Extracting Spatial Information of IoT Device Events for Smart Home Safety Monitoring

Yinxin Wan, Xuanli Lin, Kuai Xu, Feng Wang, Guoliang Xue

***Abstract*—Smart home IoT devices have been widely deployed and connected to many home networks for various applications such as intelligent home automation, connected healthcare, and security surveillance. The network traffic traces generated by IoT devices have enabled recent research advances in smart home network measurement. However, due to the cloud-based communication model of smart home IoT devices and the lack of traffic data collected at the cloud end, little effort has been devoted to extracting the spatial information of IoT device events to determine where a device event is triggered. In this paper, we examine why extracting IoT device events' spatial information is challenging by analyzing the communication model of the smart home IoT system. We propose a system named IoTDuet for determining whether a device event is triggered locally or remotely by utilizing the fact that the controlling devices such as smartphones and tablets always communicate with cloud servers with relatively stable domain name information when issuing commands from the home network. We further show the importance of extracting spatial information of IoT device events by exploring its applications in smart home safety monitoring.**

I. Introduction

With the rapid development and widespread deployment of IoT devices in smart homes, many innovative applications such as home automation, connected healthcare, and security surveillance have been brought to end users. However, poorly protected and inadequately managed IoT devices are highly susceptible to cyber attacks [2, 3, 9, 16, 27, 28, 30, 39**?**]. The pervasive IoT devices can be turned into bots for launching distributed denial-of-service (DDoS) attacks [3, 11, 16] or can be compromised for projecting cyber security hazards and physical risks, e.g., unlocking the front door of a smart home via a hacked smart lock [10, 12, 27, 28, 30].

To understand and secure the smart home ecosystem, a lot of work has been proposed for monitoring and analyzing the home network traffic to profile and characterize IoT devices [13, 15, 19, 20, 26, 29, 33]. Some recent papers further explore inferring which IoT device events happen in the smart home based on the home network traffic [1, 32, 34]. Different anomaly detection systems have been proposed [14, 22, 24, 36] for detecting and classifying attacking traffic during the reconnaissance and exploitation phase via machine learning techniques. Some anomaly detection systems at device event level have also been proposed based on the strong assumption that all IoT devices' event logs and the domain knowledge of the physical environments are available [5, 7, 8, 10]. However, these existing research efforts have not studied the problem of determining the spatial locations of the controlling devices that trigger IoT device events by analyzing the home network traffic. Such information can be crucial in understanding the behaviors of IoT devices and detecting malicious IoT device events and smart home user activities, especially in cases where IoT devices have been compromised by attackers.

In this paper, we study the problem of extracting spatial information of IoT device events by first discussing the obstacles and challenges. We then propose IoTDuet for determining whether a device event is triggered locally or remotely. IoTDuet makes efforts in extracting spatial information of IoT device events by collecting and analyzing only the home network traffic, while the outputs of IoTDuet are still informative and crucial for understanding IoT devices' behaviors and detecting safety risks in smart homes.

IoTDuet is designed based on our observations that both the IoT devices and the controlling devices communicate with the cloud servers at almost the same time when a device event is triggered. The IoT devices leave unique and static packet-level traces for each device event as found by [1, 32, 34]. We observe that the controlling devices also leave traces when triggering IoT device events. Specifically, we notice that a controlling device always communicates with the remote cloud server whose domain name or part of the domain name is relatively static and unique when triggering device events. Based on the above findings, IoTDuet examines all of the network traffic generated by or received from all networked devices in the smart home for detecting IoT device event signatures as well as the controlling commands sent by controlling devices to extract spatial information of each device event.

We then utilize the extracted spatial information for applications in home safety monitoring. We first demonstrate that we can detect abnormal device events triggered remotely while the controlling devices are still connected to the home network. We then show that the spatial information of each IoT device event can help us comprehensively monitor all possible home entrance activities and identify potential anomalies. The extensive evaluation results based on a real-world dataset collected at a smart home testbed confirm that IoTDuet can accurately extract the spatial information and we can achieve effective home safety monitoring with such knowledge.

This paper's main contributions are summarized as follows:

- We design IoTDuet to extract the spatial information about whether a smart home IoT device event is triggered locally or remotely by exploiting the fact that a controlling device will communicate with the cloud server whose

All authors are affiliated with Arizona State University. Emails: {ywan28, xlin54, kuai.xu, fwang25, xue}@asu.edu. This research was supported in part by NSF grants 2007469, 1816995, and 1717197. The information reported here does not reflect the position or the policy of the funding agency.

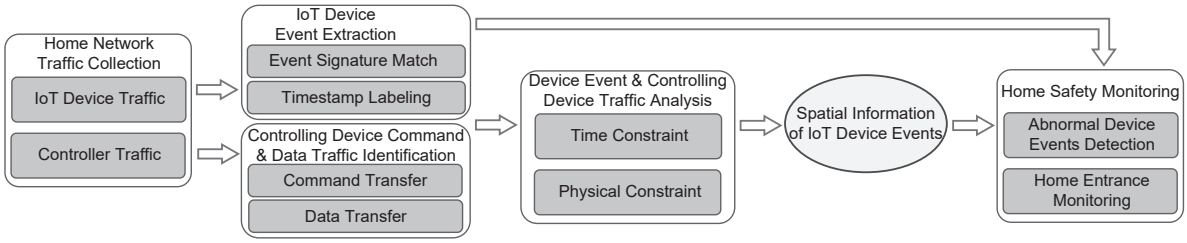

Fig. 1. The overall architecture of our proposed system.

domain name or part of the domain name is relatively static and unique when triggering a device event.

- We propose simple yet effective algorithms for home safety monitoring applications including abnormal device event detection and home entrance monitoring using the extracted spatial information.
- We implement our system on a real-world smart home testbed with 17 different IoT devices and then comprehensively evaluate our proposed system.

The remainder of this paper is organized as follows. Section II discusses the motivation and the architecture. Section III discusses the communication model of the smart home system and pinpoints the challenges in spatial information extraction. Section IV presents the design of IoTDuet in detail. Section V presents our efforts in applying the extracted spatial information of IoT device events for home safety monitoring applications. Section VI presents the evaluation results. Section VII discusses related work and Section VIII concludes this paper.

II. Motivation and System Architecture

In this section, we first discuss the motivation of our paper and then present the overall architecture of our proposed system.

A. Motivation

Our motivation comes from real-world attacking scenarios where smart home IoT devices are compromised for executing malicious device events triggered by attackers [18, 21, 30]. However, these attacks could reveal that the controlling devices triggering the events are in abnormal or suspicious places. Thus, extracting the spatial information of each IoT device event is crucial to better understand the behaviors of smart home IoT devices and users as well as to secure smart homes.

