Augmented Reality's Potential for Identifying and Mitigating Home Privacy Leaks

Stefany Cruz[1], Logan Danek[1], Shinan Liu[2], Christopher Kraemer[6], Zixin Wang[3]
Nick Feamster[2], Danny Yuxing Huang[4], Yaxing Yao[5], Josiah Hester[6]
[1]Northwestern University, [2]University of Chicago, [3]Zhejiang University
[4]New York University, [5]University of Maryland, Baltimore County, [6]Georgia Institute of Technology

***Abstract*—Users face various privacy risks in smart homes, yet there are limited ways for them to learn about the details of such risks, such as the data practices of smart home devices and their data flow. In this paper, we present Privacy Plumber, a system that enables a user to inspect and explore the privacy "leaks" in their home using an augmented reality tool. Privacy Plumber allows the user to learn and understand the volume of data leaving the home and how that data may affect a user's privacy—*in the same physical context* as the devices in question, because we visualize the privacy leaks with augmented reality. Privacy Plumber uses ARP spoofing to gather aggregate network traffic information and presents it through an overlay on top of the device in an smartphone app. The increased transparency aims to help the user make privacy decisions and mend potential privacy leaks, such as instruct Privacy Plumber on what devices to block, on what schedule (i.e., turn off Alexa when sleeping), etc. Our initial user study with six participants demonstrates participants' increased awareness of privacy leaks in smart devices, which further contributes to their privacy decisions (e.g., which devices to block).**

I. Introduction

The increasing adoption of Internet-connected smart devices has brought huge improvements to our lives. Yet, these devices also raise significant privacy concerns from their users, such as sensitive data collection [53], [51], data sharing [51], and data misuse [22], [23], [27]. Literature has suggested many types of privacy risks associated with smart devices. For example, some seemingly innocent data, such as the network traffic shapes and patterns of smart devices, may reveal sensitive personal information, such as users' daily schedule, their gender, date of birth, social security number, location, and behaviors [5], [3].

However, many risks are not obvious to users due to the opaque nature of the data practices of smart devices; the average users lack an understanding of how their data is collected, processed, and shared [51], [50], [21]. Prior research has proposed various ways to increase users' awareness of the data practices in smart homes, such as data dashboards, mobile phone apps, ambient light and sounds, and so on [44], [15], [9], [16]. Some other mechanisms (e.g., IoT Inspector [15]) focus on specific aspects of the data practices and present network traffic data to users so that they can access first-hand data of the data flow in/out of smart devices. Yet, most mechanisms we know decouple such transparency from the device themselves—i.e., users need to learn about the data practices separately from the smart devices—making the information less intuitive to consume, especially for the average user. In addition, these mechanisms do not provide users with the ability to take action if they notice unexpected data practices (e.g., blocking the data from being sent out to third parties).

In this paper, we focus on the data flow in and out of smart devices. We build a proof-of-concept smartphone-based augmented reality system called **Privacy Plumber** to increase users' awareness of the data flows of smart devices and provide them with controls to block certain data flow if needed. We focus on data flow rather than other aspects of data practices (e.g., types of data being collected) mostly due to practicality and feasibility reason, as we can reasonably capture data flow and identify its source and destination using ARP spoofing [15]. In addition, from the smart devices' perspective, these devices have multiple tiers of software, all of which entail some type of tracking. Such tracking is generally embodied in the data flow. We use augmented reality to visualize data flows *in the same physical environment* as the devices in question; this method could potentially help users establish a connection between the devices and their data flows in the same context. Users' proper understanding of data flow may help them understand the privacy implications of devices such as smart TVs [28], voice assistants [15], children's toys [10], security cameras [24], [35], and smart light bulbs [8].

The development of Privacy Plumber is inspired by the following three gaps in the literature. First, the data flows of smart devices are opaque and not visible to users. Second, existing tools to monitor network traffic of smart devices (e.g., IoT Inspector [15], open.Dash [9]) require a certain level of technical knowledge to be able to interpret the results—not to mention that the results are often decoupled from the physical environment where the smart devices are situated. Oftentimes, the results are presented on, for instance, dashboards on computers or phones, where there is a disconnection between the visualization of data flows and the smart devices that create the data flow. Third, existing tools or mechanisms do not provide users with the ability to control unnecessary or unexpected data flows. With Privacy Plumber, we aim to bridge the gaps and increase users' awareness and control of the data flow in smart devices.

Symposium on Usable Security and Privacy (USEC) 2023
27 February 2023, San Diego, CA, USA
ISBN 1-891562-91-6
https://dx.doi.org/10.14722/usec.2023.238930
www.ndss-symposium.org, https://www.usablesecurity.net/USEC/

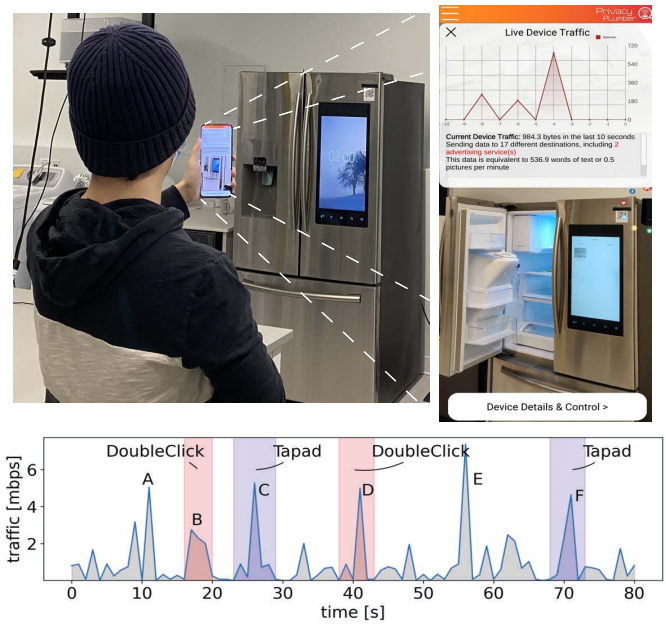

Fig. 1: Privacy Plumber lets a user find and mitigate potential privacy violations in the smart home. The figure shows a user walking around the smart home and inspecting the traffic and trackers coming out of a Samsung Smart Fridge using the Augmented Reality enabled app. Furthermore, (not shown in the picture above) users can use built-in, infrastructure-free controls to limit traffic of devices to times of day—without requiring any additional hardware or modifications to the network. The graph shows the actual network traffic as the user interacted with the Smart Fridge: A: turning on the ice maker; B: browsing recipe; C: browsing goods; D: interacting with the Bixby voice assistant of the fridge; E: opening the fridge door; F: adding items to the shopping list. During these interactions, the Smart Fridge communicated with various advertising and tracking services, such as DoubleClick and Tapad.

Privacy Plumber uses augmented reality (AR) techniques and visualizes real-time network traffic flowing in and out of smart devices through an overlay. It allows users to find potential privacy leaks in their homes by pointing the AR-based app at smart devices. As shown in Figure 1, the app adds an overlay on top of the smart devices in which it displays a real-time data flow based on the network traffic with the necessary information for users to understand it. We chose to use AR because, as privacy is highly contextual [32], it can provide strong contextual connections between the actual real-time privacy leaks, and the user actions (or inaction). This allows the smartphone to function as a viewfinder into the invisible world of data flow and identify potential privacy violations. The smartphone application relies on a companion software tool hosted on a laptop or desktop on the same home network. This tool discovers smart devices in a user's home, intercepts their traffic via ARP spoofing [48], and analyzes the data flow (e.g., what traffic is leaving the home over time)—without requiring the user to modify their network settings

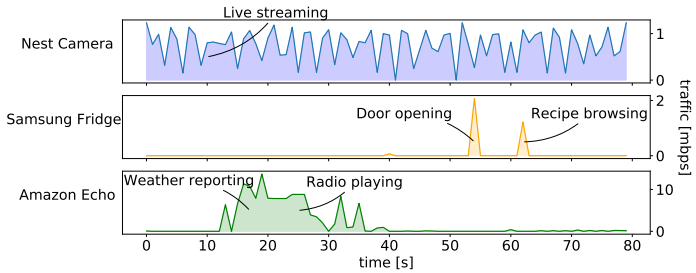

Fig. 2: Outbound network traffic from various smart home IoT devices: a Nest Camera, an Amazon Echo, and a Samsung Smart Fridge. Traffic increases or provides a fingerprint for many types of seemingly benign actions, creating a privacy leak. Current systems do not provide real-time context or ability to experiment with these devices, nor control their leakage.

or install additional hardware. When users would like to take action and block certain data flow, ARP-spoofing is used again to jam specific devices' traffic (thereby blocking the device) at the time of day set by the user.

We build a proof-of-concept prototype and conducted a pilot study with 6 participants in our lab to collect their feedback on the prototype. Our initial findings have suggested that Privacy Plumber helped participants understand the network traffic, increased their awareness of potential privacy violations, and helped them make more informed decisions on how to handle IoT devices.

This paper makes three contributions. First, to the best of our knowledge, Privacy Plumber is the first mechanism that provides users with real-time information on the data flow of their smart devices. This paper proves the possibility of using AR-based technology as a viable option to increase users' awareness of the data flows of smart devices. Second, our initial evaluation shows promising results, indicating users' potential acceptance of these technologies. Third, we summarized lessons learned from the pilot user study to inform the design and development of future systems that aim to improve users' awareness of data practices in smart homes.

