentralized and distributed data control makes fog computing a viable deployment method in a smart building to offer various features such as scalability, autonomy, manageability and security, etc. Fog nodes in the building can have security requirements as well as the connectivity and coordination to feed the larger cloud-based systems with the data needed to inform their difficult decisions.

Contributions and Plan of the Paper. This paper proposes a novel architecture based on fog computing and various IoT sensors considered for four use cases (surveillance, maintenance, environment, and concierge). IoTFC is a thorough

architecture, which provides security (such as authentication, access control, confidentiality, etc.) and privacy (in terms of location and data) features through a combination of different network technologies, middleware transport protocols, and light weight cryptographic protocols implemented over open source solutions. As compared to other related works in literature, this architecture provides mechanisms to counter a complete set of security and privacy issues expected in smart buildings.

The remainder of the paper is organized as follows: In Sect. 2, we present the selected use cases of a smart building. Section 3 discusses the communication technologies suitable for smart buildings. In Sect. 4, we present the design details of the proposed IoTFC. Section 5 compares the relevant state-of-the-art smart building solutions with IoTFC. Finally, Sect. 6 concludes the paper.

2 Use Cases

This section describes four use cases for smart buildings spanning building and occupants' security, operational efficiency, energy conservation, efficient space utilization and occupant satisfaction. Also, a brief summary of the security and privacy threats in these cases is presented.

- ***Smart surveillance:*** It enables building managers to perform the following functions: 1) manage and control surveillance devices remotely; 2) make smarter decisions based on real-time security conditions; (3) detect false alarm without physical inspection; and 4) collect and analyze data to make important improvements to security processes of the building.
- ***Smart maintenance:*** It allows the building managers and operators to improve efficiency and optimize building operations in the following ways: 1) detect temperature changes, water pressure variations, or other abnormalities, and alert operational systems to prevent damage; 2) use predictive maintenance to avoid outages or significant downtime caused by maintenance; and 3) provide a swift response to any breakdown incident.
- ***Smart environment:*** It enables the building managers to perform the following functions: 1) monitor and adjust building systems, such as lighting, air quality, etc. to match comfort levels and reduce energy waste; 2) control and monitor entry and exit points of the building for better security decision-making and disaster prevention; and 3) automatic garbage collection from the dumpsters upon reaching the fill level.
- ***Smart concierge:*** It enables the building managers to support occupants in performing following predefined day-to-day tasks: 1) locate occurring events or available meeting spaces by date, time or room size; 2) book appointments or conference halls; 3) order a taxi or rental car/bike; and 4) registering visitors/guests without the inhabitant having to personally go to the reception.

Table 1 summarizes the possible security and privacy threats of each of the aforementioned use case. Many solutions have been proposed to either provide security or privacy against the attacks mentioned in Table 1, such as protection

Table 1. Security and privacy threats.

Threats	Use cases			
	Smart Surveillance	**Smart Maintenance**	**Smart Environment**	**Smart Concierge**
Security	• Remote tampering by hackers to use attacked cameras as spy tools • Manipulation or deletion of surveillance footage stored on local or global data centers • Collection of additional sensitive information (Wi-Fi or users' credentials) • DDoS attacks • Unauthorized access to the building's sensitive or critical information	• Create network congestion with false alarms generated by hacked IoT device • Access to other sensitive building systems (e.g., payment) by third party vendors responsible of remotely maintaining IoT devices (HVAC, lighting, etc.) • Physical intrusions into the network • Eavesdropping to obtain information without authorization, or man-in-the-middle (MITM) attack	• Bypass authentication attacks to gain full control of the devices and carry out malicious actions such as injecting malware, spyware, etc. into the building system • Stealthy Deception attacks • Jamming attacks to disrupt the functionality of devices • Rogue devices connecting to the network as legitimate devices	• Vectoring and sniffing attacks for data theft • Distribution of active X scripts by authorized but malicious entities to disrupt the whole system • Malware infiltrated in the building by the infected occupants' devices • Privilege escalation to gain access to occupant's data
Privacy	• Linking data from users to their identities (e.g., surveillance cameras and image recognition can be used to identify users) • Tracking and profiling all users' activities • Misuse of microphones or cameras for invading private spaces without consent	• Indirectly tracking and profiling the activities of a user (e.g., inferring the time of cooking, showering, or watching TV by the power consumption meter)	• Inferring the indoor activities by sensor measurements (e.g., inferring sleeping patterns from the change in temperature or humidity measurements in a bedroom) • Information disclosure	• Profiling users' preferences and behaviours • Inferring users' locations and interests through their location-based queries • Inferring social relationships between building occupants • Privilege abuse of continuous monitoring for malicious activities

of data from illegal disclosure or malicious violation by using an access control mechanism (discretionary access control (DAC), mandatory access control (MaC), role-based access control (RBAC), etc.), confirming one's identity and limiting unauthorized access to the system by providing authentication (elliptic-curve or public key cryptography based techniques), identifying malicious events or jamming attacks (machine learning-based anomaly detection), providing end-to-end secure communication (Zigbee with custom security, MQTT, etc.), preventing user profiling, tracking and re-identification (privacy-preserving data mining, location privacy protection techniques, and statistical disclosure control, etc.).