B. System Architecture

Fig. 1 presents the overall architecture of our system which detects whether a device event is triggered locally or remotely. Our system consists of five major components, which are (i) home network traffic collection, (ii) IoT device event extraction, (iii) controlling device's command and data traffic identification, (iv) traffic analysis of device event and controlling device, and (v) home safety monitoring.

The first component of our system is in charge of collecting all incoming and outgoing network traffic of the smart home at the home router. The second component is implemented by adopting state-of-the-art solutions [32, 34] for accurately inferring the device event logs from the collected smart home network traffic. The third component focuses on identifying the command and data transfer traffic of the controlling devices in the home network and building profiles of the domain name information of the cloud servers that the controlling devices send commands and data to. The fourth component combines information captured by the second and third components to determine whether a device event is triggered locally or remotely. The last component leverages the extracted spatial information of IoT device events for critical home safety monitoring applications. We focus on detecting abnormal device events and comprehensively monitor all possible home entrance activities, which are closely related to the cyber and physical security of smart homes.

III. Communication Model of the Smart Home IoT System And Challenges in Device Event Spatial Information Extraction

In this section, we first discuss the communication model of most smart home IoT systems. We then explain the difficulties of extracting the spatial information of IoT device events.

A. Smart Home IoT System Communication Model

As illustrated in Fig.2, a smart home IoT device either directly connects to a home router via WiFi or Ethernet, or indirectly through a hub device via low-energy wireless protocols such as Bluetooth, Zigbee, or Z-Wave for Internet access. To control a smart home IoT device such as turning a smart bulb on, a controlling device (with companion apps installed) needs to connect to the Internet through WiFi or cellular network and then authenticate itself with the cloud services. After connections with the cloud servers are established, the controlling devices can send commands to the servers where the integrity and validity of the commands will be verified. The command messages will be routed inside the AS (autonomous system) of the cloud service providers and then forwarded to the IoT devices through the home router. Upon receiving commands from the cloud servers, the IoT devices will validate them and then take the corresponding actions.

When the controlling devices are physically close to IoT devices, they can directly communicate with the IoT devices and send commands using Bluetooth if supported. In fact, we found that August Smart lock's mobile companion app always prefers using the Bluetooth protocol when the controlling device is within the communication range of the lock even if

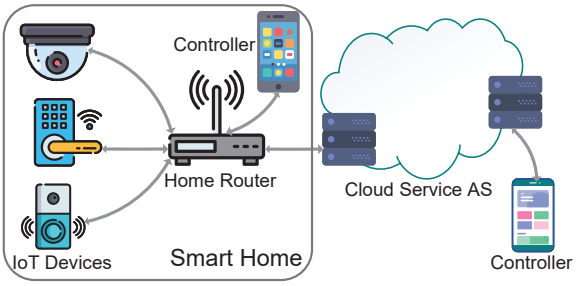

Fig. 2. Communication model of the smart home IoT system. IoT devices are connected to the home router for communicating with the cloud servers via the Internet. The controlling devices send commands to IoT devices indirectly via the cloud servers.

WiFi or cellular network is available. We also found that some IoT devices such as Reolink cameras, directly communicate with the controlling devices connected to the same home network to exchange commands and data.

B. Challenges of Device Events' Spatial Information Extraction

As we can observe from the communication model of smart home IoT devices, IoT devices and controlling devices do not directly communicate with each other for sending and receiving commands, data, and status update information in most cases. In particular, the vendors' cloud servers act as a proxy between IoT devices and controlling devices for forwarding messages. From the smart home users' perspective, capturing the network traffic sent and received by the IoT devices at the home router is practical for most off-the-shelf home routers with little modifications. However, by analyzing the network traffic collected at the home router we can only observe network packets sending to or receiving from the cloud servers. Thus, no information about the controlling devices' IP addresses can be observed by just analyzing the home network traffic.

We thus turn our attention to exploring the IP addresses of the cloud servers to check whether they can help us determine the location of the controlling devices. We examine the DNS traffic of all IoT devices deployed in our smart home testbed and notice that when IoT devices send DNS queries to get the cloud servers' IP addresses, the DNS resolution policies will pick the host that is close to the IoT devices geographically. The traffic will be routed inside the cloud service provider's AS to the host that is close to the controlling device to minimize the network latency. On the other hand, such policy also prevents us from inferring the location of the controlling device by examining the location of the cloud servers the IoT devices communicate with.

On the cloud servers' side, they can learn the IP addresses of both the controlling devices and the IoT devices but such information cannot be directly accessed by smart home users. The lack of network traffic captured at the cloud servers side prevents us from directly knowing the IP addresses of the controlling devices which carry important spatial information.

The inter-packet time intervals of packets sent between the IoT devices and the cloud servers captured at the home router were also considered for inferring the locations of the controlling devices. It can reveal whether the controlling devices are far away from the IoT devices if network latency instead of computation latency at the cloud server dominates the inter-packet time intervals. In fact, we notice that for some devices such as August Lock, sending out commands using a controlling device located on a different continent from where the IoT device is deployed will result in noticeably larger intervals compared to the case where the controlling device is in the same city as the IoT device. However, we can only tell whether the controlling device is significantly far away by analyzing the intervals without knowing the exact geographical location. The variations in network conditions also prevent us from quantifying how far the controlling device is from the IoT device. This strategy also does not apply to all IoT devices because inter-packet intervals of some devices are dominated by the computation latency of the cloud server.

IV. IoTDuet Design

In this section, we present the design of IoTDuet in detail.

A. Home Network Traffic Collection

The first component of IoTDuet is in charge of collecting smart home network traffic at the home router. We first flash the Linux-based OpenWRT operating system to an off-the-shelf home router. Then we collect all raw incoming and outgoing traffic of all devices connected to the home router using software such as tcpdump or tshark. The network traffic of different devices can be grouped by filtering with IP addresses and MAC addresses. Specifically, we identify the network traffic generated by different IoT devices using the hostname field in the header of DHCP response packets, which contain string values that can be mapped to the names of the device vendors, device names, and device models.

B. Device Event Inference from Home Network Traffic

Existing studies have revealed that different IoT device events always generate unique network traffic traces which can be modeled as device event signatures [1, 26, 32, 34]. After extracting device event signatures via controlled experiments, they can be used for inferring device events by matching them in home network traffic [32, 34]. IoTDuet adopts one of the state-of-the-art solutions [32, 34] for generating the traffic signature of each IoT device event with inter-packet interval information, and extracting IoT device events from network traffic collected by the home routers using time-sensitive subsequence matching. We label the timestamp of each extracted device event using the timestamp of the first packet that is matched to it.