II. Background and Related Work

In this section we discuss related work seeking to understand or discover privacy leaks, and the tools that exist to help users understand and mitigate them. Privacy Plumber is meant to to provide a handheld and zero-cost inspection and experimentation tool for privacy leaks of nearby smart devices in the home, and a straightforward and low burden method for mitigating those leaks.

A. Privacy Issues in Smart Home

Over the decades, privacy issues have been deeply disclosed in smart home, such as transparency of data collection, data sharing, and accessibility [20], [50], [49], [16], [53], [30], [50]. Some smart home devices have always-on sensors that capture users' offline activities in their homes and transmit relevant information outside of the home, especially for cloud services run by device manufacturers [6].

In the meantime, users are concerned about leaks of sensitive information [23], [51], [25], such as visual and auditory information which they see as private [23], [25]. Thus, users have a strong desire to protect themselves against such recordings being accessed without their permission [30], [19]. However, some information users perceived as not very sensitive also lead to privacy leaks. For example, the home temperature could be used to determine whether a house is occupied or not, as a precursor to burglary [20].

In fact, smart devices give off digital exhaust which can be used by third parties including a user's Internet Service Provider, advertisers, device manufacturers, and others, to fingerprint activities and get sensitive information. Shown in Figure 2 is the network traffic and trackers of various smart home devices.This network traffic forms the basis of most leaks.

B. Tools for Enhancing Smart Home Privacy

Most related to Privacy Plumber are tools that watch or monitor network traffic in the home and provide something of use to the user, whether visualization and information, education, or a mechanism for control.

Sophisticated, technically literate users can use systems that block advertising and tracking domains (e.g., PiHole [38] and pfSense [34]), but these methods are bespoke and often require additional or dedicated hardware (e.g., Raspberry Pi for Pi-Hole, and a supporting custom router for pfSense). Other tools have provided insight into what might be exposed from web-browsing activities, including WiFi privacy Ticker [11], but do not consider or scale to the new problems of connected devices with physical sensors and abilities in a space. Aretha [40] explores this tool space and proposed (but did not deploy) a simple firewall-based control mechanism. Aretha presents data in aggregate instead of contextually and in real-time. None of these techniques investigate a range of IoT devices, usually constrained by studies with participants in their own homes, in a time when smart home adoption is low (Aretha had three participants, and only one had more than a phone, tablet, and Alexa). None of these techniques develop a scalable (no additional hardware required) way to interpret privacy leaks and control them. Emerging smart devices are highly contextual and location sensitive, an Alexa in the bedroom versus the kitchen has different privacy exposure (i.e. the former gives sleep times, the latter exposes eating habits). Moreover, tracking these devices' privacy exposures presents a technical challenge because the traffic is not centralized through a web browser or laptop. A tool is needed to visualize privacy leaks from smart devices in real-time and in context, educate users on the consequences of these leaks, and provide control mechanisms for partially mitigating these leaks.

Wifi Privacy Ticker [11] demonstrated a first method for improving the awareness of users in terms of privacy by providing a count of the amount of sensitive data that was being transmitted unencrypted over the network awareness. By seeing this in real-time, users could adjust their behavior or find encrypted means to browse the web. Of course, this ticker was developed well before the current generation of smart devices, however the underlying concept of surfacing the invisible privacy leaks remains the same for Privacy Plumber, but for smart devices. Xray-refine [46], [47] provided smart phone users a means to visualize their exposure profile, based on the duration of app use. This method was an educational solution, but users had to adjust behavior to work around the constraints of the apps they were using, in some cases, opting out of apps to reduce privacy exposure.

Finally, recent work like Aretha [40], PriView [36], Lumos [41], and IoT Inspector [15] look at making usable visualizations and mechanisms to understand and interpret data coming from smart devices in the home. IoT Inspector is a simple-to-install desktop application that uses ARP spoofing to gather network traffic on the Wifi network of the desktop/laptop. This information is sent and collated at a server, and then viewed online by the user, listing different trackers and websites that are attached to smart device usage. Because of the ease of installation and no extra hardware requirement, IoT Inspector was deployed by thousands of users.

In comparison, Aretha is a part research tool, part exploratory users tool for exploring a design space of privacy tools and controls. Aretha helps users become aware of the network traffic flows in their homes while also educating users to regain their privacy in the connected home. Aretha suggests the use of firewall mechanisms controllable by the user, but does not implement them. Aretha, owing to a hardware requirement (a device must be attached to the Wifi router in the home) was only deployed in three homes, compared to the massive scale deployment of IoT Inspector. Similarly, PriView also has a hardware requirement; its users need to have dedicated external thermal cameras (e.g., FLIR One [36]) attached to their phones. For Lumos, there is no special hardware environment, although the focus is more on identifying hidden smart devices rather than analyzing the network traffic for privacy leaks.

Privacy Plumber is not meant as a research tool or a design space exploration tool. It is meant as an actual, real world system with a focus on scalability and ease of deployment in any home, similar to IoT inspector. Unlike both IoT inspector and Aretha, Privacy Plumber provides *real-time and contextual visualizations of privacy leaks*, real-time ability to plug those leaks (as well as automated rule setting for plugging leaks), and enables experimentation in real-time.

Finally, other significant measurement campaigns on in-home traffic have been conducted, focusing on the Wifi network itself or devices in the home [39], [18]. These have usually been for research purposes and need finding and are useful for informing the design of Privacy Plumber, but are not necessarily tools for controlling smart home device privacy.

C. Determining Home Activities from Network Traffic

Complementary to Privacy Plumber are other works which demonstrate the ability to infer activities from network traffic: whether on a phone, smart device, or laptop [4]. By analyzing the patterns of network traffic in the home, occupancy, habits such as sleeping, watching TV, listening to music, and sometimes preferences, can all be determined. *HomeSnitch* [33], *Peek-a-boo* [1], and *HoMonit* [52] all utilize machine learning with varying degrees of success to identify activities in the home from network traffic. Other tools utilized for monitoring Internet connected smart devices in the home, IoT Sentinel [26] and IoT Sense [7], have shown that particular devices can be

fingerprinted by their traffic patterns. Enabling another way for an ISP or third party to determine the activity in the home. Each of these systems and methods are complementary with Privacy Plumber; inferred activities from traffic would be useful to surface in Privacy Plumber for the user to understand privacy exposure and know when to mitigate it, and device fingerprinting provides a way for zero-registration or setup of Privacy Plumber in a home.

D. Challenges: Contextual, Real-time Privacy Understanding and Control in the Home

Despite the diverse work in the smart home privacy space, significant gaps and challenges remain, which we detail below.

C1: Users can't model what devices are doing, especially without context. With tools like IoT Inspector, a user might be able to count the number of trackers and advertisers contacted in a day from the sum of their interactions with smart devices. But how can a user know that turning on the NPR podcast on their smart fridge will send thousands of bytes of information to Bloomberg News for advertising purposes? How can they know that turning on the device sends a short burst of traffic? Users know that data captured will often be used for advertising, which often generates an adverse reaction [45]. However, with smart devices, it is not always clear what actions or contexts trigger data being transmitted [13]. Things like Privacy labels for websites and smart devices are meant to give a method for scoring devices privacy [42], [17]. However, these are static representations of the privacy exposure of a device. With tools like IoT inspector and Aretha, *aggregate* views of data are seen (as opposed to *real-time* views), not associated with very fine user actions: like the turn on the light, say command to Alexa, or open the fridge door. Because of this granularity, the mental models of what devices are doing, and what they are sharing, are very perplexing. Privacy tools must address this lack of action mapping to network traffic, enabling contextual integrity [32] in real-time.

C2: Users don't have intuitive methods to control the privacy "valve". Users want devices that provide helpful features, but they do not know the cost of this ease. One option is to just unplug the device; however, this is all or nothing. Users need a way to valve the privacy flow to something they are comfortable with, or to at least be able to analyze the tradeoffs [43]. Making privacy more "tangible" [2] is one way this can be done; where the privacy leaks are more visceral. Selective firewalls (such as pfSense [34]), or other more fine grained network mechanisms may provide a means to control the privacy valve, but this must be intuitive and understandable to the user, and they must be able to actually "see" the effect of turning this valve.

C3: Smart devices are context (location, time, action) dependent. Smart devices are necessarily scattered around the home; and this will continue as more devices become intelligent, and more applications are explored. Watching a desktop or laptop traffic meter and figuring out which device in which room is doing what at which times, is mentally trying for the user and disassociates the device from the physical space that defines its context and use. Just like when trying to find leaks in pipes, physical proximity is required. Handheld inspection tools provide mobility, and enable in-situ fixing and experimentation.

C4: Users can't experiment. Indeed, because of contextual changes in how private information is leaked, experimentation is difficult with existing tools that generally provide traffic summaries. Interactions with smart devices can last only a few seconds. Enabling a user to experiment with different actions and uses of a smart device, and then see the associated network traffic in real-time, would provide a powerful way to build a mental model. However, providing an ability to experiment is challenging with the current suite of tools.

C5: Technical challenge of scalability and deployment. If a privacy tool is to be useful and translate to the general public, it must be hardware free, or at least trivially easy to deploy to enable scalability and broad adoption. Commercial products like fing.com embed all functionality in a single phone application. Large scale deployments like with IoT Inspector are enabled through a desktop application that is easy to install. However, these methods do not provide controls since that is technically difficult to do without custom hardware put between the Wifi endpoint and the user. On the other hand, hardware requirements or custom install procedures reduce the deployment size of tools like Aretha, or narrow the user base by requiring technical ability, as with PiHole. It is not clear how to implement mechanisms of control without changing the Wifi network and infrastructure. To create scalable, user-centered, novice friendly privacy tools, mechanisms for enabling control of smart device traffic without hardware intervention must be developed.

