Analysis, Implications, and Challenges of an Evolving Consumer IoT Security Landscape

Christopher Bellman and Paul C. van Oorschot
School of Computer Science, Carleton University, Ottawa, Canada

***Abstract*—The Internet of Things (IoT) is a rapidly growing subset of our modern computing architecture, and as such, provides significant new attack surface. The history of IoT has provided a substantial body of topics to look back on: IoT's evolution, products, and major security incidents including the largest botnet ever witnessed. Unique to IoT, its architecture, interaction design, and scale make its many issues distinct from those in the Internet of Computers (IoC). Its perceptions, understandings, and definitions have evolved over time, thus requiring an updated focus from the perspective of security and cyberphysical safety. We take a fresh look at challenges and opportunities in IoT security, the characteristics that uniquely distinguish it from the IoC, and identify security-related questions that they raise. Our aim is to provide an up-to-date view of the IoT security landscape and technical security issues to help guide both existing and especially new researchers looking for challenging open problems that remain largely unaddressed.**

***Index Terms*—Internet of Things (IoT), IoT security, cyber-physical security, Internet security**

I. Introduction

The Internet of Things (IoT) is commonly described as adding network communication capabilities to everyday objects. The IoT space ranges from such low-powered devices as simple temperature sensors, to embedded devices in critical infrastructure controls for power or water systems. IoT capabilities may involve the addition of microcontrollers (integrated circuits: with processor, integrated memory, and programmable input/output peripherals; program stored in ROM) or microprocessors (CPU with separate chips for memory, peripherals). Their numbers are expected to exceed 50-billion by 2025 [1]. Given the scale of the Mirai botnet attack [2], the mainstream realization of potential IoT-related damage has made researchers aware of the threats that IoT devices pose.

IoT devices differ from traditional computing devices such as desktop computers, servers, or smart phones—devices that belong to what we know as the Internet of Computers (IoC). Unlike IoC, which can only indirectly affect the physical world, IoT has implications for not only computer security, but also safety. Fig. 1 highlights a partial taxonomy and overview of major categories of IoT. Our work herein has primary focus on consumer-grade IoT devices, which are commonly recognized as being very poorly secured [3] [4].

IoT—in both its definitions and our perceptions of it—has evolved considerably over the past decade. As such, there is need for a revised understanding of the landscape of IoT security. Past surveys—a number from the pre-Mirai era (e.g., [5]–[7])—lack technical details as work done in more specific areas of IoT security was yet to be carried out in depth. Technical progress over the past five years allows a new understanding of the challenges and opportunities in IoT security. We discuss considerations including IoT network architecture, security protocols, cryptographic implementations, and networking protocols, and their influence on IoT security. We take a fresh look at the characteristics that uniquely distinguish IoT from IoC, and analyze the implications of each for security. We tie these implications to current security challenges, and those that are expected to appear as IoT rapidly grows and permeates our environments.

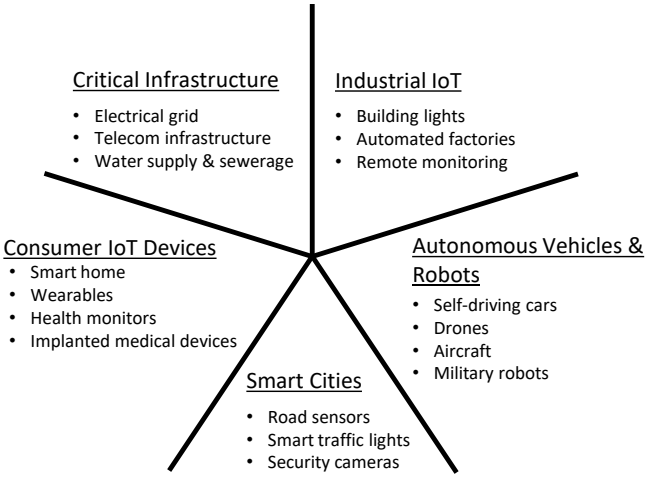

Fig. 1. Partial IoT taxonomy and examples of sub-areas.

II. Generic Architecture of Consumer-Grade IoT

The architecture of IoT is defined by the mechanisms and physical structure by which each device in the network relates to others. Fig. 2 depicts a simplified view of the network architecture for a smart home. *IoT services* include interoperability and trigger-action programming functions (e.g., IFTTT [8]), and management platforms (e.g., Amazon AWS IoT Core).