C. Identifying Controller's Command and Data Transfer Traces from Home Network Traffic

We start with an example of an unlocking command packed in JSON format sent from an iPhone controller with the Schlage Home mobile app to the cloud server whose domain name

is ‘api.branch.io’ to open the Schlage smart lock. From the key-value pairs in the JSON object illustrated in Fig. 3, we notice that there are many values that could change in different settings for keys such as ‘device_carrier’, ‘connection_type’, ‘latest_update_time’, and ‘app_version’. Thus the payload contents and the packet lengths could vary when sent from different types of controlling devices in different network conditions at different times which makes the packet-level signature hard to be extracted from the controller’s command messages, unlike the device event traces of the IoT devices.

```
"device_carrier" : "Verizon",
"connection_type" : "wifi",
"latest_update_time" : 1641964009951,
"app_version" : "4.4.0",
```

Fig. 3. Examples of the keys in the JSON object corresponding to the unlocking command sent from an iPhone controller to the cloud server with the domain name ‘api.branch.io’ to unlock the Schlage smart lock. The keys ‘device_carrier’, ‘connection_type’, ‘latest_update_time’, and ‘app_version‘ can change if the command is sent from a different controller device in different network conditions at different times.

However, we notice that the domain name of the remote server that the controller communicates with is always ‘api.branch.io’ even if we use different controlling devices such as an Android phone or a tablet. Thus, we use the domain name or part of the domain name of the remote host that command messages and data are sent to as the signature of the controller’s command transfer. As the example in Fig. 4 illustrates, if a controlling device inside the home network initiates Arlo Q camera’s stream on event, we would observe the network traffic of both the packet-level signature [32, 34] of Arlo Q camera’s stream on event and controlling device’s command message which is sent to ‘myapi.Arlo.com’ at the home router. However, when a controlling device outside the home network initiates the same device event, we can only observe the packet-level signature of Arlo Q camera’s stream on device event at the home router. Such differences in the network traffic data collected by the home router can help differentiate whether an IoT device event is triggered by the controlling device connected to the home network or not.

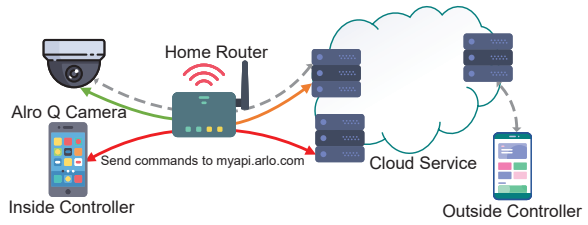

Fig. 4. An example of the network traffic collected at the home router in the scenarios of a controller inside the home network initiating a stream on event and a controller outside the home network initiating a stream on event of the Arlo Q camera.

To correctly extract the domain name information, we installed a MITM (Man-in-the-Middle) root certificate on the controlling devices and applied the MITM proxy for collecting and decrypting the network traffic generated by the controlling device. We then repeat each device event for at least 5 times at different times of the day for each kind of controlling device such as smartphone, tablet, and browser. After verifying that the domain name is consistent over different runs and different controlling platforms, we set it as the signature of the controlling command.

Some device events such as streaming have different kinds of network traffic traces on the controlling devices side because a large number of video stream packets instead of simple reply packets are sent to the controlling devices from the cloud. We observe that for some of the camera and doorbell devices in our testbed, video streams could be sent from the cloud server with the same domain name as the cloud server that the IoT device communicates with for uploading the videos. So, we can apply the same strategy used for command messages to extract the domain name information of the controller’s data transfer traffic.

After building signatures of the controllers’ command and data transfer messages, we can easily look up the packets sent to servers with these domain names in the traffic collected by the home router.

D. Extracting Spatial Information of IoT Device Event

For each device event e_i, we can efficiently extract the spatial information about whether it is triggered locally or remotely, as described in Algorithm 1. We first examine the device event name to check whether it contains keywords in the set H which includes names of the low-energy wireless communication protocols such as Zigbee, Bluetooth, and Z-Wave. Due to the limited wireless communication range of these protocols, when we observe the above keywords, we can claim that such device event is triggered by the controlling device that is close to the home and label the location of e_i as ‘inherently local’.

Algorithm 1: IoTDuet(e_i, P, H, D)

Input: An IoT device event e_i, network packets of all devices connected to the home network $P = \{p_1, p_2, \ldots, p_r\}$, a set H of device event names that are inherently local, a set D of the domain name controlling device communicate with when triggers e_i

Output: Spatial attribute of e_i

$e_i.location \leftarrow$ ‘remote’;
if $e_i.name \in H$ **then**
  $e_i.location \leftarrow$ ‘inherently local’;
else
  $start \leftarrow \arg\min_j(p_j.t, p_j.t > e_i.time - \delta)$;
  $end \leftarrow \arg\max_j(p_j.t, p_j.t < e_i.time + \delta)$;
  for $j := start$ **to** end **do**
    if $p_j.domainName \in D$ **then**
      $e_i.location \leftarrow$ ‘local’;
output $e_i.location$.

We then check the device event name to verify if the device event can only be triggered by a person or an object that is physically close to the device. In our smart home environment,

these device events include motion detection events of all camera and doorbell devices, ringing event of all doorbells, and manual locking or unlocking and autolocking events of all lock devices. These device names are also included in the set H and when observing these device events, we label the location of e_i as 'inherently local'.

If $e_i.name \notin H$ and e_i is extracted from the IoT device's network traffic with timestamp $e_i.time$, we further check the network packets P of all devices connected to the home network. When we observe the packet sent to the host of the domain name in set D which contains domain names that the controlling device will communicate with for sending the command messages or data during the time period of $[e_i.time - \delta, e_i.time + \delta]$, we claim that the device event e_i is triggered by a controlling device in the home network and label the spatial attribute of e_i as 'local'. In our environment, we set $\delta = 5s$ to cope with network latency while minimizing interference. Otherwise, if only the device event e_i is extracted from the IoT device's network traffic and there are no packets sent to the server with the domain name in D during $[e_i.time - \delta, e_i.time + \delta]$, then the device event e_i is considered to be triggered by a controlling device in a different network. For most smartphones, tablets, and laptops, they automatically connect to the home WiFi network when in the communication range unless deliberately disconnected by the user, which is a rare case. So in this condition, we will label the location of e_i as 'remote'.

V. Home Safety Monitoring

The spatial information about smart home IoT device events is critical in smart home security. We explore home safety monitoring applications of detecting abnormal device events and home entrance monitoring.

A. Abnormal Device Event Detection

The spatial information about where a device event is triggered helps us gain a deeper understanding of the device events that happen at a smart home. Thus we can apply this information for detecting if a device event is abnormal or not. We assume that when we observe the network traffic of a controlling device connected to the home network, the device events observed during this time period should be triggered locally. On the contrary, a device event that is triggered remotely during this time should be labeled as anomalies and reported to the smart home users because there is a high possibility that this device event is triggered by attackers who compromised the device remotely.