Although there exists a few solutions to counter some of the aforementioned security and privacy threats, IoTFC addresses a complete set of these threats and integrates lightweight security and privacy solutions that have not been integrated in a single architecture for a smart building. The proposed security and privacy solutions in IoTFC are provided in detail in Sect. 4.3.

3 Communication Technologies

The most essential part of IoT infrastructure is the communication system that acts as a bridge for the delivery of data and control messages between leave stations and a central processing unit (for our case, between a fog node and an end node). Since the amount of information generated over smart building scenario is assumed to be huge and increasing (due to the increase of end devices such as actuators or sensors embedded in home appliances along with the already existing in-building sensors such as smoke alarm and security window sensors, etc.), there is a need to adopt universally accepted, cost effective and scalable communication technology within IoT frame work. In this section, we provide a brief overview of the technologies and protocols that are currently being used within IoT smart building environments.

3.1 Available IoT Smart Building Systems

There are numerous smart building standards available, where some of them are proprietary solutions (e.g., Hubitat, Insteon, etc.), others are partially open standards (e.g. Zwave, EnOceon, etc.), and some of them are open source solutions (e.g., KNX, LonWorks, etc.). However, most of the open source standards support expensive devices as compared to the ones used in conventional installations. Also, these open standards utilize topologies and architectures that are predefined and are difficult to accommodate within fog computing paradigm. Moreover, these solutions are vulnerable to security attacks [1]. In this paper, we intend to utilize the fog computing-based Home Automation System (HAS) proposed in [8], which uses a custom made solution. The authors designed a home gateway based on Raspberry Pi and OpenHAB to provide home automation fog services through local gateways. Though the system is efficient as compared to traditional cloud-based systems, it lacks security and privacy solutions so as to be implemented in a building. We aim to build IoTFC architecture and its security and privacy solution over the aforementioned solution. We intend to improve the system by incorporating Raspberry Pi 4, which includes Bluetooth LE, Wi-Fi, Ethernet and increased computational capacity (required to support the lightweight security and privacy enabling solutions).

3.2 Networking Standards/technologies for Smart Buildings

From technological point of view, an architecture of a smart building can be divided into following four layers: 1) sensor, 2) network, 3) middleware, and 4) application.

Table 2. Comparison of indoor connectivity standards.

	Wi-Fi	ZigBee	Zwave	BLE	NFC
Operation range in building (m)	60	30	30	60	0.1
Maximum Data rate (kbps)	54000	250	100	1000	424
Frequency of operation (GHz)	2.4	2.4	0.86842	2.4	0.01356
Network topology	Star	Star, Mesh	Partial Mesh	Star, Mesh	Peer-to-Peer
IP layer at end devices	Yes	No	No	Yes	No
Security features	AES	AES	AES	AES	AES, RSA
Privacy features	×	×	×	Address Randomization	×

Sensor Layer. This layer detects and collects useful real-world data from the environment (i.e., temperature, humidity, etc.) or things (i.e., motion, vibration, etc.). Moreover, this layer processes information into digital form and then transmits it to the network layer. The main security issues includes DDoS attacks through malicious node placement that can result in battery depletion and physical attack on hardware component through tampering of sensors and devices.

Network Layer. This layer provides the means through which data is transferred among IoT hubs and devices to realize the integration of communication network and the perceptions. It constitutes the communication software and hardware components (i.e., topologies, network nodes, and gateways). Different aspects of these technologies are highlighted in Table 2. In terms of security, this layer is highly susceptible to DoS attacks, confidentiality and privacy attacks through eavesdropping and passive monitoring, MITM attack, illegal access, asynchronous and conspiracy attacks.

Below, we describe the details of the access methods commonly used in smart buildings and highlight the security measures of each to counter different security challenges.

- ***Wi-Fi:*** IEEE 802,11 based Wi-Fi is the most popular wireless technologies that is ubiquitously available at a global scale. The new Wi-Fi standards (IEEE 802.11ah and IEEE 802.11ax) have been particularly designed for IoT use cases. For security, Wi-Fi uses Wireless Equivalent Privacy (WEP) and Wi-Fi Protected Access version 1 or 2 (WPA, or WPA2) protocols. WEP uses a 64 or 128-bit encryption key that enforces confidentiality, access control and data integrity with the goal to protect the privacy of user data from eavesdropping. WEP, due to a brute force attack vulnerability, was superseded by WPA and WPA2. WPA supports Temporal Key Integrity Protocol (TKIP) that operates by performing per-packet key mixing with re-keying and WPA2 uses Advanced Encryption Standard (AES) block cipher TKIP protocol called Counter Mode Cipher Block Chaining MAC Protocol (CCMP) that provides stronger encryption.
- ***Zigbee:*** Zigbee is a low power, low data rate and low cost radio network standard which operated on top of the MAC and PHY layer of IEEE 802.15.4 standard for personal area standard. The Medium Access Control (MAC)

layer of IEEE 802.15.4 defines security services for access control based on MAC address, encryption using AES cryptography, frame integrity through detection and sequential freshness through MAC frame serialization. Apart from the MAC layer security features defined by IEEE 802.15.4 standard, Zigbee has its own security model which includes cryptographic key initiation, key transport, frame protection and device management. Some of the drawbacks are: assigned network key cannot be changed, eavesdropping or ejection of fake packets by adversary, and problem of inter-node coordination.

