A Survey on Sensor-Based Threats and Attacks to Smart Devices and Applications

Amit Kumar Sikder, *Member, IEEE*, Giuseppe Petracca, Hidayet Aksu, Trent Jaeger, and A. Selcuk Uluagac

***Abstract*—Modern electronic devices have become "smart" as well as omnipresent in our day-to-day lives. From small household devices to large industrial machines, smart devices have become very popular in every possible application domain. Smart devices in our homes, offices, buildings, and cities can connect with other devices as well as with the physical world around them. This increasing popularity has also placed smart devices as the center of attention among attackers. Already, several types of malicious activities exist that attempt to compromise the security and privacy of smart devices. One interesting and noteworthy emerging threat vector is the attacks that abuse the use of *sensors* on smart devices. Smart devices are vulnerable to *sensor-based threats and attacks* due to the lack of proper security mechanisms available to control the use of sensors by installed apps. By exploiting the sensors (e.g., accelerometer, gyroscope, microphone, light sensor, etc.) on a smart device, attackers can extract information from the device, transfer malware to a device, or trigger a malicious activity to compromise the device. In this paper, we explore various threats and attacks abusing sensors of smart devices for malicious purposes. Specifically, we present a detailed survey about existing sensor-based threats and attacks to smart devices and countermeasures that have been developed to secure smart devices from sensor-based threats. Furthermore, we discuss security and privacy issues of smart devices in the context of sensor-based threats and attacks and conclude with future research directions.**

***Index Terms*—Sensor-based threats, smart devices, Internet-of-Things, security, sensors.**

I. Introduction

SMART devices such as smartphones, smart watches, smart lights, smart locks, etc. have become very popular in recent years. With the tremendous growth of Internet of Things (IoT), smart devices now have advanced capabilities to interact with other devices and also with human beings and its surrounding physical world to perform a myriad of tasks [1]. In this context, the use of sensors on smart devices enables a seamless connection between the devices and the physical world. Indeed, modern smart devices come with a wide range of sensors (e.g., accelerometer, gyroscope, microphone, light sensor, etc.) that enable more efficient and user-friendly applications [2]. These sensors introduce features such as context-awareness, self-learning, and automation which improve the applicability of smart devices in various application domains [3], [4]. From personal healthcare to home appliances, from big industrial applications to smart cities, smart devices are in every possible application domain. The increasing popularity and utility of these devices in diverse application domains made the device industry grow at a tremendous rate. According to a report by Statista and Forbes, there will be 3.5 billion smart devices by 2024 with a penetration rate of 52.4% and more than 152,000 smart devices will be connected to the Internet every minute in 2025 [5], [6].

The use of sensors in smart devices inevitably increases the functionality of the devices; however, the sensors can also be used as vehicles to launch attacks on the devices or applications. Recently, there have been several attempts to exploit the security of smart devices via their sensors [7]–[9]. Attackers can use the sensors to *transfer malicious code* or *a trigger message* to activate malware planted in a device [10], [11], *capture sensitive personal information* shared between devices (e.g., smartwatches, smart home devices, etc.) [12]–[15], or even *extract encrypted information* by capturing encryption and decryption keys [16]. Moreover, attackers can use the sensors of one device as an attack platform to abuse or interrupt normal functionalities of connected devices [17]. These sensor-based threats pose a significant risk to the smart devices as manufacturers are not fully aware yet [18]. Indeed, sensor-based threats are becoming more prevalent because of the easy access to the sensors and the limited security measures that consider these threats [19]–[22].

Furthermore, attackers do not need any complicated tools to access the sensors, which make sensor-based threats easier to execute [13], [23]. Existing studies have verified that it is possible to execute sensor-based attacks without impeding the normal functions of devices. Also, there have been several real-life malware reported recently which use sensors as a means of performing malicious activities on smart devices [24]. For example, TrendMicro, a renowned security company, reported in 2019 three publicly available Android apps in Google Playstore used the motion sensor to evade malware scanners in the smartphone [25]. When a user performs any task such as making calls or texting in the smartphone, it

Manuscript received September 26, 2019; revised April 24, 2020, September 14, 2020, and January 8, 2021; accepted February 8, 2021. Date of publication March 8, 2021; date of current version May 21, 2021. This work was supported in part by the U.S. National Science Foundation under Award NSF-CAREER-CNS-1453647 and Award NSF-CNS-1718116; and in part by the Army Research Laboratory and was accomplished through Cooperative Agreement under Grant W911NF-13-2-0045 (ARL Cyber Security CRA). *(Corresponding author: Amit Kumar Sikder.)*

Amit Kumar Sikder, Hidayet Aksu, and A. Selcuk Uluagac are with the Cyber-Physical Systems Security Lab, Electrical and Computer Engineering Department, Florida International University, Miami, FL 33174 USA (e-mail: asikd003@fiu.edu; haksu@fiu.edu; suluagac@fiu.edu).

Giuseppe Petracca and Trent Jaeger are with the Systems and Internet Infrastructure Security Laboratory, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16801 USA (e-mail: gxp18@cse.psu.edu; tjaeger@cse.psu.edu).

Digital Object Identifier 10.1109/COMST.2021.3064507

creates deviation in the motion sensor. These malicious apps check the motion sensor data to determine whether the app is running in a real environment or in a sandbox environment of a malware scanner. Based on the detected sandbox environment, the malware stop executing malicious code and perform the normal operation. On the contrary, the malware inject malicious banking trojan such as *Anubis* and *Cerberus* upon detecting a real-life environment with user interaction. Hence, trivial execution, easy access to the sensors, and lack of knowledge about the sensor-based threats constitute significant risks for the smart devices.

Researchers have proposed several countermeasures such as enhancing permissions for sensor access, information flow analysis, etc. to improve the security of smart devices against sensor-based threats [26], [27]. However, these proposed solution depend on either user decision or availability of the source code of the apps. Moreover, the majority of these app-specific solutions cannot envision the passive sensor-based threats such as eavesdropping, triggering and transferring malware using sensors, abusing sensors using interference, etc. For instance, a specific on/off pattern of a smart light can trick a smart camera to capture and leak pictures containing sensitive information in a smart home environment [28]. Thus, understanding these sensor-based threats and attacks in the literature is necessary for researchers and the community to design reliable solutions to detect and prevent these threats efficiently.

*Contributions—*In this paper, we present a detailed survey of threats that can be exploited to attack sensors in smart devices. Several prior works have mentioned sensor-based threats as one of the emerging threat vectors to the smart devices [29], [30]. In particular, previous works have included sensor-based threats in the threat taxonomy as a general threats to specific smart devices such as smartphones [31] and discussed major drawbacks on the operating system (OS) level [32]. However, no taxonomy and impact analysis of sensor-based threats and attacks is provided in these works. Compared to the prior works, we conduct a detailed survey of the existing sensor-based threats to smart devices and provide a formal taxonomy of sensor-based threats to understand the attack mechanisms and effects on smart devices. We also introduce common vulnerability metrics to perform impact analysis of sensor-based threats to smart devices. Furthermore, we present a taxonomy of existing solutions that specifically focus on mitigating sensor-based threats and outline future research directions in terms of sensor security in smart devices. In summary, the contributions of this paper are:

- *First*, we present a detailed discussion regarding the security goals and requirements to protect smart devices from sensor-based threats and identify the important shortcomings of the existing systems.
- *Second*, we provide a detailed taxonomy of sensor-based threats and attacks to smart devices and discuss the mechanisms and effectiveness of the attacks in a detailed way. We also summarize the effectiveness of the threats and attacks based on known vulnerability metrics.
- *Third*, we discuss the proposed security solutions by the research community and developers for smart devices and their shortcomings in the context of sensor-based threats and attacks.
- *Fourth*, we identify several open issues and discuss future research that could contribute to secure smart devices against emerging sensor-based threats.

*Organization—*The rest of the paper is organized as follows. We discuss related works in Section II. In Section III, we give the definition and general architecture of smart devices. In Section IV, we briefly discuss the security goals and requirements to protect sensors in smart devices and how existing systems address these goals. We also summarize the shortcomings of existing systems in detecting sensor-based threats. In Section V, we classify the sensor-based threats and attacks based on key security principles and explain our scope of work. We present existing sensor-based threats and attacks in Section VI and summarize attack methods and impact of the threats based on different vulnerability metrics. In Section VII, we articulate approaches that have been proposed to secure sensors of smart devices and their shortcomings to detect reported sensor-based threats and attacks. Future research in the area of sensor-based threats and security of smart devices are described in Section VIII. Finally, we conclude this paper in Section IX.

II. Related Work

In recent years, several surveys and tutorials have been published covering different threats and defense mechanisms of smart devices and applications [31], [33]. However, these works either focus on traditional network-based threats or system vulnerabilities generated from flawed frameworks. In addition, prior works investigate security and privacy issues of wireless sensor networks focusing on communication level threats [34].

The majority of existing surveys and tutorials focus on explaining the security and privacy issues of smart devices in a generalized way overlooking the detail explanation of sensor-based threats. However, the generalized discussion and categorization of security and privacy issues presented in prior works cannot illustrate the detailed attack surface of sensor-based threats including attack methods, attack types, targeted sensors, and attack impacts. Suarez-Tangil *et al.* investigated malware evolution in smart devices and categorized the existing malware detection techniques in seven broad categories including the type of detection, type of analysis, targeted malware, etc., [31]. Yang *et al.* surveyed the general security and privacy issues of IoT devices and categorized the threats to four categories - physical/perception, network, software, and encryption attacks [35]. The authors categorized sensor-based threats under physical/perception attacks and discussed threats arising from tampered sensors and unencrypted communications only. Ammar *et al.* presented a tutorial work on security and privacy issues of IoT programming platforms such as AWS IoT, Azure, etc. and explained how security features on programming platforms are adapted in smart apps [36]. The authors mainly focused on security issues arising from authentication, access control, and secure communication in IoT layers. Khan and Shah surveyed different threats to smartphones in IoT ecosystem [37]. Cao *et al.* presented a detailed survey of network-based threats to cyber-physical systems and discussed threats on perception and sensing layer arising from malicious communication between CPS devices [38]. Hassija *et al.* summarized security threats to IoT devices and categorizes the threats based on working layers [39]. Bhat and Dutta reviewed existing threats and mitigation techniques in Android-operated smart devices and listed several sensor-based threats that exploits OS-level authorization [32]. In a recent work, Li *et al.* surveyed adversarial threats to CPS and IoT devices and categorizes adversarial threats to sensors of smart devices [29]. The authors only considered adversarial examples that exploit sensors in IoT and CPS devices and discussed how adversarial inputs can manipulate sensor data in machine learning-based models.

Security and privacy issues arising from communication protocols in smart devices have been highlighted in several surveys and tutorials. However, these works only cover sensor-based threats targeting the communication medium overlooking other sensor-based threats (e.g., keystroke inference, false sensor data, etc., [15], [40]). Dragomir *et al.* reviewed several security threats of communication protocols for IoT systems and summarized several network vulnerabilities of IoT sensor networks [41]. Tomic and McCann surveyed protocol level vulnerabilities in wireless sensor networks including eavesdropping, node tampering, and hardware threats [42]. Ngu *et al.* reviewed different aspects and issues of IoT middleware and summarized threats to sensor communication in IoT ecosystem [43]. The authors discussed various threats to sensors that emerged from IoT middleware communication such as sensor to cloud communication, sensor computing in the cloud, Web services connected with sensors, etc. Another interesting work was presented by Polla *et al.* where authors mentioned sensor sniffing as a severe threat to mobile devices (e.g., smartphone, smart watch, etc.) [44]. Caprolu *et al.* investigated security and privacy aspects of short-ranged audio channel and considered sensor eavesdropping as one of the major threats to smart devices [45]. Neshenko *et al.* outlined security requirements in IoT network by introducing a layer and security-based attack taxonomy [30]. Authors discussed existing flaws in IoT devices and platforms and pointed out several key challenges to improve the security of IoT devices. Hamad *et al.* conducted a survey to discuss security flaws in IoT devices and architecture and summarized recent advancements in security services in IoT cloud [46]. In another work, Sengupta *et al.* summarized blockchain-based security solutions to address security and privacy threats in IoT devices [47]. Newaz *et al.* surveyed security and privacy issues in emerging health IoT devices and applications [48]. Recently, Yan *et al.* presented a generalized approach to analyze security of analog sensors and presented existing vulnerabilities [49]. Besides these works, several prior works have summarized the security and privacy issues of smart devices and discussed future research directions to resolve these issues [50]–[54].

Several prior works have also surveyed existing security solutions proposed by the research community and developers to address the security and privacy issues of smart devices. However, there is no survey exploring security solutions addressing sensor-based threats to smart devices. Yan *et al.* surveyed trust management frameworks for IoT devices and discussed several approaches for sensor enriched networks [55]. Chaabouni *et al.* summarized existing network-based intrusion detection methods for IoT devices including detection methods for sensor-based threats using network traffic [56]. Butun *et al.* presented a detailed overview of intrusion detection systems (host-based and network-based) for wireless sensor networks for traditional network-based attacks [57]. Another recent work updated the aforementioned work with current trends of intrusion detection systems in wireless sensor networks [58]. In addition, Butun *et al.* surveyed existing issues of wireless sensor networks in IoT devices and categorized network layer threats emerging from sensor nodes [59]. Al-garadi *et al.* surveyed machine learning-based security mechanisms for IoT devices and discussed challenges regarding security of IoT devices [60].

Differences From the Existing Works: All the aforementioned surveys and tutorials are very useful to review the security of smart devices from the network and operating system level. While several prior works [7], [10], [18] mentioned sensor-based threats as a security issue, no prior work investigated the sensor-based threats and attacks in-depth as in this paper. We acknowledge that several prior works included sensor-based threats as a variant of network-based and side-channel attacks in the attack taxonomy of mobile or IoT devices overlooking several existing sensor-based attacks [29], [30], [39], [45]. However, these prior works lack a method to organize these sensor-based attacks to help researchers understand the diverse attack surface and the effects of these threats on smart devices as shown in Table I. For instance, several existing surveys explored defense mechanisms for mobile and IoT devices to address network-based threats and platform-specific security inadequacies that fail to assess the specific requirements for securing sensors [30], [32], [37]. Instead, our work identifies the diverse sensor functionalities in smart devices (both mobile and IoT devices) and provides a detailed taxonomy of sensor-based threats to explain how each type of attack can be performed by targeting sensors in smart devices. We also introduce common vulnerability scoring metrics that provide a systematic approach to assess the severity of existing and future sensor-based threats based on the nature of the attacks, the attacker's capabilities, the privilege requirements, and their success rate in different smart devices. In summary, we have the following key differences from prior works-

- *Sensor Security Requirements:* In this survey, we identify shortcomings of existing smart device platforms in securing sensors and summarize key security requirements to enhance sensor security in smart devices.
- *Threat Taxonomy and Modeling:* Prior works have included sensor-based threats in the threat taxonomy and explained the attacks from network and OS-perspective. Compared to prior works, we introduce a formal threat model of sensor-based threats considering attack methods, effects, and capabilities.
- *Impact Analysis:* We introduce vulnerability metrics to analyze the impact of sensor-based threats on smart

TABLE I
COMPARISON BETWEEN EXISTING SURVEYS AND OUR SURVEY

	Targeted smart device		Smart device architecture	Existing sensor security schemes	Sensor-based threats			Solutions		Future directions for sensor security
	Mobile devices	IoT devices			Taxonomy	Vulnerability metrics	Attack methods and impact	Taxonomy	Detail approach	
Suarez-Tangil et al. [31]	●	●	○	●	○	○	◑	○	◑	○
Yang et al. [35]	○	●	○	●	○	○	◑	○	○	○
Ammar et al. [36]	○	●	○	●	○	○	○	○	○	●
Dragomir et al. [41]	○	●	○	○	○	○	◑	○	○	●
Tomic et al. [42]	○	●	○	○	○	○	◑	○	◑	○
Ngu et al. [43]	●	●	○	○	○	○	◑	○	○	●
Polla et al. [44]	●	○	○	○	○	○	◑	○	○	○
Khan et al. [37]	●	●	○	○	○	○	◑	○	◑	○
zhao et al. [52]	○	●	●	●	○	○	○	○	○	○
granjal et al. [51]	○	●	○	●	○	○	◑	○	○	●
chen et al. [53]	●	●	○	●	○	○	◑	○	◑	●
Yan et al. [55]	○	●	●	●	○	○	○	○	◑	○
chaabouni et al. [56]	○	●	●	●	○	○	○	○	◑	●
Butun et al. [59]	○	●	○	○	○	○	◑	○	◑	●
Li et al. [29]	○	●	○	○	◑	○	◑	○	◑	●
Bhat et al. [32]	●	○	●	●	○	○	◑	○	◑	●
Cao et al. [38]	○	●	●	○	◑	○	◑	○	◑	●
Neshenko et al. [30]	○	●	●	○	◑	○	◑	◑	◑	●
Hassija et al. [39]	○	●	●	○	◑	○	◑	○	◑	●
Caprolu et al. [45]	●	●	○	○	◑	○	◑	○	○	○
Hamza et al. [61]	○	●	○	○	◑	○	◑	○	◑	○
Our survey	●	●	●	●	●	●	●	●	●	●

Topic covered by the work: ●; Topic partially covered by the work: ◑; Topic is not covered in the work: ○

devices. We consider seven vulnerability metrics including attack method, device access, attack complexity, privilege, user interaction, security impact, and success rate to understand the overall impact of each reported sensor-based attack.

- *Taxonomy of Existing Security Mechanisms:* Several prior works have explored existing security mechanisms available for smart devices and summarized how they enhance the overall security of smart devices. Compared to these works, we provide a taxonomy of existing security mechanisms that explicitly address sensor-based threats to smart devices. We explain how each security mechanism enhances sensor security in smart devices and outlines their shortcomings based on reported sensor-based threats. We also report solution types, device platforms, dependencies, and overhead to discuss shortcomings of existing solutions in detecting sensor-based threats.
- *Future Research Directions:* Prior works have outlined future research directions to enhance the security of smart devices which includes general recommendations for improving sensor security. However, a detailed guideline is needed to address sensor-based threats on smart devices and develop effective security measures. In this work, we outline the key open issues in smart device platforms that have triggered the growth of sensor-based threats in recent years and summarize the futures research direction to enhance sensor security in smart devices.

The differences between existing surveys and our work are summarized in Table I.

III. BACKGROUND: COMPONENTS OF SMART DEVICES

In this section, we introduce the components of smart devices as they are relevant to understand the significance of sensor-based threats and attacks.

In general, a smart device is an electronic device which has the capability to connect, share, and interact with its user, peripheral, and other smart devices using their sensors and communication protocols [31], [62], [63]. These devices in general have the following salient features:

- *Sensing* - Smart devices use sensors to sense the surrounding environment and perform different tasks based on measured events.[1]
- *Automation* - Automation is the ability of a smart device to perform a task automatically based on specific events. For example, the user presence in a room can cause a sensor to turn a light on and off.
- *Accessibility* - Smart devices offer easy and remote accessibility to the users. Users can control and monitor a smart device from a remote location. For example, users can lock or unlock a smart lock from a remote location using a smartphone.
- *Context-Awareness* - Context-awareness refers to the ability of a device to understand and analyze its surroundings. A smart device can understand what is happening around the device and perform different tasks accordingly. For example, a smart lock can automatically unlock as a specific person approaches the door. Here, the smart lock is aware of the user's presence in its proximity and unlocks the door.
- *Self-Learning* - Smart devices can learn usage patterns and change responses to perform different tasks without any manual instruction from the users. For example, a smart thermostat can learn the usage pattern of a user and adjust the temperature automatically to save power.

A smart device can have all or a subset of the aforementioned features. These features of smart devices are linked together with one common component- *sensors in smart devices*. For instance, an embedded temperature sensor can be used to trigger a smart thermostat at a pre-defined temperature which represents sensing, automation, and self-learning features of smart devices. Again, external sensors can be connected with smart devices using different communication protocols (e.g., ZigBee, Z-Wave, BLE, etc.) or via

[1] In some cases, sensors may be using or connected to actuators to perform tasks.

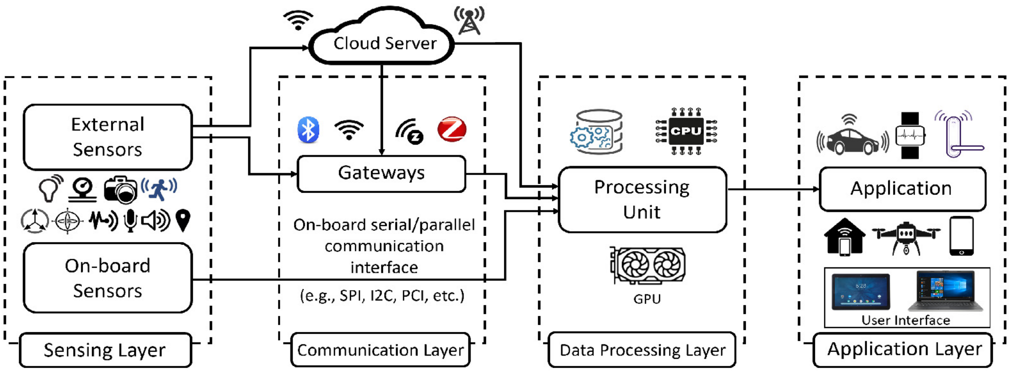


Fig. 1. Smart device architecture layers and components. Some smart devices may have all the layers (e.g., smart thermostat) or a subset of these layers (e.g., smart sensors).

cloud [64]. An external presence sensor can be configured with a smart thermostat to turn on whenever a user enters a room which depicts context-awareness. Users can also control smart devices remotely that can be associated with the embedded sensors to automate on-going tasks. Hence, sensors in smart devices are the key components that enable aforementioned salient features in smart devices.

Moreover, it is important to understand the smart device architecture to better visualize the threats and attacks. We illustrate the smart device architecture in four working layers (sensing, communication, data processing, and application) as shown in Figure 1. One unique feature of smart device architecture is the presence of sensing layer that connects surrounding environment with traditional computing layers (communication, data processing, and application layers) to automate different tasks [65]. Based on the functionalities, distinct smart devices may combine the sensing layer and the computing layers (network, processing, application) differently. For instance, a smart sensor can have sensing layer that combines with communication and data processing layer [66] to capture and analyze sensed data. However, a device with only sensing layer cannot be considered as a smart device since only sensing layer without any computing capabilities cannot provide any functionalities of smart devices (automation, context-awareness, self-learning, etc.) [63].

A. Sensing Layer

The main purpose of the sensing layer is to identify any phenomena in the devices' peripheral and obtain data from the real world. This layer consists of several sensors, where multiple sensors are typically used together by applications to collect various data [67]. The sensing layer of smart device ecosystem can consist of both on-device sensors and external independent sensors. In both cases, sensors are usually integrated through sensor hubs [68]. A sensor hub is a common connection point for multiple sensors that accumulate and forward sensor data to the processing unit of a device. A sensor hub may use several transport mechanisms (e.g., Inter-Integrated Circuit (I2C) or Serial Peripheral Interface (SPI)) for data flow between sensors and applications. For on-device sensors, the sensor hub uses Inter-Integrated Circuit (I2C) or Serial Peripheral Interface (SPI) to forward sensor data to the data processing layer. For external independent sensors, sensor data are forwarded to the cloud server from the sensor hub and smart devices can accumulate these data from the cloud server using the network layer. Sensors in smart devices can be classified into three broad categories (A detailed description of different sensors is given in Table II):

1) Motion Sensors: Motion sensors measure the change in motion as well as the orientation of the devices. There are two types of motions one can observe in a device: *linear* and *angular* motions. The linear motion refers to the linear displacement of a smart device while the angular motion refers to the rotational displacement of the device.

2) Environmental Sensors: Sensors such as light, pressure, etc. are used to sense the change in environmental parameters in the device's peripheral. The primary purpose of using environmental sensors is to help the devices to take autonomous decisions according to the changes in a device's peripheral. For instance, environment sensors are used in many applications to improve user experience (e.g., home automation systems, smart locks, smart lights, etc.).

3) Position Sensors: Position sensors deal with the physical position and location of the device. The most common position sensors used in smart devices are magnetic sensors and Global Positioning System (GPS). Magnetic sensors are usually used as a digital compass and help fix the orientation of a device's display. On the other hand, GPS is used for navigation purposes.