Algorithm 2 describes how we identify the list of abnormal device events L_e in sequence $\mathbb{E}$. If we observe a device event e_i at time $e_i.time$ triggered by a controlling device that is connected to the home network, and by observing the network traffic in the smart home collected at the home router, we can notice that the controlling device is connected to the home network until t'. In this case, we will check the spatial information of all device events $e_{i+1}, e_{i+2} \ldots, e_u$ where $e_u.time \leq t'$ and $e_{u+1}.time > t'$ extracted using IoTDuet as described in Algorithm 2. If we observe a device event e_j which is triggered by remotely, we will flag it as 'potentially abnormal' and include it in the output list L_e.

Algorithm 2: AbnormalEventsExtract($\mathbb{E}$, P, IoTDuet)

Input: Device event sequence $\mathbb{E} = (e_1, e_2, \ldots, e_m)$, network packets of all devices connected to the home network P
Output: A list L_e of all potential abnormal device events $L_e = \{e_{a_1}, e_{a_2}, \ldots, e_{a_r}\}$

```
L_e ← ∅;
for i := 0 to m do
    if IoTDuet(e_i, P, H, D) == 'local' then
        check P for t' of time that the controlling device
          disconnect the home network;
        for j := i + 1 to m do
            if e_j.time < t' then
                if IoTDuet(e_j, P, H, D) == 'remote' then
                    L_e.insert(e_j);
            else
                i ← j + 1; break;
output List L_e.
```

B. Home Entrance Monitoring

The extracted spatial information of smart home device events can help us better profile the entrance activities in a smart home. In this work, we aim to comprehensively monitor all possible home entrance activities as it is critical to the physical safety and privacy of the smart home residents. We first identify all possible entrance points in the smart home. For each entrance point, we enumerate all possible home entrance activities and build up signatures of them which consist of sequences of device events using the techniques proposed by IoTMosaic [35]. The key difference is that we embedded the spatial information into the signatures of home entrance activities, which means two occurrences of a device event with the same name but different spatial attributes will be treated differently when building up signatures for the entrance activities, unlike IoTMosaic [35].

We also comprehensively consider all kinds of entrance activities at different entrance points which not only include the common ones listed in IoTMosaic [35], but also activities that rarely happen or are potentially 'abnormal'. Based on whether an entrance activity is common for the normal user, we build up a set N consisting of potentially abnormal entrance activities which are seldom triggered by legitimate users. We also build a set T consisting of time-dependent activities which are considered to be abnormal during the specific time period, i.e., midnight time M.

By applying the $k_{\leq}$appxMatch algorithm on the device events $\mathbb{E}$, we can extract all home entrance activities $\mathbb{A} = (A_1, A_2, \ldots, A_n)$ by ordering them based on the timestamp of the first device event in each sequence that is matched to the signature of a home entrance activity. Each extracted home entrance activity A_i has two attributes, activity name $A_i.name$ and its timestamp $A_i.time$. Utilizing spatial and

temporal information of each device event, we can identify a list of abnormal home entrance activities L_a which should be reported to the smart home users using Algorithm 3. We first check the name of each entrance activity A_i and add A_i to L_a if $A_i.name \in N$. We then check the timestamp of A_i and if $A_i.name \in T$ and $A_i.time \in M$, we also include A_i in L_a. In our experiment, we set M =[1:00am, 5:00am] when the users in the smart home and the neighborhood are less active.

Algorithm 3: EntranceActMon($\mathbb{A}, N, T, M$)

Input: Home entrance activity sequence $\mathbb{A} = (A_1, A_2, \ldots, A_n)$, a set of potentially abnormal entrance activities N, a set of time-dependent abnormal entrance activities T, midnight time period M

Output: A list L_a of all potentially abnormal home entrance activities $L_a = \{e_{g_1}, e_{g_2}, \ldots, e_{g_q}\}$

```
L_a ← ∅;
for i := 0 to n do
    if A_i.name ∈ N then
        L_a.insert(A_i);
    else if A_i.name ∈ T and A_i.time ∈ M then
        L_a.insert(A_i);
output List L_a.
```

VI. Experimental Evaluation

In this section, we first present the setup of our experimental environment. Then we discuss our identified hostnames that controllers communicate with for command and data transfer. We then evaluate the performance of IoTDuet in extracting the spatial information of each IoT device event. Subsequently, we evaluate the frameworks of abnormal device event detection and home entrance safety monitoring.

A. Experimental Setup

To systematically and extensively evaluate IoTDuet, we set up a real-world smart home environment where 17 different types of IoT devices are deployed as illustrated in Table I. All of these IoT devices are ranked as popular based on Smart Home DB [31]. For each of these IoT devices, we also identify the different device events it supports and there are 53 different IoT device events in total as listed in Table I.

We identified 3 entrance points in our smart home testbed which are the window, the front door, and the back door. Fig. 5 demonstrates the layout of our smart home environment and the deployment of the IoT devices at the 3 entrance points for home entrance monitoring.

B. Domain Name Extraction of Cloud Servers

For each of the 53 IoT device events, we repeated it for at least 10 times using all possible controlling devices. We marked traffic types and remote cloud servers' host names of the device events that are 'inherently local' as 'N/A' in Table I. The rest of the IoT device events can be triggered by different kinds of controllers such as smartphones and tablets with the mobile companion apps installed. Some of them can also be

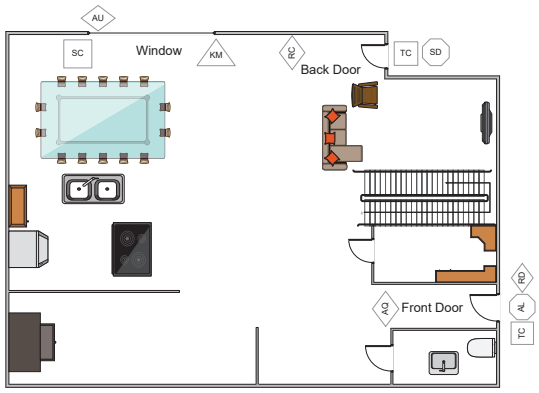

Fig. 5. The layout of our smart home testbed with 3 entrance points and the deployment of IoT devices at the entrance points for home entrance monitoring.

triggered using a web browser or the PC software provided by the vendors. We tested on all of these controlling devices to extract the domain names of the remote cloud servers that the controlling commands of data are sent to.

