Chen, Fei; Luo, Duming; Xiang, Tao; Chen, Ping; Fan, Junfeng; Truong, Linh

IoT Cloud Security Review: A Case Study Approach Using Emerging Consumer-Oriented Applications

Published in:
ACM Computing Surveys

DOI:
10.1145/3447625

Published: 01/07/2021

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Chen, F., Luo, D., Xiang, T., Chen, P., Fan, J., & Truong, L. (2021). IoT Cloud Security Review: A Case Study Approach Using Emerging Consumer-Oriented Applications. *ACM Computing Surveys*, *54*(4), Article 75. https://doi.org/10.1145/3447625

IoT Cloud Security Review: A Case Study Approach Using Emerging Consumer-Oriented Applications

FEI CHEN[*] and DUMING LUO[*], Shenzhen University, China
TAO XIANG[†], Chongqing University, China
PING CHEN[‡] and JUNFENG FAN[‡], Open Security Research, Inc., China
HONG-LINH TRUONG[§], Aalto University, Finland

Recent years have seen the rapid development and integration of the Internet of Things (IoT) and cloud computing. The market is providing various consumer-oriented smart IoT devices; the mainstream cloud service providers are building their software stacks to support IoT services. With this emerging trend even growing, the security of such smart IoT cloud systems has drawn much research attention in recent years. To better understand the emerging consumer-oriented smart IoT cloud systems for practical engineers and new researchers, this paper presents a review of the most recent research efforts on *existing, real, already deployed* consumer-oriented IoT cloud applications in the past five years using typical case studies. Specifically, we first present a general model for the IoT cloud ecosystem. Then, using the model, we review and summarize recent, representative research works on emerging smart IoT cloud system security using ten detailed case studies, with the aim that the case studies together provide insights into the insecurity of current emerging IoT cloud systems. We further present a systematic approach to conduct a security analysis for IoT cloud systems. Based on the proposed security analysis approach, we review and suggest potential security risk mitigation methods to protect IoT cloud systems. We also discuss future research challenges for the IoT cloud security area.

CCS Concepts: • **Security and privacy → Domain-specific security and privacy architectures**; *Distributed systems security*; • **Computer systems organization** → *Sensor networks*;

Additional Key Words and Phrases: IoT, cloud, security, consumer-oriented smart applications, case study

ACM Reference Format:
Fei Chen, Duming Luo, Tao Xiang, Ping Chen, Junfeng Fan, and Hong-Linh Truong. 2021. IoT Cloud Security Review: A Case Study Approach Using Emerging Consumer-Oriented Applications. *ACM Comput. Surv.* 1, 1, Article 1 (January 2021), 35 pages. https://doi.org/10.1145/3447625

1 INTRODUCTION

The Internet of Things (IoT), a network of everything, has gained prevalence in recent years. It enables users to sense information from various things and control these things back through networking technology, which provides remarkable convenience for users to interact with the surrounding world. Therefore, IoT devices have been widely deployed in different areas during the past ten years with the continuous development of hardware technology and networking technology. Moreover, according to Groupe Speciale Mobile Association (GSMA) prediction [68], IoT devices are estimated to keep being deployed in the future, reaching 25.2 billion devices globally by 2025.

Concurrently with the development of the IoT, cloud computing has also flourished in recent years and has become a new infrastructure of modern society. It provides anytime, anywhere access

Authors' addresses: Fei Chen, fchen@szu.edu.cn; Duming Luo, 1800271043@email.szu.edu.cn, College of Computer Science and Engineering, Shenzhen University, China; Tao Xiang, txiang@cqu.edu.cn, College of Computer Science, Chongqing University, China; Ping Chen, ping.chen@osr-tech.com; Junfeng Fan, fan@osr-tech.com, Open Security Research, Inc. China; Hong-Linh Truong, linh.truong@aalto.fi, Department of Computer Science, Aalto University, Finland.

to storage and computing services through mobile, web, and desktop applications. Despite cloud computing and IoT developed and evolved independently in the past decades, researchers have been integrating cloud computing and IoT in recent years to build more powerful IoT applications; indeed, the cloud can serve as a storage, messaging, and computing backend for IoT devices, which further supports remote data and compute accesses for IoT terminal applications. Mainstream cloud service providers currently all support such IoT cloud services .

IoT application developments are also moving toward an IoT cloud ecosystem direction [88, 113]. For example, Amazon provides Alexa services [118], which is a form of the IoT cloud ecosystem. In such a system, Alexa senses voice data from users and sends the captured data to the cloud. The cloud responds to Alexa after processing the data. Alexa can also control other IoT devices in a user's home, serving as a kind of smart home hub. For example, the user can ask Alexa to turn on a TV, display an image, order food, etc. If the user is not at home, the user can use the terminal application (i.e., mobile phone application) to invoke Alexa. All these are done through cloud services.

While the application of IoT cloud ecosystem architecture is growing, which covers a wide range of scenarios, including wearable devices, smart homes, smart cars (unmanned vehicles), health care, and industrial equipment, its security has gained considerable attention. Researchers have investigated recent existing IoT cloud applications [8, 11, 18, 21, 26, 30, 31, 33, 38, 41, 42, 52, 54, 56, 58, 63, 67, 74, 77, 79, 81, 83–87, 93, 96, 99, 104–106, 108, 119, 121, 123, 124, 126, 132]. It turns out that many of them have security flaws.

Because all IoT cloud systems are related to humans and some of them are related to critical infrastructures, understanding their security is very important. A deep understanding helps protect such systems and their users and further enables better system designs. Toward this end, this paper aims to present a review of the relatively emerging consumer-oriented IoT cloud system area by summarizing current knowledge and proposing future research challenges, with the hope that it provides a reference for both practical developers and researchers who are interested in this area and that it calls for better solutions for IoT cloud systems by solving existing research challenges.

1.1 Related Work and Limitations

There have been a few works that review IoT security and cloud computing security. *For the IoT security area*, Sicari et al. surveyed security research and challenges for IoT systems [101]. This paper also surveyed typical IoT system projects. Alaba et al. summarized the issues that need to be solved in IoT security, including hardware, software, and networking of IoT systems [4]. Khan and Salah reviewed potential blockchain solutions for IoT security [70]. Harbi et al. reviewed the security attacks and security requirements for IoT systems [53]. Stoyanova reviewed IoT data forensics, including challenges, theoretical frameworks, and existing solutions [109]. *For cloud security area*, Khalil et al. reviewed the security vulnerabilities and potential countermeasures of cloud computing services [69]. Singh, Chatterjee, and Basu et al. also reviewed more recent security challenges and solutions for cloud computing [14, 102]. Domingo-Ferrer et al. reviewed privacy-preserving computations on sensitive data in the cloud [36]. Ahmed et al. surveyed trust evaluation issues for cross-cloud federation [3]. Tabrizchi also surveyed up-to-date security and privacy issues of different components of cloud computing [110]. *For the integration of the IoT and cloud*, Ammar et al. reviewed the architecture and security features of mainstream IoT cloud frameworks [9]. Dizdarevic et al. surveyed communication protocols for IoT, fog, and cloud integration [34]. Celik et al. studied security and privacy issues of IoT programming platforms using program analysis techniques [20]. Kumar et al. surveyed security threats and security mechanisms for cloud-based IoT applications [72]. Almolhis also reviewed some general security issues and existing solutions for cloud-based IoT applications [7].

Table 1. Comparison with Recent Surveys.

year	paper	survey topics	IoT cloud integration	deployed, real, large-scale applications	differences in our paper
2015	[101]	IoT security	no	few	different topic
2017	[4]	IoT security	no	few	different topic
2018	[70]	blockchain for IoT security	no	few	different topic
2019	[53]	IoT security	no	few	different topic
2020	[109]	IoT data forensics	few	few	different topic
2014	[69]	cloud security	no	few	different topic
2017	[102]	cloud security	no	few	different topic
2018	[14]	cloud security	no	few	different topic
2019	[36]	privacy-preserving cloud computing	no	few	different topic
2019	[3]	cross-cloud federation trust evaluation	no	few	different topic
2020	[110]	cloud security	no	few	different topic
2018	[9]	mainstream IoT cloud framework security	yes	all	different topic; this paper focuses on IoT cloud application-level security.
2019	[34]	communication protocol security	yes	many	different topic; this paper focuses on IoT cloud application-level security.
2019	[20]	IoT programming platforms security	yes	many	different topic; this paper focuses on IoT cloud application-level security.
2019	[72]	IoT cloud application security	yes	few	more complete summarization, discussion, and real case studies; GDPR privacy discussion; more research challenges
2020	[7]	IoT cloud application security	yes	few	more complete summarization, discussion, and real case studies; GDPR privacy discussion; more research challenges
2020	this work	IoT cloud application security	yes	all	

Compared with previous review works, the integration of IoT and cloud system (i.e., IoT cloud ecosystems) for building smart consumer-oriented applications is just emerging in the consumer market. It is worth noting that IoT cloud integration has been known in the research community; however, they have only been employed by consumer applications very recently. These consumer applications are often used by *large-scale* consumers (e.g., millions of users) with different backgrounds. Understanding their security is a new area for research. It creates new security challenges and is not fully covered by previous reviews as follows.

First, a typical modern IoT cloud ecosystem features a large-scale size, which is larger than that of a traditional IoT system and that of a cloud application. For example, millions of users deploy smart home devices (e.g., smart voice assistants) at their homes. How to manage the devices and protect the data are challenging for such a scale. *Second*, an IoT cloud ecosystem brings more openness, which provides more access points. For example, the IoT end can be obtained/bought by any user who is interested in the system; the cloud end also uses public HTTP GET/PUT services for data exchange between an IoT device and the cloud. An IoT hub may also allow different devices, owned by different users, from different manufacturers to join the IoT cloud system. The different system accessing approaches from different devices and different users pose enormous

challenges for protecting security. *Third*, an IoT cloud ecosystem features more diversity. A typical IoT cloud application that is offered by a commercial company is used by many different users, each of whom buys a different device but the same type. This requires the cloud end to differentiate these different users, which is also quite challenging for the cloud end. A user definitely does not want the user's data to be accessed by other users who use the same type of IoT devices from the same manufacturer. *Fourth*, an IoT cloud ecosystem involves more human participation. It not only senses data from human lives (e.g., in smart home applications) but also offers a mobile app that enables a user to control IoT devices. A device may be shared and used by different users; different users may also have different usage patterns when using the same device. The depth of human involvement also opens a new attack point.

Another limitation of the previous review is the abstractness: it is difficult to have an intuitive and application-level understanding of real, deployed IoT systems due to the lack of detailed examples (e.g., as in [4, 9, 14, 20, 34, 70]). In contrast, this paper adopts a case study approach by presenting an intuitive, up-to-date, and concise review for the security of emerging IoT cloud applications. Table 1 also lists a summary of comparison with recent surveys. For the "IoT cloud integration" column, "no" means that all the reviewed works do not involve IoT cloud integration; "yes" means that all works involve the integration; "few" means that less than 20% works involve the integration. For the "deployed, real, large-scale applications" column, "few" means that less than 20% reviewed works involve real deployed large-scale applications; "many" means that more than 80% works involve real deployed large-scale applications; "all" means that all the works involve real deployed large-scale applications.

1.2 Our Contribution

We present a review of the most recent, representative research works in the relatively emerging area of IoT cloud ecosystem security, with a focus on practical developers/engineers and new researchers. We conduct the review using a detailed case study approach focusing on existing, real, already-deployed consumer-oriented IoT cloud applications. We believe that using a thoroughly discussed case study helps to create an intuitive and deep understanding of the security of IoT cloud ecosystems. Although the reviewed case studies are consumer-oriented emerging applications, when combined, they reflect a large portion of IoT cloud ecosystem security. This review potentially helps IoT cloud ecosystem developers avoid existing security flaws, helps researchers be aware of existing research results and future research challenges in the area of IoT cloud ecosystem security, and calls for innovative solutions to solve the research challenges.

Specifically, we propose a conceptual framework to abstract an IoT cloud ecosystem. We review typical IoT cloud applications in this framework as case studies. In total, we review ten representative applications from most recent research papers and reports. Corresponding to these case studies, we review their security issues. We also propose an analytical approach to systematically understand IoT system security. Based on the approach, we summarize potential countermeasures to these security issues in the reviewed case studies. We further discuss future open research problems in the area of IoT cloud ecosystem security.

1.3 Paper Organization

Figure 1 shows the paper's organization, which intuitively shows the whole paper in a high level and is inspired by other reviews, e.g., [5, 62]. We first present a framework to model IoT cloud ecosystems in Section 2. Then, we review detailed security issues for IoT cloud ecosystems using case studies in Section 3. We choose these case studies as representatives of existing well-adopted IoT cloud applications. Later, we present an analytical approach to understand security and summarize potential protection measures for existing security issues in Section 4. We further discuss future

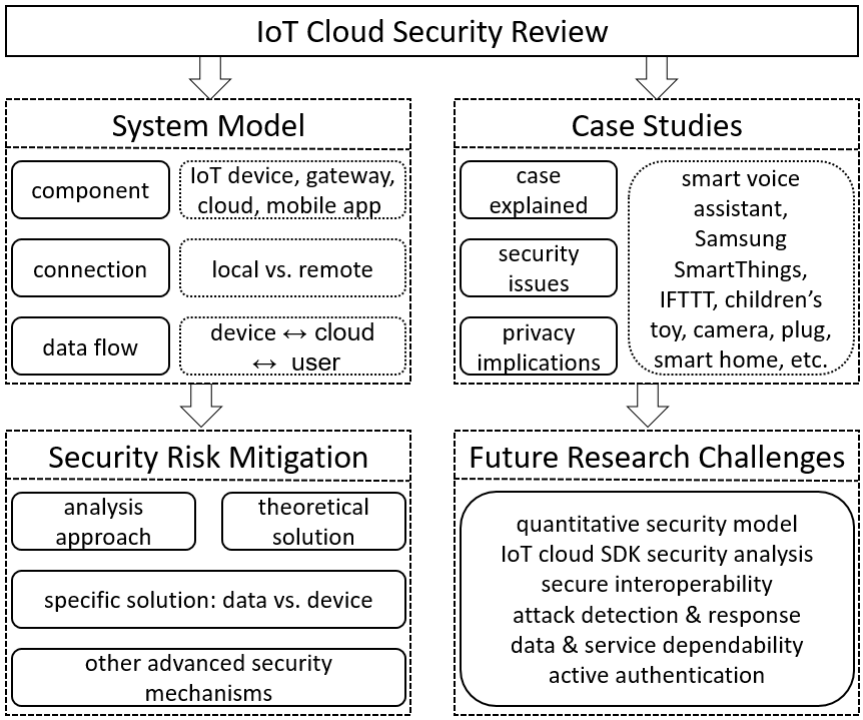

Fig. 1. Paper Organization

research challenges in Section 5. Finally, we conclude the paper in Section 6. For quick reference and high-level overview, Table 2 on page 22 presents a summarization of the reviewed security issues and potential defense approaches for the ten case studies.