III. System Design

We present Privacy Plumber as a proof-of-concept and end-to-end system to address the challenges listed of scalable and general population serving privacy tools for emerging smart homes. Privacy Plumber is inspired by various handheld tools for identifying and fixing faults in large and complex systems. For example, acoustic leak finding has been used for decades to localize leaks in gas and water pipelines. Handheld oscilloscopes, multimeters, and RF Spectrum Analyzers have helped engineers debug problems in large electrical systems. These handheld devices make the invisible signals visible and interactive. They allow real-time experimentation and debugging. Inspired by these devices, Privacy Plumber is designed to offer a general user a level of insight and control of the invisible privacy leaks that are rampant in Internet-connected smart devices in the home. Privacy Plumber is composed of two pieces as shown in Figure 3:

(1) the **IoT Network Analyzer**, a desktop application which collects real-time data on smart devices on the shared Wifi network, and provides an infrastructure and hardware free mechanism to block arbitrary devices traffic, and;

(2) the **Privacy Plumber** phone application, which serves as a viewfinder or inspector for any smart devices in view, and presents data from the desktop application, including device network traffic and potential privacy leaks to the users, along with educational content matched to what is known about the device, all in real time.

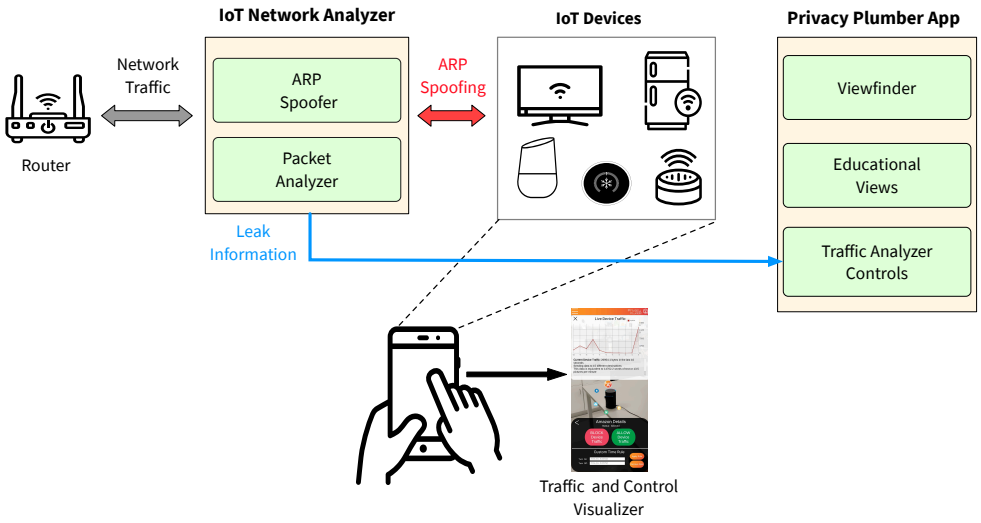

Fig. 3: System diagram of Privacy Plumber including the IoT Network Analyzer and Privacy Plumber mobile application. IoT Network Analyzer runs on a computer that is connected to a user's router. IoT Network Analyzer automatically discovers and captures IoT devices on the same network using ARP spoofing. Privacy Plumber connects with IoT Network Analyzer to present the network analysis in AR. The user can then examine their devices' network traffic and control when they want their devices to be on or off.

Overview of Usage. A user would first download, install, and run IoT Network Analyzer on their computer and the Privacy Plumber app on their mobile phone, such that both the computer and the phone are on the same local area network. Let us assume that the user is interested in inspecting a smart device like an Amazon Echo. While running the Privacy Plumber app, the user points the phone camera to Echo and speaks a voice command (e.g., "Alexa, what is the weather?") IoT Network Analyzer captures all network traffic between Echo and the Internet, analyzes the packets, and identifies destinations that are third-party advertising and tracking companies. The Privacy Plumber app extracts this information from IoT Network Analyzer and visualizes key statistics for the user—such as real-time bandwidth usage of the device and the number of advertising and tracking services contacted—as an overlay in the AR view.

When the user points the phone camera at a device, the Privacy Plumber app does not recognize devices with computer vision algorithms. Instead, for the purpose of this prototype, we print a QR code on each IoT device. The QR code includes the device's MAC address, its name, and the manufacturer. The app uses the phone's camera to scan for the QR code, identifies the device based on the QR code, and displays the device with a dial menu around it (see Figure 4a). The options in the menu allow the user to see the outbound traffic from the device as well as read a brief article stating what types of information the device may be tracking. The user may also use the Device Control menu (Figure 4c) to manually block or allow traffic from the device. Future versions of the app will use computer vision to recognize devices; see the discussion in Section V.

Privacy Threat Model and Assumptions. We assume that a user's privacy may be *potentially* violated if an IoT device exhibits either or both of the following behaviors. In **Threat 1**, an IoT device could contact an advertising and tracking service on the Internet. In **Threat 2**, an IoT device could be sending out network traffic to hosts on the Internet when the user does not expect any network activities—for example, when the user is not interacting with the device.

We design both the Privacy Plumber app and IoT Network Analyzer with this privacy threat model in mind. IoT Network Analyzer captures packets, analyzes the headers, identifies the destination hosts (based on the IP addresses, domain names, and the TLS Server Name Indication fields), and determines if a destination host is an advertising and tracking company. The Privacy Plumber app displays the number of advertising and tracking services (e.g., the red text below the graph in Figure 4b), thereby helping users toward identifying **Threat 1**. Based on the byte counters from IoT Network Analyzer, the Privacy Plumber app also shows a bandwidth graph that plots the bytes sent per second over time (e.g., the time-series graph in Figure 4b). This graph could help users correlate network activities with human interactions—or the lack thereof—with given IoT devices and thus identify possible instances of **Threat 2**. Note that IoT Network Analyzer does not parse the payload of packets to discover sensitive information within the traffic, as the network traffic is likely encrypted.

A. *Design Goals*

Privacy Plumber must make the underlying behavior of the devices in the home apparent, and enable forms of fine-grained (informed) control of the leakage of sensitive information for the user. Towards this end, and addressing the challenges described in Section II-D, we are guided by the following design goals.

(1) **Handheld and Mobile.** Smart devices are scattered throughout the home. Phone adoption is nearly universal. Using a phone as a window into the information world gives

context and a sense of place. The phone form factor increases the likelihood of adoption and allows for inspection on the go; users can trigger or interact with devices and easily watch the movement of data, instead of having to return to the desktop.

(2) **Real-time.** Seeing statistics after the fact, as in most systems, is not as impactful or helpful when developing a model of how devices operate. Moreover, real-time analysis enables experimentation, providing users with a mechanism for exploring limitless scenarios and quickly associating triggers with outcomes.

(3) **Infrastructure/Hardware Free.** Many other methods require custom hardware. This increases cost and raises the barrier to entry. We hope to enable anyone, especially those that may have limited autonomy over infrastructure (i.e. renters, low-resourced communities) to be able to inspect the devices put in their living space.

(4) **Intuitive Controls.** Complex mechanisms to control or limit the flow of privacy are not interpretable by users, and are possibly frustrating. Configuring a firewall is not a task most people would enjoy. Straightforward controls, with visible results, once those controls are put in place, are essential for users to trust the capability of the system.

(5) **Educational.** The ever-changing landscape of devices and the security/privacy arms is impossible to keep up with for privacy tools. Assisting users in understanding what makes certain devices leakier (e.g., always-on microphone) is essential.

To realize these design goals, we build the Privacy Plumber app—i.e., the handheld form factor—and IoT Network Analyzer as a two-part architecture working in tandem. Both systems must be running on the same local area network. IoT Network Analyzer, running on a computer, captures and analyzes network traffic between smart devices on the network and the Internet. IoT Network Analyzer acts as a server and provides the above information over an HTTP REST API. The Privacy Plumber app, acting as a client, regularly polls the REST API and presents the analysis as an AR overlay to users.

In the following sections, we detail the pieces of the system and how they interact to enable understanding and control of smart devices in the home. In Section III-B we discuss the IoT Network Analyzer and its role in capturing and curating privacy leak information; in Section III-D we describe the phone app design; in Section III-C we detail the mechanisms we use for controlling devices on a schedule, and finally, in Section III-E we describe a few ways to use Privacy Plumber.

B. Low Burden Home Network Traffic Capture

To use the Privacy Plumber app, the user must also have IoT Network Analyzer running on a computer (macOS, Windows, or Linux) that is on the same local area network as the phone. For our study, we run IoT Network Analyzer on a Raspberry Pi 3 Model B that is connected to the lab's network via Ethernet. We based IoT Network Analyzer's code on the open-source project, IoT Inspector [15], and made modifications according to our needs. In particular, whereas the original IoT Inspector constantly sends captured traffic's metadata to the researchers' servers, IoT Network Analyzer runs without the Internet; it processes the captured traffic locally and exposes the traffic via a REST API. Furthermore, whereas the original IoT Inspector runs on users' computers and visualizes the traffic in a browser-based dashboard, IoT Network Analyzer uses an AR-based app, Privacy Plumber, to visualize the network traffic; the mobile app reads the processed traffic through the abovementioned REST API and presents the results as an AR overlay.