To accurately identify domain names, we installed a MITM root certificate on the controlling devices and deploy the MITM proxy to collect and decrypt the network traffic sent from it. For mobile applications employing the certificate pinning technique, we patched them using Frida to bypass the certificate pinning. We investigated the packets whose payloads carry commands that are sent to the cloud servers for triggering the corresponding device events or data received from the cloud servers for device events such as video streaming. Since all the TLS packets are already decrypted using MITM, we identified the packets corresponding to the command or data exchange by matching keywords in the payloads such as 'open', 'off', and 'lock'.

We identified domain names of the remote cloud servers for each of the device events which involve the controlling device and we found that the domain name is consistent in all 10 runs in the experiments across all different controlling platforms. The results are illustrated in Table I.

C. Performance Evaluation of IoTDuet

With the domain name of the cloud servers that controllers communicate with, we further evaluated the performance of IoTDuet in detecting whether an IoT device is triggered locally or remotely. We collected the data over a 2-week period where device events are triggered by both controllers connected to the smart home network and by controllers in three different remote places including the same city as the smart home tested, a different city in US west coast, and a different city in US east coast respectively. For each of the triggered device event, we recorded its name and labeled the timestamp.

Fig. 6 illustrates the extracted device events by mapping them to a 24-hour time window. The blue points indicate the device events triggered locally the red points indicate the device events triggered remotely. We notice that IoTDuet can accurately and efficiently determine where a device event is triggered by analyzing the smart home network traffic.

TABLE I

THE DOMAIN NAMES OF THE CLOUD SERVER WHICH THE CONTROLLING DEVICE OF DIFFERENT PLATFORMS COMMUNICATES WITH FOR COMMAND AND DATA EXCHANGE OF 17 IOT DEVICES WITH 53 DIFFERENT IOT DEVICE EVENTS.

Type	Device Name	Device Event (Abbreviation)	Controlling Platform	Traffic Type	Cloud Server's Domain Name
bulb	Philips Hue	on or off (PH_{onoff})	Smartphone, Tablet	command	api2.amplitude.com
		brightness (PH_{br})	Smartphone, Tablet	command	api2.amplitude.com
	Sengled SmartLED	on (SS_{on})	Smartphone, Tablet	command	*.cloud.sengled.com
		off (SS_{off})	Smartphone, Tablet	command	*.cloud.sengled.com
		brightness (SS_{br})	Smartphone, Tablet	command	*.cloud.sengled.com
	TP-Link Bulb	on (TB_{on})	Smartphone, Tablet	command	api.tplinkra.com
		off (TB_{off})	Smartphone, Tablet	command	api.tplinkra.com
		color (TB_{cl})	Smartphone, Tablet	command	api.tplinkra.com
		brightness (TB_{br})	Smartphone, Tablet	command	api.tplinkra.com
camera	Amcrest ProHD	stream on (AP_{on})	Smartphone, Tablet, Web	command	*.compute.amazonaws.com
	Arlo - Q Indoor	stream on (AQ_{on})	Smartphone, Tablet, Web	command	myapi.arlo.com
		stream off (AQ_{off})	Smartphone, Tablet, Web	command	myapi.Arlo.com
		streaming (AQ_{str})	Smartphone, Tablet, Web	data	arlo*.*.amazonaws.com
		motion detection (AQ_{mot})	inherently local	N/A	N/A
	Arlo Ultra	stream on (AU_{on})	Smartphone, Tablet, Web	command	myapi.arlo.com
		stream off (AU_{off})	Smartphone, Tablet, Web	command	myapi.arlo.com
		streaming (AU_{str})	Smartphone, Tablet, Web	data	arlo*.*.amazonaws.com
		motion detection (AU_{mot})	inherently local	N/A	N/A
	Blink XT2	stream on (BX_{on})	Smartphone, Tablet	command	*.immedia-semi.com
		stream off (BX_{off})	Smartphone, Tablet	command	*.immedia-semi.com
		streaming (BX_{str})	Smartphone, Tablet	data	*.compute.amazonaws.com
		motion detection (BX_{mot})	inherently local	N/A	N/A
	Reolink Camera	stream on (RC_{on})	Smartphone, Tablet, Web, PC	command	apis.reolink.com
		stream off (RC_{off})	Smartphone, Tablet, Web, PC	command	apis.reolink.com
		streaming (RC_{str})	Smartphone, Tablet, Web, PC	data	direct
		motion detection (RC_{mot})	inherently local	N/A	N/A
	Yi Camera	stream on (YC_{on})	Smartphone, Tablet	command	gw-us.xiaoyi.com
		stream off (YC_{off})	Smartphone, Tablet	command	gw-us.xiaoyi.com
		streaming (YC_{str})	Smartphone, Tablet	data	*.aliyun.com
		motion detection (YC_{mot})	inherently local	N/A	N/A
doorbell	August Doorbell Cam Pro	stream on (AD_{on})	Smartphone, Tablet	command	api-production.august.com
		stream off (AD_{off})	Smartphone, Tablet	command	api-production.august.com
		streaming (AD_{str})	Smartphone, Tablet	data	*.compute.amazonaws.com
		ringing (AD_{ring})	inherently local	N/A	N/A
		motion detection (AD_{mot})	inherently local	N/A	N/A
	Ring VideoDoorbell	stream on (RD_{on})	Smartphone, Tablet, Web	command	clientapigw.ring.com
		stream off (RD_{off})	Smartphone, Tablet, Web	command	clientapigw.ring.com
		streaming (RD_{str})	Smartphone, Tablet, Web	data	*.compute.amazonaws.com
		ringing (RD_{ring})	inherently local	N/A	N/A
		motion detection (RD_{mot})	inherently local	N/A	N/A
lock	August Lock Pro	WiFi (un)locking (AL_{wlk})	Smartphone, Tablet	command	api-production.august.com
		Bluetooth (un)locking (AL_{blk})	inherently local	N/A	N/A
		autolocking (AL_{alk})	inherently local	N/A	N/A
		manual (un)locking (AL_{mlk})	inherently local	N/A	N/A
	Schlage WiFi Deadbolt	WiFi (un)locking (SD_{wlk})	Smartphone, Tablet	command	api2.branch.io
		autolocking (SD_{alk})	inherently local	N/A	N/A
		manual (un)lock (SD_{mlk})	inherently local	N/A	N/A
plug	Amazon Smart Plug	on (AP_{on})	Smartphone, Tablet	command	api.amazon.com
		off (AP_{off})	Smartphone, Tablet	command	api.amazon.com
	Gosund WiFi Smart Socket	on or off (GS_{onoff})	Smartphone, Tablet	command	device-provisioning.googleapis.com
	TP-Link Plug	on (TP_{on})	Smartphone, Tablet	command	api.tplinkra.com
		off (TP_{off})	Smartphone, Tablet	command	api.tplinkra.com
	WeMo Plug	on or off (WP_{onoff})	Smartphone, Tablet	command	appapis.xwemo.com

