Igor Zavalyshyn*, Axel Legay, Annanda Rath, and Etienne Rivière

SoK: Privacy-enhancing Smart Home Hubs

Abstract: Smart homes are IoT systems enabling the automation of household operation. The unrestricted collection and processing of data by smart home systems raises legitimate privacy concerns for their users. Over the past decade, there has been significant interest in privacy-enhancing technologies applied at the level of a local *smart hub* physically located in the home and acting as a gateway between sensors, applications, platform providers, and services in the cloud. The number and variety of projects and research proposals can, however, make their comparison a daunting and unnecessarily complex task. We systematize existing knowledge in this field through the analysis and categorization of 10 industrial and community-contributed systems and 37 research proposals from the literature of the past 11 years. Our results shed light on the diversity of system and trust models considered in the state-of-the-art and on the associated privacy-enhancing technologies. We further identify open research problems and promising approaches that would benefit the smart home hub model and the protection of smart home users' privacy.

Keywords: smart homes, privacy, security, smart hub

DOI 10.56553/popets-2022-0097

Received 2022-02-28; revised 2022-06-15; accepted 2022-06-16.

1 Introduction

The number of deployed Internet of Things (IoT) devices continues to grow, already reaching more than 12 billion by the end of 2021 [1]. A substantial portion of these devices find their way into our homes, falling under the category of *smart home* devices. From voice assistants to IP cameras and thermostats, smart home devices quickly became ubiquitous and their adoption rate is expected to grow even more in the coming years [2].

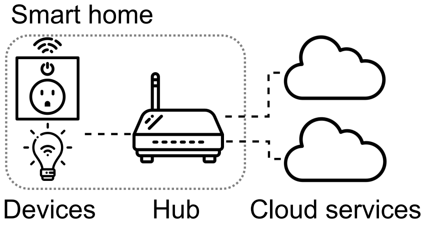

Fig. 1. A typical smart home setup, combining IoT devices, a local (smart) hub, and services in the cloud.

Figure 1 shows a typical IoT setup in a smart home environment. A vast majority of smart home devices today depend on Internet connectivity to perform their assigned task and often rely on various services deployed in the cloud to process and store sensor data. This connection is established through a *hub*, which can be a dedicated device set up by the user, a specific gateway provided as part of a commercial smart home platform such as Samsung SmartThings [3] or Amazon Alexa [4], or simply the home Wi-Fi access point. The hub often serves as a proxy that merely forwards device data and commands to and from cloud backends (a so-called *proxy hub*). In some cases, however, it can also act as an edge server and locally store and process sensor data (acting, therefore, as a *smart hub*). Although the functionality of individual smart home devices in isolation can be limited, smart home platforms stand out by supporting a range of third-party applications, or *apps*, that allow end users to integrate and create automation chains involving multiple devices.

While constant Internet connectivity allows for remote control and monitoring, privacy concerns over sensitive user data being collected and sent to the remote cloud servers constitute one of the barriers to a widespread adoption of smart home technologies [5–7]. Both device manufacturers and smart home platform providers, as well as third-party app developers have unprecedented access to sensor data and can end up violating the privacy of the end users [8–11]. The activity of apps within a smart home platform is generally subject to a certain level of isolation, such as permission-based access control rules. There is, however, no such mechanism in place to govern the actions of the platform provider itself and prevent it from accessing virtually all smart home data in the clear. The situation

***Corresponding Author: Igor Zavalyshyn:** UCLouvain, E-mail: igor.zavalyshyn@uclouvain.be
Axel Legay: UCLouvain, E-mail: axel.legay@uclouvain.be
Annanda Rath: Sirris, E-mail: annanda.rath@sirris.be
Etienne Rivière: UCLouvain, E-mail: etienne.riviere@uclouvain.be

is the same for device manufacturers who control both the devices and associated cloud services. Moreover, a growing number of off-the-shelf devices become insecure when they reach their end of life and may no longer benefit from firmware updates and security patches, leaving end users vulnerable to numerous attacks (e.g., unauthorized access [12], DDoS [13] or ransomware [14]).

To address privacy concerns linked with the uncontrolled transmission and processing of sensor data in the cloud, a rather natural approach is to act locally, enforcing privacy-enhancing measures, such as local storage and computation or flow filtering, directly at the level of the smart home hub. The use of a *privacy-enhancing smart home hub* can offer improved control to users over their devices, their data, its storage, and its processing.

Security solutions and privacy-enhancing technologies proposals revolving on the use of a local smart home hub have generated a significant surge of interest in the last decade (2010 to 2021), with numerous proposals from industry and academia building upon this concept. Some of the proposed systems operate at the connection point between IoT devices and the hub and act as *filters* upon sensor data flows either in the clear [15–19] or in an encrypted form [20–22]. Others intercept and filter communications between the local hub and a cloud service by minimizing the amount and types of shared data [23] or sanitizing sensitive data samples [18]. Some work even suggest full-fledged privacy-enhancing platforms at the edge with secure local data processing [24–26] or novel privacy-aware app programming models [27, 28]. Finally, researchers also suggested mechanisms to improve the security and privacy properties of *existing* commercial and open-source smart hub systems. Among these are novel access control mechanisms [29–34] at the app level or at the user level [35] when the same household is shared by multiple users, data flows tracking systems [36, 37], as well as secure decentralized device and app activities logging for forensics analysis and diagnostics [38, 39].

The abundance of proposals featuring privacy-enhancing smart hubs is encouraging, but also comes with a steep learning curve for practitioners and researchers interested in exploring the state of the art in this field. In addition to the raw number of documented solutions, we observe that proposals often differ slightly or radically in the considered system model (i.e., constituents of the system and their interactions) and in the considered trust assumptions (i.e., which parties are considered malicious or ill-intentioned, and how do the different stakeholders trust each other or not with the handling and processing of sensitive data). As a result, identifying an appropriate solution for a given usage scenario, or comparing different proposals, may become an error-prone and time consuming task.

Our contribution in this paper is to propose a **systematic categorization and analysis of the fast-growing field of smart-hub-based privacy and security**. This SoK (*Systematization of Knowledge*) paper aims to help the research and industrial communities interested in privacy-enhancing technologies implemented at the level of smart home hubs to identify past work and position new proposals in this field.

We selected our base material for this SoK using a systematic analysis of top publication venues in the fields of security, distributed systems, privacy-enhancing technologies, and operating systems, combined with an additional search using publication indexing databases. Furthermore, we selected a number of high-visibility, active commercial and open source projects based on (smart) home hubs, with various levels of consideration for security and privacy. Our goal was to operate an unbiased and comprehensive selection of relevant work, resulting in source material for our systematic study formed of 37 publications and 10 commercial and open source systems (**Section 2**).

Our analysis leverages an analytical framework enabling the comparison and discussion of works using heterogeneous assumptions (**Section 3**). We propose a generic *system* model that encompasses all variants in the analyzed systems and research proposals (e.g., stakeholders, components, and interactions). While few systems include all of these elements in their own model, it allows mapping each of them to the corresponding subset and clarify implicit assumptions. In addition, we map the different points of the IoT workflow in which privacy-enhancements may happen into a number of *security checkpoints*, allowing us to better identify the nature and scope of operations between different systems.

We first identify the extent to which the smart hub idea has been adopted in industrial systems and pinpoint their advantages and limitations with respect to the system and trust models assumed by researchers (**Section 4**). In the main part of the paper, we systematically map the identified published work to our analysis frameworks, drawing similarities, differences, and trends in the field and discussing relations between different traits of our models found in different categories of work (**Section 5**).

After the state of the art analysis in privacy-enhancing smart hubs, we describe other privacy-enhancing technologies that were originally tailored to cloud-based systems but we believe could benefit to the advancement of smart home hubs (**Section 6**). We continue with an overview of common trends in smart home systems design and recent advancements in privacy-enhancing technologies used in those (**Section 7**). We then identify open questions that remain unanswered by both industry and academic communities and stimulate further research in the area (**Section 8**). We finally review related surveys and systematizations of knowledge (**Section 9**) before concluding the paper (**Section 10**).

2 Methodology

The enforcement of security and privacy measures at the level of a smart hub has received considerable attention over the last few years. Our intent is to map existing industrial and academic efforts in this direction. We detail in this section how we identified our source material, i.e., existing systems and research publications, that we will analyze in the rest of the paper.

Selection of existing systems: We first proceeded to a selection of software projects and products representative of the commercial and community activity around smart home systems. Our selection has been based on popularity as well as the number of active contributors, endorsements, forks and releases at the time of analysis as indicated by corresponding code repository (e.g., GitHub) metrics. We discuss 10 selected systems in Section 4, where we compare their architecture, data processing methods, security and privacy properties, as well as guarantees offered to the end-users.

Selection of research works: Our objective in the selection of relevant work from the literature was twofold: (1) to ensure a fair selection exempt of biases (e.g., work we already knew of or authored) and (2) ensure a comprehensive selection of all relevant papers from the corresponding research communities. We first selected a number of venues that we believe the community in privacy-enhancing systems, security, distributed systems, and middleware consider as authoritative in the field. Our resulting list contains 29 venues. This includes 9 journals such as PoPETS, 14 conferences such as USENIX Security, SOUPS, or IEEE S&P, as well as 6 prominent workshops where publications on the topic of IoT security and smart environments generally receive visibility similar to that of a conference. Note that conferences currently using a journal-like publication process, such as PETS with the PoPETS journal, are listed among journals but may have belonged in other categories in the past.

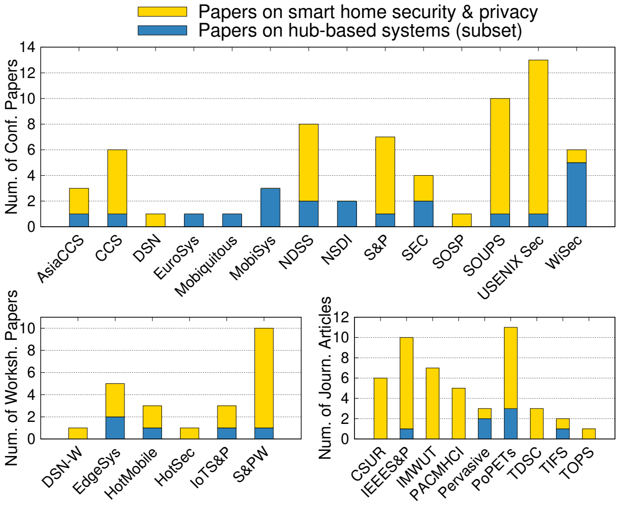

Fig. 2. Number of publications on smart home privacy and security found in the proceedings of relevant venues (Table 3 in the Appendix provides full names and range of considered years).