Low-end devices make use of lightweight communication protocols and standards. IoT-friendly (lower resource consumption) upper-layer protocols are commonly used for communicating with other devices or services [9]. Hub devices (e.g., Phillips Hue Bridge, Samsung SmartThings Hub) are used to manage local devices and bridge communications

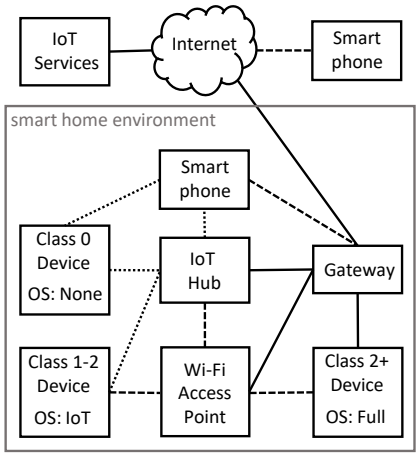

Fig. 2. Generic architecture of smart home IoT deployment. Solid lines denote wired connections, thick dashed lines represent Wi-Fi, thin dashed lines represent low-power wireless, e.g., Zigbee, Bluetooth Low-Energy. "IoT" OSs are specifically for IoT devices (Section III-A). "Full" OSs are, e.g., Linux.

between them and other hosts; these are typically less resource constrained. Devices that require an intermediate node for communication connect to a hub via low-power wireless to have messages forwarded. Higher-end devices connect directly to a gateway or Wi-Fi access point. From the gateway, traffic can be routed as normal on the Internet. Alternatively, smart devices such as wearables connect directly to a smart phone via low-energy wireless and/or to the Internet via cellular signal or Wi-Fi. Remote devices can be used to access IoT cloud services and smart home devices.

III. Distinguishing Characteristics of IoT

IoT has characteristics that distinguish it from IoC. Here we discuss select characteristics and their security implications (Table I).

In IoT, an individual user may have a variety of devices associated with them in addition to standard IoC devices (e.g., laptop, smart phone, desktop computer). Combined with the devices that are associated with environments, the scale of IoT is expected to dwarf the IoC [1]. This scale impacts essentially all characteristics as the size commonly exacerbates existing issues.

A. Low-Cost

An IoT device might simply be a standard device with a small built-in computer component. When an IoT device is referred to as "low-cost", we often mean its IoT component. Manufacturers typically minimize the cost of an IoT component, favouring market presence over security [10].

Device resource constraints are typical consequences of low costs. Some resource constraints in IoT are input/output (e.g., screen, keypad), memory sizes, processor speeds, and battery size. RFC 7228 [11] defines three classes of resource constrained devices (Table II). Class 0 devices are generally too constrained to communicate directly with hosts on the Internet securely, relying on an intermediate node to communicate via low-power protocol such as Bluetooth Low-Energy (BLE), Zigbee, or 6LoWPAN. They typically use single-purpose specialized microcontrollers [12]. Class 1 devices commonly struggle to communicate over the Internet using more standard upper-layer communication protocols (e.g., HTTP, TLS), instead using lighter-weight protocols through intermediate nodes. Class 2 devices still leverage protocols and features designed for resource constrained devices, but may (depending on hardware and software) be capable of running standard protocols to communicate on the Internet. Resource constrained devices above Class 2 exist (Class "2+" in Fig. II) but are not discussed herein.

TABLE I
IoT characteristics and their implications for security.

1. Low-Cost (Section III-A)
   - Constrained resources
   - Smaller/no OS
   - Need for more efficient protocols
   - Need for lightweight crypto
   - Over-provisioned functionality (cost-friendly component re-use)
   - Manufacturer security inexperience (IoT sub-component)
2. Non-Standard Interfaces (Section III-B)
   - New attack surfaces
   - Greater physical access to devices
   - Complicates device management, configuration, updates; exacerbated by scale
3. Cyberphysical Interaction (Section III-C)
   - Successful network attack may affect physical world
   - Implied trust in manufacturer
4. Expectation of Long-Lived Devices (Section III-D)
   - Lack of software updates may leave vulnerabilities unpatched
   - Forgotten devices remain attractive targets
   - Device outliving manufacturer impacts software updates
   - Cryptographic algorithms and protocols must be future-proofed
5. "Many-User" Devices with Unclear Authority (III-E)
   - Home guests may be denied functionality of critical services
   - Rogue guests may retain remote access
   - Difficult to differentiate authorized and unauthorized users

1) Implications for Security: Seeking cost reductions, manufacturers may use open-source software or generic hardware to build their devices on, choosing solutions that provide the required functionality for their product. Use of over-provisioned components adds unnecessary risks—complexity is the enemy of security. For example, unused modules and features, often not properly disabled, provide additional attack surface. A common example is using Linux for the OS of a device (and not disabling functions or services that are unused). This consequence is related to manufacturer inexperience as new manufacturers who do not fully understand their technical or functional needs may choose generic, potentially over-provisioned solutions.