B. Communication Layer

The communication layer acts as a channel to transfer data collected in the sensing layer to other connected devices. In addition, the communication layer also establishes a connection between the device and cloud server to accumulate data from the external independent sensors [31], [69]. In smart devices, the communication layer is realized by using diverse communication technologies (e.g., Wi-Fi, Bluetooth, Zigbee, Z-Wave, LoRa, cellular network, etc.) to allow data flow between other devices within the same network. The Communication layer also simplifies remote access to smart devices. For example, a user can control a smart light from different locations using an app on a smartphone. For on-board sensors, data communication from the sensor to the

TABLE II
SENSORS AVAILABLE IN SMART DEVICES

Sensor Type	Sensor Name	Description
Motion Sensors	Accelerometer	• An electro-mechanical device which can measure changes in acceleration forces along x, y, and z-axis. • Detects various types of motion like shake, tilt, etc. and adjusts the display of the device accordingly.
	Linear Acceleration Sensor	• An accelerometer which can detect acceleration along one axis without considering the effect of gravitational force. • Helps to adjust the display with motion.
	Gyroscope	• Measures the rate of change of angular momentum in all three axes. • Detects rotational movement of the device and adjusts display accordingly.
Environmental Sensors	Light Sensor	• A photodiode which changes characteristics with the change of light intensity. • adjusts brightness and contrast of the display of the device. • Controls automatic lighting system.
	Proximity Sensor	• IR-based sensor to detect the presence of nearby objects without any physical contact. • Reduces power consumption of the display by disabling the LCD backlight and avoids inadvertent touches.
	Temperature Sensor	• Measures temperature of the device as well as ambient temperature. • Controls and sets the temperature in a device.
	Audio Sensor	• Two types of audio sensor: microphone and speaker. • Microphone: Detects acoustic signal. • Speaker: Playbacks audio signal.
	Camera	• Deals with light intensity, device ambiance, etc. to capture pictures and videos of surroundings. • Provides live video feeds.
	Barometer	• Measures the pressure of the device peripheral.
	Heart rate	• Measures the heart rate of the user in beat per second.
	Fingerprint	• Optical or capacitive scanner to capture the fingerprint of the user. • Provides biometric authentication.
Position Sensors	GPS	• Captures signal from the satellite to infer the location of the device. • Helps in navigation systems.
	Magnetic Sensor	• Measures device's magnetic field with respect to earth's magnetic field. • It is also used to fix display position by considering the magnetic field.

processing unit is performed by different serial and parallel communication protocols such as Serial Port Interface (SPI), Inter-Integrated Circuit (I2C) protocol, Peripheral Component Interconnect (PCI), etc.

C. Data Processing Layer

The data processing layer takes data collected in the sensing layer and analyses it to make data-driven decisions. This layer provides processed data to installed applications to perform different tasks. Also, in some smart devices (e.g., smartwatch, smart home hub, etc.), the data processing layer saves the results from previous analysis to improve the user experience. For instance, the data processing layer can learn the contexts and patterns during the user interactions to take autonomous decisions. This layer may share the result with other connected devices via the network layer.

D. Application Layer

The application layer presents and renders the results of the data processing to the user. In other words, the application layer is a user-centric layer which executes various tasks for the users. There exist diverse applications, which include smart transportation, smart home, personal care, healthcare, etc., [70]. Application layer also provides user interface to the users where users can select, control, and monitor different applications of the smart devices.

IV. EXISTING SENSOR MANAGEMENT SYSTEMS AND SECURITY NEEDS IN SMART DEVICES

Modern smart devices create a many-to-many relationship between apps and sensors that OSes manage. Managing this relation is a hard task and smart device OSes need effective and practical sensor management schemes to ensure secure data flow from the sensors to the apps. In addition, the sensor management in several smart devices (e.g., smart light, thermostat, etc.) also needs to assure a secure and seamless connection with external sensors to perform multiple tasks. Hence, an effective sensor management system is required to manage and ensure the security of all the sensors in the smart devices. In this section, we discuss different security requirements and goals of smart devices and how the existing sensor management systems address these requirements. Furthermore, we also articulate the shortcomings of existing sensor management systems.

To understand the security needs in smart devices, we consider the following smart device use cases. Assume a user, Bob, has several smart devices and sensors installed in his smart home system including smart lock, thermostat, motion sensor, temperature sensor, and presence sensor. Here, temperature and presence sensors are embedded in smart thermostat while the motion sensors are external sensors connected with smart devices using different communication protocols (e.g., ZigBee, Z-Wave, BLE, etc.) or via cloud [64]. We assume all the smart devices and sensors are in the same network. Bob installed several smart apps to automate and control tasks in smart devices. For instance, Bob installed an app in the thermostat to automate temperature control using the embedded temperature sensor. Also, Bob configured the external motion sensor with the smart lock to unlock the door with the users' motion. Based on the configurations and installed apps, the following scenarios can happen-

Case 1: An attacker having access to the same network installs a malicious motion sensor without alerting Bob. How can Bob identify the legitimate sensor while configuring the smart lock with the external motion sensor?

Case 2: Bob unknowingly installs a malicious app for the smart thermostat that is trying to access all the embedded sensors (both temperature and presence sensor). How can Bob limit the sensor access of the installed app?

Case 3: An attacker with the access of device peripheral captures the network packets between external sensors and the smart lock using a sniffing device. Additionally, the attacker is trying to change environment parameters (e.g., temperature) to change sensor reading and switch on the thermostat maliciously. How can Bob ensure that the attacker fails to extract any sensitive information from captured sensor-device communication and verify whether the sensor reading is legitimate or not?

Case 4: An attacker having access to the network sends malicious connection requests to the external motion sensor to make it unavailable for performing pre-defined tasks. How can Bob confirm sensor availability while configuring the smart lock with an external motion sensor?

To address these questions, current smart device ecosystem needs, (1) a sensor authentication system to identify fake or compromised sensors, (2) a sensor authorization framework to limit malicious sensor access, (3) Secure data sharing to confirm data confidentiality and integrity in sensors, (4) seamless connectivity to ensure sensor availability. In the following sub-sections, we briefly discuss existing sensor management systems and their shortcoming in addressing the aforementioned security needs.

A. Sensor Authentication

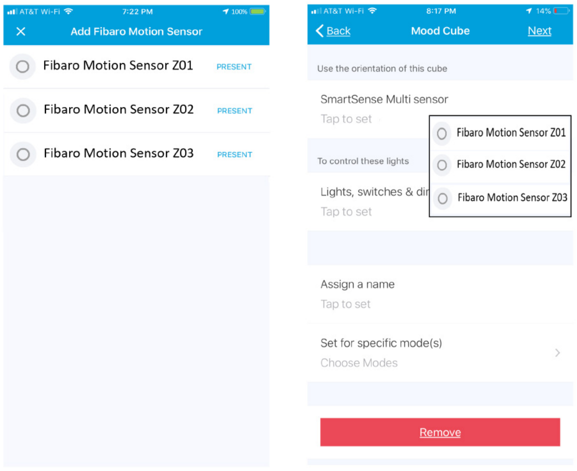
Smart devices can connect with each other to perform tasks collectively. Although this may increase the functionalities of smart devices, device authentication is needed to ensure secure communication among devices. The network layer of smart devices should have an authentication framework to connect with trusted devices and sensors. Similarly, the sensing layer of smart devices should also have an authentication framework to detect tampered sensors in the device ecosystem.

Although sensor authentication has not been a big concern for on-device sensors, an unauthenticated external sensor can perform malicious activities in connected smart devices [71]. To authenticate an external sensor in a smart device ecosystem, device fingerprinting can be utilized at the time of pairing between a smart device and an external sensor. Here, we discuss how sensor fingerprinting is implemented in the Samsung SmartThings platform. Samsung SmartThings offers a capability-based sensor management system which can control sensors of several devices from one common platform (a hub or smartphone). When a new external sensor is installed in the system, a pre-defined device handler is used to pair the sensor which specifies the capabilities of the sensor. This device handler also contains the fingerprint of the sensor. A sample device handler snippet is given in Listing 1. Here, a Fibaro motion sensor device handler for the Samsung SmartThings platform is shown. From line 4 to 11 capabilities of the sensor are defined and after initial installation, the sensor can provide these pre-defined functions. From line 13 to line 17, different benign commands are defined which allowed for Fibaro motion sensor in Samsung SmartThings ecosystem. In lines 19 and 20, the fingerprint of the sensor is defined which allows the smart device ecosystem to understand the device type and authenticate the sensor at the time of installation. This fingerprint is hard-coded in the device handler and can be manually modified to create new handler.

```
 1 metadata {
 2 definition (name: "Fibaro Motion Sensor", namespace: "
        smartthings", author: "SmartThings", ocfDeviceType: "x.com.
        st.d.sensor.motion", runLocally: true, minHubCoreVersion: '
        000.021.00001', executeCommandsLocally: true)
 3 {
 4          capability  "Motion Sensor"
 5          capability  "Temperature Measurement"
 6          capability  "Acceleration Sensor"
 7          capability  "Configuration"
 8          capability  "Illuminance Measurement"
 9          capability  "Sensor"
10          capability  "Battery"
11          capability "Health Check"
12
13          command "resetParams2StDefaults"
14          command "listCurrentParams"
15          command "updateZwaveParam"
16          command "test"
17          command "configure"
18
19          fingerprint mfr:"010F", prod:"0800", model:"2001"
20          fingerprint mfr:"010F", prod:"0800", model:"1001"
21          }
```

Listing 1. An example device handler of Fibaro Motion Sensor.

Although capability-based sensor management provides automatic authentication of the connected external sensor at the installation time, the hard-coded capabilities and fingerprint in the device handler can be easily altered. The device handler can be changed manually and an attacker can easily create a fake device handler to trick smart device user to install a compromised sensor in the smart device ecosystem. Attackers can also exploit the sensors by mimicking the hard-coded fingerprint in a compromised or fake sensor [72]. Furthermore, after initial authentication, all the sensor from the same vendor is visible to any connected users to add automation rules [73]. Hence, an adversary with access to the smart environment can use any installed sensors to add malicious automation rules. Figure 2 shows the app installation process in the Samsung SmartThings platform. Here, three different Fibaro motion sensors are available to the users and users can choose any of these sensors to create new automation rules. The current ecosystem does not allow any security measure to restrict specific sensors after initial authentication. Again,



(a) Adding sensor in the system (b) Adding sensor in an app

Fig. 2. Sensor authentication and automation in Samsung SmartThings.

if any of these three sensors is compromised, it can be used as a platform to attack connected devices sharing the smart environment.

B. Sensor Authorization

Modern smart devices use different apps to perform multiple tasks. These apps use multiple sensors to execute a task efficiently. At run-time, installed apps can ask for sensor access and it is necessary to check whether the requested access is legitimate as apps can use sensors for malicious purposes. For example, a simple flashlight app in the smartphone can access the motion sensor data which is irrelevant to the function of the app and can leak the information surreptitiously [40]. Smart devices should have a robust authorization framework to limit these unauthorized sensor accesses. Sensor authorization can be implemented in both the sensing and application layers. The sensing layer authorization can bind sensors with the apps while the application layer authorization can offer user control over sensors [22], [74].

Current smart device OSes offer a permission-based sensor management system to control on-device sensor authorization at app installation time and run-time [75], [76]. Here, we briefly discuss the Android sensor management system as Android OS holds the highest market share in the smart device domain (approximately 37%) [77]. Whenever an application wants to access a sensor in the OS, it has to communicate via a sensor manager framework (Figure 3). An application first sends a request to the sensor manager to register the desired sensor which invokes *ListenerService* service for the application. After receiving the request, the sensor manager creates a *ListenerService* for the application and maps the request with the designed sensor driver to acquire sensor data. If more than one App requests access for the same sensor, the sensor management system runs a multiplexing process to register one sensor to multiple Apps. This data acquisition path from the application to the sensor driver is initiated by the Hardware Abstraction Layer (HAL) which binds the sensor

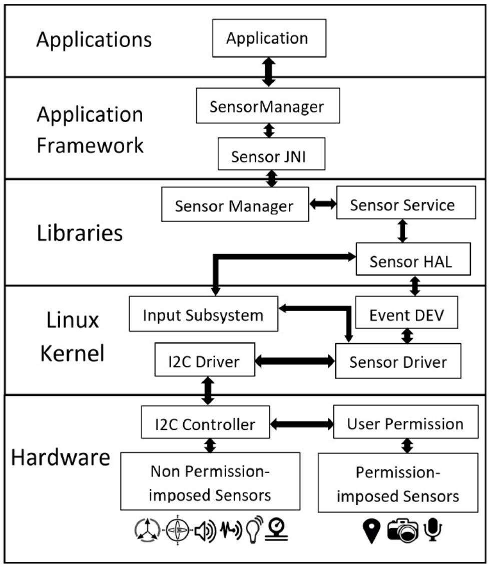


Fig. 3. Example Sensor Management System for Android.

hardware with the device driver. The sensor driver then activates the requested sensor and creates a data flow path from the sensor to the app [78]. On the other hand, Windows and Blackberry OSes use Sensor Class Extension to connect sensor hardware with the device driver [79], [80]. Windows OS also uses the User Mode Driver Framework to detect sensor access request and create a data acquisition path between sensor API and the APP. In iOS, the sensor management system is divided into four core services: Core Motion, Core Audio, Core Location, and Core Video [81]. The Core Motion service provides access to the motion sensors and some of the environmental sensors (e.g., barometer, light, proximity, etc.). The audio sensors (microphone and speakers), GPS, and the camera can be accessed via the Core Audio, the Core Location, and the Core Video services, respectively. These services provide data flow between the sensors and their apps according to the requests.

However, the main shortcoming of the permission-based sensor authorization is the dependence on the user's consent for sensor access. In most smart devices, permission-based sensor authorization is implemented for a subset of the supported sensors (e.g., GPS, camera, audio sensor). Whenever an application is installed in a smart device, it asks the users to grant permission to access various sensors. Thus, malicious applications may trick the user into allowing access to sensitive sensors to launch sensor-based attacks [7], [13], [23]. Users are typically unaware of what the malicious applications actually do with the sensed data [19], [21]. Furthermore, permissions are imposed on selected on-device sensors only (e.g., camera, microphone, and GPS) and other sensors are automatically included without any explicit permission. Thus, applications can easily access other no-permission-imposed sensors such as accelerometer, gyroscope, light sensor, etc., as discussed in the following sections in further detail. These sensors can be exploited maliciously and various sensor-based threats (e.g., information leakage, denial-of-service, etc.) can be launched on smart devices [82]–[84]. Additionally, for external sensors, the smart device ecosystem offers one-time sensor authorization at the time of sensor installation. After the initial installation, any connected smart device in the same network can access the external sensor without any additional authorization step.

C. Data Confidentiality and Integrity

One major concern is to keep the collected sensor data secure in smart devices. Smart devices use multiple sensors to perform a task and recent studies have shown that user activities on a smart device can be inferred using the sensor data [40]. The current smart device ecosystem implements different encryption methods in the network layer to encrypt sensor data before sharing with the devices. For example, Azure IoT suite, Amazon AWS, and Weave use SSL/TLS protocol to ensure secure communication [36]. Moreover, smart devices using ZigBee protocol use 128-bit AES encryption for secure communication [85]. However, most of the existing encryption schemes are available for communication between external sensors and smart devices or cloud communication. Some smart device platforms (e.g., Apple HomeKit, Weave) allows disk encryption to secure saved sensor data. But any app running in the smart devices can access these encrypted data, even collect unencrypted data from the on-device sensors [44]. These sensor data can be further processed to gain sensitive user information such as PIN code for the devices, typed information, even on-going tasks on a device [7].

D. Sensor Availability

To perform sensor-dependent tasks, smart devices should have uninterrupted sensor access. This requires sensor availability to the application layer of the devices from the sensing layer. Sensor availability is more important in external sensors than on-device sensors as attackers can target the network layer to perform a Denial-of-Service attack. The current smart device ecosystem offers firewall rules to filter unauthorized and malicious service requests to avoid unauthorized sensor access and avoid buffer overflow [86]. One possible solution can be fine-grained access control systems in the application layer to ensure continuous data availability to legit *app requests*. However, the existing schemes cannot detect sensor unavailability caused by forced changes in the sensors (e.g., hacking gyroscope using acoustic signals [8]).

E. Summary of Existing Sensor Management Systems and Their Shortcomings

Although existing sensor management systems in smart devices acknowledge the needs of securing sensors by addressing sensor authentication, authorization, and availability, there are several shortcomings that can be easily exploited by sensor-based threats.

(1) *User Dependency:* Existing sensor authorizations depend on user permission where users are asked to allow or deny sensor access permission to an app at installation time or run-time. However, no information about the nature of sensor usage is presented to the users. Hence, an app can easily trick the users to get desired sensor authorization and abuse sensors for malicious purposes [13], [87].

(2) *Selective Sensor Authorization:* Existing sensor management systems impose permission-based sensor authorization for selective sensors such as microphone, camera, and GPS. However, any installed app can access other sensors such as motion, light, magnetic, and proximity sensors without any explicit user permission. Attackers can exploit this limitation to get access to sensors and perform malicious activities including keystroke inference [88], eavesdropping [89], etc.

(3) *Passive Sensor Sniffing:* As smart devices allow external sensor integration to perform various tasks, it is possible to capture the network traffic between sensors and smart devices without interrupting normal operation. Also, both embedded and external sensors in a smart device are sensitive to environmental parameters which can be captured by a nearby smart device. For instance, typing in a keyboard creates a tap noise which can be captured by the microphone of a nearby smartphone [14], [90]. Attackers can extract sensitive information from captured traffic and sensor data even if proper encryption schemes are used to protect confidentiality [28], [85]. Hence, current sensor management systems cannot protect sensor abuse from passive sniffing.

(4) *Transitive Access:* Smart devices create a network of devices or smart environment where several devices are connected with each other to perform multiple tasks. Here, a newly installed smart device becomes automatically visible and can access other devices and sensors without any explicit privilege. As current sensor management systems use hard-coded capabilities and fingerprint to authenticate devices and sensors, attackers can introduce a compromised or fake device to capture sensitive sensor information and inject false data in the system to perform malicious activities [71], [91].

(5) *Indirect Sensor Data Injection:* Current sensor management systems do not offer any verification method to check whether a sensor input is valid or not. As a result, an attacker can target to maliciously change or control environment parameters such as light intensity or magnetic field to spoof sensor data, trigger malicious activities, or interrupt normal device activities. For instance, an attacker can use inaudible acoustic signals to trigger a voice command in voice-assisted devices and interrupt drone operations [8], [92].

V. Threat Model

In this section, we first categorize sensor-based threats in smart devices based on different security requirements and present the threat model.

A. Types of Sensor-Based Threats

A sensor-based threat exploits on-device or external sensors in a smart device ecosystem to perform attacks such as false data injection, eavesdropping, information leakage, etc. to jeopardize the proper operation of the device. Based on the nature of the threats, sensor-based threats can be categorized in two categories.

- *Passive Threats:* Passive sensor-based threats refer to the malicious sensor activities in smart devices without obstructing the normal operation of the device. For example, a malicious app installed in a smart device can run in the background and observe the sensor behavior to infer the ongoing task in the device [93]. Passive sensor-based threats can accomplish its malicious intents by performing malicious activities within a smart device or by utilizing another near-by smart device.

- *Active Threats:* Active sensor-based threats obstruct the normal operation of the smart device to perform malicious activities. An active sensor-based threats can directly abuse an on-board or external connected sensor by spoofing the sensor reading [91] or obstructing sensor signals using external device [8].

B. Attacker's Capabilities

To perform sensor attacks, we consider adversaries have the following capabilities in terms of device access, security privilege, and processing capabilities.

- *Device Access:* An adversary may need device access to perform malicious sensor activities in a smart device. Based on the type of access needed for an adversary, we categorized three different access types - direct access, transitive access, and peripheral access. In direct access, an adversary can directly access the sensors in a smart device to perform malicious activities. For example, a malicious app installed in a smartphone can directly access on-board sensors and collect data to infer sensitive information [7]. For transitive access, an adversary uses access to a smart device or sensor to perform malicious sensor activities in a targeted smart device. For example, in a smart home environment, an adversary can get access and strobe a smart light to change the output of the light sensor and a targeted smart light [91]. In peripheral access, an adversary implanted in a device (affected device) can perform malicious sensor activities in any smart device in its peripheral. Here, the affected device and the targeted device share the same environment, but are not connected with each other. For example, an adversary can use the audio sensor of a smartphone to eavesdrop to another smartphone in close proximity to infer keystrokes.
- *Security Privilege:* An adversary needs different levels of security privileges to perform malicious sensor activities in a smart device. For instance, to perform eavesdropping, an adversary needs minimum (low) privileges in the targeted device while for false data injection in a sensor, an adversary needs maximum privileges to access the sensor. In this work, we consider an adversary can have both privileges to classify the sensor-based threats and attacks correctly.
- *Processing Capability:* In smart devices, sensors mostly act as a triggering component to initiate automated applications. The sensed information in the smart device sensors often needs further processing to extract important information. Hence, an adversary needs processing capabilities to perform malicious sensor activities in smart devices. Based on the adversary's goal, the processing capacity may vary. For example, an adversary extracting keystrokes from motion sensors needs higher processing capabilities than an adversary recording phone conversation secretly off the device [40].

C. Threat Model

In this paper, we consider sensor-based threats and attacks in four working layers (sensing, communication, data processing, and application) of the smart devices. We consider adversaries that try to abuse the sensors to perform malicious tasks as a sensor-based threat. Additionally, this work considers passive threats to the sensors that do not disrupt normal functionalities of the smart devices. An adversary can be installed in a smart device to get access to the embedded sensors of the device or external sensors connected to the smart device. An adversary that has access to the peripheral of a targeted smart device to sniff the sensor data and network traffic is also within the scope of this work. Furthermore, we consider an adversary that can have direct or indirect access to the sensors of the smart devices to capture sensor data for further analysis. Specifically, we consider the following threats in our threat model.

- *Information Leakage:* An active or passive adversary may try to access the sensor data to steal sensitive information such as typing information, unlock code, PIN code, etc.
- *Transmitting Malicious Sensor Command:* An adversary may try to abuse sensors to transmit malicious sensor command to trigger malicious activities in a smart device.
- *False Sensor Data Injection:* An adversary may try to inject false sensor data to disrupt the normal functionalities of the smart devices.
- *Denial-of-Service:* An adversary may establish a sensory channel between on-device sensors and external entities (e.g., device, signal generator, etc.) to impede normal sensor operation which eventually leads to obstructing an on-going task in the smart device.

We consider these threats based on the impact of the attacks on smart devices. We expand the threat model in several sub-categories (e.g., information leakage includes keystroke inference, task inference, location inference, and eavesdropping) based on the final impact of sensor-based threats in Section VI. Each category includes other sub-threats based on how they are executed and targeted sensors in smart devices. Indeed, in total, we cover 89 different sensor-based threats reported by developers and researchers. Note that a physical sensor abuse or sensor tampering that could lead to physical damage to the smart devices is not considered and outside the scope of this work. Also, we do not consider threats arising from wireless sensor networks (WSNs) and cyber-physical systems (e.g., smart grid, industrial control systems, robotics systems, etc.) in the threat model. However, we do acknowledge that there are several interesting threats reported by the research community related to sensor exploitation in WSNs and CPSs. For instance, sensor impersonation attack in a wireless sensor network is an emerging threat where attackers implant a compromised sensor node to impersonate as valid components and perform malicious tasks such as false data injection and eavesdropping [94]–[96]. The readers are advised to check these useful studies for more information.

VI. Taxonomy of Threats, Attack Methods, and Their Impact

As existing sensor management systems and security schemes cannot provide adequate security to the sensors, attackers can exploit these sensors in various ways. In this section, we provide a detailed discussion about sensor-based threats and attacks to smart devices and survey the existing malicious attacks confirmed by the research community and developers [13], [19]–[23].

To understand the severity of sensor-based threats and attacks, we considered several common vulnerability scoring metrics for sensor-based threats in our discussion [97]. These scoring metrics give insights of the characteristics and impact of the threats. Detailed of these metrics are given below.