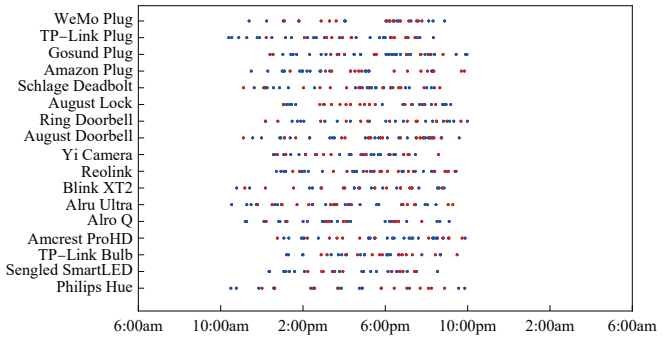

Fig. 6. Device events triggered by the controller locally and remotely over the 2 weeks that mapped to a 24-hour window. The blue points indicate the device events triggered locally while the red points indicate the device events triggered remotely.

D. Performance Evaluation of Home Safety Monitoring

The spatial information of device events extracted by IoTDuet enables us to detect potential anomalies that are critical to the safety of smart home users. We systematically evaluated the performance of our proposed Algorithm 2 and Algorithm 3 in detecting abnormal device events and home entrance activities using the data collected in our real-world smart home testbed.

1) Abnormal Device Event Detection: We detect the abnormal device events using the spatial information of each device event extracted with IoTDuet by setting alarms to the device events that are triggered remotely while the controlling devices are connected locally to the home network. To evaluate the performance of Algorithm 2 on extracting abnormal device events, we trigger each device event normally while synthetically injecting device events that are triggered remotely when the controlling device is connected to the home network.

TABLE II
HOME ENTRANCE ACTIVITIES WITH LABELS AND THEIR CORRESPONDING DEVICE EVENT SEQUENCE SIGNATURES.

Entrance Point	Entrance Activity	Device Event Sequence	Label
Window	A person or an object outside the house passes by the window	AU_{mot}	time-dependent
	A person opens the window from the inside	KM_{mot}, SC_{open}	potentially normal
	A person closes the window from the inside	KM_{mot}, SC_{close}	potentially normal
	A person opens the window from the outside	AU_{mot}, SC_{open}	potentially abnormal
	A person closes the window from the outside	AU_{mot}, SC_{close}	potentially abnormal
	A person breaks into the house through the closed window from outside and leaves the window open	$AU_{mot}, SC_{open}, KM_{mot}$	potentially abnormal
	A person breaks into the house through the closed window from outside and closes the window	$AU_{mot}, SC_{open}, KM_{mot}, SC_{close}$	potentially abnormal
	A person breaks into the house through the opened window from outside and leaves the window open	AU_{mot}, KM_{mot}	potentially abnormal
	A person breaks into the house through the opened window from outside and closes the window	$AU_{mot}, KM_{mot}, SC_{close}$	potentially abnormal
	A person breaks out the house through the closed window from inside and leaves the window open	$KM_{mot}, SC_{open}, AU_{mot}$	potentially abnormal
	A person breaks out the house through the closed window from inside and closes the window	$KM_{mot}, SC_{open}, AU_{mot}, SC_{close}$	potentially abnormal
	A person breaks out the house through the opened window from inside and leaves the window open	KM_{mot}, AU_{mot}	potentially abnormal
	A person breaks out the house through the opened window from inside and closes the window	$KM_{mot}, AU_{mot}, SC_{close}$	potentially abnormal
Back Door	A person enters the house through the back door by manually unlocking a locked door and locks the door manually	$SD_{mlk}, TC2_{open}, RC_{mot}, TC2_{close}, SD_{mlk}$	potentially normal
	A person enters the house through the back door by manually unlocking a locked door and locks the door via WiFi locally	$SD_{mlk}, TC2_{open}, RC_{mot}, TC2_{close}, SD_{wlk}.local$	potentially normal
	A person enters the house through the back door by manually unlocking a locked door and locks the door via WiFi remotely	$SD_{mlk}, TC2_{open}, RC_{mot}, TC2_{close}, SD_{wlk}.remote$	potentially abnormal
	A person enters the house through the back door by manually unlocking a locked door and leaves the door automatically locked	$SD_{mlk}, TC2_{open}, RC_{mot}, TC2_{close}, SD_{alk}$	potentially normal
	A person enters the house through the back door by manually unlocking a locked door and leaves the door unlocked	$SD_{mlk}, TC2_{open}, RC_{mot}$	potentially abnormal
	A person enters the house through the back door by unlocking a locked door via WiFi locally and locks the door manually	$SD_{wlk}.local, TC2_{open}, RC_{mot}, TC2_{close}, SD_{mlk}$	potentially normal
	4 entrance activities by unlocking a locked back door via WiFi locally with different ways of closing the backdoor are omitted due to page limit	...	...
	A person enters the house through the back door by unlocking a locked door via WiFi remotely and locks the door manually	$SD_{wlk}.remote, TC2_{open}, RC_{mot}, TC2_{close}, SD_{mlk}$	potentially abnormal
	4 entrance activities by unlocking a locked back door via WiFi remotely with different ways of closing the backdoor are omitted due to page limit	...	...
	A person exits the house through the back door by unlocking a locked door manually and leaves the door unlocked	$RC_{mot}, SD_{mlk}, TC2_{open}$	potentially abnormal
	14 exiting activities with different ways of opening and closing the back door omitted due to page limit	...	...
Front Door	A person or an object outside the house passes by the front door	RD_{mot}	time-dependent
	A person outside the house rings the doorbell and then leaves	RD_{mot}, RD_{ring}	time-dependent
	A person outside the house rings the doorbell and gets inside with door unlocked manually and leaves the door opened	$RD_{mot}, RD_{ring}, AL_{mlk}, TC1_{open}, AQ_{mot}$	potentially abnormal
	14 entrance activities by ringing the doorbell first with different ways of opening and closing the front door omitted due to page limit	...	...
	A person enters the house through the front door by manually unlocking a locked door and locks the door manually	$RD_{mot}, AL_{mlk}, TC1_{open}, AQ_{mot}, TC1_{close}, AL_{mlk}$	potentially normal
	14 entrance activities of a person with lock control with different ways of opening and closing the front door omitted due to page limit	...	...
	A person exits the house through the front door by unlocking a locked door manually and leaves the door unlocked	$AQ_{mot}, AL_{mlk}, TC1_{open}, RD_{mot}$	potentially abnormal
	14 exiting activities of a person with lock control with different ways of opening and closing the front door omitted due to page limit	...	...