2 IOT CLOUD ECOSYSTEM MODEL

2.1 A Motivating Example

Let us first begin with a representative smart home application. Assume that the application has two IoT devices, i.e., the temperature sensor and the smart air conditioner. When the temperature increases to 30 degrees, the temperature sensor uploads the current temperature data to the cloud server through a smart home hub. On receiving the data, the data analysis system deployed in the cloud server finds that the current temperature is too high. The cloud server then returns an instruction to tell the smart air conditioner that it should start working and set the appropriate temperature target. The smart air conditioner turns on and adjusts to the target temperature. However, when the user is uncomfortable due to a cold, even if the temperature is as high as 30 degrees, the user may not be willing to turn on the air conditioner. The user can also directly send an instruction to the smart air conditioner through a control terminal such as a mobile app, which prevents the air conditioner from being turned on. In summary, such an IoT cloud ecosystem can be an autonomous system that adapts itself according to predefined rules; it can also be intervened by human users.

This kind of *user-centric* application is the main focus of this review. Because these applications are emerging and being used by millions of users, their security matters for users, manufacturers, cloud

service providers, and, in general, society as a whole. By understanding their security objectively, it helps to build more secure consumer-oriented IoT cloud systems in the future.

2.2 The IoT Cloud Ecosystem

Abstracting the motivating example in the previous subsection and other applications that we will discuss later, we present an abstract model for the general IoT cloud ecosystem as in Figure 2. An IoT cloud system contains four core components, namely, the IoT device, IoT gateway, cloud, and user. The four components are connected through communication protocols. We explain them in detail.

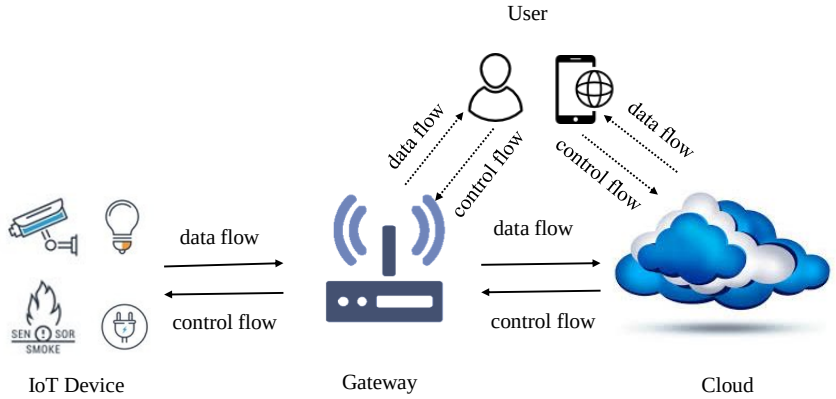

Fig. 2. IoT Cloud Ecosystem Model

2.3 Core Components

The IoT device is an entity that collects data from the environment and/or executes instructions to influence the surrounding environment. In the motivating example, the temperature sensor and the smart air conditioner are two typical IoT devices. Depending on the application scenario, an IoT device can be lightweight, have a low power supply, and have a small memory, which then cannot afford excessive computing. Thus, most IoT devices rely on external sources to enable a large amount of data storage and data analysis. A general solution currently is to rent/set up a (public/private) cloud service.

The IoT gateway is an entity that receives sensed data from an IoT device and transforms the data such that it can be sent on the current Internet infrastructure. In the motivating example, the smart home hub serves as an IoT gateway. Because IoT devices come from different manufacturers and have different design intentions, they may use different communication protocols for transmitting data, such as Bluetooth, ZigBee, and MQTT (i.e., the Message Queuing Telemetry Transport protocol). Thus, the IoT device may send data with different protocols to the IoT gateway; the IoT gateway then performs a unified format transformation and forwards the data to the backend cloud server. With the development of technology, the future IoT gateway may become smarter; it may directly process some simple data, send the processed data to the backend cloud server, or return the corresponding control instructions back to the IoT device.

Cloud is an entity that has a large pool of storage, middleware, and computing resources that are used to host and process sensed data and logs from the IoT devices. The cloud is scalable, geographically independent, can be rented on-demand, and is now the platform for many IoT

device manufacturers to store and analyze data. In general, in an IoT cloud ecosystem, the cloud may serve as (1) the authentication server, which verifies the pairing relationship between an IoT device and the human user; (2) the middleware server, which generally forwards the message between an IoT device and the human user; the storage server, which stores the data collected by the IoT device and its operation log; and (4) the data analysis server, where users can customize the data analysis service to intelligently control the IoT device and visualize the data analysis results. The cloud service can be provided by public service providers. Technically, manufacturers may build their IoT cloud ecosystems from the ground up and then deploy their IoT and web services on the cloud; they may also quickly set up IoT cloud ecosystems using public IoT cloud SDKs provided by mainstream clouds, e.g., the Amazon IoT. We use the term *cloud* to denote the technology trend; in the years before cloud computing prevailed, traditional web services were also used as backend servers.

The *User* is an entity that reads, understands the data returned by the IoT device and the aggregated data from the cloud, and possibly needs to control the IoT device using a terminal, e.g., mobile phones, tablets, and computers. Regardless of how intelligent an IoT cloud ecosystem is, it ultimately better serves people and facilitates people's lives. Because people cannot directly interact with IoT devices and the cloud, the information sent by these entities needs to be visualized by a program installed on a terminal. Similarly, the instructions sent back to the IoT device by a person need to be encoded and sent through the same program. For our focus, all the security issues related to users in this paper generally refer to the security related to the terminal, which is normally a *mobile app*.

2.4 Connection Modes

In an IoT cloud application, the IoT device needs to connect to the cloud to store sensed data; the user also wants to read/control the IoT device. Different connection modes exist for the above two requirements in current practices; the detailed connection model often depends on the application scenario naturally.

Two basic modes exist for connecting the IoT devices to the cloud through the IoT gateway. One is to use a physical device as the IoT gateway. The physical device can usually access the current Internet infrastructure, e.g., using WiFi. The physical device can be a separate device that is different from the IoT device; however, it can also be integrated with the IoT device in the form of a system on a chip (SoC). Another is to use a mobile app as the IoT gateway. In this case, there is no additional hardware, except for the IoT device. The IoT device normally uses a short-range communication mechanism to talk to the mobile app; then, the mobile app sends the sensed data from the IoT device to the cloud. Sometimes, both modes are provided by an IoT cloud application.

There are also two basic modes for the user to connect the IoT device. One is to rely on the cloud to relay the control commands back to the IoT device. This enables a user to control the IoT device remotely. Another is to connect the IoT device directly. The user may connect the IoT device through the IoT gateway remotely, or the user connects the IoT device through the mobile app when the user is near the device. Similarly, both modes are supported by some applications.

It is worth noting that the connection modes may seem complicated at first glance. However, they are simple when we focus on the underlying specific IoT cloud application. This is because they are application dependent. Some applications only require a weak IoT sensor due to its economic cost, which eliminates the need for a separate hardware IoT gateway. Some applications, however, need a strong IoT device that integrates a WiFi module on the system chip or provides a separate gateway for data/signal transformation; this naturally serves as a hardware IoT gateway, which is different from the software gateway using a mobile app.

2.5 Dataflow

In this review, we mainly focus on application-level dataflow, which is user-centric. Existing surveys have covered network level and operating system level dataflows, and we refer to recent related surveys for interested readers [4, 34, 53].

From an application-level view, data are exchanged between the four core components, i.e., the IoT device, IoT gateway, cloud, and user. The data include information collected by the IoT device, operational logs generated by the IoT device, error feedback, current status of the IoT device, authentication information between the user and the cloud, and control instructions sent by the user. These data enable the cloud to judge the status quo of the IoT cloud ecosystem, analyze and predict the data, and formulate the most suitable solution for adjusting the IoT cloud ecosystem. The user may also send the proper instructions to the device according to the working state of the device and the current energy consumption.

The dataflow of an IoT cloud ecosystem is as follows. The IoT device first senses environmental data; then, the data are sent to the IoT gateway, which further sends the raw/processed data to the cloud. Upon receiving the sensed data, the cloud stores the data and processes the data using predefined programs. When needed, the user requests the IoT system data from the cloud, which reflects the running state of the IoT system. Under necessary conditions, the user may also send control instructions to change the state of the IoT device.

2.6 Remark: Industrial IoT, Edge Computing, Fog Computing

We remark that the architecture in Fig. 2 has been known in the research community; it has only been employed by large-scale commercial applications very recently with the age of smart computing. In addition to consumer-oriented IoT cloud applications, researchers are investigating industrial IoT (IIoT) cloud applications for the business sector and the manufacturing sector. In IIoT applications, edge computing and fog computing are used between the IoT gateway and the cloud. This architecture can speed up IIoT data processing; it thus supports real-time, critical IIoT applications. We note that we do not focus on the IIoT in this review. As in Fig. 2, the architecture for consumer-oriented IoT cloud applications only has four components: IoT device, gateway, cloud, and mobile application, but no edge/fog devices.

Compared with industrial IoT applications that use edge/fog devices, the architecture in Fig. 2 is simpler. We use this architecture modeling for three reasons. First, we focus on user-centric emerging IoT cloud applications, which have an impact on millions of users. This architecture is abstracted from and used by the surveyed user-centric IoT cloud applications. Second, this architecture captures the application-level dataflow in a clear and direct manner. Existing surveys have covered network level and operating system level dataflow security for IoT systems, e.g., [4, 34, 53]. Application-level dataflow security is more related to the user for emerging consumer-oriented IoT cloud applications. It deserves attention. Third, when combining all the surveyed consumer-oriented IoT cloud applications, they should reflect a large portion of emerging IoT cloud applications. Real security analysis for consumer-oriented IoT cloud applications is easier because these applications are ready to be obtained and studied from the consumer market. In contrast, real and already-deployed IIoT applications are not easy to approach, which also makes real security analysis hard. Although the reviewed works are consumer oriented, when combined, they should reflect a large portion of emerging IoT cloud systems.

3 IOT CLOUD ECOSYSTEM SECURITY REVIEW: CASE STUDY GUIDED

In this section, we review existing security issues in emerging IoT cloud ecosystems using case studies. Due to regulations such as the European general data protection regulation (GDPR) and

California consumer privacy act (CCPA) in the United States, we also focus on privacy implications for the end user in each reviewed case study. We chose the reviewed cases according to three criteria. First, they should reflect general, representative security issues for an IoT cloud ecosystem. Second, they should be well documented in existing studies. Third, they should represent different aspects of people's lives. As a result, we chose ten representative case studies in total. It is worth noting that the reviewed security issues are not exhaustive; however, when combining all found security issues in these case studies together, they should present a representative, high-level view for the security of emerging consumer-oriented IoT cloud systems.

We organize the structure of each case review as follows. For each use case, we first show how it works. Then, we discuss its security issues found in existing studies, with an aim to give the reader a more intuitive and informative understanding. Next, we present a short summary of the reviewed case for readers' quick reference. We also present the privacy implications for each case study and note the main stakeholder that should handle the reported security issues.

3.1 Case Study: Smart Voice Assistant

3.1.1 Background. Smart voice assistants are one of the most commonly used IoT cloud applications [25, 32, 60, 78, 128, 130]. Their main purpose is to help users call other services through voice commands, such as obtaining weather forecasts, making phone calls, and turning on the TV. Although voice assistants may come from different IoT device manufacturers, the IoT cloud system architecture is basically the same. Typically, the IoT device is a voice assistant. The voice assistant is equipped with a Bluetooth module and a WiFi module. With these two modules and a user mobile application, the device can connect to the Internet. Thus, the voice assistant also acts as an IoT gateway. Device manufacturers generally use third-party cloud services to empower voice assistants with artificial intelligence power, such as Amazon, Google, and their third-party applications in the cloud market space. The user mobile application may also support more voice assistant applications.

Smart voice assistant applications work roughly as follows. First, after the user installs the mobile app, the user can configure parameters for device networking by pairing the device with a Bluetooth connection or using a WiFi connection. Then, the voice assistant starts running by monitoring user voice commands. Once the voice assistant detects the user's voice, it uploads the voice data to the cloud service for voice command analysis. The cloud may analyze the command; the cloud may also transfer the command to third-party service providers that register in the cloud application market space. After the cloud/third-party analysis is completed, the command is issued to the target IoT device. After the target IoT device successfully starts the service, it returns a response to the cloud. Finally, the cloud returns the result to the voice assistant.

3.1.2 Security issues. The voice assistant was reported to have security issues on the cloud side [25, 32, 60, 78, 128, 130]. Because voice assistants require advanced speech recognition technology and natural language processing technology, many voice assistants generally choose third-party cloud services as technical support for IoT clouds. However, third-party cloud services are not as secure and reliable as expected. First, the Google Home/Amazon Alexa third-party application market does not adequately detect the security of a third-party smart voice app, which causes malicious tasks to enter the application market. Using a malicious app, tasks may be called without user installation.

Second, the third-party cloud service has insufficient accuracy for voice command recognition. The review of each task's calling command is also insufficient, which increases the probability of malicious skill being called. It is easy for attackers to reproduce a voice squatting attack (VSA). In such an attack, a malicious application first uses a task name that is similar to a victim's name by

adding some popular human modal particles. For example, a voice command "Alexa, open Capital One please" is transferred to a malicious app called "Capital One please" while the victim app is "Capital One". Even some meaningless commands with the same phoneme can successfully trigger voice commands, which makes it possible for users to inadvertently open tasks they do not need. This reduces the user experience.

Third, third-party cloud services may not accurately identify user intent. This causes the cloud not to accurately switch tasks. Some malicious tasks may reproduce other voice commands. Inadvertently, users can leak their privacy. Fourth, the IoT cloud uses the same private key for the same assistant to encrypt all tasks. This makes it possible to implement man-in-the-middle attacks.

3.1.3 Short summary. This case is summarized as follows for quick reference.

- Entities. Smart voice assistant; mobile apps; cloud server; third-party voice apps.
- Flows. Sensing flow: voice assistant -> cloud server -> third-party voice app. Control flow: mobile app/third-party voice app -> cloud server -> voice assistant.
- Security issues. Insufficient security reviews for third-party voice apps; insufficient recognition of voice commands; unable to accurately identify user intent; usage of the same private key for the cloud to encrypt all voice commands.
- Privacy implications. User privacy is subject to leakage because his/her voice can be used by malicious tasks without the user's consent.
- Key stakeholders. Service provider (e.g., Amazon, Google).

3.2 Case Study: Samsung SmartThings and IFTTT Platform

3.2.1 Background. **Samsung SmartThings**. SmartThings is an IoT ecosystem proposed by Samsung, which includes the IoT cloud, hub, and open application development programming framework [21, 41, 42, 96, 129]. Because Samsung's IoT ecosystem advocates interoperability between different devices, SmartThings also allows third-party developers to further develop device functions through this programming framework. This also means that one device allows multiple apps to control it. By providing the Samsung IoT cloud and IoT gateway, SmartThings achieves this universal functionality.

The workflow of a typical IoT application in this ecosystem is as follows. First, one user finds and installs a mobile app in the SmartThings application market. After installation, the app starts to enumerate IoT devices by using the user's Samsung account information. The user can then view the device statuses and control them through the app. When the user controls one device through the app, the app sends a command to the IoT cloud, which then forwards the command to the hub. Finally, the hub forwards the command to the target device. It is worth noting that in SmartThings, all commands need to go through the IoT cloud to be forwarded to the device.