- *Attack Method (AM):* Attack method reflects how the threats penetrate the smart device to perform malicious sensor events. For sensor-based threats, we consider three methods to assess the severity of the threat- active, passive, and combination.
- *Device Access (DA):* To initiate a malicious sensor activity in a smart device, sensor-based threats need to access the device directly or indirectly. Based on the nature of the threat, we categorize the device access of sensor-based threats in three categories - *direct access*, *transitive access*, and *peripheral access*. Direct access refers to the threats that need access to the targeted device. In transitive access, a sensor-based threat can preform malicious sensor activities by accessing a device that is connected with the targeted device. For example, a sensor-based threat can perform malicious activities in a smart light by accessing a connected light sensor [28]. A sensor-based threat can also execute malicious sensor-activities by accessing the peripheral of the targeted device. For instance, keystrokes in a smartphone can be captured by a nearby smart speaker or smart watch [98].
- *Attack Complexity (AC):* Sensor-based threats and attacks can target one single sensor or multiple sensors to perform malicious tasks in smart devices. As abusing more than one sensor at a time may require immense effort from the attacker side, we consider two different levels (high and low) of complexity for sensor-based threats.
- *Required Privilege (RP):* To get access to the sensors for initiating malicious activities, sensor-based threats need to exploit existing security mechanisms of smart devices. As we explained in Section IV, sensors in a smart device can be categorized in two categories based on access permission: no-permission imposed sensor and permission imposed sensor. To access the no-permission based sensors, an adversary needs no excessive privilege while an adversary targeting permission imposed sensors needs high privilege. Hence, based on permission needed for accessing sensors in a smart device we consider two categories - high privilege threats (need excessive permission) and low privilege threats (need no permission).
- *User Interaction (UI):* This scoring metric portrays the need of user interaction other than the attacker to compromise the sensor functionalities in smart devices. Low user interaction indicates the higher impact of the sensor-based threats to smart devices.
- *Attack Impact (AI):* This scoring metric represents the impact of the sensor-based threats to various security requirements of the smart device. For sensor-based threats, we choose three important security features that might get affected - confidentiality, integrity, and availability.
- *Success Rate (SR):* Success rate of the sensor-based attacks is the fraction or percentage of success of an attack to perform malicious activities in a smart device among a number of attempts. We categorize this metric in three categories - high (success rate >90%), medium (success rate 70-90%), and low (success rate $<$ 70%).

In the following sub-sections, we summarize existing sensor-based threats and attacks in four broad categories based on the purpose and nature of the threats (presented in Section V).

A. Information Leakage

Information leakage is the most common sensor-based threat for smart devices and their applications. Sensors on smart devices can reveal sensitive data like passwords, secret keys of a cryptographic system, credit card information, etc. This information can be used directly to violate user privacy or to build a database for future attacks. An adversary (e.g., malicious app) can get access to the sensor data by exploiting vulnerabilities of existing sensor managements systems such as selective sensor authorization and user dependency (Section IV: use case 2). Only one sensor can be enough for information leakage (e.g., eavesdropping using microphone [13]) or multiple sensors can be exploited to create a more complex attack (e.g., keystroke inference using the gyroscope and audio sensors [99]). Moreover, sensors of one smart device can be used to leak information from a nearby device (passive information leakage) (Section IV: use case 3). In general, information leakage can be accomplished for the purpose of *(1) keystroke inference, (2) task inference, (3) location inference,* or *(4) eavesdropping* as explained below.

1) Keystroke Inference: Keystroke inference is a generic threat to smart devices. Most of the smart devices provide input medium such as the touchscreen, touchpad, keyboard (external or built-in virtual or real). Whenever a user types or gives input to a device, the device tilts and turns which creates deviations in data recorded by sensors (e.g., accelerometer, gyroscope, microphone, light sensor, etc.). These deviations in sensor data can be used to infer keystrokes in a smart device. Keystroke inference can be performed on the device itself or on a nearby device using sensors of the smart device. Keystroke inference can be performed actively (using on-board sensors) or passively (using external sensors). Here, we summarize different keystroke inferences based on the targeted sensors in the smart devices.

Keystroke Inference With Light Sensors: Light sensors in smart devices are usually associated with the display unit. In general, the display unit of the smart devices is touch-sensitive and provides a user interface to take inputs. For a constant state and unchangeable ambiance, the readings of the light sensor are constant. Each time a user touches and uses the touchscreen to interact with the device, he/she tilts and changes the orientation of the device, which causes changes in the readings of the light sensor. Each input may have a dissimilar light intensity recorded by the sensor. These changes in the readings of the light sensor of a device can be utilized to infer keystrokes of that particular device. An attacker can derive the various light intensities recorded by the light sensor by trying several keystrokes in a device and then construct a database. When users put their PINs or type something in the touchpad, attackers can capture the data maliciously from the device and collate these data with the database to decode keystroke information. As an example, some researchers developed a method named *PIN Skimming* to use the data from an ambient light sensor and RGBW (red, green, blue and white) sensor to extract PIN input of the smartphone [100].

Keystroke Inference with Motion Sensors: The main purpose of using the embedded motion sensors (e.g., accelerometer, gyroscope, linear acceleration sensor) in smart devices is to detect changes in motion of the devices such as shake, tilt, etc.. Accelerometer and linear acceleration sensor measure acceleration force that is applied to a device while gyroscope measures the rate of rotation in the devices. In smart devices with user interface (e.g., smartphone, smart watch, tablet, etc.), the value given by the motion sensors depends on the orientation of the device and user interactions (striking force of the finger on the device display, resistance force of the hand, the location of the finger on the touchpad of the device, etc.). Thus, when a user gives inputs to a device, the motion sensors' data changes accordingly. Generally, smart devices use two types of user interface to take user input – on-screen user interface (e.g., touchpad) and external user interface (e.g., keyboard, keypad, etc.). For both user interfaces, input keys are in a fixed position and for a single keystroke, the motion sensors give a specific value [101]. As attackers do not need any user permission to access the motion sensors, it is easy to access the motion sensor data.

One common keystroke inference attack can be performed by exploiting accelerometer. As mentioned above, accelerometer gives a specific reading for each user input on a smart device, thus, attackers can build a database of pre-processed accelerometer readings with diverse input scenarios and make a matching vector of sensor data and keystrokes to extract users' input [88], [102]. The data extracted from these attacks vary from text inputs to PINs and numbers typed in the touchpad which is much more serious as attackers can acquire the PIN or credit card information [82], [103]. Owusu *et al.* developed an app named *ACCessory* which can identify the area of the touchscreen by analyzing accelerometer data of smart devices [83]. *ACCessory* can infer PIN input on smart devices based on the detected area from accelerometer data. Accelerometer data can also be used to infer keystroke from a nearby keyboard. Marquardt *et al.* presented an attack scenario where accelerometer data of a smart device can be used to guess input on a nearby keyboard [84]. Whenever a user types on the keyboard, a vibration occurs and the accelerometer of the smart devices can catch this vibration and keystrokes can be identified correctly by analyzing this data [104].

Another method of keystroke inference can be achieved by analyzing the gyroscope data of a smart device. Gyroscope measures the angles of rotation in all the three axes which vary based on the specific area of the touch on the screen. Many smart devices such as smartphones, tablets, etc. have a feature when users input something on the touchpad the device vibrates and gyroscope is also sensitive to this vibrational force. The orientation angle recorded in the gyroscope and the vibration caused by the input can be used to distinguish inputs given by the users. Moreover, the data of the gyroscope can be combined with the tap sound of each key recorded via the microphone which can increase the success rate of inferring keystrokes [99], [105], [106]. The combination of accelerometer and gyroscope data can also be used for keystroke inference which yields more accurate results [107]–[112]. Additionally, the use of pattern recognition and deep learning algorithms can improve the success rate of keystroke inference attacks to smart devices [113].

In most wearables (smart bands, smartwatches, etc.), the motion sensors are utilized for monitoring the movement of the devices. A smartwatch, which is one of the most common wearables, maintains constant connectivity with smartphones via Bluetooth. While wearing a smartwatch, if a user moves his/her hands from an initial position, the motion sensor calculates the deviation and provides the data regarding the change of the position of the smartwatch [114]. Typing in the touchpad of a smart device while wearing a smartwatch will change the data recorded by the motion sensors of the smartwatch depending on user gestures. For a specific user input interface such as *QWERTY* keyboard of smartphones which has a specific distance between keys, the motion sensors' data of the smartwatch can be used to infer the keystrokes [15], [90], [115], [116]. Modern wearable devices (e.g., Apple Watch 5, Samsung Gear VR, etc.) also provide a user interface where users can provide inputs to the devices. Researchers showed that it is possible to infer the user input in wearables by observing hand movements [117]. A recent work showed it is possible to infer the unlock code of a smart lock from the gyroscope data of a smart watch [118].

Keystroke Inference With Audio Sensors: High precision microphones used in smart devices can sense the acoustic signals emanating from keyboards (built-in or nearby) which can be used to infer the keystrokes on a smart device. Asonov and Agrawal proposed an experiment to record the sound of key tapping and infer the correct key from it [119]. In this experiment, the attacker is assumed to record the acoustic signal emanating from the device while the user types on the keyboard. Then, the attacker matches this signal with a training dataset recorded stealthily while the same user was typing in the training period.

Zhuang *et al.* showed that it would be possible to infer keystrokes by just analyzing the acoustic emanation without having a training data set [14]. In this attack scenario, a specific key is assigned to a pre-defined class according to the frequency of the acoustic signal it generates while being typed. The attacker then takes a ten-minute of recording of the acoustic signal of typing on a keyboard. This recorded signal is analyzed using machine learning and speech recognition feature named *Cepstrum* to match with the previously defined key classes and infer the input of a keyboard.

In another work, Halevi and Saxena introduced a new technique named *Time-Frequency Decoding* to improve the accuracy of keystroke inference from the acoustic signal [120]. In this technique, machine learning and the frequency-based calculations are combined to match the recorded acoustic signal data from a smart device with a training dataset and increase the success rate of the attack scenario. This technique also considers the typing style of users to minimize the error rate of keystroke inference.

Berger *et al.* divided a PC keyboard in regions based on tap sound generated by keys and modeled a dictionary attack [121]. This attack utilizes signal processing and cross-correlation functions to process acoustic signal emanations from a nearby keyboard. Kune and Kim proposed a timing attack on a number pad used in smartphone and ATMs using the audio feedback beeps generated while entering PIN [122]. Inter-keystroke timing and distance between the numbers on the keypad are the main two features which are used to infer the input PIN in this attack. By analyzing the audio feedback recorded using the microphone of a nearby smart device, these two features are extracted and using Hidden Markov Model, the input numbers and PINs are inferred. Lu *et al.* proposed *KeyListener*, a context-aware inference method to predict the keystroke in QWERTY keyboard of smartphones and tablets using embedded microphones [123]. *KeyListener* uses a binary search tree algorithm to predict the typed information and achieves over 90% success rate. Similar to *KeyListener*, Shumailov *et al.* presented an acoustic side-channel attack which uses the tap noises of a virtual keyboard to infer the typed information in a smartphone [124]. Kim *et al.* further improved this work by capturing tap noises using multiple embedded microphones and combining the patterns of the acoustic signals [125]. Here, researchers developed a tapstroke detection and localization algorithms which can infer the typed information with 85.4% accuracy. In a recent work, Zhou *et al.* presented *PatternListener+*, an inference attack to predict the unlock patterns on an Android device using acoustic signal [126]. *PatternListener+* uses the speaker of a smartphone to play an inaudible sound and capture the reflected signal from users' fingertips using the embedded microphone. The reflected signal contains information of the hand movement which is further analyzed with a tree structure to infer the pattern of the lock. Backes *et al.* showed that acoustic signal emanated from a dot matrix printer which was collected by a nearby microphone of a smart device can be analyzed to predict the text printed on a paper [127]. In the training phase of this attack, words from a list are being printed, the acoustic signal is recorded and the data is stored. The audio signal processing and speech recognition techniques are used to extract the features of the acoustic signal to create a correlation between the number of needles used in the printer and the intensity of the audio signal. In the real attack scenario, the audio signal is captured by a nearby audio sensor and matched with the previous dataset to infer the printed text.

Zhu *et al.* showed a context-free attack scenario using the keyboard's acoustic emanation recorded in a smartphone to infer keystrokes [128]. In this attack scenario, the acoustic signals emanated from the keyboards are recorded by two or more smartphones. For each pair of microphones of smartphones, the recorded acoustic signal strength will depend on the distance between the typed key and the smartphones. By calculating the time-difference of the arrival of the acoustic signal, the position of the key can be inferred.

In a similar attack, Chhetri *et al.* introduced a method to reconstruct the design source code sent to a 3-D printer [129]. In this attack scenario, the acoustic signal emanated the 3-D printer is being recorded by a recorder placed in close proximity of a 3-D printer and the recorded file is processed for extracting time and frequency domain features. These features are then cross-matched with a training dataset collected in a learning phase to infer the correct design. Song *et al.* improved this attack by adding magnetic sensor data to accurately reconstruct the physical prints and their G-code [130].

Keystroke Inference With Video Sensors: Modern smart devices come with powerful cameras which can both take still pictures and record high definition videos. By applying image processing techniques in captured images, keystroke inference can be done. Simon and Anderson developed a malware named *PIN skimmer* which uses the front camera of a smartphone and microphone to infer PIN input in a smartphone [87]. PIN skimmer records the tap sound on the touchpad of a smartphone and records video using the front camera of the phone. The movement recorded in the video is then analyzed to detect which part of the touchscreen is used. This information is then combined with the tap sound to infer the inputs correctly.

Another potential malware attack on the smart devices using the camera is *Juice Filming Attack* [131]. In this attack scenario, a malicious app uses the camera to take screenshots when any user-input is given in the touchpad and save the images on the storage unit (internal ROM or external memory card) of the device. Most of the smart devices use USB for heterogeneous applications (e.g., charging, data transfer, etc.) and when the compromised device is connected to the laptop or any other device with a storage unit, the app transfers the stored pictures to the storage device from which attackers can easily extract the information.

Shukla *et al.* showed a method to infer the PIN input by analyzing the hand position using the recorded video [132]. In this method, a background application gets access to the camera of the smartphone and records a video when a user starts typing in a touchpad. Then, analyzing the hand position and the position of the smartphone, an attacker can extract the inputs given in a touchpad. Another version of this attack is to record the typing scenario using an external camera. In this scenario, a camera of a smart device (e.g., smartphone, smart glass, smart surveillance system, etc.) is used to record the video of typing the PIN. In both cases, the input PIN can be inferred with high accuracy.

Aviv introduced another type of attack named *Smudge Attack* using an external camera to infer pattern lock of a smart device [133]. In this attack scenario, a smart device is placed in between two cameras of other smart devices (smartphone or smart glass) and high definition pictures are taken. Whenever the user gives the unlock pattern in the touchpad, some smudge marks are left on the screen, and captured by the cameras, which leak information about the unlock pattern to an attacker.

Raguram *et al.* developed a process named *iSpy* which can reconstruct the typed text by analyzing the reflection of the touchscreen in a reflective surface such as sunglass or smart glass [134]. The experimental setup of *iSpy* includes a high definition camera which can capture the video of the reflective surface while a user types in the touchpad of a phone. The reflection of the phone is being extracted from the video and consecutive frames are analyzed to extract stable pictures of the phone screen. Features (hand position, motion in the screen, etc.) are extracted from stable pictures extracted from the video and by using machine learning techniques, key press detection is done and typed text can be inferred successfully. In more recent work, Wang *et al.* proposed *GazeRevealer*, a novel side-channel attack to infer keystrokes in a smart device using the eye movement of the users [135]. *GazeRevealer* uses the front camera to capture video and analyzes to extract multiple features such as eye movement, head position, etc. These features are used to train a classifier which can predict the keystroke in real-time with high accuracy.

Keystroke Inference With Magnetic Sensors: Besides the aforementioned attack scenarios, electromagnetic emanations from the keyboard can be used to infer the input of a computer. As magnetic sensors of smart devices are sensitive to electromagnetic emanations, they can be used as the attack medium. Vuagnoux and Pasini showed that both wired and wireless keyboards emit electromagnetic signals when a user types and this signal can be further processed to infer keystroke [136]. In this method, electromagnetic radiation is measured by the magnetic sensor of a smart device when a key is pressed and using the *falling edge transition technique*, an attacker can infer the keystrokes.

Lessons Learned for Keystroke Inference: We summarize the aforementioned threats and attacks in Table III with common vulnerability metrics. We can see smart devices with user input module (touchscreen, keypad, numeric keypad) are mostly the targeted device for keystroke inference. These threats and attacks affect the confidentiality of the sensor data. Another interesting fact we observe is the majority of the threats and attacks targets motion sensor (22 out of 42 reported threats and attacks) which does not require any permission to access in current smart device security schemes. Thus, these threats and attacks can easily access sensor data and extract keystroke information easily. For the targeted layer, we can notice the keystroke inference in smart devices only targets sensing (34 out of 42) and application (8 out of 42) layer. We can also observe a trade-off between attack complexity and required privilege in sensor-based threats targeting sensors in smart devices. For example, keystroke inference from the motion sensor (e.g., accelerometer) does not require any privilege to perform while keystroke inference from the audio sensor needs permission to access the microphone. However, accessing the motion sensor needs active vulnerability which may disrupt the on-going task in the smart device. On the other hand, capturing keystroke using the audio sensor can be both active and passive which increases the severity of the threat or attack. The outcomes of keystroke inference also have diverse effects on smart devices and users. As keystroke inference is directly related to user activities in smart devices, it impacts sensitive user information. Attackers can infer various typed information including device unlock code, password, banking information, typed and printed information, etc. These inferred information can be used to initiate another attack or directly used for malicious purposes such as ransom, data hijacking, identity theft, etc. In summary, passive keystroke inference with minimum required privilege (e.g., [15], [116], [136]) can severely affect the confidentiality of the smart devices.

2) Task Inference: Task inference refers to a type of attack which reveals the information of an on-going task or an application in a smart device. Task inference reveals information about the state of the device and attackers can replicate this device state to launch an attack without alerting security policies implemented in the device. Sensors associated with smart devices show deviation in the reading for various tasks running on the devices. This deviation in the reading can be used to infer the running process inside a device and application of the device.

Task Inference With Light Sensor: Light sensor of a smart device can be used to infer an on-going task on a device. Smart devices with display emit lights with distinct intensity for different tasks. For example, playing separate videos in a smart TV will change the emitted light intensity based on the background and video quality. This change in light intensity can be used to infer an on-going task on the display. Chakraborty *et al.* showed that light intensity changed in a flat panel display (e.g., smart TV, smart monitor, etc.) can be used to infer what is written on the screen by a light sensor of a smartphone [137]. In this attack, an Android-powered smartphone is placed in front of the display to capture the light intensity emitted from the screen. These captured light signals can be sampled and deconvoluted to infer the task on the monitor such as on-going videos, specific Web pages, etc. Celik *et al.* used a smart light to passively leak the status of a smart home [72]. In this attack, if no user is present inside the home, a smart light will maliciously trigger an on-off pattern to notify the user. Maiti and Jadliwala proposed a new attack vector to infer the audio and video of a smart TV using the light emitted from a smart light [93]. Here, researchers used the multimedia-visualization feature of smart light which creates a vibrant lighting effect in conjunction with audio and video playing nearby. Based on the light intensity emitted in audio frequencies, researchers successfully inferred an on-going audio or video.

Task Inference With Magnetic Sensors: Magnetic sensors in smart devices has the role to fix the orientation of the device with respect to Earth's magnetic field. Data recorded by a magnetic sensor change in the presence of an external magnetic field in the device's peripheral. This deviation in data

TABLE III
SUMMARY OF KEYSTROKE INFERENCE VIA SENSORS IN SMART DEVICES

Attack name	Target device	Target sensor	Target layer	Vulnerability metrics†						Ref.
				AM	DA	AC	RP	UI	SR	
Pin Skimming	Smartphone	Light	Sensing	●	●	○	○	○	◑	[100]
Text Inference	Smartphone	Accelerometer, Gyroscope	Application	●	●	●	○	○	○	[110]
Motion-based keystroke inference	Smartphone	Accelerometer, Gyroscope	Sensing	●	●	●	○	○	○	[101]
Keystroke inference on Android	Smartphone	Accelerometer, Gyroscope	Sensing	●	●	●	○	○	●	[88]
Input extraction via motion sensor	Smartphone	Accelerometer, Magnetometer	Sensing	●	●	●	○	○	◑	[82]
Accelerometer side channel attack	Smartphone	Accelerometer	Sensing	●	●	○	○	○	◑	[103]
ACCessory	Smartphone	Accelerometer	Sensing	●	●	○	○	●	○	[83]
(sp)iphone	Smartphone	Accelerometer	Sensing	○	○	○	○	●	◑	[84]
Single-stroke language-agnostic keylogging	Smartphone	Gyroscope, Microphone	Sensing	●	●	●	●	○	●	[99]
Touchlogger	Smartphone	Accelerometer, Gyroscope	Sensing	●	●	●	○	●	◑	[105]
Taplogger	Smartphone	Accelerometer, Gyroscope	Application	●	●	●	○	●	●	[107]
I Know What You Type	Smartphone	Accelerometer, Gyroscope, Light	Sensing	●	●	●	○	●	○	[113]
Type and leak	smartphone	Accelerometer	Sensing	○	○	○	○	●	◑	[117]
Risk Assessment of motion sensor	smartphone	Accelerometer	Sensing	○	◑	○	○	●	○	[102]
Infer tapped and traced user input	Smartphone	Accelerometer, Gyroscope	Application	●	●	●	○	○	◑	[109]
Motion-based side-channel attack	Smartphone	Accelerometer, Gyroscope	Sensing	●	●	●	○	○	◑	[106]
When good becomes evil	Smart watch	Accelerometer	Sensing	○	○	○	○	●	○	[115]
Mole	Smart watch	Accelerometer	Application	○	○	○	○	●	○	[90]
(Smart) watch your taps	Smart watch	Accelerometer	Sensing	○	◑	○	○	○	●	[15], [98]
Wristsnoop	Smart watch	Accelerometer	Sensing	○	◑	○	○	●	○	[116]
Inferring Mechanical Lock Combinations	Smart lock	Gyroscope	Application	○	○	○	○	○	●	[118]
Inference of private information	Smartphone	Accelerometer, gyroscope	Sensing	●	●	●	○	○	●	[111]
KeyListener	Smartphone	Microphone	Sensing	●	●	○	●	○	●	[123]
aLeak	Smart watch	Accelerometer, Gyroscope	Sensing	○	◑	●	○	○	●	[112]
Keyboard acoustic emanation	Smartphone	Microphone	Sensing	○	○	○	●	○	◑	[119]
Keyboard acoustic emanations revisited	Smartphone	Microphone	Sensing	○	○	○	●	○	●	[14]
A closer look at keyboard acoustic emanations	Smartphone	Microphone	Sensing	○	○	○	●	○	○	[120]
TapSnoop	Smartphone	Microphone	Sensing	●	○	○	●	○	◑	[125]
Dictionary attacks using keyboard acoustic	Smartphone	Microphone	Sensing	○	○	○	●	○	●	[121]
Timing attacks	Smartphone	Microphone	Sensing	●	○	○	●	○	-	[122]
Acoustic Side-Channel Attacks on Printers	Smartphone	Microphone	Sensing	○	○	○	●	○	◑	[127]
Context-free keyboard acoustic emanations	Smartphone	Microphone	Sensing	○	○	○	●	○	◑	[128]
PatternListener+	Smartphone	Microphone, speaker	Sensing	●	●	●	●	●	●	[126]
Hearing your touch	Smartphone	Microphone, speaker	Sensing	●	●	○	●	○	○	[124]
PIN skimmer	Smartphone	Microphone, Camera	Sensing	●	●	●	●	●	○	[87]
Juice filming attack	Smartphone	Camera	Application	●	○	○	●	○	-	[131]
Beware, your hands reveal your secrets!	Smartphone	Camera	Sensing	○	○	○	●	○	◑	[132]
Smudge attack	Smartphone	Camera	Sensing	○	○	○	●	○	○	[133]
iSpy	Smart security camera	Camera	Application	○	○	○	●	○	●	[134]
GazeRevealer	Smartphone	Camera	Application	●	●	○	●	●	◑	[135]
Compromising electromagnetic emanations	Smartphone	Magnetic	Sensing	○	○	○	○	○	●	[136]
My Smartphone Knows What You Print	Smart printer	Microphone, magnetic	Sensing	○	○	●	●	○	●	[130]

† Attack Method (AM): Active- ●, Passive- ○; Device Access (DA): Direct-●, Transitive-◑, Peripheral-○; Attack Complexity (AC): High-●, Low- ○; Required Privilege (RP): High privilege- ●, Low privilege- ○; User Interaction (UI): Needed - ●, not needed - ○; Success Rate (SR): High (>90%) - ●, medium (70-90%) - ◑, low (<70%) - ○.

‡ Any type of keystroke inference impacts the confidentiality of the smart device.

can be used to identify the tasks running on a device. Many smart devices have a storage unit and whenever data is written or read from this storage unit, a change in the reading of the magnetic sensor can be observed. Magnetic sensors of a smart device can be used not only to infer information of the device itself, but can also be used as a medium to fetch information from a nearby device. Biedermann *et al.* showed that the magnetic sensor of a smartphone could be used to infer on-going tasks in a storage unit like the hard drives of the computers and servers [138]. When an application is running on a computer, the hard drives generate a magnetic field which can be sensed by a magnetic sensor of a smartphone. Various actions cause distinct readings on the magnetic sensor which can be used to track the users' action. This can be considered as a serious threat to the device and attackers can fetch valuable information in this way. Ning *et al.* proposed *DeepMag+*, a side-channel attack to exploit on-board magnetic sensor for inferring smart apps installed in a smart device [139]. *DeepMag+* captures the on-board magnetic sensor data while executing installed apps in a smart device and uses convolutional neural network to fingerprint the apps. Additionally, *DeepMag+* can combine motion sensor data with magnetic sensor to increase the inference accuracy up to 98%. Similar to this work, Matyunin *et al.* presented *MagneticSpy*, a novel website and application fingerprinting method exploiting magnetic sensors of a smart device [140]. *MagneticSpy* analyzes the electromagnetic disturbances caused by the mobile processors which are proportional to the CPU workload. By analyzing the deviance in different working conditions, *MagneticSpy* can infer the on-going CPU activity with high accuracy (up to 90%).