- ***Z-wave:*** It is on of the oldest low power wireless technology specifically designed for home automation. To enable faster and simpler development of application, Z-wave uses a simpler protocol architecture. The physical and data link layer is included as standard G.9959 by the International Telecommunication Union (ITU). In order to provide protection against authentication, confidentiality and replay attack, Z-wave defines a distinct security layer with two classes of security: Security 0 (S0) for lightweight, and Security 2 (S2) for stronger security. Both the classes provide confidentiality by encrypting information with AES-128. In S0 class, the network key is shared with all devices in a network. While in S2, each new network of subclass prevents a low-security class device from compromising higher-security device. A major security issue for Z-wave protocol is the requirement to support older devices that do not include encryption and authentication procedures. Also, Z-wave has been found to be susceptible to eavesdropping and spoofing attacks. Moreover, implementation of brute force resistant asymmetric cryptography at the end nodes can be challenging due to limited processing power and energy.
- ***Bluetooth Low Energy (BLE):*** Bluetooth, which is based on the IEEE 802.15.1 standard, is a low power, low cost wireless communication technology. BLE was introduced as version 4 of Bluetooth in 2011 and was designed for applications requiring periodic transmission of small amount of data. It provides confidentiality using encryption and uses Cyclic Redundancy Checks (CRC) with hashing and AES encryption to ensure integrity of data. In addition, BLE also aims to improve the availability by providing protection against MITM and DoS attacks through the use of secure connection pairing. To enable privacy, BLE protocol includes a privacy mode which uses random MAC addresses to help achieve anonymity. BLE has shown vulnerability to eavesdropping for nodes that lack display capability to present a six digit number and thus follow the just work association model.
- ***Near Field Communication (NFC):*** NFC, which is breakthrough of the Radio-frequency identification (RFID), is a short range and high frequency P2P (Peer-to-Peer) wireless technology used for wireless identification. It operates by storing information within tiny micro chips (tags) which is transmitted to readers within a certain physical range. In comparison to Bluetooth, NFC does not require pairing before sending the data. Although short-range, NFC technology is vulnerable to many security challenges such as eavesdropping, unauthorized manipulation of data and MITM attacks, which can be countered by using hardware secure elements. The communication between a tag and the reader is performed using AES-128 encryption. The tag is capable

of producing random ID to improve privacy. Authentication is provided using Secure Unique NFC (SUN) procedure.

Middleware Layer IoT Network Technologies. This layer is responsible to provide abstraction to application layer. It receives data from network layer and stores it in a database. Also, it is responsible for ubiquitous computing and information processing. Due to wide the rapid and wide-spread evolution of IoT devices, different application layer protocols have been proposed. In this section, we discuss a few important application protocols that are being used within smart building paradigm.

- ***Message Queue Telemetry Transport (MQTT):*** It is the most popular light machine weight M2M protocol optimized for centralized data collection and analysis. MQTT uses a publish and subscribe architecture and is used for constrained devices operating with low-bandwidth, high-latency, and unreliable networks. It operates on top of TCP and uses the Transport Layer Security (TLS) protocol to provide encryption, authentication, and integrity. For security, each MQTT message contains a variable header with a user name and password for authentication support. The disadvantage of MQTT is the delay caused by TLS and the lack of support for priority messages. On the contrary, MQTT is the most widely used in IoT applications.
- ***Advanced Message Queuing Protocol (AMQP):*** It is a message-oriented light weight middle ware open standard that aims to create an open, asynchronous messaging protocol. Similar to MQTT, AMPQ uses a publish and subscribe architecture and the protocol is built on top of TCP. In terms of security, AMQP does provide higher security mechanisms (including Secure Sockets Layer (SSL) and Kerberos) over the cost of more computational power and resource. Therefore, it is difficult to implement AMQP on IoT devices with limited resources.
- ***Extensible Messaging and Presence Protocol (XMPP):*** It is a an IETF defined protocol developed for near real-time messaging and is based on extensible markup language that helps different entities within a network to communicate. It uses XML as data model and is build on top of TCP. XMPP supports both request/response and publish/subscribe models. The drawbacks of XMPP are: higher processing power and more bandwidth consumption, no QoS guarantee, restriction to simple data and lack of end-to-end encryption.
- ***Constrained Application Protocol (CoAP):*** This is also an IETF defined protocol developed for constrained devices that are capable of connecting to the Internet. It supports a variant of publish and subscribe and request and response architectures. It uses UDP (as opposed to TCP) with poor level of reliability and is assumed be more power efficient than MQTT. In terms of security, CoAP uses a lighter version of TLS. Key management and heavy cost of computation are considered as the main drawbacks of CoAP.