We proceeded to a first systematic selection among all papers published in these venues over the last 11 years (i.e., 2010 to 2021, both inclusive), picking all papers with a general focus on privacy-enhancing technologies and security for smart home systems. This initial list contains 137 papers, with the distribution per venue shown by the outer (yellow) bars in Figure 2. We analyzed in detail these papers to identify those that, in their proposed solution and/or system model, suggest adopting privacy-enhancing measures at the level of a specific device installed inside the smart home and acting as the connection point for devices, following our (smart) hub model. 34 papers matched this criteria. Their distribution is shown by the inner (blue) bars in Figure 2. In addition, and in order to avoid missing important work not published in this set of venues, we used the Google Scholar bibliographic database. We searched using the following keywords over the period of 2010-2021: *"IoT"*, *"smart home"*, *"smart hub"*, *"privacy"*, and *"security"*. We scrutinized the first 100 results sorted by *relevance* and selected papers that (1) were published in peer-reviewed international venues, (2) were not preceding works from papers already selected in our list, and (3) had a strong focus on the privacy-enhancing smart hub model. This search resulted in only three additional papers published in the TRON forum [40], in PerCom workshops [41], and at the IOTSMS conference [42]. Our analysis is thus based on a total of **37 papers**.

Insights: We observe a general interest from the academic and industrial research communities in smart home privacy and security issues. While the majority

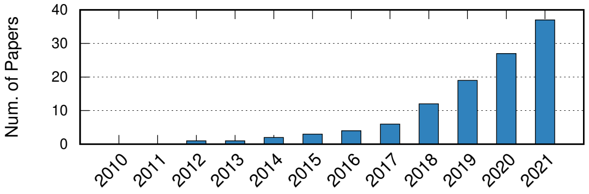

Fig. 3. Cumulative number of publications from 2010 to 2021 on privacy- and security-enhancing smart home systems based on the (smart) hub concept.

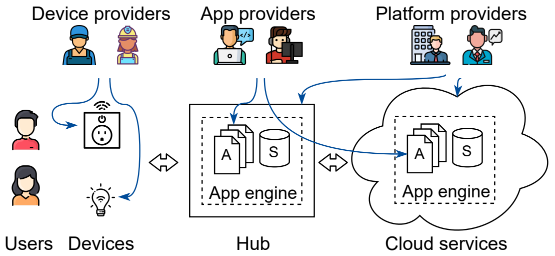

Fig. 4. A generic smart home system model and stakeholders.

of papers on this topic get published at traditional security- and privacy-oriented venues, a few emerge from the venues that are usually associated with topics such as operating systems design or distributed computing. Figure 3 presents the evolution of the (cumulative) number of papers over the studied period of 11 years. We particularly notice a gradual increase in the number of smart home systems proposed over the last five years that exploit edge computing and employ a local server or a hub in their design. This is in line with a growing demand for privacy-friendly smart home technologies from both the end-users and privacy advocates [5–7]. It is certainly also an illustration of the rising number of academic proposals that suggest moving sensitive computation to the edge of the network, and provide the end users with the ways to control the amount and type of data being shared with various service providers.

3 Analytical framework

In this section we describe the analytical framework we derived from our review of selected papers and systems. The objective of this framework is to allow comparing and positioning works that do not necessarily share the same constituents and interactions, or that apply privacy-enhancing mechanisms at different points in the interaction workflow between their constituents.

3.1 Generic system model

During our analysis of the selected material, we observed that the system model (i.e., the components, stakeholders and their modes of interaction) was often different to either a small or large margin, sometimes defined in different terms or using implicit assumptions, thus making a direct comparison between systems difficult. More importantly, these differences in system models have direct implications on the security and privacy features that can be provided, or even make sense in a given system. This observation led us to the conclusion that a more generic model, that does not refer to a specific system but instead encompasses components and stakeholders that can be found in any of them, is required for a proper categorization. In other words, this generic system model is a superset of features found in the different surveyed papers and systems.

Our generic system model, illustrated by Figure 4, includes four types of stakeholders. ① *Device providers* manufacture and maintain commercial off-the-shelf smart home devices and make those available on the market. ② *Platform providers* offer hardware (i.e., hubs) and software (i.e., cloud services) components to control various smart home devices that may not be necessarily compatible with each other but that can nevertheless interact via a single, and often platform-specific, protocol. ③ *App providers* develop and make available to end users (e.g., through platforms' app stores) the software components (i.e., apps) that implement a variety of home automation rules and scenarios. These apps rely on platforms' APIs to interact with connected devices and in order to use storage and network resources. Depending on the system, the apps can either run directly on the hub (e.g., as in openHAB [43]), at a cloud server (e.g., as with Samsung SmartThings [3]) or at both of these locations in a *hybrid* mode (e.g., as in PAIGE [44]). Finally, we have the ④ *users* of these devices, platforms and apps. We consider multi-tenancy, i.e., multiple user(s) can share the same household and have access to devices, platforms, and apps.

Sensor data from connected devices flows to a *hub* and/or to the *cloud services*, where it is shared with and processed by various *apps*. The location of the *app engine* where apps are executed largely defines the system model and has a strong impact on the amount and granularity of data flows leaving the home environment. On the one hand, systems with *hub-centric* and *hub-only* app engines tend to process raw sensor events in place

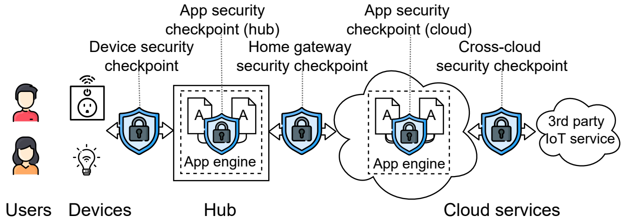
App security checkpoint (hub)
App security checkpoint (cloud)
Device security checkpoint
Home gateway security checkpoint
Cross-cloud security checkpoint
A
A
App engine
A
A
App engine
3rd party IoT service
Users
Devices
Hub
Cloud services

Fig. 5. Security checkpoints in a generic system model.

without necessarily sending them to cloud services. On the other hand, *cloud-centric* systems establish permanent sensor data flows from the home environment to the cloud server where the app engine is running, consequently increasing the risks of data exposure and widening the attack surface. Hybrid systems may execute apps on both hub and cloud. They can overcome the risks of cloud-centric systems by minimizing the amount of raw sensor data leaving the home environment.

3.2 Generic security checkpoints

Without any privacy-enhancing mechanisms in place, all stakeholders including device, platform, and app providers are potentially able to access, analyze, and retain sensitive device data, user actions, and activity in the household. While smart home systems generally employ means to prevent unauthorized access to user and device data, they do not necessarily consider the same set of actors and stakeholders in their threat and trust models. These assumptions result in different approaches to securing system components. To better understand these differences we again view a superset of implemented security mechanisms and pinpoint places in the processing workflow (i.e., *security checkpoints*) where those are applied in the generic system model (see Figure 5). The location of these security checkpoints reflects the diversity of suggested mechanisms in existing smart home systems and research proposals.

Smart home systems that support third-party apps generally recognize threats associated with malicious app providers and implement a variety of security measures to ensure isolation. Various permission- or context-based access control mechanisms restrict app activities to those that were specifically authorized by the users (e.g., permission for an app to access certain devices or device types). Similarly, additional mechanisms that verify cross-app interactions or perform a safety analysis of executed app actions are often deployed. All of these mechanisms are applied at the app engine level and more specifically at an *app security checkpoint* which, depending on the system type, can be found at the hub or at the cloud service.

Other systems implement security mechanisms that restrict sensor data flows from the connected devices to either device or platform providers. These mechanisms are often applied at a *device security checkpoint* where unauthorized device activity can be stealthily detected and the transmitted data can be reviewed, filtered or suppressed, or at a *home gateway security checkpoint* where network flows stemming from individual devices or from applications running on the hub can be analyzed and altered before leaving the smart home environment.

Finally, several of the analyzed smart home systems address the problem of cross-cloud sensor data flows and specifically target popular cloud-based trigger-action platforms (TAPs). These systems implement security mechanisms at *a cross-cloud checkpoint* with additional user-controlled components deployed in the hub. These mechanisms aim to restrict data visibility for platform providers and their affiliated third-party services when processing sensor data events.

Insights: Smart home systems have a rather complex structure and combine multiple technical components operated and deployed by various stakeholders. As a consequence, smart home systems tend to focus their security enforcement on threats originating from specific stakeholders, which defines where the countermeasures are applied. We see that, depending on the place where these countermeasures are applied, end users gain more or less control over the sensitive data flows generated by their devices. While app security checkpoints at the hub or at the cloud service levels have direct access to sensor data and app activities, and offer rich control options to the users, checkpoints at the device or home gateway levels often only have indirect access to sensor data due to the abundance of proprietary software components and the use of encrypted communication protocols, and thus provide a more limited set of options.

4 Existing smart home systems

In this section we analyze six commercial (three standard trigger-based: Samsung SmartThings, Philips Hue, IFTTT; and three voice-activated: Amazon Echo/Alexa, Google Nest/Assistant and Apple HomeKit) and four community-contributed (openHAB, Home Assistant, Domoticz and HomeGenie) smart home systems that we have selected, as discussed in Section 2.

Property	Samsung SmartThings	Amazon Echo/Alexa	Google Nest/Assist.	Apple HomeKit	Philips Hue	IFTTT	openHAB	Home Assistant	Domoticz	HomeGenie
Reference	[3]	[4]	[45]	[46]	[47]	[48]	[43]	[49]	[50]	[51]
Maintainer type	commercial	commercial	commercial	commercial	commercial	commercial	community	community	community	community
Open source	no	no	no	no	no	no	yes	yes	yes	yes
System model	cloud-centric	cloud-centric	cloud-centric	hub-centric	hub-centric	cloud-centric	hub-only	hub-only	hub-only	hub-only
3rd-party apps	no	yes	yes	yes	no	yes	no	no	no	no
App distribution	n/a	app store	app store	app store	n/a	app store	n/a	n/a	n/a	n/a
App access control	n/a	permission	permission	permission	n/a	permission	n/a	n/a	n/a	n/a
App isolation	n/a	yes	yes	yes	n/a	yes	n/a	n/a	n/a	n/a
Stakeholders:										
Device provider	yes	yes	yes	yes	yes	no	no	no	no	no
Platform provider	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
App provider	yes	yes	yes	yes	yes	no	no	no	no	no
Threat model:										
Device provider	◒	◒	◒	●	○	○	○	○	○	○
Platform provider	○	○	○	◒	○	○	●	●	●	●
App provider	○	◒	◒	◒	○	○	○	○	○	○
External	●	●	●	●	●	●	●	●	●	●

Table 1. Existing commercial and open-source smart home systems compared. "n/a" stands for *not applicable*. In the threat models analysis, ● means that a system *implements* a security countermeasure against threats associated with this class of stakeholders, while ◒ and ○ respectively mean that only *partial* or *no* such countermeasure(s) are in place.