Once running, IoT Network Analyzer automatically discovers IoT devices on the network, captures their network traffic via ARP spoofing, produces traffic statistics (e.g., bandwidth usage and identifying advertising and tracking services) over a local HTTP REST API, and blocks select devices (if desired by the user). We explain each of these features below.

Discovering IoT devices. Upon launch, IoT Network Analyzer automatically broadcasts ARP packets to the local area network and discovers active devices. To identify IoT devices, Huang *et al*. [15] describe an algorithm that infers the likely identities of IoT devices based on MAC OUI (i.e., Organizationally Unique Identifier, basically the first three octets of a MAC address), DNS, and UPnP messages. For the prototype in this paper, we only use the MAC OUI. Within the code of IoT Network Analyzer, we have already hard-coded the mapping between OUIs and names of five IoT devices in our lab (which we can find out beforehand). In this way, IoT Network Analyzer can instantaneously identify the IoT device in our lab without relying on the device identification algorithm in Huang *et al*. [15].

Capturing network traffic. Once IoT Network Analyzer identifies a known IoT device on the lab's network, it automatically starts intercepting network traffic between the device and the Internet via ARP spoofing, a technique used in the original IoT Inspector implementation and which incurs an overhead of 3.4 Kbps, given that we have five IoT devices in the lab [15].[1]

Obtaining traffic statistics. All traffic to and from IoT devices in our lab is redirected through IoT Network Analyzer. In doing so, IoT Network Analyzer is able to obtain statistics about the network traffic for every device, including the device's MAC address (from which to extract the OUI and determine the device's identity based on our hard-coded mapping); the number and size of packets (from which to infer the bandwidth usage); the remote IP addresses, DNS requests and responses, and the Server Name Indication field within TLS packets (from which to infer the remote hostname and whether the hostname is associated with a known advertising and tracking company, based on the Disconnect block list [12]. IoT Network Analyzer presents all these statistics and information via an HTTP REST API that the Privacy Plumber app can access over the local area network. For example, if the computer running IoT Network Analyzer has a local IP address of I_i, then the Privacy Plumber app (on the same local network) can access the traffic information via *http://[*I_i*]/get_traffic*.

Phone Application: App Implementation. The Privacy Plumber mobile app was implemented in Unity using C# and is cross-platform, tested on Android and iPhone. The

[1]Per Huang *et al*. [15], our setup includes $N = 5$ devices. It follows that $N(N+1) = 30$ spoofed ARP packets are sent every two seconds. As each ARP packet has 28 bytes, the overhead is $28 \times 30/2 * 8 = 3,360$ Bits/second or 3.4 Kbps.

(a) View finder (b) Traffic view (c) Controls (d) Education

Fig. 4: Illustration of mobile application design. (a) Device recognition with interactive menu. (b) Live traffic inspection. (c) Rule-based device traffic control (i.e., blocking and unblocking). (d) Educational material on privacy details.

app works by communicating with IoT Network Analyzer via HTTP GET requests, as described in the previous paragraph, to obtain JSON-encoded information about the devices on the network and their traffic. Parsing these JSON objects, the app visualizes the information as charts and text on the AR display (e.g., Figure 4b). The app also shows an interface where users could block an IoT device's traffic, e.g., Figure 4c. Once the user confirms, the app sends the corresponding request to IoT Network Analyzer via the HTTP REST API, and IoT Network Analyzer would subsequently block the device by jamming the device with corrupt ARP packets.

C. User Control of Privacy Leaks from a Phone

With Privacy Plumber we also want to help the user feel more empowered by allowing them to take control of their devices with the ability to block device traffic. Users can manually block or allow device traffic indefinitely, or they can set rules to govern when they want their device to be on or off and for how long (Figure 4c). Users are also given the option to physically power off their device altogether. In this way, Privacy Plumber provides a closed-loop system where users can analyze the information flow out of a given device, then immediately apply direct control over that device in response and receive immediate feedback via the traffic view.

To illustrate how a user might control an IoT device's traffic, let us say that a user feels uncomfortable with an IoT device communicating with the Internet. The user can use the Privacy Plumber app to block Internet access on the device. As shown in Figure 4c, the user can click "Block Traffic" on the app to indefinitely block the device, or specify when to block and unblock the device. Moreover, the app sends an HTTP request to IoT Network Analyzer, using the REST API [2] (where I_i is the IP address of the running instance of IoT Network Analyzer). During the period of blocking, IoT Network Analyzer jams the communication of the device by using a corrupt source MAC address in the spoofed ARP packets (as described in Section III-B). IoT Network Analyzer stops this process at *[unblock_time]*, upon which IoT Network Analyzer sends out spoofed ARP packets without the corrupt source MAC address. This gives users the ability to control the times of day when they want their devices to be on or off.

Privacy Plumber's software-based device blocking offers several advantages over simply turning off or disconnecting a device. First, users do not need physical access to the device; for instance, many surveillance cameras are mounted on ceilings and are difficult to power off. Second, users can temporarily disable a device when they are feeling uncomfortable, e.g., blocking Amazon Echo for an hour during a sensitive phone call or conversation, through Privacy Plumber. Such temporary blocking is difficult to achieve through Echo's app (no such feature) or manually (e.g., the user has to remind themselves to re-connect Echo again). Third, though not currently implemented, Privacy Plumber, with the help of IoT Network Analyzer, can block based on the context (i.e., future work). For example, when IoT Network Analyzer detects the presence of a user's phone on the network (e.g., by checking if the phone responds to periodic ARP requests), IoT Network Analyzer automatically blocks all indoor cameras; when the phone leaves the network (e.g., when the user is out), IoT Network Analyzer could automatically unblock all indoor cameras.

Technical Mechanism for Blocking Devices. A major difference with respect to IoT Inspector's original implementation is

[2] http://[I_i]/block/[device_id]/[block_time]/[unblock_time]

that we have added the capability of blocking devices in IoT Network Analyzer. Using the HTTP REST API [3], the Privacy Plumber app can request IoT Network Analyzer to block a certain device at a particular time (for instance, because the user does not want the device to be communicating to the Internet). Upon receiving this request, IoT Network Analyzer jams the network communication of the device by sending it spoofed ARP packets with corrupt MAC addresses.

To illustrate this process, let us assume that the computer running IoT Network Analyzer has a MAC address M_i and IP address I_i. Let us also assume that IoT Network Analyzer is about to intercept the communication from the gateway (with MAC address M_g and IP address I_g) to a particular device (with MAC M_d and IP I_d) without blocking. To do so, every two seconds, IoT Network Analyzer sends an ARP packet to the device, such that the ARP packet has a source MAC of M_i and a source IP of I_g, along with a destination MAC of M_d and a destination IP of I_d. In contrast, let us assume that IoT Network Analyzer is to block the device. It sends the same ARP packet to the device, except that the ARP packet's source MAC is a series of random numbers (instead of M_i) that represent an unreachable MAC address on the local area network.

D. Visualizing and Understanding Traffic in Real-Time

One of the goals of Privacy Plumber is to show users contextualized network activities of IoT devices to help them pinpoint the potential privacy risks. In this section, we discuss how Privacy Plumber utilizes Augmented Reality to help users contextually visualize their devices' network traffic information in real-time, provide a chart of network traffic in real-time, and provide links to other research in which the privacy concerns of the inspected device have been studied (including home behavior inference, sleeping behaviors, and personal data). Lastly, users are able to send feedback and bug reports.

Use of Augmented Reality. The use of AR visualization makes the interaction with the device the user is inspecting more tangible and contextual. While IoT Inspector [15] and IoT Network Analyzer are text-only data-driven analyzers that can only be accessed using browser HTTP requests, Privacy Plumber is a fully-fledged interactive application due to the utilization of AR. By pointing their camera at the device being inspected, the user can see, in their environment, the traffic coming out of the device that they are physically inspecting. Users can interact with their devices and receive immediate feedback about data output and communication with advertisers. Combined with manual device control, this is intended to help the user feel informed and in control of the IoT devices that physically surround them, similar to the use of a TV remote control.

Learning About Privacy Threats. We aim to educate and inform users on how their IoT devices expose their network traffic information to third parties. In Figures 4d and 5, the app shows icons surrounding the IoT device. When any of these icons are pressed, they provide links to other research materials—which we have manually curated in advance—where the privacy concerns of the inspected device have been studied. Depending on the device, Privacy Plumber provides the following categories of potential privacy violations represented by icons:

- Location: Your physical location either roughly (your address) or fine-grained (room you are in).
- Activity: Your physical activity such as walking, talking, sleeping.
- Screen: Your online activity, such as when you browse videos on YouTube or surf the web.
- Identity: Attributes that can identify you such as your face or voice.
- Shopping: Data on your usage of money or products.
- Health: Can infer different health markers without consent (heart rate, breathing, and others).

E. Privacy Plumber Example Use Cases

In this section, we illustrate two example use cases of the Privacy Plumber app. We focus on the ability of Privacy Plumber to enable experimentation and the usefulness of a real-time inspector. We will describe the users' reactions in Section 4.3.

Example 1: Is Echo Always Listening?

A user may use the Privacy Plumber app to correlate network activities on an Amazon Echo device with the user's interactions—or the lack thereof—with it. While pointing the AR camera at the device, the user could invoke a voice command, such as "Alexa, what is the weather", while observing the device's bandwidth usage graph on the Privacy Plumber app. Afterward, the user may physically press the mute button on Echo, repeat the same voice command, and observe the bandwidth usage graph on the app.