IFTTT. It is a comprehensive platform that supports cross-device, cross-service interactions and applications [117]. These applications can automate work/life flows. For example, assume that an air conditioner and a smart window from two vendors are connected to IFTTT. IFTTT then supports the interaction and automation of the two devices. IFTTT allows a user to set a rule as follows: "if the temperature is higher than 30 degrees, then open the window." This is called the *trigger-action paradigm* by IFTTT. In addition to IoT devices, the scope of IFTTT is much broader: it supports interactions with various traditional web services, e.g., Gmail, Twitter, and Facebook.

The workflow of a typical trigger-action application in IFTTT is as follows. First, a user sets up different IoT devices from different vendors using different IoT platforms. This is the same as typical IoT cloud applications. The new functionality supported by IFTTT is that the different vendors set up new services in IFTTT. These new services support trigger and action APIs; in IFTTT, it is formed as a rule for easy user usage. A trigger involves a query of a specific device. According to

the query value, an action is performed by the same device or other devices that are invoked by IFTTT through the services published by the device vendors.

3.2.2 Security issues. **Samsung SmartThings**. Researchers have found that SmartThings has security issues in both the IoT cloud and the app [21, 41, 42, 96, 129]. Because SmartThings allows multiple apps to coexist, the interaction between different apps and the interoperability between devices also cause security issues.

In terms of the IoT cloud, its programming framework gives apps too much privilege. First, the app only needs to have the identifier of the IoT device to read all device events. Second, SmartThings does not review the functions required by the app. Instead, the programming framework defaults to granting all privileges to the app, including functions it does not need. This means that the app installed by the user can spy on the user's additional privacy information. There is also a risk that events of other devices may be forged, which leads to the risk of misuse of IoT device functions and thus incurs poor user experience. Third, SmartThings relies too much on password protection for account security. It adopts a single-factor protection mechanism. Once SmartThings account passwords are obtained by an attacker, sensitive data are completely leaked. When the user and the attacker are in the same WiFi environment, the attacker can easily obtain the real reset password link and reset the password.

In terms of the app, its security issue is mainly due to the differences in developers' program development experiences and awareness of program security. As a result, smart apps in the application market have bugs, which weakens the security of the devices in the SmartThings framework. First, the external input is insufficiently cleaned. Smart apps are vulnerable to command injection attacks. Second, the free SMS service of SmartThings can be abused, which leads to the risk of sensitive data leakage.

There are also interoperability cased security issues between different devices because SmartThings advocates an open IoT ecosystem. This means that devices can access each other and that different users can access the same device through different smart apps. Device-to-device access may make the entire IoT ecosystem more intelligent, but it may also cause misuse of functions between devices. For example, the forgery of a device fire alarm event in a smart home system puts other devices in the home at alert status. Because different users interoperate on the same device through different apps, this may cause conflicts between common users of the device or may cause the device to fail to correctly accept the correct command and conduct the correct function.

IFTTT. Existing research has reported that the IFTTT platform mainly has security risks on the interactions between different trigger-action rules. Different trigger-action rules from different devices and different users may conflict with each other. For the above smart air conditioner and smart window example, an attacker aims to enter a user home. Instead of attacking the smart window directly, the attacker may try to attack the smart air conditioner to increase the temperature so that the window will open automatically. This security risk comes from the interaction defined by the trigger-action rule.

For IFTTT, researchers have identified several vulnerability issues for the IFTTT platform [117]. First, the platform has a condition bypassing issue for the trigger-action rule. The trigger can be bypassed by other rules from the user or other devices. Second, action reverting is also possible in the platform. A device can be disconnected from a home WiFi when it is turned on by some trigger-action rule set by other devices. Third, some actions can be executed in a loop, which results in a denial of service attack. Fourth, some rules conflict with each other. For example, the following rules are conflicting: "rule 1: arm the Scout Alarm when the user enters an area; rule 2: turn off the user's home WiFi when the user enters an area". Fifth, similar to rule conflict, rule action can

also be duplicated from different service developers. For example, the actions can be "turn off one device vs. turn off all devices".

3.2.3 Remark. The main goal of this case study is to show that IoT interoperability is important and may incur security risks. Interoperability in IoT cloud ecosystems involves device interoperability, vendor interoperability, platform interoperability, and application-level interoperability. The advantage of involving different devices, vendors, platforms, and higher-level applications is to boost system robustness, prevent a single point of failures, and most importantly, enhance system functionalities. While promoting IoT cloud system interoperability adds considerable value, it also creates security risks due to the more interactions introduced. In this case, we review existing security issues on real IoT cloud systems. However, none of the reviewed cases involves the interoperability of different large-scale IoT platforms, e.g., AWS IoT, and Azure IoT. In our understanding, the main reason is that there are few real, influential IoT cloud systems that are built on top of them *simultaneously*. It can be envisaged that more security issues can be reported in the future when they are adopted on a larger scale. Therefore, we note that improving interoperability security is very important.

3.2.4 Short summary. Samsung SmartThings is summarized as follows for quick reference.

- Entities. IoT device; mobile app; SmartThings cloud server; smart apps in the cloud's application market.
- Flows. Sensing flow: IoT device -> IoT hub -> cloud server -> smart app. Control flow: mobile app/smart app -> cloud server -> IoT device.
- Security issues. Insufficient interoperability control; insufficient privilege/access control; single-factor authentication mechanism; insufficiently cleaned external input; cloud and user SMS services abuse.
- Privacy implications. User privacy is subject to leakage because his/her private information can be accessed by malicious apps without the user's consent.
- Key stakeholders. Service provider, i.e., Samsung SmartThings.

IFTTT is summarized as follows for quick reference.

- Entities. Different IoT devices and platforms; IFTTT.
- Flows. Sensing flow: IoT device/platform -> IFTTT. Control flow: IFTTT -> IoT device/platform.
- Security issues. Rule condition bypassing; action reverting; looped actions; rule conflict; rule duplications.
- Privacy implications. User privacy has a leakage risk if bad rules that leak user privacy exist.
- Key stakeholders. In addition, IFTTT should also help reduce users' risks.

3.3 Case Study: Children's Toys Application

3.3.1 Background. The market is providing a kind of child's toys that runs under the general model of an IoT cloud ecosystem [12, 26, 59, 61, 115]. A hydration tracker is such an application that can scientifically track whether a child drinks water in a timely manner. The application asks a user to input physical information such as the height and weight of a child; based on such information, the application calculates the amount of water that is needed for the good health of a child. The drinking bottle records how much water has been drunk by the child. When the child does not drink enough water, the bottle reminds the child using some animation on the bottle, which sets a target consumption of water on the bottle. This information is also recorded on the mobile app and cloud server provided by the toy's manufacturer. For this application, the IoT device is the bottle. The bottle is also integrated with a WiFi module; thus, the bottle also serves as an IoT gateway.

The application works roughly as follows. First, the water bottle connects to the mobile app, through which the bottle connects to the cloud server. Then, the sensor on the bottle collects the drinking data and sends it to the mobile phone app. The app then sends the data to the manufacturer's server for backup and to multiple third-party cloud servers for analysis. Finally, the cloud returns the analysis results to the app and the bottle.

3.3.2 Security Issues. The hydration tracker was reported to have security issues on the cloud and the user's app [12, 26, 59, 61, 115]. We first discuss the cloud side, which was reported to have four problems. First, POST tokens can be reused. Each time a user drinks water, the user's mobile app sends data reporting the event to the cloud. The attacker can pretend to be a legitimate user by capturing the packet and reusing the HTTP header. Then, the attacker can send arbitrary content to the cloud, including remotely executable codes. Second, access control is flawed. The attacker is able to obtain the profile images by reusing the captured authentication tokens between the user and the cloud. This means that once eavesdropping occurs in the authentication between the app and the cloud, the attacker can generate a valid request to obtain the user's private file stored on the cloud. Third, the HTTP response code leaks private information. Researchers found that the URL token for user pictures consists of 12 letters. As long as the first three letters pair with a legitimate token, the cloud returns the HTTP 301 response; however, the cloud returns HTTP 404 for a nonexistent token. This reduces the time to guess the correct tokens. Combined with the previous two security risks, the attacker can use a brute force attack to exhaust the remaining 9 letters to bulk crawl the user's profile image. Fourth, expired files are not deleted from the cloud. Researchers found that after the user uploads a new profile image, the user can still use the previous link to access the user's previous profile image. This means that the cloud does not delete the information.

The user's mobile app has three security problems. First, plaintext API communication is used. The app does not encrypt the data when forwarding it to the cloud server, which leaks the user's personal privacy to eavesdroppers. The attacker can also learn the interaction format between the app and the cloud, which enables packet spoofs. The spoofed packets may trigger codes and cause smart water bottle or app malfunctions. Second, personal information is leaked. When an error occurs on the network, the app sends a crash report to the third-party analytics platform. However, the report contains personally identifiable information, such as name, gender, birthday, and weight. Third, third-party services are embedded in the user's app. The mobile app uses four third-party analytics and performance monitoring services. The traffic for the connections with these third-party services is encrypted. Thus, it is not known whether private information is leaked.

3.3.3 Short summary. This case is summarized as follows for quick reference.

- Entities. Drinking bottle with sensor; mobile app; cloud server.
- Flows. Sensing flow: bottle -> mobile app -> cloud server. Control flow: cloud server -> mobile app -> bottle.
- Security issues. Reused POST token; authentication token freshness; HTTP response privacy leakage; expired data management; plaintext API communication; third-party analytic package; HTTPS cannot solve all problems.
- Privacy implications. User privacy is subject to leakage because the mobile app leaks the user's information in plaintext.
- Key stakeholders. Service provider.

3.4 Case Study: Web Camera

3.4.1 Background. Web cameras are a popular IoT cloud application [6, 40, 86, 99, 111, 122] that serve as a monitoring tool. It records the scene in front of the camera and stores the video stream. Web cameras come from different manufacturers; while different manufacturers use different business strategies, the system architecture is roughly the same. The IoT device is the camera. The camera is integrated with a WiFi module that can connect the camera to the Internet; thus, the camera also serves as the IoT gateway. The device manufacturers also provide a backend cloud service for storing the data and offer data access from the user using mobile apps.

The web camera application roughly works as follows. After the camera is started, it sends packets to the cloud for registration. Some cameras also send the recorded video in the cloud for longer storage and for remote access to users. The user can connect to the camera in two ways. If the mobile phone and the camera are not in the same local area network (LAN), the mobile app sends its own authentication data to the cloud to obtain the camera information, e.g., IP address. After the cloud matches the information, the camera information is forwarded to the user's mobile app; then, the two are connected. If the mobile phone and the camera are located on the same LAN, there are two situations. One is that the mobile app sends a broadcast packet to find the camera. After the camera receives the information, it authenticates and connects the mobile app. Another is that the mobile app uses the cloud to connect the mobile app to the camera. The camera, on receiving the connection request from the cloud, finds and connects the mobile app in the LAN. Once connected, the camera sends the video stream directly to the user's mobile app. Some applications also use the cloud to forward the camera video stream.

3.4.2 Security Issues. This application has security issues on the IoT device and the cloud [6, 40, 86, 99, 111, 122]. We first discuss the IoT device, for which three issues were reported. First, the debugging interface in the camera may be abused. The interface is generally used to debug the device, but it is not protected at the factory, which enables the attacker to easily enter the device shell. An attacker can modify parameters or inject commands that may directly damage the device or reveal user privacy information. Second, flash memory lacks protection. Some certificates, private authentication algorithms, keys, and other encrypted information are stored in the flash memory, which can be extracted from the camera through reverse engineering. This causes the private encryption mechanisms to be exposed to the attacker and leaks the user's private information. Third, a hardcoded password is used. Some manufacturers use hardcode passwords, certificates, and other encryption mechanisms in flash memory. Combined, flash memory without encryption protection is very easy to crack. Additionally, because the password is hardcoded in the flash memory, it cannot be changed. Once the password is broken by an attacker, the device has no way to update the password.

For the cloud, four issues were reported. First, a plaintext API is used for communication. When the camera communicates with the mobile app and the cloud, the data are not encrypted. The attacker can perform data analysis by capturing packets. Thus, the attacker can obtain the SSID and password of the WiFi and even obtain the password of the mobile app when it connects with the camera remotely. Second, insecure communication protocols are used. Specifically, the applications use private communication protocols without security evaluation but also use the unencrypted HTTP protocol instead of HTTPS. The studied cameras use an unsecured communication protocol to transmit video streams (clear text transmission video streams) and use a proprietary protocol with a low-security level to transmit data. Thus, the attacker can easily crack the video stream of the camera by capturing and cracking the communications. Third, the authentication mechanism is imperfect. Some regular fixed identifiers are used as the main basis for the authentication between the IoT device and the cloud. In the studied cameras, the MAC or a deformed MAC is used as the

basis for identification. Attackers can use this security vulnerability to create large-scale imitation device attacks, which can cause real devices to fail in connection with the cloud server, which brings bad experience to users and damages the reputation of the manufacturer. Fourth, the data collection mechanism is not perfect. The data uploading process from the camera to the cloud does not need to be authenticated again. Thus, the attacker can construct the URL using the same format as the camera; then, the attacker can send a faked video stream to the cloud server without detection. If the cloud server uses a capacity-based billing scheme for individual users, the fake video stream uses the free capacity of the individual user, which indirectly damages the user's property and reduces the user's product experience.

3.4.3 Short summary. This case is summarized as follows for quick reference.

- Entities. Web camera; mobile app; cloud server.
- Flows. Sensing flow: camera -> mobile app; camera -> cloud server. Control flow: cloud server -> mobile app -> camera.
- Security issues. Unprotected debugging interface; unprotected flash memory; hardcoded password; plaintext API for communication; insecure ad-hoc communication protocols; imperfect authentication mechanism; imperfect data collection mechanism.
- Privacy implications. User privacy is subject to leakage because the video stream is transmitted in plaintext on the network.
- Key stakeholders. Service provider.

3.5 Case Study: Indoor Localization for Healthcare Facilities

3.5.1 Background. The indoor localization application monitors the location of patients, medical staff, and various medical devices in hospitals [18, 39, 80]. It uses WiFi signal strength to locate a specific device and uses a cloud service to record the location data. The IoT device in this application is a tiny embedded system, which is called a tracker. It is also equipped with a WiFi module that connects the tracker to the wireless access points in the hospital. Thus, the trackers also serve as IoT gateways that transform and send the sensed data to the cloud server. The application also supports access to cloud data using a mobile app.

The application works as follows. When the tracker is first powered on, it connects to the wireless access points using its local configuration file. The tracker also runs a backend process that opens the wireless signal monitoring mode on the first startup. The process uses the signal strength to locate the tracker and sends the location information to the cloud server. The transmission runs every 5-7 seconds. Each time the data are transmitted, the username of the database, the hash of the table name and password are sent. After the system is running, the user may use a mobile phone to view the location information of various entities in the hospital.