Class 0 devices are highly resource constrained—they are typically specialized microcontrollers that have very static and specific functions [12]. Those above Class 2 with much higher operating specifications may be capable of running a full operating system like Linux. Devices in between (classes 1

TABLE II
RESOURCE-CONSTRAINED DEVICE CLASSES: MEMORY LIMITATIONS [11], OPERATING SYSTEMS (IF ANY), AND COMMUNICATION METHODS [12].

Class	Volatile memory	Non-volatile memory	OS & Communication
0	$<<$10 KiB	$<<$100 KiB	Function-specific hardware, few IoT OSs. Basic health indicator and keep-alive messages, requires intermediate node.
1	$\sim$10 KiB	$\sim$100 KiB	IoT-specific OS. Lightweight wireless (e.g., BLE)/wired, UDP-based protocols.
2	$\sim$50 KiB	$\sim$250 KiB	IoT-specific OS. Lightweight wireless/wired, UDP-based protocols, commonly-used upper-layer protocols.
2+	$>$50 KiB	$>$250 KiB	IoT-specific, or full OS. Commonly-used upper-layer protocols.

through 2) can make use of a variety of open- and closed-source OSs specifically for resource constrained devices (e.g., *Contiki*, *TinyOS*, and *FreeRTOS* [12]).

A wide variety of low-power protocols (e.g., BLE, 6LoWPAN, Zigbee) are typically used to communicate with other physically-near devices. Depending on the hardware, less resource constrained devices are capable of running Wi-Fi and common upper-layer protocols for Internet communication. IETF work currently underway, to support resource constrained devices, is developing new or adapted suites of protocols designed for IoT (e.g., CoAP, MQTT [9]).

Both communication and cryptography functions require processing and memory, so lightweight cryptographic algorithms and wireless protocols need to be used, especially for Class 0 devices. Devices must be able to run common cryptographic algorithms at acceptable speeds to meet secure communication requirements. Generating and storing sufficiently-long asymmetric cryptographic keys (e.g., $\geq$2048 bits for RSA [13]) in IoT is more challenging than in IoC. Further, best practices for key sizes will grow over time (e.g., 3072 bits recommended for RSA by 2031 [14]), which is increasingly problematic for IoT. This is the major motivation for adoption of elliptic curve (EC) cryptographic algorithms in IoT environments.

Now that IoT has become attractive for manufacturers, the "IoT" label on a device may be used as a feature to invest in; however, the addition of IoT functionality is often not accompanied by security expertise (for IoT sub-components). Manufacturer inexperience amplifies safety and ubiquity issues as any weaknesses with a device adds to the potential impact of attacks and problems related to non-standard interfaces.

2) New Problems/What is Different: Class 1 and 2 devices are capable of running lightweight OSs designed specifically for IoT. Common requirements include low memory usage ($\approx$10–50 KiB volatile, $\approx$100–250 KiB non-volatile memory [11]), diverse hardware support (8–32-bit microcontroller architectures, varying amount of on-board RAM), communications (wired or wireless), low power usage (operate for months without battery replacement), low-delay processing (commonly requires real-time responses), and built-in security mechanisms (crypto/security protocols, access control) [12]. A common characteristic of IoT OSs is their development in the C language [12]. C and C++ have historically been the choice for IoC operating systems and tools; however, they bring with them a number of vulnerabilities such as memory safety errors (e.g., buffer overflows), integer-based vulnerabilities, and race conditions. Many of the lessons learned from IoC OS design will need to be remembered lest all the same problems reappear in IoT.

For wide-spread adoption of secure communication, toolkits will need to support both expert and non-expert developers. Many such toolkits exist for IoC and higher-end IoT devices (e.g., OpenSSL, NSS, wolfCrypt—commonly written in C). Libraries such as *micro-ecc*, *TinyECC*, and *μNaCl* bring limited crypto functions to heavily resource constrained 8-bit microcontrollers. Some IoT development boards make use of dedicated hardware-based processors (e.g., Microchip ATECC608A) for cryptographic algorithms and key storage. Elliptic-curve cryptography appears as a candidate to replace RSA for asymmetric-key operations due to faster computation and smaller key sizes [15] (e.g., A 224-bit ECDSA key has comparable strength to a 2048-bit RSA key [14]).