Note that two other open-source systems were evaluated, namely IoBroker and OpenMotics, but not included due to their similarity with the selected ones and their relative lower visibility.

Existing smart home systems feature largely heterogeneous system and threat models. They involve different combinations of stakeholders, sometimes allowing multiple different actors with the same role to coexist. They also have different, and sometimes conflicting, views on apps' and users' security objectives and guarantees. We provide a comparison of these systems in Table 1 using the following criteria. The *maintainer* might be a commercial entity (e.g., Apple for HomeKit or Samsung for SmartThings) or a community effort (e.g., openHAB); in general, the latter is associated with an *open source* model and the existence of an online community of developers and users. We distinguish between *system models* that focus on collecting and processing data in the cloud or at a local hub. We did not identify systems with a hybrid system model, i.e., with data processing at both locations. Some systems may support *3rd-party apps*, in which case we also list the mechanism used for the *app distribution*, the associated *access control model*, and whether the runtime for these apps enforces *isolation* between their respective executions. We do not differentiate between standard trigger-based apps (as in IFTTT) and voice-activated ones (as in Amazon Echo or Google Nest) since they all follow an *"if-this-then-that"* logic. We then list for each system which *stakeholders* roles are held by the *platform provider* itself, i.e., whether it also acts as a *device provider* (or, in contrast, rely on off-the-shelf and 3rd-party devices) or as an *app provider* (or, in contrast, rely solely on 3rd-party apps and/or user-defined apps). Finally, we compare the threat models of the selected systems, and analyze which threats they address and to what extent they do so. In the following we discuss common patterns, similarities, and shared or specific weaknesses of these systems.

Commercial vs. community-based systems: Commercial smart home systems (e.g., Samsung SmartThings or Amazon Alexa) are more well-spread due to the wide range of supported devices, number of offered built-in or third-party apps and automations, and a simple and user-friendly design. These commercial products use proprietary software stacks and communication protocols and do not share their source code publicly. This comes in sheer contrast with community-based systems (e.g., openHAB [43] or Home Assistant [49]) that, while often having a more limited set of supported features as compared to commercial products, have publicly-available source code amenable to security verification by a community of developers. Furthermore, with direct access to the source code and packaged versions, end users can run some or all of the system components on their own devices or cloud servers. This may limit data exposure and minimize the number of stakeholders involved in the processing and storage of sensor data.

System model: Depending on the location where sensor data and user actions are processed by various apps and automations (hub or cloud), smart home systems can be divided into two categories: hub-only or cloud-centric. On the one hand, we observe that with two exceptions, namely, Apple HomeKit and Philips Hue, commercial smart home systems implement a cloud-centric system model. While some of these systems offer a hub device (e.g., Samsung SmartThings [3] or Amazon Echo [4]), this device merely acts as a proxy and forwards data from the local devices to a cloud server and receives feedback information from it. A hub-centric system model in which a local hub device can collect sensor data and execute automation rules or user commands has been adopted by Apple HomeKit and Philips Hue. Privacy concerns of the users are the main reason for adopting such a model in HomeKit [52], while Hue suggests operational requirements as a motivation [53]. In both cases, cloud endpoints are only used for remote access and/or encrypted backups. In contrast again, all open-source systems implement a hub-only model with local data processing and a hub device, without depending on external cloud services. While cloud-centric systems offer superior computational and storage resources, hub-centric and hub-only ones provide better privacy and security control to end-users by restricting communication with external services. A difference between commercial and open source hub-based systems lies in the amount of control offered to users. Commercial systems allow sensor data management only via system-specific and often proprietary tools, while open source systems use more standardized data formats that facilitate data export and management.

Apps: Smart home systems generally allow defining automations as trigger-action rules that execute device commands (i.e., an *action*) when a certain condition is met (i.e., a *trigger*). An app consists of one or several such rules. Open-source systems, however, additionally support more complex apps with network and file system access, logging and debug options. The apps can be offered by the platform provider, a third-party app provider, or manually created by the users. The majority of commercial smart home systems support third-party apps and maintain official stores for app distribution. Third-party apps constitute a high-risk threat for end-users' privacy, as they access sensitive device data and can potentially cause data leaks. To prevent this, commercial smart home systems implement a permission-based access control which requires the users to grant an app an explicit right to access a given device and/or user information (e.g., location, contact information, etc.). To prevent potentially incorrect and insecure code execution, apps are running in sandboxes that block unauthorized API calls and unauthorized inter-app communication. While third-party apps are predominant in commercial smart home systems, they are not common in open-source alternatives. In the latter, end users are typically expected to create their own apps (either using a visual UI for simple trigger-action apps with a fixed format, or by manually writing code for more complex custom apps in a free format). This eliminates to a certain extent the risk of a malicious third-party app provider but creates a barrier for adoption among less tech-savvy users.

Stakeholders share: Next, we compare the stakeholder shares across the ten selected systems. In the majority of cases, commercial smart home systems vendors hold multiple roles acting as device providers, app providers, and platform providers at the same time. As a result, a single entity can have unprecedented access to sensitive device and user data. We see that for open-source systems such a monopolistic approach does not apply: only hub software is offered by the platform provider which relies on third-party device providers and user-defined apps.

Threat models: Finally, we compare the threat models considered by the selected smart home systems that often target different actors and, hence, implement different security mechanisms. Commercial smart home systems address device providers threats partially by making mandatory certification and independent testing of both device software and hardware. However, the main goal of these procedures is to ensure correct device behavior and API usage rather than to prevent sensor data abuse or unauthorized sharing. Device manufacturers usually impose their own privacy policy and service agreement which may differ from the smart home system ones. Apple HomeKit stands out by requiring all certified devices to incorporate a hardware-based security module (Apple Authentication Coprocessor) which enforces end-to-end encrypted and mutually authenticated communication with user devices and a HomeKit Hub. Open source platforms consider all connected devices trusted by default. Threats associated with the platform provider are only partially addressed by Apple HomeKit, which uses user-specific public-private key pairs to secure end-to-end communication between connected smart home appliances and user devices. However, as other commercial systems, HomeKit uses a proprietary software stack and communication protocols making any independent security assessment difficult. Open source systems allow their users to not only inspect but also alter the system behavior as needed. Threats associated with the app provider are generally recognized and addressed in all of the commercial systems that support third-party apps, often through a user-defined permission-based access control. The development of apps that control smart home devices in voice-activated systems is restricted to corresponding device manufacturers which limits the risks of potential data exposure. All these apps go through certification procedures that, however, only check compliance with general interoperability requirements but shift responsibility for data privacy to app providers. HomeKit follows a similar approach but also allows regular third-party iOS apps to access smart home data. These particular apps go through a standard app review process and must request permission to access a given smart home device, but once this access is granted the apps can freely aggregate and send sensitive sensor data elsewhere. Finally, all the reviewed systems follow industry standards and best practices for securing their communications (TLS, OAuth2, two-factor authentication, etc.) from external attackers. While some inference attacks were shown to be possible at the network level [22, 54], those offer no direct access to raw sensor or user data.

Insights: Smart home systems tend to have different views on security and privacy depending on their system model and support for third-party apps. Commercial smart home systems are often closed-sourced and follow a cloud-based model, making it difficult to verify if their internal activities are in line with user expectations. Open-source smart home systems, on the other hand, can run on user-controlled devices and their source code can be inspected by the expert users and third-party developers. Threat models are also different. Certification procedure for devices compatible with a given system (if available) ensures operational compliance but provides no incentive for device providers to respect user privacy. Similarly, common across commercial systems is the assumption that platform providers are to be considered fully trusted. However, the very same platforms tend to be obscure about their internal structure and data handling practices, raising numerous privacy concerns [55–57]. Open source systems, on the other hand, are fully transparent in this regard. An independent trusted entity can perform a security audit of the source code and verify the system's data handling processes and practices. Similarly, an open community of developers facilitates code review and incremental improvements of system functionality and security. While commercial systems consider potentially buggy or malicious app providers, and implement various access control and sandboxing mechanisms, their open-source alternatives shift the security responsibility towards end-users who must take special care when creating apps manually. When considering stakeholders we notice an alarming trend towards monopoly among commercial smart home systems. When acting as multiple stakeholders at once platform providers are potentially able to access and aggregate more user data than when acting as a single stakeholder. This constitutes a major threat to user privacy and motivates further research into privacy-enhancing technologies and platforms.

5 Analysis of privacy-enhancing smart hubs research

We now offer an analysis of the 37 research works selected in the literature following the methodology detailed in Section 2. Similarly to commercial and open-source systems, these privacy-enhancing approaches following a hub-based design also tend to have different system and threat models. Depending on their system model (i.e., where sensor data processing takes place), these systems act at different security checkpoints and hence provide different security and privacy guarantees. We compare these systems side-by-side and group them based on their functionality and the security mechanisms that they implement. Our analysis identifies seven distinctive categories of approaches. The systems in these different categories range from those that offer network traffic obfuscation and protection from external network observers, to those that implement data minimization and obfuscation techniques to minimize the exposure of private user data.

Table 2 summarizes the results of our analysis. We distinguish hub-based systems using the following criteria. As with systems described in Section 4 we distinguish between *system models* that perform sensor data collection and processing in the cloud (cloud-centric), on the hub (hub-centric when cloud services are still involved, cloud-only when not) or at both places at the same time (hybrid). We then proceed to identify the stakeholders that the reviewed systems consider in their *threat models*. Depending on the system, the threat models can consider a single or multiple stakeholders as possibly ill-behaved and, therefore, untrusted for handling sensor data without restrictions. Finally, we pinpoint the *security checkpoints* where these systems operate privacy-enhancing mechanisms. A system with multiple components may deploy such mechanisms at a combination of different checkpoints. In the following we discuss similarities and common patterns among each group of systems and highlight some general trends.