3.5.2 Security Issues. The indoor localization application in healthcare facilities [18] was reported to have security issues on the IoT device and the cloud. The IoT device has two security problems. First, a weak password is used. It was found that the IoT device has a built-in SSH service and that the default password is well documented. The user may use weak passwords for SSH remote login. Developers generally use the default SSH password and do not enhance the password strength after the product leaves the factory. Thus, the attacker may crack the SSH login password using a dictionary attack. Second, the memory card is not protected. The IoT device of this application is located in public areas. Thus, the attacker can remove the memory card inside the device and read the card data with a card reader. It is found that the memory card has internal system files. An attacker can view private information and modify the configuration file.

The cloud also has one security problem, i.e., using plaintext API communication. The abovementioned health facility devices send the database username, table name, and password hash value to the cloud every time data are transmitted. An attacker can capture the communication packets and then know the packet format and database-related information. Once the packet format is obtained by the user, combined with a brute force attack, the attacker can log in to the cloud service to obtain private information. The paper [18] also noted that attackers can use denial of service attacks to take the whole application down, either attacking the IoT device or the cloud.

3.5.3 Short summary. This case is summarized as follows for quick reference.

- Entities. Indoor localization tracker; mobile app; cloud server.
- Flows. Sensing flow: tracker -> cloud server. Control flow: cloud server -> tracker.
- Security issues. Weak password; unprotected memory card; plaintext API communication.
- Privacy implication. User privacy is subject to leakage because patients can be tracked when using medical devices.
- Key stakeholders. Service provider.

3.6 Case Study: Edimax Smart Plug Application

3.6.1 Background. The Edimax plug application is a smart home application that can switch a plug on/off remotely [45, 73, 74]. In addition to controlling whether one home appliance is powered on using a mobile app, the Edimax plug enables viewing the real-time power consumption data, history of the daily, weekly and monthly statistics of the appliance, and can automatically control the power consumption budget. In the event of an emergency, the application notifies the user immediately. The smart plug is the IoT device; it also serves as an IoT gateway that is responsible for connecting the smart plug with the Internet. The application has a cloud service that authenticates the plug and the user; the cloud service can also relay the user's control commands to the plug. In addition to remote control, the user can also connect and control the smart plug locally, e.g., in the same local area network.

The Edimax smart plug application works as follows. First, the Edimax plug connects the user's mobile app, through which the plug knows how to connect to the Internet. Then, the plug sends a packet to the Google server to confirm whether the device has accessed the Internet successfully. The plug also connects a time server to configure the time. After that, the plug registers itself to the manufacturer's remote cloud server. To control the smart plug, the user has two different approaches depending on whether the user and the device lie in the same LAN. If yes, the mobile app immediately sends a specific broadcast packet to find the device; later, on receiving the broadcast packet, the plug responds to the mobile app. The two then perform TCP connections, and the mobile app can operate the device. Otherwise, the two do not locate in the same LAN. The mobile app connects to the cloud server, authenticates the user, requests the cloud to connect the plug and finally relays the user's control command.

3.6.2 Security Issues. The Edimax plug application was reported to have security issues in the IoT device, the cloud, and the user's mobile app [45, 73, 74]. For the IoT device, two problems exist. The Edimax plug uses plaintext communication with the user's mobile app when they are in the same LAN. An attacker, once located on the same LAN, can eavesdrop on the communication. The firmware is not well protected. The attacker can create malicious firmware that suggests to the user to upgrade the IoT device. When the user installs the malicious firmware, the malicious firmware can establish a reverse channel so that the attacker can log in to the device remotely and install different malware to damage the device.

The cloud also has two problems. First, an insecure communication protocol is used. When the Edimax plug interacts with the cloud server, it uses its own encryption protocol. However, this encryption algorithm is simple and has been cracked by attackers. Thus, the attacker obtains the communication packets between the IoT device and the cloud. Second, an imperfect authentication mechanism is used. The authentication between the cloud server and the IoT device depends too much on a regular fixed identity, e.g., the MAC address is one of the identifiers for authentication. In addition, the cloud service allows a brute force attack by enumerating the passwords.

The user's mobile app also has one potential security problem: a weak password. When the user logs into the app, the user may use weak passwords, which are default, short, or simple passwords, such as their birthdays and cell phone numbers.

3.6.3 Short summary. This case is summarized as follows for quick reference.

- Entities. Edimax Smart Plug; mobile app; cloud server.
- Flows. Sensing flow: plug -> cloud server; plug -> mobile app. Control flow: mobile app-> cloud server -> plug.
- Security issues. Plaintext communication in the same LAN; unprotected firmware; insecure ad-hoc communication protocol; problematic authentication mechanism; weak password.
- Privacy implication. User privacy is subject to leakage because the user's pattern on the plug usage can be identified.
- Key stakeholders. Service provider; in addition, users should also use strong passwords.

3.7 Case Study: Industrial Equipment Application

3.7.1 Background. **Smart meter**. Industrial applications have also started employing the IoT cloud ecosystem [116]. The smart meter is one such application in the electrical power industry [10, 27, 120]. The application uses a smart meter to record a user's power consumption and uses a reader to send the detailed consumption data to the backend supplier server. In this application, the smart meter is the IoT device, and the reader is the IoT gateway. Whether a mobile app and remote control are supported is not known, although we can imagine such an evolution in the future. The application works as follows. The smart meter manages and measures energy usage and reports to nearby meter readers. The reader then reports the data to the backend server; the detailed technology can use the current mobile communication infrastructure, which was not reported in detail [120]. The backend supplier server can then compute the energy usage of a specific user, charge the user, and count energy consumption statistics on certain areas.

Temperature-based control systems. Temperature sensors are widely used in the automatic control of temperature-sensitive critical applications [114]. For example, it is used to detect and alarm fire hazards for buildings. It is also used to preserve a critical temperature for cold chain applications. In this kind of application, the IoT device is an analog temperature sensor. The sensor data are sent to the administrator using a gateway. Depending on the detailed application, the administrator may use a terminal computer, a mobile app, or a cloud service to process the sensed temperature.

3.7.2 Security Issues. **Smart meter**. This application mainly has hardware security issues related to the IoT device [120]. Two issues were reported. First, the hardware interface is not protected. After Wurm et al. opened the smart meter shell, the memory can be read and written by re-enabling JTAG (i.e., the debug interface), which means that the attacker can easily read the information in the memory through the hardware interface. Second, there is no sufficient internal hardware data protection mechanism. Wurm et al. analyzed the hardware memory dump and modified the device identity. This causes the smart meter to send the wrong identity information corresponding to

the power electricity user. This not only makes other users suffer economic losses but also makes stealing electricity possible.

Temperature-based control systems. In this application, researchers reported interesting security issues on the IoT device relating to analogy signal processing [114]. The temperature sensor is an analogy device. The sensed temperature can be maliciously modified remotely without being detected by the backend server. The sensor is a nonlinear circuit that has a rectification effect. The circuit output can be changed by electromagnetic interference signals. The security risk is physical in nature. This nonlinear physical phenomenon also exists in other industrial equipment. Thus, this kind of physical security deserves attention. This security risk was validated in a medical application, a laboratory application, and a PID control application [114].

3.7.3 Short summary. Smart meter applications are summarized as follows for quick reference.

- Entities. Smart meter; mobile app; cloud server.
- Flows. Sensing flow: smart meter -> cloud server -> mobile app. Control flow: mobile app-> cloud server -> smart meter.
- Security issues. Unprotected hardware interface; insufficient internal hardware data protection.
- Privacy implication. User privacy is subject to leakage because the user's power consumption pattern can be leaked.
- Key stakeholders. Service provider.

The temperature-based control system application is summarized as follows for quick reference.

- Entities. Temperature sensor; gateway; terminal device/service (e.g., PC, mobile app, cloud)
- Flows. Sensing flow: temperature sensor -> gateway -> terminal device
- Security issues. Electromagnetic interference signals change the nonlinear circuit output and the sensed value without being detected.
- Privacy implications. There is no explicit privacy issue because this application is used to monitor temperature changes.
- Key stakeholders. Service provider.

3.8 Case Study: Baby Monitor Application

3.8.1 Background. Baby monitor applications on the market generally have functions such as video surveillance, temperature monitoring, sleep monitoring, and two-way communication [22, 48, 57, 108]. The IoT device is a video camera integrated with other sensors. Normally, the IoT device is also integrated with communication chips, e.g., WiFi and 3G/4G chipsets, which thus also serve as the IoT gateway. That is, the IoT device and the IoT gateway are integrated into a single piece of equipment. The manufacturer also provides a backend cloud service to store the video data and other sensed data, which enables easy access from the user using a mobile app.

The application roughly works as follows. When the baby monitor starts working, the IoT device/gateway authenticates to the cloud; then, the baby monitor sends the collected data to the cloud. The user may employ the mobile app to request stored data or request two-way communication with the baby monitor. The entire process requires the cloud to forward the data.

3.8.2 Security Issues in Baby Monitor. This application has security risks on the IoT device and the cloud [108]. For the IoT device, five problems were reported. First, the device sends the message using plaintext APIs when the mobile app and the device are on the same local area network. Once the attacker stays in the same LAN, the attacker can eavesdrop on the data collected by the IoT device. Second, the data collected by the IoT device (including its industry standard, encryption format, and key) are stored in the memory card of the IoT device in cleartext. Once the memory

card of the IoT device is removed from the device by the attacker, the privacy of the user and the intellectual property of the manufacturer are greatly threatened. Third, remote shell access is not disabled in the IoT device. When the manufacturer is developing the device, for the sake of convenience, the engineer may choose to remotely debug the device but not cancel the function after the device leaves the factory. Fourth, the default development account is not disabled in the IoT device. The default account has root privileges; its password is simple or just the default password. Even if these accounts are protected by a unique password, an attacker may use password cracking software to deduce the password. Fifth, the hardware universal asynchronous receiver/transmitter (UART) access is not protected. The UART interface is used for diagnosing IoT device problems. Connections through the UART interface generally bypass conventional authentication. An attacker can thus easily make parameter changes to the device through the UART interface. The original parameters of the equipment are designed based on the environment, market, and human needs. Once the parameters are changed, it may endanger people's lives and properties.

For the cloud, two problems were found. First, the cloud communicates with the user's mobile app using plaintext APIs. Once the attacker intercepts the packet on the public network, the user's data are completely leaked. Second, the authentication between the cloud and the IoT device is questionable. It depends on some fixed identity, such as the MAC address, which can be enumerated. This allows the attacker to conduct a large-scale spoofing attack on the cloud server, resulting in other users being unable to use their device for a short period of time; therefore, the manufacturer is also damaged in this attack.

3.8.3 Short summary. This case is summarized as follows for quick reference.

- Entities. Baby monitor; mobile app; cloud server.
- Flows. Sensing flow: baby monitor -> cloud server -> mobile app. Control flow: mobile app -> cloud server -> baby monitor.
- Security issues. Plaintext APIs; stored data in clear text; remote shell access is allowed; the default development account is allowed; unprotected hardware interface access; imperfect authentication between the cloud and the IoT device.
- Privacy implications. The user is subject to privacy leakage because the application stores and transmits the data in plaintext.
- Key stakeholders. Service provider.

3.9 Case Study: Smart Home Application

3.9.1 Background. Researchers have studied different smart home applications [28, 29, 35, 76, 85, 94, 105, 131]. We review two specific applications as follows.

Nest smoke-alarm. This application detects the concentration of carbon monoxide gas, temperature, lighting, and human movements in a house. The IoT device of this application is the Nest smoke-alarm sensor, which collects information about harmful gases, smoke concentrations, light switches, and human motion trajectories. The IoT gateway is integrated with the sensor and uses WiFi to connect to the Internet. The cloud in this application is the server held by the manufacturer and stores the data from the user. The application also provides a mobile app for users to learn about the status of the sensor and to notify when the sensor detects an emergence. The application works as follows. First, the Nest smoke-alarm sensor starts collecting data, uploading approximately 20 kb of sensed data every day to the cloud server. Then, the user opens the mobile app to read the sensed data. The app sends approximately 1 kb of authentication data to the cloud server. The Nest smoke-alarm application uses the OAuth token to authenticate the user, and the server matches the corresponding mobile app according to the internal record. Finally, when the device sensor

detects a dangerous situation, the cloud server immediately sends an emergency notification to the mobile app.

Hue light-bulbs. This application can customize the current light color according to some pictures selected by the user. It can also automatically turn off the light after the user leaves the room. The IoT device is the smart bulb. The IoT gateway is a separate device that can control the smart bulb and communicate with the manufacturer's cloud service. The user can also control the smart bulb through the IoT gateway using a mobile app. When the user is not at home, the user can also control the smart bulb through the cloud service. The application mainly works as follows. First, the smart bulb turns on and is verified at the IoT gateway. Then, the user can use a mobile app or the device-specific website to send control commands or data to the bulb. Specifically, the command is first sent to the cloud, which computes appropriate parameters for the bulb. Then, the command as well as the parameters are sent to the IoT gateway. Finally, the IoT gateway uniformly converts the format and then sends the appropriate signal to the bulb.

3.9.2 Security Issues in Smart Homes. **Nest smoke-alarm**[29, 76, 85]. This application is primarily suspected of collecting too much of users' private information. The Nest smoke-alarm device sends less than 1 KB of verification data to the cloud server when it starts. However, the device sends 20 KB of unknown encrypted data to the cloud server every day. The size of the unknown encrypted packet sent by the device is much larger than the authentication data, which is suspicious. It was also found that after the device is forbidden to send 20 KB of unknown encrypted data to the cloud, the cloud still sends a notification to the user's mobile app in an emergency. Thus, it is suspicious that the unknown encrypted data are somehow private data collected without the user's permission. Depending on the function of the device, the data may include the movement of the person and can determine the state of the user according to the lighting condition. If the above suspicion is true, then the device leaks the user's private information. The cloud server may also abuse the unknown encrypted data, seriously violating user privacy.

Hue Light-Bulb [29, 76, 85]. This application was reported to have security issues on the IoT device and the cloud. The IoT device uses an unchangeable username. It was found that the Hue Light-Bulb uses a fixed hash value as the basis for IoT gateway authentication. In this case, the attacker only needs to capture the hash value of the Hue Light-Bulb to join the IoT gateway; with this hash, the attacker can later construct a packet to fool the IoT gateway. Although the manufacturer later allows the creation of a username, the problem still exists if the username is captured by an attacker.

For the cloud, researchers reported that the cloud communicates with the user's mobile app using plaintext APIs. That is, the interaction data sent by the app to the cloud are not encrypted. However, the plaintext of the packet includes the current state of the light bulb (including brightness, color and warning information); thus, an attacker can know the user's private bulb information by intercepting the data packet.

3.9.3 Short summary. The Nest smoke-alarm is summarized as follows for quick reference.

- Entities. Smoke-alarm; mobile app; cloud server.
- Flows. Sensing flow: Smoke-alarm -> cloud server -> mobile app. Control flow: mobile app -> cloud server -> Smoke-alarm.
- Security issues. Too much encrypted, unknown data collection; privacy concern.
- Privacy implications. There is some potential privacy concern because the application transmits a large quantity of unknown, encrypted data.
- Key stakeholders. Service provider.