Prevention and/or mitigation of malicious action is required to address device compromise. The recently-proposed Manufacturer Usage Descriptions (MUDs) are manufacturer-provided descriptions of how their devices are designed to behave [16]. Obtaining this directly from the manufacturer should enable easier misbehaviour detection and, if implemented correctly, will make anomalous activity far more easily detected. In the absence of MUDs, automatic generation of communication policies at the network level may be done [17] [18].

B. Non-Standard Interfaces

Device interfaces vary wildly between IoC and IoT. For usability, the challenge is often greater in the configuration of a device rather than in its standard function. Interaction design is about what ways a user interacts with a device. In IoC, this is done almost exclusively using a keyboard and monitor, or combined touch screen. IoT devices commonly require some alternative method for device setup or configuration (e.g., smart phone app, cloud management service). This leads to a number of challenges for users to manage device updates, configuration, and decommissioning.

IoT is still fairly new and device diversity is high. Diversity is amplified by the wide range of what we define as an IoT device and makes it difficult to standardize hardware. This exacerbates problems such as secure device configuration or communication between devices. Coupled with hardware differences, at very low ends, software running on devices is specialized for a specific task making it difficult to produce software for, update, and manage a wide variety of devices.

This is ameliorated in devices with IoT-focused OSs where common code bases can be utilized [12].

1) Implications for Security: New interaction designs mean new attack surfaces. Voice commands have proven useful in smart home devices, taking any sound in the environment as a potential command. Sensor inputs (temperature, noise) can be abused to provide falsified data (e.g., manually altering sensor readings). Cloud services present a new attack surface, although the core cloud services take on IoC challenges. The scale of IoT plays a significant role. The more things there are, the greater the potential attack surface and hosts in a botnet. Finally, physical access to IoT devices is an additional attack surface; it is likely easier for a guest or intruder to steal a small IoT device than a laptop or desktop computer due to their placement and ubiquity. Once stolen, the attacker could attack the home using this device or recover sensitive data from its storage.

Non-standard interfaces make device management more difficult. These problems are exacerbated by the scale of devices to be maintained—the more devices, the more of an impact any interface inefficiencies have. It is one thing to configure a small handful of devices, but scaling the numbers up may cause users to become frustrated and ignore configuration. If a device functions correctly (from the user's perspective) without a secure configuration, users may choose to avoid configuration entirely for the rest of their devices. Enforcing device configuration before allowing it to function would solve this issue [3], but it would impact usability and frustrate users.

2) New Problems/What is Different: For each new input mode, new methods may be needed to protect them or mitigate attacks conducted via their use. For voice inputs, there is the potential for unauthorized users to send inputs to a device via voice commands. For example, "Hidden voice commands" that can trigger functions on voice-activated devices without being understandable voice commands to the human ear [19], leading to audible attacks that users are unable to identify as malicious. Sensors will need to determine if a reading has been falsified or manipulated. Cross-checking of device readings may be viable in some cases (e.g., comparing a thermometer's reading to a reading from across the room) to determine consistency, but requires communication either between (possibly heterogeneous) sensors or with a central hub device. Remote management requires that each communication step between a device and its remote access point (Fig. 2) be secure. IoT's resource constraints require new solutions for secure constrained end-device communication. IoT cloud services are a more usable approach to interacting with devices compared to individually connecting to each one, but it means the security of such a platform is out of the control of the users and they must trust the service provider to maintain security of the service infrastructure and communications.

As smaller, more pervasive devices are easier to be physically accessed or stolen, each individual device will be required to protect itself from physical or digital attacks. One means is by minimizing the sensitive data it stores (such as user data or non-critical communication information), under the assumption that it will be stolen and attacked offline. Devices should be provided only with the minimum amount of information required for them to successfully function.

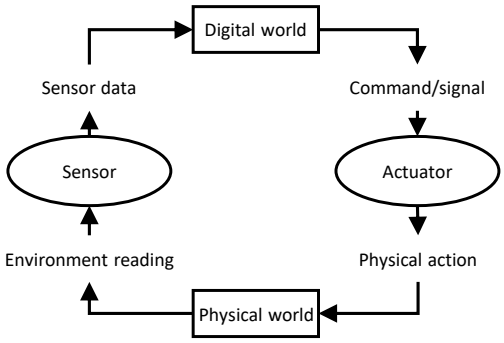

Fig. 3. Relationship between sensor devices and actuator devices, and their role in IoT space.

IoT devices—particularly consumer devices—should be simple to setup and maintain for the average user. Given a large number of devices, users might deploy their devices with a default configuration. This makes designs involving safe default settings critically important.

