IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 November 2023, accepted 21 January 2024, date of publication 29 January 2024, date of current version 7 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3359442

RESEARCH ARTICLE

A Problem Analysis of Smart Home Automation: Toward Secure and Usable Communication-Based Authorization

SIOK WAH TAY[1,2], NING ZHANG[2], AND SALEM ALJANAH[3,2]
[1]Faculty of Information Science and Technology (FIST), Multimedia University, Bukit Beruang, Melaka 75450, Malaysia
[2]Department of Computer Science, The University of Manchester, M13 9PL Manchester, U.K.
[3]College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11432, Saudi Arabia

Corresponding author: Siok Wah Tay (swtay@mmu.edu.my)

This work was supported by The University of Manchester.

ABSTRACT The advent of the Internet of Things (IoT) and Artificial Intelligence (AI) have led to the rising popularity of Smart Home Automation (SHAuto). SHAuto uses a variety of interconnected smart devices to provide life-enhancing services such as smart energy control, smart entertainment, smart healthcare, and so on. If these devices are compromised, sensitive data may be disclosed and the compromised devices or other connected devices may be maliciously controlled, threatening the privacy and safety of home occupants. Therefore, controlling access to devices in SHAuto is of paramount importance. However, due to the characteristics of the SHAuto environment, this has become a challenging issue. As a first step towards addressing this challenging issue, this paper provides a comprehensive problem analysis of SHAuto. The problem analysis consists of two parts. The first part is an in-depth analysis of various SHAuto use case scenarios covering three aspects, i.e., device control modes, automation modes, and device communications. This analysis has led to the formulation of a generic model for SHAuto. Based on this model, the second part analyses potential vulnerabilities and threats in relation to authorisation. The comprehensive problem analysis has led to a hypothesis that access to the devices can be controlled by governing device communications and the specification of a set of requirements for the design of secure and usable communication-based access control solutions for SHAuto environments.

INDEX TERMS Access control, authorisation, Internet of Things (IoT), smart home automation.

I. INTRODUCTION

The IoT can be described as a network of connected devices and the technology that enables communication and data exchange between the devices [1], [2]. Smart Home or Smart Home Automation (SHAuto) is one of the most popular IoT applications. SHAuto can be referred to as the utilisation of technology within the home environment to offer convenience, comfort, security, and energy efficiency to the home occupants [3]. The complexity of SHAuto systems arises from inter-connected devices and their integration with the Internet. This connectivity not only promises enhanced functionality and convenience but also introduces risks.

In SHAuto systems, a range of heterogeneous smart devices are deployed to automate a variety of home services such as lighting and HVAC (heating, ventilation, and air conditioning) control, entertainment, home safety and security, and healthcare and wellbeing, and so on. These devices produce and/or consume diverse data including sensitive data, e.g., health data. They may also perform safety-critical operations such as fire detection or door locking and unlocking. These devices are commonly equipped with Internet connectivity, interconnected with other devices, and capable of communication among themselves. They can be

The associate editor coordinating the review of this manuscript and approving it for publication was Eyhab Al-Masri.

monitored and controlled manually and/or automatically. In the latter case, the control may be done locally and/or remotely via third-party services such as Cloud services. If these devices are compromised, unauthorised parties may gain access to sensitive data produced or consumed by the devices.

Internet-connected devices, such as household appliances, could also facilitate unauthorised surveillance or invasions of privacy [4]. For example, criminals could track homecomings and departures of occupants by exploiting an Internet-connected door lock. The devices may also be manipulated to perform unauthorised operations or to gain control over other connected devices. For example, a compromised device may publish commands to maliciously control kitchen appliances (e.g., ovens, stoves, and toasters), causing them to malfunction, or even worse, causing a house fire. A compromised device may not only pose threats to the home occupants' security and privacy but also cause physical harm to the occupants, putting their safety at risk. While security is essential to safeguarding an SHAuto system, safety is equally paramount. Ensuring the security of an SHAuto system is vital to guarantee its overall safety. In essence, security is about keeping the system safe, and safety is about keeping the occupants safe. It is, therefore, crucial to restrict how devices can be accessed and controlled to preserve security and safety in an SHAuto environment. This is where access control comes into play. Access control encompasses authentication and authorisation.

Authorisation is about granting or denying access privileges on system resources to subjects [5], [6]. While efforts have been made to improve authorisation in SHAuto environments, to the best of our knowledge, there is only a limited amount of work (e.g., [7], [8], [9], [10], [11], [12], [13]) that has taken into account device-to-device communications. As a first step towards achieving secure and usable communication-based authorisation in SHAuto environments, in this paper, we have conducted an in-depth problem analysis of SHAuto, focusing on device-to-device communications. More specifically, the contributions of this paper are summarised as follows.

- A comprehensive use-case analysis of SHAuto covering three aspects: device control modes, automation modes, and communications among entities. The entities refer to a set of interconnected devices, services, and applications (apps) that collectively offer SHAuto services.
- The formulation of a generic SHAuto model which captures different SHAuto use-case scenarios.
- A threat analysis that identifies potential vulnerabilities and authorisation-related threats in an SHAuto environment.
- The specification of a set of functional, security, usability, and performance requirements for the design of secure, usable, and efficient communication-based access control solutions for SHAuto environments. The specification of the requirements has taken into account the characteristics of SHAuto and the findings from the problem analysis conducted above.

The rest of the paper is structured as follows. Section II reviews the existing security analyses of smart homes. Section III describes the characteristics, architecture, and communication model of SHAuto. Section IV presents the analysis of the SHAuto scenarios. Section V describes the generic SHAuto model. Section VI performs the threat analysis based on the model. Section VII specifies the set of requirements. Finally, Section VIII concludes the paper.

II. RELATED WORK

In recent years, several studies have examined security risks in smart homes. In a study conducted by Fernandes et al. [14], an empirical security analysis was performed on Samsung SmartThings [15], a popular smart home platform. The analysis identified framework design flaws in two domains: the SmartThings permission/capability model and the event subsystem. The authors also demonstrated how these design flaws can be exploited by attackers to steal lock pin-codes, disable a vacation mode SmartApp, and cause fake fire alarms.

Geneiatakis et al. [16] conducted a security and privacy threat analysis on a smart home architecture, with a focus on the interactions among various off-the-shelf IoT devices. However, the architecture only considered scenarios where the IoT devices, organised in islands, were connected to a hub, and were not directly accessed by other devices. Ali et al. [17] used different scenarios to investigate security attacks which violate the security goals including confidentiality, integrity, availability and so on. Ray and Bagwari [18] presented a set of security threats in existing IoT communication protocols (e.g., Bluetooth, ZigBee, WiFi, LoRaWAN) and proposed a secure and cost-effective communication protocol with desirable attributes for smart homes. In a more recent study, Girish et al. [19] conducted an analysis of local network communications involving a broad spectrum of consumer smart IoT devices and mobile apps. The study identified security and privacy threats associated with local network traffic in smart homes. In another study, Li et al. [20] introduced ZPA, a system designed to investigate privacy and security issues in ZigBee-based smart home networks, utilising ZigBee-encrypted traffic. ZPA employs state-of-the-art machine-learning models to identify smart home devices' type and status that could potentially leak users' private information.

Meng et al. [21] surveyed security challenges in smart homes based on different attacking interfaces such as the physical layer, network layer, mobile applications, access control, and voice user interface. They also discussed the existing proposed countermeasures to these challenges. Kavallieratos et al. [22] performed a threat analysis on the smart home ecosystem using the STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege) threat analysis method

and Microsoft's threat modeling tool. The analysis focused on threats in relation to the physical components of the ecosystem and information transmitted between these components.

Different from the reviewed work discussed above, the risk analysis conducted on a smart home automation system in [23] took into account the human factor. This analysis encompassed six components of the system: connected sensors/devices, in-house gateway, cloud server, API, mobile device, and smartphone apps. Each of these components was analysed to identify vulnerabilities and threats related to hardware, software, information, network communication, and human aspects. Similar to [23], the study conducted by [24] took into account human-related attacks such as social engineering attacks. This study explored security challenges in smart homes based on the four IoT layers, i.e., application layer, perception layer, physical layer, and network layer.

Many of the reviewed work focused on specific aspects, such as established smart home platforms [14], specific device interaction scenarios [16], particular smart home applications [17], IoT communication protocols [18], [20], and the local home network [19]. In contrast, broader aspects were explored in the studies [21], [22], [23], [24], encompassing different IoT layers and the human factor. However, threats associated with authorisation were not extensively investigated in these studies. Unlike the reviewed work, our analysis took a comprehensive approach to investigating authorisation-related threats. We formulated a generic model of smart home automation that encompasses a wide range of use case scenarios, including device control modes, automation modes, and device communications.

III. SHAUTO CHARACTERISTICS, ARCHITECTURE AND COMMUNICATION MODEL

This section presents the characteristics, architecture and communication model of SHAuto.

A. THE CHARACTERISTICS

SHAuto has a number of characteristics which make governing access to devices a challenging issue. These characteristics include device heterogeneity, dynamic nature of SHAuto, varying access requirements, and access purposes.

Device Heterogeneity: Smart devices may come from different vendors and have different functionalities, energy capacities, processing and communication capabilities. For example, in an SHAuto system, some devices support cooking functions while others support lighting control, and so on. From a security perspective, devices can differ in sensitivity levels. For instance, devices that produce health-related data and/or perform safety-critical operations typically have a higher sensitivity level. In addition, some devices may have shared ownership [25]. For instance, in homes with multiple occupancies, a single occupant may own a subset of devices and some devices may be owned by multiple occupants.

Dynamic System: An SHAuto system is typically a dynamic system where context changes are expected. New devices may be added by the homeowners gradually and obsolete devices may be removed when no longer needed. Additionally, devices may be added to or removed from the system in an ad hoc manner. They may also be powered off or disconnected from the home network from time to time [26]. Furthermore, devices may experience a context change, e.g., a device relocation.

Varying and Changing Access Requirements: Depending on various factors, access requirements may vary across different SHAuto systems and these requirements may change. For example, SHAuto systems may differ in terms of household sizes, the number of devices, and so on, thus imposing different access requirements. For example, in an SHAuto system with multiple occupants, access to devices could be based on device ownership. However, this requirement does not apply to an SHAuto system with only a single occupant. In addition, the requirements may change from time to time due to context changes, e.g., a device relocation may cause a change in the role of the device, hence changing the requirements under which it can be accessed.

Different Access Purposes: In an SHAuto system, a device may be accessed for data access, for controlling other devices, or for both of these purposes.

Due to the diversified access requirements and the dynamic nature of SHAuto, SHAuto should support fine-grained and context-aware access control that makes access decisions based on device context. With the increased granularity and the incorporation of device context into decision making, managing authorisation in an SHAuto environment has become complex and challenging. Unlike industrial IoT applications where access control is often managed by dedicated security professionals, in SHAuto, access control administration generally relies on homeowners who typically lack security knowledge or expertise, and/or may not spend time to configure the system or specify access policies or assign access privileges adequately [25]. For example, assigning privileges for every single device can be a complex and time-consuming process, requiring excessive user effort. The process is also prone to human errors. All these factors can prevent homeowners from implementing access control properly. Additionally, in a broader sense, human users' decisions to grant permissions to technology elements (e.g., apps) or adopt these elements may also be affected by their risk perception [27], [28], which could be influenced by the ease or difficulty of retrieving relevant concerns for the decisions [29].

B. THE ARCHITECTURE

The SHAuto architecture used in our problem analysis is derived based on the architectures of both the IoT and Smart Home published by previous studies [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42]. The architecture, as shown in FIGURE 1, consists of four layers: device layer, gateway later, cloud layer, and application layer.