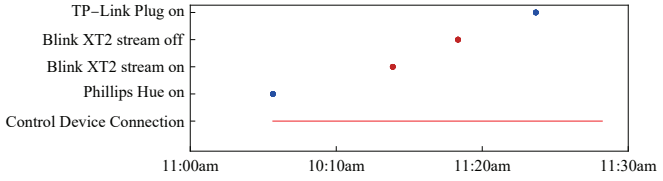

Fig. 7. An example of detected abnormal Blink XT2 stream on and off device events which are triggered remotely (marked in red) when the controlling device is connected to the home network.

Fig. 7 illustrates an example of detected abnormal Blink XT2 stream on and off device events. These two device events are triggered remotely and are marked in red in Fig. 7. We also noticed Phillips Hue on event and TP-Link Plug on event which are triggered locally and the controlling device is connected to the home network during the time that Blink XT2 stream on and off device events are triggered remotely. So, we raise an alarm for these two device events to notify the smart home user that the Blink XT2 may be compromised for generating abnormal device events. Our evaluation on the data collected at the smart home with synthetic abnormal data shows that our framework can correctly identify abnormal device events with suspicious spatial patterns which are triggered remotely while the controlling device still connects to the home network.

2) Home Entrance Monitoring: In our smart home testbed, we identified 3 critical entrance points including the window, front door, and back door. For each of the 3 entrance points,

we further enumerated all possible entrance activities in order to comprehensively monitor the home entrance safety. Table II lists all 90 home entrance activities that could happen at our smart home. Different from IoTMosaic [35], which considers only the common user activities, we study all kinds of home entrance activities that could happen including intrusive activities and abnormal activities. For example, we not only consider the opening window from the inside activity of normal users, but also include the activity that a person breaks into the house through the window from the outside, as listed in Table II. In addition, we also take the spatial information of each device event into consideration when identifying the entrance activities and treat the device events with same name but different spatial information as unique when building up the signatures.

As shown in Table II, we categorize each entrance activity as 'potentially normal', 'potentially abnormal', and 'time-dependent' based on whether the entrance activity is considered to be different from smart home users' normal behaviors. If an entrance activity is performed by legitimate users for most of the time, we consider it as 'potentially normal'. If an entrance activity is intrusive or not common, we consider it as 'potentially abnormal'. If an entrance activity is suspicious during a particular time period, i.e., midnight, we label it as 'time-dependent' because we need the timestamp information of the activity to help us make the decision.

We applied the signature extraction and approximate matching algorithms proposed by IoTMosaic [35] with k set as 0 on the dataset collected in our smart home testbed where each home entrance activity was repeated for at least 10 times over 2 weeks. For each matched entrance activity, we first check if its name corresponds to a 'potentially abnormal' activity and then check the timestamp to see if it is 'time-dependent' and it happens during midnight following Algorithm 3. For both of the above cases, we will mark the home entrance activity as a potential anomaly and report it to the users. Our experiments of applying Algorithm 3 on the dataset which contains all 90 different kinds of entrance activities confirm that our system can effectively identify intrusive and abnormal home entrance activities for home safety monitoring.

VII. Related Work

The raising security and privacy threats of smart home IoT devices have drawn great research attention [1, 4–8, 10, 17, 19, 22, 23, 25, 26, 37, 38]. Most of the existing work focuses on either detecting abnormal flows of known attacks such as port scanning and DoS attacks at the network traffic level [6, 22, 36] or applying the domain knowledge such as smart home layout and IoT devices' deployment location for anomaly detection at the device event level with the assumption that the event logs of all IoT devices are available [5, 7, 8, 10]. Some researchers have been working on inferring the device event information from the home network traffic by utilizing the fact that IoT devices generate unique signatures when device events are triggered [1, 32, 34]. However, little effort has been devoted to extracting the controller spatial information from the home network traffic.

IoTArgos [36] designs a multi-layer anomaly detection framework which adopts a two-phase technique where both supervised learning and unsupervised anomaly detection are applied to detect known and unknown attacks at different network layers. DÏoT [22] proposes a federated learning model for detecting abnormal network traffic. Survey paper [6] reviews the literature work in network traffic level anomaly detection using machine learning. These solutions achieve high accuracy and efficiency in detecting cyber attacks such as port scanning, brute forcing credential, and DoS attacks, but fail to work if the IoT devices have already been compromised by attackers.

HAWatcher [10] is a semantics-aware anomaly detection system for appified smart homes by generating and enforcing hypothetical correlations based on semantic information. The work in [7] identifies device event constraints in physical channels. Peeves [5] verifies the validity of a device event based on the state and sensory data of nearby IoT devices. However, all of them rely on the strong assumption that the complete device event logs or device state information are always available.

Pingpong [32] proposes to extract the IoT device event information from the home network traffic by matching the unique packet-level signature of each device event using the state transmission machine. IoTAthena [34] further identifies the inter-packet time interval as an important feature in generating more accurate signatures and designs a time-sensitive signature matching algorithm for device event extraction. However, both of them do not further study the spatial attribute of the extracted device events.

IoTDuet, to the best of our knowledge, is the first effort in studying the spatial information of IoT device events from the home network traffic. IoTDuet is able to determine whether a device event is triggered locally and remotely by monitoring the traffic of all devices connected to the home network. Such spatial information is crucial for understanding the behavior of the smart home ecosystem and home safety monitoring.

VIII. Conclusion and Future Work

In this paper, we propose a system named IoTDuet for extracting spatial information of IoT device events about whether they are triggered by controlling devices connected to the home network. We explore the extracted spatial information of each IoT device event for applications of abnormal device event detection and home entrance safety monitoring. We extensively evaluated the performance of IoTDuet and applications of home safety monitoring on a real-world smart home testbed. We plan to implement our system in a large number of smart homes and other IoT environments for spatial information extraction and safety monitoring. Our future work is centered on exploring more applications of utilizing the spatial information of device events for protecting the security and privacy of smart homes.

References

[1] A. Acar, H. Fereidooni, T. Abera, A. K. Sikder, M. Miettinen, H. Aksu, M. Conti, A.-R. Sadeghi, and S. Uluagac, "Peek-a-Boo: I See Your Smart Home Activities, Even Encrypted!" in *Proc. of ACM WiSec*, 2020.

[2] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "SoK: Security Evaluation of Home-Based IoT Deployments," in *Proc. of IEEE S&P*, 2019.

[3] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai Botnet," in *Proc. of USENIX Security*, 2017.

[4] V. Bhosale, L. De Carli, and I. Ray, "Detection of Anomalous User Activity for Home IoT Devices," in *Proc. of IoTBDS*, 2021.