C. Cyberphysical Interaction

Over time, the terms "cyberphysical system" and "IoT" have merged, and are now commonly used interchangeably [20]. Both share the characteristic of linking the physical environment to the digital space, so we refer to a cyberphysical device as a device that interacts with its environment. Fig. 3 depicts two common classes of cyberphysical IoT components—sensors and actuators—and their relationship to one another and the world. Sensors convert environmental readings into data, and actuators take commands and provide output in the form of a physical manipulation of an environment.

1) Implications for Security: Since low-level IoT devices are primarily comprised of sensors and actuators, interacting with them is an interaction with an environment. In IoC, attacking a system means that you are interacting with data, but in IoT there is now an impact on the physical world. The new risk this introduces is that standard network attacks can now affect physical environments through actuation.

With devices surrounding us, it is possible to link data from various sources. Recorded data from our environment can provide entities with great knowledge about us and our environments if they have access to this data. Further, they could link various sources to generate new data about a user or environment. Purchasing and using a product demands an implicit trust in the manufacturer and service providers, as the data may be stored and/or processed by them.

2) New Problems/What is Different: In IoT, affecting an actuator device alters the physical environment, thus device security for actuators needs to be a priority and threat models need to consider this [21]. With IoT, we have the ability to better contextualize access granted to devices or functions as these actions are more clearly tied to real-world effects [22].

The potential for data linking remains a problem. Currently, environmental monitoring across a variety of devices is difficult due to the heterogeneity inherent in IoT [23]; however, with greater standardization and interoperability between devices and ecosystems, tracking data on a greater scale will be easier. Even if the data itself is unusable (encrypted, anonymized), metadata could be linked to glean personal information about the user or their environment [24] [25]. However, data linking may also provide a defense against misbehaviour [23]. Multiple devices can use contextual data or metadata provided by its surrounding devices to sanity-check other input data.

D. Expectation of Long-Lived Devices

Consumers expect their IoT devices to operate for a long period. This requires that devices remain functional and secure. For many types of device, interaction is designed to be kept at a minimum, as a "set-and-forget" device. Once these devices are set, users expect them to continue functioning without frequent maintenance. Devices such as smart light bulbs should function properly as light bulbs, but not require the user to check for updates or perform maintenance periodically.

1) Implications for Security: The need for device software updates is a significant threat to IoT devices as a lack of updates means potentially unpatched vulnerabilities. It may be a boon to usability if manufacturers can take responsibility for the update process, however users may need to be made aware of the existence of vulnerabilities and updates, should automatic, manufacturer-controlled updates be unavailable. Further complicating the issue of update strategy, a device could outlive its manufacturer—there will not necessarily always be software updates for a product's lifetime. Will third-party developers be able to fill the gap as is common in the IoC? With (relatively) few operating systems in IoC (Windows, Mac OS, Linux), software development toolkits are more commonly accessible. In IoT where low-end devices make use of specialized hardware and commonly run no OSs at all, building third-party software is more difficult (no accessible toolkits/developer APIs) and may be unrewarding.

Devices may continue to be used, but not always for their IoT functionality. For example, smart light bulbs may be set once to a standard white colour and high brightness, then never changed again—only turned on and off like a normal light bulb. In cases where devices remain network connected but no longer being used (for their IoT functionality), it is likely their availability remains for attackers to probe or make use of (if exploited). This network connectivity, lack of monitoring, and potentially little or no update strategy from the manufacturer leads to devices remaining vulnerable for significant periods of time.

Another potential issue is quantum computing and its impact on currently infeasible calculation problems. A risk is that, once quantum computing is more advanced, currently-popular public-key cryptography algorithms such as RSA and ECC will be easily defeated [26]. Quantum-proof crypto is applicable to IoC as well. The issue remains a serious problem for IoT devices that are constrained and not easily updated.

2) New Problems/What is Different: Updates must be received securely and verified as legitimate. Out-of-date devices will first need to be identified, so a method of device identification will be required to point users towards devices requiring updates and to enforce security policies [18]. To provide trustworthy updates, reliance on signing and verification operations requires at least minimal trust infrastructure. In the case of a bad update (whether by accidental corruption or a malicious image), a device may operate with vulnerabilities or attacker access, or may not function at all. The ability to roll back an update may help; however, this function might be unknown to users or impossible if the update was malicious or crippling for the device. Given the set-and-forget nature of IoT, updates presented to the user (if any) may well be ignored or avoided. Push-based updates may be more appealing in the scale of IoT (as opposed to users manually pulling updates). This is, however, less easily accomplished than in IoC where near constant TCP/IP Internet connections are expected.