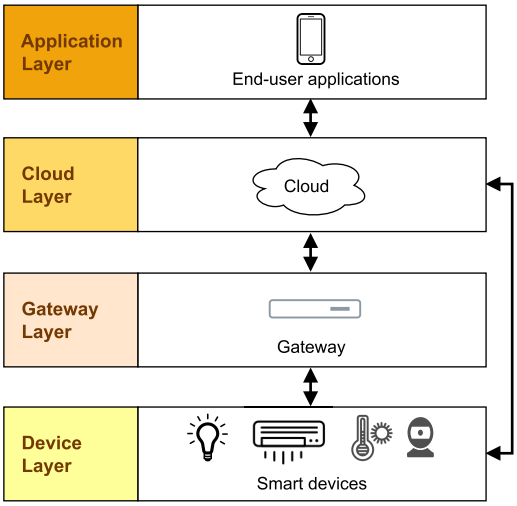

FIGURE 1. A general architecture of SHAuto.

Device Layer: This layer is also referred to as the object layer [32], the perception layer [39], [40], [41], [42], or the physical layer [38], in the literature. Its main tasks are to collect data and receive commands [33]. These tasks are accomplished by a myriad of interconnected smart devices, including smart appliances and IoT devices equipped with communication and sensing and/or actuating capabilities [34], [35], [43]. As described in Section III-A, these devices can be heterogenous in terms of functionalities, processing and communication capabilities, and so on.

Gateway Layer: This layer is responsible for mediating communications between heterogeneous SHAuto devices. It also serves as a gateway between a smart home and the cloud. The gateway function is typically provided by connecting the devices to a central device, i.e., the SHAuto gateway. The SHAuto gateway, also known as the SHAuto hub, is located in the local home network [35]. It enables interconnections between devices from various vendors or between devices that use different communication protocols such as WiFi, ZigBee and Bluetooth. The gateway is connected to the Internet, thus the cloud, via a home router [16]. It connects to the cloud on behalf of devices that are not WiFi-enabled. [44] classifies non-WiFi-enabled devices and WiFi-enabled devices, respectively, as hub-connected devices (known as gateway-connected devices hereafter in this paper) and cloud-connected devices. In addition to the provision of connectivity, the gateway also provides services to support SHAuto, some of which may include device control and management, data processing, storage and analytics, and so on.

Cloud Layer: This layer is represented by the provider cloud. It provides support for most of the services of the gateway layer [32]. Most of the smart home platforms today are cloud-based [45], i.e., the core services supporting smart homes are mainly provided in the cloud layer, instead of the gateway layer. Examples of such platforms include Samsung SmartThings, Amazon AWS (Amazon Web Services) IoT [46], and IBM Watson IoT [47]. The provider cloud may collaborate with third-party clouds [32] or integrate with third-party services to offer richer features. For example, SmartThings allows users to download and install SmartThings apps (or SmartApps), written by third-party developers [48], into their SmartThings cloud account [13] to automate a home collaboratively. In addition, some smart home platforms (e.g., SmartThings) also support user-defined automation rules provided by third-party platforms (e.g., IFTTT) [49]. IFTTT ("If This, Then That") [50] is a Trigger-Action platform that allows users to customise automation rules. Users may create their own Applets (or automation rules) [51] or adopt published Applets written by third-party developers [52].

Application Layer: This layer provides home users with an interface to access SHAuto services. For example, the users can remotely control devices, manage devices and their communications [32], and customise automation rules [44], [49] through end-user applications.

In SHAuto environments, artificial intelligence could be employed to achieve automated personalised decision making in a non-intrusive manner [53]. Depending on the specific SHAuto system being used, the automated decision-making process may occur at the device, cloud or gateway layer of the SHAuto architecture. A more detailed discussion of automated decision making is presented in Section IV.

C. THE COMMUNICATION MODEL

The communication model used in SHAuto systems is typically an event-driven communication model [14], [54], in which automations are triggered by events. This communication model allows an SHAuto system to act on events as they occur. An event is a change that takes place in an SHAuto system. It could be a sensor reading that exceeds a threshold, or a user action such as pressing a toggle button on a mobile app. We call the former a sensor-generated event and the latter a user-generated event.

An event-driven SHAuto system consists of event producers and event consumers. An event producer is an entity that detects an event and generates a message to represent the event. This message contains data or a command. The event producer then transmits the message to the event consumer(s). An event consumer is an entity that executes an action in response to a received message. Examples of actions include the control of a device (e.g., switching on or off a device) and the generation of another event.

IV. SHAUTO SCENARIOS

Through synthesising the literature [13], [15], [16], [34], [35], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [55], [56], [57], [58], [59], [60], and [61], we have identified a number of SHAuto scenarios. In this section, we analyse

these scenarios from three aspects: device control modes, automation modes, and entity communications.

We have chosen these aspects for this analysis because of the following reasons. Firstly, as stated in Section I, unauthorised control of devices can lead to safety concerns. Therefore, in order to protect an SHAuto system from such unauthorised control, it is necessary to anaylse the ways in which devices can be controlled. Secondly, as an SHAuto system comprises interconnected devices, it is crucial to explore how these devices interact with one another. This can be accomplished through an investigation of automation modes and entity communications. The automation modes facilitate the investigation of how the devices may interact within or across different layers of the SHAuto architecture. Entity communications delve deeper into potential communications among the entities based on factors such as entity automation roles and communication levels and patterns.

A. DEVICE CONTROL MODES

A device control mode determines how a device is, or should be, controlled. As discussed in Section III-C, there are two types of events, user-generated events and sensor-generated events. Accordingly, device controls can also be classified into two modes, a manual mode and an automatic (auto) mode. A device controlled via a user-generated event is said to be in the manual mode, whereas a device controlled via a sensor-generated event is in the auto mode.

1) MANUAL CONTROL MODE

In the manual control mode, device control actions are triggered by user-generated events. Current SHAuto platforms generally allow users to control home devices via input devices [61]. For example, devices can be controlled via applications (apps) running on end-user devices such as smartphones, tablets, personal computers, and so on [34]. These end-user apps are typically the companion app for a device or an SHAuto platform offered by the provider. Through these apps, users can issue a control command to control devices locally from within the home network, or remotely over the Internet when they are away from home.

The manual device control can be explained using a lighting control use case. FIGURE 2 illustrates a high-level automation workflow of the manual control scenario of the use case. The workflow involves three steps: (1) a homeowner launches a mobile app and changes the state of a toggle button, which controls a light bulb in a room of the home, from *off* to *on*; (2) when this state change of the button (an event) occurs, the app (an event producer) sends a message containing a *switch-on* command to the corresponding light bulb (an event consumer); (3) the light bulb then switches itself on (an action) in response to the command.

2) AUTO CONTROL MODE

In the auto control mode, device control actions are triggered by sensor-generated events (e.g., sensor outputs) without any human intervention (e.g., a user command). In other words, devices are automatically controlled by the SHAuto system which makes automated decisions in response to sensor-generated events.

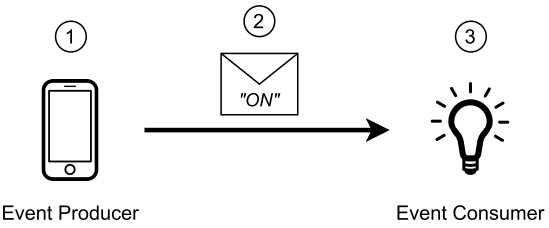

FIGURE 2. Automation workflow of manual lighting control use case scenario.

A similar lighting control use case can be used to explain this mode. FIGURE 3 shows a high-level automation workflow of the auto control scenario of the use case. It can be seen that more steps are involved in this mode in comparison to the manual control mode. The steps are: (1) a motion sensor detects a motion in the room (an event); (2) the motion sensor (an event producer) sends a message containing some sensed data to a smart lighting control service (an event consumer); (3) the lighting control service (an event producer) processes the message and makes a decision to switch on the light bulb (an event); (4) the service sends a message containing a *switch-on* command to the light bulb (an event consumer); (5) the light bulb switches itself on (an action) in response to the command.

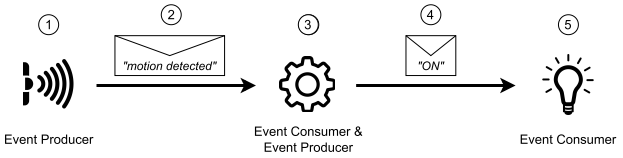

FIGURE 3. Automation workflow of auto lighting control use case scenario.

From the scenarios discussed in the two device control modes above, two observations can be made. First, there are two types of communications in SHAuto, i.e., data and command communications, and an automation task may involve one (e.g., use case scenario in Section IV-A1) or more (e.g., use case scenario in Section IV-A2) types of communications. The communication type is determined based on the content (data or command) of the message transmitted. As such, event producers and event consumers can be classified into four communication roles, i.e., data producer, command producer, data consumer, and command consumer. Take the auto lighting control use case scenario in Section IV-A2 for example. The smart lighting control service is a data consumer in the communication with the motion sensor, but a command producer in the communication with the light bulb. This example also indicates that an entity can take on more than one communication role in an automation task.

Second, according to their capability, entities participating in SHAuto are heterogenous in their role in supporting automation (hereinafter referred to as automation role). An SHAuto system generally consists of three types of entities, sensors, controllers, and actuators [56]. Each of these entity types has a specific capability, i.e., sensing, automation processing, and actuating. For example, in the auto control scenario as described in Section IV-A2, the motion sensor is of type sensor, the lighting control service is of type controller, and the light bulb is of type actuator. In addition to these three types, we have added an additional type of entity, i.e., users, to capture the command producer role in the manual control mode. This leads to a classification of the entities participating in SHAuto (hereinafter referred to as SHAuto Entities) based on their capability into four automation roles: Sensor, Controller, Actuator, and User.

Sensor: An entity with sensing capability. It could be an entity that senses or measures particular properties (e.g., temperature) of physical objects (e.g., a room in the house) and transforms them into digital data. It could also be an entity that monitors devices and detects changes of state in devices. Sensors are data producers.

Controller: An entity that enables automated controls of smart devices. It has the capability of processing automations, i.e., it is capable of making automated SHAuto decisions based on sensor data to determine how a device should operate in response to the data. It could be a device programmed to make decisions for specific device operations, or software such as an app or a service. The apps and services can range from simple automation rules to complex services (e.g., home climate control service). Controllers are typically both data consumers and command producers.

Actuator: An entity with actuating capability. It could be a device or a device component that acts on control commands received from Controller(s). There are two cases of actuating. Firstly, an Actuator actuates itself by performing own operations. Secondly, an Actuator actuates devices (e.g., relays, LEDs) connected to it. An example Actuator of the former case is an air conditioner that switches itself on or off upon receiving a command from an air conditioning service. An example Actuator of the latter case is a control unit, which actuates a simple linear actuator interfaced with it, to control the opening and closing of a window, upon receiving a command from a home climate control service. Actuators are command consumers.

User: An end-user app used by human users (e.g., home-owner) to control Actuators. These end-user apps are command producers.

B. AUTOMATION MODES

Automation can be cloud-managed or locally-managed, or a mixture of both [59], depending on where the Controller is hosted. The two automation modes, cloud-managed automation and locally-managed automation, are described below.

1) CLOUD-MANAGED AUTOMATION

As mentioned in Section III-B, smart home platforms today are primarily cloud-based, thereby supporting cloud-managed automation. These platforms rely on cloud services, which act as the Controllers, to automate a home. Depending on the communication capability of devices, cloud-managed automation can be performed in two ways, i.e., with or without an SHAuto gateway. Cloud-connected devices can directly connect to the Internet and interact with the cloud [44], i.e., to exchange data and/or commands with the cloud. Gateway-connected devices, on the other hand, communicate with the cloud via the gateway.