Hue light-bulb is summarized as follows for quick reference.

- Entities. Light-bulb; mobile app; cloud server.
- Flows. Sensing flow: light-bulb -> cloud server. Control flow: mobile app -> light-bulb.
- Security issues. Using a fixed hash value for IoT gateway authentication; plaintext APIs.
- Privacy implications. There is no explicit privacy issue because this application is used to control light bulbs.
- Key stakeholders. Service provider.

3.10 Case study: Smart Home Human Factors

3.10.1 Background. The purpose of people using smart home equipment is to facilitate their home life and protect their rights when they are not home. However, the privacy of multiple users is often involved in the home environment. Through this case discussion, we review the interpersonal privacy impact of the installation of smart home devices from a multiuser usage perspective [46, 127, 131].

Consider typical smart home devices, such as smart door locks, smart TVs, smart cameras, and smart speakers. In the smart home scenario, once a user buys and installs a smart IoT device, the device is later shared by the users in the same home. The owner of the device is also called a driver [46]. The sharing causes privacy issues according to [46]. The following facts exist. First, cohabitants may be family members or friends of the user. Second, in the selection and installation of smart home equipment, many drivers do not seek cohabitants' opinions on the installation of smart home equipment. Third, drivers have much greater control over smart home devices than cohabitants. Fourth, drivers have the habit of uploading device data to the IoT cloud.

3.10.2 Security Issues. The fact that smart home IoT devices are shared among several users causes privacy concerns [46, 64, 127, 131]. Compared with drivers, cohabitants are more worried about third-party companies snooping on their privacy. In terms of interpersonal relationships, different users interacting with the same device also cause conflicts between them. As children users grow up, children's privacy is also a concern, which becomes a potential conflict between them and the drivers. Regarding the operation of the device, although the driver will learn how to operate the device seriously, the cohabitants are less enthusiastic at learning. Therefore, when the device fails, users complain more that the device is not intelligent enough. Additionally, as time goes by, once one of the users moves away from the home, the data saved by the cloud service about that user is not deleted. Moreover, the user's account is not automatically deleted. The possibility of the user logging into the account and reoperating the device is not excluded.

3.10.3 Short summary. This case is summarized as follows for quick reference.

- Entities. Smart home IoT devices; shared users; mobile app; cloud server.
- Flows. Sensing flow: Smart home -> cloud server -> mobile app. Control flow: mobile app/shared users -> cloud server -> smart home.
- Security issues. Privacy leakage; data sharing; user authorization; data deletion.
- Privacy implications. There is an implicit privacy issue because many users share the same device while having different privacy requirements.
- Key stakeholders. User.

3.11 Summary of Reported Security Issues

Summarizing the security issues found in the above case studies, we find that most of the problems in current IoT cloud ecosystems show significant similarity. Moreover, these problems are some (maybe novel or combined) variants of traditional computer and network security problems, including operating system security, network security, and Web security. Most of the time, the security issues

come from the complicated interactions between the IoT device, the IoT gateway, the cloud, and the user's app using different communication technologies, e.g., Bluetooth, WiFi, RFID, HTTP, and HTTPS. Table 2 lists a summary of the security issues found. We summarize the details as follows.

Table 2. IoT Cloud Ecosystem Security Review and Risk Mitigation Summary

Cases	Security Issues	Device	Cloud	App	Potential Mitigation
Smart Voice Assistant	insufficient security reviews for third-party skill		√		enhance access control
	insufficient recognition of voice commands		√		enhance AI service
	unable to accurately identify user intent		√		enhance AI service
	reuse of the same private key		√		enhance authentication
SmartThings and IFTTT	insufficient interoperability control		√		enhance access control
	insufficient privilege/access control		√		enhance access control
	single-factor authentication mechanism			√	enhance authentication
	insufficiently cleaned external input		√		enhance authentication, authorization
	cloud and user SMS services abuse		√	√	enhance authorization
	potential trigger-action rule manipulation		√		enhance rule checking
Children's Toy	POST tokens reused		√		establish sound authentications
	authentication token freshness		√		establish sound authentications
	HTTP response privacy leakage		√		enhance privacy policy
	expired data management		√		enhance privacy policy
	plaintext API communication			√	encrypt communication traffic
	third-party analytic package			√	enhance privacy policy
	HTTPS cannot solve all problems			√	establish sound authentications
Web Camera	unprotected debugging interface	√			close hardware debugging interface
	unprotected flash memory	√			encrypt sensitive data
	hardcoded password	√			establish sound authentications
	plaintext API for communication		√		encrypt communication traffic
	insecure ad-hoc communication protocols		√		use sound encryptions
	imperfect authentication mechanism		√		enhance authentication
	imperfect data collection mechanism		√		establish sound privacy policy
Indoor localization for healthcare facilities	weak password	√			use complex passwords
	unprotected memory card	√			encrypt sensitive data
	plaintext API communication		√		encrypt communication traffic
Edimax Smart Plug	plaintext communication in the same LAN	√			encrypt communication traffic
	unprotected firmware	√			enhance firmware protection
	insecure ad-hoc communication protocol		√		encrypt communication traffic
	problematic authentication mechanism		√		establish sound authentications
	weak password			√	use complex passwords
Industrial Equipment	unprotected hardware interface	√			close hardware debugging interface
	insufficient internal hardware data protection	√			encrypt sensitive data
	physical signal interference	√			improve sensor circuit
Baby monitor	plaintext APIs	√	√		encrypt communication traffic
	stored data in clear text	√			encrypt sensitive data
	remote shell access is allowed	√			disable remote login services
	the default development account is allowed	√			disable remote login services
	unprotected hardware interface access	√			close hardware debugging interface
	imperfect authentication for cloud and device		√		establish sound authentications
Smart Home	too much unknown encrypted data	√	√		establish sound privacy policy
	unknown data collection	√	√		establish sound authentications
	privacy concern	√	√		establish sound privacy policy
	using fixed hash value for authentication	√			establish sound authentications
	plaintext APIs		√	√	encrypt communication traffic
Smart home Human factors	privacy leakage		√		establish sound privacy policy
	data sharing		√		enhance access control
	user authorization		√		enhance access control
	data deletion		√		establish sound privacy policy

IoT device. The security issues involved in IoT devices are mainly hardware related. Attackers may obtain private information and encryption keys through the device hardware and even take root access to devices through reverse analysis of unprotected hardware. The attacker can read and even control devices through insecure interfaces and embed malicious code into upgraded drivers. In addition to hardware issues, the IoT device may also communicate with the IoT gateway using the local plaintext API, which leaks data to eavesdroppers.

IoT gateway. The communication between the IoT gateway and the cloud is often not well secured. The reasons can be hardcoded passwords, plaintext communication, insecure authentication mechanisms, private schemes that are not well studied, and insecure Web communications. Another issue is that it may allow remote login with weak credential protection.

Cloud. One main security problem for clouds is correct authentication and access control. The cloud may leak users' private information to attackers who bypass the authentication mechanism or impersonate other devices/users. Another security problem is the use of the plaintext API.

The user. Most often, the user uses a mobile app to read and control the IoT device. The main security issue is that the app may leak private information about the user and the IoT device. Leakage occurs in various forms. For instance, the app binary code contains private information; the app uses third-party services other than the manufacturer; the app sends out private information on system crashes.

Data. The core of an IoT cloud ecosystem is the data. The main security relates to its privacy, integrity, and availability. Most often, privacy is broken due to plaintext transmission and weak encryption. Integrity may be caused by a faked cloud or user. Availability may be caused by denial of service attacks.

A final remark. The reviewed case studies represent current typical IoT cloud ecosystems in the market. It is quite likely that these issues may also occur in future common IoT cloud ecosystems if care is not taken in the system design stage. Thus, we suggest future IoT cloud application designers and developers review their systems against these security issues.

4 IOT CLOUD ECOSYSTEM SECURITY RISK MITIGATION

The research community has not only reported security issues for case studies but also *discussed* potential mitigation methods, which we review in this section. We first propose a security analysis approach to argue for security logically; then, we review detailed mitigation suggestions. Table 2 also provides a quick reference for the reviewed defenses. Because the case studies share some common security problems, they also have some common mitigation methods. Thus, instead of discussing defenses for each case study, we summarize the defenses in two groups following our security analysis approach. One group is from the data protection perspective; another is from the device protection perspective. We note that these mitigation methods are sorted in ascending order according to their importance.

4.1 Security Analysis Approach

For developers and researchers of new IoT cloud ecosystems, it is naturally expected to have a method for analyzing the security of the system thoroughly [19]. While it is possible to list all found issues in Section 3 and check whether they exist in the new system, it is more desirable to have a systematic method for conducting security analysis. In this subsection, based on the experience after reviewing the case studies in the previous section, we propose such an approach.

We summarize the approach using two-step reasoning, as shown in Figure 3. First, we understand the security goal (i.e., security model) and potential attacking points (i.e., threat model). Second, following the dataflows of the IoT cloud ecosystem, check whether the attacking points in the flow have flaws that may break the security goals. We explain more details as follows.

In the first step, security goals include protecting data and protecting devices. Protecting data means that we need to protect the confidentiality, integrity, and availability of the data transmitted and stored in the system. Protecting devices means that we should prevent any malicious modification of the system in both hardware and software. Attacking points are the core components of the IoT cloud system. To protect the data and the device, we need to understand potential threats they face; extensive threat analysis can also be found in recent surveys

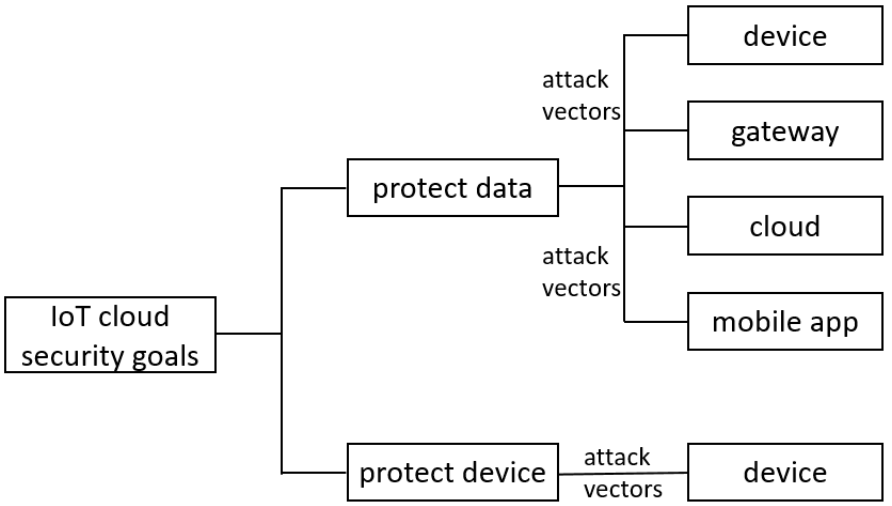

Fig. 3. IoT Cloud Ecosystem Security Analysis Approach

[4, 9, 14, 20, 34, 51, 53, 69, 70, 101, 102]. Specifically, the data can be eavesdropped on during transmission. The eavesdropping can happen in the user's local area network; it can also happen on the cloud side. Attackers can locate near the user or make extensive exploits on the cloud. For the device, although an attacker cannot obtain the user's device easily, the attacker can buy such a device on the market. With the same type of device from the same manufacturer, the attacker can hack the device, learn how it works, and interact with the gateway and the cloud.

In the second step, we first need to understand the data and control flows in the system. This can be achieved by understanding how the system works. For each flow, we determine the sender and the receiver, which are potential attacking points. Then, we analyze how the flow can be broken by an attacker, whose interest is either on the data or on the device. In general, the flow has two directions: one corresponds to the data sensing flow, and the other corresponds to the control flow. One is from the IoT device to the IoT gateway, to the mobile app, and to the cloud. The other direction is from the mobile app to the cloud, to the IoT gateway, and to the IoT device. We note that these two flows differ with the underlying IoT applications, as can be observed in the surveyed case studies. Some IoT cloud systems only support data sensing flow, while others support control flow as well. For most consumer IoT cloud applications, both flows are supported. However, the more flows there are, the more attacking vectors the IoT cloud systems face. For more extensive data flow analysis of IoT cloud applications, we refer to [47, 89, 98, 103].

Finally, it is worth noting that the proposed approach not only enables an abstract security analysis model of an IoT cloud ecosystem but also serves as a logical framework to propose mitigation mechanisms, which we review in the next section. We aim to keep this approach simple and relatively straightforward compared with other surveys; however, it does capture the main points for analyzing IoT cloud system security. The underlying consideration is that we aim to deliver the most important information directly to balance both researchers and engineers.

4.2 Theoretical Solutions

Before showing the specific mitigation suggestions, we discuss general theoretical solutions and enabling techniques, which are universal. In the next subsection, we review detailed mitigation

approaches using the technical mechanisms listed below. Securing data is the core and ultimate goal in both protecting data and protecting devices. Specifically, we need to protect the confidentiality, integrity, and availability of the data.

Confidentiality means the secrecy of the user's data again nonauthorized attackers and users who have no privilege to read the data. General solutions rely on encryption schemes [16, 50, 92]. For example, an IoT cloud system can use standard encryption algorithms (e.g., AES [92]) when transferring the data over the Internet and storing the data in the cloud. We note that general encryption may not be sufficient when there are multiple users and when the stored data are later shared. Each user should have a different secret key; authorization should be supported for access control. In this case, advanced encryption algorithms (e.g., attribute-based encryption [16, 50]) can be used.

Integrity means the requirement that the data should not be modified by an attacker. This applies to the data in transit over the Internet and the data at rest in the cloud. General solutions can use integrity mechanisms, such as cryptographic hash functions, message authentication codes, and digital signatures [15, 37, 65]. Different mechanisms feature different performances and fit different data usage scenarios. The former two approaches rely on private key cryptography, which is normally efficient. The last method is based on public-key cryptography, which is computationally expensive but offers diverse functionalities, e.g., public verification. A user can also combine these techniques to audit his/her data regularly.

Availability is the requirement that the data should never be lost and that the user can access the data whenever the user sends the access request. General solutions use error-correcting schemes and backup schemes [17, 24, 66]. An error-correcting scheme can recover small data corruption using reasonably small additional storage costs. Backup can recover large-scale data lost by storing multiple copies of the data at different locations. The two can be combined in practice.

From a system perspective, an IoT cloud ecosystem needs to employ different technologies to achieve confidentiality, integrity, and availability protection goals. Using HTTPS communication normally protects the data transmission process over the Internet. However, it is only a common communication tool, and using only HTTPS is far from sufficient. An IoT cloud ecosystem needs to further employ proper encryption, authentication, and backup schemes on the device and the cloud side. It is also worth noting that no universal protection mechanism exists. Different IoT cloud application scenarios require different combinations of the above discussed enabling mechanisms according to their user, data, and system scale.