What happens if a vulnerability is found but a user can not patch it (unavailability or no knowledge of patch existence)? If unable to provide updates, the responsibility could be transferred from the manufacturer to another entity to formally provide the updates. Regardless of the OS a device is using or the status of a manufacturer, a software update policy set by the manufacturer could be used to enforce how and when updates are applied, and who takes responsibility for providing these updates [10]. While many IoT-specific OSs are in development and deployment (Section III-A), over time their numbers may thin, making it less costly to develop across OSs. This is highly related to the lifespan of a device as short-lived devices would be seen as less important to spend resources developing software for. Regardless, a solution for long-lived vulnerable devices is needed, whether it is a solution for updates or deactivation of devices past their manufacturer-supported lifetimes. Inactive and vulnerable devices must have their access revoked in order to protect the integrity of the network.

E. "Many-User" Devices with Unclear Authority

In IoC, systems are commonly labeled as "multi-user" or "single-user" based on their architecture and usage. In these systems a user is identified (username, remote device ID, user ID) and they interact with a system within the context of being an identified user. In IoT, devices commonly belong to an environment and are generally not linked to an individual user or group of users, thus being a "many-user" device (e.g., sensors, voice assistants, lights). While this is not the case for all IoT devices, it is a characteristic that is common.

1) Implications for Security: Device-to-device access control is a matter of configuration by the owner or trusted user, which could be (and commonly is) solved by standard role-based or discretionary access control [27]. IoT devices may now belong to a single environment rather than a user. This makes it difficult for systems to differentiate between

authorized and unauthorized users. Modern smart home hubs provide varying approaches to access control such as differing levels of control (privileged vs. unprivileged user), guest accounts, and time-/location-based policies [27]; however, these are for hub devices—individual "things" in an environment require different approaches as standard role-based or discretionary access controls are not applicable when an individual user can not be uniquely identified.

A home owner who configures their house with smart devices has full access to the devices they own. The owner of a home using a rental service (e.g., Airbnb, VRBO) requires administrator access to critical devices (e.g., lighting, heating) but guests should be granted basic user access to at least operate the devices [27]. In IoT where it is more difficult to tell which individual users are using devices, managing access remains a problem. Different hubs and more advanced IoT devices may have interfaces that can make use of user accounts for access control [27]. For devices incapable of this or ones that are used frequently, new solutions for access control are required as it becomes tedious to authenticate frequently. To highlight the access control issue, rogue guests could continue to access devices after leaving and returning to an environment and, for example, unlock the front door of a rental unit if their credentials were not cleared.

2) New Problems/What is Different: A single solution for access control that covers all permissions and devices within a system is a challenge. Some ecosystems' hub devices may allow for access control of configuration activities, but providing a user with access to use a light bulb and thermostat of two different brands becomes more difficult. Further, access control may be done differently in different devices or ecosystems, making access control more open to misconfiguration in environments with great device heterogeneity.

Facilitating access control methods for many-user devices is a challenge. Individual user access control could be done if each user was forced to authenticate each command or interaction (e.g., using a smart phone to confirm each interaction), but this becomes tedious and against the spirit of easy IoT interaction. With explicit users and interfaces as is the common case in IoC, access control is a fairly well-understood (albeit non-trivial) task. In a many-user environment the challenge of identifying individual authorized users remains. A potential solution could be considering the device itself as an untrusted entry point into the network, thus having a hub device set access control policies to the device's ability to communicate through it. This solution, however, makes management tedious as a change in users of an environment may require a modification of access control policies for a device.

Once a user is provided with access, it may need be revoked. Not all devices provide adequate access control mechanisms to separate privileged and unprivileged actions [3]. Administrators should have the ability to configure revocation strategies; however, some of these methods require the identification of individual users. Alternatively, solutions such as time-limited access tokens provide automatic revocation, but require careful configuration of user access windows to balance user access and exposure window (time after legitimate token use but before token revocation). For devices that are accessed via smart phone, this is not a problem as identification becomes linked to the user's smart phone.