Cloud-managed automation can be described using a smart fan control use case, in which a smart fan is switched on when the room temperature exceeds a threshold. FIGURE 4 shows the workflow of an example scenario involving two gateway-connected devices, i.e., a temperature sensor and a smart fan. The workflow involves five steps: (1) the temperature sensor sends room temperature data to the gateway; (2) the gateway relays the data to the cloud; (3) a cloud service, which is the Controller, processes the data and executes automation rules; (4) the cloud service sends the gateway a *switch-on* command; (5) the gateway relays the command to the fan.

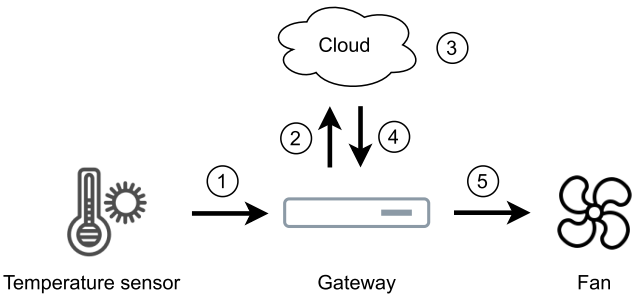

FIGURE 4. Workflow of example cloud-managed automation.

2) LOCALLY-MANAGED AUTOMATION

Locally-managed automation allows automation to be processed locally by using Controller(s) hosted on machines(s) located within a home. In other words, connected Actuators can be controlled locally from within the home local network, without using cloud services. Depending on the type of hosting machines used, this approach can be further classified into two types: gateway-managed automation and device-managed automation.

In gateway-managed automation, Controllers are often hosted on an SHAuto gateway which centrally processes all the automation tasks. Some examples of this approach can be found in [55] and [60]. Gateway-managed automation can be explained using the same smart fan control use case presented above. As illustrated in FIGURE 5, this workflow involves three steps: (1) the temperature sensor sends room temperature data to the gateway; (2) the gateway processes received data and executes automation rules; (3) the gateway sends a *switch-on* command to the fan.

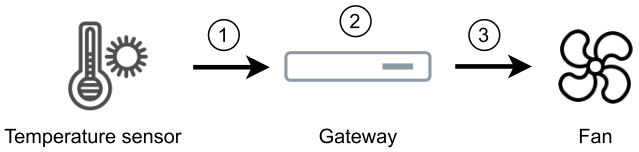

FIGURE 5. Workflow of example gateway-managed automation.

With the advances in AI and the vision for personalised SHAuto, it is envisioned that more and more AI-powered devices will be integrated into homes. An example of an AI-powered device is the LG ThinQ air conditioner that can detect human presence in a room and adjust the temperature accordingly [57]. AI-powered devices can collect data via their interactions with other devices, the underlying environment, and the users, and make automated decisions based on the collected data. We call this on-device automation processing device-managed automation.

From the scenarios discussed above, we can make two observations. Firstly, an SHAuto Entity may operate across different layers of the SHAuto architecture. For example, a Controller may reside in the cloud layer (in cloud-managed automation), the gateway layer (in gateway-managed automation), or the device layer (in device-managed automation). Secondly, an SHAuto Entity (e.g., an AI-powered device) can take on more than one automation role. For example, the air conditioner presented in Section IV-B2 has the Sensor, Controller and Actuator automation roles. To capture such a device, we derive four automation roles from the original ones described in Section IV-A. These roles are: SensorActuator, ControllerActuator, SensorController, and SensorControllerActuator.

SensorActuator: An entity with sensing and actuating capabilities. Such an entity is usually an Actuator which is also a Sensor, and is capable of detecting a change in its state.

ControllerActuator: An entity with automation processing and actuating capabilities. Such an entity can make automated decisions based on sensor data and actuate itself accordingly.

SensorController: An entity with sensing and automation processing capabilities. Such an entity makes an automated decision based on self-generated data and issues a command to the corresponding Actuator.

SensorControllerActuator: An entity with sensing, automation processing, and actuating capabilities. Such an entity is typically a self-contained device which can operate independently. This type of device makes an automated decision based on data generated by its on-device Sensor, and actuates itself accordingly.

C. SHAUTO ENTITY COMMUNICATIONS

There are a number of potential communication scenarios among SHAuto Entities, depending on their automation roles, communication levels, and communication patterns.

For clarity, we first create a classification of SHAuto Entities prior to discussing the scenarios. Based on their automation roles, SHAuto Entities can be classified into two generic types, Simple Entities and Composite Entities. A Simple Entity is defined as an entity having only a single component (a Sensor, a Controller, or an Actuator) with a specific capability. A Composite Entity refers to an entity made up of more than one component. Such an entity may have components with the same or different capabilities. An example of the former is a smart camera with a motion sensor; each of these two components (camera and sensor) has a sensing capability. An example of the latter is a smart camera that, in addition to the camera function, can also make automated decisions.

1) AUTOMATION ROLES

SHAuto Entities with different automation roles may communicate with one another. In the following, these communications are explained using five scenarios. The communication involving the User role is straightforward and has already been discussed in Section IV-A1, hence is excluded from discussion.

a: SCENARIO-1: SENSOR, CONTROLLER AND ACTUATOR

This scenario describes communications involving three Simple Entities (a Sensor, a Controller, and an Actuator). These inter-entity communications are of two types: (i) data communication between the Sensor and the Controller and (ii) command communication between the Controller and the Actuator, as shown in FIGURE 6. An example use case is an air conditioning service (Controller) that receives room temperature data from a temperature sensor (Sensor), and sends a command to switch a smart air conditioner (Actuator) on when the temperature exceeds a threshold.

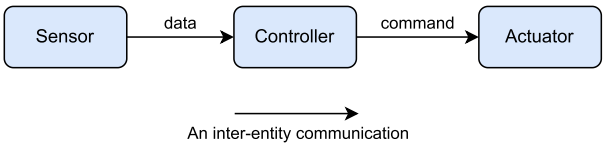

FIGURE 6. Communications in scenario-1.

b: SCENARIO-2: SENSORACTUATOR AND CONTROLLER

This scenario describes communications between a Composite Entity (e.g., a SensorActuator) and a Simple Entity (e.g., a Controller). There are two types of such communications in this scenario, data and command communications, as depicted in FIGURE 7. An example use case is a door lock service (Controller) that receives the state of a smart lock from the smart lock (SensorActuator) and sends a command to the smart lock to lock itself after the smart lock is left unlocked for 30 seconds.

c: SCENARIO-3: SENSOR AND CONTROLLERACTUATOR

This scenario describes communications involving a Simple Entity (e.g., a Sensor) and a Composite Entity

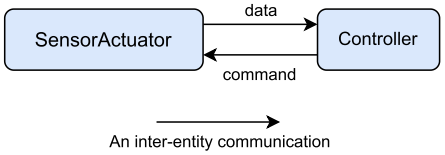

FIGURE 7. Communications in scenario-2.

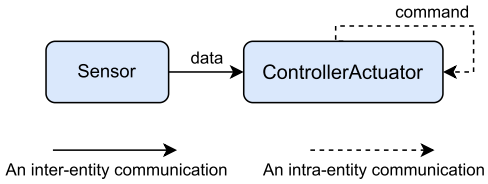

FIGURE 8. Communications in scenario-3.

(e.g., a ControllerActuator). As illustrated in FIGURE 8, there is an inter-entity data communication between the Sensor and the ControllerActuator, and an intra-entity command communication within the ControllerActuator. An example use case is a smart air conditioner (ControllerActuator) that makes an automated decision based on the room temperature data received from a temperature sensor (Sensor), and switches itself on when the temperature exceeds a threshold.

d: SCENARIO-4: SENSORCONTROLLER AND ACTUATOR

This scenario describes communications involving a Composite Entity (e.g., a SensorController) and a Simple Entity (e.g., an Actuator). As shown in FIGURE 9, there is an inter-entity command communication between the SensorController and the Actuator, and an intra-entity data communication within the SensorController. An example use case is a smart temperature sensor (SensorController) that monitors the room temperature and sends a command to switch a smart air conditioner (Actuator) on when the room temperature exceeds a threshold.

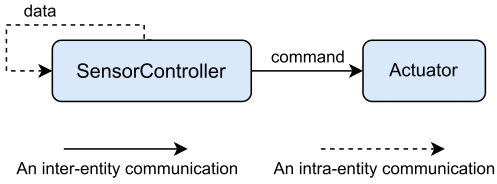

FIGURE 9. Communications in scenario-4.

e: SCENARIO-5: SENSORCONTROLLERACTUATOR

This scenario describes communications involving only a single Composite Entity (e.g., a SensorControllerActuator). As shown in FIGURE 10, in this scenario, there are only intra-entity communications which communicate data and a command within the SensorControllerActuator. An example use case is a smart fridge that provides a child lock feature. The fridge automatically locks itself to prevent children from accessing harmful items (e.g., medicine) when the parents are not around. The fridge has multiple components including a smart exterior camera (Sensor), a child lock service (Controller) and a smart door lock (Actuator). These components work collectively to provide the child lock feature. When a child is approaching the fridge, the camera captures the image of the child which serves as input for the child lock service. The service then performs face recognition to verify if the person is a child, makes a decision, and sends a lock command to the smart door lock to lock itself.

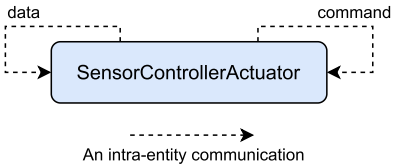

FIGURE 10. Communications in scenario-5.

2) COMMUNICATION LEVELS

Communications among SHAuto Entities can happen at the same level or across different levels. There are two types of same-level communications, i.e., entity-to-entity communications and component-to-component communications, and one type of cross-level communications, i.e., entity-to-component communications.

a: SCENARIO-6: ENTITY-TO-ENTITY COMMUNICATIONS

An entity-to-entity communication occurs at the entity level and can happen between two Simple Entities, two Composite Entities, or a Simple Entity and a Composite Entity. In this scenario, an SHAuto Entity communicates with others as a single unit, regardless of how many components it has. An example of entity-to-entity communications is a data communication between two Simple Entities, e.g., data is communicated between a Sensor and a Controller as described in Scenario-1.

b: SCENARIO-7: COMPONENT-TO-COMPONENT COMMUNICATIONS

A component-to-component communication occurs at the component level and may happen within a single Composite Entity or between two Composite Entities. An example of this communication which happens within a Composite Entity is depicted in FIGURE 11. This example provides a detailed view of the intra-entity communications described in Scenario-5. As shown in the figure, communications occur internally within the Composite Entity (smart fridge d_1) among its components (the smart exterior camera s_1, the child lock service c_1, and the smart door lock a_1).

An example of a component-to-component communication taking place between two Composite Entities (the smart fridge d_1 and a device d_2) is shown in FIGURE 12. In addition to the components described above, the fridge d_1 also has a smart interior camera s_2 which keeps track of what is in the fridge and makes this information accessible to devices

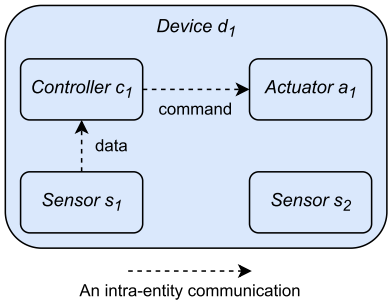

FIGURE 11. An example of component-to-component communications within a composite entity.