[5] S. Birnbach, S. Eberz, and I. Martinovic, "Peeves: Physical Event Verification in Smart Homes," in *Proc. of ACM CCS*, 2019.

[6] A. A. Cook, G. Mısırlı, and Z. Fan, "Anomaly Detection for IoT Time-Series Data: A Survey," *IEEE Internet of Things Journal*, vol. 7, no. 7, 2020.

[7] W. Ding and H. Hu, "On the Safety of IoT Device Physical Interaction Control," in *Proc. of ACM CCS*, 2018.

[8] W. Ding, H. Hu, and L. Cheng, "IOTSAFE: Enforcing Safety and Security Policy with Real IoT Physical Interaction Discovery," in *Proc. of NDSS*, 2021.

[9] E. Fernandes, J. Jung, and A. Prakash, "Security Analysis of Emerging Smart Home Applications," in *Proc. of IEEE S&P*, 2016.

[10] C. Fu, Q. Zeng, and X. Du, "HAWatcher: Semantics-Aware Anomaly Detection for Appified Smart Homes," in *Proc. of USENIX Security*, 2021.

[11] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet," in *Proc. of NDSS*, 2019.

[12] G. Ho, D. Leung, P. Mishra, A. Hosseini, D. Song, and D. Wagner, "Smart Locks: Lessons for Securing Commodity Internet of Things Devices," in *Proc. of ACM ASIACCS*, 2016.

[13] S. Hui, H. Wang, D. Xu, J. Wu, Y. Li, and D. Jin, "Distinguishing Between Smartphones and IoT Devices via Network Traffic," *IEEE Internet of Things Journal*, vol. 9, no. 2, 2022.

[14] Y. Jia, Y. Xiao, J. Yu, X. Cheng, Z. Liang, and Z. Wan, "A Novel Graph-based Mechanism for Identifying Traffic Vulnerabilities in Smart Home IoT," in *Proc. of IEEE INFOCOM*, 2018.

[15] Y. Jia, L. Xing, Y. Mao, D. Zhao, X. Wang, S. Zhao, and Y. Zhang, "Burglars' IoT Paradise: Understanding and Mitigating Security Risks of General Messaging Protocols on IoT Clouds," in *Proc. of IEEE S&P*, 2020.

[16] G. Kambourakis, C. Kolias, and A. Stavrou, "The Mirai botnet and the IoT Zombie Armies," in *Proc. of IEEE MILCOM*, 2017.

[17] D. Kumar, K. Shen, B. Case, D. Garg, G. Alperovich, D. Kuznetsov, R. Gupta, and Z. Durumeric, "All Things Considered: An Analysis of IoT Devices on Home Networks," in *Proc. of USENIX Security*, 2019.

[18] H. Liu, T. Spink, and P. Patras, "Uncovering Security Vulnerabilities in the Belkin WeMo Home Automation Ecosystem," in *Proc. of IEEE PerCom Workshops*, 2019.

[19] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. O. Tippenhauer, and Y. Elovici, "ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis," in *Proc. of ACM SAC*, 2017.

[20] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "IoT Sentinel: Automated Device-Type Identification for Security Enforcement in IoT," in *Proc. of IEEE ICDCS*, 2017.

[21] Milan, Fránik and Miloš, Čermák, "Serious Flaws Found in Multiple Smart Home Hubs: Is Your Device among Them?" https://www.welivesecurity.com/2020/04/22/serious-flaws-smart-home-hubs-is-your-device-among-them/.

[22] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A Federated Self-learning Anomaly Detection System for IoT," in *Proc. of IEEE ICDCS*, 2019.

[23] T. OConnor, R. Mohamed, M. Miettinen, W. Enck, B. Reaves, and A.-R. Sadeghi, "HomeSnitch: Behavior Transparency and Control for Smart Home IoT Devicess," in *Proc. of ACM WiSec*, 2019.

[24] Q. Qin, K. Poularakis, and L. Tassiulas, "A Learning Approach with Programmable Data Plane towards IoT Security," in *Proc. of IEEE ICDCS*, 2020.

[25] P. Rashidi, D. J. Cook, L. B. Holder, and M. Schmitter-Edgecombe, "Discovering Activities to Recognize and Rrack in a Smart Environment," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 4, pp. 527–539, 2011.

[26] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi, "Information Exposure From Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach," in *Proc. of ACM IMC*, 2019.

[27] E. Ronen, C. O'Flynn, A. Shamir, and A.-O. Weingarten, "IoT Goes Nuclear: Creating a ZigBee Chain Reaction," in *Proc. of IEEE S&P*, 2017.

[28] E. Ronen and A. Shamir, "Extended Functionality Attacks on IoT Devices: The Case of Smart Lights," in *Proc. of IEEE EuroS&P*, 2016.

[29] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, 2018.

[30] V. Sivaraman, D. Chan, D. Earl, and R. Boreli, "Smart-Phones Attacking Smart-Homes," in *Proc. of ACM WiSec*, 2016.

[31] SmartHomeDB, "Smart Home DB - The Smart Home Database," https://www.smarthomedb.com/.

[32] R. Trimananda, J. Varmarken, A. Markopoulou, and B. Demsky, "PingPong: Packet-Level Signatures for Smart Home Device Events," in *Proc. of NDSS*, 2019.

[33] Y. Wan, K. Xu, F. Wang, and G. Xue, "Characterizing and Mining Traffic Patterns of IoT Devices in Edge Networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 89–101, 2021.

[34] Y. Wan, K. Xu, F. Wang, and G. Xue, "IoTAthena: Unveiling IoT Device Activities from Network Traffic," *IEEE Transactions on Wireless Communications*, vol. 21, no. 1, January 2022.

[35] Y. Wan, K. Xu, F. Wang, and G. Xue, "IoTMosaic: Inferring User Activities from IoT Network Traffic in Smart Homes," in *Proc. of IEEE INFOCOM*, 2022.

[36] Y. Wan, K. Xu, G. Xue, and F. Wang, "IoTArgos: A Multi-Layer Security Monitoring System for Internet-of-Things in Smart Homes," in *Proc. of IEEE INFOCOM*, 2020.

[37] Q. Wang, W. Hassan, A. Bates, and C. Gunter, "Fear and Logging in the Internet of Things," in *Proc. of NDSS*, 2018.

[38] W. Zhang, Y. Meng, Y. Liu, X. Zhang, Y. Zhang, and H. Zhu, "HoMonit: Monitoring Smart Home Apps from Encrypted Traffic," in *Proc. of ACM CCS*, 2018.

[39] T. Zillner, "ZigBee Exploited: The good, the Bad and the Ugly," in *Proc. of the DeepSec Conferences in Depth Security*, 2017.