IV. Related Work

As with IoT itself, literature surrounding IoT is growing. Here we highlight selected security-focused surveys of IoT security. A number of general IoT security surveys consider current challenges and opportunities [5]–[7], [28]. Fernandes et al. [29] provides a high-level overview of the IoT space for consumer-grade, industrial, and vehicular IoT within the context of determining what challenges can be solved using known IoC techniques. Bertino and Islam [30] discuss the significance of the weak security of IoT devices and its relationship with botnets. Primarily citing the Mirai botnet as the most significant attack (as does Kolias et al. [2]), they outline reasons why IoT devices are so heavily targeted for botnets, and strategies for preventing IoT device infection. Angrishi [31] does similarly, exploring the structure of IoT botnets, their functions, and common vulnerabilities that lead to their exploitation. Greer et al. [20], Hahm et al. [12], and Smith [32] provide overviews of the IoT security space for non-technical audiences.

V. Discussion & Concluding Remarks

One of the common themes that appears in all characteristics and problems discussed herein is the scale of IoT (Section III). Scale exacerbates all security and usability challenges. Engineering properly secured devices is difficult enough in IoC for a home environment; IoT brings orders of magnitude more devices, complicating access control (fine-grained access control becomes infeasible with scale), secure configuration (of a larger number of devices), and device software updates (secure protocol design, acquisition, integrity, authorization, and installation).

It is generally acknowledged [3] [4] that many consumer-grade IoT devices have easily exploited security vulnerabilities, but what tools are available for IoT device manufacturers? Given IoT hardware capabilities, cryptographic toolkits require re-engineering to meet the constrained capabilities, and commonly relied upon algorithms in IoC need to be re-evaluated to meet performance challenges (e.g., choosing ECC over RSA [15]). Research over the past decade has explored lightweight cryptography in a number of areas including wireless sensor networks [15] and smart grid applications [33]. Findings in these areas can be applied in IoT.

IoT currently lacks critical software tools and protocols to develop secure systems (Section III-A1). As discussed, lightweight tools (crypto libraries, OSs) are under development to improve system security; however, these are as yet thinly deployed and highly platform specific. Manufacturers lacking expertise in security, looking to develop devices with IoT capabilities, release products before they are security-ready [4]. Who is then responsible for attacks based on these insecure devices? The (in)security track record to date supports

the argument for government regulation, as safety becomes an issue and safety regulation is a traditional government role. Standards are helpful, but are not necessarily enforced. California (USA) lawmakers have passed a bill requiring device manufacturers to take reasonable measures to protect devices, including unique default passwords or secure first-time authentication schemes [34]. Regulation such as California's may promote best practices elsewhere.

As research directions, there are a number of avenues to pursue. One is development of new security protocols, algorithms, and tools that support IoT constraints. Another is development and adoption of IoT security best practices. Design of OSs for resource-constrained devices is underway as discussed. Use of standardized APIs will aid developers. A critical aspect of the IoT is communication, and its impact on battery power. As discussed, low-power communication protocols are widely used for connecting devices; secure variations of these remain a challenge. To further support IoT-friendly lower-layer communication protocols, new upper-layer protocols will be required. Separate from these technical aspects, the usability of IoT devices plays a critical role in security. As noted, the ability of users to configure their devices without security errors and with reliable access control methods is of critical importance. IoT's lack of standard interfaces, in addition to scale, exacerbates configuration issues.

Further research challenges will no doubt arise as the IoT continues its growth and evolution. We expect the pace of technical change to increase, rather than subside. As such, we encourage security researchers and industry experts to give increased attention to the many practical research problems and opportunities opened up by the Internet of Things—a gift that keeps on giving, in perhaps too many ways.

Acknowledgments. We thank the members of the Carleton Computer Security Lab and anonymous referees for the feedback on this work. Van Oorschot is Canada Research Chair in Authentication and Computer Security, and acknowledges NSERC for funding the chair and a Discovery Grant.

REFERENCES

[1] G. F. Hurlburt, "The Internet of Things... of all things," *ACM Crossroads*, vol. 22, no. 2, pp. 22–26, 2015.

[2] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[3] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "SoK: security evaluation of home-based IoT deployments," in *IEEE Symp. Security and Privacy*, 2019.

[4] M. Antonakakis, T. April, M. Bailey, E. Bursztein, J. Cochran, Z. Durumeric, J. Alex Halderman, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai botnet," in *USENIX Security Symposium*, 2017.

[5] Z. K. Zhang, M. C. Y. Cho, C. W. Wang, C. W. Hsu, C. K. Chen, and S. Shieh, "IoT security: ongoing challenges and research opportunities," in *International Conference on Service-Oriented Computing and Applications*, 2014.

[6] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: the road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[7] J. A. Stankovic, "Research directions for the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.