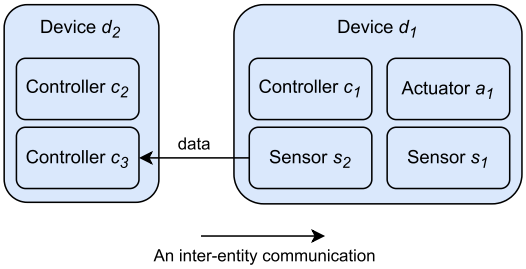

FIGURE 12. An example of component-to-component communication between composite entities.

which need it. d_2 has two Controllers, c_2 and c_3, running on it. c_3 is a grocery shopping service which automatically orders groceries based on available items in the fridge as well as users' consumption and purchasing habits. To perform automated ordering, c_3 needs to know what items are left in the fridge and it obtains this information from the smart interior camera s_2 of the fridge.

c: SCENARIO-8: ENTITY-TO-COMPONENT COMMUNICATIONS

An entity-to-component communication is a cross-level communication which may happen between a Simple Entity and a component of a Composite Entity, or between a Composite Entity and a component of another Composite Entity. This communication can be explained using an automated cooking recipe generation example illustrated in FIGURE 13. This automation involves a smart fridge d_1 and a smart cooker d_3. The cooker can generate various recipes based on the ingredients available in the fridge. The ingredient information consumed by the cooker is captured and sent by the camera s_2 installed inside the fridge.

3) COMMUNICATION PATTERNS

In our previous work [62], [63], we have identified three types of device-to-device interactions showing three communication patterns, i.e., one-to-one, many-to-one, and one-to-many communication patterns. In this work, we expand this concept of interactions in two ways. Firstly, in addition to the interactions between devices, we also consider the interactions between other SHAuto Entities such as services and apps as

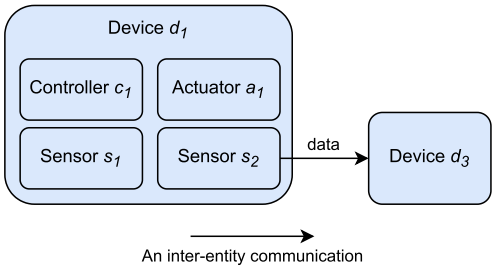

FIGURE 13. An example of entity-to-component communication.

well as their interactions with the devices. Secondly, we also consider the many-to-many communication pattern. This expansion results in four communication patterns of SHAuto Entities: one-to-one, many-to-one, one-to-many, and many-to-many. In the following, we use a smart air conditioning use case to describe each of these patterns.

a: SCENARIO-9: ONE-TO-ONE COMMUNICATIONS

In this scenario, an SHAuto Entity or component sends data or a command to another SHAuto Entity or component. For example, as shown in FIGURE 14, a temperature sensor sends room temperature data to an air conditioning service, which, in turn, issues a command to a smart air conditioner to switch itself on when the room temperature exceeds a threshold.

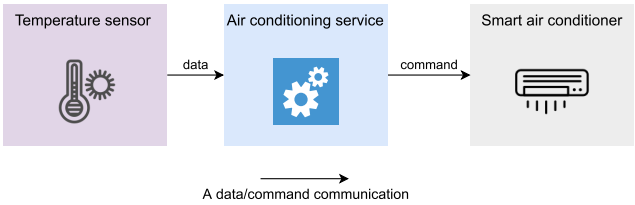

FIGURE 14. An example of one-to-one communication.

b: SCENARIO-10: MANY-TO-ONE COMMUNICATIONS

In this scenario, multiple SHAuto Entities and/or components send data and/or commands to another SHAuto Entity or component. FIGURE 15 shows an example of this scenario, in which two temperature sensors are used to provide room temperature data to an air conditioning service.

c: SCENARIO-11: ONE-TO-MANY COMMUNICATIONS

In this scenario, an SHAuto Entity or component provides data or a command to multiple SHAuto Entities and/or components. FIGURE 16 depicts an example of this scenario, in which, an air conditioning service commands two smart air conditioners.

d: SCENARIO-12: MANY-TO-MANY COMMUNICATIONS

In this scenario, multiple SHAuto Entities and/or components provide data and/or commands to multiple SHAuto Entities and/or components. An example of this scenario, as shown in FIGURE 17, is that two temperature sensors provide

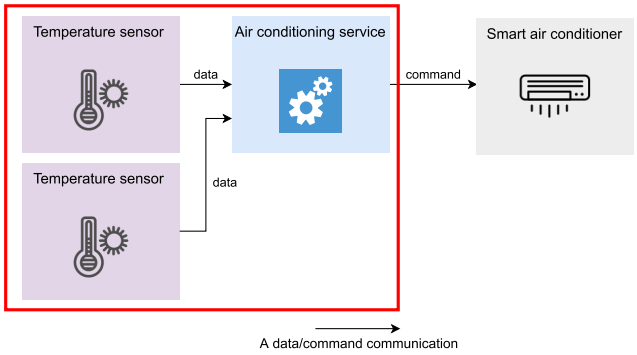

FIGURE 15. An example of many-to-one communication.

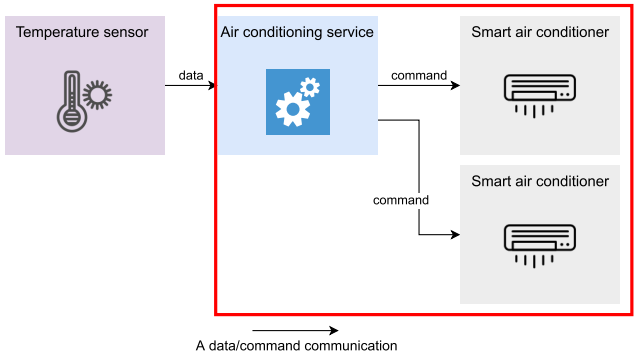

FIGURE 16. An example of one-to-many communication.

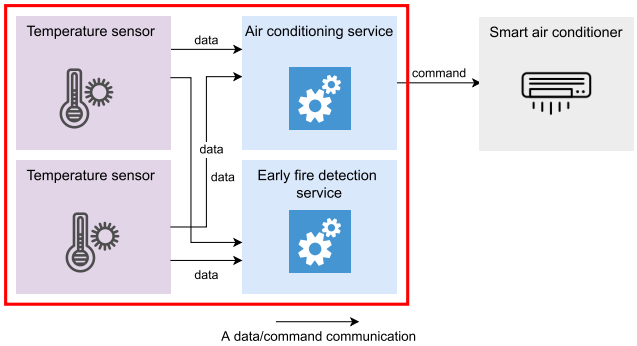

FIGURE 17. An example of many-to-many communication.

temperature data to two services, an air conditioning service and an early fire detection service.

V. A GENERIC MODEL OF SHAUTO

Based on the use case analysis in the previous section, we have formulated a generic model for SHAuto (hereinafter referred to as the G-SHAuto model). The G-SHAuto model captures the different SHAuto use case scenarios presented above, in a communication view called the PubSub Space. The PubSub Space represents a logical space where heterogeneous SHAuto Entities can communicate with one another, regardless of the layer they reside in the SHAuto architecture. FIGURE 18 depicts the G-SHAuto model, how SHAuto Entities are represented in the PubSub Space and how they communicate among themselves in the PubSub Space. A detailed description of the model including its messaging pattern and elements and the aforementioned communications is provided below.

A. MESSAGING PATTERN

We have chosen Publish/Subscribe (PubSub) [64] as the underlying messaging pattern for the G-SHAuto model. In a PubSub system, clients, i.e., publishers and subscribers, exchange messages through a central message broker. Subscribers register their interest in receiving messages through subscriptions, and only receive messages, generated and sent by the publishers that match their registered interest [64]. The matching of messages with relevant subscribers is done by the broker by using a filtering approach. Topic-based filtering and content-based filtering are among the most widely used approaches [64].

In the topic-based filtering approach, subscribers receive all the messages published to the topics to which they subscribe. As shown in FIGURE 19, topics serve as logical channels between publishers and subscribers. In the content-based filtering approach, subscribers receive messages if the content of the messages matches the constraints defined by the subscribers during their subscriptions [65].

The G-SHAuto model adopts the topic-based PubSub messaging pattern for the following reasons:

- The messaging pattern provides asynchronous, loosely-coupled, and many-to-many communications between message producers (publishers) and message consumers (subscribers) [66], [67], making it suitable for SHAuto communications. PubSub has been used in considerable smart home applications, which not only include established IoT platforms such as AWS IoT and IBM Watson IoT, but also research projects (e.g., [68], [69], [70], [71], [72], [73], [74]).
- The messaging pattern supports an event-driven communication model [64], [75], [76], which is commonly seen in SHAuto.
- In addition to one-to-one communications, topic-based PubSub also supports topic-based group communications, thus enabling one-to-many, many-to-one and many-to-many communication patterns described in Section IV-C3. The support for group communications is particularly useful in SHAuto scenarios where a publisher sends an identical piece of data/command to multiple subscribers. Without PubSub, the publisher would have to send the data/command separately and respectively to each of the subscribers. With PubSub, the publisher would only need to publish the data/command once to a topic and the data/command will be delivered to all the subscribers by the broker, thus reducing the number of messages to be transmitted by the publisher.
- PubSub supports persistent sessions [64]. Take the Message Queuing Telemetry Transport (MQTT) proto-

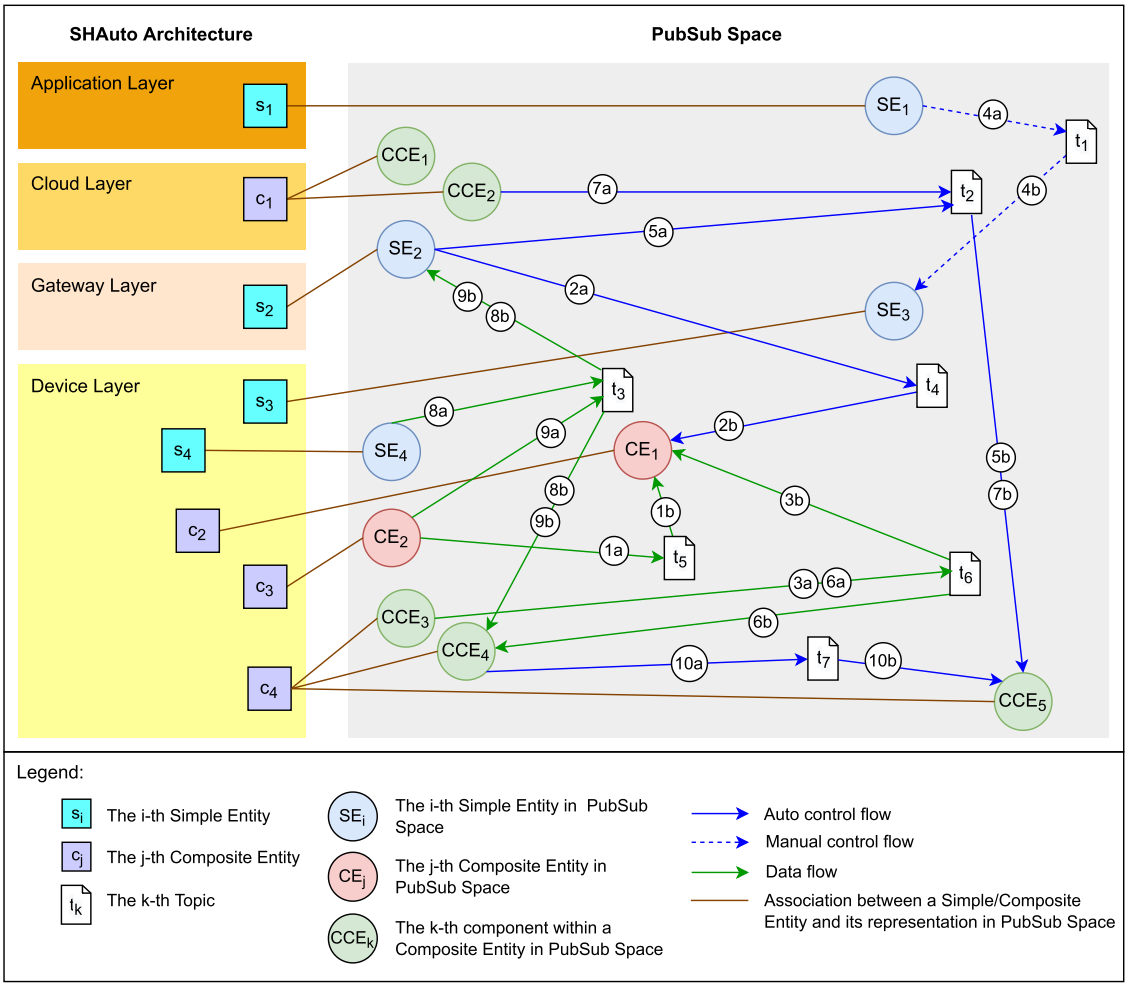

FIGURE 18. The G-SHAuto model.