Example 2: What is this App on My Smart Fridge?

Many smart fridges have built-in touch-screen panels. For example, the Samsung Smart Fridge has a tablet-like touch-screen panel to control various settings of the fridge (such as temperature). The panel also allows users to access various built-in apps, such as checking recipes or ordering ingredients online. A user who is concerned with the privacy of such apps may point the AR camera at the fridge, interact with an app, and observe the advertising and tracking services counter on the app. This counter shows, in real-time, the total number of advertising and tracking services that the fridge has communicated with, based on the Disconnect block list [12].

IV. Pilot User Study

To test how users react to Privacy Plumber and inform its future iteration, we conducted a pilot study with 6 participants to experiment with, understand, and control the potential privacy violations of IoT devices. It should be noted that the pilot study would be best conducted in participants' homes. However, due to University research restrictions, the COVID-19 pandemic has made it difficult for us to recruit real users, distribute hardware (e.g., phones powerful enough for AR and Raspberry Pi's for running IoT Network Analyzer), and conduct a free-living study.

[3] http://[I_i]/block/[device_id]/[block_time]

We conducted a one-day controlled lab study in our IoT Lab with 6 participants. Participants were invited to use the Privacy Plumber app while interacting with several IoT devices in the lab, including Samsung Smart Fridge, Amazon Echo, Google Home, Samsung Smart TV, and Google's Nest Cam. Our goal is to assess whether using augmented reality to display network traffic (i.e., by using Privacy Plumber) influenced the participants' awareness of privacy and changed their behaviors.

In the following sections, we present the details of the pilot study and discuss some highlights in the results as well as lessons learned to inform the next iterative of Privacy Plumber.

A. *Participants Recruitment and Demographics*

We recruited 6 graduate students from our institution through our university mailing list. We did so rather than recruiting from a broader population sample because of the constraints our university implemented during the COVID-19 pandemic (i.e., external members were not permitted to enter our buildings). Our sample consisted of four males and two females. Three of the participants were between the ages of 18-24, two participants were between the ages of 25-34, and one participant was between the ages of 35-44.

B. *Study Procedure and Data Collection*

For safety reasons and to implement social distancing procedures, only two people were allowed in the IoT Lab during the study. Aside from the participant, one of the co-authors in this paper served as the research coordinator. They were present throughout the user study to help guide the participants or troubleshoot any technical difficulties that could arise during the study procedure.

Before the study began, each participant filled out a background pre-survey on a computer in the IoT lab. We asked questions about their demographics, how technically savvy they are, their smart device experiences, their general understanding of privacy, and their concerns about their information being exposed to third parties.

After completing the survey, our research coordinator handed each participant a script and an Android mobile phone that had Privacy Plumber installed. Following the script, each participant opened the Privacy Plumber app, kept it running in the foreground, and interacted with one IoT device at a time. Regardless of the IoT device, each interaction consists of the following steps, as prescribed in the script:

1) Using the Privacy Plumber app, the participant scanned the QR code that we had placed on the IoT device. The QR code encodes the device's MAC address, device name, and manufacturer. Based on the QR code, Privacy Plumber shows the corresponding device's AR model on the screen.
2) The participant used the app to inspect the device's traffic, while not doing anything to the device.
3) The participant interacted with the device (which we will describe in detail). During the interactions, the participants observed the network traffic graph on the app.

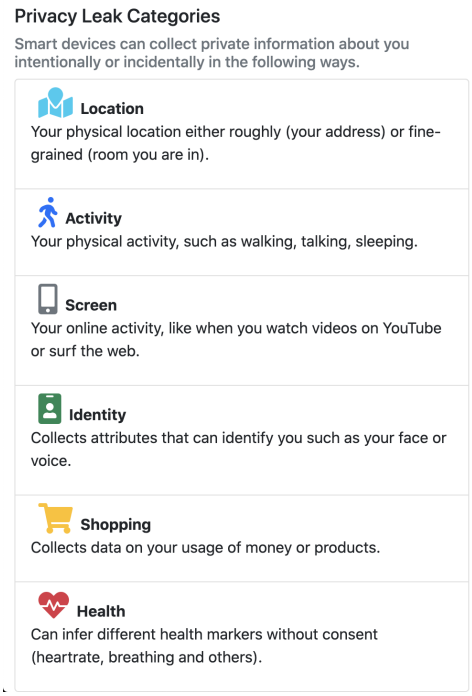

Fig. 5: This screen on the phone application describes the different categories of privacy leaks that different devices have, based on a database that we manually curated in advance.

4) Using the app, the participant clicked on any of the icons surrounding the AR model of the device and read the educational material.

After interacting with all the IoT devices, participants returned the phone to the research coordinator and responded to a post-survey that asked the same questions as in the pre-survey, along with a usability survey. We discuss the results in more depth in Sections IV-C and IV-D. We also include our pre- and post-surveys in the Appendix.

Below, we describe each participant's scripted interactions with each device—i.e., showing Step 3 in detail. During the interactions with the devices, users can access the educational content which is summarized from Mozilla's "privacy not included" handout [29] and academic literature. Each device is described by the categories of privacy exposure they create, those categories are shown in Figure 5.

Samsung Smart Fridge. The fridge has a built-in touchscreen on the door. Through the touchscreen, users have the ability to interact with several built-in apps, such as managing the shopping list, checking what is inside the fridge, and searching for recipes online. Users can also interact with the touchscreen using voice commands, using the trigger word, "Bixby."

Per the script, the research coordinator instructed the participant to follow the following three tasks. (i) Once the participant scanned the QR code of the smart fridge, they said the voice command, "Hey Bixby, do we have mangoes?" Bixby, the fridge's voice assistant, would say "no,". The participant then said, "Hey Bixby, can you add mangos to my shopping list?" Immediately, the participant looked at the

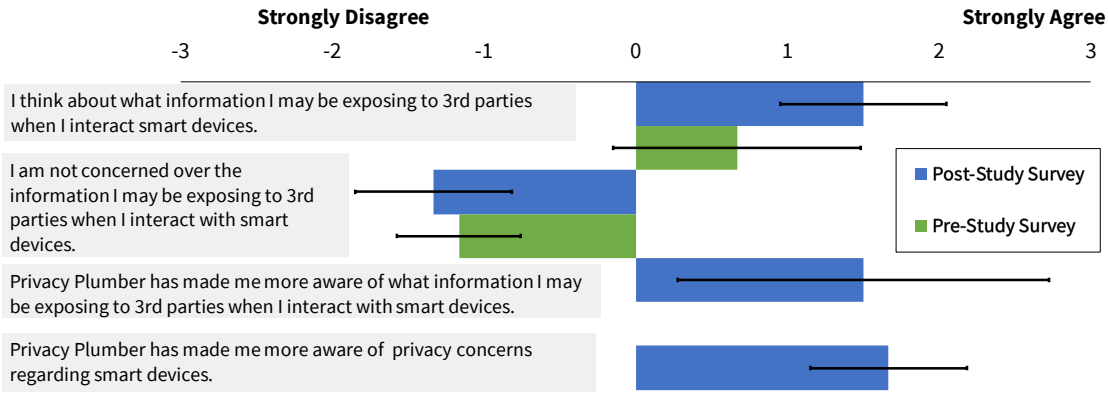

Fig. 6: Representation of participants' average agreement ratings relating to statements about information being exposed to third parties and privacy concerns caused by interacting with IoT devices. Participants rated the first two statements before and after the study, while the last two statements were rated at the end of the study. The results show that after the study, participants displayed an increase in awareness and concern about how their information is being handled when interacting with IoT devices.

Privacy Plumber app and observed the network traffic emitting from the fridge for about 30 seconds. (ii) The participant said, "Hey Bixby, find me a Ramen recipe." The recipe app popped up on the touchscreen. Using the finger, the participant browsed through the available recipes on the screen, while observing the network traffic on Privacy Plumber for 30 seconds. (iii) The participant opened the fridge door and then closed it. Once again, they inspected the fridge's network traffic through the Privacy Plumber app for 30 seconds.

Amazon Echo. Interactions with Echo consists of the following 3 tasks. (i) The participant said the voice command, "Alexa, play a thunderstorm sound." Immediately, the participant observed the network traffic on the app for 30 seconds. (ii) The participant physically pressed the "mute" button on the Echo and watch the device's network traffic for 15 seconds. (iii) The participant said the same voice command as in Task (i) and observed the traffic in the app.

Google Home. The participant said a voice command, "Hey Google, what was the final score in the Super Bowl last year?" The participant immediately started observing the network traffic on the app for 30 seconds.

Samsung Smart TV. The participant used the TV's remote control to navigate to the Bloomberg app on the home screen. They then started streaming a live video on the Bloomberg app for one minute while they observed the network traffic with the Privacy Plumber app.

Nest Cam. Interactions with the camera consists of the following 2 tasks. (i) The participant walked into the field of view of the camera and stay there for five seconds, walked out of the camera's field of view, and observed inspect the network traffic with the Privacy Plumber app. They repeated this task as many times as they liked. (ii) The participant blocked the network traffic to and from the camera using the built-in feature on the Privacy Plumber app. The participant observed the network traffic for 10 seconds, walked in front of the camera's field of view, waited for another ten seconds, and unblocked the device using the Privacy Plumber functionality.