An electromagnetic (EM) emanation is a common phenomenon for smart devices. Electromagnetic emanations occur whenever current passes through a device and a task is running on a device. EM emanation attacks can also be observed in FPGA-based (Field-programmable gate array)

TABLE IV
SUMMARY OF TASK INFERENCE, LOCATION INFERENCE, AND EAVESDROPPING VIA SENSORS IN SMART DEVICES

	Attack name	Target device	Target sensor	Target layer	Vulnerability metrics[†]						Ref.
					AM	DA	AC	RP	UI	SR	
Task Inference	LightSpy	Smartphone	Light	Application	○	○	○	○	○	◑	[137]
	IoTBench	Smart light	Light	Application	○	◑	○	●	●	-	[72]
	Hard drive side-channel attacks	Smartphone	Magnetic	Sensing	○	○	○	○	○	◑	[138]
	Electro-magnetic Analysis of smart cards	Smart card	Magnetic	Sensing	○	○	○	○	○	-	[141]
	Power analysis attack	Smart devices	Any embedded sensors	Sensing	●	○	○	○	○	-	[147]
	Light Ear	Smart light	Light	Application	○	○	○	○	○	-	[93]
	Peek-a-boo	Smart home devices	Motion, light, temperature	Communication	○	◑	●	○	○	●	[85]
	DeepMag+	smartphone	Magnetometer	Sensing	●	●	○	○	●	●	[139]
	MagAttack	smartphone	Magnetometer	Sensing	●	●	○	○	○	◑	[145]
	MagneticSpy	smartphone	Magnetometer	Sensing	●	●	○	○	●	●	[140]
Location inference	VoipLoc	Smartphone	Microphone, speaker	Sensing	●	●	●	●	○	◑	[149]
	ACComplice	Smartphone	Accelerometer	Application	●	●	○	○	●	-	[150]
	Inferring Your Secrets from Android	Smart Navigation device	Microphone, speaker	Application	●	●	●	●	○	◑	[151]
	Permission less Location Attack	Smartphone	Magnetic	Application	●	●	○	○	○	◑	[152]
	Inferring User Routes and Locations	Smartphone	Accelerometer, gyroscope, magnetic	Application	●	●	●	○	○	○	[153]
	MISSILE	Smartphone	Accelerometer, gyroscope	Sensing	●	●	●	○	●	◑	[154]
	Prying into Private Spaces	Smartphone	Accelerometer, gyroscope	Sensing	●	●	●	○	●	◑	[155]
Eavesdropping	Soundcomber	Smartphone	Microphone	Application	●	●	○	●	●	◑	[13]
	VoicEmployer	Smartphone	Microphone, Speaker	Application	●	●	●	●	●	●	[156]
	CPVT	Smartphone	Microphone, Speaker	Application	●	●	●	●	●	-	[157]
	Hidden voice commands	Smart car	Microphone	Application	○	○	○	○	●	●	[158]
	Gyrophone	Smartphone	Gyroscope	Sensing	●	●	○	○	○	◑	[89]
	Spearphone	smartphone	Accelerometer	Sensing	○	○	○	○	●	●	[159]
	I Can Hear Your Alexa	smart speaker	Microphone	Communication	○	○	○	○	●	○	[160]

[†] Attack Method (AM): Active- ●, Passive- ○; Device Access (DA): Direct-●, Transitive-◑, Peripheral-○; Attack Complexity (AC): High-●, Low- ○; Required Privilege (RP): High privilege- ●, Low privilege- ○; User Interaction (UI): Needed - ●, not needed - ○; Success Rate (SR): High (>90%) - ●, medium (70-90%) - ◑, low (<70%) - ○.

[‡] Any type of information leakage impacts the confidentiality of the smart device.

smart devices [141]–[143]. Attackers can record electromagnetic emission data generated from the FPGA-based smart devices to deduct which kind of application is running in the system and also the states of logic blocks of the devices. Such information leakages make the system vulnerable to the user. Smart cards also emit EM waves while performing various tasks which can be captured by a radio frequency (RF) antenna and the task can be inferred from the radiation [144]. Cheng *et al.* proposed *MagAttack*, a side-channel attack to abuse the magnetic sensor of smart mobile devices [145]. User activities such as application launching and operation has a slight but significant effect on CPU's power consumption, and hence in the EM emissions. An attacker can capture this EM emission using the magnetic sensors of a smart device and infer the on-going user activities in a laptop or workstation.

Task Inference With Power Analysis: Power analysis is a form of sensor-based threat where an attacker studies the power consumption and power traces of the sensors for extracting information from the devices [146]. O'Flynn and Chen introduced an attack scenario where the power analysis attack is launched against IEEE 802.15.4 nodes [147], which is a standard low power wireless protocol used in smart devices. Low power smart devices use this protocol standard for various communication purposes such as connecting to a network, communicating with other devices, etc.. In this attack scenario, an attacker uses differential power analysis in the sensors. As packets transmitted from the smart devices are encrypted, power analysis on the sensors can infer which encryption process is running in the device. Again, diverse encryption process leads to diverse power profiles which reveal associated information (e.g., key size, block size, etc.) about the encryption process. Encryption process also depends on the packet size which can be observed in the power profile and attackers can infer what type of information is being transmitted based on the packet size.

Task Inference From Sniffing Sensor Data: In a connected environment such as smart home, several smart devices are connected with each other and with multiple sensors. These sensors communicate with the devices using various protocols (e.g., WiFi, ZigBee, Z-Wave, etc.) and work as triggering devices for several automated tasks. An attacker can sniff the communication traffics in the smart environment and infer user and device actions which can be considered as a privacy violation [148]. Acar *et al.* showed that it is possible to infer user activities and devices states by capturing the communication packets and extracting sensor data in a smart home environment [85]. In this attack scenario, an adversary in close proximity of the smart environment can sniff the communication packets and infer the states of the devices (active/inactive). In addition, authors showed that the attacker could deduce the actions of the users (e.g., walking, presence, etc.) using machine learning techniques in captured traffics.

Lessons Learned for Task Inference: Similar to keystroke inference, task inference in smart devices also affect the confidentiality of the devices. From Table IV, we can observe the majority of the task inference threats (6 out of 10 reported threats and attacks) are passive which indicates the high impact on the smart devices. Another interesting fact is the majority of these threats does not need any additional privilege (9 out of 10) to bypass existing security schemes. Also, task inference threats target sensing (6 reported threats), application (3 reported threats), and communication (1 reported threat) which indicates a broad attack surface of these threats. One limitation of reported task inference attacks is the lack of extensive evaluation of the attacks. To understand the effectiveness of a sensor-based attack, it is necessary to check the success rate of the attack on real-life smart devices. The majority of the task inference attacks are not appropriately evaluated with known evaluation metrics such as success rate, error rate, precision, etc.. Without proper evaluation metrics, especially without reported success rate, it is hard to understand the effectiveness and feasibility of task inference attacks on a smart device. Task inference directly impacts the confidentiality and privacy of the smart device users by leaking sensitive information such as user activity, installed security measures, installed apps on smart devices, etc. Attackers can profile a user based on task inference attacks to perform diverse types of malicious activities such as gaining access to the smart device and environment, bypassing security measures to leak data, manipulate or obstruct on-going tasks, etc., [85].

3) Location Inference: Researchers developed a novel location-privacy attack based on acoustic side-channels [149]. The attack is based on acoustic information embedded within foreground-audio disseminated in a closed environment (i.e., conference room). The researchers studied how audio, generated by secure messaging clients in voice-call mode, can be abused to generate a location fingerprint. The attack leverages the pattern of acoustic reflections of the human voice at the user's location and does not depend on any characteristic background sounds. The attack can be used to compromise location privacy of participants of an anonymous VoIP session, or even to carry out confirmation attacks that verify if a pair of audio recordings originated from the same location regardless of the speakers. Other researchers have also shown that several heuristics can be used to identify sensitive locations (i.e., home and work locations) of a victim whose personal device is under an adversary control [20]. Han *et al.* showed that it is possible to infer the location of a user using the accelerometer of a smartphone [150]. Here, researchers first derived an approximate motion trajectory from accelerometer reading and correlated the trajectory with the map to infer the exact location of the user. Zhou *et al.* showed that it is possible to infer the location of the user by analyzing verbal directions provided by navigation apps of a smart device [151]. Researchers measured the on/off times of the speaker controlled by the navigation app to leak the driving instructions to the attacker. In a more recent work, Block and Noubir introduced a new location inference technique using the smartphone's magnetometer [152]. Here, researchers used small fluctuations originated by nearby magnetic fields while the smartphone is in motion to build a trajectory path of the user. Narain *et al.* proposed a combination of sensor data (accelerometer, gyroscope, and magnetometer) to further improve the accuracy of the inferred location [153]. In a recent work, Zheng and Hu proposed a location eavesdropping attack using the mobile inertia/motion sensors [154]. Here, researchers showed that in the presence of specific indoor structures (e.g., elevators, fire stop doors, etc.), motion sensors display specific patterns which can be utilized to infer the location correctly. Similar to this work, Fyke *et al.* used the motion sensors data to recreate user's movement and plot maps and landmarks in private spaces (e.g., home, workplace, etc.) [155].

Lessons Learned for Location Inference: Although location inference attacks impact the confidentiality of smart devices, all of the threats (7 reported threats and attacks) are active which limits the consequences (Table IV). Also, to execute malicious sensor activities, these threats need direct access to the devices which affect the easy deployability of these threats in real-life smart devices. One can also observe from Table IV that the success rate of these attacks is low to medium range. Compared to keystroke and task inference attacks, location inference poses less effects on the security of the smart devices. However, leaking location information can violate user's privacy and propagate other attacks including a targeted physical attack on the user's vehicle [161].

4) Eavesdropping: Many smart devices such as voice-enabled speakers use audio sensors for making calls, recording audio messages, receiving voice commands, etc. Eavesdropping refers to a type of attack where a malicious app records a conversation stealthily by exploiting audio sensors and extract information from the conversation. An attacker can save the recorded conversation on a device or listen to the conversation in real-time. One of the recent examples of eavesdropping via the microphone of a smartphone is *Soundcomber* [13]. In this example, a malicious app covertly records when a conversation is initiated from the device. As the recording is done in the background, a user does not have any idea about the recording. Several companies like banks, social security offices, credit card companies, etc. have automated voice messaging systems and users have to say their private information such as credit card numbers or social security numbers at the beginning of the call. Thus, *Soundcomber* does not have to record all the conversations to extract data. Only the beginning part of the conversations will be enough for extracting private information of the user. Moreover, a specific conversation can also be recorded by identifying the dialed number on a smartphone. The touchpad of the smartphone creates corresponding tones when any number is dialed. This tone can be recorded and processed to identify the dialed number. After that, when the desired number is dialed, the conversation can be recorded and then processed to extract information.

Another way to exploit microphones is to attack through voice assistant apps, e.g., Apple's Siri and Google Voice Search. Most of the smart devices nowadays have built-in voice search apps. Diao *et al.* developed a malware named *VoicEmployer* which can be installed on the device to record the voice command given in a smartphone [156]. This malware can use the recorded command for various malicious activities such as replicate malicious voice command, transfer information to paired devices, etc. *Cyber Physical Voice privacy Theft Trojan horse (CPVT)* is another malware which uses the microphone of smartphones to record conversations [157]. The recording of the conversation can be controlled by external control channels like SMS, Wi-Fi, or Sensory channels [18]. An attacker can trigger *CPVT* and create command about when to start recording and when to stop recording using SMS, Wi-FI, or even sensors. Recorded conversations are stored in the device and the attacker can gain the stored files using e-mail, SMS, or connecting via USB. Carlini *et al.* showed that it is possible to exploit voice assistant apps by inserting hidden voice commands [158]. In this attack, the attacker first records voice commands of the user and extracts features from the recorded audio clips. From the extracted features, a new command is generated which is not understandable by humans, but recognized by the voice assistant apps. In a recent work, Kennedy *et al.* showed that it is possible to infer the voice command given to a voice assistant device (e.g., Amazon Alexa) by capturing the network packet and using natural language processing [160].

The gyroscope on smart devices is also sensitive to an acoustic signal. The typical sampling rate of gyroscope covers some frequency of audible range which can be used to reconstruct the speech of a user. Michalevsky *et al.* proposed a new way of eavesdropping by analyzing vibrational noise in gyroscope caused by an acoustic signal [89]. As the gyroscope does not cover the full audible range, this new process can distinguish speakers and one-syllable words by using signal processing and machine learning techniques. In a recent work, Anand *et al.* showed that the on-board accelerometer could be used to eavesdrop and reconstruct the speech of a user [159]. While a user talks on a smartphone, the loudspeaker of a smartphone shows some reverberations which impact the accelerometer reading. This deviation in accelerometer can be further analyzed to extract sensitive information such as speaker identification and gender classification.

Lessons Learned for Eavesdropping: Eavesdropping mostly affects smart devices with audio sensors and impacts the confidentiality of the devices. From Table IV, it is visible that the majority of the eavesdropping are active attacks (4 out of 5 reported threats and attacks) and require additional privileges (4 out of 5 threats and attacks) to bypass the existing security schemes. These threats also need users to interact with the system to perform malicious tasks that limit the impact of these threats. For performing eavesdropping, the majority of the threats and attacks also need direct access (4 reported threats) on a targeted smart device. Because of these dependencies, the impact of eavesdropping is lower than other types of information leakage attacks. Nevertheless, the information captured in the eavesdropping attack can be used to perform various malicious activities such as leaking private conversation, gaining physical access to a secured environment, etc., [161].

B. Transmitting Malicious Sensor Commands

Sensors available in the smart devices can be used to transmit malicious sensor patterns or triggering commands to activate malware that may have been implanted in a victim's device [18]. Sensors may be employed to create unexpected communication channels between device peripherals. Such channels can be used to exchange critical sensor parameters (e.g., devices' motion, light intensity, magnetic field, etc.) or to transmit malicious commands (Section IV: use case 3).

Transmitting via Light Sensors: Light sensors can be used as a potential method of transmitting signals and malicious commands [177]. It is easier to transfer a bit stream via a light source by turning it on and off. Since the light sensor of a smart device can distinguish the intensity of the light source, the light intensity change can be decoded as a bit stream in the device. By controlling the voltage of a light source, an attacker can easily transfer trigger messages and can activate malware implanted in a device. Hasan *et al.* showed that TV screen or laptop monitor could also be used to transfer trigger messages to a compromised smart device by changing the light intensity of the monitor [11]. Fernandes *et al.* showed that a smart light could be maliciously programmed to strobe the light at a high rate and if the user has the health problem of seizure, this action will trigger the user's seizures which is really dangerous [33]. Celik *et al.* showed that a smart light could be programmed to operate in a specific pattern to trigger a smart camera and take pictures surreptitiously [72].

Transmitting via Magnetic Sensors: As mentioned earlier, magnetic sensors of a smart device are sensitive to the magnetic fields of the device's peripherals. By changing the magnetic field of the device ambiance, one can easily change the readings of the magnetic sensor which can be used as a triggering message of malware. Triggering messages encoded by an electromagnet can be sent to a smart device and there will be some deviations in the magnetic sensor's readings of the device due to this message. These deviations can be calculated and the triggering message can be extracted from this electro-magnetic signal. Moreover, the magnetic field deviations can be calculated in x, y, and z-axis and divergent values of the magnetic field deviations can be interpreted as disparate triggering messages [11].

Transmitting via Audio Sensors: Audio sensors can be used to transmit malicious commands to activate a malicious application in a smart device. Hasan *et al.* showed that a triggering message embedded in an audio song can be detected by the microphone and can trigger a malicious app in a smartphone [11]. Moreover, microphones used in modern smart devices can detect audio signals with a frequency lower than the audible range. Malware can be transferred using this audio channel as a covert channel to bypass the security measures of the device. Deshotels showed that the ultrasonic sound could be used to send information to smartphones without alerting the user or any security measurement implemented on the device [162]. Subramanian *et al.* showed that a trojan can be transferred by encoding it in an audio signal and transferring it using a buzzer [10]. In a recent work, Zhang *et al.* showed that it is possible to transmit an inaudible acoustic signal to smart speakers to trigger malicious activities [164]. Yan *et al.* performed a feasibility study of previous work and concluded that it is possible to trigger several malicious events in smart devices including making a phone call, changing state of connected devices, etc., [92]. Kumar *et al.* showed that valid voice commands could be used to trick smart speakers (e.g., Amazon Alexa, Amazon Echo, etc.) to perform malicious activities in skill squatting attack [163]. Here, researchers used misinterpretations of valid commands made by the smart speaker to trigger a malicious activity. For example, "test your luck" can be misinterpreted by the smart speaker as "test your lock" which can unlock the door. In a recent work, Zhang *et al.* proposed *Vaspy*, a malicious app installed in the smart device to exploit voice assistant devices [165]. *Vaspy* silently observes smartphone activities and captures the phone call conversations to extract the voice activation commands. Upon extracting the voice command, *Vaspy* uses a machine learning model to analyze user behavior and choose a specific time to launch an attack surreptitiously.

Lessons Learned for Transmitting Malicious Sensor Commands: The threat of transmitting malicious sensor

TABLE V
SUMMARY OF OTHER SENSOR-BASED ATTACKS IN SMART DEVICES

	Attack name	Target device	Target sensor	Target layer	Vulnerability metrics†							Ref.
					AI	AM	DA	AC	RP	UI	SR	
Transmitting malicious sensor commands	Out-of-band command	Smartphone	Light	Sensing	I	●	○	○	○	○	●	[11]
	Creating seizures using strobed light	Smart light	Light	Application	I	○	◑	○	●	●	●	[33]
	IoTBench- Side channel attack	Smart light	Light	Application	I	○	◑	○	●	●	●	[72]
	Out-of-band command via magnetic sensor	Smartphone	Magnetic	Sensing	I	●	○	○	○	○	●	[11]
	Out-of-band command via audio sensor	Smartphone	Microphone	Sensing	I	●	○	○	●	○	●	[11]
	Inaudible sound as a covert channel	Smartphone	Microphone	Sensing	I	●	○	○	●	○	●	[162]
	Sensor side channels	Smartphone	Microphone	Sensing	I	●	○	○	●	○	-	[10]
	Skill squatting attack	Smart Speaker	Microphone	Sensing	C, I	●	○	●	●	●	●	[163]
	DolphinAttack	Smart Speaker	Microphone, Speaker	Application	C, I	●	○	●	●	●	-	[164]
	Injecting inaudible voice commands	Smart voice assistant	Microphone, Speaker	Application	C, I	●	○	●	●	●	-	[92]
	Vaspy	smartphone	Microphone	Application	C, I	●	●	●	●	●	●	[165]
False sensor data injection	GPS spoofing attack	Smart navigation device	GPS	Application	I	●	●	○	●	○	-	[166]
	GPS jamming	Smart navigation device	GPS	Application	I	●	●	○	●	○	-	[167]
	Spy-sense	Smart sensor network	-	Data processing	I	●	◑	○	○	○	-	[168]
	Injected and Delivered	Smart cars, drone	Accelerometer, gyroscope	Sensing	I	○	○	○	○	○	◑	[169]
	This ain't your dose	Smart medical device	Light	Sensing	I	○	○	○	○	○	●	[170]
	Illusion and dazzle	Smart car	Light	Application	I	○	○	○	○	○	-	[171]
	Remote attacks on automated vehicles	Smart car	Light, Camera	Application	I	●	○	●	●	○	●	[172]
	REEVE	Smart voice assistant	Microphone	Application	I	○	○	○	○	●	-	[173]
	Using AI to Hack IA	Smart voice assistant	Microphone, Speaker	Application	I	●	●	●	●	●	-	[174]
	Light Commands	Smart voice assistant	Light	Application	I	○	○	○	○	○	-	[175]
Denial-of-Service	Rocking drones	Smart drone	Gyroscope	Communication	A	●	○	○	○	●	●	[8]
	Pairjam	Smart home device	Microphone	Communication	A	●	◑	○	●	○	◑	[176]

† Attack Impact (AI): Confidentiality (C), Integrity (I), Availability (A); Attack Method (AM): Active- ●, Passive- ○; Device Access (DA): Direct-●, Transitive-◑, Peripheral-○; Attack Complexity (AC): High-●, Low- ○; Required Privilege (RP): High privilege- ●, Low privilege- ○; User Interaction (UI): Needed - ●, not needed - ○; Success Rate (SR): High (>90%) - ●, medium (70-90%) - ◑, low (<70%) - ○.

commands mostly affects the integrity of smart devices (Table V). The majority of the threats and attacks (9 out of 11 reported) needs an additional privilege to bypass the existing security schemes. Also, upon successful attack, malicious sensor commands trigger malicious activities in the smart devices which obstruct normal operations. Thus, transmitting malicious sensor commands (all reported threats and attacks) act as active attacks to the smart devices. Another interesting fact we observe transmitting malicious sensor commands do not need any direct connection to the device. Only transitive (2 out of 11 threats) or peripheral access (9 out of 11) is enough to transmit malicious commands to the targeted smart devices. However, the success rate of most of the threats and attacks is high, indicating the high impact on smart devices. This tradeoff between excessive privilege and success rate determines the effects of the threats and attacks.

C. False Sensor Data Injection

The applications of smart devices largely depend on data collected by sensors available on the devices. By altering the sensor data, one can control the applications of smart devices. False sensor data injection refers to an attack where the sensor data used in the smart applications is forged or intentionally changed to perform malicious activities. The false sensor data can be injected in the devices by accessing the device physically or by using various communication mediums (Bluetooth, ZigBee, Z-Wave, Wi-Fi, cellular network, etc.) covertly. An attacker can also introduce fake sensors in the IoT environment to inject false generated data and initiate malicious activities (Section IV: use cases - 1 and 3) [72], [178]. Moreover, the sensors of smart devices can also be used to alter data typed or stored on the devices.

Tu *et al.* presented a *spoof attack*, where an out-of-band signal is inserted in smart devices via motion sensor [169]. This signal injection results in deviation in sensor output which disrupts the normal functionality of the smart devices. Park *et al.* used infrared light to disrupt normal operation of a smart medical device [170]. Here, researchers used a medical infusion pump to inject the spoof light signal and change the dose of the medicine in the device. In another recent work, Shin *et al.* exploited the light sensor of a smart car to change the output of the automatic obstruction detection system [171]. Petit *et al.* improved this attack by combining camera reading of a smart car to change the output of autonomous vehicle [172]. In a recent work, Zhou *et al.* proposed an attack to exploit the voice assistant of a smart car [161]. In this attack, the adversary inserts malicious commands in an audio or video file which can inject malicious commands to the voice assistant apps upon playing.

The smart voice assistant is deployed in several smart devices such as smartphone, smart speaker, smart home hub, etc. These smart assistants usually triggered with a specific command such as “Hi Google”, “Hey Siri”, or “Alexa”. Recent researches showed that it is possible to inject malicious commands to smart voice assistants by exploiting the microphone of the smart devices. As smart voice assistants constantly scan for desired a triggering command, an adversary needs no additional privilege to inject malicious audio signals to the device. Yuan *et al.* proposed *REEVE*, a stealthy voice manipulation attack to smart voice assistant [173]. *REEVE* uses benign audio signals such as TV or radio as a medium and insert malicious trigger commands which can be detected by a nearby voice assistant device. The researchers tested this attack on consumer voice assistant devices (Amazon Echo) and achieved high success rate. Zhang *et al.* improved this attack by eliminating the need of external audio signals [174]. Here, researchers developed a spyware which can abuse the microphone of a smartphone to record phone conversations and detect the trigger messages. Later, the spyware replays the recorded command using the speaker of the same smartphone to inject false commands to the voice assistant service.

Tippenhauer *et al.* showed another attack scenario in GPS-enabled devices to change the real location of the device [166]. In this attack scenario, a vehicle with a GPS enabled device is used. The attacker transmits a forged GPS signal to the device to alter the location of the vehicle. In this way, the real location of the vehicle is disguised and the attacker can perform any physical attack on the disguised vehicle. The GPS data used in the smartwatches can expose the location of a user and this GPS data can then be forged and a new location can be given as a false input in the GPS [167].