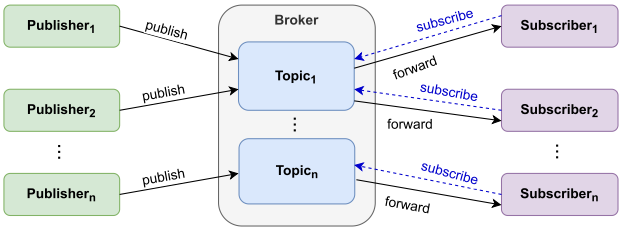

FIGURE 19. Topic-based PubSub architecture.

col [77], a lightweight PubSub protocol, for example. In a persistent session, the broker stores subscription information even if the subscriber is offline, and delivers undelivered messages to the subscriber as soon as it has reconnected [78].

- PubSub protocols such as MQTT allow publishers to mark a message published to a topic to be retained by the broker [67]. In MQTT, such a message is known as a retained message, and will be sent by the broker to any new subscribers immediately upon their subscription to the topic to which the message was published. This can be very useful in the SHAuto context where devices may join or leave (or switch-on or switch-off) dynamically. For example, a smart door lock publishes its status only when changed. With the retained message, a new subscriber will automatically receive the latest status of the smart door lock upon subscription, without the need to wait for the next status change.

B. MODEL ELEMENTS

The G-SHAuto model has three types of elements, PubSub Entities, a Broker and Topics.

PubSub Entities: PubSub Entities are PubSub clients made up of SHAuto Entities that communicate with one another in the PubSub Space to provide SHAuto. For the sake of simplicity, PubSub Entities are hereinafter referred to as entities. Entities include devices, device components, automation rules or services, and end-user apps provided by the SHAuto platform or third parties. To capture the different communication levels as described in Section IV-C2, entities are categorised into three types: SE (Simple Entity),

CE (Composite Entity), and CCE (Component within a Composite Entity). A Composite Entity may communicate in the PubSub Space as a single entity or multiple entities. In the former case, the Composite Entity (e.g., c_2 in FIGURE 18) is represented by an CE (e.g., CE_1 in FIGURE 18). In the latter case, the Composite Entity (e.g., c_1 in FIGURE 18) is represented by multiple CCEs (e.g., CCE_1 and CCE_2 in FIGURE 18). In other words, an CE represents a Composite Entity in the PubSub Space, while an CCE represents a component within a Composite Entity in the PubSub Space.

Entities can be both publisher (i.e., data or command producer) and subscriber (i.e., data or command consumer), that is to say, the communication role of an entity may differ across automation tasks. An example of such an entity is a smart fridge which participates in two automation tasks, ice dispensing and recipe generation. In the former case, the smart fridge is an Actuator which dispenses ice in response to a command received from a smart door lock with a camera when the door lock detects the homeowner has arrived. In the communication with the door lock, the fridge is the command consumer. On the other hand, in the latter case, the fridge is a Sensor which sends the information about ingredients stored in the fridge to a smart cooker which then generates recipes based on the information received. In the communication with the cooker, the fridge is a data producer. In addition, an entity (e.g., the smart lighting control service described in Section IV-A2) may have more than one communication role in an automation task, but can only take on one communication role in a communication, either a data/command producer or a data/command consumer.

Topics: A topic is a logical channel between a pair, or a group, of data producer(s) and data consumer(s) or command producer(s) and command consumer(s). Topics can be dynamically created when an entity publishes or subscribes to them, without any initialisation beforehand (as supported by MQTT [77], [79]).

Broker: A PubSub broker, though not explicitly shown in FIGURE 18, acts as a central server, to which all entities are connected, to enable message transmission between entities via topics. Depending on the automation modes described in Section IV-B, the Broker could be cloud-hosted or locally-hosted on a machine (usually an SHAuto gateway) within the home.

C. COMMUNICATIONS AMONG PUBSUB ENTITIES

The G-SHAuto model involves two forms of communications, namely, inter-entity communications and Entity-Broker communications.

1) INTER-ENTITY COMMUNICATIONS

To facilitate SHAuto, entities communicate with one another by exchanging messages and each such message contains data or commands. These communications are called inter-entity communications and can be categorised into six groups: CE and CE, CE and SE, CE and CCE, SE and SE, SE and CCE, and CCE and CCE. These inter-entity communications are summarised in Table 1. Each group of communication is provided with an example composed of communication flows depicted in FIGURE 18. It is worth noting that in each inter-entity communication, an entity takes on one automation role and one communication role. Take the communication between CCE_3 and CE_1 via topic t_6 for example. In this communication, CCE_3 has the data producer communication role and may assume the Sensor automation role. On the other hand, CE_1 has the data consumer communication role and may assume the ControllerActuator automation role.

TABLE 1. Inter-entity communications in SHAuto.

Inter-entity Communication	Description	Example Communication Flow
CE and CE	A communication between two CEs	Data flows 1a and 1b via topic t_5
CE and SE	A communication between an CE and an SE	Command flows 2a and 2b via topic t_4
CE and CCE	A communication between an CE and an CCE	Data flows 3a and 3b via topic t_6
SE and SE	A communication between two SEs	Command flows 4a and 4b via topic t_1
SE and CCE	A communication between an SE and an CCE	Command flows 5a and 5b via topic t_2
CCE and CCE	A communication between two CCEs. There are two types of this communication: Local CCE and CCE and Global CCE and CCE	
Local CCE and CCE	A communication between two CCEs of the same Composite Entity	Data flows 6a and 6b via topic t_6
Global CCE and CCE	A communication between two CCEs of different Composite Entities	Command flows 7a and 7b via topic t_2

The inter-entity communications shown in TABLE 1 cover different SHAuto scenarios discussed in Section IV. The scenarios include:

- Communications involving different device control modes:
  - Manual control (e.g., command flows 4a and 4b via t_1) and
  - Auto control (e.g., all communication flows except for command flows 4a and 4b).
- Communications involving different automation modes:
  - Cloud-managed automation (e.g., the control of CCE_5 by CCE_2 via t_2),
  - Gateway-managed automation (e.g., the control of CE_1 by SE_2 via t_4), and
  - Device-managed automation (e.g., the control of CCE_5 by CCE_4 via t_7).
- Communications involving different communication patterns:
  - One-to-one (e.g., SE_1 to SE_3 via t_1),
  - Many-to-one (e.g., SE_2 and CCE_2 to CCE_5 via t_2),

– One-to-many (e.g., CCE_3 to CCE_4 and CE_1 via t_6), and
– Many-to-many (e.g., SE_4 and CE_2 to SE_2 to CCE_4 via t_3).

2) ENTITY-BROKER COMMUNICATIONS

The Broker mediates the inter-entity communications described above. This results in four types of Entity-Broker communications: Subscribe, Publish, Forward, and Unsubscribe.

Subscribe: A Subscribe communication involves an entity subscribing to the topic(s) to request messages, containing data or commands, that it is interested in receiving. For example, a smart air conditioner subscribes to a topic to receive control commands from other entities. In a Subscribe communication, an entity sends a Subscribe request in the form of a message to the Broker. We refer to this message as a Subscribe message. A Subscribe message contains information about a subscription such as the identifier of the topic(s) of interest.

Publish: A Publish communication involves an entity publishing a message, containing data or commands, to which other entities might be subscribing. As can be seen from the examples illustrated in FIGURE 18, each of the inter-entity communications is enabled through two sequential communications involving the Broker; the first being the Publish communication. In a Publish communication, an entity publishes a message to a topic managed by the Broker. We call this message an inbound Publish message (hereinafter referred to as Publish message for simplicity). This message serves as a Publish request and may include the identifier of the topic that it will be published to and a control command (e.g., on/off) or sensor data (e.g., room temperature reading). An example of Publish communication is captured by the data flow 1a in FIGURE 18.

Forward: A Forward communication involves the Broker forwarding a message published by a publisher to a subscriber. This is the subsequent communication that completes an inter-entity communication after a Publish communication. A Forward communication is established by the Broker in response to the receipt of a Publish message from a publisher. The Broker then relays the message as an outbound Publish message to every single subscriber of the topic to which the message was published. We name the outbound Publish message the Forward message. An example of Forward communication is exhibited by the data flow 1b in FIGURE 18.

Unsubscribe: An Unsubscribe communication involves a subscriber unsubscribing to remove a request for messages. To unsubscribe from messages, a subscriber sends an Unsubscribe request to the Broker to stop receiving any subsequent messages published to a particular topic. An Unsubscribe request is sent in the form of an Unsubscribe message and it includes the identifier of the topic.

VI. THREAT ANALYSIS

While SHAuto has the potential to offer numerous benefits, it also provides opportunities for attacks. External and internal parties may take advantage of the interconnected smart devices to illegitimately automate a home. In the following, we identify and discuss these opportunities (vulnerabilities and authorisation threats) as well as the impacts they may impose.

A. VULNERABILITIES

Vulnerabilities are known as the weaknesses in a system that an attacker could exploit to cause harmful impacts on the system [80]. In the G-SHAuto model, vulnerabilities can be classified into two categories: unconstrained communications within the PubSub Space and overprivileged entities.

1) UNCONSTRAINED COMMUNICATIONS WITHIN THE PUBSUB SPACE

Interconnected entities, with heterogeneous functionalities, communication capabilities, providers, and so on, can freely communicate with one another through publishing or subscribing to topics to provide SHAuto. This promises enhanced convenience and comfort to home occupants. However, a single compromised entity may potentially compromise the entire SHAuto system. For example, an entity could leverage the group-based communication of PubSub to control numerous other entities by publishing a single command to a topic that they have all subscribed to, without having to compromise every single one of them. For instance, in a one-to-many communication scenario illustrated in 16, a compromised smart air conditioning service could potentially manipulate all smart air conditioners subscribed to it by issuing a single command.

2) OVERPRIVILEGED ENTITIES

An overprivileged entity can access more data than it needs or control more devices than it should. Overprivileges can be attributed to the following features offered by SHAuto platforms.

- The support for automation rule customisation and management of devices and device communications through end-user applications: This provides homeowners with control over their SHAuto system but may also introduce risks (e.g., misconfigurations) due to human error, negligence, or a lack of knowledge in IT or security as described in Section III-A. For example, the homeowner may unintentionally grant a device more permissions than it requires, causing the device to be overprivileged.
- The support for third-party app development and integration (see Section III-B): These apps enrich the SHAuto experience but may be overprivileged [14].