4.3 Data Security Risk Mitigation

Store data securely. This method applies to case studies [18, 86, 99, 108, 120]. Encryption, integrity, and availability mechanisms can be used. For the IoT device and the IoT gateway, important data (such as configuration files, keys, hardware device IDs, and passwords) in the flash memory of the device should be protected, e.g., through encryption. For clouds, sensitive data (such as ID number and mobile phone number) is encrypted. For the user's mobile app, sensitive data should not be hardcoded in the app. While cryptographic protections enhance security, they also create some burden for the cloud to manage the data. For example, the cloud needs to decrypt the data before further processing; encrypted data also weaken system performance. One method is to use code obfuscation and application hardening. Code obfuscation maximizes the optimization of byte codes by removing useless classes, fields, methods, and comments, increasing the difficulty of an attacker understanding the decompiled mobile app binary. Application hardening prevents apps from being reverse analyzed, decompiled, or repackaged. The drawback of obfuscation is that it makes debugging the mobile app complicated and harder.

Encrypt communication traffic. This applies to case studies [18, 26, 74, 85, 86, 99, 105, 108, 120]. The IoT device, the IoT gateway, the cloud, and the user in the IoT cloud ecosystem transfer data through the network. The communications between them should be encrypted. A good choice is to use the HTTPS protocol, which inherently supports encryption and integrity protection; in case HTTPS is not available due to network or certificate problems, end-to-end encryption is needed. While HTTPS incurs performance costs when setting up a connection, SSL sessions can be enabled to improve performance.

Establish sound authentication. This applies to case studies [18, 26, 60, 74, 78, 85, 86, 99, 105, 108, 128]. This is also the hardest part for constructing secure IoT cloud ecosystems. Cryptographic authentication protocols can be used to enhance authentication [49, 55, 91, 100]. We note that using HTTPS alone is not sufficient for sound authentication; examples can be found at [60, 78, 85, 128]. Several methods can be used to enhance authentication. First, add more randomness in the authentication. Current authentication mechanisms often employ a fixed identifier (e.g., MAC address of the device) to conduct authentication between the IoT device, the gateway, and the cloud. However, the MAC address can be enumerated by an attacker for a specific type of device because the MAC addresses of one manufacturer often have regularities, which can be employed by an attacker. In addition to the fixed identifier, embedding more randomness increases security for authentication. Second, the authentication key should be well protected. The key should not be stored in plaintext in the devices and should not be transmitted in plaintext. A secret key management framework can be employed. For instance, the IoT device can have a master secret key stored in its tamper-resistant area. The authentication keys can be freshly, randomly generated from the master secret key for each different authentication. Third, limit the number of erroneous IP connections. In the authentication mechanism, even if more randomness is embedded, the attacker may still use a brute force attack or a dictionary attack on the device's identity or its password. In this case, the cloud needs to limit the number of incorrect connection requests for one specific IP address and directly block this abnormal traffic. Fourth, increase the frequency of reauthentication. Appropriately increasing the frequency of authentication can increase the difficulty of an attacker. Once a device is hacked, the attacker can be detected if more authentications are required by the cloud, the app, etc. Similarly, the validity of the authentication token in an HTTP/HTTPS session should be kept valid only within a reasonable and safe time limit. However, we note that reliable (re)authentication is costly and thus needs to be balanced.

Establish sound access control mechanisms. This applies to case studies [26, 41, 85, 96, 129]. The cloud is mainly responsible for this security protection. Users with different priority levels should have different operation rights for files. For example, an administrator can read and write configuration files. However, ordinary users can only read such files. Any operation involving writing critical files should be notified to the administrator, and the administrator should conduct the final review. When the IoT device initializes the binding with the user's mobile app, the first bound account is the default administrator account; after that, adding a new account is preferably only initiated by the administrator account. Advanced cryptographic approaches can be used to support access control [16, 50]. For example, attribute-based encryption allows flexible access control according to different user attributes. However, efficiency should be balanced for this approach. Access control requires additional computation and storage costs. System resources can be prioritized and then necessary access control can be enforced.

Establish sound privacy policy. This applies to case studies [46, 85, 127, 131]. The IoT device should not collect and send those data without user consent; the cloud should not leak the data to others. This can be achieved by carefully selecting third-party libraries when implementing the IoT device, the cloud backend system, and the mobile app. Data encryption is a potential method for protecting user privacy. Different users should employ different encryption keys. Similarly, encryption may

incur a performance cost. Choosing a proper third-party library also poses a challenge for system developers.

4.4 Device Security Risk Mitigation

Disable remote login services. This applies to [108, 120]. Most users are unwilling to let others remotely log in their own equipment. Thus, due to privacy protection and user experience, the system designer may disable remote login services.

Protect hardware debugging interface. This applies to case studies [86, 99, 108, 120]. The hardware debugging interface is generally used by developers to debug hardware devices or access important files in their operating system before the device leaves the factory. In theory, after the hardware device is shipped, the device manufacturer should disable the interface. However, since the hardware device has the possibility of returning to the factory for repair, it is somehow hard to completely disable the interface. Therefore, device manufacturers should hide the hardware debug interface as much as possible or close the interface if possible.

Enhance firmware protection. This applies to case studies [74, 86, 99, 120]. The firmware updates should be authenticated to ensure that an attacker cannot install a modified firmware on the IoT device and the gateway. The code of the firmware can also be obfuscated so that it requires a high cost for the attacker to understand and modify the code. One drawback is that obfuscation makes debugging harder. This can be mitigated by allowing a small portion of unprotected firmware to collect bug information. Important obfuscation techniques can be found at [13, 75, 97].

Use complex passwords. This applies to case studies [18, 74, 120]. This mitigation can be used for the IoT devices, the IoT gateway, and the user's mobile app. The reason for using weak passwords on the IoT device and IoT gateway is that developers use weak passwords in default (developer) accounts for ease of use. In this regard, developers need to complicate the default passwords or even disable this account before the product leaves the factory. The weak password is used for the user's mobile app because the user's security awareness may not be sufficient. Thus, in addition to reminding users to use complex passwords, the system designer can use more advanced authentication mechanisms with some server-end performance cost.

5 FUTURE RESEARCH CHALLENGES

The IoT cloud ecosystem applications are relatively new and show a growing trend. Compared with traditional computing devices and services, the IoT device and the IoT gateway vary greatly in their computing power, memory capacity, and network bandwidth; the cloud service may also be out of the system administrator's control. Thus, the security of this new computing paradigm is not well understood. We outline some future research challenges.

5.1 Quantitative Security Evaluation/Prediction Model

Establishing a theoretical model for IoT cloud ecosystem security can be useful for both system designers and users (e.g., [90]). For such a model, the input is the configurations of the IoT device, IoT gateway, cloud, and user's mobile app; the output is a numerical score that represents the likely security degree of the underlying application. In addition to the score of the system as a whole, such a model may also output a corresponding score for each component, i.e., the IoT device, the IoT gateway, the cloud, and the user's mobile app.

How to obtain such a model is future work. Intuitively, a machine learning approach can be used. Those applications with no known security issues and those that do have security issues are collected. Their configurations are used as the input. It is also reasonable to collect their behavior patterns (e.g., using dynamic network behavior analysis) as the input. The output can be the score value we want.

5.2 Public IoT Cloud SDK Security Analysis

It is an important future work to analyze existing public IoT cloud software development kits (SDKs) (e.g., [95]). In contrast to developing an IoT cloud ecosystem from the ground up, manufacturers may choose to use existing public IoT cloud SDKs provided by mainstream clouds. The advantage of using an existing SDK includes reducing initial development and time cost when setting up a new IoT cloud ecosystem. The drawback lies in maintaining costs that are charged by cloud providers.

Security analysis for the SDK may include data confidentiality, integrity, and availability. The methodology discussed in Section 4.1 can be used for the SDK security analysis. In addition, because this trend of using SDKs is just emerging, the system dependability of such an approach remains to be understood.

5.3 Security Interoperability Mechanisms

Interoperability refers to the interactions of different elements in a hybrid IoT cloud ecosystem, which is built using different IoT devices and cloud platforms. This kind of hybrid IoT cloud system is emerging recently and has attracted research interests [43, 44, 82, 107, 125]. Indeed, various manufacturers are providing different IoT devices that can sense different data. These devices differ in their computing and networking capabilities as well as cost. Integrated IoT cloud platforms (e.g., AWS IoT, IBM Watson IoT, MS Azure, Google IoT, etc.) are also providing infrastructures to build IoT cloud ecosystems more rapidly and dependably. Due to cost, dependability, stability, and scalability considerations, such a hybrid architecture using different IoT cloud elements is attracting.

While a hybrid IoT cloud ecosystem offers various merits, how to inter-operate between different devices and clouds securely needs extensive research. Important problems include how to protect data confidentiality, integrity, and availability in such a setting. An IoT device, when allowing another IoT device to access its data that may be in the local device or in the remote cloud platform, needs to employ proper mechanism to ensure that only intended data is shared, but not other sensitive data. An identify management mechanism that supports fine-grained authentication and authorization could be useful.

5.4 Attack Detection Mechanism

How to detect whether the IoT cloud application is attacked is worth studying (e.g., [1]). Computing systems have complexity in their nature; it seems that avoiding security issues is a hard task. Indeed, designing an extremely secure system is costly and may not be affordable for consumers. Instead of perusing a 100% secure system, the application may relax the security requirement but offer attack detection mechanisms.

How to design and deploy attack detection requires detailed research. It can be deployed on the IoT device, the IoT gateway, and the cloud. Compared with traditional intrusion detection mechanisms, the IoT cloud application has its own challenges in terms of computing capability. For the IoT device and the IoT gateway, more efficient and affordable attack detection for weak devices deserves research effort. For clouds, which are much more powerful computing entities, more accurate detection methods need to be investigated, which can leverage the data and traffic patterns of the new IoT cloud ecosystem.

5.5 Attack Response Strategy

Following attack detection, how to respond to an IoT cloud application deserves further research (e.g., [2]). A timely and appropriate response can minimize the damage of an attack. The user can stop the loss caused by the attack; the cloud can also resume its service more securely.

To design a response, the IoT application needs to take it into the system design phase. The goals include protecting the attacked user's data and device immediately, protecting other users that are not currently attacked, and protecting its cloud services to function continuously and normally. This poses significant design challenges for the underlying IoT cloud application.

5.6 Data and Service Dependability

It is also important to protect the data and service on the cloud (e.g., [23, 66]). The cloud role in the IoT cloud ecosystem also poses challenges for the underlying application. Because some IoT cloud applications store all/partial data and service on a third-party cloud deployed on a cloud server, what if the cloud service is not available? In addition, an attacker may modify the data and service codes on the cloud.

However, this poses design challenges for an IoT cloud application. The system designer needs to propose security protocols to validate the data storage and the remote cloud service. These security protocols also need to balance factors such as security, performance, and cost.

5.7 Active Authentication Mechanism

One interesting research direction is to design a more active authentication mechanism (e.g., [71, 112]). Current authentications often rely on the user's password and the device's unique identity. This information is static information; they can also be leaked by eavesdropping. In addition to this information, the cloud may ask the user/IoT gateway to input additional information to authenticate itself. For example, location or time can be used in authentication. This may restrict application usage in specific locations or time periods but provide additional security.

The research challenges, however, are devising new ideas for active authentication. The designed authentication protocol also needs to be usable, sound, and efficient. If active authentication is somehow more expensive than traditional authentication, the IoT cloud application may further investigate randomized authentication strategies to combine the use of both authentication schemes. However, this does add more complexity to an IoT cloud application.

6 CONCLUSION

In this paper, we reviewed representative IoT cloud applications and its securities using ten emerging consumer-oriented case studies in different application areas. We also surveyed potential security risk mitigation mechanisms. Finally, we discussed future research issues to enable a more secure IoT cloud application. Because the application of IoT cloud integration is emerging and growing, this survey could help future system designers avoid common security issues and build more secure systems using the security analysis framework. This survey could also serve as a guide for IoT cloud security researchers on state-of-the-art work and future research challenges.

ACKNOWLEDGMENTS

We are thankful for the anonymous reviewers' insightful and helpful comments. This work was partially supported by the National Natural Science Foundation of China under Grant No. (61872243, 62072062, U20A20176, U1713212), Guangdong Basic and Applied Basic Research Foundation (2020A151501489), Natural Science Foundation of Guangdong Province-Outstanding Youth Program under Grants 2019B151502018, Shandong Provincial Key Research and Development Program (Major Scientific and Technological Innovation Project, No.2019JZZY010133), and Key-Area Research and Development Program of Guangdong Province (No.2019B010140001).

REFERENCES

[1] Abebe Abeshu and Naveen Chilamkurti. 2018. Deep learning: The frontier for distributed attack detection in fog-to-things computing. *IEEE Communications Magazine* 56, 2 (2018), 169–175.

[2] Neha Agrawal and Shashikala Tapaswi. 2019. Defense Mechanisms Against DDoS Attacks in a Cloud Computing Environment: State-of-the-Art and Research Challenges. *IEEE Communications Surveys & Tutorials* 21, 4 (2019), 3769–3795.

[3] Usama Ahmed, Imran Raza, and Syed Asad Hussain. 2019. Trust Evaluation in Cross-Cloud Federation: Survey and Requirement Analysis. *Comput. Surveys* 52, 1, Article Article 19 (Feb. 2019), 37 pages.

[4] Fadele Ayotunde Alaba, Mazliza Othman, Ibrahim Abaker Targio Hashem, and Faiz Alotaibi. 2017. Internet of Things security: A survey. *Journal of Network and Computer Applications* 88 (2017), 10–28.

[5] A. S. Albahri, A. A. Zaidan, O. S. Albahri, B. B. Zaidan, and M. A. Alsalem. 2018. Real-Time Fault-Tolerant mHealth System: Comprehensive Review of Healthcare Services, Opens Issues, Challenges and Methodological Aspects. *Journal of Medical Systems* 42, 8 (2018), 137.

[6] Rana Alharbi and David Aspinall. 2018. An IoT Analysis Framework: An Investigation of IoT Smart Cameras' Vulnerabilities, In Living in the Internet of Things: Cybersecurity of the IoT - 2018. *IET Conference Proceedings*, 47 (10 pp.). https://doi.org/10.1049/cp.2018.0047

[7] Nawaf Almolhis, Abdullah Mujawib Alashjaee, Salahaldeen Duraibi, Fahad Alqahtani, and Ahmed Nour Moussa. 2020. The Security Issues in IoT - Cloud: A Review. In *IEEE International Colloquium on Signal Processing Its Applications*. 191–196.

[8] Omar Alrawi, Chaz Lever, Manos Antonakakis, and Fabian Monrose. 2019. SoK: Security evaluation of home-based IoT deployments. In *IEEE Symposium on Security and Privacy*. 208–226.

[9] Mahmoud Ammar, Giovanni Russello, and Bruno Crispo. 2018. Internet of Things: A survey on the security of IoT frameworks. *Journal of Information Security and Applications* 38 (2018), 8–27.

[10] Ross Anderson and Shailendra Fuloria. 2011. Smart meter security: a survey. https://www.cl.cam.ac.uk/~rja14/Papers/JSAC-draft.pdf. (2011).

[11] Noah Apthorpe, Dillon Reisman, and Nick Feamster. 2017. A smart home is no castle: Privacy vulnerabilities of encrypted IoT traffic. *arXiv preprint arXiv:1705.06805* (2017).