The power analysis attack on smart devices can also be used for injecting false data. The power analysis on smart devices running an encryption algorithm can reveal information about the encryption process including the block size, key size, even the actual encryption key [179]. This information can be used to encrypt a false data and replace the original data on the device. Thus, attackers can inject false encrypted data in the communication channel to change the action of a device for specific commands. Giannetsos and Dimitriou introduced a malicious app named *Spy-sense*, which monitors the behavior of the sensors in a device and can manipulate data by deleting or modifying it [168]. *Spy-sense* exploits the active memory region of a device and alters the data structure and reports back important data to a server covertly.

Lessons Learned for False Data Injection: False data injection impacts the integrity of the smart devices as these threats and attacks disrupt the output of an on-going task. From Table V, it is evident that the majority of the threats and attacks are simple and do not need any user interaction (8 out of 11 reported threats and attacks) to perform malicious tasks. Also, false data injection attacks are passive by nature (6 out of 11 threats) and do not need any excessive privilege (7 out of 11 threats) to perform the attack. Another interesting fact we observe is the effect of the successful attack directly impact the on-going activities of the smart devices. Hence, false data injection attacks are method-wise passive, but effect-wise active. However, the majority of the existing false data injection attacks did not report any success rate. Without proper evaluation, it is hard to understand the effectiveness of the reported attacks in real-life smart devices. Hence, further investigation is needed to properly evaluate the effectiveness of these attacks on real-life smart devices. The effects of false data injection are diverse as it can manipulate the targeted smart device to perform numerous malicious activities. For instance, false data injected in smart voice assistant can give the attacker access to any connected device in a smart environment which can cause device theft, undesired physical access to properties, unauthorized bank transactions and online shopping, etc., [173], [174].

D. Denial-of-Service

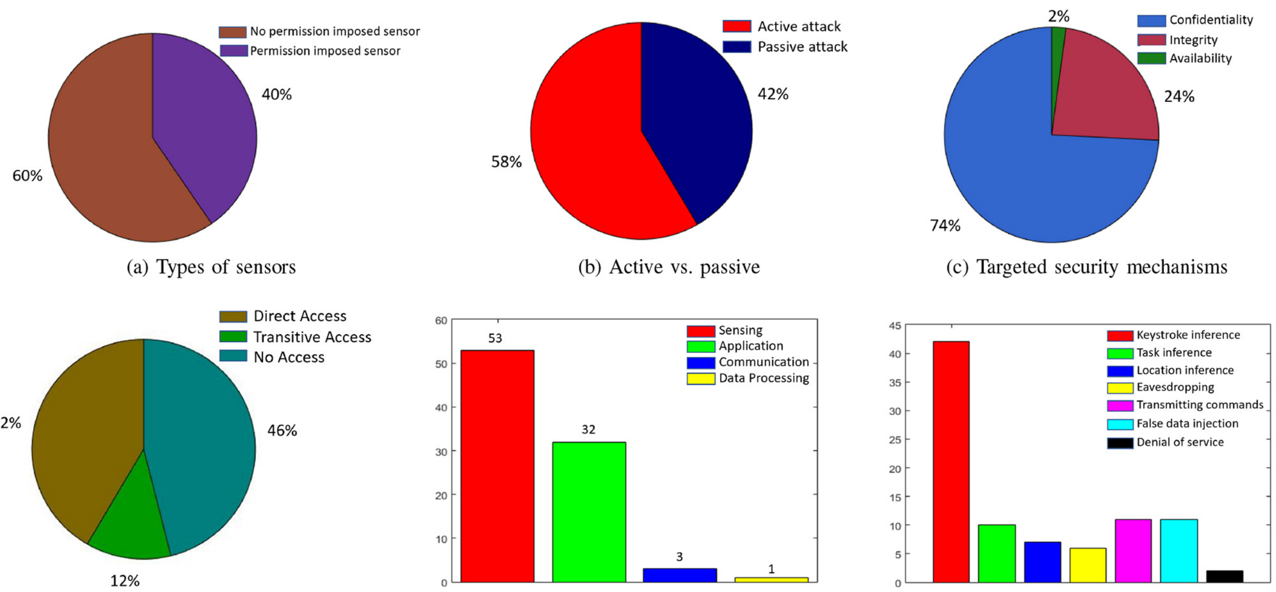
Denial-of-Service (DoS), by definition, is a type of attack where the normal operation of a device or application is denied maliciously. DoS attacks can be active attacks where an application or task is refused forcefully or passive attacks where attacking one application can stop another on-going task on the device. An adversary with access to smart device network and peripheral can send unauthorized access request or malicious signals to interrupt an ongoing task in smart devices (Section IV: use case 4). Indeed, recently, ICS-CERT published an active alert for a list of accelerometers used in smart devices which can be exploited using vibrational force [17]. Every accelerometer has a working frequency and if an external vibrational force can match this frequency, it is possible to turn off the devices forcefully. This reported threat is applicable for 20 different types of MEMS accelerometer which are used in multiple commercial and consumer smart devices. Hence, the impact of this threat is severe in real-life smart devices. Son *et al.* showed that it is possible to obstruct the flight control of a drone by exploiting gyroscope using a sound signal [8]. The MEMS Gyroscopes deployed in drones have a sensing mass inside of the sensor which is constantly vibrating. The gyroscope measures the rotational motion of the device with respect to the sensing mass. When the resonant frequency of the gyroscope is matched by an audio signal, an attacker can obstruct the normal performance of the gyroscope and change the course of the drone, or even turn it off. In a recent work, Mao *et al.* presented *Pairjam*, a DoS attack that uses inaudible noises to disrupt pairing between smart devices [176]. In a smart environment, multiple smart devices are connected with each other to perform various tasks. The interconnection between the devices follows a device authentication/pairing method to ensure secure communication. *Pairjam* abuses the audio sensor of smart devices the inject inaudible noise signal in the smart environment which disrupts the normal pairing method and makes a targeted smart device unavailable for pairing.

Lessons Learned for DoS: Denial-of-Service impacts the availability of the targeted sensor and disrupts an on-going task immediately in a smart device. There are only two reported DoS attack on smart devices which are passive and achieves high success rate (Table V). However, one of the reported DoS attack [8] uses the MEMS accelerometer and gyroscope which is used by a vast number of smart devices [17]. This increases the impact of DoS attacks on smart devices. Moreover, both of these DoS attacks are applicable to both standalone and connected smart devices which widens the attack surface. Hence, DoS attacks have a high impact on smart devices.

E. Summary of the Threats and Attacks

We categorize 89 reported sensor-based threats and attacks by the research community and industry in four categories. Additionally, we explain the attack methods and discuss the impacts of the sensor-based attacks based on common vulnerability metrics (attack impact, attack method, attack complexity, required privilege, user interaction, success rate). Some interesting findings of aforementioned sensor-based threats and attacks are listed below-

- *Type of Sensors Targeted:* Existing threats and attacks target nine different sensors including both permission and no-permission imposed sensors discussed in Section IV. One interesting fact we observe that the majority of the threats and attacks target no-permission imposed sensors which make the existing permission-based sensor



(a) Types of sensors

(b) Active vs. passive

(c) Targeted security mechanisms

(d) Device access

(e) Targeted layers

(f) Types of threats

Fig. 4. Overview of sensor-based threats and attacks to smart devices.

management system ineffective. From Figure 4(a), one can notice that 60% of the total threats and attacks target no-permission imposed sensors whereas only 40% of the reported threats and attacks target permission-imposed sensor which needs to bypass existing sensor management systems.

- *Active vs. Passive:* As previously mentioned, sensor-based threats and attacks can be active or passive depending on the attack method. From Figure 4(b), one can notice the high percentage of active sensor-based threats and attacks on smart devices. While the majority of these are active, passive attacks and threats are also a point of interest to the attackers. As passive threats or attacks do not affect the normal functionalities of the devices, these may remain unnoticed to the implemented security mechanisms and perform malicious activities surreptitiously.
- *Targeted Security Mechanisms:* Sensor-based threats and attacks target various security mechanisms of the devices (e.g., confidentiality, integrity, availability) which make them hard to detect. Figure 4(c) shows different security mechanisms targeted by the sensor-based threats and attacks. One can notice that most of these threats and attacks aim to violate data confidentiality (74%) followed by integrity (24%) and availability (2%) of the sensors.
- *Device Access Needs:* To perform malicious sensor activities, sensor-based threats and attacks need device access (direct or transitive). Figure 4(d) illustrates the device access needs of sensor-based threats. While approximately 54% of the threats need direct or transitive access (42% direct and 12% transitive), 47% of the threats do not need any access to execute the malicious activity. As a sensor-based threat without any need of device access can easily bypass any security mechanism, the impact of the threats is high. Additionally, the exclusiveness of device access needs of sensor-based threats manifests the shortcomings of the existing permission-based sensor management system.
- *Affected Layers:* As discussed in Section III, smart devices typically have four architectural layers and attackers can target any of these layers to initiate a sensor-based attack. From Figure 4(e), it is evident that the most affected layer is the sensing layer followed by the application layer. As modern smart devices offer diverse sets of apps that use sensors for enhanced functionalities, attackers target these layers to modify and perform malicious activities in smart devices. The apps in smart devices directly bind the sensing and application layer which is the main reason for increasing threats to these layers.
- *Interest of the Attackers:* From the discussion above, we can infer the most common sensor-based threats and attacks to smart devices is keystroke inferences followed by task inference and transmitting malicious sensor commands. As keystroke inference attacks typically target smart devices with user interfaces, we can observe higher number of sensor-based attacks in smartphones and smart watches. Figure 4(f) shows the common sensor-based threats and attacks to the smart devices.
- *Effect of the Sensor-Based Threats and Attacks:* In Table VI, we summarize the effects of sensor-based threats and attacks on smart devices. One can notice that keystroke inference attacks can leak diverse typing information such as passwords, PIN input, hand gestures, printed texts, etc. by exploiting a smart device directly or using a smart device to extract information from a nearby device. Task inference attacks reveal the nature of on-going tasks on smart devices either in the user interface of the device or in a connected smart environment. Sensor-based threats can also infer the geo-location of a smart device user as well as create a location map

TABLE VI
EFFECT OF SENSOR-BASED ATTACKS ON SMART DEVICES

Attack type	Effect	Reference
Keystroke Inference	Information of lock code or PIN of a smart device.	[82], [83], [87], [88], [99]–[101], [105]–[107], [109], [111], [113], [123]–[126], [132], [134]
	Typing information (typed text, notes, commands, etc.) in an embedded or virtual keyboard.	[110]
	Drawing patterns or unlock patterns on a smart device.	[103], [131], [133]
	Typing information of a nearby keyboard or device.	[14], [15], [84], [90], [98], [102], [112], [115]–[122], [128], [135]
	Printed information in a printer.	[127]
Task Inference	on-going task reflected on user interface of a smart device.	[93], [137]
	On-going task by a user in a single smart device or connected smart environment.	[72], [85]
	Information of an on-going task in a near by device.	[138]–[141], [145], [147]
Location Inference	Geo-location of a user interacting with a smart device.	[149], [150], [152]
	Mapping smart device users' motion and routes to detect location.	[153]–[155]
Eavesdropping	Capturing conversation of the users using on-board sensors surreptitiously.	[13], [89], [159]
	Stealthily record a conversation or command and replaying it to get access to a smart device and perform malicious activities.	[156], [157]
Transmitting malicious commands	Transmitting malicious sensor command to a nearby smart device and execute malicious activities.	[10], [11], [92], [162]–[165]
	Transmitting malicious commands using a smart device in a connected smart environment.	[33], [72]
False Sensor Data Injection	Injecting false sensor data to change the output of a smart device or a specific application running in a smart device application.	[166]–[173], [175]
Denial-of-Service	Using sensory channel to disrupt an on-going task in a smart device.	[8]
	Injecting inaccurate sensor signal to make a device unavailable in a smart environment.	[176]

of users' route. By performing an eavesdropping attack using sensors, an adversary can capture users' conversations and smart device commands to extract information and accessing a targeted device. An adversary transmitting malicious sensor commands can trigger malicious events on a smart device which can be propagated to nearby smart devices. Additionally, introducing false sensor data in a smart device can change the output of a smart device and make a device or sensor unavailable for performing a task (DoS). Also, external signal can interrupt an on-going task in smart devices by obstructing benign sensor activities.

F. Attack Comparison and Our Findings

Sensor-based threats are emerging attack vectors and have diverse malicious effects on smart devices. In this survey, we surveyed 89 directly reported threats and observed that the majority of the threats aim to extract sensor readings to infer sensitive user information. The selective sensor authorization in existing smart devices is the main reason for increasing number of information leakage attacks on smart devices' sensors. Besides, embedded sensors in smart devices are prune to passive sniffing which also leads to several information leakage attacks, including keystroke inference and eavesdropping. Compared to the information leakage, other types of sensor-based threats such as false data injection and transmitting malicious sensor commands are mostly active attacks which need to bypass implemented security measures. Hence, these active attacks are harder to execute in real-life smart devices.

Another interesting observation is the correlation of success rate and attack method in sensor-based threats and attacks. We observed that passive sensor-based attacks are easy to execute. However, the average success rate of passive attacks is lower than the active attacks as attackers need efficient analytical tools to perform the attack. For instance, passive information leakage attack uses sniffing techniques to capture sensor data. But, to extract information from passively captured data, attackers need extra analytical tools that can learn the sensor patterns and decode sensitive information from raw sensor data. On the contrary, active attacks on smart device sensors have higher success rate as attackers usually exploit the sensors by directly accessing the device. For instance, in an active eavesdropping attack, attackers can directly capture and record users' conversation by exploiting audio sensors [13], [157]. Here, no analytical tool is usually needed to extract information from the captured data.

As smart devices allow third-party app installation, the majority of the reported threats utilize third-party apps to exploit the sensors. We found several cases where a benign app source code is altered to capture sensitive sensor information from smart devices. In these cases, the installed third-party apps with malicious code snippets can directly capture sensitive information or raw sensor data which needs further analysis to extract information. We also observe a correlation between attack type and targeted sensors. As existing smart device platforms offer selective authorization for sensors such as GPS, microphone, and camera, threats targeting these sensors need to bypass the sensor authorization mechanism. Hence, sensor-based threats targeting GPS, microphone, and camera are usually active in nature and need excessive privilege to perform an attack. In contrast, threats targeting motion sensors and light sensors are relatively easy to execute and do not need any user interaction and excessive privileges.

In conclusion, our study show that passive sensor-based threats targeting no-permission imposed sensors can easily bypass existing security mechanisms in smart devices and can cause severe effect in terms of security and privacy of smart devices. However, while the issue of sensor-based threats have received attention from the developer and research communities, an additional investigation is needed to understand the diverse effect of sensor-based threats on smart devices. Also, researchers should study the existing sensor-based threats in detail to learn the future trends of zero-day sensor threats and enhance sensor security on smart devices accordingly.

VII. Existing Security Mechanisms to Prevent Sensor-Based Threats and Attacks

Researchers have identified a diverse set of sensor-based threats to smart devices. Tables III, IV, and V lists a summary of the existing sensor-based threats to smart devices. Although there are several threats, no comprehensive security mechanism to prevent such threats has been developed yet. Indeed, the use of a wide range of sensors in smart devices and applications has made it hard to secure all the sensors by one effective framework. Moreover, with the rise of IoT, securing sensors of a single smart device does not guarantee the security of all the connected devices. Furthermore, the lack of knowledge of the existing sensor-based threats and differences in sensor characteristics make it hard to establish a complete and comprehensive security measure to secure all the sensors of smart devices against the sensor-based threats [7]. In this section, we discuss three main approaches proposed by researchers in an attempt to design security mechanisms for sensor-based threats on smart devices. A summary of the existing solutions is given in Table VII.

A. Enhancing Existing Sensor Management Systems

One approach toward securing the sensors in smart devices is to enhance existing sensor management systems of smart device OSes. For instance, Xu and Zhu proposed an extension of the Android sensor management system named *SemaDroid*, which provides users with a monitoring and logging feature to make the usage of sensors by apps explicit [26]. Also, with *Semadroid*, users can specify policies to control whether and with what level of precision third-party apps can access to sensed data. Moreover, *SemaDroid* creates mock data to verify how applications, from unknown vendors, use sensed data and, thus, prevents malicious behaviors.

Furthermore, system designers have long struggled with the challenge of determining how to let the user control when applications may perform operations using privacy-sensitive sensors securely and effectively. Current commercial systems request that users authorize such operations once (i.e., on install or first use), but malicious apps may abuse such authorizations to collect data stealthily using such sensors. Proposed research methods enable systems to infer the operations associated with user input events [187]–[189], but malicious applications may still trick users into allowing unexpected, stealthy operations. To prevent users from being tricked, Petracca *et al.* proposed to bind applications' operation requests to the associated user input events and how such events are obtained explicitly, enabling users to authorize operations on privacy-sensitive sensors unambiguously [19], [21]. To demonstrate this solution, they implemented the *AWare* authorization framework for Android, extending the Android Middleware to control access to privacy-sensitive sensors. They evaluated the effectiveness of *AWare* in: (1) a laboratory-based user study, finding that at most 7% of the users were tricked by examples of four types of attacks when using *AWare*, instead of 85% on average for prior approaches; (2) a field study, showing that the user authorization effort increases by only 2.28 decisions on average per application; (3) a compatibility study with 1,000 of the most downloaded Android apps, demonstrating that such applications can operate effectively under *AWare*. A similar work is presented in *EnTrust*, where researchers implemented an improved authorization framework in Android to regulate sensor authorization based on input events and delegation graphs generated from co-operating programs [180]. *EnTrust* user authorization for any sensor authorization requests and remember the user's decision for similar requests for future authorization. Another recent work, *ContexIoT*, proposed an enhanced permission model for smart home devices [71]. ContexIoT observes the inter-procedure control and data flow in an app to determine the context of the app and forward the detail information to the users before allowing sensor access to the apps. Although, the context-aware approach gives users more information before allowing permissions, ContexIoT still depends on user decisions and an app can trick the user by obfuscating the code.

Lessons Learned for Enhancing Sensor Management Systems: Existing permission-based sensor management system lacks in securing all the sensors in smart devices and the aforementioned solutions enhance the existing system to be more robust and secure in terms of sensor security. However, permission-based sensor management still relies on user permissions which can be easily tricked as users may not be aware of the threats. Additionally, in a smart environment, sensor-based threats can use transitive permissions to access a smart device using the sensors of a connected device. In this case, permission-based sensor management systems may fail to protect the sensors from malicious attacks. In summary, enhanced sensor management systems improve the sensor security in smart devices significantly, but not comprehensively.

B. Intrusion Detection System

One common approach to secure a system from external attacks is to install an intrusion detection system (IDS). An IDS monitors the device and sensor states to detect suspicious activity and alert the system upon finding any vulnerability. In recent years, several prior works have proposed IDSs specifically to detect sensor-based threats to smart devices. A sensor-based threat detection method is proposed in *6thSense*, where researchers proposed a context-aware framework to detect the sensor-based threats in IoT devices [7], [40]. This framework is built upon the observation that for any user activity on an IoT device, a specific set of sensors becomes active. *6thSense* builds a comprehensive context-aware model for each user activity based on this observation. Different from other works, *6thSense* utilizes all the sensor data in real-time and determines whether the present context of the sensors is malicious or not using various machine learning-based approaches. Researchers tested the proposed framework with 50 real-life user data and confirmed that *6thSense* can detect various sensor-based threats with approximately 98% accuracy and F-score. In a later version of this work, the researchers implemented *6thSense* in a smartwatch and tested against several sensor-based threats [40]. Here, researchers collected user activity data from 100 real-life smartwatch users

TABLE VII
SUMMARY OF EXISTING SECURITY MECHANISMS TO PREVENT SENSOR-BASED THREATS

Solution Type	Solution name	Implemented platform	Summary	Performance	Limitations	Ref.
Enhancing existing sensor management systems	SemaDroid	Smart phone	• An extension of the Android sensor management system. • Covers both permission and no-permission imposed sensors. • Uses mock data to verify risky apps and offers flexible policy control to the users for sensors.	• Tested against existing sensor-based malware with 100% accuracy.	• Rely on user decisions. • Vendor-specific solution. • Ineffective against passive sensor-based threats.	[26]
	AWare	Smart phone	• An authorization framework to enable user authorization for sensitive sensor operations. • Binds applications' operation requests to the associated user input events and extends control access to privacy-sensitive sensors.	• Tested with real users to understand the user decisions. • Tested successfully with 1000 most downloaded Android Apps to check the compatibility	• Depends on user authorization. • Compatible with specific O/S	[19], [21]
	ContexIoT	Smart home devices	• An enhanced permission model for the connected smart home devices. • Analyzes the sensor access by installed apps at run-time and asks for user permission before executing a task.	• Tested with multiple types of malicious apps for the smart home devices. • Implemented in Samsung SmartThings platform and performed app patching to 283 commodity IoT apps.	• Depends on user permission. • Cannot detect cross-app interference at run-time.	[71]
	EnTrust	Smart phone	• An sensor authorization framework for Android platform. • Generates authorization requests based on input events of co-operating programs and delegation graphs.	• Tested in a lab environment. • Implemented in an Android smartphone with low overhead.	• Depends on user authorization and authorizes access to similar sensor requests.	[180]
Intrusion Detection System	6thSense	Smart phone, Smart watch	• A context-aware IDS for smart devices. • Considers all the sensors' states to build user activity contexts and detect malicious activities in the devices. • Uses multiple machine learning techniques to detect sensor-based threats.	• Tested against several sensor-based threats and achieved 96% accuracy. • Implemented in a smart phone and smartwatch and reveals minimal overhead.	• Needs initial training data for user activity and malicious apps.	[7]
	Aegis	Smart home devices	• An IDS to detect malicious apps abusing sensors in a connected smart environment. • Captures user behavior to build a context-aware model and train machine learning algorithms to detect malicious app and user behavior on sensors.	• Implemented in a Samsung SmartThings supported smart environment. • Tested with 22 types of smart devices and multiple layouts.	• Needs high number of smart devices in the system to capture user behaviors. • Depends on user feedback to improve performance.	[28], [181]
Protecting Sensed Data	Location-Privacy Preserving Mechanism	Smart Phone	• A novel approach to replicate real data with synthetic data to create a robust defense against white-box attack. • Uses targeted maneuvers to augment real sensors' data with synthetic data and obtain a uniform distribution of data points to increase diversity in data points.	• Tested with real dataset and reduced the probability of white-box attack by 3%.	• Sensor specific solution.	[20]
	SIRO	FPGA-based smart devices	• Uses single inverter ring oscillator to provide frequency hopping scheme in cipher.	• Tested with real FPGA devices and achieves minimal overhead.	• Only effective for specific devices and attacks.	[182]
	Protection against power analysis attack	FPGA-based smart devices	• A novel approach to minimize power analysis attack. • Correlates between power consumption and the number of bit transition to understand the effect of power analysis. • Uses randomize bit transition to minimize the effect of the power analysis attack.	• Conducted a detailed study to understand the effectiveness of proposed solution. • Performed security vs performance trade-off.	• No definite evaluation is provided. • Only effective against power analysis attack.	[183]
	AuDroid	Smart phone	• A trust management framework to analyze over-privilege sensor access requests by the apps. • Takes autonomous decisions on whether a sensor access is legitimate or not.	• Implemented in an Android powered smart phone. • Tested with 17 popular apps and achieved high accuracy.	• Sensor-specific solution.	[22]
	IoTDots	Smart home devices	• A forensic analysis tool to analyze sensor and device states in a connected smart environment to detect malicious behaviors of the devices. • Uses machine learning algorithms to detect malicious app and user behavior on sensors.	• Implemented in a Samsung SmartThings supported smart environment. • Tested with 22 different devices and achieved high accuracy and F1-Score.	• Focuses only on forensic analysis. • No real-time detection method implemented.	[184]
	Ditio	Smartphone, IoT devices	• A trust auditing tool to analyze sensor activity and defined policies for untrusted behavior. • Logs sensor activity in a server using authentication protocol and detects malicious sensor activity at a given time.	• Implemented in ARMJuno development board and Nexus 5 smartphone. • Low performance overhead in terms of CPU and memory usage.	• High power consumption. • No real-time detection method implemented.	[185]
	Peek-a-Boo	Smart home devices	• A network traffic spoofing method to protect sensor activities from packet sniffing.	• Simulated false packet injection to protect sensor information in network traffic.	• No real implementation.	[85]
App analysis	FlowFence	IoT devices	• A static analysis framework to enforce sensitive data flow control using opacified computation. • Inserts code snippets in an IoT app to track sensitive data flow and runs the app in a sandbox to detect malicious flow.	• Tested with several IoT apps from three different IoT platforms. • Achieves minimum overhead (4.9% latency)	• Ineffective against passive sensor-based attacks.	[186]
	SaINT	Smart home devices	• A static analysis framework to detect sensitive information flow using taint analysis. • Identifies a complete set of sources and sinks for smart home apps and detects information leakage.	• Tested with a total of 230 smart home apps. • Tested against 27 unique information leakage attacks in smart home.	• Ineffective against passive sensor-based attacks.	[72]

and achieved approximately 97% accuracy in detecting different sensor-based threats. In another recent work, researchers proposed *Aegis*, a context-aware intrusion detection system (IDS) for the smart connected environment [28], [181]. *Aegis* observes on-going user activities in a smart environment to learn how the state of smart devices change. Based on that, *Aegis* builds a context-aware model to detect malicious sensor activities in a smart environment. Researchers tested *Aegis* in several smart environment configurations and achieved over 97% accuracy in detecting different sensor-based threats.