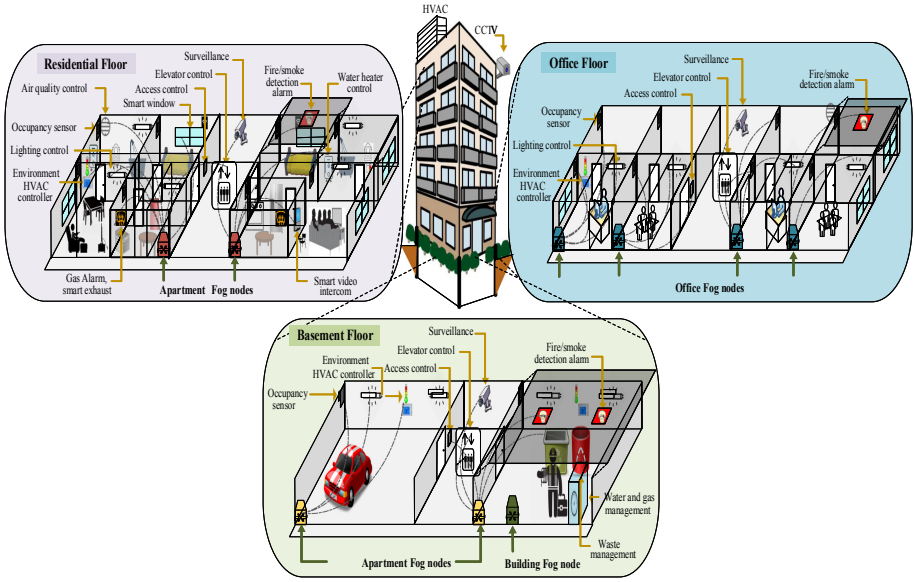

Fig. 1. Smart building architecture.

Application Layer IoT Network Technologies. This layer is used to provide application specific functionalities requested by customers. It provides an interface to lower layer protocols for end users to access data and to communicate with the IoT devices. The application layer typically includes Machine to Machine (M2M) communication protocols, cloud computing, a middle ware and a service support platform. The main security issues related to this layer are data leakage (stealing of data by attackers by knowing the vulnerabilities), malicious code injection (upload malicious codes in software), DDoS attack (to disrupt the availability), inability to receive security patches, and sniffing attack (corruption of the system through the injection of sniffer application).

4 System Architecture

The proposed IoTFC architecture is depicted in Fig. 1. As can be observed, the building is a mixed-use building that includes both offices and residential spaces. The residential portion takes up more square footage than the commercial part. The office floors inhabited by companies consist of open office areas with a few private office rooms, meeting rooms, reception area, open spaces for informal meetings, printing area, storage space, cafeteria, corridors and WCs. In each residential floor, there are 8–10 small to large-sized apartments that may consist of two to four bedrooms, living room, study/library, laundry area, storage space, kitchen, WC(s), terraces and balconies. Also, the building features a fitness centre, a heated indoor swimming pool, smart elevators and a large

reception area. The lowest floor (basement) houses heating stations, parking, electric cars charging points, waste collection points, water tanks and pumps, and technical support area.

4.1 Entities

The proposed smart building architecture consists of the following stakeholders:

- Management staff: Their goal is to operate the building as smoothly and efficiently as possible. It can be further divided into three categories: (1) operational staff who bridge the technology-operations conversations with their technological counterpart; (2) technological staff who manage the network infrastructure and information backbone of the building; and (3) maintenance staff who performs day-to-day activities (e.g. repairs, inspections, etc.) related to electrical, mechanical equipment and civic duties.
- Occupants: These are the true users who utilize advanced technology of the building to achieve their goals of productivity, health, comfort, privacy, and well-being.
- Visitors: A non-resident who may be visiting the building for a brief time.

4.2 Functionality of Layers

The proposed IoTFC model is designed as a four-tier architecture as shown in Fig. 2. A brief functionality of each layer is presented below:

- **IoT Sensor layer:** Following are a few sensors used in the selected four use cases: smart surveillance (night vision security camera), smart maintenance and environment (digital temperature and humidity (DHT11), NFC reader (EZ430), motion detection (HC-SR 501), ultrasonic (HC-SR 04), gas leakage (MQ2), air quality control (MQ135), fire safety), and smart concierge (ultrasonic (HC-SR 04), motion detection (HC-SR 501), light sensing (LDR)).
- **Connectivity layer:** The aforementioned actuator nodes can relay the useful information through any of the communication technologies mentioned in Sect. 3.2. Apart from the default support of Wi-Fi and BLE, the 26 dedicated general-purpose input-output (GPIO) pins available in Raspberry Pi 4 can be used to connect a ZigBee coordinator (XeBee module). Also, Z-wave coordinator can be connected through the available USB port. Due to the comprehensive security and privacy features, in this work, we aim to utilize BLE. For this purpose, we utilize ESP32 wireless micro controller developed by Espressif, which contains an integrated Wi-Fi/BLE chip-set, RF core, amplifier, power management module and a builtin antenna. ESP32 can be programmed using Arduino IDE and can establish a secure connection to a MQTT broker.
- **Fog layer:** In IoTFC (Fig. 1), fog nodes are deployed at each apartment and office areas of the building closed to IoT devices for an efficient real-time processing and data analytics at the edges of the network. Each of these fog node will be connected to the sensors monitoring temperature, humidity,