[12] Noah Apthorpe, Sarah Varghese, and Nick Feamster. 2019. Evaluating the Contextual Integrity of Privacy Regulation: Parents' IoT Toy Privacy Norms Versus COPPA. In *USENIX Security Symposium*. 123–140.

[13] Sebastian Banescu, Christian Collberg, Vijay Ganesh, Zack Newsham, and Alexander Pretschner. 2016. Code obfuscation against symbolic execution attacks. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*. 189–200.

[14] Srijita Basu, Arjun Bardhan, Koyal Gupta, Payel Saha, M. Pal, Mahasweta Bose, Kaushik Basu, Saunak Chaudhury, and Pritika Sarkar. 2018. Cloud Computing Security Challenges & Solutions - A Survey. In *IEEE 8th Annual Computing and Communication Workshop and Conference*. 347–356.

[15] Daniel J Bernstein. 2005. The Poly1305-AES message-authentication code. In *International Workshop on Fast Software Encryption*. Springer, 32–49.

[16] John Bethencourt, Amit Sahai, and Brent Waters. 2007. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*. IEEE, 321–334.

[17] Kevin D Bowers, Ari Juels, and Alina Oprea. 2009. HAIL: A high-availability and integrity layer for cloud storage. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*. 187–198.

[18] Cole Bradley, Samy El-Tawab, and M Hossain Heydari. 2018. Security analysis of an IoT system used for indoor localization in healthcare facilities. In *Systems and Information Engineering Design Symposium*. IEEE, 147–152.

[19] David D Brandt, Kenwood Hall, Mark Burton Anderson, Craig D Anderson, and George Bradford Collins. 2016. System and methodology providing automation security analysis and network intrusion protection in an industrial environment. US Patent 9,412,073. (2016).

[20] Z. Berkay Celik, Earlence Fernandes, Eric Pauley, Gang Tan, and Patrick McDaniel. 2019. Program Analysis of Commodity IoT Applications for Security and Privacy: Challenges and Opportunities. *Comput. Surveys* 52, 4, Article Article 74 (Aug. 2019), 30 pages.

[21] Z. Berkay Celik, Patrick McDaniel, and Gang Tan. 2018. SOTERIA: Automated IoT Safety and Security Analysis. In *Proceedings of the 2018 USENIX Conference on Usenix Annual Technical Conference (USENIX ATC '18)*. USENIX Association, Berkeley, CA, USA, 147–158.

[22] Silvio Cesare. 2014. Breaking the Security of Physical Devices. Presentation at Blackhat, http://regmedia.co.uk/2014/08/06/dfgvhbhjkui867ujk5ytghj.pdf. (2014).

[23] Fei Chen, Tao Xiang, Yuanyuan Yang, and Sherman SM Chow. 2015. Secure cloud storage meets with secure network coding. *IEEE Trans. Comput.* 65, 6 (2015), 1936–1948.

[24] Henry CH Chen and Patrick PC Lee. 2013. Enabling data integrity protection in regenerating-coding-based cloud storage: Theory and implementation. *IEEE Transactions on Parallel and Distributed Systems* 25, 2 (2013), 407–416.
[25] Yuxuan Chen, Xuejing Yuan, Jiangshan Zhang, Yue Zhao, Shengzhi Zhang, Kai Chen, and XiaoFeng Wang. 2020. DevilâĂŹs Whisper: A General Approach for Physical Adversarial Attacks against Commercial Black-box Speech Recognition Devices. In *29th USENIX Security Symposium*.
[26] Gordon Chu, Noah Apthorpe, and Nick Feamster. 2018. Security and privacy analyses of Internet of Things children's toys. *IEEE Internet of Things Journal* 6, 1 (2018), 978–985.
[27] S. Cleemput, M. A. Mustafa, and B. Preneel. 2016. High Assurance Smart Metering. In *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE)*. 294–297.
[28] B. Copos, K. Levitt, M. Bishop, and J. Rowe. 2016. Is Anybody Home? Inferring Activity From Smart Home Network Traffic. In *2016 IEEE Security and Privacy Workshops (SPW)*. 245–251.
[29] Brittany D Davis, Janelle C Mason, and Mohd Anwar. 2020. Vulnerability Studies and Security Postures of IoT Devices: A Smart Home Case Study. *IEEE Internet of Things Journal* (2020). https://doi.org/10.1109/JIOT.2020.2983983
[30] Tamara Denning, Tadayoshi Kohno, and Henry M. Levy. 2013. Computer Security and the Modern Home. *Communications of the ACM* 56, 1 (Jan. 2013), 94–103.
[31] N Dhanjani. 2013. Hacking lightbulbs: Security evaluation of the philips hue personal wireless lighting system. https://www.dhanjani.com/docs/Hacking Lighbulbs Hue Dhanjani 2013.pdf. (2013).
[32] Wenrui Diao, Xiangyu Liu, Zhe Zhou, and Kehuan Zhang. 2014. Your voice assistant is mine: How to abuse speakers to steal information and control your phone. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*. 63–74.
[33] Wenbo Ding and Hongxin Hu. 2018. On the Safety of IoT Device Physical Interaction Control. In *Proceedings of the 2018 ACM Conference on Computer and Communications Security (CCS '18)*. ACM, New York, NY, USA, 832–846.
[34] Jasenka Dizdarevic, Francisco Carpio, Admela Jukan, and Xavier Masipbruin. 2019. A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration. *ACM Computing Surveys* 51, 6 (2019), 116.
[35] Quang Do, Ben Martini, and Kim-Kwang Raymond Choo. 2018. Cyber-physical systems information gathering: A smart home case study. *Computer Networks* 138 (2018), 1 – 12.
[36] Josep Domingo-Ferrer, Oriol Farras, Jordi Ribes-Gonzalez, and David Sanchez. 2019. Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges. *Computer Communications* 140-141 (2019), 38 – 60.
[37] Morris J. Dworkin. 2015. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Federal Information Processing Standards (NIST FIPS) âĂŞ 202. (2015).
[38] Pardis Emami-Naeini, Sruti Bhagavatula, Hana Habib, Martin Degeling, Lujo Bauer, Lorrie Faith Cranor, and Norman Sadeh. 2017. Privacy Expectations and Preferences in an IoT World. In *Proceedings of the Thirteenth USENIX Conference on Usable Privacy and Security (SOUPS '17)*. USENIX Association, Berkeley, CA, USA, 399–412.
[39] Bahar Farahani, Farshad Firouzi, Victor Chang, Mustafa Badaroglu, Nicholas Constant, and Kunal Mankodiya. 2018. Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare. *Future Generation Computer Systems* 78 (2018), 659 – 676.
[40] Margherita Favaretto, Tu Tran Anh, Juxhino Kavaja, Michele De Donno, and Nicola Dragoni. 2020. When the Price Is Your Privacy: A Security Analysis of Two Cheap IoT Devices. In *Proceedings of 6th International Conference in Software Engineering for Defence Applications*, Paolo Ciancarini, Manuel Mazzara, Angelo Messina, Alberto Sillitti, and Giancarlo Succi (Eds.). Springer International Publishing, Cham, 55–75.
[41] Earlence Fernandes, Jaeyeon Jung, and Atul Prakash. 2016. Security analysis of emerging smart home applications. In *IEEE Symposium on Security and Privacy*. IEEE, 636–654.
[42] E. Fernandes, A. Rahmati, J. Jung, and A. Prakash. 2017. Security Implications of Permission Models in Smart-Home Application Frameworks. *IEEE Security & Privacy* 15, 2 (March 2017), 24–30.
[43] Giancarlo Fortino, Claudio Savaglio, Carlos E Palau, Jara Suarez de Puga, Maria Ganzha, Marcin Paprzycki, Miguel Montesinos, Antonio Liotta, and Miguel Llop. 2018. Towards multi-layer interoperability of heterogeneous IoT platforms: The INTER-IoT approach. In *Integration, interconnection, and interoperability of IoT systems*. Springer, 199–232.
[44] M. Ganzha, M. Paprzycki, W. PawÅĆowski, P. Szmeja, and K. Wasielewska. 2017. Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective. *Journal of Network and Computer Applications* 81 (2017), 111–124.
[45] C. Gao, Z. Ling, B. Chen, X. Fu, and W. Zhao. 2018. SecT: A Lightweight Secure Thing-Centered IoT Communication System. In *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 46–54.
[46] Christine Geeng and Franziska Roesner. 2019. WhoâĂŹs In Control? Interactions In Multi-User Smart Homes. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI âĂŹ19)*. Association for

Computing Machinery, New York, NY, USA, Article Paper 268, 13 pages.
[47] N. K. Giang, M. Blackstock, R. Lea, and V. C. M. Leung. 2015. Developing IoT applications in the Fog: A Distributed Dataflow approach. In *2015 5th International Conference on the Internet of Things (IOT)*. 155–162.
[48] Assaf Glazer. 2016. Systems and methods for configuring baby monitor cameras to provide uniform data sets for analysis and to provide an advantageous view point of babies. US Patent 9,530,080. (2016).
[49] Prosanta Gope, Jemin Lee, and Tony Q. S. Quek. 2018. Lightweight and Practical Anonymous Authentication Protocol for RFID Systems Using Physically Unclonable Functions. *IEEE Transactions on Information Forensics and Security* 13, 11 (2018), 2831–2843.
[50] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. 2006. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*. 89–98.
[51] J. Granjal, E. Monteiro, and J. SÃą Silva. 2015. Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. *IEEE Communications Surveys Tutorials* 17, 3 (2015), 1294–1312.
[52] Daniel Hahn, Noah Apthorpe, and Nick Feamster. 2018. Detecting Compressed Cleartext Traffic from Consumer Internet of Things Devices. *arXiv preprint arXiv:1805.02722* (2018).
[53] Yasmine Harbi, Zibouda Aliouat, Saad Harous, Abdelhak Bentaleb, and Allaoua Refoufi. 2019. A Review of Security in Internet of Things. *Wireless Personal Communications* 108 (2019), 1–20.
[54] George Hatzivasilis, Othonas Soultatos, Sotiris Ioannidis, Christos Verikoukis, Giorgos Demetriou, and Christos Tsatsoulis. 2019. Review of Security and Privacy for the Internet of Medical Things (IoMT). In *15th International Conference on Distributed Computing in Sensor Systems*. IEEE, 457–464.
[55] Debiao He, Sherali Zeadally, Neeraj Kumar, and Wei Wu. 2016. Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. *IEEE Transactions on Information Forensics and Security* 11, 9 (2016), 2052–2064.
[56] Weijia He, Maximilian Golla, Roshni Padhi, Jordan Ofek, Markus Dürmuth, Earlence Fernandes, and Blase Ur. 2018. Rethinking Access Control and Authentication for the Home Internet of Things (IoT). In *Proceedings of the 27th USENIX Conference on Security Symposium (SEC'18)*. USENIX Association, Berkeley, CA, USA, 255–272.
[57] Kashmir Hill. 2014. Baby Monitor Hacker Still Terrorizing Babies And Their Parents. https://www.reshareworthy.com/hacked-baby-monitor/. (2014).
[58] Grant Ho, Derek Leung, Pratyush Mishra, Ashkan Hosseini, Dawn Song, and David Wagner. 2016. Smart Locks: Lessons for Securing Commodity Internet of Things Devices. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIA CCS '16)*. ACM, New York, NY, USA, 461–472.
[59] Donell Holloway and Lelia Green. 2016. The Internet of Toys. *Communication Research and Practice* 2, 4 (2016), 506–519. https://doi.org/10.1080/22041451.2016.1266124 arXiv:https://doi.org/10.1080/22041451.2016.1266124
[60] Hang Hu, Limin Yang, Shihan Lin, and Gang Wang. 2020. Security Vetting Process of Smart-home Assistant Applications: A First Look and Case Studies. *arXiv preprint arXiv:2001.04520* (2020). https://arxiv.org/abs/2001.04520
[61] Patrick C. K. Hung, Farkhund Iqbal, Shih-Chia Huang, Mohammed Melaisi, and Kevin Pang. 2016. A Glance of Child's Play Privacy in Smart Toys. In *International Conference on Cloud Computing and Security*, Xingming Sun, Alex Liu, Han-Chieh Chao, and Elisa Bertino (Eds.). Springer International Publishing, Cham, 217–231.
[62] Muzammil Hussain, A.A. Zaidan, B.B. Zidan, S. Iqbal, M.M. Ahmed, O.S. Albahri, and A.S. Albahri. 2018. Conceptual framework for the security of mobile health applications on Android platform. *Telematics and Informatics* 35, 5 (2018), 1335 – 1354.
[63] Catherine Jackson and Angela Orebaugh. 2018. A study of security and privacy issues associated with the Amazon Echo. *International Journal of Internet of Things and Cyber-Assurance* 1, 1 (2018), 91–100.
[64] William Jang, Adil Chhabra, and Aarathi Prasad. 2017. Enabling Multi-User Controls in Smart Home Devices. In *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*. Association for Computing Machinery, New York, NY, USA, 49âĂŞ54.
[65] Don Johnson, Alfred Menezes, and Scott Vanstone. 2001. The elliptic curve digital signature algorithm (ECDSA). *International journal of information security* 1, 1 (2001), 36–63.
[66] Ari Juels and Alina Oprea. 2013. New approaches to security and availability for cloud data. *Commun. ACM* 56, 2 (2013), 64–73.
[67] Kaushal Kafle, Kevin Moran, Sunil Manandhar, Adwait Nadkarni, and Denys Poshyvanyk. 2019. A Study of Data Store-based Home Automation. In *Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy (CODASPY '19)*. ACM, New York, NY, USA, 73–84.
[68] Sylwia Kechiche. 2018. IoT: the next wave of connectivity and services. https://data.gsmaintelligence.com/research/research/research-2018/iot-the-next-wave-of-connectivity-and-services. (4 2018).
[69] Issa M. Khalil, Abdallah Khreishah, and Muhammad Azeem. 2014. Cloud Computing Security: A Survey. *Computers* 3, 1 (2014), 1–35.

[70] Minhaj Ahmad Khan and Khaled Salah. 2018. IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems* 82 (2018), 395 – 411.

[71] Hokeun Kim, Eunsuk Kang, David Broman, and Edward A. Lee. 2020. Resilient Authentication and Authorization for the Internet of Things (IoT) Using Edge Computing. *ACM Transactions on Internet of Things* 1, 1, Article Article 4 (March 2020), 27 pages. https://doi.org/10.1145/3375837

[72] Raj Kumar, Pramod Kumar, and Vivek Singhal. 2019. A Survey: Review of Cloud IoT Security Techniques, Issues and Challenges. In *Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE)*.

[73] Z. Ling, K. Liu, Y. Xu, Y. Jin, and X. Fu. 2017. An End-to-End View of IoT Security and Privacy. In *2017 IEEE Global Communications Conference (GLOBECOM)*. 1–7.

[74] Zhen Ling, Junzhou Luo, Yiling Xu, Chao Gao, Kui Wu, and Xinwen Fu. 2017. Security vulnerabilities of internet of things: A case study of the smart plug system. *IEEE Internet of Things Journal* 4, 6 (2017), 1899–1909.