Lessons Learned for IDS: With context-aware IDS proposed in [7] and [28], security of sensors in smart devices can be improved. However, to build the context-aware model, the system needs a higher number of sensors to correctly understand the user and device behavior model (the ground truth) [190]. Thus, proposed context-aware solutions are suitable for sensor-riched smart devices and the environment. Monitoring sensor data continuously can also increase the overhead in terms of power and CPU usage. For the real-life implementation of proposed IDSs, researchers should perform overhead analysis and proposed possible solutions to reduce the resource usage. In summary, context-aware sensor-based IDSs ensure comprehensive security to the smart device sensors, but introduce overhead in the system.

C. Protecting Sensed Data

Another approach toward securing smart devices against the sensor-based threats is to protect the sensed data in transfer and at rest. Indeed, malicious applications record sensor data and transmit it later when the device is locked or when security mechanisms are turned off. For instance, sensed location data may be subject to inference attacks by cyber-criminals that aim to obtain sensitive locations such as the victim's home and work locations to launch a variety of attacks.

Location-Privacy Preserving Mechanisms (LPPMs) exist to reduce the probability of success of inference attacks on location data. However, such mechanisms have been shown to be less effective when the adversary is informed of the protection mechanism adopted, also known as *white-box attacks*. Petracca *et al.* proposed a novel approach that makes use of targeted maneuvers to augment real sensors' data with synthetic data and obtain a uniform distribution of data points, which creates a robust defense against *white-box attacks* [20]. Such maneuvers are systematically activated in response to specific system events (i.e., internal state of sensors) to rapidly and continuously control the rate of change in system configurations and increase diversity in the space of readings, which would decrease the probability of success of inference attacks by an adversary. For instance, in the event of stationary states, devices leak more information about the location of the users such as stop position, home location, etc. The proposed solution activates random obfuscation as a maneuver which selects one protection mechanism from a set of mechanisms to increase the number of required guesses of an adversary. This ends up in reducing success rate of the adversary to leak location information. Proposed technique also implements two other maneuvers (spatial and temporal distribution) to deceit the adversary if there is no new data point over a longer period. Experimental results performed on a real data set showed that the adoption of such maneuvers reduces the probability of success of white-box attacks to 3% on average compared to 57% when using the state-of-the-art LPPMs. Acar *et al.* proposed a new approach to inject false packet in network traffic to protect sensor information from network sniffing. Here, the authors showed that it is possible to prevent sensor information leakage by modifying the feature vectors of the network packets and protect the sensor data.

Furthermore, power analysis attacks and electromagnetic emanation attacks exploit information from the power consumption and electromagnetic emissions of active sensors from the device. One proposed countermeasure to immune electromagnetic emanation attacks is to use a single inverter ring oscillator (SIRO) [182]. In this proposed system, a multiclock system with cipher embodiment is used with SIRO-based synchronization. The absence of external oscillator and unsynchronized nature of SIRO makes the system more immune to electromagnetic emission. Again, the SIRO-based system provides a frequency hopping scheme in cipher which increases immunity to timing and power analysis attacks. Standaert *et al.* proposed an approach to minimize the effect of power analysis attack which is based on the correlation between the power consumption measurements and a simple prediction developed on the number of bit transitions within the devices [183]. The use of random pre-charges in the devices can minimize the probability of power analysis attack on the FPGA-based smart devices.

More general solutions to address the protection of the sensed data have also been proposed. For example, Roman *et al.* proposed the use of public-key encryption to secure sensor data from devices [191]. They proposed the encryption of sensor data collected and stored it in the device before sharing it with third-party apps or other devices. Devices connected to each other can share their public key through a key management system and use their assigned private key to decrypt the sensor data. Third-party apps installed in the device can also use a public key encryption scheme to use sensor data for various applications.

Trust management frameworks can also be leveraged for secure information flow among sensors, secure communication of sensor data with other devices, and to certify authorized access of sensors by trusted software and apps in the system. Trust management frameworks can detect over-access requests on sensors and take decisions based on whether the requests are legitimate or not. For instance, a framework named *AuDroid* was proposed to secure communications via audio channels when applications make use of the device's microphones and speakers [22]. *AuDroid* leverages the SELinux kernel module to build a reference monitor which enforces access control policies over dynamically created audio channels. It controls information flows over audio channels and notifies users whenever an audio channel is created between processes at runtime. Mirzamohammadi *et al.* developed *Ditio*, a trustworthy auditing framework to capture and verify sensor activities with pre-defined policies [185]. *Ditio* uses an authentication protocol to connect with a secured server and log sensor activity to check compliance with enforced usage policies. It detects any untrusted sensor activity at a specified time by analyzing the logged data. Babun *et al.* proposed a forensic analysis tool to detect malicious user and app behavior on sensors in a smart environment. Authors considered the state of smart devices and sensors to build a state model of a smart environment and use machine learning algorithms to detect malicious behaviors on sensors.

Lessons Learned for Protecting Sensed Data: One interesting difference between existing sensor management systems and protecting sensed data is that many of the enhanced sensor management systems prevent sensor access to prevent sensor-based threats before executing whereas protecting sensed data schemes target to secure sensed data at run-time. Protecting sensed data can ensure sensor security against specific types of sensor-based threats and attacks (active information leakage, eavesdropping, etc.). However, passive sensor-based threats can still bypass the aforementioned solutions and execute malicious sensor activities. In addition, encrypting sensor data in a sensor-rich smart device can introduce overhead in terms of resource usage and latency.

D. App Analysis for Security and Privacy Invasion

Smart devices such as smartphones, smartwatches, smart home devices, etc. support different apps to provide multiple functions to the users. These apps can use the embedded sensors of the smart devices or external connected sensors (e.g., motion sensors in smart home systems) to perform various tasks. As current sensor management systems in smart devices offer selective restrictions on sensors, a malicious app can abuse the sensors to perform malicious activities in a device. One effective way to prevent sensor-based threats and attacks is to perform app analysis to detect malicious apps in the devices. There are two approaches to perform app analysis for smart devices – *static analysis* and *dynamic analysis*. In static analysis, the source code of the apps is analyzed to detect any malicious activity such as information leakage, transferring malicious codes, etc. One common static analysis approach is *taint analysis* where data entry points (source) and exit points (sink) are tainted to observe the information flow inside the app. In a smart device app, sensors are considered as sources and any communication method such as the Internet, text messaging, Bluetooth, etc. are considered as sinks. The taint analysis observes how the collected sensor data from the sources link to the sinks and any sensitive sensor information leakage via the sinks is revealed. Fernandes *et al.* proposed a static analysis framework named, *FlowFence*, which offers a language-independent taint analysis approach to detect information leakage in IoT apps [186]. FlowFence takes the source code of an untrusted IoT app, inserts code snippets to track sensitive data flow between the sources and sinks, and runs the app in a sandbox to detect malicious sensor data flow in the apps. FlowFence is tested with different IoT apps to determine its effectiveness against information leakage. However, FlowFence can only detect information leakage from the apps and fails to protect the sensors from side-channel attacks. Another static analysis tool, *SaINT*, is proposed by Celik *et al.* for the smart home platform [72]. SaINT specifically performs static analysis of smart home apps and detects sensitive information leakage by performing the taint analysis of the source code. SaINT analyzed a total of 230 smart home apps and reported sensitive information flow including sensor data leakage in smart home systems. However, SaINT fails to detect passive attacks such as trigger malware or transferring malicious codes using sensors.

Lessons Learned for app Analysis: As smart devices often use different smart apps to perform various tasks, app analysis can be an effective solution to detect sensor-based threats in the application layer. However, the majority of the sensor-based threats target the sensing layer (Section VI) and app analysis techniques often cannot detect these anomalies in source code [7]. Existing app analysis techniques are mostly static analysis [72] which needs the source code of an app to perform the analysis. This is a major drawback as the source code of the app may not be available to the users. Unlike enhanced sensor management systems and IDS, performing app analysis depends on user interaction (interacting with the tools) which requires technical knowledge. As users may not have required technical expertise, performing app analysis to detect sensor-based threats can be ineffective in real-life.

E. Shortcomings of Proposed Security Mechanisms

Although the aforementioned solutions address sensor-based threats and attacks, there are still limitations that need to be overcome.

(1) Most of the proposed security mechanisms for smart devices are anomaly detection frameworks at the application level which are not suitable for detecting sensor-based threats or attacks at the system level [90], [192]–[194]. Sikder *et al.* analyzed the performance of several sensor-based threats with respect to real-life malicious software scanners available in *VirusTotal* website and observed that no scanner can recognize sensor-based threats [7]. Celik *et al.* showed that Apps with malicious sensor logic in smart home devices cannot be detected via static analysis [72].

(2) With the growing popularity of the IoT concept, more and more smart devices are being interconnected with each other and the security of these devices becomes difficult to manage. Many smart devices are severely resource-limited, small devices and it is hard to implement a complex security mechanism considering the limited resources of the devices [195].

(3) Proposed security mechanisms only target a subset of sensitive sensors available in smart devices nowadays. For instance, commercial sensor management systems use an explicit permission-based security model for only some of the sensors (e.g., camera, GPS, and microphone). Similarly, *AuDroid* provides a policy-enforced framework to secure the audio sensors of smart devices explicitly [22]; however, such framework was not designed to protect other sensitive sensors. Other proposed solutions only provide protection against power analysis and electromagnetic emanation-based attacks, respectively [182], [183]. A step forward was made with *AWare* and *6thSense* that covered a wider set of privacy-sensitive sensors available in current smart devices to build a context-aware model and determine whether a sensor usage scenario is malicious.

(4) In solutions where users' decisions are utilized to build the sensor use policy for third-party apps, such as in *Semadroid*

TABLE VIII
COMPARISON BETWEEN EXISTING SECURITY MECHANISMS TO PREVENT SENSOR-BASED THREATS IN SMART DEVICES

Proposed Solution	Solution type	Attacks Covered[†]				Smart device Platform	OS Dependency	User Dependency	Sensor and Device dependency	Analytical model	Detection type	Overhead
		IL	TC	FD	DoS							
SemaDroid [26]	Sensor management	●	○	○	○	Standalone	●	●	●	○	●	○
AWare [19], [21]	Sensor management	●	○	○	○	Standalone	●	●	●	○	●	○
ContexIoT [71]	Sensor authorization	●	○	●	○	Connected	○	●	○	○	●	○
EnTrust [180]	Sensor authorization	●	○	○	○	Standalone	●	●	○	○	●	○
6thSense [7]	Intrusion detection	●	●	●	○	Standalone	○	○	○	●	●	●
Aegis [28]	Intrusion detection	●	●	●	●	Connected	○	○	○	●	●	○
Location-Privacy Preserving Mechanism [20]	Data spoofing	●	○	●	○	Standalone	○	○	●	○	N/A	○
SIRO [182]	Signal masking	●	○	○	○	Standalone	○	○	●	○	N/A	●
Protection against power analysis [183]	Data correlation and randomization	●	○	○	○	Standalone	●	○	●	○	N/A	●
AuDroid [22]	Sensor authorization	●	○	○	○	Standalone	●	○	●	○	●	○
IoTDots [184]	Forensic analysis	●	●	●	○	Connected	○	○	○	●	○	○
Ditio [185]	Sensor management	●	○	○	○	Standalone	○	○	○	○	○	●
Peek-a-Boo [85]	Data spoofing	●	○	○	○	Connected	○	○	○	●	○	○
FlowFence [186]	Static analysis	●	○	●	○	Connected	●	●	○	○	○	○
SaINT [72]	Static analysis	●	○	○	○	Connected	●	●	○	○	○	○

[†] IL - Information leakage, TC - Transmitting malicious sensor command, FD - False data injection, DoS - Denial-of-Service.

[‡] Smart device platform: Standalone smart device, connected smart devices; OS dependency: Dependent - ●, Independent - ○; User dependency: Dependent - ●, Independent - ○; Sensor and device dependency: Dependent - ●, Independent - ○; Analytical model: Supervised - ●, Unsupervised - ○; Detection type: Real-time - ●, static/after incident - ○; Overhead: High - ●, Low - ○.

and *AWare*, if a user allows an application to use a sensor without any restriction, then the application is blindly treated as secure by the system.

(5) Encrypting sensor data using public key encryption schemes provides protection to sensor data, but it also consumes high power to run in smaller smart devices [191]. This power-performance trade-off is impractical for resource-limited smart devices. In conclusion, a complete and comprehensive solution for autonomous policy enforcement, comprehensive coverage of all the sensors, and an efficient power-performance trade-off are yet to be designed.

F. Comparison of Security Mechanisms and Our Findings

While existing security mechanisms address sensor-based threats to some extent, further research is needed to develop comprehensive and efficient security mechanisms and tools to prevent sensor-based threats effectively. A comparison between existing security mechanisms is given in Table VIII. We notice that the majority of the existing security mechanisms only address information leakage by enhancing OS sensor management system and encrypting sensor data. Hence, threats such as transmitting malicious sensor command and false data injection can easily bypass these security mechanisms and exploit sensors in smart devices. Compared to a sensor management and authorization framework, an intrusion detection system can perform efficiently as it can address several sensor-based threats at run-time [7], [28]. However, the existing intrusion detection systems for smart devices use supervised learning which are ineffective for zero-day threats.

Another interesting observation is the user dependency in existing sensor management, authorization, and static analysis tools [72]. As attackers can easily trick the users to bypass installed sensor management systems and authorization rules, user dependency in security mechanisms may become ineffective against sensor-based threats. Again, static analysis tools require app source code and user interaction which can be undesirable to novice smart device users. Hence, compared to intrusion detection systems, the existing sensor management, authorization, and static analysis techniques perform ineffectively against sensor-based threats.

To protect sensor data, several prior works have proposed data encryption and spoofing mechanisms. While the proposed mechanisms effectively address information leakage at sensor level, the majority of the encryption and spoofing methods are sensor-specific and OS-dependent solutions. Also, data encryption at sensor level introduces high overhead which can affect the normal operation of smart devices. Researchers may study efficient end-to-end and sensor-independent encryption schemes to protect sensor data at rest and run-time. We also notice that there is no effective security mechanism that address denial-of-service (DoS) attacks in standalone smart devices (e.g., smartphones, smart watches, etc.). Although there are only two reported threats that perform DoS attacks in smart devices, lack of security measures may encourage attackers to develop novel DoS attacks targeting sensors.

In conclusion, while several prior works have proposed various security mechanisms to protect sensors in smart devices, we notice the absence of comprehensive understandings and security mechanisms to protect sensors from diverse sensor-based threats reported by research community and industry. Also, we observe the OS and user dependency in existing security mechanisms which impact the effectiveness in detecting sensor-based threats. Hence, further investigation is needed to understand the robustness of existing security mechanisms in different smart devices and platforms which would provide valuable insights to develop comprehensive mitigation techniques against sensor-based threats.

VIII. OPEN ISSUES, FUTURE DIRECTIONS, AND RECOMMENDATIONS

The concept of making devices "smart" is no longer in the developing stage and new research ideas related to smart devices are emerging these days. In this section, we discuss open issues and future research directions in the context of sensor-based threats and attacks to smart devices.

A. Open Issues and Future Directions

Due to the lack of knowledge among users and research communities, sensor-based threats become compelling to the attackers to exploit the security of smart devices and perform malicious activities. There are several open issues that exist in smart devices that need attention from developers, researchers, and users. These open issues can be categorized in three major areas - (1) Smart device architectures and platforms, (2) Further investigation of existing threats, and (3) Solutions to detect sensor-based threats. In the following discussion, we briefly explain these open issues and summarize future research directions needed to counter sensor-based threats.

Smart Device Architectures and Platforms: The smart device industry is growing rapidly and these smart devices are different from each other in terms of hardware, software, implementation, and functionalities. To understand the sensor-based threats, it is important to understand the smart device architecture and functionalities properly. Researchers and developers can investigate the following open issues in smart device architecture to understand the consequences of sensor-based threats properly.

Study of Smart Device Architectures and Sensor Operations: With the introduction of IoT, the number of smart devices in different domains is increasing rapidly. The smart devices have several internal architectures (i.e., software and hardware) with less knowledge available, which is an obstacle to secure sensors in these devices. For instance, there are several operating systems (e.g., Linux, Android, Contiki, TinyOS, etc.) available for smart devices which vary in terms of functionalities, operations, and integrated security features. Moreover, smart devices can connect with each other and create a network of smart devices to perform various tasks. The lack of knowledge of device architectures can affect the security of the devices as security flaws in one smart device can cause the compromise of other connected smart devices. Additionally, in a smart connected environment, multiple smart devices use one sensor to automate various tasks [28]. Hence, compromising one sensor can trigger malicious activities in several connected smart devices. Researchers and developers should study the smart device architectures (both standalone and connected smart devices) and functionalities to understand the sensor mechanism which will help to understand the consequences of emerging sensor-based threats.

Adoption of Standard Security Mechanisms: Currently, there exist several operating systems for smart devices that manage their on-board and external connected sensors in distinctive ways (Section IV). These dissimilarities make it hard to converge for a general security scheme to protect sensors of the smart devices [196]. For example, in a smart environment, several smart devices from different vendors can share the same sensors and physical environment. Any sensor-based threats compromising normal functionalities of a sensor can propagate to several connected smart devices. In this scenario, installing vendor-specific sensor security schemes surely increase the security of smart devices from a specific vendor. However, sensor-based threats targeting smart devices from another vendor can compromise connected smart devices even with an installed vendor-specific security scheme [28]. Moreover, installing different security schemes in different smart devices can lead to high resource usage and introduce overhead in the smart environment. Hence, a comprehensive vendor-independent sensor security scheme is needed to secure sensors of smart devices in a connected smart environment. One of the future research efforts should be the standardization of development platforms for smart devices which will make it easier for researchers to come up with universal security measures to defend against sensor-based threats and attacks. Therefore, researchers should investigate the possibility of a common security mechanism for authentication of sensor data as well as authorization of legitimate sensor access.

Fine-grained Control of Sensors: Existing sensor management systems of smart devices offer permission-based sensor management which completely depends on user consent. Apps generally ask for permission to access specific sensors on installation time and once the permissions are granted, users have less control over the sensors' usage by the apps. Again, the user permission is enforced only to secure a limited number of the on-board sensors (e.g., microphone, camera, GPS). Granting permission to these sensors automatically grant permission for other sensors such as accelerometer, gyroscope, light sensor, etc. In recent years, researchers have also shown that both permission-enforced (microphone, camera, GPS) and no permission-enforced (accelerometer, gyroscope, etc.) sensors are vulnerable to sensor-based threats and attacks. Therefore, a fine-grained sensor management system is needed to verify compliance between sensor access and user intent.

Further Investigation of Sensor-Based Threats: Several prior works have reported many sensor-based threats to smart devices in recent years. However, these sensor-based threats are unique from one another in terms of attack methods, targeted devices, and attack consequences. To understand sensor-based threats, it is important to study the existing threats and use the knowledge to enhance the security of smart devices to tackle new sensor-based threats.

Study of Malicious Sensor Behavior and User Perspectives: Sensor-based threats are relatively new and there are only a few comprehensive studies available to understand the threats properly. This lack of knowledge is lucrative for attackers to target and trick smart device users to install malicious apps and perform malicious sensor activities [197]. Users carelessly install any third-party apps with illegitimate sensor permissions which can compromise smart devices [74], [198]. Therefore, to secure sensors in smart devices, it is important to understand how users, smart devices, and apps are using sensors to perform and automate various tasks and what their views of sensor-based threats are. Researchers may perform additional usability studies to better understand how users can contribute to improving sensor access control via their inputs in smart devices.

Prevent Leakage of Sensor Data: Smart devices can autonomously sense their surrounding environment which can be used to prevent information leakage from the devices.

Sensors in smart devices can anticipate an on-going task and detect the pattern of information accessed by the task. These sensor patterns vary for different activities and by observing these sensor behaviors, it is possible to prevent information leakage in smart devices [7].

Control Sharing of Data among Sensors: Communication on smart devices become more sensor-to-sensor (i.e., machine-to-machine) compared to human-to-sensor or sensor-to-human (human-to-machine or machine-to-human) and the introduction of a huge number of sensors in smart devices is speeding up this shift. As smart devices deal with sensitive personal data, sensor-to-sensor communication channels should be secured, which helps in end-to-end security for the devices. Secure end-to-end communication from sensors to the devices and among devices is vital to avoid information leakage [199], [200]. Devices should share encrypted sensor data to avoid any information leakage via packet sniffing [85]. Sensor data should also be available to all the connected devices continuously to ensure unimpeded performance.

Privacy Concerns in Smart Device Sensors: Sensors in smart devices are associated with several tasks on smart devices that capture sensitive user inputs including user credentials, typed information, PIN code, etc. Hence, raw sensor data leaked from smart devices can lead to privacy violations in smart devices [201]. Attackers can utilize advanced techniques and analytical engines to learn sensitive information from sensor data and emulate user inputs to perform malicious activities such as accessing the device, alter device settings, etc. For instance, a malware installed in a smart device can capture keystrokes from sensor input and start injecting false keystrokes to perform malicious tasks while the device is on sleep mode [202]. Hence, it is important to ensure sensor data confidentiality in ongoing tasks of a smart device to protect user privacy. One effective solution can be run-time encryption of sensor data which can prevent information leakage from raw sensor data. Another possible solution can be sensor-assisted continuous authentication in connected smart devices to detect emulated user input [201], [203]. Further investigation is needed from the research community to develop emulated user input detection techniques and sensor encryption schemes to ensure user privacy at the sensor level.

Sensor-based Threats in Other Domains: Sensors have become ubiquitous not only in modern smart devices, but also sensor-assisted technologies are gaining popularity in various application domains. The diverse use of sensors in smart devices opens up the possibility of new security threats adopted from different application domains. For instance, sensor impersonation attack is a common threat vector in wireless sensor networks which can be easily adapted to exploit sensors in a multi-device smart environment [204], [205]. Also, a compromised sensor node is an interesting security issue in cyber-physical systems such as smart grid which can be modified to exploit sensors in smart devices [206], [207]. To address such adopted sensor-based threats, researchers may study OS-level and user-level sensor behavior to differentiate benign and compromised sensors in smart devices [208]. Another interesting research direction to address such threats can be to investigate correlation between user behavior and sensor behavior to identify compromised sensor nodes in smart devices. Hence, the study of adopted sensor-based threats in interconnected application domains can be an emerging research topic for both industry and research communities.

Security Measures for Sensor-Based Threats: As mentioned in Section VII, there are no comprehensive solutions to detect sensor-based threats in smart devices. The existing solutions focus on specific threats or sensors which are ineffective in addressing sensor-based threats extensively. Researchers and developers should focus on the following open issues to develop effective security measures to detect sensor-based threats properly.

Device Independent Security Measure: The majority of the existing solutions to secure sensors in smart devices focus on smartphone overlooking the security needs of other smart devices [31]. However, the number of different smart devices are also increasing rapidly. Several prior works have verified that not only smartphones but all the smart devices (e.g., smart watch, smart home devices, etc.) are vulnerable to emerging sensor-based threats [28], [209]. Additionally, smartphones can be used as a platform to launch sensor-based threats to other smart devices as smartphones act as controller device for several smart devices such as smart lock, smart camera, etc., [37]. Hence, researchers should consider sensor-based threat as a general threat to smart devices to develop device independent security measures.

Protect Sensor Data When at Rest: Smart device applications deal with multiple sensor data at a time and tampered data in the smart devices can impact the normal behavior of applications. To ensure the authenticity of sensor data, various end-to-end encryption mechanisms may be applied from the sensors to the program requesting it. Various security features of the hardware such as ARM TrustZone may be adopted to achieve secure data flow inside the devices [210]. Researchers may also invest their effort in studying the adoption of the blockchain technology as a way of designing highly distributed systems able to provide attestation and verification among multiparty and heterogeneous components part of a larger smart device ecosystem.

Protect Integrity of Sensor Operations: The research community has not invested enough effort in studying the design and development of tools for automated detection and analysis of sensors-based threats. For instance, no tool is available to automatically identify and analyze adversary-controlled sensors that would compromise the integrity of sensor operations, as well as the integrity of the data generated or modified by such operations. Also, no tool is available to automatically identify dangerous configurations in enforced access control policies, which may lead to risky operations by trusted programs that may compromise the integrity of the entire connected smart device environment.