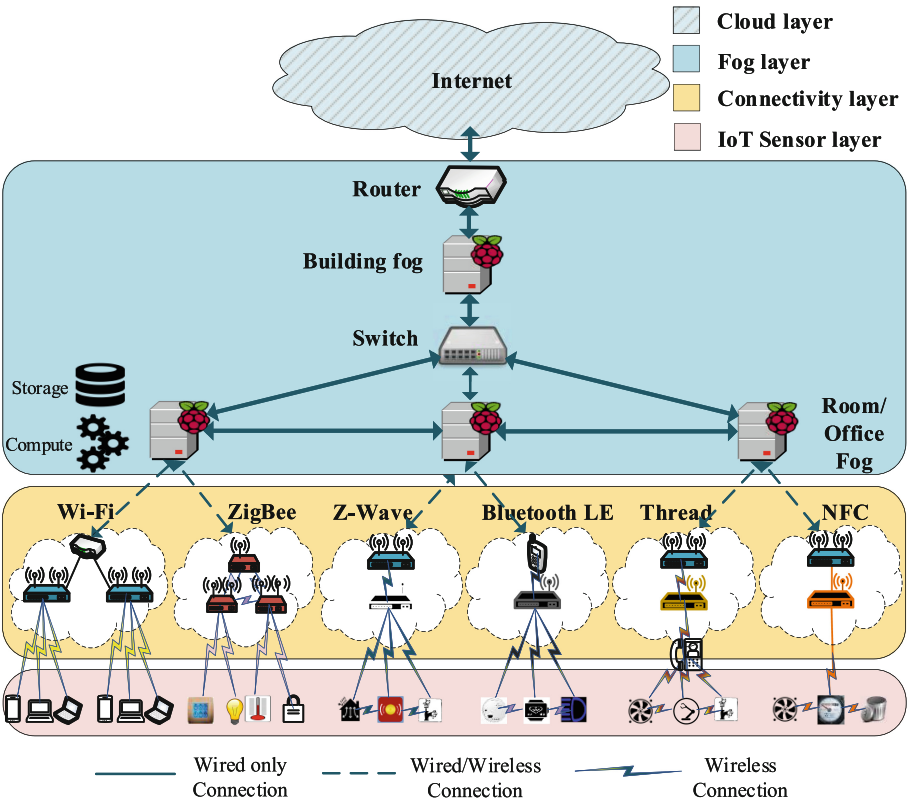

Fig. 2. IoTFC Smart building architecture.

air quality, occupancy, energy usage, and numerous other factors via Wi-Fi, Bluetooth or Zigbee. The fog nodes will have enough analytical power to perform real-time actions such as conserving plug load and HVAC usage when any room is unoccupied, to learn an occupant's preferred temperature and maintain the space at that comfort level, to perform access control, other security (cryptographic) operations, fault tolerance of a room, and privacy-preserving operations, and provide storage (for 1–2 h), and to make an escape route in an emergency situation. By deploying these services in the fog layer, lower latency and improved QoS can be achieved to deliver outstanding user experience. These nodes will be associated with the building's connectivity access infrastructure (wired or wireless), and will be able to perform continuous security scan of all devices connected to them. A building fog node communicates with all the apartment/office area fog nodes and takes slower, more deliberate actions, such as to set equipment maintenance schedules, optimize load balancing (also for connectivity layer) and fault tolerance of floor, move applications around if any apartment/office area fog node fails or

becomes overloaded, and share with the cloud any information that requires non-critical decision making or very high storage space in tens of terabytes. The apartment/office area fog nodes and the building fog node is assumed to be Raspberry Pi 4, which runs a Linux distribution called Raspbian and utilizes OpenHAB (popular IoT integration platform) with additional bindings (to extend the functionality for parallel integration of different systems and devices) so as to create a distributed fog environment. Every fog node is assigned a static IP address and is connected to a common Ethernet switch, resulting in formation of a network. The building fog node will act as a master node connected to cloud and will be connected to the Internet via USB-Ethernet adaptor. Data from the IoT devices to the cloud are passed through the respective fog nodes using MQTT broker. It is pertinent to highlight here that the security and privacy solutions proposed in this paper can be added as additional bindings to OpenHAB.