[8] IFTTT. (2019) IFTTT. [Online]. Available: https://ifttt.com/

[9] C. Gündogan, P. Kietzmann, M. Lenders, H. Petersen, T. C. Schmidt, and M. Wählisch, "NDN, CoAP, and MQTT: a comparative measurement study in the IoT," in *ACM Conference on Information-Centric Networking (ICN)*, 2018.

[10] P. Morgner and Z. Benenson, "Exploring security economics in IoT standardization efforts," in *Workshop on Decentralized IoT Security and Standards (DISS)*, 2018.

[11] C. Bormann, M. Ersue, and A. Keranen, "Terminology for constrained-node networks," RFC Editor, RFC 7228, May 2014.

[12] O. Hahm, E. Baccelli, H. Petersen, and N. Tsiftes, "Operating systems for low-end devices in the Internet of Things: a survey," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 720–734, 2016.

[13] E. B. Barker and A. L. Roginsky, "Transitioning the use of cryptographic algorithms and key lengths," NIST Special Pub 800-131A, March 2019.

[14] E. B. Barker, "Recommendation for key management," NIST Special Pub 800-57 Part 1, January 2016.

[15] A. Liu and P. Ning, "TinyECC: a configurable library for elliptic curve cryptography in wireless sensor networks," in *International Conference on Information Processing in Sensor Networks*, 2008.

[16] A. Hamza, D. Ranathunga, H. H. Gharakheili, M. Roughan, and V. Sivaraman, "Clear as MUD: generating, validating and applying IoT behaviorial profiles," in *Workshop on IoT Security and Privacy*, 2018.

[17] D. Barrera, I. Molloy, and H. Huang, "Standardizing IoT network security policy enforcement," in *Workshop on Decentralized IoT Security and Standards (DISS)*, 2018.

[18] M. Miettinen, S. Marchal, I. Hafeez, T. Frassetto, N. Asokan, A. R. Sadeghi, and S. Tarkoma, "IoT Sentinel: automated device-type identification for security enforcement in IoT," in *International Conference on Distributed Computing Systems*, 2017.

[19] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. A. Wagner, and W. Zhou, "Hidden voice commands," in *USENIX Security*, 2016.

[20] C. Greer, M. Burns, D. Wollman, and E. Griffor, "Cyber-physical systems and Internet of Things," NIST Special Pub 1900-202, March 2019.

[21] M. Wolf and D. Serpanos, "Safety and security in cyber-physical systems and Internet-of-Things systems," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 9–20, 2018.

[22] Y. J. Jia, Q. A. Chen, S. Wang, A. Rahmati, E. Fernandes, Z. M. Mao, and A. Prakash, "ContexIoT: towards providing contextual integrity to appified IoT platforms," in *NDSS*, 2017.

[23] Q. Wang, W. U. Hassan, A. Bates, and C. Gunter, "Fear and logging in the Internet of Things," in *NDSS*, 2018.

[24] W. Zhang, Y. Meng, Y. Liu, X. Zhang, Y. Zhang, and H. Zhu, "HoMonit: monitoring smart home apps from encrypted traffic," in *ACM CCS*, 2018.

[25] M. McCool and E. Reshetova, "Distributed security risks and opportunities in the W3C Web of Things," in *Workshop on Decentralized IoT Security and Standards (DISS)*, 2018.

[26] O. M. Guillen, T. Poppelmann, J. M. Bermudo Mera, E. F. Bongenaar, G. Sigl, and J. Sepulveda, "Towards post-quantum security for IoT endpoints with NTRU," in *Design, Automation & Test in Europe Conference & Exhibition*, 2017.

[27] S. Mare, L. Girvin, F. Roesner, and T. Kohno, "Consumer smart homes: where we are and where we need to go," in *Mobile Computing Systems and Applications*, 2019.

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

[29] E. Fernandes, A. Rahmati, K. Eykholt, and A. Prakash, "Internet of Things security research: a rehash of old ideas or new intellectual challenges?" *IEEE Security & Privacy*, vol. 15, no. 4, pp. 79–84, 2017.

[30] E. Bertino and N. Islam, "Botnets and Internet of Things security," *Computer*, vol. 50, no. 2, pp. 76–79, 2017.

[31] K. Angrishi, "Turning Internet of Things (IoT) into Internet of Vulnerabilities (IoV): IoT botnets," *arXiv preprint arXiv:1702.03681*, 2017.

[32] S. Smith, *The Internet of Risky Things*. O'Reilly Media, 2017.

[33] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "Cryptanalysis of an elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Generation Computer Systems*, vol. 81, pp. 557–565, 2018.

[34] Jackson, "Senate Bill SB-327 information privacy: connected devices," September 28, 2018.