System Name	System Model	Threat Model				Security Checkpoint				
		DP	AP	PP	Ext.	Device	App (Hub)	Gateway	App (Cloud)	Cross-cloud
Network Traffic Obfuscation										
Acar et al. [22]	cloud-centric	✗	✗	✗	✓	○	○	●	○	○
Apthorpe et al. [20]	cloud-centric	✗	✗	✗	✓	○	○	●	○	○
Wang et al. [54]	cloud-centric	✗	✗	✗	✓	○	○	●	○	○
Yoshigoe et al. [40]	cloud-centric	✗	✗	✗	✓	○	○	●	○	○
Local Data Processing										
Bolt [26]	hybrid	✗	✓	✗	✗	○	●	○	●	○
HomeOS [58]	hub-only	✗	✓	✗	✗	○	●	○	○	○
NoCloud [59]	hub-centric	✓	✓	✓	✗	●	●	○	○	○
PAIGE [44]	hybrid	✗	✓	✗	✗	○	●	○	●	○
Zhao et al. [24]	hub-centric	✗	✓	✗	✗	○	●	○	○	○
Device Activity Control and Patching										
Capture [60]	cloud-centric	✗	✗	✗	✓	●	○	●	○	○
Chandrasekaran et al. [61]	cloud-centric	✗	✓	✓	✓	●	○	○	○	○
Charyyev et al. [62]	cloud-centric	✗	✓	✗	✓	○	○	●	○	○
Doshi et al. [63]	cloud-centric	✓	✗	✗	✓	○	○	●	○	○
E-Spion [64]	cloud-centric	✗	✗	✗	✓	●	○	●	○	○
Hestia [65]	cloud-centric	✗	✗	✗	✓	○	○	●	○	○
HomeSnitch [66]	cloud-centric	✓	✓	✗	✓	○	○	●	○	○
Magpie [67]	cloud-centric	✗	✗	✗	✓	●	○	●	○	○
Mhaidli et al [68]	cloud-centric	✓	✓	✓	✓	●	○	○	○	○
SecWIR [69]	cloud-centric	✗	✗	✗	✓	○	○	●	○	○
Simpson et al. [41]	cloud-centric	✗	✗	✗	✓	○	○	●	○	○
TLS-RaR [70]	cloud-centric	✓	✗	✓	✓	○	○	●	○	○
Vigilia [71]	cloud-centric	✗	✓	✗	✓	○	○	●	○	○
Ye et al. [42]	hybrid	✗	✗	✗	✓	○	○	●	○	○
Cross-cloud Data Flows Control										
DTAP [72]	cloud-centric	✗	✗	✓	✓	○	●	○	○	●
eTAP [73]	cloud-centric	✗	✓	✓	✓	○	●	○	○	●
App Activity Control										
dSpaces [74]	hybrid	✗	✓	✗	✗	○	●	○	○	○
HomePad [28]	hub-centric	✗	✓	✗	✗	○	●	○	○	○
HoMonit [32]	cloud-centric	✗	✓	✗	✗	●	○	○	○	○
PatrIoT [75]	cloud-centric	✗	✓	✓	✓	○	●	○	●	○
SafeHome [76]	hub-only	✗	✓	✗	✗	○	●	○	○	○
Siegel et al. [77]	cloud-centric	✗	✓	✗	✓	○	●	○	○	○
Data Minimization										
Mandalari et al. [78]	cloud-centric	✓	✗	✗	✗	○	○	●	○	○
PFirewall [23]	cloud-centric	✗	✗	✓	✗	●	○	●	○	○
Data Obfuscation										
Davies et al. [79]	hub-centric	✗	✓	✗	✗	○	●	●	○	○
MegaMind [18]	cloud-centric	✗	✓	✓	✗	●	○	○	○	○
Psychoula et al. [80]	cloud-centric	✗	✓	✗	✗	○	●	●	○	○
Vaidya et al. [81]	cloud-centric	✗	✓	✓	✗	●	○	○	○	○

Table 2. Hub-based research proposals' threat models (consisting of device (**DP**), app (**AP**) and platform (**PP**) providers, as well as external (**Ext.**) attackers) and security checkpoints.

5.1 Network traffic obfuscation

Smart home devices generate enormous amounts of network traffic making it possible for any network observer (an attacker having access to the home network or an Internet service provider) to intercept this traffic and infer privacy-sensitive information, such as the number, types and states of individual devices or daily user activities and schedule. The first group of works addresses this problem through the obfuscation of smart home network traffic. The general goal of this research line is to make this traffic less revealing to potential interceptors before it reaches a dedicated cloud service.

Apthorpe *et al.* [20] show that inference attacks are possible even with encrypted device traffic and suggest a system that combines various mitigation techniques based on traffic shaping, tunneling and rate-limitation. These techniques make it harder to identify and pinpoint individual smart home events by masquerading those within a given traffic trace. Alternative systems suggest dummy or custom spoofed traffic generation [22] to cover real traffic spikes and their frequency, adaptive padding combined with differential privacy [54] to make individual home's traffic indistinguishable from others, and synthetic packet-injection [40] to hide distinctive device traffic patterns. All of these works consider a cloud-centric system model and thus can only act on network flows rather than on the individual sensor data values that are transmitted. While this limits the range of potential privacy-enhancing techniques it allows these approaches to be readily compatible with existing smart home systems that use proprietary communication protocols. Consequently, the only security checkpoint where such systems can operate is at the home gateway level, which is well-positioned to intercept all incoming and outgoing smart home network flows. Due to the focus on a cloud-centric system model, only external attackers are considered as a threat: other stakeholders must be viewed as trusted.

5.2 Local data processing

The second category of works enforces local sensor data processing at the hub, rather than sending it to the cloud. We map five papers to this category. All these works focus on protecting against malicious app providers (AP), i.e., preventing an app from leaking sensitive sensor data it accesses for its legitimate purpose. Dixon *et al.* propose HomeOS – an operating system for the smart home with a local execution model and an easy to use interface to control all connected smart home devices and apps [58]. While the original HomeOS design did not consider a specific adversary model, its extension named Bolt [26] suggested splitting sensitive data into individual chunks and encrypting those with rolling keys before sharing with potentially malicious third-party cloud services and apps. NoCloud suggests strictly on-device or hub-only data processing [59], while others aim for a hybrid hub-cloud architecture [24, 44]. In such hybrid architectures, users can decide the type of data that will be shared with various IoT service providers and the granularity of this data, before the data leaves the premises of a smart home. Most of these works consider app or service providers in their threat models and implement countermeasures at the gateway level or at device security checkpoints. The system models featured in this category of work alternate between exclusively hub-only to hybrid hub-cloud models. While the former offer better control over users' privacy, the latter recognize the utility of virtually unlimited cloud resources and connectivity with external cloud services. Consequently, the chosen system model defines a tradeoff between the privacy protection and performance of a given system.

5.3 Device activity control and patching

Our third group of hub-based privacy-enhancing proposals aim to monitor smart home device activity and prevent unauthorized actions. A subgroup of these works targets smart speaker devices specifically and suggests physical intervention techniques that block the device ability to record user conversations when not used, based on user-defined time periods [61], or gaze direction and voice volume [68]. Other works propose methods to detect smart speaker misactivations based on network traffic analysis, by comparing legitimate and unauthorized voice commands traffic patterns [62].

A number of security-oriented solutions have also been proposed to address the problem of vulnerable IoT devices and cloud services. Some of these such as E-Spion [64], Hestia [65], HomeSnitch [66] or Vigilia [71] among others, implement intrusion detection systems that mitigate the risks of external attacks exploiting unpatched device vulnerabilities [63, 67]. Other solutions revert to local device software patching, as in Capture [60], or securing device communication with an in-hub security manager [41, 69], sometimes even by forwarding requests to compromised or out-of-service cloud servers towards local alternatives [42]. Finally, systems like TLS-RaR [70] suggest novel secure communication protocols for IoT devices and services that allow end users to inspect device activity and transferred data.

The cloud-centric system model prevails across proposals in this group, with a sole exception of a system proposed by Ye *et al.* [42] that employs a hybrid model and allows for local data processing when possible. The considered threat models, however, are rather diverse. We observe proposals that consider a single or several (sometimes even all) stakeholders as potential attackers. Some common patterns emerge, though. First, the systems that aim to secure device software and communication protocols tend to concentrate on external attackers and disregard threats from other stakeholders. These systems usually operate on a device security checkpoint where they can monitor device activity and patch vulnerabilities. Secondly, systems that target device abuse scenarios often consider both app and platform providers in their threat model besides external attackers. These systems act at a device or, more often, on a gateway security checkpoint. Finally, a small set of systems exclusively target malicious app providers. Due to the use of a cloud-based system model these can only operate at a home gateway security checkpoint.

5.4 Cross-cloud data flows control

Our fourth group targets privacy enhancement using a hub for popular trigger-action platforms (TAPs), such as IFTTT [48]. Such TAP platforms in the cloud are associated with significant privacy and security risks since their users are forced to trust the platform provider not only with access to their smart home devices but also to various unrelated cloud services they use on a daily basis (e.g., Dropbox, Google Mail, Slack). Traditionally, access rights in TAP platforms are regulated via OAuth tokens that are generated and authorized by the user connecting multiple trigger-action services together. However, a compromised or malicious TAP provider can have unlimited access to user activity within and outside of a smart home. Furthermore, various TAP trigger-action services may also act maliciously by abusing their access rights and collecting more information that they actually need to perform a given task. Research proposals such as DTAP [72] and eTAP [73] suggest a decentralized way of managing OAuth tokens. Such an approach relies on a trusted client device (or a local hub) to generate unique rule-specific tokens. These tokens cannot be re-used outside of strictly-defined scenarios. A set of cryptographic primitives allows to ensure the authenticity of trigger events and to protect the integrity of transmitted sensor data. Both DTAP [72] and eTAP [73] assume in their threat models a malicious or a compromised TAP provider and any external attacker capable of intercepting communication with the TAP platform. In addition, eTAP [73] also considers semi-honest trigger-action service providers. By including the latter in its threat model, eTAP significantly reduces the risks of data exposure and allows for easy access revocation. Both systems act on a hub and cross-cloud security checkpoints.

5.5 App activity control

Our next identified group focuses on the privacy and security risks of third-party apps running as part of a smart home platform. Some of the work in this category, in particular HomePad [28], PatrIoT [75], or dSpaces [74] propose novel programming models that make app activities explicit and subject to the verification of their compliance with user-defined privacy policy rules. Alternatively, the HoMonit [32] system reverts to device network traffic analysis to detect and block app activities that deviate from the advertised ones in the app description presented to the user. Others implement a context firewall to identify and prevent anomalous app behavior [77], or offer mechanisms to maintain safety, atomicity and isolation (serializability) of concurrent app actions, as in SafeHome [76].