– **Cloud layer:** Data from the fog layer that is less time sensitive is sent to the cloud for long-term storage, historical analysis, and big data analytics to determine operational aspects of the building. Over period of time, the cloud performs analysis on the received data from fog nodes to gain business insight, and based on these insights can send new or update application rules to the fog nodes to further optimize building operations.

4.3 Security and Privacy Solutions

There are mainly four kinds of data services in fog computing: data storage, data sharing, data query, and data computation. All these services demand different unique data security (access control, authorization revocation, availability, confidentiality of inputs, outputs and computing tasks) and privacy (identity, query, location, and preventing tracking and profiling) requirements. In IoTFC architecture, we propose light-weight solutions to mitigate the security and privacy issues mentioned in Table 1 for the four use cases.

Security in IoTFC. Three-tier light-weight security model is proposed in the following:

Access Control: In multi-users fog computing environment such as a smart building, permissibility of certain network or data resources must be allowed to only those users or devices, who possess certain rights to use the requested resources. The existing access control mechanisms (mentioned in Sect. 2) offer some advantages and a few limitations, e.g., RBAC model is more scalable than the DAC and MaC models. However, in RBAC, roles are assigned statically, whereas, in order to fulfill fog access control requirements (latency, efficiency, resource restriction, policy management), RBAC model can be evolved to support dynamic environment since the user's access privileges not only depend on "who the user is" but also on "where the user is" and "what is the user's state and the state of the user's environment". In IoTFC, we propose a dynamic RBAC

(dRBAC) model in IoTFC architecture that dynamically grants and adapts permissions to users according to context. Since the fog devices are resourceful and are used closed to IoT devices, therefore, there is no need to deploy access control mechanism at IoT sensors layer.

dRBAC consist of the following elements: 1) subject: an entity such as user, fog node, etc. that accesses objects; 2) object: any computing or data resources; 3) roles: an entity associated with specific authority and responsibility, e.g., building management, occupants, visitors, etc.; 4) permissions: operations such as read, write, delete, etc. that bind objects and a set of actions that can be executed by a subject; 5) session: a set of interactions between subjects and objects; 6) context agent: an entity that collects context information such as domain, hierarchical role (mapping between inheritance relationships and roles, e.g., operational staff inherits the permissions of maintenance staff), and location; 7) user assignment list (UA): mapping between roles and subjects; 8) permission assignment list (PA): mapping between permissions and roles, and 9) conflict-of-interest (COI): a list mapping conflicting and non-conflicting permissions to roles.

In dRBAC, a policy maker center (PMC) is deployed at the building fog node that creates and maintains UA, PA and COI. The intermediate fog nodes (apartment/office) act as sub-coordinators and perform part of the policy decision-making tasks delegated by the building fog node. To successfully access building services or objects, a subject initially sends access right request to the PMC to get an access token. Given registered subject information established in profile database, the PMC evaluates the access request by enforcing defined authorization policy rules. If the access request is granted, the access token encapsulating access right is generated with issuer's signature and sent back to the subject. Otherwise, the access right request is rejected. Then, the subject can request access to other services or objects by presenting a valid access token to the service provider.

For each query, the coordinator or sub-coordinators would perform the following evaluation:

$$\begin{aligned} &User(?user) \wedge Role(?role) \wedge Permission(?action) \wedge \\ &Session(?activeorinactive) \wedge Context(?attributes) \wedge \\ &Conflict(?action) \wedge Accessdecision(?decision) \end{aligned} \tag{1}$$

For example, the subject (building resident) using his/her access token looks for an available meeting room by checking the meeting room schedule (that gets updated from the occupancy sensor). The coordinator of the office area verifies the access right policies and conditional constraints against the provided access token. If satisfied, it grants the access request to the resident, who can book the available time-slot of the meeting room. The building resident can update the schedule, but he/she is not allowed to delete or change any other information in the schedule such as deleting other booked time-slots.

Authentication: As services in IoTFC are offered to a large number of building users by fog nodes, authentication becomes a critical issue. Fog nodes need to

authenticate at different levels so as to ensure security. Also, in an IoT multi-user fog environment, the IoT sensors layer has two types of devices, mobile and fixed IoT devices. The mobile IoT devices are carried by their owners (e.g., smart phones, tablet, NFC tags etc.), while the fixed IoT devices (e.g., occupancy sensors, NFC reader, etc.) are pre-deployed in specific areas to provide services. This mobility of users creates a problem in designing identity authentication with minimal latency. Hence, in IoTFC architecture, we need to design an identity control mechanism at three layers (IoT sensors, fog, cloud) with minimal latency.

The device level authentication is performed by running MQTT broker (open source Mosquitto) on the apartment/office fog node, which is a Raspberry Pi 4 and thus can act as both publisher and subscriber. A token authentication (TA) server is assumed to be deployed in IoTFC, which is responsible to issue a token and check its validity against any authentication request. The device first sends its credential i.e., valid ID (universal unique identifier (UUID) or a MAC address), and username to TA server for obtaining a token. Once the authentication server checks the requested credentials against its database, it then returns a valid token containing the header, payload and signature. Upon receiving the token, the device sends the "CONNECT" message to the broker by providing the access token as its username and the password. The broker can use the token to perform various validations, such as: check the validity of the signature, check the expiry date of the token, and check the token authentication server to see whether the token was revoked. If token is valid, the device is then allowed to publish the sensor data. MQTT payload encryption is performed between the publisher and the broker for secure communication.