[75] Cullen Linn and Saumya Debray. 2003. Obfuscation of executable code to improve resistance to static disassembly. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*. 290–299.

[76] Anindya Maiti and Murtuza Jadliwala. 2019. Light Ears: Information Leakage via Smart Lights. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3, Article Article 98 (Sept. 2019), 27 pages. https://doi.org/10.1145/3351256

[77] Vincentius Martin, Qiang Cao, and Theophilus Benson. 2017. Fending off IoT-hunting Attacks at Home Networks. In *Proceedings of the 2nd Workshop on Cloud-Assisted Networking (CAN '17)*. ACM, New York, NY, USA, 67–72.

[78] Ioannis Agrafiotis Mary K. Bispham and Michael Goldsmith. 2019. Nonsense Attacks on Google Assistant and Missense Attacks on Amazon Alexa. In *Proceedings of the 5th International Conference on Information Systems Security and Privacy*. SciTePress, Prague, Czech Republic.

[79] J. Max. 2016. Backdooring the Frontdoor Hacking a "perfectly secure" smart lock. https://media.defcon.org/DEFCON24/DEFCON24presentations/DEFCON-24-Jmaxxz-Backdooring-the-Frontdoor.pdf. (2016).

[80] T. D. McAllister, S. El-Tawab, and M. H. Heydari. 2017. Localization of Health Center Assets Through an IoT Environment (LoCATE). In *2017 Systems and Information Engineering Design Symposium*. 132–137.

[81] Jack McBride, Julio Hernandez-Castro, and Budi Arief. 2017. Earworms Make Bad Passwords: An Analysis of the Nokē Smart Lock Manual Override. In *International Workshop on Secure Internet of Things*. IEEE, 30–39.

[82] Vittorio Miori, Dario Russo, and Luca Ferrucci. 2019. Interoperability of home automation systems as a critical challenge for IoT. In *2019 4th International Conference on Computing, Communications and Security (ICCCS)*. IEEE, 1–7.

[83] Richard Mitev, Markus Miettinen, and Ahmad-Reza Sadeghi. 2019. Alexa Lied to Me: Skill-based Man-in-the-Middle Attacks on Virtual Assistants. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security (Asia CCS '19)*. ACM, New York, NY, USA, 465–478.

[84] Dang Tu Nguyen, Chengyu Song, Zhiyun Qian, Srikanth V. Krishnamurthy, Edward J. M. Colbert, and Patrick McDaniel. 2018. IotSan: Fortifying the Safety of IoT Systems. In *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '18)*. ACM, New York, NY, USA, 191–203.

[85] Sukhvir Notra, Muhammad Siddiqi, Hassan Habibi Gharakheili, Vijay Sivaraman, and Roksana Boreli. 2014. An experimental study of security and privacy risks with emerging household appliances. In *IEEE Conference on Communications and Network Security*. IEEE, 79–84.

[86] Johannes Obermaier and Martin Hutle. 2016. Analyzing the security and privacy of cloud-based video surveillance systems. In *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security*. ACM, 22–28.

[87] Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama, and Christian Rossow. 2015. IoTPOT: Analysing the Rise of IoT Compromises. In *Proceedings of the 9th USENIX Conference on Offensive Technologies (WOOT'15)*. USENIX Association, Berkeley, CA, USA, 9–9.

[88] T. Pflanzner and A. Kertesz. 2018. A Taxonomy and Survey of IoT Cloud Applications. *EAI Endorsed Transactions on Internet of Things* 3, 12 (4 2018).

[89] Deepak Puthal, Rajiv Ranjan, Surya Nepal, and Jinjun Chen. 2018. IoT and Big Data: An Architecture with Data Flow and Security Issues. In *Cloud Infrastructures, Services, and IoT Systems for Smart Cities*, Antonella Longo, Marco Zappatore, Massimo Villari, Omer Rana, Dario Bruneo, Rajiv Ranjan, Maria Fazio, and Philippe Massonet (Eds.). Springer International Publishing, Cham, 243–252.

[90] Alex Ramos, Marcella Lazar, Raimir Holanda Filho, and Joel JPC Rodrigues. 2017. Model-based quantitative network security metrics: A survey. *IEEE Communications Surveys & Tutorials* 19, 4 (2017), 2704–2734.

[91] Bradley Reaves, Logan Blue, and Patrick Traynor. 2016. Authloop: End-to-end cryptographic authentication for telephony over voice channels. In *25th USENIX Security Symposium*. 963–978.

[92] Vincent Rijmen and Joan Daemen. 2001. Advanced encryption standard. *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology* (2001), 19–22.
[93] E. Ronen and A. Shamir. 2016. Extended Functionality Attacks on IoT Devices: The Case of Smart Lights. In *2016 IEEE European Symposium on Security and Privacy*. 3–12.
[94] Eyal Ronen and Adi Shamir. 2016. Extended functionality attacks on IoT devices: The case of smart lights. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 3–12.
[95] Amirfardad Salami and Alireza Yari. 2018. A framework for comparing quantitative and qualitative criteria of IoT platforms. In *2018 4th International Conference on Web Research (ICWR)*. IEEE, 34–39.
[96] Florian Schmeidl, Bara' Nazzal, and Manar H. Alalfi. 2019. Security Analysis for SmartThings IoT Applications. In *Proceedings of the 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft '19)*. IEEE Press, Piscataway, NJ, USA, 25–29.
[97] Sebastian Schrittwieser, Stefan Katzenbeisser, Johannes Kinder, Georg Merzdovnik, and Edgar Weippl. 2016. Protecting software through obfuscation: Can it keep pace with progress in code analysis? *Comput. Surveys* 49, 1 (2016), 1–37.
[98] J. Schuette and G. S. Brost. 2018. LUCON: Data Flow Control for Message-Based IoT Systems. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. 289–299.
[99] Yogeesh Seralathan, Tae Tom Oh, Suyash Jadhav, Jonathan Myers, Jaehoon Paul Jeong, Young Ho Kim, and Jeong Neyo Kim. 2018. IoT security vulnerability: A case study of a Web camera. In *20th International Conference on Advanced Communication Technology*. IEEE, 172–177.
[100] Jian Shen, Ziyuan Gui, Sai Ji, Jun Shen, Haowen Tan, and Yi Tang. 2018. Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. *Journal of Network and Computer Applications* 106 (2018), 117–123.
[101] S. Sicari, A. Rizzardi, L.A. Grieco, and A. Coen-Porisini. 2015. Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks* 76 (2015), 146 – 164.
[102] Ashish Singh and Kakali Chatterjee. 2017. Cloud security issues and challenges: A survey. *Journal of Network and Computer Applications* 79 (2017), 88 – 115.
[103] Jatinder Singh, Thomas Pasquier, Jean Bacon, Hajoon Ko, and David Eyers. 2015. Twenty security considerations for cloud-supported Internet of Things. *IEEE Internet of things Journal* 3, 3 (2015), 269–284.
[104] Vijay Sivaraman, Dominic Chan, Dylan Earl, and Roksana Boreli. 2016. Smart-Phones Attacking Smart-Homes. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec '16)*. ACM, New York, NY, USA, 195–200.
[105] Vijay Sivaraman, Hassan Habibi Gharakheili, Arun Vishwanath, Roksana Boreli, and Olivier Mehani. 2015. Network-level security and privacy control for smart-home IoT devices. In *IEEE 11th International Conference On Wireless And Mobile Computing, Networking And Communications*. IEEE, 163–167.
[106] Inc. Somerset Recon. 2016. Hello Barbie Initial Security Analysis. https://static1.squarespace.com/static/543effd8e4b095fba39dfe59/t/56a66d424bf1187ad34383b2/1453747529070/HelloBarbieSecurityAnalysis.pdf. (2016).
[107] Sergios Soursos, Ivana Podnar Å¡arko, Patrick Zwickl, Ivan Gojmerac, Giuseppe Bianchi, and Gino Carrozzo. 2016. Towards the cross-domain interoperability of IoT platforms. In *European Conference on Networks and Communications*. 398–402.
[108] Mark Stanislav and Tod Beardsley. 2015. *Hacking IoT: A case study on baby monitor exposures and vulnerabilities*. Technical Report. Rapid 7. https://media.kasperskycontenthub.com/wp-content/uploads/sites/63/2015/11/21031739/Hacking-IoT-A-Case-Study-on-Baby-Monitor-Exposures-and-Vulnerabilities.pdf
[109] Maria Stoyanova, Yannis Nikoloudakis, Spyridon Panagiotakis, Evangelos Pallis, and Evangelos K. Markakis. 2020. A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues. *IEEE Communications Surveys Tutorials* 22, 2 (2020), 1191–1221.
[110] Hamed Tabrizchi and Marjan Kuchaki Rafsanjani. 2020. A survey on security challenges in cloud computing: issues, threats, and solutions. *The Journal of Supercomputing* 76, 12 (2020), 9493–9532.
[111] Ali Tekeoglu and Ali Saman Tosun. 2015. Investigating Security and Privacy of a Cloud-Based Wireless IP Camera: NetCam. In *International Conference on Computer Communication and Networks*. 1–6.
[112] Abhijeet Thakare, Euijong Lee, Ajay Kumar, Valmik B Nikam, and Young-Gab Kim. 2020. PARBAC: Priority-Attribute Based RBAC Model for Azure IoT Cloud. *IEEE Internet of Things Journal* (2020). https://doi.org/10.1109/JIOT.2019.2963794
[113] Hong-Linh Truong and Schahram Dustdar. 2015. Principles for engineering IoT cloud systems. *IEEE Cloud Computing* 2, 2 (2015), 68–76.
[114] Yazhou Tu, Sara Rampazzi, Bin Hao, Angel Rodriguez, Kevin Fu, and Xiali Hei. 2019. Trick or Heat? Manipulating Critical Temperature-Based Control Systems Using Rectification Attacks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS âĂŹ19)*. Association for Computing Machinery, New

York, NY, USA, 2301âĂŞ2315.
[115] Junia Valente and Alvaro A. Cardenas. 2017. Security & Privacy in Smart Toys. In *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy (IoTS&P âĂŹ17)*. Association for Computing Machinery, New York, NY, USA, 19âĂŞ24. https://doi.org/10.1145/3139937.3139947
[116] Pal Varga, Sandor Plosz, Gabor Soos, and Csaba Hegedus. 2017. Security threats and issues in automation IoT. In *IEEE 13th International Workshop on Factory Communication Systems*. 1–6.
[117] Qi Wang, Pubali Datta, Wei Yang, Si Liu, Adam Bates, and Carl A. Gunter. 2019. Charting the Attack Surface of Trigger-Action IoT Platforms. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS âĂŹ19)*. Association for Computing Machinery, New York, NY, USA, 1439âĂŞ1453.
[118] Wikipedia. 2020. Amazon Alexa. https://en.wikipedia.org/wiki/Amazon_Alexa. (8 2020).
[119] Daniel Wood, Noah Apthorpe, and Nick Feamster. 2017. Cleartext data transmissions in consumer IoT medical devices. In *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*. ACM, 7–12.
[120] Jacob Wurm, Khoa Hoang, Orlando Arias, Ahmad-Reza Sadeghi, and Yier Jin. 2016. Security analysis on consumer and industrial IoT devices. In *21st Asia and South Pacific Design Automation Conference*. IEEE, 519–524.
[121] Haitao Xu, Fengyuan Xu, and Bo Chen. 2018. Internet protocol cameras with no password protection: An empirical investigation. In *International Conference on Passive and Active Network Measurement*. Springer, 47–59.
[122] Haitao Xu, Fengyuan Xu, and Bo Chen. 2018. Internet Protocol Cameras with No Password Protection: An Empirical Investigation. In *Passive and Active Measurement*, Robert Beverly, Georgios Smaragdakis, and Anja Feldmann (Eds.). Springer International Publishing, Cham, 47–59.
[123] Moosa Yahyazadeh, Proyash Podder, Endadul Hoque, and Omar Chowdhury. 2019. Expat: Expectation-based Policy Analysis and Enforcement for Appified Smart-Home Platforms. In *Proceedings of the 24th ACM Symposium on Access Control Models and Technologies (SACMAT '19)*. ACM, New York, NY, USA, 61–72.
[124] Tianlong Yu, Vyas Sekar, Srinivasan Seshan, Yuvraj Agarwal, and Chenren Xu. 2015. Handling a Trillion (Unfixable) Flaws on a Billion Devices: Rethinking Network Security for the Internet-of-Things. In *Proceedings of the 14th ACM Workshop on Hot Topics in Networks (HotNets-XIV)*. ACM, New York, NY, USA, Article 5, 7 pages.
[125] Ivana Podnar Žarko, Szymon Mueller, Marcin Płociennik, Tomasz Rajtar, Michael Jacoby, Matteo Pardi, Gianluca Insolvibile, Vasileios Glykantzis, Aleksandar Antonić, Mario Kušek, et al. 2019. The symbIoTe Solution for Semantic and Syntactic Interoperability of Cloud-based IoT Platforms. In *2019 Global IoT Summit*. IEEE, 1–6.
[126] Eric Zeng, Shrirang Mare, and Franziska Roesner. 2017. End User Security & Privacy Concerns with Smart Homes. In *Proceedings of the Thirteenth USENIX Conference on Usable Privacy and Security (SOUPS '17)*. USENIX Association, Berkeley, CA, USA, 65–80.
[127] Eric Zeng and Franziska Roesner. 2019. Understanding and improving security and privacy in multi-user smart homes: a design exploration and in-home user study. In *USENIX Security Symposium*. 159–176.
[128] Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. 2019. Dangerous skills: Understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems. In *IEEE Symposium on Security and Privacy*. IEEE, 0.
[129] Wei Zhang, Yan Meng, Yugeng Liu, Xiaokuan Zhang, Yinqian Zhang, and Haojin Zhu. 2018. HoMonit: Monitoring Smart Home Apps from Encrypted Traffic. In *Proceedings of the 2018 ACM Conference on Computer and Communications Security (CCS âĂŹ18)*. Association for Computing Machinery, New York, NY, USA, 1074âĂŞ1088.
[130] Yangyong Zhang, Lei Xu, Abner Mendoza, Guangliang Yang, Phakpoom Chinprutthiwong, and Guofei Gu. 2019. Life after Speech Recognition: Fuzzing Semantic Misinterpretation for Voice Assistant Applications. In *Proceedings of NDSS*.
[131] Serena Zheng, Noah Apthorpe, Marshini Chetty, and Nick Feamster. 2018. User Perceptions of Smart Home IoT Privacy. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW, Article Article 200 (Nov. 2018), 20 pages.
[132] Wei Zhou, Yan Jia, Yao Yao, Lipeng Zhu, Le Guan, Yuhang Mao, Peng Liu, and Yuqing Zhang. 2019. Discovering and Understanding the Security Hazards in the Interactions Between IoT Devices, Mobile Apps, and Clouds on Smart Home Platforms. In *Proceedings of the 28th USENIX Conference on Security Symposium (SEC'19)*. USENIX Association, Berkeley, CA, USA, 1133–1150.