Works in this group target the entire spectrum of system models but consider only a malicious or compromised app provider in their threat models. The only exception is the PatrIoT system [75] which, besides the app providers, lists platform providers as well as external attackers as potential threats and offers additional security countermeasures based on Intel SGX enclaves and their capability for remote attestation [82]. PatrIoT stands out as it targets both an untrusted cloud environment and a local hub as long as SGX-enabled hardware is available. To efficiently monitor app activities all of these systems but one must operate at an app security checkpoint. In contrast, the HoMonit [32] system does not have direct access to app code and runtime environment, and has to operate on a device security checkpoint and intercept encrypted device traffic.

5.6 Data minimization

Our next-to-last category is focused on minimizing the risk of data exposure through *data minimization* principles, i.e., sharing just enough sensor data to ensure a desired system operation. Since all commercial systems use proprietary software, designing hubs and cloud endpoints that can verify a platform's compliance with user privacy and security preferences becomes a challenging task. In this group we placed proposals that aim to reduce the likelihood of privacy violations and limit the overall exposure of users' and of their homes' activities.

Mandalari *et al.* [78] implement a system that automatically identifies and blocks non-essential device traffic. It does so by sequentially probing network flows *to* and *from* a given smart device, and evaluating their impact on device functionality. PFirewall [23] relies, instead, on application logic analysis to determine the minimum amount of data that needs to be disclosed to a platform provider to fulfill a desired automation. It acts at the device or gateway level between the devices and a hub or cloud endpoint respectively. It performs automatic sensor data filtering using generated minimization policies. Both systems consider cloud-based system model since they target existing commercial smart home systems. However, their threat models are rather distinct: while Mandalari *et al.* [78] consider malicious device providers, PFirewall [23] targets malicious or compromised platform providers. The key difference lies in granularity: while Mandalari *et al.* [78] concentrate on individual device data flowing to the device provider (lower exposure), PFirewall [23] abstracts away and targets the data flows from *all* connected devices that can flow to a single platform provider (higher exposure).

5.7 Data obfuscation

In this final category we group research proposals that perform data *obfuscation* on a smart hub. This category differs from the previous category of data *minimization* as it assumes direct access to raw sensor data and the ability to alter personally identifiable information in a way that makes it indistinguishable from the data of other users.

Several academic systems implement obfuscation mechanisms for smart voice assistants by forcing pitch and tone shifting (to prevent user identification and profiling), as do Vaidya *et al.* [81], or by sanitizing voice commands inputs and outputs (to prevent accidental or hidden data release and block unwanted content), as in MegaMind [18]. Davies *et al.* [79] as well as Psychoula *et al.* [80] suggest the idea of so-called *privacy mediators* running at a local hub or gateway. These mediators dynamically enforce user-defined privacy policies on all raw sensor data flows towards cloud services. Depending on the target device type and ability to execute app logic locally, these systems operate on a device or a hub/gateway level. They all consider malicious app providers in their threat models but those that target smart speaker devices, e.g., Vaidya *et al.* [81] and MegaMind [18], also consider malicious platform providers.

5.8 Insights and discussion

All of the hub-based smart home research proposals reviewed in this section address specific privacy and security problems of existing commercial smart home devices and platforms. We see that a significant number of these systems implement security mechanisms to protect vulnerable and often unpatched device software components and communication protocols from external attackers. Due to a prevalent closed-source nature of smart home products, a home gateway (router) overseeing all the network flows remains the most common (and often the only) place where privacy-enhancing mechanisms can be applied.

System targeting malicious app providers in their threat models often suffer from the inability to alter the behavior of existing smart home platforms. As an alternative, they suggest novel platform designs, app programming models and approaches to sensitive data handling. While novel system designs with enhanced privacy controls bring numerous benefits to end users, we see that there are still just a few systems considering existing platform providers in their threat models, and even less so for device providers. The ability to verify the actions of these stakeholders and make them accountable for any violation is not only required by data protection regulations (e.g., EU GDPR [83] or the upcoming update of California's CCPA [84]) but is also an extremely important feature now that more and more of smart home devices appear on the market.

6 Adopting cloud-based PETs in smart home hubs

In the previous section we reviewed privacy-enhancing systems and research proposals that explicitly rely on the use of a local smart hub to protect smart home users from various threats, including those associated with malevolent app and platform providers. In this section, we review existing techniques, related to static or dynamic app code analysis, novel access control mechanisms, data flows tracking and device monitoring outside of our selected working literature material, that were specifically designed to improve the security and privacy properties of cloud-based smart home systems. While these techniques target a cloud environment without the use of local smart hubs, we argue that the privacy-enhancing techniques and approaches they feature have strong potential to be implemented and successfully used in hub-based solutions. We group them based on a scope of implemented security mechanisms and briefly describe the specifics of each technique.

App access control: Multiple techniques have been proposed to address the problem of over-privileged third-party apps [8] suggesting novel access control and risk assessment mechanisms. For instance, Rahmati *et al.* propose the Tyche development framework [29] that aims to minimize the security risks posed by a given app. Apps developed using this framework may only access the device data and actions restricted according to the risk factor identified by the user: e.g., a *door.lock()* action is associated with a low risk while the *door.unlock()* action is typically associated with a high risk.

A more elaborate approach is used by the SmartAuth system [30], which relies on knowledge about a context in which a given device action takes place to determine if an app is allowed to execute it or not. Through source code and app description analysis at install time, SmartAuth can predict potential security and privacy violations and help the user to make a conscious decision when installing a given app. Similarly, the Soteria system [31] uses model checking to determine violation of security and safety properties defined by the user within various third-party apps installed in the system. Zhang *et al.* [32] suggest an alternative approach to detecting malicious activity of third-party smart home apps. Their approach uses information inferred from the network activity of a given app to determine if it is in line with an advertised app functionality. Illegal app actions are then blocked by the system.

Cross-app interaction Cross-app interaction chains that may accidentally or intentionally (as part of a carried attack) violate the security and even the safety of smart home users proved to be possible in commercial smart home systems. The IoTMon system [33] aims to detect such violations at app installation time by checking if a new app's activity is in conflict with any of the previously installed apps. An alternative approach is used by the IoTGuard system [34] that relies on app code *instrumentation* to dynamically enforce safety and security policies at runtime, by blocking the execution of violating device actions within a single app or a chain of interacting apps.

Multi-user access control: Existing smart home systems are often ill-designed for multi-user environments and offer a limited set of access control mechanisms. The academic community has highlighted the shortcomings of existing approaches by reviewing popular smart home platforms, and suggested novel access control mechanisms and alternative design principles [85, 86]. Among others, the Kratos system [35] suggests a multi-user and multi-device access control mechanism which uncovers conflicting preferences among co-living users and automatically tries to suggest appropriate solutions. Such a mechanism is essential when multiple users share the same smart home devices and may naturally have different expectations and views on devices activities and operation modes. Furthermore, Zeng *et al.* [85] suggest access control based on location, roles or delegated authorization, while Geeng *et al.* [86] urge platform providers to consider different relationship types (e.g., between a landlord and a renter), temporal or intermittent users, and changing relationships (e.g., married vs. divorced) in the design of their systems.

Sensitive data flows tracking: Smart home apps need access to sensitive sensor data to perform a given task but may abuse this access by sharing information about the user activities with unauthorized parties (i.e., causing a data leak). To prevent such data leaks various systems have been proposed over the last few years to protect the privacy of the end users. The Flowfence system [27] suggests splitting application logic into components that work with sensitive data and those that do not. Components requiring access to sensitive data run in isolated sandboxes and communicate with other components only through well-defined API calls that allow taint-tracking and security enforcement. The Saint system [36] uses static code analysis to identify all potentially-sensitive flows between data sources (API calls that return raw sensor values) and data sinks (API calls that provide access to network and messaging services). A similar approach has been used to track and analyze sensitive flows in the IFTTT platform's JavaScript-based applets [37].

Logging for diagnostics and forensics: Modern smart home platforms provide their users with little to no access to app and device activity logs, making it hard to perform security audits or diagnose problems. Several novel logging mechanisms propose to overcome this problem. Among those, the ProvThings [38] and IoTDots systems [39] rely on app code and device API instrumentation to generate provenance logs in real time. Such logs can later be used to analyze the system behavior and determine the root cause of certain device state or action at any given time.

Insights: We see a tremendous amount of proposals from the research community to improve the privacy and security properties of existing cloud-based smart home systems. These proposals highlight numerous flaws and discrepancies in system designs and/or their threat and trust models. While these proposals target cloud-based systems, hub-based systems often suffer from the same flaws and hence can benefit from intelligent access control for app and user activities, as well as data flows tracking and logging, among others. While directly applying these novel techniques at hub-based systems might not be a straightforward process (e.g., due to incompatible APIs, limited resources, or lack of access to global databases available in the cloud) we argue that core security principles could remain unchanged.

7 Discussion

Some general trends became apparent during our analysis. Among industrial systems we still see a prevailing cloud-centric system model, however a hub-based model has gained traction over the last few years, particularly with the Apple HomeKit and Philips Hue systems. We also notice an upcoming shift towards a hub-based design among other industry leaders, such as Amazon Alexa [87] and Samsung SmartThings [88]. We see a similar trend within scientific communities. Early academic research on smart home systems strongly advocated for a local hub-centric design (e.g. HomeOS [58]), but later switched to hybrid- [26, 44] and increasingly cloud-centric architectures [27, 30, 34] following the design of commercial systems. Today, however, we see a backward trend with computation and storage functionality moving back to the edge where fine-grained data flow control is possible [24, 28]. While this is a promising trend, overall the smart home market is still very fragmented and lacks common open standards for device connectivity, application development and data transfer. Such standards would not only allow the users to easily migrate from one platform to another, but would also make it possible to inspect and customize a given system according to user privacy preferences and needs.

Smart home devices especially those equipped with actively-listening microphones constitute a major threat to privacy. The research community responded with numerous systems aiming to detect and prevent illegal device activity [64, 70]. These efforts, however, will have limited efficacy unless these devices are freed from vendor or platform specific dependencies, i.e., proprietary communication protocols and software stacks, static network routes, and unchangeable data storage options.