In order to perform fog node authentication, first it needs to be registered in IoTFC. To perform registration, TA chooses a unique identity (ID_{FDi}) for each fog device (FD_i), and then calculates its pseudo identity $PID_{FDi} = H(K||ID_{FDi})$ using the its own secret key K, and generates corresponding temporary credentials $TC_{FDi} = H(K||TS||ID_{FDi}||nonce)$, where TS is the timestamp. TA stores these credentials in FD_i's database before it is deployed. Similar procedure is deployed for cloud server registration. Once registered, the fog nodes can communicate with other fog nodes or cloud through a secure key management protocol based on these temporary credentials to establish a secret key for secure communication.

Anomaly Detection: In order to identify events that appear to be anomalous in nature with respect to the normal behavior, IoTFC will utilize anomaly based intrusion detection system (AIDS) and network based intrusion detection system (NIDS). All sensor devices will be analyzed by AIDS of respective fog nodes (i.e., anomaly detection operates in a distributed manner). In the event of an unusual event, an alarm is sent by a fog node to the building fog node and to the IoT sensors. AIDS, operating at apartment/office fog node, comprises of a learning module (contains traffic patterns observed dynamically, such as, request count, failed authentication count, device usage at different time periods, bandwidth consumed, etc. and a known behavior models for different attacks, such as DDoS, TCP flooding, etc.), Classification module (Support Vector Machine (SVM) to

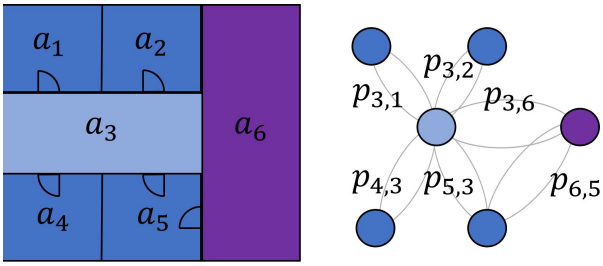

Fig. 3. Swap Areas and Markov Chain obtained from their measurements.

classify and detect anomaly) and flow management (methods to block a flow in the event of true detection and blacklisting of the node). NIDS, implemented at the building fog node, will monitor the entire network traffic and can centrally detect a malicious activity by using a light-weight SVM algorithm. The overall hierarchical approach will allow detection of both malicious end nodes as well as a malicious fog node. In addition, this procedure will ensure no access to a node black listed within the network.

Privacy in IoTFC. To guarantee privacy in case data stored for smart surveillance, "pixelating technique" will be used to automatically obscure the identities of persons, while all of their movements and actions remain recognizable. This approach provides important information to support decisions, appropriately plan interventions, and minimize damage and costs; all without violating the privacy of individuals.

To protect device level data, we propose to aggregate room sensors' measurements, called *Area nodes* $(a_1, a_2, \ldots, a_6)$ as shown in Fig. 3, that will act as Mix Zones [2]. Additional privacy can be provided by increasing the size of the Mix Zones by joining adjacent locations (e.g., aggregating a_1 and a_2 nodes) in Fig. 3 to obtain a super node a_{12}, which may also guarantee k-anonymity. With IoTFC, we also provide privacy for location-based services, users are protected from re-identification by the servers and from inference of their interests and habits from the location information contained in their queries [13]. We adopt a solution from [12] that is suited for IoTFC, in which the operation is performed by a component called *SwapMob*, which is integrated to the area nodes. We also consider privacy profiles for each of the user roles used for role-based access control. To comply with consent acquisition (as required by GDPR), we modify the SwapMob collection mechanism to a role-based swapping in which the fog nodes will only swap the currentIDs depending some pre-specified rules on the roles.

For example, in a smart building, a technician that comes to the building for maintenance will not expect to have the same privacy as an inhabitant or permanent worker from the building, however if there is a team of technicians all of them would be expected to have the same level of privacy. Therefore, in this

case the Fog node may swap their currentIDs but only among the technicians or inhabitants, not between a inhabitant and a technician.

The SwapMob mechanism will carry out the privacy protection for all mobile IoT devices. It preserves the sufficient statistics for discrete time Markov chains specified by time interval τ and spatial resolution χ, hence will preserve the Global Markov Model [4]. It will be used for predicting locations, and analyzing collective patterns of mobility inside the building. All of these tasks will be performed without the need to know the specific locations for particular individuals. In Fig. 3, the global mobility Markov chain is represented as a directed graph in which $p_{i,j}$ denotes the probability of a device that is in area a_i to move to area a_j.