B. THREATS

Entities have access to an SHAuto system, and they may be malicious, curious or compromised. These entities may

misuse their access privilege, intentionally or unintentionally, by exploiting the aforementioned vulnerabilities. We call the misuse of access privilege an authorisation threat. Authorisation threats can be classified into two classes: (1) privilege escalation and (2) privilege blocking. These threats are discussed below and summarised in FIGURE 20.

1) PRIVILEGE ESCALATION

Privilege escalation refers to the act of gaining privileges beyond what is intended for an entity. This threat can happen in entity control and command/data access.

a: PRIVILEGE ESCALATION IN ENTITY CONTROL

Privilege escalation in entity control refers to an event where an entity takes control of other entities which it should not have control over. The entities involved in this threat can be classified into two types: actor and victim. An actor is an entity which causes the threat. A victim, on the other hand, is an entity being controlled as a result of the threat. A victim is usually an entity with actuating capability (e.g., an Actuator, a SensorActuator, or a ControllerActuator). This threat can take two forms, direct control and indirect control.

A direct control threat refers to a circumstance in which an actor directly controls the victim(s). It can be caused by an illegitimate publication by an actor. Such a publication occurs if an actor publishes a command or data to a topic subscribed to by the victim(s), causing the victim(s) to act following the received command/data. For example, a smart coffee maker illegitimately publishes a command to a topic subscribed to by a burglar alarm, a device it should not control, in an attempt to deactivate it. An illegitimate publication can happen in any of the following inter-entity communications:

(COM1) An entity publishing a command to an Actuator or a SensorActuator.
(COM2) An entity publishing data to a ControllerActuator.

The threats of indirect control refer to situations in which actors indirectly control victims through intermediaries. Intermediaries are Controllers which are authorised to control the victims. An indirect control threat happens in two stages: (1) an illegitimate publication by an actor and (2) publication(s) by one or more intermediaries. For simplicity, the stages are explained using a case involving only one victim and one intermediary. In the first stage, an actor publishes data to a topic subscribed to by an intermediary. In the second stage, the intermediary publishes a command, in response to the received data, to a topic subscribed to by the victim. The publication by the actor is illegitimate as it allows the actor to indirectly control the victim which the actor should not gain control over. The publication by the intermediary is legitimate if performed in response to a legitimate publication; however, if carried out as a result of an illegitimate publication as described in the first stage, it will lead to an indirect control threat. An example of indirect control threats involves a motion detector within a room (i.e., an actor) illegitimately sending a data item to a home security service (i.e., an intermediary), which then issues a command to either lock or unlock the main door of the house (i.e., a victim). An illegitimate publication by an actor can occur in the inter-entity communication below:

(COM3) An entity publishing data to a Controller.

b: PRIVILEGE ESCALATION IN COMMAND/DATA ACCESS

Privilege escalation in command/data access arises when an entity consumes commands or data which are not intended for it. This threat may occur as a result of an illegitimate subscription or an unconstrained message forwarding. An illegitimate subscription occurs when an entity subscribes to a topic to which it should not have access. An example is a coffee maker illegitimately subscribing to a topic to read the status of the main door of the house (e.g., On or Off) which is not intended for it. An unconstrained message forwarding, on the other hand, happens when the Broker broadcasts a message, which contains data or a command, to all the subscribers of the topic to which the message was published, irrespective of the fact that only a subset of these subscribers should receive the message. For example, the Broker might broadcast a *switch-on* command to all smart light bulbs in the house, irrespective of the fact that only the light bulbs in the kitchen are intended to receive it. A privilege escalation in command/data access can happen in any of the following inter-entity communications:

(COM4) An Actuator or a SensorActuator consuming a command published by a command producer.
(COM5) A ControllerActuator, a Controller, a Sensor, a SensorController, or a SensorControllerActuator consuming a command published by a command producer.
(COM6) A ControllerActuator or a SensorControllerActuator consuming data published by a data producer.
(COM7) A Controller or a SensorController consuming data published by a data producer.
(COM8) A Sensor, an Actuator, or a SensorActuator consuming data published by a data producer.

The illegitimate publications and subscriptions discussed above may be performed intentionally by a curious, malicious or compromised entity, or unintentionally as the result of device permission misconfigurations by their owners and so on.

2) PRIVILEGE BLOCKING

Privilege blocking is the act of blocking legitimate access to rightful resources [81]. In SHAuto, privilege blocking occurs when the legitimate consumption of commands/data by entities is prevented.

A privilege-blocking threat may occur as a result of an illegitimate subscription cancellation. Such a cancellation happens when a legitimate subscriber unsubscribes itself from a topic illegitimately, stopping itself from receiving commands or data intended for it. This action, which may be performed intentionally or unintentionally by entities, can

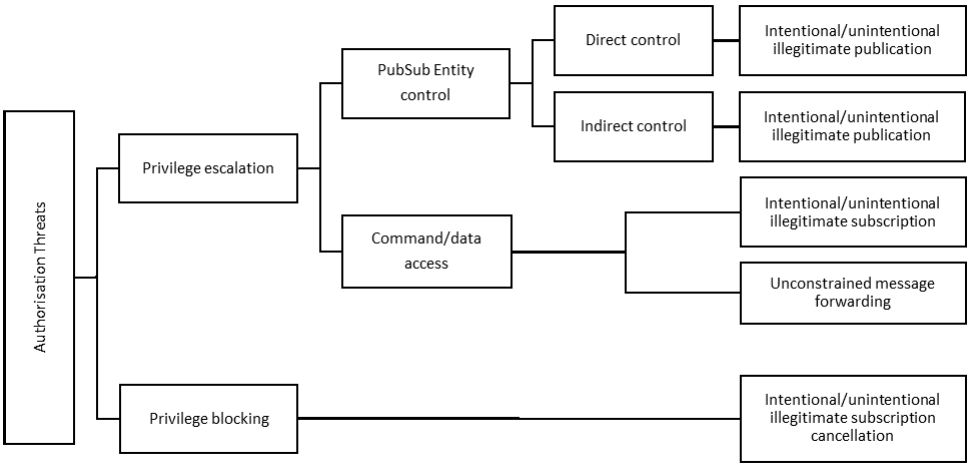

FIGURE 20. Summary of authorisation threats in SHAuto.

result in an availability violation in which commands/data are made unavailable to legitimate subscribers. For example, a compromised smoke detection service might unsubscribe itself from receiving data from smoke detectors, preventing itself from sending alerts to the homeowner's smartphone.

C. IMPACTS OF THREATS

The authorisation threats discussed above could cause one or more adverse impacts on an SHAuto system. These threats not only impact the security and safety of an SHAuto system but also its efficiency. Below, impacts that have the potential to cause safety concerns are discussed before other impacts as they have the highest severity (e.g., endangering occupants' lives).

Operation Obstruction: This occurs when an entity is prevented from operating. It could be caused by a privilege blocking as described above. Take a security camera for example. The camera switches itself on after receiving a command when home occupants leave the house. If the camera is compromised, it could be manipulated to unsubscribe itself from such control commands, rendering it inoperable when occupants are away from home.

Device Misoperation: This occurs when a device fails to operate as intended. Device misoperation can be of two types: internal misoperation and external misoperation. The former means a misoperation posed by an entity (i.e., the device itself), whereas, the latter refers to a device misoperation posed by other entities. An internal misoperation of a device may result from a privilege escalation that occurs in (COM4) or (COM6). On the other hand, an external misoperation of a device could be caused by a privilege escalation that happens in (COM1) to (COM3), or (COM7). Misoperations of devices, which perform critical SHAuto functions, could be disastrous. They may cause device or property damage, putting residents' lives at risk. An example of malicious control of devices, e.g., kitchen appliances, by a compromised entity has been discussed in Section I.

Service Disruption: This is an interruption in the delivery of SHAuto services. It can result from an event of operation obstruction or device misoperation. Multiple occurrences of such an event may potentially result in a whole SHAuto system disruption.

Disclosure of Information: It refers to the dissemination of information to any parties who are not authorised to access that information. It can result from a privilege escalation in command/data access that happens in any of the inter-entity communications described in (COM4) to (COM8). For example, a compromised entity might be used to subscribe to data published by entities (e.g., smart door lock) that are not intended for it, with the aim of learning about the current state of the smart home, such as whether the house is locked or unlocked and if the occupants are home or away. This information could then be used in facilitating home intrusion and burglary.

Impact on Efficiency: The threat actions of illegitimate topic subscription and unconstrained message forwarding may impact the efficiency of an SHAuto system. For example, an illegitimate topic subscription, which is performed by an entity unintentionally, could lead to additional overhead in the entity in processing unwanted messages. The situation is even worse if the entity is a resource-constrained device. Additionally, the threat actions can lead to a rise in the volume of communications, putting an additional burden on the Broker. This extra burden may cause the reduced performance of the messaging system in SHAuto.

VII. REQUIREMENT SPECIFICATION

Based on the SHAuto characteristics (Section III-A), the G-SHAuto model and the threat analysis, this section specifies a set of functional, security, usability, and performance requirements developed by the authors for a secure, usable, and efficient access control solution for SHAuto environments.

A. FUNCTIONAL REQUIREMENTS

(FR1) Communication-based access control: All inter-entity communications, including both data and command communications, should be governed by access control. In other words, access control should be incorporated into all four types of Entity-Broker communications, i.e., Publish, Subscribe, Unsubscribe, and Forward.

(FR2) Support for IoT communication characteristics: The characteristics of IoT communications in SHAuto environments such as dynamic topic creation by entities, persistent sessions, retained messages, and communication patterns should be taken into consideration.

B. SECURITY REQUIREMENTS

The security requirements (SR1 and SR2) are specified to satisfy the safety property as defined by NIST, i.e., an access control system is considered safe if no privilege can be escalated to unauthorised or unintended entities and the correct privileges are always accessible to authorised entities [81].

(SR1) Least privilege communication: An entity should be granted only the minimum privilege for the minimum time frame necessary to perform its task.

(SR2) Availability: The legitimate privilege of an entity to communicate with other entities should not be blocked.

(SR3) Fine-grained access control: The following requirements are specified to achieve fine-grained access control.

- (SR3a) Communication-based access control should be enforced at the lowest level, the CCE level.
- (SR3b) The attributes of entities, including their contextual attributes, should be taken into account when making authorisation decisions.

C. USABILITY REQUIREMENTS

(UR1) Flexibility: The solution should be flexible to accommodate varying access control requirements of different SHAuto systems and the changing needs of an SHAuto system, as described in Section III-A.

(UR2) Ease of use: The solution should support ease of authorisation administration, e.g., policy and privilege/permission administration and privilege/permission review, without requiring user expertise in IT and security.

(UR3) Efficiency of use: The solution should be efficient to use in such a way that human effort or intervention involved in authorisation administration is kept as minimal as possible.

D. PERFORMANCE REQUIREMENT

(PR1) Performance efficiency: Access delays should be kept as short as possible while meeting the functional, security, and usability requirements. In general, an access delay can be defined as the amount of time it takes for an access control solution to produce a decision for an access request.

VIII. CONCLUSION AND FUTURE WORK

SHAuto offers a host of life-enhancing benefits to home occupants but may also pose security and safety problems. In this paper, we have performed a comprehensive problem analysis of an SHAuto environment. In doing so, we have analysed SHAuto use case scenarios from three aspects, i.e., device control modes, automation modes, and communications among entities, identifying communication types, communication patterns, communication levels and the roles of entities in the communications. We have then constructed a generic SHAuto model, based on which, we have identified potential vulnerabilities and threats in relation to authorisation. The problem analysis has led to the specification of a set of requirements, which will be used to guide the next phase of our work, i.e., the design of a secure, usable, and efficient communication-based access control solution for SHAuto environments.