In terms of application development, we see that smart home systems increasingly move towards built-in (native) or user-defined applications as opposed to third-party ones, e.g., SmartThings [89]. Such a move significantly reduces an attack surface eliminating the application developer threat. While applications offered by the device or service providers can still pose a significant risk, these are generally included in a trust model.

A platform provider threat became more evident recently which motivated both industry and academic efforts on designing secure- and private-by-design smart home systems. We see a rise of systems implementing software- [73] and hardware-based [75] PETs that rely on end-to-end encryption protocols, or use Trusted Execution Environments (TEEs) or Trusted Platform Modules (TPMs) that enable secure computation at the untrusted cloud (e.g., Intel SGX) or device (e.g., ARM TrustZone) hardware. These PETs restrict sensor data access to end users only and make it impossible for platform or infrastructure providers to eavesdrop.

Various well-known security- and privacy-enhancing techniques have been effectively used in a broader IoT context (smart cities, factories and grids) for some time, but are still not common within a smart home scenario. This is the case, in particular, for techniques that rely on fully or partially homomorphic encryption for secure IoT data collection and processing in untrusted cloud environments [90–93], differential privacy for private data sharing and IoT traffic obfuscation [94, 95], or oblivious RAM techniques for hiding data access at compromised cloud servers [96].

Overall, smart home systems in face of rising privacy concerns and regulations are gradually shifting towards more user-centered and privacy-friendly operation modes. However, there are still many open challenges and unanswered research questions remaining. We list some of those in the next section.

8 Open research questions

There is a number of open questions that remain unanswered still and should motivate future research into privacy-enhancing hub-based smart home systems. We list the prominent questions that emerged during our analysis of the state of the art.

How to control the device activity? We see a lack of security and privacy-enhancing solutions targeting threats associated with malicious device providers. Due to the use of proprietary software and communication protocols, inspecting the device activities and generated data flows is a challenging and often error-prone task.

Emerging security solutions based on manufacturer usage description (MUD) [97] constitute a promising solution but still require collaboration and willingness to comply from device manufacturers. Solutions that verify the advertised description with an actual device behavior and gracefully recover from detected conflicts are therefore needed. A smart hub is well-positioned in the smart home ecosystem for implementing such solutions.

What should be the appropriate user interfaces when smart home systems are, by nature, embedded and their activity is often invisible? How do we raise awareness of privacy issues among current and future users, in particular people who have limited technological literacy and experience, or have accessibility issues (e.g., elderly or disabled users)? Both industrial and academic smart home systems often assume tech-savvy users. In reality, end-users are often confused by, and lost in, the complexity and unpredictability of these systems [5, 98, 99]. Elderly users, for instance within an *aging in place* scenario that some envision to benefit from the deployment of smart home solutions [100, 101], tend to be unaware of and susceptible to various security and privacy risks posed by smart home systems [102]. There is a growing need for user interfaces that would facilitate the configuration of smart home systems according to one's security and privacy expectations regardless of technological literacy, as well as effective mechanisms to translate user mental models into actual system behavior.

What are the *"killer apps"* that would promote hub-based systems adoption? As became evident in our SoK analysis, cloud-centric smart home systems still prevail on the market. However, we witness a general shift towards systems that utilize local devices for sensitive data processing. In particular, federated machine learning has been successfully used to protect user privacy in mobile systems [103], but it is still relatively new to a smart home environment. On-hub model training and inference (e.g., for object or user activity recognition) can be effectively done without sending raw sensor data to the cloud services. Complex models can be collaboratively trained this way across multiple smart home hubs without putting their owners' privacy at risk. Such privacy-aware collaboration opens up doors to various novel smart hub use case scenarios that would not be otherwise possible under the umbrella of a single smart home system. While we see some early work in this direction, such as the IOTFLA [104] privacy-conscious federated learning platform for smart homes, much remains to be done in this area.

9 Related work

A number of surveys and SoK papers were recently published on the topic of IoT security and privacy in smart homes. We analyze in this section the contributions of these works with respect to our own SoK paper.

Alrawi *et al.* [105] carried out a security analysis of smart home system components and reviewed common attack scenarios, involved stakeholders and mitigation techniques. While we recognize the importance of securing system components and highlight several relevant security mechanisms implemented at the device or hub level, in the current paper we also target end-user privacy threats and showcase systems that implement corresponding countermeasures.

He *et al.* [106] surveyed academic literature on context sensing for access control in smart homes, and pointed out flaws in existing systems under adversarial attacks. Context-sensing is just one of the approaches adopted by the systems we reviewed in this SoK paper. We additionally review numerous other techniques to verify the authenticity, safety and permission validity of a given device or user action.

Finally, Babun *et al.* [107] performed an in-depth analysis of popular smart home platforms. As in our SoK the authors review commercial and open-source platforms and compare their system and app programming models, communication protocols, third-party components support, as well as point out their limitations when dealing with sensitive sensor data and apps. However, the contributions of the present SoK paper go beyond existing systems review and analysis, offering a comprehensive study of privacy-enhancing hub-based systems and additionally of alternative cloud-oriented techniques that can be applied locally.

10 Conclusion

In this paper we examined **10** industrial and **37** academic smart home systems and compared their system and threat models, implemented security and privacy-enhancing mechanisms as well as the ways they deal with sensitive sensor and user data. To facilitate this comparison we derived an analytical framework that accounts for heterogeneous system design, uneven stakeholders shares and different views on where the security and privacy control should be applied.

Among many of our findings, the main ones suggest that cloud-based system model is still prevailing among smart home systems, although various systems and techniques have been proposed to either process sensor data locally at the smart hub, or at least minimize the data exposure by obfuscating or sanitizing sensitive user data before it leaves the home environment. We see however a gradual shift towards, and a general interest in, hub-based or hybrid system models that offer better security and privacy protection to the end-users. Our systematization of knowledge should provide important insights for system developers and stimulate further research in privacy-enhancing technologies targeting smart home systems.

11 Acknowledgments

This work was supported by the Brussels Institute for Research and Innovation (Innoviris) under the "Smart and Social Home Care" research project. We thank the anonymous reviewers for their precious time and for their valuable suggestions and comments.

References

[1] Satyajit Sinha. State of IoT 2021: Number of connected IoT devices growing 912.3 billion globally, cellular IoT now surpassing 2 billion. https://iot-analytics.com/number-connected-iot-devices/, 2021. Accessed: June 2022.

[2] John Koetsier. Amazon's full-on smart home assault: All the highlights. https://bit.ly/39BxUsg, 2021. Accessed: June 2022.

[3] Samsung SmartThings. https://www.smartthings.com, 2021. Accessed: June 2022.

[4] Amazon Alexa. https://www.amazon.com/smart-home-devices, 2021. Accessed: June 2022.

[5] Madiha Tabassum, Tomasz Kosinski, and Heather Richter Lipford. "i don't own the data": End user perceptions of smart home device data practices and risks. In *15th Symposium on Usable Privacy and Security*, SOUPS, pages 435–450, 2019.

[6] Julie Haney, Yasemin Acar, and Susanne Furman. "it's the company, the government, you and i": User perceptions of responsibility for smart home privacy and security. In *30th USENIX Security Symposium*, 2021.

[7] Serena Zheng, Noah Apthorpe, Marshini Chetty, and Nick Feamster. User perceptions of smart home IoT privacy. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–20, 2018.

[8] Earlence Fernandes, Jaeyeon Jung, and Atul Prakash. Security analysis of emerging smart home applications. In *IEEE symposium on security and privacy*, S&P. IEEE, 2016.

[9] Adam Clark Estes. Yes, Your Amazon Echo Is an Ad Machine. https://gizmodo.com/yes-your-amazon-echo-is-an-ad-machine-1821712916. Accessed: June 2022.

[10] Jay McGregor. Here's How Amazon's Ring Doorbell Police Partnership Affects You. https://bit.ly/3G0wKCQ. Accessed: June 2022.

[11] Christine Hauser. Police Use Fitbit Data to Charge 90-Year-Old Man in Stepdaughter's Killing. https://nyti.ms/2Oz8P5j. Accessed: June 2022.

[12] Dara Kerr. Ftc and trendnet settle claim over hacked security cameras. https://www.cnet.com/tech/services-and-software/ftc-and-trendnet-settle-claim-over-hacked-security-cameras/, 2013. Accessed: June 2022.

[13] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. Understanding the mirai botnet. In *26th USENIX security symposium*, 2017.

[14] Dan Goodin. When coffee makers are demanding a ransom, you know iot is screwed. https://arstechnica.com/information-technology/2020/09/how-a-hacker-turned-a-250-coffee-maker-into-ransom-machine/, 2020. Accessed: June 2022.

[15] Phuthipong Bovornkeeratiroj, Srinivasan Iyengar, Stephen Lee, David Irwin, and Prashant Shenoy. RepEL: A utility-preserving privacy system for iot-based energy meters. In *IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation*, IoTDI. IEEE, 2020.

[16] Omid Hajihassnai, Omid Ardakanian, and Hamzeh Khazaei. Obscurenet: Learning attribute-invariant latent representation for anonymizing sensor data. In *International Conference on Internet-of-Things Design and Implementation*, IoTDI, 2021.

[17] Mohammad Malekzadeh, Richard G Clegg, and Hamed Haddadi. Replacement autoencoder: A privacy-preserving algorithm for sensory data analysis. *arXiv preprint arXiv:1710.06564*, 2017.

[18] Seyed Mohammadjavad Seyed Talebi, Ardalan Amiri Sani, Stefan Saroiu, and Alec Wolman. MegaMind: a platform for security & privacy extensions for voice assistants. In *19th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys, 2021.

[19] Suman Jana, Arvind Narayanan, and Vitaly Shmatikov. A scanner darkly: Protecting user privacy from perceptual applications. In *IEEE symposium on Security and Privacy*, S&P. IEEE, 2013.

[20] Noah Apthorpe, Danny Yuxing Huang, Dillon Reisman, Arvind Narayanan, and Nick Feamster. Keeping the smart home private with smart(er) IoT traffic shaping. *Proceedings on Privacy Enhancing Technologies (PoPETS)*, 2019(3), 2019.

[21] Frederik Möllers. Energy-efficient dummy traffic generation for home automation systems. *Proceedings on Privacy Enhancing Technologies (PoPETS)*, 2020(4), 2020.

[22] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and Selcuk Uluagac. Peek-a-boo: I see your smart home activities, even encrypted! In *13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec, 2020.