Adoption of Intrusion Mechanisms to Detect Attacks: In recent years, multiple efficient techniques (e.g., machine learning (ML) and neural network (NN)) were applied to detect threats in various application domains. These detection techniques should be explored in detail to design novel intrusion detection mechanism, for smart devices and applications, able to identify when unsafe operations are authorized. Therefore, researchers should investigate NN and ML classification algorithms as viable solutions to identify and differentiate legitimate from illegal sensing activities. Another interesting approach is to study adversarial effects on sensor-based threat detection. Prior works showed ML-based intrusion detection can be pruned to adversarial attacks [211]. Hence, researchers and industry practitioners should investigate and develop mitigation strategies against adversarial attacks on sensors.

B. Recommendations

Vendors: Vendors have to consider the emerging sensor-based threats and attacks and get the security requirements right for every embedded and connected sensors. With the introduction of IoT, sensors can be external devices and connected via different communication means. Hence, vendors need to consider sensors as embedded components as well as independent before implementing security measures. Smart device vendors also should have a strong research strategy to understand the sensor-based threats and attacks and its consequences to secure the devices.

End-users: The main victims of the sensor-based threats and attacks are end-users. Attackers mostly target end-users with less technical knowledge of sensor-based threats to perform malicious activities such as information leakage, task inference, etc. Although it is hard to understand the technical part of different sensor-based threats and attacks, end-users should know the consequences of these threats and attacks and be cautious before using any risky apps in the devices. Additionally, end-users can follow good security practices such as rejecting any suspicious sensor access, disabling automatic data sharing between apps, etc. to secure their devices and information. Users can also raise their concerns to the vendors regarding sensor-based threats and attacks.

Developers: Developers can play an important role in securing smart devices against sensor-based threats and attacks. Modern app-based platforms increase the popularity of smart devices rapidly and developers can build numerous apps and publish them in app markets. To secure the devices from the sensor-based threats and attacks, developers can follow the guidelines published by the vendors to minimize the sensor data abuse in the apps [212]. Developers can also follow good app developing practices such as the use of encrypted sensor data in the app, trusted data flow path, use of only essential sensor permission, etc. Developers can also help the vendors to build specific security measures against the sensor-based threats and attacks.

Research community: Several on-going research efforts have already confirmed the necessity of securing sensors in smart devices [7], [163]. The research community can help the industry to address the sensor-based threats and attacks efficiently and propose various solutions. Researchers along with the industry experts should jointly propose a standard practice in app development to minimize the sensor abuses in smart devices. Furthermore, researchers should report newly found sensor-based threats to the vendors immediately to reduce the consequences.

Summary: In summary, there are several interesting research problems that may be tackled by the research community toward improving the security of sensors in smart devices and applications. While following the above directions toward better protection mechanisms against the sensor-based threats and attacks, researchers have to identify the key characteristics that differentiate IoT security from the commodity system security. Such unique characteristics will guide toward the design of innovative mechanisms that will be robust against the sensor attacks.

IX. Conclusion

The growing popularity of topics like smart home, smart office, smart city is increasing attention towards security issues in smart devices and applications. In this paper, we surveyed a lesser-known yet serious family of *sensor-based threats and attacks to smart devices*. To the best of our knowledge, this survey is the first one to address sensor-based threats and attacks as a major security issue to smart devices and classify these emerging threats and attacks formally. We presented a comprehensive overview of sensors in smart devices and existing sensor management systems used in commodity smart devices. We provided a detailed analysis of recent sensor-based threats and attacks and discussed how these threats and attacks can be used to exploits various sensors in smart devices. We also summarized several security approaches proposed by researchers in the attempt to address critical shortcomings for the security of these devices, and discussed some of the challenges for future research work in this area. In conclusion, we believe this survey will have a positive impact in the research community by documenting recent sensor-based threats and attacks to smart devices and motivating researchers to develop further comprehensive security schemes to secure these devices against sensor-based threats and attacks.

Acknowledgment

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes not with standing any copyright notation here on.

References

[1] N. Bari, G. Mani, and S. Berkovich, "Internet of Things as a methodological concept," in *Proc. IEEE 4th Int. Conf. Comput. Geospatial Res. Appl. (COM. Geo)*, 2013, pp. 48–55.

[2] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 140–150, Sep. 2010.

[3] Y. Yu, J. Wang, and G. Zhou, "The exploration in the education of professionals in applied Internet of Things engineering," in *Proc. IEEE 4th Int. Conf. Distance Learn. Educ. (ICDLE)*, 2010, pp. 74–77.

[4] A. K. Sikder, A. Acar, H. Aksu, A. S. Uluagac, K. Akkaya, and M. Conti, "IoT-enabled smart lighting systems for smart cities," in *Proc. IEEE 8th Annu. Comput. Commun. Workshop Conf. (CCWC)*, 2018, pp. 639–645.

[5] Statista. *Smart Home*. Accessed: Sep. 2019. [Online]. Available: https://www.statista.com/outlook/279/109/smart-home/united-states

[6] M. Kanellos. *152,000 Smart Devices Every Minute in 2025: IDC Outlines the Future of Smart Things*. Accessed: Mar. 2016. [Online]. Available: https://www.forbes.com/sites/michaelkanellos/2016/03/03/152000-smart-devices-every-minute-in-2025-idc-outlines-the-future-of-smart-things/#f055f744b63e

[7] A. K. Sikder, H. Aksu, and A. S. Uluagac, "6thSense: A context-aware sensor-based attack detector for smart devices," in *Proc. 26th USENIX Security Symp. (USENIX Security)*, Vancouver, BC, Canada, 2017, pp. 397–414.

[8] Y. Son *et al.*, "Rocking drones with intentional sound noise on gyroscopic sensors," in *Proc. USENIX Security*, 2015, pp. 881–896.

[9] A. Nahapetian, "Side-channel attacks on mobile and wearable systems," in *Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2016, pp. 243–247.

[10] V. Subramanian, A. S. Uluagac, H. Cam, and R. A. Beyah, "Examining the characteristics and implications of sensor side channels," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2013, pp. 2205–2210.

[11] R. Hasan, N. Saxena, T. Haleviz, S. Zawoad, and D. Rinehart, "Sensing-enabled channels for hard-to-detect command and control of mobile devices," in *Proc. 8th ACM SIGSAC Symp. Inf. Comput. Commun. Security*, 2013, pp. 469–480.

[12] R. Wijewickrama, A. Maiti, and M. Jadliwala, "deWristified: Handwriting inference using wrist-based motion sensors revisited," in *Proc. 12th Conf. Security Privacy Wireless Mobile Netw.*, 2019, pp. 49–59.

[13] R. Schlegel, K. Zhang, X.-Y. Zhou, M. Intwala, A. Kapadia, and X. Wang, "SoundComber: A stealthy and context-aware sound trojan for smartphones," in *Proc. NDSS*, vol. 11, 2011, pp. 17–33.

[14] L. Zhuang, F. Zhou, and J. D. Tygar, "Keyboard acoustic emanations revisited," *ACM Trans. Inf. Syst. Security*, vol. 13, no. 1, p. 3, 2009.

[15] A. Maiti, M. Jadliwala, J. He, and I. Bilogrevic, "(Smart) watch your taps: Side-channel keystroke inference attacks using smartwatches," in *Proc. ACM Int. Symp. Wearable Comput.*, 2015, pp. 27–30.

[16] S. M. D. Pozo, F.-X. Standaert, D. Kamel, and A. Moradi, "Side-channel attacks from static power: When should we care?" in *Proc. Design Autom. Test Europe Conf. Exhibit.*, 2015, pp. 145–150.

[17] *MEMs Accelerometer Hardware Design Flaws (Update A)*. Accessed: May 2017. [Online]. vailable: https://ics-cert.us-cert.gov/alerts/ICS-ALERT-17-073-01A

[18] A. S. Uluagac, V. Subramanian, and R. Beyah, "Sensory channel threats to cyber physical systems: A wake-up call," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2014, pp. 301–309.

[19] G. Petracca, A.-A. Reineh, Y. Sun, J. Grossklags, and T. Jaeger, "AWare: Preventing abuse of privacy-sensitive sensors via operation bindings," in *Proc. 26th USENIX Security Symp. (USENIX Security)*, 2017, pp. 379–396.

[20] G. Petracca, L. M. Marvel, A. Swami, and T. Jaeger, "Agility maneuvers to mitigate inference attacks on sensed location data," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, 2016, pp. 259–264.

[21] G. Petracca, A. Atamli, Y. Sun, J. Grossklags, and T. Jaeger, "AWare: Controlling app access to I/O devices on mobile platforms," 2016. [Online]. Available: arXiv:1604.02171.

[22] G. Petracca, Y. Sun, T. Jaeger, and A. Atamli, "AuDroid: Preventing attacks on audio channels in mobile devices," in *Proc. ACM 31st Annu. Comput. Security Appl. Conf.*, 2015, pp. 181–190.

[23] R. Templeman, Z. Rahman, D. Crandall, and A. Kapadia, "PlaceRaider: Virtual theft in physical spaces with smartphones," in *Proc. 20th Annu. Netw. Distrib. Syst. Security Symp. (NDSS)*, Feb. 2013, pp. 158–169.

[24] M. Kumar. (2019). *New Android Malware Apps Use Motion Sensor to Evade Detection*. [Online]. Available: https://thehackernews.com/2019/01/android-malware-play-store.html

[25] T. Micro. (2019). *Google Play Apps Drop Anubis Banking Malware, Use Motion-Based Evasion Tactics*. [Online]. Available: https://blog.trendmicro.com/trendlabs-security-intelligence/google-play-apps-drop-anubis-banking-malware-use-motion-based-evasion-tactics/

[26] Z. Xu and S. Zhu, "SemaDroid: A privacy-aware sensor management framework for smartphones," in *Proc. 5th ACM Conf. Data Appl. Security Privacy*, 2015, pp. 61–72.

[27] S. Arzt *et al.*, "FlowDroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps," *ACM SIGPLAN Notices*, vol. 49, no. 6, pp. 259–269, 2014.

[28] A. K. Sikder, L. Babun, H. Aksu, and A. S. Uluagac, "Aegis: A context-aware security framework for smart home systems," in *Proc. 35th Annu. Comput. Security Appl. Conf.*, 2019, pp. 28–41.

[29] J. Li, Y. Liu, T. Chen, Z. Xiao, Z. Li, and J. Wang, "Adversarial attacks and defenses on cyber-physical systems: A survey," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5103–5115, Jun. 2020.

[30] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet-scale IoT exploitations," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2702–2733, 3rd Quart., 2019.

[31] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and A. Ribagorda, "Evolution, detection and analysis of malware for smart devices," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 961–987, 2nd Quart., 2013.

[32] P. Bhat and K. Dutta, "A survey on various threats and current state of security in Android platform," *ACM Comput. Surveys*, vol. 52, no. 1, pp. 1–35, 2019.

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

[34] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 8, no. 2, pp. 2–23, 2nd Quart., 2006.

[35] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017.

[36] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: A survey on the security of IoT frameworks," *J. Inf. Security Appl.*, vol. 38, pp. 8–27, Feb. 2018.

[37] M. H. Khan and M. A. Shah, "Survey on security threats of smartphones in Internet of Things," in *Proc. 22nd Int. Conf. Autom. Comput. (ICAC)*, 2016, pp. 560–566.

[38] L. Cao, X. Jiang, Y. Zhao, S. Wang, D. You, and X. Xu, "A survey of network attacks on cyber-physical systems," *IEEE Access*, vol. 8, pp. 44219–44227, 2020.

[39] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[40] A. K. Sikder, H. Aksu, and A. S. Uluagac, "A context-aware framework for detecting sensor-based threats on smart devices," *IEEE Trans. Mobile Comput.*, vol. 19, no. 2, pp. 245–261, Feb. 2020.

[41] D. Dragomir, L. Gheorghe, S. Costea, and A. Radovici, "A survey on secure communication protocols for IoT systems," in *Proc. IEEE Int. Workshop Secure Internet Things (SIoT)*, 2016, pp. 47–62.

[42] I. Tomić and J. A. McCann, "A survey of potential security issues in existing wireless sensor network protocols," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1910–1923, Dec. 2017.

[43] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "IoT middleware: A survey on issues and enabling technologies," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 1–20, Feb. 2017.

[44] M. La Polla, F. Martinelli, and D. Sgandurra, "A survey on security for mobile devices," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 446–471, 1st Quart., 2012.

[45] M. Caprolu, S. Sciancalepore, and R. Di Pietro, "Short-range audio channels security: Survey of mechanisms, applications, and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 32, no. 1, pp. 311–340, 1st Quart., 2021.

[46] S. A. Hamad, Q. Z. Sheng, W. E. Zhang, and S. Nepal, "Realizing an Internet of Secure Things: A survey on issues and enabling technologies," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1372–1391, 2nd Quart., 2020.

[47] J. Sengupta, S. Ruj, and S. D. Bit, "A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT," *J. Netw. Comput. Appl.*, vol. 149, Jan. 2020, Art. no. 102481.

[48] A. Newaz, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, "A survey on security and privacy issues in modern healthcare systems: Attacks and defenses," 2020. [Online]. Available: arXiv:2005.07359.

[49] C. Yan, H. Shin, C. Bolton, W. Xu, Y. Kim, and K. Fu, "SoK: A minimalist approach to formalizing analog sensor security," in *Proc. IEEE Symp. Security Privacy (SP)*, 2020, pp. 233–248.

[50] S. Alnefaie, S. Alshehri, and A. Cherif, "A survey on access control in IoT: Models, architectures and research opportunities," *Int. J. Security Netw.*, vol. 16, no. 1, pp. 60–76, 2021.

[51] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1294–1312, 3rd Quart., 2015.

[52] K. Zhao and L. Ge, "A survey on the Internet of Things security," in *Proc. IEEE 9th Int. Conf. Comput. Intell. Security*, 2013, pp. 663–667.

[53] L. Chen *et al.*, "Robustness, security and privacy in location-based services for future IoT: A survey," *IEEE Access*, vol. 5, pp. 8956–8977, 2017.

[54] N. Yousefnezhad, A. Malhi, and K. Främling, "Security in product lifecycle of IoT devices: A survey," *J. Netw. Comput. Appl.*, vol. 171, Dec. 2020, Art. no. 102779.

[55] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for Internet of Things," *J. Netw. Comput. Appl.*, vol. 42, pp. 120–134, Jun. 2014.

[56] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2671–2701, 3rd Quart., 2019.

[57] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 266–282, 1st Quart., 2013.

[58] O. Can and O. K. Sahingoz, "A survey of intrusion detection systems in wireless sensor networks," in *Proc. IEEE 6th Int. Conf. Model. Simulat. Appl. Optim. (ICMSAO)*, 2015, pp. 1–6.

[59] I. Butun, P. Österberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, attacks and countermeasures," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 616–644, 1st Quart., 2020.

[60] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1646–1685, 3rd Quart., 2020.

[61] A. Hamza, H. H. Gharakheili, and V. Sivaraman, "IoT network security: Requirements, threats, and countermeasures," 2020. [Online]. Available: arXiv:2008.09339.

[62] S. Poslad, *Ubiquitous Computing: Smart Devices, Environments and Interactions*. Hoboken, NJ, USA: Wiley, 2011.

[63] M. Silverio-Fernández, S. Renukappa, and S. Suresh, "What is a smart device? A conceptualisation within the paradigm of the Internet of Things," *Visual. Eng.*, vol. 6, no. 1, p. 3, 2018.

[64] *Smartthings Developer Documentation*. Accessed: Jul. 7, 2017. [Online]. Available: http://docs.smartthings.com/en/latest/architecture/index.html

[65] C. Salzmann and D. Gillet, "Smart device paradigm, standardization for online labs," in *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, 2013, pp. 1217–1221.

[66] G. C. Meijer *et al.*, *Smart Sensor Systems*, vol. 7. New York, NY, USA: Wiley, 2008.

[67] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things architecture, possible applications and key challenges," in *Proc. IEEE 10th Int. Conf. Front. Inf. Technol. (FIT)*, 2012, pp. 257–260.

[68] C. Perera, P. Jayaraman, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Dynamic configuration of sensors using mobile sensor hub in Internet of Things paradigm," in *Proc. IEEE 8th Int. Conf. Intell. Sens. Sensor Netw. Inf. Process.*, 2013, pp. 473–478.

[69] A. F. A. Rahman, M. Daud, and M. Z. Mohamad, "Securing sensor to cloud ecosystem using Internet of Things (IoT) security framework," in *Proc. Int. Conf. Internet Things Cloud Comput.*, 2016, pp. 1–5.

[70] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.

[71] Y. J. Jia *et al.*, "ContexIoT: Towards providing contextual integrity to appified IoT platforms," in *Proc. NDSS*, vol. 2, no. 2, 2017, p. 2.

[72] Z. B. Celik *et al.*, "Sensitive information tracking in commodity IoT," in *Proc. 27th USENIX Security Symp. (USENIX Security)*, Baltimore, MD, USA, 2018, pp. 1687–1704.

[73] A. K. Sikder *et al.*, "Kratos: Multi-user multi-device-aware access control system for the smart home," in *Proc. 13th ACM Conf. Security Privacy Wireless Mobile Netw.*, 2020, pp. 1–12.

[74] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *Proc. ACM 8th Symp. Usable Privacy Security*, 2012, p. 3.

[75] *Apple Developer Documentation*. Accessed: Dec. 1, 2015. [Online]. Available: https://developer.apple.com/documentation

[76] *Sensor Overview*. Accessed: Oct. 23, 2017. [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_overview.html

[77] *Who Leads OS Share in Internet of Things Era?* Accessed: Oct. 23, 2017. [Online]. Available: https://spectrummattersindeed.blogspot.com/2017/04/who-leads-os-share-in-internet-of.html

[78] *Sensor Stack*. Accessed: Mar. 10, 2017. [Online]. Available: https://source.android.com/devices/sensors/sensor-stack.html

[79] *Introduction to the Sensor and Location Platform in Windows*. Accessed: Mar. 10, 2017. [Online]. Available: https://msdn.microsoft.com/en-us/library/windows/desktop/dd318936(v=vs.85).aspx

[80] *Sensors*. Accessed: Mar. 10, 2017. [Online]. Available: https://developer.blackberry.com/native/documentation/device_comm/sensors/

[81] *Core Motion*. Accessed: Oct. 23, 2017. [Online]. Available: https://developer.apple.com/documentation/coremotion

[82] C. Shen, S. Pei, Z. Yang, and X. Guan, "Input extraction via motion-sensor behavior analysis on smartphones," *Comput. Security*, vol. 53, pp. 143–155, Sep. 2015.

[83] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "Accessory: Password inference using accelerometers on smartphones," in *Proc. 12th Workshop Mobile Comput. Syst. Appl.*, 2012, p. 9.

[84] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp) iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers," in *Proc. 18th ACM Conf. Comput. Commun. Security*, 2011, pp. 551–562.

[85] A. Acar *et al.*, "Peek-a-boo: I see your smart home activities, even encrypted!" in *Proc. 13th ACM Conf. Security Privacy Wireless Mobile Netw.*, 2020, pp. 207–218.

[86] W. M. Kang, S. Y. Moon, and J. H. Park, "An enhanced security framework for home appliances in smart home," *Human Centric Comput. Inf. Sci.*, vol. 7, no. 1, p. 6, 2017.

[87] L. Simon and R. Anderson, "Pin skimmer: Inferring pins through the camera and microphone," in *Proc. 3rd ACM Workshop Security Privacy Smartphones Mobile Devices*, 2013, pp. 67–78.

[88] A. Al-Haiqi, M. Ismail, and R. Nordin, "Keystrokes inference attack on Android: A comparative evaluation of sensors and their fusion," *J. ICT Res. Appl.*, vol. 7, no. 2, pp. 117–136, 2013.

[89] Y. Michalevsky, D. Boneh, and G. Nakibly, "Gyrophone: Recognizing speech from gyroscope signals," in *Proc. USENIX Security Symp.*, 2014, pp. 1053–1067.

[90] H. Wang, T. T.-T. Lai, and R. R. Choudhury, "MoLe: Motion leaks through smartwatch sensors," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 155–166.

[91] B. Celik *et al.* (2018). *A Micro-Benchmark Suite to Assess the Effectiveness of Tools Designed for IoT Apps*. [Online]. Available: https://github.com/IoTBench/

[92] C. Yan, G. Zhang, X. Ji, T. Zhang, T. Zhang, and W. Xu, "The feasibility of injecting inaudible voice commands to voice assistants," *IEEE Trans. Depend. Secure Comput.*, early access, Mar. 19, 2019, doi: 10.1109/TDSC.2019.2906165.

[93] A. Maiti and M. Jadliwala, "Light ears: Information leakage via smart lights," 2018. [Online]. Available: arXiv:1808.07814.

[94] H.-W. Choi and H. Kim, "Impersonation attacks on anonymous user authentication and key agreement scheme in wireless sensor networks," *J. Digit. Converg.*, vol. 14, no. 10, pp. 287–293, 2016.

[95] M. Barbeau, J. Hall, and E. Kranakis, "Detecting impersonation attacks in future wireless and mobile networks," in *Proc. Int. Workshop Secure Mobile Ad Hoc Netw. Sensors*, 2005, pp. 80–95.

[96] S. A. Salehi, M. A. Razzaque, P. Naraei, and A. Farrokhtala, "Security in wireless sensor networks: Issues and challanges," in *Proc. IEEE Int. Conf. Space Sci. Commun. (IconSpace)*, 2013, pp. 356–360.

[97] First.org. (2019). *Common Vulnerability Scoring System Version 3.1: Specification Document*. [Online]. Available: https://www.first.org/cvss/specification-document

[98] A. Maiti, M. Jadliwala, J. He, and I. Bilogrevic, "Side-channel inference attacks on mobile keypads using smartwatches," *IEEE Trans. Mobile Comput.*, vol. 17, no. 9, pp. 2180–2194, Sep. 2018.

[99] S. Narain, A. Sanatinia, and G. Noubir, "Single-stroke language-agnostic keylogging using stereo-microphones and domain specific machine learning," in *Proc. ACM Conf. Security Privacy Wireless Mobile Netw.*, 2014, pp. 201–212.

[100] R. Spreitzer, "Pin skimming: Exploiting the ambient-light sensor in mobile devices," in *Proc. 4th ACM Workshop Security Privacy Smartphones Mobile Devices*, 2014, pp. 51–62.

[101] L. Cai and H. Chen, "On the practicality of motion based keystroke inference attack," in *Proc. Int. Conf. Trust Trustworthy Comput.*, 2012, pp. 273–290.

[102] Y. Huang, X. Guan, H. Chen, Y. Liang, S. Yuan, and T. Ohtsuki, "Risk assessment of private information inference for motion sensor embedded iot devices," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 3, pp. 265–275, Jun. 2020.

[103] A. J. Aviv, B. Sapp, M. Blaze, and J. M. Smith, "Practicality of accelerometer side channels on smartphones," in *Proc. ACM 28th Annu. Comput. Security Appl. Conf.*, 2012, pp. 41–50.

[104] A. Acar, H. Aksu, A. S. Uluagac, and K. Akkaya, "WACA: Wearable-assisted continuous authentication," in *Proc. IEEE Security and Privacy Workshops (SPW)*, May 2018, pp. 264–269.

[105] L. Cai and H. Chen, “TouchLogger: Inferring keystrokes on touch screen from smartphone motion,” *Proc. HotSec*, vol. 11, 2011, p. 9.

[106] J. Lin and J. Seibel. (2009). *Motion-Based Side-Channel Attack on Mobile Keystrokes*. [Online]. Available: https://pdfs.semanticscholar.org/95cb/6a266e7a7319334775d8c89e353adf9b514e.pdf

[107] Z. Xu, K. Bai, and S. Zhu, “TapLogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors,” in *Proc. 5th ACM Conf. Security Privacy Wireless Mobile Netw.*, 2012, pp. 113–124.

[108] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, “TapPrints: Your finger taps have fingerprints,” in *Proc. ACM 10th Int. Conf. Mobile Syst. Appl. Services*, 2012, pp. 323–336.

[109] T. Nguyen, “Using unrestricted mobile sensors to infer tapped and traced user inputs,” in *Proc. IEEE 12th Int. Conf. Inf. Technol. New Gener. (ITNG)*, 2015, pp. 151–156.

[110] D. Hodges and O. Buckley, “Reconstructing what you said: Text inference using smartphone motion,” *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 947–959, Apr. 2019.

[111] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, “Deep learning based inference of private information using embedded sensors in smart devices,” *IEEE Netw.*, vol. 32, no. 4, pp. 8–14, Jul./Aug. 2018.

[112] Y. Liu and Z. Li, “aLeak: Context-free side-channel from your smart watch leaks your typing privacy,” *IEEE Trans. Mobile Comput.*, vol. 19, no. 8, pp. 1775–1788, Aug. 2020.