C. Analysis of Pre-Study and Post-Study Surveys

We asked each participant to complete two surveys: (i) a pre-Study Survey that they filled out on a dedicated computer at the beginning of the study, i.e., before the participants interacted with the Privacy Plumber app or the IoT devices; (ii) a post-Study Survey that the participants filled out on the dedicated computer after interacting with all the five IoT devices. We present the results below.

In Figure 6 we present the participants' agreement rating responses for two statements that were asked in the pre-study survey and post-study survey. We observe that for those two statements participants seemed less concerned by how their information is exposed to third parties when they interact with IoT devices before they performed the activities in the study. After participants completed the study, they were more aware and concerned about how their information was disclosed to third parties. The last two statements of Figure 6 were only given in the post-study survey, which asked participants to rate whether Privacy Plumber was useful in helping them become more aware of privacy concerns and how their information is being shared with third parties. On average, participants somewhat agreed that Privacy Plumber helped raise their awareness and privacy concerns. Participants found that Privacy Plumber was helpful in that it helped them visualize what information was being shared.

Additionally, we discuss the results of participants' responses with the IoT devices before and after the study. We show that after the study participants felt less safe with how IoT devices handle their data. Participants were presented with three statements and were asked to rate whether they agree or disagree with these statements on a scale of one to five, where a 1 meant they strongly agree and a 5 represents a strongly disagree rating. Table I demonstrates the average change in attitudes participants had before the study and after the study. We note that before the study, on average participants neither agreed nor disagreed with the statements presented in Table I. After completing the study, the average rating agreement score increased to "somewhat agree" on the last two statements on all IoT devices. The exception was in the first statement, the scores for the Amazon Echo and Google Home. This indicates

Survey Question	Smart Fridge		Amazon Echo		Google Home		Smart TV		Nest Cam	
	pre	*post*	*pre*	*post*	*pre*	*post*	*pre*	*post*	*pre*	*post*
I think this device could be (or is) useful or valuable to my daily life and routine.	3	3.17	2.86	2.5	2.71	2.5	2.43	2.33	2.71	3
I am comfortable having this device in my house and always on.	2.29	3.5	3.86	4.17	3.86	4.17	2.29	3.5	3.43	4.17
I am comfortable having this device in my house if I can automatically control when it is on, or off.	1.29	2.17	2.29	2.5	2	2.33	1.14	2	2.29	2.83
	Strongly Disagree (5) to Strongly Agree (1)									

TABLE I: Results of the survey on user awareness and comfort with smart devices, before and after using Privacy Plumber to inspect those devices. Scores are listed for both pre- and post-study surveys for each device. The higher the scores, the more strongly the participant disagreed with the survey question statement.

that after using Privacy Plumber in the study, participants felt that the Amazon Echo and Google would still find it useful to use in their households.

We also observe that the Smart Fridge, Smart TV, and the Nest cam had the most significant change in attitude. We gathered a few quotes from participants in which they describe how they felt about interacting with these IoT devices and using Privacy Plumber to inspect their network traffic:

> *IoT devices provide more information to third parties than people thought. I think apps like Privacy Plumber can help people to make better decisions when using IoT devices* — **(P1)**

> *Cool to see when and how much traffic each device sends at any given moment!* — **(P5)**

> *I think the app does make me more aware about how the traffic is associated with the behavior of the device. Having some control over the traffic is nice. That being said, if I do have the device in my home, I probably would like to use it, and in that case, I have to allow traffic, which I have no control about what could pass or could not pass. In that sense, I can only accept certain privacy risks.* — **(P2)**

> *It was interesting to see the potential privacy leaks shown next to the device. Some leaks/ privacy implications were surprising. Liked the ability to allow/block traffic, it was also cool to see the real-time traffic including communication with third-party advertisers. Liked the app interface.* —**(P6)**

These quotes, along with results from Figure 6 and Table I, suggest that Privacy Plumber helped participants understand the network traffic, increased their awareness of potential privacy violations, and helped them make more informed decisions on how to handle IoT devices.

D. Analysis of the Usability Survey

At the end of the study, each participant completed the usability survey. Overall, most participants indicated that they would use Privacy Plumber in their home network, found it easy to use and user-friendly, and agreed that most people would learn to use Privacy Plumber quickly. We summarize the results below:

- When asked if they would use the Privacy Plumber mobile app to inspect the data the IoT devices in their homes, two participants said they strongly agreed with the statement and four participants said they somewhat agreed to use Privacy Plumber.
- When participants were asked if they found Privacy Plumber easy to use, four of them somewhat agreed, one participant strongly agreed, and one participant somewhat disagreed.
- When presented the statement "I imagine that most people would learn to use Privacy Plumber very quickly", the responses were across the board spectrum. Three participants rated that comment as strongly agreed, one participant rated the statement with a somewhat agree, one participant responded that they felt neither agreed or disagreed with the statement, and one participant somewhat disagreed.
- When participants were asked to rate the overall user-friendliness's of Privacy Plumber, four participants rated the Privacy Plumber app as good and two participants rated Privacy Plumber as fair.

We gave participants an open-ended question if they would improve the usability of Privacy Plumber, and if so, how. We show their responses in Appendix B. All in all, participants seemed to respond somewhat positively towards Privacy Plumber. It shows that Privacy Plumber may have the potential to be distributed to the general public after further studies. We hope to build off our current platform and implement the suggestions our participants gave us in future work.

E. Performance: System Overhead and Battery Life Impact

Network Overhead. IoT Network Analyzer intercepts the network traffic of select IoT devices via ARP spoofing, a technique that could introduce network overhead especially to the targeted IoT devices. This overhead comes from two sources. First, the spoofed ARP packets consume extra bandwidth, although the overhead is relatively small—i.e., less than 60 Kilobytes/second even if 50 IoT devices are under ARP

spoofing [15]). The second source of overhead comes from the Raspberry Pi 3 Model B, where we run IoT Network Analyzer in the lab. The Raspberry Pi is connected to the lab's network via Ethernet. For all IoT devices to which IoT Network Analyzer sends spoofed ARP packets, all inbound (i.e., download) and outbound (i.e., upload) traffic to and from the IoT devices has to first go through the Raspberry Pi before IoT Network Analyzer forwards the traffic to the targeted device and to the Internet respectively. Effectively, the Raspberry Pi introduces a bottleneck for the ARP-spoofed devices.

To measure the overhead as a result of the Raspberry Pi bottleneck, we conduct the following experiment. We install the Ookla Speed Test app on an Android phone that is connected to the the lab's WiFi network. We have the Ookla app run 15 back-to-back speed tests, which measure the inbound and outbound traffic rates with respect to a server in our city, as well as the latency of packets. Using the same setup, we repeat the same experiment, except that we have IoT Network Analyzer inspect the phone's traffic via ARP spoofing.

We find significant overhead as a result of IoT Network Analyzer. Without ARP spoofing, the app achieves, on average, an inbound rate of 293.6 ± 15.4 Mbps, an outbound rate of 94.1 ± 0.2 Mbps, and a latency of 5.7 ± 0.5 milliseconds. With ARP spoofing by IoT Network Analyzer, the app achieves, on average, an inbound rate of 41.4 ± 74.6 Mbps, an outbound rate of 72.8 ± 14.1 Mbps, and a latency of 5.9 ± 0.5 milliseconds. Compared with the case without ARP spoofing, IoT Network Analyzer reduces the inbound rate by 85.9% and outbound rate by 22.6%, while increasing the latency by 3.5%.

Despite the seemingly significant reduction in bandwidth, we argue that IoT Network Analyzer is unlikely to degrade usability, as the network analyzer is not always running (only when inspecting, or blocking a specific device). Additionally, the overhead can be reduced with improved hardware. According to Netflix, 25 Mbps of inbound rate is sufficient to stream Ultra HD contents [31]. A user who inspects a smart TV using IoT Network Analyzer is likely to enjoy Ultra HD streaming given the reduced inbound rate of 41.4 ± 74.6 Mbps under ARP spoofing. If a user desires to reduce the network overhead, the user could upgrade the computer that runs IoT Network Analyzer, as Raspberry Pi 3 is anecdotally known for its poor networking performance [37], [38]. Possible upgrade option could include a computer—or ODroid if the user needs the compact form factor [14]—that is shipped with a fast CPU and a Gigabit Ethernet card.

Battery Lifetime. We used AccuBattery on android, to try to understand the energy cost. This does not hold across phones, so we compare the energy cost against YouTube and TikTok for ten minutes of streaming video. With all the background application killed, 10 minutes of Privacy Plumber impacts 3.98% (159mAh) of the battery lifetime, while YouTube costs 2.63% (105mAh) and TikTok costs 3.9% (156mAh). Privacy Plumber is only meant for point inspection and short usage to analyze new devices in the home, or experiment with different setups, so it should not impact battery lifetime too much since it is not always on. Moreover, the battery lifetime cost is similar to that of streaming videos online, a normal function, therefore users should not expect significant battery lifetime loss due to usage of Privacy Plumber.

V. Discussion on Limitations and Future Work

Comparing users' mental models against actual contents of IoT network traffic. Our results show that users' mental model of how IoT devices communicate with the Internet may be inconsistent with how devices *appear* to behave, but it is unclear whether this mental model is consistent with the *actual* contents of the communication. For example, two participants in our study did not expect network traffic from Amazon Echo when the device's microphone was on mute. Presumably, the participants expected Amazon not to send any audio data back to Amazon during mute. In this case, Echo's *apparent* behavior was the communication with the Internet on mute; in contrast, whether Echo *actually* sent out audio data was unknown. Our system did not extract the contents of the communication, which could be encrypted based on previous results [4].