[23] Haotian Chi, Qiang Zeng, Xiaojiang Du, and Lannan Luo. Pfirewall: Semantics-aware customizable data flow

control for home automation systems. *arXiv preprint arXiv:1910.07987*, 2019.

[24] Yuchen Zhao, Hamed Haddadi, Severin Skillman, Shirin Enshaeifar, and Payam Barnaghi. Privacy-preserving activity and health monitoring on databox. In *3rd ACM International Workshop on Edge Systems, Analytics and Networking*, EdgeSys, 2020.

[25] Hossein Shafagh, Lukas Burkhalter, Sylvia Ratnasamy, and Anwar Hithnawi. Droplet: Decentralized authorization and access control for encrypted data streams. In *29th USENIX Security Symposium*, 2020.

[26] Trinabh Gupta, Rayman Preet Singh, Amar Phanishayee, Jaeyeon Jung, and Ratul Mahajan. Bolt: Data management for connected homes. In *11th USENIX Symposium on Networked Systems Design and Implementation*, NSDI, 2014.

[27] Earlence Fernandes, Justin Paupore, Amir Rahmati, Daniel Simionato, Mauro Conti, and Atul Prakash. Flowfence: Practical data protection for emerging iot application frameworks. In *25th USENIX security symposium*, 2016.

[28] Igor Zavalyshyn, Nuno O Duarte, and Nuno Santos. Homepad: A privacy-aware smart hub for home environments. In *2018 IEEE/ACM Symposium on Edge Computing*, SEC, 2018.

[29] Amir Rahmati, Earlence Fernandes, Kevin Eykholt, and Atul Prakash. Tyche: A risk-based permission model for smart homes. In *2018 IEEE Cybersecurity Development*, SecDev. IEEE, 2018.

[30] Yuan Tian, Nan Zhang, Yueh-Hsun Lin, XiaoFeng Wang, Blase Ur, Xianzheng Guo, and Patrick Tague. SmartAuth: User-centered authorization for the internet of things. In *26th USENIX Security Symposium*, 2017.

[31] Z Berkay Celik, Patrick McDaniel, and Gang Tan. Soteria: Automated IoT safety and security analysis. In *2018 USENIX Annual Technical Conference*, 2018.

[32] Wei Zhang, Yan Meng, Yugeng Liu, Xiaokuan Zhang, Yinqian Zhang, and Haojin Zhu. Homonit: Monitoring smart home apps from encrypted traffic. In *ACM SIGSAC Conference on Computer and Communications Security*, CCS, 2018.

[33] Wenbo Ding and Hongxin Hu. On the safety of iot device physical interaction control. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 832–846, 2018.

[34] Z Berkay Celik, Gang Tan, and Patrick D McDaniel. IoTGuard: Dynamic enforcement of security and safety policy in commodity IoT. In *Network and Distributed System Security Symposium*, NDSS, 2019.

[35] Amit Kumar Sikder, Leonardo Babun, Z Berkay Celik, Abbas Acar, Hidayet Aksu, Patrick McDaniel, Engin Kirda, and A Selcuk Uluagac. Kratos: Multi-user multi-device-aware access control system for the smart home. In *13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec, 2020.

[36] Z Berkay Celik, Leonardo Babun, Amit Kumar Sikder, Hidayet Aksu, Gang Tan, Patrick McDaniel, and A Selcuk Uluagac. Sensitive information tracking in commodity IoT. In *27th USENIX Security Symposium*, 2018.

[37] Iulia Bastys, Musard Balliu, and Andrei Sabelfeld. If this then what? controlling flows in IoT apps. In *ACM SIGSAC conference on computer and communications security*, CCS, 2018.

[38] Qi Wang, Wajih Ul Hassan, Adam Bates, and Carl Gunter. Fear and logging in the internet of things. In *Network and Distributed Systems Symposium*, NDSS, 2018.

[39] Leonardo Babun, Amit Kumar Sikder, Abbas Acar, and A Selcuk Uluagac. Iotdots: A digital forensics framework for smart environments. *arXiv preprint arXiv:1809.00745*, 2018.

[40] Kenji Yoshigoe, Wei Dai, Melissa Abramson, and Alexander Jacobs. Overcoming invasion of privacy in smart home environment with synthetic packet injection. In *2015 TRON Symposium*, TRONSHOW. IEEE, 2015.

[41] Anna Kornfeld Simpson, Franziska Roesner, and Tadayoshi Kohno. Securing vulnerable home iot devices with an in-hub security manager. In *Workshops of the International Conference on Pervasive Computing and Communications Workshops*, PerCom Workshops. IEEE, 2017.

[42] Chenghao Ye, Praburam Prabhakar Indra, and David Aspinall. Retrofitting security and privacy measures to smart home devices. In *Sixth International Conference on Internet of Things: Systems, Management and Security*, IOTSMS. IEEE, 2019.

[43] openHAB: empowering the smart home. https://www.openhab.org, 2021. Accessed: June 2022.

[44] Yilei Liang, Dan O'Keeffe, and Nishanth Sastry. PAIGE: towards a hybrid-edge design for privacy-preserving intelligent personal assistants. In *Third ACM International Workshop on Edge Systems, Analytics and Networking*, EdgeSys, 2020.

[45] Google Home. https://developers.google.com/home, 2021. Accessed: June 2022.

[46] Apple Home Kit: Developing Apps and Accessories for the Home. https://developer.apple.com/homekit/, 2021. Accessed: June 2022.

[47] Smart Lighting: Philips Hue. https://www.philips-hue.com/en-us, 2022. Accessed: June 2022.

[48] IFTTT: Developers page. https://ifttt.com/developers, 2021. Accessed: June 2022.

[49] Home Assistant: Open source home automation that puts local control and privacy first. https://www.home-assistant.io, 2021. Accessed: June 2022.

[50] Domoticz: Home automation system. https://www.domoticz.com, 2021. Accessed: June 2022.

[51] Home Genie: The open source, programmable, home automation server for smart connected devices and applications. https://homegenie.it, 2021. Accessed: June 2022.

[52] Homekit data security. https://support.apple.com/guide/security/homekit-data-security-sec49613249e/web, 2021. Accessed: June 2022.

[53] Philips hue: Get started. https://developers.meethue.com/develop/get-started-2/, 2022. Accessed: June 2022.

[54] Chenggang Wang, Sean Kennedy, Haipeng Li, King Hudson, Gowtham Atluri, Xuetao Wei, Wenhai Sun, and Boyang Wang. Fingerprinting encrypted voice traffic on smart speakers with deep learning. In *13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec, 2020.

[55] Natasha Lomas. Google ordered to halt human review of voice ai recordings over privacy risks. https://techcrunch.com/2019/08/02/google-ordered-to-halt-human-review-of-voice-ai-recordings-over-privacy-risks/, 2019. Accessed: June 2022.

[56] Jon Brodkin. Google workers listen to your "ok google" queries—one of them leaked recordings. https://arstechnica.com/information-technology/2019/07/google-defends-listening-to-ok-google-queries-after-voice-recordings-leak/, 2019. Accessed: June 2022.

[57] Alex Hern. Amazon staff listen to customers' alexa recordings, report says. https://www.theguardian.com/technology/2019/apr/11/, 2019. Accessed: June 2022.

[58] Colin Dixon, Ratul Mahajan, Sharad Agarwal, AJ Brush, Bongshin Lee, Stefan Saroiu, and Paramvir Bahl. An operating system for the home. In *9th USENIX Symposium on Networked Systems Design and Implementation*, NSDI, 2012.

[59] Reza Rawassizadeh, Timothy J Pierson, Ronald Peterson, and David Kotz. NoCloud: Exploring network disconnection through on-device data analysis. *IEEE Pervasive Computing*, 17(1):64–74, 2018.

[60] Han Zhang, Abhijith Anilkumar, Matt Fredrikson, and Yuvraj Agarwal. Capture: Centralized library management for heterogeneous iot devices. In *USENIX Security Symposium*, 2021.

[61] Varun Chandrasekaran, Suman Banerjee, Bilge Mutlu, and Kassem Fawaz. PowerCut and Obfuscator: An exploration of the design space for privacy-preserving interventions for smart speakers. In *Seventeenth Symposium on Usable Privacy and Security*, SOUPS, 2021.

[62] Batyr Charyyev and Mehmet Hadi Gunes. Misactivation detection and user identification in smart home speakers using traffic flow features. In *14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec, 2021.

[63] Rohan Doshi, Noah Apthorpe, and Nick Feamster. Machine learning DDoS detection for consumer internet of things devices. In *Workshops of the IEEE Security and Privacy*, S&P workshops, 2018.

[64] Anand Mudgerikar, Puneet Sharma, and Elisa Bertino. E-spion: A system-level intrusion detection system for IoT devices. In *ACM Asia conference on computer and communications security*, AsiaCCS, 2019.

[65] Sanket Goutam, William Enck, and Bradley Reaves. Hestia: simple least privilege network policies for smart homes. In *12th Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec, 2019.

[66] TJ OConnor, Reham Mohamed, Markus Miettinen, William Enck, Bradley Reaves, and Ahmad-Reza Sadeghi. HomeSnitch: behavior transparency and control for smart home IoT devices. In *12th Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec, 2019.

[67] Ryan Heartfield, George Loukas, Anatolij Bezemskij, and Emmanouil Panaousis. Self-configurable cyber-physical intrusion detection for smart homes using reinforcement learning. *IEEE Transactions on Information Forensics and Security*, 16:1720–1735, 2020.

[68] Abraham Mhaidli, Manikandan Kandadai Venkatesh, Yixin Zou, and Florian Schaub. Listen only when spoken to: Interpersonal communication cues as smart speaker privacy controls. *Proceedings on Privacy Enhancing Technologies (PoPETS)*, 2020(2):251–270, 2020.

[69] Xinyu Lei, Guan-Hua Tu, Chi-Yu Li, Tian Xie, and Mi Zhang. SecWIR: securing smart home IoT communications via wi-fi routers with embedded intelligence. In *18th International Conference on Mobile Systems, Applications, and Services*, MobiSys, 2020.

[70] Judson Wilson, Riad S Wahby, Henry Corrigan-Gibbs, Dan Boneh, Philip Levis, and Keith Winstein. Trust but verify: Auditing the secure internet of things. In *15th International Conference on Mobile Systems, Applications, and Services*, MobiSys, 2017.

[71] Rahmadi Trimananda, Ali Younis, Bojun Wang, Bin Xu, Brian Demsky, and Guoqing Xu. Vigilia: Securing smart home edge computing. In *2018 IEEE/ACM Symposium on Edge Computing*, SEC. IEEE, 2018.