[113] L. Bo, L. Fengjun, and W. Guanghui, “I know what you type on your phone: Keystroke inference on Android device using deep learning,” Ph.D. dissertation, Elect. Eng. Comput. Sci., Univ. Kansas, Lawrence, KS, USA, 2019.

[114] Z. Ji, Z.-Y. Li, P. Li, and M. An, “A new effective wearable hand gesture recognition algorithm with 3-axis accelerometer,” in *Proc. IEEE 12th Int. Conf. Fuzzy Syst. Knowl. Disc. (FSKD)*, 2015, pp. 1243–1247.

[115] X. Liu, Z. Zhou, W. Diao, Z. Li, and K. Zhang, “When good becomes evil: Keystroke inference with smartwatch,” in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Security*, 2015, pp. 1273–1285.

[116] A. Sarkisyan, R. Debbiny, and A. Nahapetian, “WristSnoop: Smartphone pins prediction using smartwatch motion sensors,” in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, 2015, pp. 1–6.

[117] Z. Ling, Z. Li, C. Chen, J. Luo, W. Yu, and X. Fu, “I know what you enter on gear VR,” in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2019, pp. 241–249.

[118] A. Maiti, R. Heard, M. Sabra, and M. Jadliwala, “Towards inferring mechanical lock combinations using wrist-wearables as a side-channel,” in *Proc. 11th ACM Conf. Security Privacy Wireless Mobile Netw.*, 2018, pp. 111–122.

[119] D. Asonov and R. Agrawal, “Keyboard acoustic emanations,” in *Proc. IEEE Symp. Security Privacy*, 2004, pp. 3–11.

[120] T. Halevi and N. Saxena, “A closer look at keyboard acoustic emanations: Random passwords, typing styles and decoding techniques,” in *Proc. 7th ACM Symp. Inf. Comput. Commun. Security*, 2012, pp. 89–90.

[121] Y. Berger, A. Wool, and A. Yeredor, “Dictionary attacks using keyboard acoustic emanations,” in *Proc. 13th ACM Conf. Comput. Commun. Security*, 2006, pp. 245–254.

[122] D. Foo Kune and Y. Kim, “Timing attacks on pin input devices,” in *Proc. 17th ACM Conf. Comput. Commun. Security*, 2010, pp. 678–680.

[123] L. Lu *et al.*, “KeyListener: Inferring keystrokes on qwerty keyboard of touch screen through acoustic signals,” in *Proc. IEEE Int. Conf. Comput. Commun.*, 2019, pp. 775–783.

[124] I. Shumailov, L. Simon, J. Yan, and R. Anderson, “Hearing your touch: A new acoustic side channel on smartphones,” 2019. [Online]. Available: arXiv:1903.11137.

[125] H. Kim, B. Joe, and Y. Liu, “TapSnoop: Leveraging tap sounds to infer tapstrokes on touchscreen devices,” *IEEE Access*, vol. 8, pp. 14737–14748, 2020.

[126] M. Zhou *et al.*, “Stealing your Android patterns via acoustic signals,” *IEEE Trans. Mobile Comput.*, vol. 20, no. 4, pp. 1656–1671, 2021.

[127] M. Backes, M. Dürmuth, S. Gerling, M. Pinkal, and C. Sporleder, “Acoustic side-channel attacks on printers,” in *Proc. USENIX Security Symp.*, 2010, pp. 307–322.

[128] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, “Context-free attacks using keyboard acoustic emanations,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2014, pp. 453–464.

[129] S. R. Chhetri, A. Canedo, and M. A. Al Faruque, *Poster: Exploiting Acoustic Side-Channel for Attack on Additive Manufacturing Systems*, Univ. California at Irvine, Irvine, CA, USA, 2016.

[130] C. Song, F. Lin, Z. Ba, K. Ren, C. Zhou, and W. Xu, “My smartphone knows what you print: Exploring smartphone-based side-channel attacks against 3D printers,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2016, pp. 895–907.

[131] W. Meng, W. H. Lee, S. Murali, and S. Krishnan, “Charging me and i know your secrets! Towards juice filming attacks on smartphones,” in *Proc. 1st ACM Workshop Cyber Phys. Syst. Security*, 2015, pp. 89–98.

[132] D. Shukla, R. Kumar, A. Serwadda, and V. V. Phoha, “Beware, your hands reveal your secrets!” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2014, pp. 904–917.

[133] A. J. Aviv, “Side channels enabled by smartphone interaction,” Ph.D. dissertation, Comput. Inf. Sci., Pennsylvania State Univ., State College, CA, USA, 2012.

[134] R. Raguram, A. M. White, D. Goswami, F. Monrose, and J.-M. Frahm, “iSPY: Automatic reconstruction of typed input from compromising reflections,” in *Proc. 18th ACM Conf. Comput. Commun. Security*, 2011, pp. 527–536.

[135] Y. Wang, W. Cai, T. Gu, and W. Shao, “Your eyes reveal your secrets: An eye movement based password inference on smartphone,” *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2714–2730, Nov. 2020.

[136] M. Vuagnoux and S. Pasini, “Compromising electromagnetic emanations of wired and wireless keyboards,” in *Proc. USENIX Security Symp.*, 2009, pp. 1–16.

[137] S. Chakraborty, W. Ouyang, and M. Srivastava, “Lightspy: Optical eavesdropping on displays using light sensors on mobile devices,” in *Proc. IEEE Int. Conf. Big Data*, 2017, pp. 2980–2989.

[138] S. Biedermann, S. Katzenbeisser, and J. Szefer, “Hard drive side-channel attacks using smartphone magnetic field sensors,” in *Proc. Int. Conf. Financial Cryptography Data Security*, 2015, pp. 489–496.

[139] R. Ning, C. Wang, C. Xin, J. Li, and H. Wu, “DeepMag+: Sniffing mobile apps in magnetic field through deep learning,” *Pervasive Mobile Comput.*, vol. 61, Jan. 2020, Art. no. 101106.

[140] N. Matyunin, Y. Wang, T. Arul, K. Kullmann, J. Szefer, and S. Katzenbeisser, “Magneticspy: Exploiting magnetometer in mobile devices for website and application fingerprinting,” in *Proc. 18th ACM Workshop Privacy Electron. Soc.*, 2019, pp. 135–149.

[141] J.-J. Quisquater and D. Samyde, “Electromagnetic analysis (EMA): Measures and counter-measures for smart cards,” in *Proc. Smart Card Program. Security*, 2001, pp. 200–210.

[142] V. Carlier, H. Chabanne, E. Dottax, and H. Pelletier, “Electromagnetic side channels of an FPGA implementation of AES,” in *Proc. Cryptol. Eprint Archive*, 2004, p. 145.

[143] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, “Theem side—Channel (s),” in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2002, pp. 29–45.

[144] Y. Ren and L. Wu, “Power analysis attacks on wireless sensor nodes using CPU smart card,” in *Proc. IEEE 22nd Wireless Opt. Commun. Conf. (WOCC)*, 2013, pp. 665–670.

[145] Y. Cheng *et al.*, “MagAttack: Guessing application launching and operation via smartphone,” in *Proc. ACM Asia Conf. Comput. Commun. Security*, 2019, pp. 283–294.

[146] S. B. Örs, E. Oswald, and B. Preneel, “Power-analysis attacks on an FPGA—First experimental results,” in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2003, pp. 35–50.

[147] C. O’Flynn and Z. Chen, “Power analysis attacks against IEEE 802.15. 4 nodes,” in *Proc. Int. Workshop Construct. Side Channel Anal. Secure Design*, 2016, pp. 55–70.

[148] A. I. Newaz, A. K. Sikder, L. Babun, and A. S. Uluagac, “HEKA: A novel intrusion detection system for attacks to personal medical devices,” in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2020, pp. 1–9.

[149] Anonymous. (2016). *VoIPLoc: Compromising Location-Privacy Via Acoustic Side-Channel Attacks*,” [Online]. Available: https://www.semanticscholar.org/paper/VoipLoc-%3A-Compromising-location-privacy-via-attacks/b3a04badcab8e68491277735ceb4dcd12c3e3f71

[150] J. Han, E. Owusu, L. T. Nguyen, A. Perrig, and J. Zhang, “Accomplice: Location inference using accelerometers on smartphones,” in *Proc. IEEE 4th Int. Conf. Commun. Syst. Netw.*, 2012, pp. 1–9.

[151] X. Zhou *et al.*, “Identity, location, disease and more: Inferring your secrets from Android public resources,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2013, pp. 1017–1028.

[152] K. Block and G. Noubir, “My magnetometer is telling you where i’ve been? A mobile device permission less location attack,” in *Proc. 11th ACM Conf. Security Privacy Wireless Mobile Netw.*, 2018, pp. 260–270.

[153] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir, “Inferring user routes and locations using zero-permission mobile sensors,” in *Proc. IEEE Symp. Security Privacy (SP)*, 2016, pp. 397–413.

[154] H. Zheng and H. Hu, "MISSILE: A system of mobile inertial sensor-based sensitive indoor location eavesdropping," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3137–3151, 2020.

[155] Z. Fyke, I. Griswold-Steiner, and A. Serwadda, "Prying into private spaces using mobile device motion sensors," in *Proc. IEEE 17th Int. Conf. Privacy Security Trust (PST)*, 2019, pp. 1–10.

[156] W. Diao, X. Liu, Z. Zhou, and K. Zhang, "Your voice assistant is mine: How to abuse speakers to steal information and control your phone," in *Proc. 4th ACM Workshop Security Privacy Smartphones Mobile Devices*, 2014, pp. 63–74.

[157] L. Lei, Y. Wang, J. Zhou, D. Zha, and Z. Zhang, "A threat to mobile cyber-physical systems: Sensor-based privacy theft attacks on Android smartphones," in *Proc. 12th IEEE Int. Conf. Trust Security Privacy Comput. Commun.*, 2013, pp. 126–133.

[158] N. Carlini *et al.*, "Hidden voice commands," in *Proc. 25th USENIX Security Symp.*, 2016, pp. 513–530.

[159] S. A. Anand, C. Wang, J. Liu, N. Saxena, and Y. Chen, "SpearPhone: A speech privacy exploit via accelerometer-sensed reverberations from smartphone loudspeakers," 2019. [Online]. Available: arXiv:1907.05972.

[160] S. Kennedy, H. Li, C. Wang, H. Liu, B. Wang, and W. Sun, "I can hear your alexa: Voice command fingerprinting on smart home speakers," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2019, pp. 232–240.

[161] M. Zhou, Z. Qin, X. Lin, S. Hu, Q. Wang, and K. Ren, "Hidden voice commands: Attacks and defenses on the VCS of autonomous driving cars," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 128–133, Oct. 2019.

[162] L. Deshotels, "Inaudible sound as a covert channel in mobile devices," in *Proc. 8th USENIX Workshop Offensive Technol.*, 2014, pp. 1–4.

[163] D. Kumar *et al.*, "Skill squatting attacks on Amazon Alexa," in *Proc. 27th USENIX Security Symp.*, 2018, pp. 33–47.

[164] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "dolphinAttack: Inaudible voice commands," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 103–117.

[165] R. Zhang, X. Chen, S. Wen, and J. Zheng, "Who activated my voice assistant? A stealthy attack on android phones without users' awareness," in *Proc. Int. Conf. Mach. Learn. Cyber Security*, 2019, pp. 378–396.

[166] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful GPS spoofing attacks," in *Proc. 18th ACM Conf. Comput. Commun. Security*, 2011, pp. 75–86.

[167] J. Coffed, *The Threat of GPS Jamming: The Risk to an Information Utility*, EXELIS, Herndon, VA, USA, 2014.

[168] T. Giannetsos and T. Dimitriou, "Spy-Sense: Spyware tool for executing stealthy exploits against sensor networks," in *Proc. 2nd ACM Workshop Hot Topics Wireless Netw. Security Privacy*, 2013, pp. 7–12.

[169] Y. Tu, Z. Lin, I. Lee, and X. Hei, "Injected and delivered: Fabricating implicit control over actuation systems by spoofing inertial sensors," in *Proc. 27th USENIX Security Symp.*, 2018, pp. 1545–1562.

[170] Y. Park, Y. Son, H. Shin, D. Kim, and Y. Kim, "This Ain't your dose: Sensor spoofing attack on medical infusion pump," in *Proc. 10th {USENIX} Workshop Offensive Technol.*, 2016, pp. 1–6.

[171] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications," in *Proc. Int. Conf. Cryptograph. Hardw. Embedded Syst.*, 2017, pp. 445–467.

[172] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," in *Proc. Black Hat Europe*, vol. 11, 2015, p. 2015.

[173] X. Yuan *et al.*, "All your alexa are belong to us: A remote voice control attack against echo," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2018, pp. 1–6.

[174] R. Zhang, X. Chen, J. Lu, S. Wen, S. Nepal, and Y. Xiang, "Using Ai to hack IA: A new stealthy spyware against voice assistance functions in smart phones," 2018. [Online]. Available: arXiv:1805.06187.

[175] T. Sugawara, B. Cyr, S. Rampazzi, D. Genkin, and K. Fu, "Light commands: Laser-based audio injection attacks on voice-controllable systems," in *Proc. 29th USENIX Security Symp. (USENIX Security)*, 2020, pp. 2631–2648.

[176] J. Mao, S. Zhu, and J. Liu, "An inaudible voice attack to context-based device authentication in smart IoT systems," *J. Syst. Archit.*, vol. 104, 2020, Art. no. 101696.

[177] G. J. Persial, M. Prabhu, and R. Shanmugalakshmi, "Side channel attack-survey," *Int. J. Adv. Sci. Res. Rev.*, vol. 1, no. 4, pp. 54–57, 2011.

[178] M. A. Hakim, H. Aksu, A. S. Uluagac, and K. Akkaya, "U-Pot: A honeypot framework for UPNP-based IoT devices," in *Proc. IEEE 37th Int. Perform. Comput. Commun. Conf. (IPCCC)*, 2018, pp. 1–8.

[179] M. Yoshikawa and Y. Nozaki, "Hierarchical power analysis attack for falsification detection cipher," in *Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf. (CCWC)*, 2017, pp. 1–6.

[180] G. Petracca, Y. Sun, A.-A. Reineh, P. McDaniel, J. Grossklags, and T. Jaeger, "EnTrust: Regulating sensor access by cooperating programs via delegation graphs," in *Proc. 28th USENIX Security Symp.*, 2019, pp. 567–584.

[181] A. K. Sikder, L. Babun, and A. S. Uluagac, "Aegis+: A context-aware platform-independent security framework for smart home systems," *Digit. Threats Res. Pract.*, vol. 2, no. 1, pp. 1–33, 2021.

[182] Y. Zafar and D. Har, "A novel countermeasure enhancing side channel immunity in FPGAs," in *Proc. IEEE Int. Conf. Adv. Electron. Micro Electron.*, 2008, pp. 132–137.

[183] F.-X. Standaert, F. Macé, E. Peeters, and J.-J. Quisquater, "Updates on the security of FPGAs against power analysis attacks," in *Proc. Int. Workshop Appl. Reconfig. Comput.*, 2006, pp. 335–346.

[184] L. Babun, A. K. Sikder, A. Acar, and A. S. Uluagac, "IoTDoTs: A digital forensics framework for smart environments," 2018. [Online]. Available: arXiv:1809.00745.

[185] S. Mirzamohammadi, J. A. Chen, A. A. Sani, S. Mehrotra, and G. Tsudik, "Ditio: Trustworthy auditing of sensor activities in mobile & IoT devices," in *Proc. 15th ACM Conf. Embedded Netw. Sensor Syst.*, 2017, pp. 1–14.

[186] E. Fernandes, J. Paupore, A. Rahmati, D. Simionato, M. Conti, and A. Prakash, "FlowFence: Practical data protection for emerging iot application frameworks," in *Proc. USENIX Security Symp.*, 2016, pp. 531–548.

[187] K. Onarlioglu, W. Robertson, and E. Kirda, "Overhaul: Input-driven access control for better privacy on traditional operating systems," in *Proc. 46th Annu. IEEE/IFIP Int. Conf. Depend. Syst. Netw. (DSN)*, 2016, pp. 443–454.

[188] F. Roesner, T. Kohno, A. Moshchuk, B. Parno, H. J. Wang, and C. Cowan, "User-driven access control: Rethinking permission granting in modern operating systems," in *Proc. IEEE Symp. Security Privacy (SP)*, 2012, pp. 224–238.

[189] T. Ringer, D. Grossman, and F. Roesner, "AUDACIOUS: User-driven access control with unmodified operating systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2016, pp. 204–216.

[190] A. K. Sikder, H. Aksu, and A. S. Uluagac, "Context-aware intrusion detection method for smart devices with sensors," U.S. Patent 10 417 413, Sep. 17, 2019.

[191] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, "Key management systems for sensor networks in the context of the Internet of Things," *Comput. Elect. Eng.*, vol. 37, no. 2, pp. 147–159, 2011.

[192] M. Sun, M. Zheng, J. Lui, and X. Jiang, "Design and implementation of an Android host-based intrusion prevention system," in *Proc. ACM 30th Annu. Comput. Security Appl. Conf.*, 2014, pp. 226–235.

[193] W.-C. Wu and S.-H. Hung, "DroidDolphin: A dynamic android malware detection framework using big data and machine learning," in *Proc. Conf. Res. Adapt. Convergent Syst.*, 2014, pp. 247–252.

[194] G. G. Sundarkumar, V. Ravi, I. Nwogu, and V. Govindaraju, "Malware detection via api calls, topic models and machine learning," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, 2015, pp. 1212–1217.

[195] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Comput. Netw.*, vol. 76, pp. 146–164, Jan. 2015.

[196] M. W. Live. (Mar. 2016). *Analysis: Mobile World Congress 2016 Wrap-Up*. [Online]. Available: http://www.mobileworldlive.com/mwc16-articles/analysis-mwc16-wrapup/

[197] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial Internet of Things," in *Proc. ACM 52nd Annu. Design Autom. Conf.*, 2015, p. 54.

[198] A. P. Felt *et al.*, "How to ask for permission," *HotSec*, vol. 12, 2012, p. 7.

[199] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the Internet of Things," in *Proc. IEEE World Congr. Services (SERVICES)*, 2015, pp. 21–28.

[200] R. H. Weber, "Internet of Things—New security and privacy challenges," *Comput. Law Security Rev.*, vol. 26, no. 1, pp. 23–30, 2010.

[201] L. Babun, Z. B. Celik, P. McDaniel, and A. S. Uluagac, "Real-time analysis of privacy-(un) aware IoT applications," in *Proc. Privacy Enhanc. Technol.*, 2021, pp. 145–166.

[202] N. Farhi, N. Nissim, and Y. Elovici, "Malboard: A novel user keystroke impersonation attack and trusted detection framework based on side-channel analysis," *Comput. Security*, vol. 85, pp. 240–269, Aug. 2019.

[203] A. Acar, H. Aksu, A. S. Uluagac, and K. Akkaya, "A usable and robust continuous authentication framework using wearables," *IEEE Trans. Mobile Comput.*, early access, Feb. 18, 2021, doi: 10.1109/TMC.2020.2974941.

[204] W. Aman, M. M. U. Rahman, J. Qadir, H. Pervaiz, and Q. Ni, "Impersonation detection in line-of-sight underwater acoustic sensor networks," *IEEE Access*, vol. 6, pp. 44459–44472, 2018.

[205] T.-H. Lin, C.-C. Lee, and C.-H. Chang, "Wsn integrated authentication schemes based on Internet of Things," *J. Internet Technol.*, vol. 19, no. 4, pp. 1043–1053, 2018.

[206] C. Hartung, J. Balasalle, and R. Han, "Node compromise in sensor networks: The need for secure systems," Dept. Comput. Sci., Univ. Colorado at Boulder, Boulder, CO, USA, 2005.

[207] A. Aseeri and R. Zhang, "Secure data aggregation in wireless sensor networks: Enumeration attack and countermeasure," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–7.

[208] L. Babun, H. Aksu, and A. S. Uluagac, "A system-level behavioral detection framework for compromised CPS devices: Smart-grid case," *ACM Trans. Cyber Phys. Syst.*, vol. 4, no. 2, pp. 1–28, 2019.

[209] A. I. Newaz, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, "HealthGuard: A machine learning-based security framework for smart healthcare systems," in *Proc. IEEE 6th Int. Conf. Soc. Netw. Anal. Manag. Security*, 2019, pp. 389–396.

[210] C. Namiluko, A. J. Paverd, and T. De Souza, "Towards enhancing Web application security using trusted execution," in *Proc. WASH*, 2013.

[211] A. Newaz, N. I. Haque, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, "Adversarial attacks to machine learning-based smart healthcare systems," 2020. [Online]. Available: arXiv:2010.03671.

[212] *Security for Android Developers*. Accessed: Oct. 23, 2018. [Online]. Available: https://developer.android.com/topic/security/

Amit Kumar Sikder (Member, IEEE) received the bachelor's degree in electrical and electronic engineering from the Bangladesh University of Engineering and Technology. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Florida International University, where he is also a Research Assistant and as a Member of the Cyber-Physical Systems Security Lab. His research interests are focused on the security of cyber-physical systems and Internet of Things. He also has worked in areas related to security of smart devices, security of smart home, smart city, and wireless communication. More information can be obtained from: http://web.eng.fiu.edu/asikd003/.

Giuseppe Petracca received the B.S. degree and the M.S. degree in computer science and engineering from the Sapienza University of Rome, Italy. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Pennsylvania State University, where he is also a Research Assistant. He also collaborates for the Cyber Security Collaborative Research Alliance, sponsored by the Army Research Laboratory. His industry experience includes a summer internship as a Graduate Researcher with Intel in 2013, a Graduate Technical Engineer with Intel Labs in 2014, and summer internship as a Software Engineer and a Security Researcher with Samsung Research America in 2016, and Google in 2017. His research interest focuses on mobile systems and cloud computing security. More information can be obtained from: http://sites.psu.edu/petracca/.

Hidayet Aksu received the B.S., M.S., and Ph.D. degrees from the Department of Computer Engineering, Bilkent University in 2005, 2008, and 2014, respectively. He is currently a Postdoctoral Associate with the Department of Electrical and Computer Engineering, Florida International University. Before that, he worked as an Adjunct Faculty with the Computer Engineering Department, Bilkent University. He conducted research as a Visiting Scholar with IBM T. J. Watson Research Center, USA, from 2012 to 2013. He also worked for Scientific and Technological Research Council of Turkey (TUBITAK). His research interests include security for cyber-physical systems, Internet of Things, security for critical infrastructure networks, IoT security, security analytics, social networks, big data analytics, distributed computing, wireless networks, wireless ad hoc and sensor networks, localization, and p2p networks.

Trent Jaeger received the M.S. and Ph.D. degrees in computer science and engineering from the University of Michigan, Ann Arbor, in 1993 and 1997, respectively, and spent nine years at IBM Research prior to joining Penn State. He is a Professor with the Computer Science and Engineering Department, Pennsylvania State University, where he is the Co-Director of the Systems and Internet Infrastructure Security Laboratory. He has published over 100 refereed papers on these topics and the book *Operating Systems Security*, which examines the principles behind secure operating systems designs. He has made a variety of contributions to open source systems security, particularly to the Linux Security Modules framework, SELinux, integrity measurement in Linux, and the Xen security architecture. His research interests include systems security and the application of programming language techniques to improve security. He was the Chair of the ACM Special Interest Group on Security, Audit, and Control. More information can be obtained from: http://www.cse.psu.edu/˜trj1/.

A. Selcuk Uluagac received the M.S. and Ph.D. degrees from the Georgia Institute of Technology, and M.S. degree from Carnegie Mellon University. He is currently a Member of the faculty with the Department of Electrical and Computer Engineering, Florida International University as an Eminent Scholar Chaired Associate Professor, where he directs the Cyber-Physical Systems Security Lab, focusing on security and privacy of Internet of Things and Cyber-Physical Systems. Before joining Florida International University, he was a Senior Research Engineer with the School of Electrical and Computer Engineering, Georgia Institute of Technology and a Senior Research Engineer with Symantec. He received the U.S. National Science Foundation CAREER Award and the U.S. Air Force Office of Sponsored Research's Summer Faculty Fellowship in 2015. He has served on the program committees of top-tier security conferences such as IEEE S&P, NDSS, ASIACCS, inter alia. He was the General Chair of ACM Conference on Security and Privacy in Wireless and Mobile Networks in 2019. He serves on the editorial boards of the IEEE TRANSACTIONS ON MOBILE COMPUTING, *Computer Networks* (Elsevier), and the IEEE COMMUNICATIONS AND SURVEYS AND TUTORIALS. More information can be obtained from: http://nweb.eng.fiu.edu/selcuk.