Despite the encrypted contents, man-in-the-middling is possible (e.g., per Moghaddam *et al.* [28]). In future in-lab studies, we plan to modify IoT Network Analyzer to intercept and decrypt IoT traffic, assuming that devices do not validate certificates and/or do not use certificate pinning. We hope to extract the payload from some of the TLS connections, identify exactly what devices are sending to the Internet, and compare it against users' mental models.

Automated, contextualized blocking of devices. The current prototype allows users to set a block/unblock schedule for IoT devices. Although this feature provides users with fine-grained control, it requires manual effort from the user both in setting what devices to block and when to block.

We plan to augment this feature with automated device blocking based on contextualized information that IoT Network Analyzer already collects. For example, a user could create a rule on IoT Network Analyzer that would automatically block surveillance cameras if IoT Network Analyzer detects the presence of mobile phones (based on ARP and pings) in the home network (which could suggest that the residents are home); otherwise, it can unblock the cameras to capture, say, unauthorized entry into the property. As another example, let's say a user has an Amazon Echo and a smart TV in the living room. The user could create another rule that lets IoT Network Analyzer automatically block Amazon Echo if it detects active streaming traffic from the smart TV, as the user may not want Echo to capture any conversations while the family is watching TV in the living room. In short, by leveraging the IoT traffic that IoT Network Analyzer already collects, users could create automated, contextualized rules to block IoT devices from collecting sensitive data.

Deployment roadmap and challenges. We plan to deploy the Privacy Plumber app and IoT Network Analyzer to real-world users at scale. Based on our current prototype, we plan to make the following modifications.

Operating system support. Once deployed, our system will have the same two-component architecture, although we will expand the Privacy Plumber app to both iOS and Android (current prototype), and IoT Network Analyzer to all major non-mobile operating systems including macOS, Windows,

and Linux (current prototype). This process will likely be straightforward, as we developed both components with cross-OS platforms (Unity for the app and pure Python for IoT Network Analyzer).

Network-based device identification. We will develop network-based device identification mechanisms to help users distinguish among their devices and identify the device(s) of interest. The current prototype identifies devices based on a hard-coded mapping between MAC OUIs and device names, because we already know the inventory of IoT devices in the lab. For real-world deployment, we will incorporate IoT Inspector's device identification algorithm [15], so that our system will dynamically infer device names based on the network signature, which includes not only OUIs, but also DNS queries, UPnP banners, mDNS names, and DHCP hostnames. We will also use information in the 802.11 frames to discover and locate devices [41].

Image-based device identification. To complement the network-based approach, we will also develop image-based device identification mechanisms for the AR camera. Currently, the Privacy Plumber app identifies devices based on printed QR codes on or near select IoT devices, such that the QR codes encode the MAC addresses and the names of devices. For real-world deployment, we will use computer vision to train a model of common IoT device types, such as voice assistants, smart TVs, and surveillance cameras (where security and privacy issues are commonly found in the literature). This model will help the AR app recognize *possible* IoT devices (e.g., "likely a smart TV"). The app will then refine the recognition with the network-based device identification algorithm (e.g., "whether the device is indeed a smart TV based on the network signatures") and manual user input if necessary. Both the network- and image-based approaches will hopefully help the app identify IoT devices in real-world settings.

Expanded user study. The user study, as a pilot, has a small sample size and is limited to graduate students, who may be more inquisitive or technically-inclined than the general population. We hope to scale out the testing to a larger userbase, both in lab and in real homes, in future work. We will also compare the participants' changes in privacy awareness against other visualization tools (e.g., IoT Inspector [15] and Aretha [40]). Finally, we will conduct in-depth studies on various ways to visualize privacy leaks in AR (e.g., icon overlays and animations).

VI. Summary

This paper presented Privacy Plumber, an end-to-end system demonstrating how a general population of end users can potentially have insight into the network traffic of smart home IoT devices, and how these users can control when these smart devices could communicate with the Internet with one click of a button. Designed after the concept of a leak detector, Privacy Plumber is a phone app with a tethered desktop application—IoT Network Analyzer—that provides an inspect and correct interface supported by network traffic analysis (inspect) and automated and timed network traffic jamming (correct).

Privacy Plumber is the first real-world inspection and control system that can be deployed in any home without new hardware or router modifications. Using AR, the tool aims to help users model IoT device activities within the context of the physical environment and of user interactions (addressing challenges C1 and C3, per Section II-D); it gives users the option to block IoT devices and control the privacy "valve" (C2); it provides users with an interface to visualize IoT device activities as users interact with devices (C4); and it requires a modern AR-supported phone and computer, without any dedicated or specialized hardware (C5).

We evaluated Privacy Plumber inside an instrumented smart home space with a variety of devices not previously evaluated for any privacy-enhancing tool, including a smart fridge, a smart TV, voice assistants, and Internet-connected surveillance cameras. We found that using Privacy Plumber improved users' awareness of potential privacy violations of devices and that the system was generally easy to use and afforded useful controls. In the future, we hope tools like Privacy Plumber will give mechanisms back to the user for stymieing the flow of private information outside the home, especially as our homes and living spaces become smarter, often without our consent.

Acknowledgment

This research is based upon work supported by the National Science Foundation under award numbers CNS-2219867, CNS-1739809, and CNS-1915847. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The research is also based on work supported by gifts from Consumer Reports and Meta.

References

[1] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and Selcuk Uluagac. Peek-a-boo: I see your smart home activities, even encrypted! In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 207–218, 2020.

[2] Imtiaz Ahmad, Rosta Farzan, Apu Kapadia, and Adam J Lee. Tangible privacy: Towards user-centric sensor designs for bystander privacy. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–28, 2020.

[3] Omar Alrawi, Chaz Lever, Manos Antonakakis, and Fabian Monrose. Sok: Security evaluation of home-based iot deployments. In *2019 IEEE symposium on security and privacy (sp)*, pages 1362–1380. IEEE, 2019.

[4] Noah Apthorpe, Danny Yuxing Huang, Dillon Reisman, Arvind Narayanan, and Nick Feamster. Keeping the smart home private with smart (er) iot traffic shaping. *Proceedings on Privacy Enhancing Technologies*, 2019(3):128–148, 2019.

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

[6] Noah Apthorpe, Dillon Reisman, Srikanth Sundaresan, Arvind Narayanan, and Nick Feamster. Spying on the smart home: Privacy attacks and defenses on encrypted iot traffic. *arXiv preprint arXiv:1708.05044*, 2017.

[7] Bruhadeshwar Bezawada, Maalvika Bachani, Jordan Peterson, Hossein Shirazi, Indrakshi Ray, and Indrajit Ray. Iotsense: Behavioral fingerprinting of iot devices. *arXiv preprint arXiv:1804.03852*, 2018.

[8] Patrick Bombik, Tom Wenzel, Jens Grossklags, and Sameer Patil. A multi-region investigation of the perceptions and use of smart home devices. *Proceedings on Privacy Enhancing Technologies*, 3:6–32, 2022.

[9] Nico Castelli, Corinna Ogonowski, Timo Jakobi, Martin Stein, Gunnar Stevens, and Volker Wulf. What happened in my home? an end-user development approach for smart home data visualization. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 853–866, 2017.

[10] Gordon Chu, Noah Apthorpe, and Nick Feamster. Security and privacy analyses of internet of things children's toys. *IEEE Internet of Things Journal*, 6(1):978–985, 2018.

[11] Sunny Consolvo, Jaeyeon Jung, Ben Greenstein, Pauline Powledge, Gabriel Maganis, and Daniel Avrahami. The wi-fi privacy ticker: improving awareness & control of personal information exposure on wi-fi. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 321–330, 2010.

[12] Disconnect, Inc. Disconnect tracking protection, 2021.

[13] Yuanyuan Feng, Yaxing Yao, and Norman Sadeh. A design space for privacy choices: Towards meaningful privacy control in the internet of things. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2021.

[14] HardKernel. ODROID-XU4, 2021.

[15] Danny Yuxing Huang, Noah Apthorpe, Frank Li, Gunes Acar, and Nick Feamster. Iot inspector: Crowdsourcing labeled network traffic from smart home devices at scale. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2):1–21, 2020.

[16] Haojian Jin, Boyuan Guo, Rituparna Roychoudhury, Yaxing Yao, Swarun Kumar, Yuvraj Agarwal, and Jason I Hong. Exploring the needs of users for supporting privacy-protective behaviors in smart homes. In *CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2022.

[17] Patrick Gage Kelley, Joanna Bresee, Lorrie Faith Cranor, and Robert W Reeder. A" nutrition label" for privacy. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, pages 1–12, 2009.

[18] Christian Kreibich, Nicholas Weaver, Boris Nechaev, and Vern Paxson. Netalyzr: Illuminating the edge network. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 246–259, 2010.

[19] Hosub Lee and Alfred Kobsa. Understanding user privacy in internet of things environments. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 407–412. IEEE, 2016.

[20] Huichen Lin and Neil W Bergmann. Iot privacy and security challenges for smart home environments. *Information*, 7(3):44, 2016.

[21] Heather Richter Lipford, Madiha Tabassum, Paritosh Bahirat, Yaxing Yao, and Bart P Knijnenburg. Privacy and the internet of things. *Modern Socio-Technical Perspectives on Privacy*, page 233, 2022.