[72] Earlence Fernandes, Amir Rahmati, Jaeyeon Jung, and Atul Prakash. Decentralized action integrity for trigger-action iot platforms. In *Network and Distributed System Security Symposium*, NDSS, 2018.

[73] Yunang Chen, Amrita Roy Chowdhury, Ruizhe Wang, Andrei Sabelfeld, Rahul Chatterjee, and Earlence Fernandes. Data privacy in trigger-action systems. In *2021 IEEE Symposium on Security and Privacy*, S&P, 2021.

[74] Silvery Fu and Sylvia Ratnasamy. DSpace: Composable abstractions for smart spaces. In *ACM SIGOPS 28th Symposium on Operating Systems Principles*, SOSP, 2021.

[75] Igor Zavalyshyn, Nuno Santos, Ramin Sadre, and Axel Legay. My House, My Rules: A private-by-design smart home platform. In *17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous, 2020.

[76] Shegufta B Ahsan, Rui Yang, Shadi A Noghabi, and Indranil Gupta. Home, safehome: smart home reliability with visibility and atomicity. In *Sixteenth European Conference on Computer Systems*, EuroSys, 2021.

[77] Joshua Siegel and Sanjay Sarma. A cognitive protection system for the internet of things. *IEEE Security & Privacy*, 17(3):40–48, 2019.

[78] Anna Maria Mandalari, Daniel J Dubois, Roman Kolcun, Muhammad Talha Paracha, Hamed Haddadi, and David Choffnes. Blocking without breaking: Identification and mitigation of non-essential iot traffic. *Proceedings on Privacy Enhancing Technologies (PoPETS)*, 2021(4):369–388, 2021.

[79] Nigel Davies, Nina Taft, Mahadev Satyanarayanan, Sarah Clinch, and Brandon Amos. Privacy mediators: Helping iot cross the chasm. In *17th International Workshop on Mobile Computing Systems and Applications*, HotMobile, 2016.

[80] Ismini Psychoula, Liming Chen, and Oliver Amft. Privacy risk awareness in wearables and the internet of things. *IEEE Pervasive Computing*, 19(3):60–66, 2020.

[81] Tavish Vaidya and Micah Sherr. You talk too much: Limiting privacy exposure via voice input. In *Workshops of the IEEE Security and Privacy*, S&P workshops. IEEE, 2019.

[82] Victor Costan and Srinivas Devadas. Intel SGX explained. *IACR Cryptol. ePrint Arch.*, 2016(86):1–118, 2016.

[83] European Commission. General data protection regulation art. 5: Principles relating to processing of personal data. https://gdpr-info.eu/art-5-gdpr/.

[84] State of California. Senate bill no. 1121: California consumer privacy act.

[85] Eric Zeng and Franziska Roesner. Understanding and improving security and privacy in multi-user smart homes: a design exploration and in-home user study. In *28th USENIX Security Symposium*, 2019.

[86] Christine Geeng and Franziska Roesner. Who's in control? interactions in multi-user smart homes. In *ACM CHI Conference on Human Factors in Computing Systems*, CHI, 2019.

[87] Amanda Silberling. New amazon echo devices will have local voice processing, giving users more privacy. https://techcrunch.com/2021/09/28/new-amazon-echo-devices-will-have-local-voice-processing-giving-users-more-privacy/, 2021. Accessed: June 2022.

[88] Smartthings edge provides reliable, faster smart home experiences. https://blog.smartthings.com/tag/smartthings-edge/, 2021. Accessed: June 2022.

[89] Announcement | changes to our legacy smartthings platform. https://community.smartthings.com/t/announcement-changes-to-our-legacy-smartthings-platform/197958, 2020. Accessed: June 2022.

[90] Hossein Shafagh, Anwar Hithnawi, Lukas Burkhalter, Pascal Fischli, and Simon Duquennoy. Secure sharing of partially homomorphic encrypted iot data. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, SenSys '17, New York, NY, USA, 2017. Association for Computing Machinery.

[91] Hossein Shafagh, Anwar Hithnawi, Andreas Dröscher, Simon Duquennoy, and Wen Hu. Talos: Encrypted query processing for the internet of things. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*, pages 197–210, 2015.

[92] Jaweher Zouari, Mohamed Hamdi, and Tai-Hoon Kim. A privacy-preserving homomorphic encryption scheme for the internet of things. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1939–1944. IEEE, 2017.

[93] Marin Matsumoto and Masato Oguchi. Speeding up encryption on iot devices using homomorphic encryption. In *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 270–275, 2021.

[94] Jacob Marks, Brandon Montano, Jiwan Chong, Manjusha Raavi, Raisa Islam, Tomas Cerny, and Dongwan Shin. Differential privacy applied to smart meters: a mapping study. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pages 761–770, 2021.

[95] Ahmed Alshehri, Jacob Granley, and Chuan Yue. Attacking and protecting tunneled traffic of smart home devices. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, pages 259–270, 2020.

[96] Vincent Bindschaedler, Muhammad Naveed, Xiaorui Pan, XiaoFeng Wang, and Yan Huang. Practicing oblivious access on cloud storage: the gap, the fallacy, and the new way forward. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 837–849, 2015.

[97] Ayyoob Hamza, Dinesha Ranathunga, Hassan Habibi Gharakheili, Matthew Roughan, and Vijay Sivaraman. Clear as mud: Generating, validating and applying iot behavioral profiles. In *ACM SIGCOMM Workshop on IoT Security and Privacy*, WISP, 2018.

[98] Noura Abdi, Kopo M Ramokapane, and Jose M Such. More than smart speakers: security and privacy perceptions of smart home personal assistants. In *Fifteenth Symposium on Usable Privacy and Security*, SOUPS, 2019.

[99] Abdulmajeed Alqhatani and Heather Richter Lipford. "there is nothing that i need to keep secret": Sharing practices and concerns of wearable fitness data. In *Fifteenth Symposium on Usable Privacy and Security*, SOUPS, pages 421–434, 2019.

[100] MW Raad and Laurence Tianruo Yang. A ubiquitous smart home for elderly. *Information Systems Frontiers*, 11(5):529, 2009.

[101] Amaya Arcelus, Megan Howell Jones, Rafik Goubran, and Frank Knoefel. Integration of smart home technologies in a health monitoring system for the elderly. In *Workshops of the 21st International Conference on Advanced Information Networking and Applications*, volume 2 of *AINAW*. IEEE, 2007.

[102] Alisa Frik, Leysan Nurgalieva, Julia Bernd, Joyce Lee, Florian Schaub, and Serge Egelman. Privacy and security threat models and mitigation strategies of older adults. In *Fifteenth Symposium on Usable Privacy and Security*, SOUPS, 2019.

[103] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063, 2020.

[104] Ulrich Matchi Aïvodji, Sébastien Gambs, and Alexandre Martin. IOTFLA: A secured and privacy-preserving smart home architecture implementing federated learning. In *Workshops of IEEE Security and Privacy*, S&P workshops, 2019.

[105] Omar Alrawi, Chaz Lever, Manos Antonakakis, and Fabian Monrose. SoK: Security evaluation of home-based IoT deployments. In *2019 IEEE symposium on security and privacy*, S&P, 2019.

[106] Weijia He, Valerie Zhao, Olivia Morkved, Sabeeka Siddiqui, Earlence Fernandes, Josiah Hester, and Blase Ur. SoK: Context sensing for access control in the adversarial home IoT. In *IEEE European Symposium on Security and Privacy*, EuroS&P. IEEE, 2021.

[107] Leonardo Babun, Kyle Denney, Z Berkay Celik, Patrick McDaniel, and A Selcuk Uluagac. A survey on IoT platforms: Communication, security, and privacy perspectives. *Computer Networks*, 192, 2021.

Appendix

Venue acronym	Full name of the venue	Period covered
Conferences		
AsiaCCS	ACM Asia Conference on Computer and Communications Security	2010-2021
CCS	ACM SIGSAC Conference on Computer and Communications Security	2010-2021
DSN	Annual IEEE/IFIP International Conference on Dependable Systems and Networks	2010-2021
EuroSys	European Conference on Computer Systems	2010-2021
Mobiquitous	EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services	2010-2021
MobiSys	Annual International Conference on Mobile Systems, Applications, and Services	2010-2021
NDSS	Annual Network and Distributed System Security Symposium	2010-2021
NSDI	USENIX Symposium on Networked Systems Design and Implementation	2010-2021
OSDI	USENIX Symposium on Operating Systems Design and Implementation	2010-2021
S&P	IEEE Symposium on Security and Privacy	2010-2021
SEC	ACM/IEEE Symposium on Edge Computing	2016-2021
SOSP	ACM Symposium on Operating Systems Principles	2011-2021
SOUPS	Symposium On Usable Privacy and Security	2010-2021
USENIX Sec	USENIX Security Symposium	2010-2021
WiSec	ACM Conference on Security and Privacy in Wireless and Mobile Networks	2012-2021
Workshops		
DSN-W	Workshops of the Annual IEEE/IFIP International Conference on Dependable Systems and Networks	2010-2021
EdgeSys	International Workshop on Edge Systems, Analytics and Networking (co-located with EuroSys)	2019-2021
HotMobile	International Workshop on Mobile Computing Systems and Applications	2010-2021
HotSec	USENIX Workshop on Hot Topics in Security (co-located with USENIX Sec)	2010-2012
IoTS&P	Workshop on Internet of Things Security and Privacy (co-located with CCS)	2017,2019
S&PW	IEEE Security and Privacy Workshops (co-located with S&P)	2012-2021
Journals		
CSUR	ACM Computing Surveys	2010-2021
IEEES&P	IEEE Security & Privacy	2010-2021
IMWUT	ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies	2017-2021
PACMHCI	ACM on Human-Computer Interaction	2017-2021
Pervasive	IEEE Pervasive Computing	2010-2021
PoPETs	International Symposium on Privacy Enhancing Technologies	2010-2021
TDSC	IEEE Transactions on Dependable and Secure Computing	2010-2021
TIFS	IEEE Transactions on Information Forensics and Security	2010-2021
TOPS	ACM Transactions on Privacy and Security	2010-2021
Additional venues selected through keyword search in bibliographic databases		
TRON	TRON Symposium	2015
PerCom-W	IEEE International Conference on Pervasive Computing and Communications Workshops	2017
IOTSMS	Sixth International Conference on Internet of Things: Systems, Management and Security	2019

Table 3. Venues selected for the proceedings analysis and the range of years covered.