5 Comparative Analysis

This section carries out a comparative analysis of the proposed architecture with the relevant state-of-the-art smart building solutions. The comparison, summarized in Table 3, considers the following properties: access control, authentication, anomaly detection, data and location privacy, and middleware.

Table 3. Comparison of IoTFC with the related state-of-the-art smart building solutions.

Properties	Lilis et al. [10]	Ramos et al. [9]	Wissam et al. [14]	Pappachan et al. [11]	Ferrandez et al. [7]	Boyer [3]	IoTFC
Access control	✓	✓	✓	×	×	×	✓
Authentication	✓	✓	✓	×	×	×	✓
Anomaly detection	✓	×	×	×	×	×	✓
Data privacy	×	✓	✓	✓	✓	✓	✓
Location privacy	×	×	×	✓	×	✓	✓
Middleware	✓	✓	✓	×	✓	×	✓

Table 3 shows that IoTFC offers a secure and privacy-preserving middleware architecture. In comparison, the architecture in [10] offers security properties but fails to provide data and location privacy. The secure middleware architectures proposed in [9,14] offer access control, authentication and data privacy, while fail to provide anomaly detection and location privacy. The systems proposed in [11] and [3] fail to provide the guaranteed security properties but offer both data and location privacy. In [7], the authors have proposed a middleware architecture that ensures data privacy while fail to provide any other security property.

6 Conclusions

In this paper, we have proposed IoTFC, a novel secure and privacy preserving architecture for smart building based on fog computing. This paper is a first

step towards implementing both security and privacy by design in a smart city use case. Most of the available smart building solutions are proprietary based, but IoTFC has been designed with open source communication technologies and standards to provide interoperability. Light weight access control and authentication protocols have been proposed to provide resilience against known security IoT attacks. The privacy solution provides means of protecting occupants from profiling and tracking. In future, we aim to provide a complexity, security and privacy analysis of IoTFC.

References

1. Antonini, A., Maggi, F., Zanero, S.: A practical attack against a KNX-based building automation system. In: ICS and SCADA Cyber Security Research, pp. 53–60 (2014)
2. Beresford, A.R., Stajano, F.: Location privacy in pervasive computing. IEEE Pervasive Comput. **2**(1), 46–55 (2003)
3. Boyer, J.P., Tan, K., Gunter, C.A.: Privacy sensitive location information systems in smart buildings. In: Clark, J.A., Paige, R.F., Polack, F.A.C., Brooke, P.J. (eds.) SPC 2006. LNCS, vol. 3934, pp. 149–164. Springer, Heidelberg (2006). https://doi.org/10.1007/11734666_12
4. Chen, M., Liu, Y., Yu, X.: NLPMM: a next location predictor with Markov modeling. In: Tseng, V.S., Ho, T.B., Zhou, Z.-H., Chen, A.L.P., Kao, H.-Y. (eds.) PAKDD 2014. LNCS (LNAI), vol. 8444, pp. 186–197. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06605-9_16
5. Costante, E.: Gsmart buildings: trends and challenges for a secure future (2017). https://www.forescout.com/company/blog/smart-buildings-trends-and-challenges-for-a-secure-future/
6. Faro, C.: Nearly four in ten smart buildings targeted by malicious attacks in h1 (2019). https://usa.kaspersky.com/about/press-releases/2019_smart-buildings-threat-landscape
7. Ferrández-Pastor, F.J., Mora, H., Jimeno-Morenilla, A., Volckaert, B.: Deployment of IoT edge and fog computing technologies to develop smart building services. Sustainability **10**(11), 3832 (2018)
8. 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 **18**(8), 2660 (2018)
9. Hernández-Ramos, J.L., Moreno, M.V., Bernabé, J.B., Carrillo, D.G., Skarmeta, A.F.: SAFIR: secure access framework for IoT-enabled services on smart buildings. J. Comput. Syst. Sci. **81**(8), 1452–1463 (2015)
10. Lilis, G., Kayal, M.: A secure and distributed message oriented middleware for smart building applications. Autom. Constr. **86**, 163–175 (2018)
11. Pappachan, P., et al.: Towards privacy-aware smart buildings: capturing, communicating, and enforcing privacy policies and preferences. In: IEEE ICDCSW, pp. 193–198 (2017)
12. Salas, J., Megías, D., Torra, V.: SwapMob: swapping trajectories for mobility anonymization. In: Privacy in Statistical Databases, pp. 331–346 (2018)
13. Shin, K.G., Ju, X., Chen, Z., Hu, X.: Privacy protection for users of location-based services. IEEE Wirel. Commun. **19**(1), 30–39 (2012)

14. Wissam, R., Daniele, S., Kouichi, S.: A new security middleware architecture based on fog computing and cloud to support IoT constrained devices. In: IML 2017, pp. 1–8 (2017)
15. Ziegeldorf, J.H., Morchon, O.G., Wehrle, K.: Privacy in the Internet of Things: threats and challenges. Secur. Commun. Netw. **7**(12), 2728–2742 (2014)