The specified requirements have important implications for the design of such an access control solution. The functional requirements ensures that the access control solution can be designed to be more secure and efficient in handling IoT communications in SHAuto environments. Incorporating access control in all inter-entity communications can ensure that only authorised entities can produce and/or consume data and/or commands. Taking into consideration the characteristics of IoT communications such as dynamic topic creation allows entities to create new topics on the fly. This can be useful for handling ad hoc communication needs.

The least privilege communication requirement is essential to prevent privilege escalation to mitigate the damage that a compromised or malicious entity could cause to an SHAuto system. The principle of least privilege ensures that each entity will only be granted the communication privileges that are absolutely required, and the privileges should be revoked when no longer needed. As a result, if an entity is compromised, the impact of the attack can be confined to the minimal entities that the compromised entity was permitted to communicate with. The availability requirement is crucial to prevent privilege blocking. The fine-grained access control requirement ensures that the solution can be designed to provide more secure access control. This requirement allows the solution to control inter-entity communications based on the attributes of entities and such control is enforced at the lowest level (i.e., the CCE level).

By providing flexibility, ease of use, and efficiency of use, the system can be designed to be more usable and efficient in managing access control requirements, allowing new access control policies and rules to be added as an SHAuto system evolves. Additionally, by ensuring that access delays are kept as short as possible, the solution can provide efficient access control, which is critical for ensuring safety and security in SHAuto environments. An example use case is a smart oven and a smart camera that collectively provide a child safety service. The camera detects if a child is approaching the oven without the presence of an adult. When the detection happens, the camera communicates the information to the oven so that the oven can lock its door to prevent the child from opening the door. To protect children from severe injury, it is critical that the access delay imposed on the communication between the camera and the oven can be minimised.

In our future work, we will employ the specified requirements to design a secure and usable communication-based access control solution for SHAuto environments. In addition, we will draw insights from related white papers (e.g., [82]) to refine our requirements comprehensively. Finally, we will evaluate our solution in terms of security and usability.

REFERENCES

[1] *What is the Internet of Things (IoT)*. Accessed: Aug. 2022. [Online]. Available: https://aws.amazon.com/what-is/iot/

[2] R. Ross, V. Pillitteri, and K. Dempsey, "Assessing enhanced security requirements for controlled unclassified information," Nat. Inst. Standards Technol., Gaithersburg, MA, USA, Tech. Rep. NIST Special Publication (SP) 800-172A, Mar. 2022. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-172a/final

[3] V. H. Bhide and S. Wagh, "L-learning IoT: An intelligent self learning system for home automation using IoT," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2015, pp. 1763–1767.

[4] M. O'Neill, "The Internet of Things: Do more devices mean more risks?" *Comput. Fraud Secur.*, vol. 2014, no. 1, pp. 16–17, Jan. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361372314700089

[5] V. Hu, D. Ferraiolo, and R. Kuhn, "Assessment of access control systems," Nat. Inst. Standards Technol., Gaithersburg, MA, USA, Tech. Rep. NIST Internal or Interagency Report (NISTIR) 7316, Sep. 2006. [Online]. Available: https://csrc.nist.gov/publications/detail/nistir/7316/final

[6] V. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to attribute based access control (ABAC) definition and considerations," Nat. Inst. Standards Technol., Gaithersburg, MA, USA, Tech. Rep. NIST Special Publication (SP) 800-162, Aug. 2019. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-162/final

[7] B. Bezawada, K. Haefner, and I. Ray, "Securing home IoT environments with attribute-based access control," in *Proc. 3rd ACM Workshop Attribute Access Control*, New York, NY, USA, Mar. 2018, pp. 43–53, doi: 10.1145/3180457.3180464.

[8] A. L. Marra, F. Martinelli, P. Mori, and A. Saracino, "Implementing usage control in Internet of Things: A smart home use case," in *Proc. IEEE Trustcom/BigDataSE/ICESS*, Aug. 2017, pp. 1056–1063.

[9] A. La Marra, F. Martinelli, P. Mori, A. Rizos, and A. Saracino, "Improving MQTT by inclusion of usage control," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage* (Lecture Notes in Computer Science), G. Wang, M. Atiquzzaman, Z. Yan, and K.-K. R. Choo, Eds. Cham, Switzerland: Springer, 2017, pp. 545–560.

[10] T. Dimitrakos, T. Dilshener, A. Kravtsov, A. La Marra, F. Martinelli, A. Rizos, A. Rosetti, and A. Saracino, "Trust aware continuous authorization for zero trust in consumer Internet of Things," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 1801–1812.

[11] Y. J. Jia, Q. A. Chen, S. Wang, A. Rahmati, E. Fernandes, Z. M. Mao, and A. Prakash, "ContexIoT: Towards providing contextual integrity to appified IoT platforms," in *Proc. Network Distrib. Syst. Secur. Symp.* San Diego, CA, USA: Internet Society, 2017. [Online]. Available: https://www.ndss-symposium.org/ndss2017/ndss-2017-programme/contexlot-towards-providing-contextual-integrity-appified-iot-platforms/

[12] Y. Tian, N. Zhang, Y.-H. Lin, X. Wang, B. Ur, X. Guo, and P. Tague, "SmartAuth: User-centered authorization for the Internet of Things," in *Proc. 26th USENIX Secur. Symp.*, 2017, pp. 361–378. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/tian

[13] A. Rahmati, E. Fernandes, K. Eykholt, and A. Prakash, "Tyche: A risk-based permission model for smart homes," in *Proc. IEEE Cybersecurity Develop. (SecDev)*, Sep. 2018, pp. 29–36.

[14] E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in *Proc. IEEE Symp. Secur. Privacy*, Aug. 2016, pp. 636–654.

[15] Samsung. *Samsung SmartThings—For Your Connected Smart Home*. Accessed: Aug. 2022. [Online]. Available: https://www.samsung.com/uk/smartthings/

[16] D. Geneiatakis, I. Kounelis, R. Neisse, I. Nai-Fovino, G. Steri, and G. Baldini, "Security and privacy issues for an IoT based smart home," in *Proc. 40th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2017, pp. 1292–1297.

[17] W. Ali, G. Dustgeer, M. Awais, and M. A. Shah, "IoT based smart home: Security challenges, security requirements and solutions," in *Proc. 23rd Int. Conf. Autom. Comput. (ICAC)*, Sep. 2017, pp. 1–6.

[18] A. K. Ray and A. Bagwari, "Study of smart home communication protocol's and security & privacy aspects," in *Proc. 7th Int. Conf. Commun. Syst. Netw. Technol. (CSNT)*, Nov. 2017, pp. 240–245.

[19] A. Girish, T. Hu, V. Prakash, D. J. Dubois, S. Matic, D. Y. Huang, S. Egelman, J. Reardon, J. Tapiador, D. Choffnes, and N. Vallina-Rodriguez, "In the room where it happens: Characterizing local communication and threats in smart homes," in *Proc. ACM Internet Meas. Conf.*, New York, NY, USA, Oct. 2023, pp. 437–456.

[20] R. Li, W. Zhang, L. Wu, Y. Tang, and X. Xie, "ZPA: A smart home privacy analysis system based on ZigBee encrypted traffic," *Wireless Commun. Mobile Comput.*, vol. 2023, Jan. 2023, Art. no. e6731783. [Online]. Available: https://www.hindawi.com/journals/wcmc/2023/6731783/

[21] Y. Meng, W. Zhang, H. Zhu, and X. S. Shen, "Securing consumer IoT in the smart home: Architecture, challenges, and countermeasures," *IEEE Wireless Commun.*, vol. 25, no. 6, pp. 53–59, Dec. 2018.

[22] G. Kavallieratos, V. Gkioulos, and S. K. Katsikas, "Threat analysis in dynamic environments: The case of the smart home," in *Proc. 15th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2019, pp. 234–240.

[23] A. Jacobsson, M. Boldt, and B. Carlsson, "On the risk exposure of smart home automation systems," in *Proc. Int. Conf. Future Internet Things Cloud*, Aug. 2014, pp. 183–190.

[24] H. Touqeer, S. Zaman, R. Amin, M. Hussain, F. Al-Turjman, and M. Bilal, "Smart home security: Challenges, issues and solutions at different IoT layers," *J. Supercomput.*, vol. 77, no. 12, pp. 14053–14089, Dec. 2021, doi: 10.1007/s11227-021-03825-1.

[25] T. H. Kim, L. Bauer, J. Newsome, A. Perrig, and J. Walker, "Access right assignment mechanisms for secure home networks," *J. Commun. Netw.*, vol. 13, no. 2, pp. 175–186, Apr. 2011.

[26] M. Funk, L.-L. Chen, S.-W. Yang, and Y.-K. Chen, "Addressing the need to capture scenarios, intentions and preferences: Interactive intentional programming in the smart home," *Int. J. Design*, vol. 12, no. 1, pp. 53–66, 2018. [Online]. Available: http://www.ijdesign.org/index.php/IJDesign/article/view/2995

[27] M. A. Harris, R. Brookshire, and A. G. Chin, "Identifying factors influencing consumers' intent to install mobile applications," *Int. J. Inf. Manage.*, vol. 36, no. 3, pp. 441–450, Jun. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0268401215301122

[28] G. C.-C. Shen, "Users' adoption of mobile applications: Product type and message framing's moderating effect," *J. Bus. Res.*, vol. 68, no. 11, pp. 2317–2321, Nov. 2015.

[29] S. W. Tay, P. S. Teh, and S. J. Payne, "Reasoning about privacy in mobile application install decisions: Risk perception and framing," *Int. J. Hum.-Comput. Stud.*, vol. 145, Jan. 2021, Art. no. 102517. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1071581920301191

[30] A. Abdullah, H. Kaur, and R. Biswas, "Universal layers of IoT architecture and its security analysis," in *New Paradigm in Decision Science and Management* (Advances in Intelligent Systems and Computing), S. Patnaik, A. W. H. Ip, M. Tavana, and V. Jain, Eds. Singapore: Springer, 2020, pp. 293–302.

[31] P. Agarwal and M. Alam, "Investigating IoT middleware platforms for smart application development," in *Smart Cities—Opportunities and Challenges* (Lecture Notes in Civil Engineering), S. Ahmed, S. M. Abbas, and H. Zia, Eds. Singapore: Springer, 2020, pp. 231–244.

[32] A. Alshehri and R. Sandhu, "Access control models for cloud-enabled Internet of Things: A proposed architecture and research agenda," in *Proc. IEEE 2nd Int. Conf. Collaboration Internet Comput. (CIC)*, Nov. 2016, pp. 530–538.

[33] K. Bing, L. Fu, Y. Zhuo, and L. Yanlei, "Design of an Internet of Things-based smart home system," in *Proc. 2nd Int. Conf. Intell. Control Inf. Process.*, vol. 2, Jul. 2011, pp. 921–924.

[34] (2016). *Cloud Customer Architecture for IoT*. Cloud Standards Customer Council. [Online]. Available: https://www.omg.org/cloud/deliverables/cloud-customer-architecture-for-iot.htm

[35] S. Jaouhari, E. Palacios-Garcia, A. Anvari-Moghaddam, and A. Bouabdallah, "Integrated management of energy, wellbeing and health in the next generation of smart homes," *Sensors*, vol. 19, no. 3, p. 481, Jan. 2019.

[36] M. A. Jabraeil Jamali, B. Bahrami, A. Heidari, P. Allahverdizadeh, and F. Norouzi, "IoT architecture," in *Towards the Internet of Things: Architectures, Security, and Applications* (EAI/Springer Innovations in Communication and Computing), M. A. J. Jamali, B. Bahrami, A. Heidari, P. Allahverdizadeh, and F. Norouzi, Eds. Cham, Switzerland: Springer, 2020, pp. 9–31, doi: 10.1007/978-3-030-18468-1_2.