[22] Nathan Malkin, Julia Bernd, Maritza Johnson, and Serge Egelman. "what can't data be used for?" privacy expectations about smart tvs in the us. In *Proceedings of the 3rd European Workshop on Usable Security (EuroUSEC), London, UK*, 2018.

[23] Nathan Malkin, Joe Deatrick, Allen Tong, Primal Wijesekera, Serge Egelman, and David Wagner. Privacy attitudes of smart speaker users. *Proceedings on Privacy Enhancing Technologies*, 2019(4), 2019.

[24] Shrirang Mare, Franziska Roesner, and Tadayoshi Kohno. Smart devices in airbnbs: Considering privacy and security for both guests and hosts. *Proc. Priv. Enhancing Technol.*, 2020(2):436–458, 2020.

[25] Emily McReynolds, Sarah Hubbard, Timothy Lau, Aditya Saraf, Maya Cakmak, and Franziska Roesner. Toys that listen: A study of parents, children, and internet-connected toys. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 5197–5207, 2017.

[26] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. Iot sentinel: Automated device-type identification for security enforcement in iot. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2177–2184. IEEE, 2017.

[27] Phoebe Moh, Noel Warford, Pubali Datta, Nathan Malkin, Adam Bates, and Michelle L Mazurek. Characterizing misuse and snooping in home iot devices.

[28] Hooman Mohajeri Moghaddam, Gunes Acar, Ben Burgess, Arunesh Mathur, Danny Yuxing Huang, Nick Feamster, Edward W Felten, Prateek Mittal, and Arvind Narayanan. Watching you watch: The tracking ecosystem of over-the-top tv streaming devices. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 131–147, 2019.

[29] Mozilla. Privacy Not Included., 2021.

[30] Pardis Emami Naeini, Sruti Bhagavatula, Hana Habib, Martin Degeling, Lujo Bauer, Lorrie Faith Cranor, and Norman Sadeh. Privacy expectations and preferences in an {IoT} world. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 399–412, 2017.

[31] Netflix. Internet Connection Speed Recommendations, 2021.

[32] Helen Nissenbaum. *Privacy in context: Technology, policy, and the integrity of social life*. Stanford University Press, 2009.

[33] TJ OConnor, Reham Mohamed, Markus Miettinen, William Enck, Bradley Reaves, and Ahmad-Reza Sadeghi. Homesnitch: behavior transparency and control for smart home iot devices. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 128–138, 2019.

[34] pfSense.org. World's MOST Trusted Open Source Firewall, 2021.

[35] James Pierce, Richmond Y Wong, and Nick Merrill. Sensor illumination: Exploring design qualities and ethical implications of smart cameras and image/video analytics. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2020.

[36] Sarah Prange, Ahmed Shams, Robin Piening, Yomna Abdelrahman, and Florian Alt. Priview– exploring visualisations to support users' privacy awareness. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[37] Raspberry Pi Discussion Forum. RPi 3B+ gigabit ethernet bad download speeds, 2018.

[38] Raspberry Pi Dramble. Networking Benchmarks, 2021.

[39] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, and Phillipa Gill. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. 2018.

[40] William Seymour, Martin J Kraemer, Reuben Binns, and Max Van Kleek. Informing the design of privacy-empowering tools for the connected home. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2020.

[41] Rahul Anand Sharma, Elahe Soltanaghaei, Anthony Rowe, and Vyas Sekar. Lumos: Identifying and localizing diverse hidden IoT devices in an unfamiliar environment. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1095–1112, Boston, MA, August 2022. USENIX Association.

[42] Yun Shen and Pierre-Antoine Vervier. Iot security and privacy labels. In *Annual Privacy Forum*, pages 136–147. Springer, 2019.

[43] Madiha Tabassum, Tomasz Kosinski, and Heather Richter Lipford. " i don't own the data": End user perceptions of smart home device data practices and risks. In *Fifteenth Symposium on Usable Privacy and Security ({SOUPS} 2019)*, 2019.

[44] Parth Kirankumar Thakkar, Shijing He, Shiyu Xu, Danny Yuxing Huang, and Yaxing Yao. "it would probably turn into a social faux-pas": Users' and bystanders' preferences of privacy awareness mechanisms in smart homes. In *CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2022.

[45] Blase Ur, Pedro Giovanni Leon, Lorrie Faith Cranor, Richard Shay, and Yang Wang. Smart, useful, scary, creepy: perceptions of online behavioral advertising. In *proceedings of the eighth symposium on usable privacy and security*, pages 1–15, 2012.

[46] Max Van Kleek, Reuben Binns, Jun Zhao, Adam Slack, Sauyon Lee, Dean Ottewell, and Nigel Shadbolt. X-ray refine: Supporting the exploration and refinement of information exposure resulting from smartphone apps. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2018.

[47] Max Van Kleek, Ilaria Liccardi, Reuben Binns, Jun Zhao, Daniel J Weitzner, and Nigel Shadbolt. Better the devil you know: Exposing the data sharing practices of smartphone apps. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 5208–5220, 2017.

[48] Sean Whalen. An introduction to arp spoofing. *Node99 [Online Document]*, 2001.

[49] Peter Worthy, Ben Matthews, and Stephen Viller. Trust me: doubts and concerns living with the internet of things. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, pages 427–434, 2016.

[50] Yaxing Yao, Justin Reed Basdeo, Smirity Kaushik, and Yang Wang. Defending my castle: A co-design study of privacy mechanisms for smart homes. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–12, 2019.

[51] Eric Zeng, Shrirang Mare, and Franziska Roesner. End user security and privacy concerns with smart homes. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 65–80, 2017.

[52] Wei Zhang, Yan Meng, Yugeng Liu, Xiaokuan Zhang, Yinqian Zhang, and Haojin Zhu. Homonit: Monitoring smart home apps from encrypted traffic. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1074–1088, 2018.

[53] Serena Zheng, Noah Apthorpe, Marshini Chetty, and Nick Feamster. User perceptions of smart home iot privacy. *Proceedings of the ACM on human-computer interaction*, 2(CSCW):1–20, 2018.

APPENDIX

SURVEY QUESTIONS

All questions require responses in Likert scales, ranging from "Strongly Agree" (1) to "Strongly Disagree" (5).

A. Pre-Study Survey Questions

1) When I am in a smart home, I think about what information I may be exposing to vendors, companies, and 3rd parties when I interact with or sit in the same space with smart devices in the home.
2) I am not concerned about the information I may be exposing to 3rd parties when I interact with or sit in the same space as smart devices in a smart home.
3) I think this device could be (or is) useful or valuable to my daily life and routine.
   - Smart Fridge
   - Google Home
   - Amazon Echo
   - Smart TV
   - Nest Cam
4) I am comfortable having this device in my house and always on.
   - Smart Fridge
   - Google Home
   - Amazon Echo
   - Smart TV
   - Nest Cam

B. Post-Study Survey Questions

1) When I am in a smart home, I think about what information I may be exposing to vendors, companies, and 3rd parties when I interact with or sit in the same space with smart devices in the home.
2) I am not concerned about the information I may be exposing to 3rd parties when I interact with or sit in the same space as smart devices in a smart home.
3) Privacy Plumber has made me more aware of what information I may be exposing to 3rd parties when I interact with smart devices in the home.
4) I feel Privacy Plumber has made me more aware of privacy and security concerns surrounding IoT devices.
5) I think this device could be (or is) useful or valuable to my daily life and routine.
   - Smart Fridge
   - Google Home
   - Amazon Echo
   - Smart TV
   - Nest Cam
6) I am comfortable having this device in my house and always on.
   - Smart Fridge
   - Google Home
   - Amazon Echo
   - Smart TV
   - Nest Cam
7) Finally, please provide any other thoughts or observations from participating in this experiment with Privacy Plumber (open ended).

ADDITIONAL RESPONSES FROM THE USABILITY SURVEY

We gave participants an open-ended question if they would improve the usability if privacy plumber, if so how. We obtained the following responses from each participant.

I would include more guidance or instructions in the app for first-time users. **(P1)**

I think the app is generally easy-to-use, although I might want more functionalities in the app. There are certain latencies in the app, which can be annoying. It would be more helpful if I can know if the device is not sending any traffic, or it is just simply late (e.g., adding a loading icon). **(P2)**

Make it possible to view past trends (a la net microscope) and scroll backwards in time, so I can get the context of how much traffic is regularly sent. Give me a global view of the worst offenders. Still some work to do on basic stability. It only works on devices that people have obviously ALREADY DECIDED TO BUY, which is a weird sample. Obviously, I don't have QR codes printed out on all of my household electronics. **(P3)**

I had difficulties trying to access the buttons, and the images seemed lagged a little. But the info was very useful overall. **(P4)**

Fix where the traffic and 'learn more about the device' buttons once you've scanned the QR code. It's a bit awkward to have to hold the phone back up to the device. Maybe add the units (byte/kB) to the left hand side of the graph instead of above it for the traffic visualization. **(P5)**

The plots are not super-intuitive but I liked the representations in terms of text/pictures which is easier to comprehend. I would also be interested to see what advertisers the information is being leaked to. While the AR thing is cool, I would also like the option to just scroll through a list of devices. That ways I do not have to be close to the device and would also be able to monitor its activity when I am not close to the device. In fact, I would be interested in seeing the device communication (including interaction w/ advertisers) in that case. **(P6)**