[37] N. M. Kumar and P. K. Mallick, "The Internet of Things: Insights into the building blocks, component interactions, and architecture layers," *Proc. Comput. Sci.*, vol. 132, pp. 109–117, Jan. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918309049

[38] G. Mokhtari, A. Anvari-Moghaddam, and Q. Zhang, "A new layered architecture for future big data-driven smart homes," *IEEE Access*, vol. 7, pp. 19002–19012, 2019.

[39] J. C. Sapalo Sicato, P. K. Sharma, V. Loia, and J. H. Park, "VPNFilter malware analysis on cyber threat in smart home network," *Appl. Sci.*, vol. 9, no. 13, p. 2763, Jul. 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/13/2763

[40] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of Internet of Things (IoT): A survey," *J. Netw. Comput. Appl.*, vol. 161, Jul. 2020, Art. no. 102630. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804520301041

[41] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, and H.-Y. Du, "Research on the architecture of Internet of Things," in *Proc. 3rd Int. Conf. Adv. Comput. Theory Eng. (ICACTE)*, vol. 5, Aug. 2010, pp. V5-484–V5-487.

[42] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, "IoT gateway: Bridging-Wireless sensor networks into Internet of Things," in *Proc. IEEE/IFIP Int. Conf. Embedded Ubiquitous Comput.*, Dec. 2010, pp. 347–352.

[43] A. Banerjee, F. Sufyanf, M. S. Nayel, and S. Sagar, "Centralized framework for controlling heterogeneous appliances in a smart home environment," in *Proc. Int. Conf. Inf. Comput. Technol. (ICICT)*, Mar. 2018, pp. 78–82.

[44] W. Zhou, Y. Jia, Y. Yao, L. Zhu, L. Guan, Y. Mao, P. Liu, and Y. Zhang, "Discovering and understanding the security hazards in the interactions between IoT devices, mobile apps, and clouds on smart home platforms," in *Proc. 28th USENIX Secur. Symp. (USENIX Security)*, 2019, pp. 1133–1150. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/zhou

[45] T. T. Doan, R. Safavi-Naini, S. Li, S. Avizheh, M. Venkateswarlu, and P. W. L. Fong, "Towards a resilient smart home," in *Proc. Workshop IoT Secur. Privacy*. New York, NY, USA: Association for Computing Machinery, Aug. 2018, pp. 15–21, doi: 10.1145/3229565.3229570.

[46] *AWS IoT for the Connected Home*. Accessed: Mar. 2022. [Online]. Available: https://aws.amazon.com/iot/solutions/connected-home/

[47] *IBM Watson IoT Platform*. Accessed: Mar. 2022. [Online]. Available: https://internetofthings.ibmcloud.com/internetofthings.ibmcloud.com

[48] E. Fernandes, A. Rahmati, J. Jung, and A. Prakash, "Security implications of permission models in smart-home application frameworks," *IEEE Secur. Privacy*, vol. 15, no. 2, pp. 24–30, Mar. 2017.

[49] Z. Chen, F. Zeng, T. Lu, and W. Shu, "Multi-platform application interaction extraction for IoT devices," in *Proc. IEEE 25th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2019, pp. 990–995.

[50] *IFTTT*. Accessed: Sep. 2022. [Online]. Available: https://ifttt.com/

[51] *What is IFTTT?* Accessed: Sep. 2022. [Online]. Available: https://ifttt.com/explore/new_to_ifttt

[52] *IFTTT Smart Home*. Accessed: Sep. 2022. [Online]. Available: https://ifttt.com/solutions/smart_home

[53] S. Asaithambi, S. Venkatraman, and R. Venkatraman, "Big data and personalisation for non-intrusive smart home automation," *Big Data Cognit. Comput.*, vol. 5, no. 1, p. 6, Jan. 2021. [Online]. Available: https://www.mdpi.com/2504-2289/5/1/6

[54] B. Ur, E. McManus, M. P. Yong Ho, and M. L. Littman, "Practical trigger-action programming in the smart home," in *Proc. SIGCHI Conf. Human Fact. Comput. Syst.*, New York, NY, USA, 2014, pp. 803–812.

[55] HomeSeer. *Home Controller Systems For Every Need & Budget bar HomeSeer*. [Online]. Available: https://homeseer.com/home-controllers/

[56] W. A. Jabbar, T. K. Kian, R. M. Ramli, S. N. Zubir, N. S. M. Zamrizaman, M. Balfaqih, V. Shepelev, and S. Alharbi, "Design and fabrication of smart home with Internet of Things enabled automation system," *IEEE Access*, vol. 7, pp. 144059–144074, 2019.

[57] LG. (Jan. 2020). *LG Unveils New Framework for advancing AI Technology at CES 2020*. [Online]. Available: https://thinq.developer.lge.com/en/news/news-list/7-jan-2020-article-ces-keynote/

[58] S. Maxim. (Mar. 2021). *Local Control Made Easy With Ezlo Hubs*. [Online]. Available: https://getvera.com/blogs/our-blog/local-control-made-easy-with-ezlo-hubs

[59] D. Meyer, J. Haase, M. Eckert, and B. Klauer, "A threat-model for building and home automation," in *Proc. IEEE 14th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2016, pp. 860–866.

[60] openHAB. *Empowering the Smart Home*. Accessed: Sep. 2022. [Online]. Available: https://www.openhab.org/

[61] E.-M. Schomakers, H. Biermann, and M. Ziefle, "Users' preferences for smart home automation–investigating aspects of privacy and trust," *Telematics Informat.*, vol. 64, Nov. 2021, Art. no. 101689. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736585321001283

[62] S. AlJanah, N. Zhang, and S. W. Tay, "A survey on smart home authentication: Toward secure, multi-level and interaction-based identification," *IEEE Access*, vol. 9, pp. 130914–130927, 2021.

[63] S. AlJanah, N. Zhang, and S. W. Tay, "A multifactor multilevel and interaction based (M2I) authentication framework for Internet of Things (IoT) applications," *IEEE Access*, vol. 10, pp. 47965–47996, 2022.

[64] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Comput. Surveys*, vol. 35, no. 2, pp. 114–131, Jun. 2003, doi: 10.1145/857076.857078.

[65] Y. Tian, B. Song, and E.-N. Huh, "A parameterized privacy-aware pub-sub system in smart work," in *Security Technology* (Communications in Computer and Information Science), T.-H. Kim, H. Adeli, W.-C. Fang, J. G. Villalba, K. P. Arnett, and M. K. Khan, Eds. Berlin, Germany: Springer, 2011, pp. 204–214.

[66] S. Khare, H. Sun, K. Zhang, J. Gascon-Samson, A. Gokhale, X. Koutsoukos, and H. Abdelaziz, "Scalable edge computing for low latency data dissemination in topic-based publish/subscribe," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2018, pp. 214–227.

[67] D. Happ, N. Karowski, T. Menzel, V. Handziski, and A. Wolisz, "Meeting IoT platform requirements with open pub/sub solutions," *Ann. Telecommun.*, vol. 72, nos. 1–2, pp. 41–52, Feb. 2017, doi: 10.1007/s12243-016-0537-4.

[68] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "CASAS: A smart home in a box," *Computer*, vol. 46, no. 7, pp. 62–69, Jul. 2013.

[69] R. Kishore Kodali, S. C. Rajanarayanan, L. Boppana, S. Sharma, and A. Kumar, "Low cost smart home automation system using smart phone," in *Proc. IEEE R10 Humanitarian Technol. Conf. (R10-HTC)*, Nov. 2019, pp. 120–125.

[70] J. Prabaharan, A. Swamy, A. Sharma, K. N. Bharath, P. R. Mundra, and K. J. Mohammed, "Wireless home automation and security system using MQTT protocol," in *Proc. 2nd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2017, pp. 2043–2045.

[71] Z. Li, Y. Xiao, S. Liang, and S. Wang, "Design of smart home management system based on MQTT and FBP," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2018, pp. 3086–3091.

[72] S.-M. Kim, H.-S. Choi, and W.-S. Rhee, "IoT home gateway for auto-configuration and management of MQTT devices," in *Proc. IEEE Conf. Wireless Sensors (ICWiSe)*, Aug. 2015, pp. 12–17.
[73] Y. Upadhyay, A. Borole, and D. Dileepan, "MQTT based secured home automation system," in *Proc. Symp. Colossal Data Anal. Netw. (CDAN)*, Mar. 2016, pp. 1–4.
[74] F. Ozturk and A. M. Ozdemir, "Content-based publish/subscribe communication model between IoT devices in smart city environment," in *Proc. 7th Int. Istanbul Smart Grids Cities Congr. Fair (ICSG)*, Apr. 2019, pp. 189–193.
[75] T. R. Sheltami, A. A. Al-Roubaiey, and A. S. H. Mahmoud, "A survey on developing publish/subscribe middleware over wireless sensor/actuator networks," *Wireless Netw.*, vol. 22, no. 6, pp. 2049–2070, Aug. 2016, doi: 10.1007/s11276-015-1075-0.
[76] S. Oh, J.-H. Kim, and G. Fox, "Real-time performance analysis for publish/subscribe systems," *Future Gener. Comput. Syst.*, vol. 26, no. 3, pp. 318–323, Mar. 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X09001344
[77] A. Banks, E. Briggs, K. Borgendale, and R. Gupta. (Mar. 2019). *MQTT Version 5.0*. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html
[78] (Feb. 2015). *Persistent Session and Queuing Messages—MQTT Essentials: Part 7*. [Online]. Available: https://www.hivemq.com/blog/mqtt-essentials-part-7-persistent-session-queuing-messages/
[79] (Aug. 2019). *MQTT Topics, Wildcards, & Best Practices—MQTT Essentials: Part 5*. [Online]. Available: https://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices/
[80] M. Jouini, L. B. A. Rabai, and A. B. Aissa, "Classification of security threats in information systems," *Proc. Comput. Sci.*, vol. 32, pp. 489–496, Jan. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050914006528
[81] V. Hu, R. Kuhn, and D. Yaga, "Verification and test methods for access control policies/models," *NIST Special Publication*, vol. 800, p. 192, Jun. 2017. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-192/final
[82] S. Hanna, S. Kumar, and D. Weber, *IIC Endpoint Security Best Practices*. Boston, MA, USA: Industrial Internet Consortium, 2018.

SIOK WAH TAY received the B.Sc. degree in security technology from Multimedia University, Malaysia, the M.Sc. degree in human–computer interaction from the University of Bath, U.K., and the Ph.D. degree in computer science from The University of Manchester, U.K. Her research interests include the IoT security, human–computer interaction, and usable security. She is a fellow of the Higher Education Academy (FHEA), U.K.

NING ZHANG received the B.Sc. degree (Hons.) in electronics engineering from Dalian Maritime University, China, and the Ph.D. degree in electronics engineering from the University of Kent, U.K.

Since 2000, she has been with the Department of Computer Science, The University of Manchester, U.K., where she is currently a Senior Lecturer. Her research interests include security in networked and distributed systems, applied cryptography, data privacy, trust, and digital right managements.

SALEM ALJANAH received the M.Sc. degree in information systems and technology from the University of Michigan, USA, and the Ph.D. degree in the Internet of Things (IoT) security from The University of Manchester, U.K.

He is currently an Assistant Professor of cyber security with the College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Saudi Arabia, and an Honorary Research Fellow in systems and software security with the Department of Computer Science, The University of Manchester. His research interests include the IoT security, applied cryptography, authentication, secure communications, and network security.